IBM SPSS Modeler 14.2 アルゴリズム ガイド

注 ： サポートされている情報および製品をご利用いただく前に、「注意事項」(p.346)の一般情報をお読みください。

本マニュアルには、SPSS Inc., an IBM Company が所有する情報が含まれています。これらの情報は使用許諾契約書に基づいて提供され、著作権法によって保護されています。本文書に記載されている情報には、製品の保証は含まれていません。また本マニュアルに記載されている文は製品の保証を規定しないものとします。

IBM または SPSS に情報を送信すると、あなたに対する義務を負うことなく、適切とする方法でその情報を使用または配布する非独占的権利と IBM および SPSS 付与するものとします。

はじめに

IBM® SPSS® Modeler は、IBM Corp. が開発した企業強化用のデータ マイニング ワークベンチです。SPSS Modeler を使用すると、企業はデータを詳しく調べることにより顧客および一般市民とのリレーションシップを強化することができます。企業は、SPSS Modeler を使って得られた情報に基づいて利益をもたらす顧客を獲得し、抱き合わせ販売の機会を見つけ、新規顧客を引き付け、不正を発見し、リスクを減少させ、政府機関へのサービスの提供を改善することができます。

SPSS Modeler の視覚的インターフェイスを使用すると、特定ビジネスの専門知識を適用し、より強力な予測モデルを実現し、解決までの時間を短縮します。SPSS Modeler では、予測、分類、セグメント化、および関連性検出アルゴリズムなど、さまざまなモデル作成手法を提供しています。モデルを作成した後は、IBM® SPSS® Modeler Solution Publisher により、企業全体の意思決定者やデータベースにモデルを配布することが可能になります。

IBM Business Analytics について

IBM Business Analytics ソフトウェアは、意思決定者がビジネス パフォーマンスを向上させるために信頼する完全で、一貫した正確な情報を提供します。ビジネス インテリジェンス、予測分析、財務実績および戦略管理、および分析アプリケーションの包括的なポートフォリオを利用することによって、現在の実績を明確、迅速に理解し、将来の結果を予測することができます。豊富な業界のソリューション、実績ある実例、専門サービスと組み合わせ、さまざまな規模の組織が、高い生産性を実現、意思決定を自信を持って自動化し、より良い決定をもたらします。

このポートフォリオの一部として、IBM SPSS Predictive Analytics ソフトウェアを使用する組織は、将来のイベントを予測し、その洞察に基づいて積極的に行動し、より優れた業績を実現することができます。全世界の企業、政府、学術分野のお客様が IBM SPSS の技術を活用し、不正行為を減少させ、リスクを軽減させながら、顧客の獲得、保持、成長において、競争優位を高めることができます。IBM SPSS ソフトウェアを日々の業務に取り入れることによって、組織は業務目標を達成し、大きな競争的優位を獲得することができるよう、意思決定を方向付け、自動化することができるようになります。お問い合わせは、http://www.ibm.com/spss を参照してください。

テクニカル サポート

お客様はテクニカル サポートをご利用いただけます。IBM Corp. 製品の使用方法、または対応するハードウェア環境へのインストールについてサポートが必要な場合は、テクニカル サポートにご連絡ください。テクニカル サポートの詳細は、IBM Corp. Web ページ http://www.ibm.com/support を参照してください。ご本人、組織、サポートの同意を確認できるものをご用意ください。

内容

章

1

調整済み傾向アルゴリズム

調整済み傾向スコアは、モデル構築のプロセスの一部として計算され、そのほかでは使用できません。モデルが構築されると、テスト データ区分または検証データ区分からのデータを使用してスコアリングされます。調整済み傾向スコアを算出する新しいモデルは、そのデータ区分に対する元のモデルの精度を分析することによって構築されます。モデルの種類によって、2 つの手法のうちいずれかが調整済み傾向スコアの計算に使用されます。

モデル ベースの方法

ルール セットおよびツリー モデルについて、次の方法が使用されます。

1. テスト データ区分または検証データ区分でモデルをスコアリングします。
2. **ツリー モデル:**ツリー ノードにスコアリングされたレコードの対象値の分布を反映し、テスト/検証データ区分を使用して各ツリー ノードで各カテゴリの度数を計算します。

   ルール セット モデル:テスト/検証データ区分のモデルのパフォーマンスを反映し、テスト/検証データ区分を使用して、各ルールのサポートおよび各進度を計算します。

   新しいルール セットまたはツリー モデル内の結果は元のモデルとともに格納されます。元のモデルが新しいデータに適用されるたびに、新しいモデルが調整のない傾向スコアに適用されて調整済みスコアが生成されます。

一般的な目的の方法

他のモデルについては、次の方法が使用されます。

1. テスト データ区分または検証データ区分でモデルをスコアリングし、予測値および予測された調整のない傾向を計算します。
2. 予測値または観測値の欠損値があるすべてのレコードを削除します。
3. 真の観測値を 1、偽の観測値を 0 で観測傾向を計算します。
4. 100 の度数が等しいタイルを使用し、予測された調整のない傾向に従って、レコードを分割します。
5. 各分割の予測された調整のない傾向の平均値、および観測傾向の平均値を計算します。
6. 観測傾向の平均値を対象値に、予測された調整のない傾向を予測値に、ニューラル ネットワークを作成します。ニューラル ネットワークは、次のように設定します。

   Use a random seed, value 0
   Use the "quick" training method
   Stop after 250 cycles

Do not use prevent overtaining option
Use expert mode
Quick Method Expert Options:
  Use one hidden layer with 3 neurons and persistence set to 200
Learning Rates Expert Options:
  Alpha 0.9
  Initial Eta 0.3
  High Eta 0.1
  Eta decay 50
  Low Eta 0.01

結果として、ニューラル ネットワーク モデルは、調整のない傾向を、テスト データ区分または検証データ区分で元のモデルのパフォーマンスを考慮に入れるより正確な推定に対応付けようとします。スコアリング時に調整済み傾向を計算するには、このニューラル ネットワークを元のモデルのスコアリングから取得した調整のない傾向に適用します。

章

2

異常値検出のアルゴリズム

概要

異常値検出プロシージャは、クラスタ グループの平均からの偏差に基づいて、例外的なケースを検索します。このプロシージャは、任意の推論的データ分析に先立つ予備的なデータ分析ステップで、データ監査の目的で例外的なケースを素早く検出するために設計されました。このアルゴリズムは一般的な異常値検出のために設計されています。つまり、異常ケースの定義は、異常値の定義が適切にできる、医療保険業界での普通でない支払いパターンの検出や金融業界での不正資金浄化（マネーローンダリング）のような特定のアプリケーションに特有なものではありません。

一次計算

表記

この章では特に明記しない限り、次の表記を使用します。

ID	データ ファイル内の各ケースの識別変数。
n	学習データ X_{train} 内のケースの数。
X_{ok}, k = 1, …, K	学習データ内の入力変数のセット。
M_k, k ∈ {1, …, K}	$_{ok}$ が連続型変数の場合、M_k は、変数の総平均つまり学習データ全体の変数の平均を表します。
SD_k, k ∈ {1, …, K}	$_{ok}$ が連続型変数の場合、M_k は、総標準偏差つまり学習データ全体の標準偏差を表します。
X_{K+1}	分析で作成される連続型変数。各ケースに欠損値がある変数（k = 1, …, K）のパーセンテージを表します。
X_k, k = 1, …, K	欠損値の処理が適用された後に処理される入力変数のセット。 詳細は、 p.4 モデリング段階 を参照してください。
H、または H の境界：$[H_{min}, H_{max}]$	H は、事前に定義された作成するクラスタ グループの数です。または、境界 $[H_{min}, H_{max}]$ は、クラスタ グループの最小および最大数を指定するのに使用できます。
n_h, h = 1, …, H	学習データに基づいた、クラスタ h、h = 1、…、H 内のケースの数。
p_h, h = 1, …, H	学習データに基づいた、クラスタ h、h = 1、…、H 内のケースの割合。各h に対し、$p_h = n_h/n$ です。
M_{hk}, k = 1, …, K+1, h = 1, …, H	X_k が連続型変数の場合、M_{hk} は、クラスタ平均、つまり学習データに基づいたクラスタ内の変数の平均を表します。X_k がカテゴリ型変数の場合、その変数はクラスタ モードを表します。つまり、学習データに基づいたクラスタ内の変数の、最も一般的なカテゴリ値を表します。
SD_{hk}, k ∈ {1, …, K+1}, h = 1, …, H	X_k が連続型変数の場合、SD_{hk} は、クラスタ標準偏差、つまり学習データに基づいたクラスタ内の変数の標準偏差を表します。

$\{n_{hkj}\}$, k ∈ {1, …, K}, h = 1, …, H, j = 1, …, J_k	度数セット $\{n_{hkj}\}$ は、X_k がカテゴリ型変数のときにのみ定義されます。X_k に J_k カテゴリがある場合は、n_{hkj} がカテゴリ j に分類されるクラスタ h 内のケースの数です。
m	連続型変数とカテゴリ型変数との間の影響のバランスをとるために使用される調整用の重み。正数で、デフォルトは 6 です。
VDI_k, k = 1, …, K+1	ケースの変数偏差の指標は、変数 X_k のクラスタ基準値からの偏差の測定値です。
GDI	ケースのグループ偏差指標 GDI は、対数-尤度距離 d(h, s) です。これは、変数偏差指標 $\{VDI_k, k = 1, …, K+1\}$ の、すべての合計です。
異常値指標（インデックス）	ケースの異常値指標は、ケースが属するクラスタ グループの平均 GDI に対する GDI の比率です。
変数寄与率の測定値	ケースに対する変数 X_k の変数寄与率の測定値は、ケースの対応する GDI に対する VDI_k の比率です。
$pct_{anomaly}$ または $n_{anomaly}$	事前定義の値 $pct_{anomaly}$ で、異常値と見なされるケースのパーセンテージを決定します。または、事前定義の整数値 $n_{anomaly}$ で、異常値と見なされるケースの数を決定します。
$cutpoint_{anomaly}$	事前に指定されている分割点で、$cutpoint_{anomaly}$ より大きな異常値指標値のケースは、異常値と見なされます。
$k_{anomaly}$	事前に指定された整数のしきい値 $1 \le k_{anomaly} \le K+1$ で、ケースが異常であると識別される理由として考えられる変数の数を決定します。

アルゴリズムのステップ

このアルゴリズムは、次の 3 段階に分けられます。

モデリング: 入力変数のセットの類似性に基づき、ケースがクラスタ グループ内に配置されます。ケースのクラスタ グループを決定するのに使用されるクラスタリング モデルと、クラスタ グループの平均値を計算するのに使用される十分な統計量が格納されます。

スコアリング:このモデルはクラスタ グループを識別するために各ケースへ適用され、所属するクラスタ グループに関連してケースの異常性を測定するために、いくつかの指標が各ケース用に作成されます。すべてのケースは、異常値指標の値でソートされます。ケース リストの上位部分が異常値のセットとして識別されます。

理由:各異常ケースについて、変数が対応する変数の偏差指標でソートされます。上位の変数、その値、および対応する平均値が、ケースが異常と識別された理由として表示されます。

モデリング段階

この段階で、次のタスクが実行されます。

1. **学習セットの形成**: 指定された変数とケースで始め、連続型変数で極端に大きな値 (1.0E+150 より大きい) のケースを削除します。欠損値の処理が実行されていない場合は、変数に欠損値のあるケースも削除します。すべて定数の非欠損値またはすべて欠損値の変数を削除します。残りのケースと変数が、異常値検出モデルの作成に使用

されます。プロシージャによるピボット テーブルへの統計出力はこの学習セットに基づきますが、データセットへ保存される変数はすべてのケース用に計算されます。

2. **欠損値の処理 (任意選択):** 各入力変数 X_{ok}, k = 1, …, K について、X_{ok} が連続型変数の場合は、総平均 M_k と総標準偏差 SD_k を計算するためにその変数のすべて有効な値を使用します。変数の欠損値を総平均で置き換えます。X_{ok} がカテゴリ型変数の場合、すべての欠損値を「欠損値」カテゴリに結合します。このカテゴリは、有効なカテゴリとして処理されます。{X_{ok}} の処理された形式を {X_k} で示します。

3. **欠損値 Pct 変数の作成 (任意選択):** 新しい連続型の変数 X_{K+1} は、各ケース内の欠損値のある変数（連続型とカテゴリ型の両方）のパーセンテージを示します。

4. **クラスタ グループの識別:** 処理される入力変数 {X_k, k = 1, …, K+1} は、クラスタリング モデルの作成に使用されます。2 ステップのクラスタリング アルゴリズムが、ノイズ処理をオンにされないで使用されます（詳細は TwoStep クラスタ アルゴリズム ドキュメントを参照）。

5. **十分な統計量ストレージ:**クラスタ モデルとクラスタによる変数の十分な統計結果は、スコアリング段階向けに格納されます。
   - 各連続型変数の総平均 M_k および標準偏差 SD_k は、k ∈ {1, …, K+1} に格納されます。
   - 各クラスタに対し、h = 1, …, H, でサイズ n_h を使用します。.X_k が連続型変数の場合、クラスタ h 内のケースに基づいて、変数のクラスタ平均 M_{hk} と標準偏差 SD_{hk} を格納します。X_k がカテゴリ型変数の場合、クラスタ h 内のケースに基づいて、変数の各カテゴリ j の 度数 n_{hkj} を格納します。 また、モーダルなカテゴリ M_{hk} を格納します。.これら十分な統計量が、クラスタとケース s 間の対数-尤度距離 d(h, s) の計算に使用されます。

スコアリング段階

この段階で、スコアリング（検定中または学習中）のデータに対し、次のタスクが実行されます。

1. **新規有効カテゴリのスクリーニング:**スコアリング データには、学習データ内に入力変数 {X_{ok}, k = 1, …, K} が含まれている必要があります。さらに、スコアリング データの変数の形式は、モデリング段階で学習データ ファイル内にあったデータと同じである必要があります。

   スコアリング データ内のケースは、学習データ内に出現しない有効なカテゴリのカテゴリ型変数を含んでいる場合は排除されます。たとえば、Region が学習データ内でカテゴリ IL、MA、CA を含むカテゴリ型変数の場合、Region に有効なカテゴリ FL を含むスコアリング データ内のケースは、分析から排除されます。

2. **欠損値の処理 (任意選択):** 各入力変数 X_{ok} について、X_{ok} が連続型変数の場合は、総平均 M_k と総標準偏差 SD_k を計算するためにその変数のすべて有効な値を使用します。変数の欠損値を総平均で置き換えます。X_{ok} がカテゴリ型変数の場合、すべての欠損値を欠損値のカテゴリに入れます。このカテゴリは、有効なカテゴリとして処理されます。

3. **欠損値 Pct 変数の作成 (モデリング段階により任意選択):** X_{K+1} がモデリング段階で作成される場合は、スコアリング データも計算されます。

4. **各ケースを最も近い非ノイズ クラスタへ割り当てる:**モデリング段階のクラスタリング モデルは、各ケースのクラスタ ID を作成するために、スコアリング データ ファイルの処理済み変数に適用されます。ノイズ クラスタへ属するケースは、最も近くにある非ノイズ クラスタへ再割り当てされます。ノイズ クラスタについての詳細は、TwoStep アルゴリズムのドキュメントを参照してください。

5. **変数偏差の指標を計算する:** 指定されたケース s に最も近いクラスタ h が検出されます。変数 X_k の変数偏差の指標 VDI_k は、対数-尤度距離 d(h, s) への変数の寄与率 $d_k(h, s)$ として定義されます。対応する平均値は M_{hk} で、X_k が連続型の場合は X_k のクラスタ サンプル平均であり、X_k がカテゴリ型の場合は X_k のクラスタ モードです。

6. **グループ偏差の指標を計算する:** ケースのグループ偏差指標 GDI は、対数-尤度距離 d(h, s) です。これは、すべての変数偏差指標 $\{VDI_k, k = 1, \cdots, K+1\}$ の合計です。

7. **異常値指標および変数寄与率測定値を計算する:**2 つの追加の指標が計算されますが、グループ偏差指標および変数偏差指標よりも解釈が簡単です。

   ケースの異常値指標は GDI の代替物で、ケースが属するクラスタの平均 GDI に対するケースの GDI の比率として計算されます。この指標の値を増やすと平均からの偏差がそれに従って大きくなり、よりふさわしい異常値候補が示されます。

   ケースの変数の可変寄与率測定値は VDI の代替物であり、ケースの GDI に対する変数の VDI の比率として計算されます。これは、ケースの偏差に対する変数の比例的寄与率です。この測定値が大きくなればなるほど、変数の偏差に対する寄与率が大きくなります。

異常な状況

ゼロによるゼロの除算

ケースの GDI がゼロでありケースが属するクラスタの平均 GDI もゼロの状況は、クラスタが単一であったり、同一のケースから構成され問題のケースも同一ケースと同じ場合があり得ます。このケースが異常値と見なされるかどうか、またはクラスタを構成する同一ケースの数が多いか少ないかは問題ではありません。たとえば、学習中の合計 10 ケースがあり、1 つのクラスタには単一、つまり 1 ケース、もう 1 つのクラスタには 9 ケースの 2 つのクラスタがあるとします。この状況で、シングルトンクラスタ内のケースが異常と見なされます。大きいほうのクラスタに属さないためです。この状況の異常値指標を計算する 1 つの方法は、クラスタ h のサイズに対する平均クラスタ サイズの比率としてその指標を設定することです。次のようになります。

$$\frac{n/H}{n_h}$$

10 ケースの例を続けると、シングルトン クラスタに属するケースの異常値指標は (10/2)/1 = 5 です。これは、アルゴリズムにとってこの状況を異常として捕捉するには十分な大きさです。この状況では変数寄与率測定値は 1/(K+1) に設定されます。(K+1) は、分析で処理済みの変数の数です。

ゼロによる非ゼロの除算

ケースの GDI が非ゼロであってもケースが属するクラスタの平均 GDI が 0 の状況は、クラスタがシングルトンであったり、クラスタが同一のケースから構成されても問題のケースが同一ケースと同じでない場合は、あり得ます。平均 GDI が 0 の、つまり $\text{average}(GDI)_h = 0$ のクラスタ h に属するケース i を想定します。ただし、ケース i とクラスタ h 間の GDI は非ゼロ、つまり $GDI(i, h) \neq 0$ です。ケース i の異常値指標算出のための選択の 1 つは、他のすべてのクラスタの重み付けされた平均 GDI として分母を設定することです。これは、この値が 0 でない場合です。0 の場合は、クラスタ h のサイズに対する平均クラスタ サイズの比率として計算を設定します。つまり、次のようになります。

$$\begin{cases} \dfrac{GDI(i,h)}{\frac{1}{(n-n_h)}\Sigma_{s=1,\neq h}^{H} n_s \cdot average(GDI)_s} & \text{if } \dfrac{1}{(n-n_h)}\displaystyle\sum_{s=1,\neq h}^{H} n_s \cdot average(GDI)_s \neq 0 \\ \dfrac{n/H}{n_h} & \text{otherwise} \end{cases}$$

この状況は、ケースが同一の複数ケースから構成されるクラスタに割り当てられているという警告のきっかけとなります。

理由付けの段階

これで各ケースは、グループ偏差指標および異常値指標を持ち、変数偏差指標と変数寄与率測定値のセットを持つことになります。この段階の目的は、異常の可能性の高いケースをランク付けし、異常と疑う理由を提供します。

1. **最も異常なケースを識別する:** 異常値指標の値の降順にケースをソートします。上位の $\text{pct}_{\text{anomaly}}$ %（またはその代わりとして上位の n_{anomaly}）で、異常値リストが作成されます。このリストは、異常値指標が $\text{cutpoint}_{\text{anomaly}}$ 以下のケースは異常と見なされないという制限があることが条件です。

2. **ケースが異常と見なされる理由を提供する:**各異常ケースについて、変数が対応する VDI_k 値の降順でソートされます。上位 k_{anomaly} の変数名、その値（対応する元の変数 X_{ok} の）、および期待値が理由として表示されます。

空白の処理

異常値と欠損値は、「アルゴリズムのステップ 」（ p.4 ）に説明があるように、ユーザーの設定に基づいてモデル作成時に処理されます。

生成されたモデル／スコアリング

異常値検出の生成されたモデルは、元の学習データ内で検出されたパターンに基づいて新規データ内の異常レコードを検出するのに使用できます。スコアリング対象の各レコードに対して異常値のスコアが生成され、異常性のステータスや異常値スコアを示すフラグが、新しいフィールドとして付加されます。

予測値

「スコアリング段階 」（ p. 5 ）の説明にあるように、モデルが構築されたときに作成されたクラスタ モデルに基づいて、各レコードの異常値のスコアが計算されます。異常値フラグが要求された場合は、「理由付けの段階 」（ p. 7 ）の説明にあるように、フラグが決定されます。

空白の処理

生成されたモデル内で、モデル構築時に使用された設定に従って空白が処理されます。 詳細は、 p. 5 スコアリング段階 を参照してください。

章

3

Apriori アルゴリズム

概要

Apriori は、データからアソシエーション ルールを抽出するためのアルゴリズムです。このアルゴリズムは、ルールの検索範囲を限定するために、多頻度アイテムセットを探して、多頻度アイテムセットから構成されたルールだけを調査します(Agrawal および Srikant, 1994)。

Apriori は、トランザクションから成り立つアイテムおよびアイテムセットを処理します。**アイテム**は、あるトランザクション中の特定の事柄の存在または非存在を示すフラグ型の条件です。**アイテムセット**は、トランザクション内で同時に発生する傾向がある、または発生する傾向がないアイテムのグループです。

IBM® SPSS® Modeler は、Christian Borgelt 氏による Apriori を実装しています。この Apriori の詳細は、次の場所で参照することができます。

http://fuzzy.cs.uni-magdeburg.de/~borgelt/doc/apriori/apriori.html.

ルールの派生

Apriori では、2 つのステージに分けて処理が行われます。まず、データ中の多頻度アイテムセットを識別し、次に多頻度アイテムセットのテーブルからルールを生成します。

多頻度アイテムセット

Apriori では、まず多頻度アイテムセットが識別されます。多頻度アイテムセットは、ユーザーにより指定された最小範囲の閾値 s_{min} 以上の範囲を持つアイテムセットと定義されます。アイテムセットの範囲は、アイテムセットが見つかったレコード数をレコードの総数で除算した値になります。

まずデータを調べて、この基準を満たす単一アイテムのアイテムセット（個別のアイテム、または長さ 1 のアイテムセット）を識別します。潜在的なアイテムをアイテムセットに追加すると常に潜在的なアイテムセットになるため、基準を満たさない単一アイテムがこれ以降考慮されることはありません。

次に Apriori は以下の作業を行って、再帰的に大きいアイテムセットを生成します。

- 長さ k（k アイテムを含む）のアイテムセット候補を生成するために、長さ $(k-1)$ の既存のアイテムセットを結合します。

  長さ $(k-1)$ の多頻度アイテムセット p と q の、可能なすべての組み合わせに対して、最初の $(k-2)$ アイテムを比較します（辞書式順序で）。両者が同じで、q の最後

のアイテムが（辞書式順序で）p の最後のアイテムより大きい場合、q の最後のアイテムを p の最後に追加して、長さ k の新しい候補アイテムセットが作成されます。

- 各候補アイテムセットの長さ $(k-1)$ のサブセットをチェックして候補セットを剪定します。すべてのサブセットが多頻度アイテムセットでなければなりません。そうでないと、候補アイテムセットは潜在的アイテムセットになり、以降の処理から除外されてしまいます。
- 候補セット中の各アイテムセットの範囲を次のように算出します。

$$?? = \frac{N_i}{N}$$

ここで N_i はアイテムセットに一致するレコード数を、N は学習データ中のレコード数を表します(このサポートの定義は、CARMA およびシーケンス ノードで使われているものとは異なります)。

- 範囲 $\geq s_{min}$ のアイテムセットが、多頻度アイテムセットのリストに追加されます。
- 長さ k の多頻度アイテムセットが見つかり、k がユーザーにより設定された最大ルール サイズ k_{max} 未満の場合、処理を繰り返して長さ $(k+1)$ の多頻度アイテムセットを探します。

ルールの生成

すべての多頻度アイテムセットが識別されたら、次に多頻度アイテムセットからルールが抽出されます。長さ $k > 1$ の各多頻度アイテムセット L に対して、次の処理が行われます。

- アイテムセットの長さ $(k-1)$ のすべてのサブセット A を、A 中のすべてのフィールドが入力フィールドで、アイテムセット中の他のすべてのフィールド（A 中に「存在していない」フィールド）が出力フィールドになるように計算します。以降のサブセット $\tilde{A}$ を呼び出します(最初の反復では、これは 1 つのフィールドですが、後の反復では複数のフィールドになります)。
- 各サブセット A に対して、ルール $A \Rightarrow \tilde{A}$ に対する評価測定（デフォルトではルール確信度）を以下のように計算します。
- 評価測定が、ユーザーにより定義された閾値より大きい場合、そのルールをルールテーブルに追加します。そして、A の長さ k’ が 1 より大きい場合は、長さが $(k'-1)$ である A の、可能なすべてのサブセットをテストします。

評価測定

Apriori は、保持するルールを判断するための、さまざまな評価測定法を提供しています。測定法に応じてルールの強調される側面も異なります。詳細は、『IBM® SPSS® Modeler User’s Guide』を参照してください。値は事前確信度と事後確信度に基づいて、次のように定義されます。

$$C_{prior} = \frac{c}{N}$$

であり、

$$C_{posterior} = \frac{r}{a}$$

ここで c は結果の範囲、a は前提条件の範囲、r は前提条件と結果の結合範囲、そして N は学習データ中のレコード数を表します。

ルール確信度: デフォルトのルール用評価測定は、ルールの事後確信度になります。

$$e = C_{posterior}$$

確信度との差異 (事前確信度との差の絶対値): この測定法は、事前確信度と事後確信度の値の差に基づいています。

$$e = |C_{posterior} - C_{prior}|$$

確信度の比 (確信係数と 1 の差異): この測定法は、事後確信度の事前確信度に対する比率に基づいています。

$$e = 1 - min\left(\frac{C_{posterior}}{C_{prior}}, \frac{C_{prior}}{C_{posterior}}\right)$$

情報の差 (事前確信度との情報の差): この測定法は、C5.0 ツリーの構築で使用されているのと同様の、情報の対応基準に基づいています。計算は次のように行われます。

$$e = \frac{r \cdot \log\left(\frac{r}{a \cdot c}\right) + (a - r)\log\left(\frac{a - r}{a \cdot \overline{c}}\right) + (c - r)\log\left(\frac{c - r}{\overline{a} \cdot c}\right) + (1 - a - c + r)\log\left(\frac{1 - a - c + r}{\overline{a} \cdot \overline{c}}\right)}{\log(2)}$$

ここで r はルールの範囲、a は前提条件の範囲、c は結果の範囲、$\overline{a} = 1 - a$ は前提条件の範囲の補数、そして $\overline{c} = 1 - c$ は結果の範囲の補数を表します。

カイ 2 乗値の正規化検定 (カイ 2 乗値の正規化検定の測定): この測定法は、カテゴリ データの独立に対するカイ 2 乗検定に基づいています。

$$e = \frac{(a \cdot c - r)^2}{a \cdot \overline{a} \cdot c \cdot \overline{c}}$$

空白の処理

Apriori アルゴリズムでは、空白は無視されます。このアルゴリズムでは、入力フィールドに対して空白を含むレコードは処理されます。ただし、そのようなレコードは、1 つまたは複数のフィールドに対して空白値を持つ任意のルールと一致するとはみなされません。

オプションの効果

ルールの最小範囲／最小確信値：これらの値は、どのルールがテーブルに取り込まれるかに関する制約を定義します。範囲および確信度値が、指定された値を超えたルールだけが、ルール テーブルに取り込まれます。

最大前提条件数：ルールに対して調べられる、前提条件の最大数を定義します。ルールの前提条件部にある条件数が指定された値と等しい場合、それ以上ルールが特殊化されることはありません。

フラグは真 (true) の値のみ：このオプションを選択した場合、偽の値を持つルールが入力フィールドまたは出力フィールドとみなされることはありません。

速度／メモリーの最適化 ： 処理速度とメモリー消費量のバランスを制御します。[**速度**] を選択すると、直接多頻度アイテムセット テーブルで条件値が使用され、可能な限りトランザクションがメモリーにロードされます。[**メモリー**] を選択すると、多頻度アイテムセット テーブル中の値テーブルでポインタが使用されます。多頻度アイテムセット テーブルでポインタを使用することにより、大きな問題に対してアルゴリズムが消費するメモリー量を節約することができます。ただしモデル構築時に、参照または逆参照を行うために余分な処理が必要になります。また、[**メモリー**] オプションを選択すると、トランザクションはメモリーにロードされずに、ファイルから処理されます。

生成されたモデル/スコアリング

Apriori ノードから生成されたモデルと新規データのスコアリングは、GRI 生成モデルと同様に処理されます。ただし、Apriori モデルは数値型フィールドをサポートしていないため、数値型フィールド中の空白値に対する警告は適用されません。

章 4

自動データ準備アルゴリズム

自動データ準備の目的は、学習の速度、予測の制度、準備されたデータに対するモデル適合度の強固さを全般的に改善するためにデータセットを準備することです。

これらのアルゴリズムでは、どのモデルが学習済み事後データ準備となるかは想定しません。自動データ準備の最後に、推奨された予測値の予測精度を出力します。予測精度は対象が連続型またはカテゴリ型かによって、線型回帰モデル、または Naïve Bayes モデルから計算されます。

表記

この章では特に明記しない限り、次の表記を使用します。

X	連続型変数またはカテゴリ変数
x_i	ケース i の変数 X の値。
f_i	ケース i の度数の重み。整数でない正の数は、最も近い整数に丸められます。度数の重み変数がない場合、すべて $f_i = 1$ となります。ケースの度数の重みが 0、負の数、または欠損地の場合、このケースは無視されます。
w_i	ケース i の分析の重み。分析の重み変数がない場合、すべて $w_i = 1$ となります。ケースの分析の重みが 0、負の数、または欠損地の場合、このケースは無視されます。
n	データセット内のケースの数。
N_X	$\sum_{i=1}^{n} f_i I\left(x_i \text{ は欠損値ではありません}\right)$。この場合、$I\left(\text{式}\right)$ は、式が真の場合に値 1 ととる指標関数で、偽の場合は 0 となります。
W_X	$\sum_{i=1}^{n} f_i w_i I\left(x_i \text{ は欠損値ではありません。}\right)$
N_{XY}	$\sum_{i=1}^{n} f_i I\left(x_i \text{ および } y_i \text{ は欠損値ではありません。}\right)$
W_{XY}	$\sum_{i=1}^{n} f_i w_i I\left(x_i \text{ および } y_i \text{ は欠損値ではありません。}\right)$
$\overline{x}$	変数 X の平均値、$\frac{1}{W_X}\sum_{i=1}^{n} f_i w_i x_i I\left(x_i \quad \text{は欠損値ではありません。}\right)$
M_X^r	$\sum_{i=1}^{n} f_i w_i (x_i - \overline{x})^r$
$\overline{x}_y$	$\frac{1}{W_{XY}}\sum_{i=1}^{n} f_i w_i x_i I\left(x_i \text{ および } y_i \text{ は欠損値ではありません。}\right)$
M_{XY}	$\sum_{i=1}^{n} f_i w_i \left(x_i - \overline{x}_y\right)\left(y_i - \overline{y}_x\right)$

欠損値についての注意

リストごとの削除が次のセクションで使用されます。

- 単変量統計収集 p. 15
- 基本変数のスクリーニング p. 17
- 測定レベルの変更 p. 18
- 欠損値の処理 p. 19
- 外れ値の識別と処理 p. 18
- 連続型予測値の変換 p. 20
- 対象の処理 p. 21
- カテゴリの並べ替え p. 26
- 非監視結合 p. 31

ペアごとの削除が次のセクションで使用されます。

- 2 変量統計収集 p. 22
- 監視結合 p. 26
- 監視されたデータ分割 p. 32
- フィールド選択と構築 p. 33
- 予測精度 p. 36

度数の重みおよび分析の重みについての注意

度数の重み変数は、ケース反復の重みとして処理されます。たとえば、ケースの度数の重みが 2 の場合、このケースは 2 ケースとしてカウントされます。

分析の重みは、ケースの分散を調整します。たとえば、変数 X の x_i に分析の重み w_i がある場合、$x_i \sim N\left(\mu, \frac{\sigma^2}{w_i}\right)$ のように想定します。

度数の重みおよび分析の重みは、自動準備で使用されますが、データセットのそれらの重み自体は変更されません。

日付/時刻の処理

日付の処理

日付変数がある場合、日付の要素（年、月、日）を順序変数として抽出します。必要に応じて、ユーザー指定の基準日（デフォルトは現在の日付）以降経過した日数/月数/年数も計算します。ユーザーによって指定されない場合、「最適な」期間が次のように選択されます。

1. 経過した最少の日数が 31 日未満の場合、日数を最適な単位として使用します。

2. 経過した最少の日数が 366 日未満で 31 日以上の場合、月数を最適な単位として使用します。2 つの日付の間の月数は、1 か月の平均日数（30.4375）に基づいて計算します。月数 = 日数/ 30.4375.
3. 経過した最少の日数が 366 日以上の場合、年数を最適な単位として使用します。2 つの日付の間の年数は、1 年間の平均日数（365.25）に基づいて計算します。年数 = 日数/ 365.25

日付の要素が抽出され、期間が取得されると、元の日付変数は、分析の残りから除外されます。

時間の処理

時間変数がある場合、時間の要素（秒、分、時間）を順序変数として抽出します。必要に応じて、ユーザー指定の基準時間（デフォルトは現在の時間）以降経過した秒数/分数/時間数も計算します。ユーザーによって指定されない場合、「最適な」期間が次のように選択されます。

1. 経過した最少の秒数が 60 秒未満の場合、秒数を最適な単位として使用します。
2. 経過した最少の秒数が 366 秒未満で 60 秒以上の場合、分数を最適な単位として使用します。
3. 経過した最少の秒数が 3600 秒以上の場合、時間数を最適な単位として使用します。

時間の要素が抽出され、期間が取得されると、元の時間の予測が除外されます。

単変量統計収集

連続変数

それぞれの連続変数について、次の統計量を計算します。

- 欠損値の数 ：$N_X^{missing} = \sum_{i=1}^{n} f_i I\left(x_i \text{ は欠損値です。}\right)$
- 有効値の数 ：N_X
- 最小値(M)：$\min_i x_i$
- 最大値(A)：$\max_i x_i$
- 平均値、標準偏差、歪度（下記参照）
- 異なる値の数 I。
- 各異なる値のケースの数 s_i：$c_i = \sum_{j=1}^{n} f_j I\left(x_j = s_i\right)$
- 中央値 ：X の各値が昇順に保存されている場合、 $s_1 < s_2 < \cdots < s_I$、中央値は $Median\left(X\right) = \min\left\{s_i : \frac{cc_i}{N_X} \geq 0.5\right\}$ によって計算できます。この場合、$cc_i = \sum_{j=1}^{i} c_i$ となります。

注 ： 異なる値の数が閾値より大きい場合（デフォルトは 5)、異なる値の数、および各値のケース数の更新を停止します。また、中央値を計算しません。

カテゴリ数値変数

それぞれのカテゴリ数値変数について、次の統計量を計算します。

- 欠損値の数 ：$N_X^{missing} = \sum_{i=1}^{n} f_i I\left(x_i \text{ は欠損値です。}\right)$
- 有効値の数 ：N_X
- 最小値(M)：$\min_i x_i$ （順序変数のみ）
- 最大値(A)：$\max_i x_i$ （順序変数のみ）
- カテゴリ数。
- 各カテゴリの度数。
- 平均値、標準偏差、歪度（順序変数のみ）。（下記参照）
- 最頻値（名義変数のみ）。複数の値が最高の頻度を共有する場合は、最も小さい値の最頻値が使用されます。
- 中央値（順序変数のみ）：X の異なる値が昇順に保存され、 $s_1 < s_2 < \cdots < s_I$ の場合、中央値は $Median\left(X\right) = \min\left\{s_i : \frac{cc_i}{N_X} \geq 0.5\right\}$ を使用して計算されます。この場合、$cc_i = \sum_{j=1}^{i} c_i$ となります。

注：

1. 順序型予測に指定された閾値（デフォルトは 10）より多くのカテゴリが含まれている場合、カテゴリ数、および各カテゴリのケース数の更新を停止します。また、最頻値および中央値の計算も行いません。
2. 順序型予測に指定された閾値（デフォルトは 100）より多くのカテゴリが含まれている場合、統計量の収集を停止し、変数に閾値カテゴリ以上のカテゴリが含まれているという情報を保存します。

カテゴリ文字列変数

それぞれの文字列変数について、次の統計量を計算します。

- 欠損値の数 ：$N_X^{missing} = \sum_{i=1}^{n} f_i I\left(x_i \text{ は欠損値です。}\right)$
- 有効値の数 ：N_X
- カテゴリ数。
- 各カテゴリの度数。
- 最頻値 ：複数の値が最高の頻度を共有する場合は、最も小さい値の最頻値が使用されます。

注 ： 文字列型予測に指定された閾値（デフォルトは 100）より多くのカテゴリが含まれている場合、統計量の収集を停止し、予測に閾値カテゴリ以上のカテゴリが含まれているという情報を保存します。

平均、標準偏差、歪度

モーメントを更新して、平均値、標準偏差、ワイドを計算します。

1. はじめに $N_X^{(0)} = W_X^{(0)} = \overline{x}^{(0)} = M_X^{2(0)} = M_X^{3(0)} = 0$ を計算します。

2. j=1, . . , n について、次を計算します。

$N_X^{(j)} = N_X^{(j-1)} + f_j I\left(x_j \text{ は欠損値ではありません。}\right)$

$W_X^{(j)} = W_X^{(j-1)} + f_j w_i I\left(x_j \text{ は欠損値ではありません。}\right)$

$v_j = \frac{f_j w_j}{W_X^{(j)}}\left(x_j - \overline{x}^{(j-1)}\right)$

$\overline{x}^{(j)} = \overline{x}^{(j-1)} + v_j$

$M_X^{2(j)} = M_X^{2(j-1)} + \frac{W_X^{(j)} W_X^{(j-1)}}{f_j w_j} v_j^2$

$M_X^{3(j)} = M_X^{3(i-1)} - 3v_j M_X^{2(j-1)} + \frac{W_X^{(j)} W_X^{(j-1)}}{(f_j w_j)^2}\left(W_X^{(j)} - 2f_j w_j\right) v_j^3$

3. 最後のケースを処理した後、次を計算します。

平均値 $: \overline{x} = \overline{x}^{(n)}$

標準偏差 $: sd = \sqrt{\frac{M_X^{2(n)}}{N_X - 1}}$

歪度 $: skew = \frac{\frac{N_X}{(N_X - 2)} \frac{1}{(N_X - 1)} M_X^{3(n)}}{sd^3}$

$N_X \leq 2$ または $sd^2 < 10^{-20}$の場合、歪度は計算されません。

基本変数のスクリーニング

1. 欠損値の割合が閾値より大きい場合（デフォルトは 50%)、後続の分析から変数を除外します。

2. 連続型変数について、最大値が最小値と等しい場合、後続の分析から変数を除外します。

3. カテゴリ変数で、最頻値に指定された割合より多くのケースが含まれている場合（デフォルトは 95%)、後続の分析から変数を除外します。

4. 文字列変数に指定された閾値より多くのカテゴリが含まれている場合（デフォルトは 50%)、後続の分析から変数を除外します。

チェックポイント 1:終了?

このチェックポイントは、アルゴリズムを終了するかどうかを指定します。スクリーニングの手順の後、

1. 対象（指定されている場合）が後続の分析から削除されている場合、または

2. すべての予測値が後続の分析から削除されている場合、

アルゴリズムを終了し、エラーを生成します。

測定レベルの変更

各連続型変数について、異なる値の数が閾値より少ない場合（デフォルトは 5)、順序型変数に変更されます。

各数値順序型変数について、カテゴリの数が閾値より少ない場合（デフォルトは 10)、連続型変数に変更されます。

注 ： 連続型-順序型の閾値は、順序型-連続型閾値より小さくなければなりません。

外れ値の識別と処理

ここでは、連続型変数の外れ値を識別し、外れ値を分割値または欠損値に設定します。識別は、外れ値の割合が最大 5% であることを仮定することによって推定された強固な平均値と強固な標準偏差に基づいています。

識別

1. 生のデータから平均値と標準偏差を計算します。連続型変数を非交差区間に分割します。$I_i = (\overline{x} + (i-1) \times sd_w, \overline{x} + i \times sd_w], i = -3, -2, \cdots, 2, 3, 4$ この場合、$I_{-3} = (-\infty, \overline{x} - 3sd_w]$， $I_4 = (\overline{x} + 3sd_w, +\infty]$ および $sd_w = sd \times \sqrt{\frac{N_X - 1}{W_X - 1}}$となります。
2. 各区間で単変量統計を計算します。

   $N_{I_i} = \sum_{j=1}^{n} f_j I(x_j \in I_i)$， $W_{I_i} = \sum_{i=1}^{n} f_i w_i I(x_j \in I_i)$

   $\overline{x}_{I_i} = \frac{\sum_{j=1}^{n} f_j w_j x_j I(x_j \in I_i)}{W_{I_i}}$， $M_{I_i}^2 = \sum_{j=1}^{n} f_j w_j (x_j - \overline{x}_{I_i})^2 I(x_j \in I_i)$
3. $l = -3$， $r = 4$ および $p = 0$ とします。
4. 末尾の 2 つの区間 I_l および I_r の間に、最少数のケースを含む区間があります。
5. $N_{I_l} \le N_{I_r}$ の場合、$p_{current} = \frac{N_{I_l}}{N_X}$ となります。$p + p_{current}$ が閾値 $p_{threshold}$ より小さいことを確認してください（デフォルトは 0.05)。閾値より小さい場合、$p = p + p_{current}$ および $l = l + 1$ となり、ステップ 4 に進みます。そうでない場合はステップ 6 に進みます。

   または $p_{current} = \frac{N_{I_r}}{N_X}$ となります。$p + p_{current}$ が閾値 $p_{threshold}$ より小さいことを確認してください。閾値より小さい場合、$p = p + p_{current}$ および $r = r - 1$ となり、ステップ 4 に進みます。そうでない場合はステップ 6 に進みます。
6. 範囲 $(\overline{x} + (l-1) \times sd, \overline{x} + r \times sd]$ 内の強固な平均値 $\overline{x}_{robust}$ および強固な標準偏差 sd_{robust} を計算します。詳細は以下を参照してください。
7. x_i が次の条件を満たす場合、

   $\sqrt{w_i}(x_i - \overline{x}_{robust}) < -cutoff \times sd_{robust}$ or $\sqrt{w_i}(x_i - \overline{x}_{robust}) > cutoff \times sd_{robust}$

   この場合、cutoff は正の数（デフォルトは 3）となり、x_i は外れ値として検出されます。

処理

外れ値は次の方法のいずれかを使用して処理されます。

- 外れ値を分割値に除外します。$\sqrt{w_i}(x_i - \overline{x}_{robust}) < -cutoff \times sd_{robust}$ の場合、 x_i by $\overline{x}_{robust} - cutoff \times sd_{robust}/\sqrt{w_i}$置き換え、 $\sqrt{w_i}(x_i - \overline{x}_{robust}) > cutoff \times sd_{robust}$ の場合、 x_i を $\overline{x}_{robust} + cutoff \times sd_{robust}/\sqrt{w_i}$.
- 外れ値を欠損値に設定します。

単変量統計の更新

外れ値を処理したあと、データ パスを実行して、欠損値の数、最小値、最大値、平均値、標準偏差、歪度、外れ値の数など、各連続型変数の単変量統計を計算します。

強固な平均および標準偏差

範囲 $(\overline{x} + (l-1) \times sd, \overline{x} + r \times sd]$ 内の強固な平均および標準偏差は、次のようにして計算します。

$$\overline{x}_{robust} = \frac{\sum_{i=l}^{r} W_{I_i} \overline{x}_{I_i}}{\sum_{i=l}^{r} W_{I_i}}$$

そして

$$sd_{robust} = \sqrt{\frac{M^2_{robust}}{\sum_{i=l}^{r} N_{I_i} - 1}}$$

この場合、$M^2_{robust} = \sum_{i=l}^{r} A_{I_i}$ そして $A_{I_i} = M^2_{I_i} + W_{I_i}(\overline{x}_{robust} - \overline{x}_{I_i})^2$ となります。

欠損値の処理

連続型変数: 欠損値は平均値と置き換えられ、次の統計量が更新されます。

- 標準偏差 :$sd \times \sqrt{\frac{N_X - 1}{N-1}}$、この場合、$N = N_X + N_X^{missing}$ となります。
- 歪度 : $skew \times \frac{L_1}{L_2}$、この場合、$L_1 = \left(\frac{N}{N-2}\right)\left(\frac{N_X - 2}{N_X}\right)$ and $L_2 = \sqrt{\frac{N_X - 1}{N-1}}$
- 欠損値の数 :$N_X^{missing} = 0$
- 有効値の数 :$N_X = N$

順序型変数: 欠損値は中央値と置き換えられ、次の統計量が更新されます。

- 中央カテゴリ内のケース数 :$c_{median} + N_X^{missing}$、この場合、$c_{median}$ は、中央カテゴリ内のケースの元の数です。
- 欠損値の数 :$N_X^{missing} = 0$
- 有効値の数 :$N_X = N$

名義型変数: 欠損値は最頻値と置き換えられ、次の統計量が更新されます。

- モーダルなカテゴリ内のケース数 :$c_{mode} + N_X^{missing}$、この場合、c_{mode} は、モーダルなカテゴリ内のケースの元の数です。
- 欠損値の数 :$N_X^{missing} = 0$
- 有効値の数 :$N_X = N$

連続型予測値の変換

Z スコア変換を使用して連続型予測値を変換し、ユーザー指定の平均値 $\overline{x}_{user}$（デフォルト 0）および sd_{user}（デフォルト 1）が含まれるように、または最小-最大変換を使用して変換し、最小値 $\min_{user}$（デフォルト 0）および最大値 $\max_{user}$（デフォルト 100）が含まれるようにします。

Z スコア変換

連続型変数に平均値 $\overline{x}$ および標準偏差 sd が含まれているものとします。z スコア変換は次のようになります。

$$x_i^{'} = \frac{sd_{user}}{sd} \times (x_i - \overline{x}) + \overline{x}_{user}$$

ここで、$x_i^{'}$ は、ケース i の連続型変数 X の変換された値となります。

再スケール式では分析の重みを考慮しないため、再スケール化された値 $x_i^{'}$ は、通常の分布 $N\left(\overline{x}_{user}, \frac{sd_{user}^2}{w_i}\right)$ に従います。

単変量統計の更新

Z スコア変換の後、次の単変量統計が更新されます。

- 欠損値の数 :$N_{X'}^{missing} = N_X^{missing}$
- 有効値の数 :$N_{X'} = N_X$
- 最小値:$\min\left(x_i^{'}\right) = \frac{sd_{user}}{sd} \times (\min x_i - \overline{x}) + \overline{x}_{user}$
- 最大値:$\max\left(x_i^{'}\right) = \frac{sd_{user}}{sd} \times (\max x_i - \overline{x}) + \overline{x}_{user}$
- 平均値 :$\overline{x}^{'} = \overline{x}_{user}$
- 標準偏差 :$sd\left(x^{'}\right) = sd_{user}$
- 歪度 :$skew\left(x^{'}\right) = skew\,(x)$

Min-Max 変換

連続型変数に最小値 $\min x_i$ と最大値 $\max x_i$ が含まれていると仮定します。Min-Max 変換は次のようになります。

$$x_i^{'} = \frac{\max_{user} - \min_{user}}{\max x_i - \min x_i} \times (x_i - \min x_i) + \min_{user}$$

ここで、$x_i^{'}$ は、ケース i の連続型変数 X の変換された値となります。

単変量統計の更新

min-max 変換の後、次の単変量統計が更新されます。

- 欠損値の数 ：$N_{X'}^{missing} = N_X^{missing}$
- 有効値の数 ：$N_{X'} = N_X$
- 最小値(M)：$\min\left(x_i^{'}\right) = \min_{user}$
- 最大値(A)：$\max\left(x_i^{'}\right) = \max_{user}$
- 平均値 ：$\overline{x}^{'} = \frac{\max_{user} - \min_{user}}{\max x_i - \min x_i} \times (\overline{x} - \min x_i) + \min_{user}$
- 標準偏差 ：$sd\left(x^{'}\right) = \frac{\max_{user} - \min_{user}}{\max x_i - \min x_i} \times sd$
- 歪度 ：$skew\left(x^{'}\right) = skew\,(x)$

対象の処理

名義型対象

名義型対象の場合、最小度数から最大度数まで、カテゴリを再調整します。度数にタイがある場合、タイはデータ値の昇順のソートまたは文字順に解決されます。

連続型目標

Box および Cox（1964）変換によって提唱された変換では、連続型変数をより世紀に分布する変数に変換します。Box-Cox 変換の後に z スコア変換を行うと、再スケールされた対象にはユーザ指定の平均値および標準偏差が含まれます。

Box-Cox 変換：これにより、非名義型変数 Y がより正規に分布するよう変換されます。

$$g_i(\lambda) = g(y_i, \lambda) = \begin{cases} \frac{\left((y_i - c)^{\lambda} - 1\right)}{\lambda} & \lambda \neq 0 \\ \ln(y_i - c) & \lambda = 0 \end{cases}$$

ここで、$y_i, i = 1, 2, \cdots, n$ は変数 Y の観測、c はすべての値 $y_i - c$ が正の数となるような定数です。ここで、$c = \min(Y) - 1$ を選択します。

パラメータ λ を選択して、対数尤度関数を最大化します。

$$L(\lambda) = -\frac{N_Y}{2} \ln\left[\frac{N_Y - 1}{N_Y}(sd\,(g\,(\lambda)))^2\right] + (\lambda - 1)\sum_{i=1}^{n} f_i \ln(y_i - c)$$

この場合、$(sd(g(\lambda)))^2 = \frac{1}{N_Y - 1}\sum_{i=1}^{n} f_i w_i (g_i(\lambda_j) - \overline{g}(\lambda_j))^2$ そして $\overline{g}(\lambda) = \frac{1}{W_Y}\sum_{i=1}^{n} f_i w_i g_i(\lambda)$ となります。

増分 s のユーザ指定の有限セット［a, b ］でグリッド検索を実行します。デフォルトでは、a=−3、b=3、s=0.5 です。

アルゴリズムは、次のように説明されます。

1. $\lambda_j = a + (j-1) * s$ を計算します。この場合、j は、 $a \le \lambda_j \le b$ となるような整数です。
2. 各 λ_j について、次の統計量を計算します。

   平均値 ：$\overline{g}(\lambda_j) = \frac{1}{W_Y}\sum_{i=1}^{n} f_i w_i g_i(\lambda_j)$

   標準偏差 ：$sd(g(\lambda_j)) = \sqrt{\frac{1}{N_Y - 1}\sum_{i=1}^{n} f_i w_i (g_i(\lambda_j) - \overline{g}(\lambda_j))^2}$

   歪度 ：$skew(g(\lambda_j)) = \frac{\frac{N_Y}{(N_Y-2)}\frac{1}{(N_Y-1)}\sum_{i=1}^{n} f_i w_i (g_i(\lambda_j) - \overline{g}(\lambda_j))^3}{sd(g(\lambda_j))^3}$

   対数変換の合計 ：$\sum_{i=1}^{n} f_i \ln(y_i - c)$
3. 各 λ_j について、対数尤度関数 $L(\lambda_j)$ を計算します。最大対数尤度関数で j の値を検索し、λ_j の最小値を選択してタイを解決します。また、対応する統計 $\overline{g}(\lambda^*)$, $sd(g(\lambda^*))$ および $skew(g(\lambda^*))$ を検索します。
4. 対象を変換して、ユーザーの平均値 $\overline{y}_{user}$（デフォルトは 0）、標準偏差 sd_{user}（デフォルトは 1）を反映します。

$$y_i^{'} = \frac{sd_{user}}{sd(g(\lambda^*))} \times (g_i(\lambda^*) - \overline{g}(\lambda^*)) + \overline{y}_{user}$$

この場合、$\overline{g}(\lambda^*) = \frac{1}{W_Y}\sum_{i=1}^{n} f_i w_i g_i(\lambda^*)$ そして $sd(g(\lambda^*)) = \sqrt{\frac{1}{N_Y - 1}\sum_{i=1}^{n} f_i w_i (g_i(\lambda^*) - \overline{g}(\lambda^*))^2}$.

単変量統計の更新： Box-Cox 変換および Z スコア変換の後、次の単変量統計が更新されます。

- 最小値: $\frac{sd_{user}}{sd(g(\lambda^*))} \times (g(\min(y_i) - c, \lambda^*) - \overline{g}(\lambda^*)) + \overline{y}_{user}$
- 最大値: $\frac{sd_{user}}{sd(g(\lambda^*))} \times (g(\max(y_i) - c, \lambda^*) - \overline{g}(\lambda^*)) + \overline{y}_{user}$
- 平均値 ：$\overline{y}_{user}$
- 標準偏差 ：sd_{user}
- 歪度 ：$skew(g(\lambda^*))$

2 変量統計収集

対象/予測の各ペアについて、対象および予測値の測定レベルにしたがって、次の統計量が収集されます。

連続型対象または対象なしおよびすべての連続型予測値

1 つの連続型対象および複数の連続型予測値がある場合、連続型変数のすべてのペア間の共分散および相関を計算する必要があります。連続型対象がない場合、連続型予測値のすべてのペア間の共分散および相関を計算します。m 個の連続型変数があると仮定し、共分散行列を $C_{m\times m}$、要素は c_{ij}、相関行列を $R_{m\times m}$、要素は r_{ij} と表記します。

2 つの連続型変数 X および Y 間の共分散を次のように定義します。

$$c_{XY} = \frac{1}{N_{XY}-1}\sum_{i=1}^{n} f_i w_i \left(x_i - \overline{x}_y\right)\left(y_i - \overline{y}_x\right)$$

この場合、$\overline{x}_y = \frac{1}{W_{XY}}\sum_{i=1}^{n} x_i I\left(x_i \text{ および } y_i \text{ は欠損値ではありません。}\right)$ そして $\overline{y}_x = \frac{1}{W_{XY}}\sum_{i=1}^{n} y_i I\left(x_j \text{ および } y_j \text{ は欠損値ではありません。}\right)$ となります。

共分散は、暫定平均アルゴリズムで計算されます。

1. まず、$N_{XY}^{(0)} = W_{XY}^{(0)} = \overline{x}_y = \overline{y}_x = M_{XY}^{(0)} = 0$ を計算します。
2. j=1,..,n について、次の計算を行います。

   $N_{XY}^{(j)} = N_{XY}^{(j-1)} + f_j I\left(x_j \text{ および } y_j \text{ は欠損値ではありません。}\right)$

   $W_{XY}^{(j)} = W_{XY}^{(j-1)} + f_j w_j I\left(x_j \text{ および } y_j \text{ は欠損値ではありません。}\right)$

   $v_{xj} = \frac{f_j w_j}{W_{XY}^{(j)}}\left(x_j - \overline{x}_y\right)$

   $\overline{x}_y = \overline{x}_y + v_{xj}$

   $v_{yj} = \frac{f_j w_j}{W_{XY}^{(j)}}\left(y_j - \overline{y}_x\right)$

   $\overline{y}_x = \overline{y}_x + v_{yj}$

   $M_{XY}^{(j)} = M_{XY}^{(j-1)} + \left(x_j - \overline{x}_y\right)\left(y_j - \overline{y}_x\right)\left(f_j w_j - \frac{\left(f_j w_j\right)^2}{W_{XY}^{(j)}}\right)$

   最後のケースを処理した後、次を取得します。

   $M_{XY} = M_{XY}^{(n)} = \sum_{i=1}^{n} f_i w_i \left(x_i - \overline{x}_y\right)\left(y_i - \overline{y}_x\right)$
3. X および Y 間の 2 変量統計を計算します。

   有効ケースの数 :N_{XY}

   共分散 :$c_{XY} = \frac{M_{XY}}{N_{XY}-1}$

   相関 :$r_{XY} = \frac{c_{XY}}{\sqrt{c_{XX}}\sqrt{c_{YY}}}$

   注 :ペアごとの削除を行っている場合に有効なケースがない場合、$c_{XY} = 0$ および $r_{XY} = 0$ とします。

カテゴリ対象およびすべての連続型予測値

値が $i = 1, 2, \cdots, J$ のカテゴリ対象 Y および値が $x_1, \cdots x_n$ の連続型予測値 X について、2 変量統計は次のようになります。

Y=i , i=1,...,J のそれぞれの場合の X の平均値は次のようになります。

$$\overline{x}_{\cdot i} = \frac{\sum_{j=1}^{n} f_j w_j x_j I\left(y_j = i\right)}{\sum_{j=1}^{n} f_j w_j I\left(y_j = i\right)}$$

Y=i, i =1,...,J のそれぞれの場合の X の平方和誤差は次のようになります。

$$M_{\cdot i}^2 = \sum_{j=1}^{n} f_j w_j (x_j - \overline{x}_{\cdot\ i})^2 I\left(y_j = i\right)$$

Y =i, i=1,...,J のそれぞれの場合の度数の重みの合計は次のようになります。

$$N_{\cdot i} = \sum_{j=1}^{n} f_j I\left(y_j = i \wedge x_j \text{ は欠損値ではありません。}\right)$$

無効ケースの数 :

$$N_{XY} = \sum_{i=1}^{J} N_{\cdot i}$$

Y =i, i=1,...,J の重みの合計（度数の重みX分析の重み）は次のようになります。

$$W_{\cdot i} = \sum_{j=1}^{n} f_j w_i I\left(y_j = i \wedge x_j \text{ は欠損値ではありません。}\right)$$

連続型対象およびすべてのカテゴリ予測値

値が i=1,...,J の連続型対象 Y およびカテゴリ予測値 X の場合、2 変量統計には次のものが含まれます。

X について条件がある Y の平均値 :

$$\overline{y}_x = \frac{\sum_{i=1}^{I} \sum_{j=1}^{n} f_j w_j y_j I\left(x_j = i\right)}{\sum_{i=1}^{I} \sum_{j=1}^{n} f_j w_j I\left(x_j = i\right)}$$

Y の平方和の誤差 ：

$$M_{X.}^2 = \sum_{j=1}^{n} f_j w_j (y_j - \overline{y}_x)^2$$

$X = i$, i=1,...,J のそれぞれの場合の Y の平均値は次のようになります。

$$\overline{y}_{i.} = \frac{\sum_{j=1}^{n} f_j w_j y_j I(x_j = i)}{\sum_{j=1}^{n} f_j w_j I(x_j = i)}$$

$X = i$, i=1,...,J のそれぞれの場合の Y の平方和の誤差は次のようになります。

$$M_{i.}^2 = \sum_{j=1}^{n} f_j w_j (y_j - \overline{y}_{i.})^2 I(x_j = i)$$

$X = i$, i=1,...,J のそれぞれの場合の度数の重みの合計は次のようになります。

$$N_{i.} = \sum_{j=1}^{n} f_j I(x_j = i \wedge y_j \text{ is not missing})$$

$X = i$, i=1,...,J のそれぞれの場合の重みの合計（度数の重み X 分析の重み）は次のようになります。

$$W_{i.} = \sum_{j=1}^{n} f_j w_j I(x_j = i \wedge y_j \text{ は欠損値ではありません。})$$

カテゴリ対象およびすべてのカテゴリ予測値

値が j=1,...,J のカテゴリ対象 Y および値が i=1,...,I のカテゴリ予測値 X の場合、2 変量統計は次のようになります。

$x_k = i$ および $y_k = j$ の各組み合わせの度数の重みの合計 ：

$$N_{ij} = \sum_{k=1}^{n} f_k I(x_k = i \wedge y_k = j)$$

$x_k = i$ および $y_k = j$ の各組み合わせの重みの合計（度数の重み X 分析の重み）：

$$W_{ij} = \sum_{k \in 1}^{n} f_k w_k I(x_k = i \wedge y_k = j)$$

カテゴリ変数の処理

この手順では、単変量統計または 2 変量統計を使用して、カテゴリ予測値を処理します。

カテゴリの並べ替え

名義型予測値の場合、最小度数から最大度数まで、カテゴリを再調整します。度数にタイがある場合、タイはデータ値の昇順のソートまたは文字順に解決されます。新しいフィールド値は、頻度が最も少ないカテゴリの 0 から始まります。元のフィールドが文字列型である場合でも、新しいフィールドは数値型になります。たとえば、名義型フィールドのデータ値が「A」、「A」、「A」、「B」、「C」、「C」の場合、自動データ準備は「B」を 0 に、「C」を 1 に、「A」を 2 に再コード化します。

高度に関連するカテゴリ フィールドの識別

データ セットに対象がある場合、p が α レベルの $\alpha_{selection}$ より大きくない場合（デフォルトは 0.05）、順序型/名義型予測値を選択します。これらの p 値の計算の詳細については、「P 値の計算 」（ p.28 ）を参照してください。

2 変量統計の収集時にペアごとの削除を使用して欠損値を処理するため、ケースが 0 件のカテゴリがいくつか発生する場合があります。つまりカテゴリ予測値のカテゴリ i の場合、$N_{i.} = 0$ となります。p 値を計算する場合、これらのカテゴリが除外されます。

ケースが 0 のカテゴリを除外した後カテゴリが 1 つだけまたは 0 になった場合、p 値を 1 に設定し、この予測値は選択されません。

監視結合

1 レベルの深さの CHAID ツリーに類似した監視方法を使用して、順序型/名義型予測値のカテゴリを結合します。

1. ケース度数が 0 のすべてのカテゴリを除外します。
2. X にカテゴリがない場合、除外されたすべてのカテゴリを 1 つのカテゴリに結合し、停止します。
3. X のカテゴリが 1 つの場合、ステップ 7 に進みます。
4. その後、最も類似した X のカテゴリの使用可能なペアを探します。それは、検定統計が対象に関する最も大きい P 値を与えるペアです。順序型予測のカテゴリの使用可能なペアは、2 つの隣接するカテゴリです。名義型予測の場合、任意の 2 つのカテゴリです。順序型予測について、i 番目のカテゴリと j 番目のカテゴリが 0 ケースのために除外された場合、i 番目のカテゴリと j 番目のカテゴリが隣接したカテゴリとなります。これらの p 値の計算の詳細については、「P 値の計算 」（ p.28 ）を参照してください。

5. もっとも大きな p 値を持つペアの場合、p 値が指定された α レベルの $\alpha_{selection}$ より大きいかどうかを確認してください（デフォルトは 0.05)。大きい場合、このペアは、この新しいカテゴリの 2 変量統計を計算するのと同時に、単一の結合されたカテゴリに結合されます。その後、X のカテゴリの新しいセットが形成されます。小さい場合、ステップ 6 へ進みます。

6. ステップ 3 に進みます。

7. 順序型予測値の場合、新しいそれぞれのカテゴリの最大値を検索します。これらの最大値を昇順でソートします。r 件の新規カテゴリがあるとすると、最大値は $i_1 < i_2 < \cdots < i_r$ となり、結合ルールが次のようになります。最初の新規カテゴリには $X \leq i_1$ となるすべての元のカテゴリが含まれ、2 番目の新規カテゴリには $i_1 < X \leq i_2, \cdots$ となるすべての元のカテゴリ、最後の新規カテゴリには $X > i_{r-1}$ となる元のカテゴリがすべて含まれます。

   名義型予測値の場合、ステップ 1 で除外されたすべてのカテゴリは、最も度数の低い新規カテゴリに結合されます。最も低い度数のカテゴリにタイがある場合、最も度数の低い新規カテゴリを形成した元のカテゴリ地の昇順ソートまたは文字順ソートによって、最も値の小さいカテゴリを選択して、タイを解決します。

新しいカテゴリの 2 変量統計の計算

2 つのカテゴリが新しいカテゴリに結合される場合、この新規カテゴリの 2 変量統計を計算する必要があります。

尺度対象 :カテゴリ i および i' を p 値に基づいて結合でき、2 変量統計を次のように計算する必要があります。

$$N_{i,i'\,.} = N_{i\,.} + N_{i'\,.}$$

$$W_{i,i'\,.} = W_{i\,.} + W_{i'\,.}$$

$$\overline{y}_{i,i'\,.} = \overline{y}_{i\,.} + \frac{W_{i'\,.}}{W_{i,i'\,.}} \left(\overline{y}_{i'\,.} - \overline{y}_{i\,.}\right)$$

$$M^2_{i,i'\,.} = M^2_{i\,.} + M^2_{i'\,.} + W_{i\,.} \left(\overline{y}_{i,i'\,.} - \overline{y}_{i\,.}\right)^2 + W_{i'\,.} \left(\overline{y}_{i,i'\,.} - \overline{y}_{i'\,.}\right)^2$$

カテゴリ対象 :カテゴリ i および i' を p 値に基づいて結合でき、2 変量統計を次のように計算する必要があります。

$$N_{i,i'j} = N_{ij} + N_{i'j}$$

$$W_{i,i'j} = W_{ij} + W_{i'j}$$

単変量統計および 2 変量統計の更新

監視結合手順の終わりに、各新規カテゴリの 2 変量統計を計算します。単変量統計の場合、各新規カテゴリの度数は、新規カテゴリを形成した元のカテゴリの度数の合計となります。「単変量統計収集 」 （ p. 15 ） の式に従ってそのほかの統計量を更新しますが、統計量は、新規カテゴリおよびこれらのカテゴリの数値型ケースの基づいて更新する必要があります。

P 値の計算

各 p 値の計算は、予測値および対象間の関連に関する適切な統計量検定に基づきます。

尺度対象

F 統計量を計算します。

$$F = \frac{\sum_{i=1}^{I} W_{i\cdot}(\overline{y}_{i\cdot} - \overline{y}_x)^2 / (I-1)}{\sum_{i=1}^{I} M_{i\cdot}^2 / \left(\sum_{i=1}^{I} N_{i\cdot} - I\right)}$$

ここで $\overline{y}_x = \frac{\sum_{i=1}^{I} W_{i\cdot}\overline{y}_{i\cdot}}{\sum_{i=1}^{I} W_{i\cdot}}$.

F 統計量に基づくと、p 値は次のようになります。

$$p = \Pr\left(F\left(I-1, \sum_{i=1}^{I} N_{i\cdot} - I\right) > F\right)$$

ここで、$F\left(I-1, \sum_{i=1}^{I} N_{i\cdot} - I\right)$ は、自由度が $I-1$ および $\sum_{i=1}^{I} N_{i\cdot} - I$ の F に従った無作為変数です。

結合の手順では、F 統計量および X の 2 つのカテゴリ i および i' 間の p 値を次のように計算します。

$$F = \frac{W_{i\cdot}\left(\overline{y}_{i\cdot} - \overline{y}_{i,i'\cdot}\right)^2 + W_{i'\cdot}\left(\overline{y}_{i'\cdot} - \overline{y}_{i,i'\cdot}\right)^2}{\left(M_{i\cdot}^2 + M_{i'\cdot}^2\right) / \left(N_{i\cdot} + N_{i'\cdot} - 2\right)}$$

$$p = \Pr\left(F\left(1, N_{i\cdot} + N_{i'\cdot} - 2\right) > F\right)$$

ここで、$\overline{y}_{i,i'\cdot}$ は、i と i' で結合された新しいカテゴリ i, i' の Y の平均値です。

$$\overline{y}_{i,i'\cdot} = \overline{y}_{i\cdot} + \frac{W_{i'\cdot}}{W_{i\cdot} + W_{i'\cdot}}\left(\overline{y}_{i'\cdot} - \overline{y}_{i\cdot}\right)$$

そして $F(I-1, N_{i\cdot}+N_{i'\cdot}-2)$ は、自由度が 1 および $N_{i\cdot}+N_{i'\cdot}-2$ の F 分布に従った無作為変数です。

名義型対象

X および Y の独立に対する帰無仮説が検定されます。最初の分割（または度数）表は、Y のクラスを列、予測値 X を行ととして作成されます。帰無仮説での期待セル度数が推定されます。観測セル度数と期待セル度数を使用して Pearson カイ 2 乗統計と p 値が次のように計算されます。

$$X^2 = \sum_{j=1}^{J}\sum_{i=1}^{I}\frac{\left(N_{ij}-\hat{m}_{ij}\right)^2}{\hat{m}_{ij}}$$

ここで、$N_{ij}=\sum_{k\in D} f_k I\left(x_k=i \wedge y_k=j\right)$ は観測セル度数で、 $\hat{m}_{ij}$ は独立モデルに従ったセルの推定期待セル度数 $(x_k=i, y_k=j)$ です。$\hat{m}_{ij}=0$ の場合、$\frac{(N_{ij}-\hat{m}_{ij})^2}{\hat{m}_{ij}}=0$ となります。$\hat{m}_{ij}$ の推定方法は、下記で説明しています。

対応する p 値は $p=\Pr\left(\chi_d^2 > X^2\right)$ で指定されます。この場合、χ_d^2 は、自由度が $d=(J-1)(I-1)$ のカイ 2 乗分布に従います。

X の 2 つのカテゴリ i および i' を結合できるかどうかを検証する場合、Pearson カイ 2 乗統計量は次のようになります。

$$X^2 = \sum_{j=1}^{J}\left(\frac{\left(N_{ij}-\hat{m}_{ij}\right)^2}{\hat{m}_{ij}}+\frac{\left(N_{i'j}-\hat{m}_{i'j}\right)^2}{\hat{m}_{i'j}}\right)$$

p 値は、$p=\Pr\left(\chi_{J-1}^2 > X^2\right)$ によって算出されます。

順序型対象

X のカテゴリが I 個、Y の順序型カテゴリが J 個あると仮定しますX と Y の独立に関する帰無仮説が、Goodman (1979) によって提唱された行効果モデルに対して検定されます（行は X のカテゴリ、列は Y のクラス)。2 つのセットの期待セル度数 $\hat{m}_{ij}$（独立に対する帰無仮説）と $\hat{\hat{m}}_{ij}$（データが行効果モデルに従うという仮説）が推定されます。尤度比統計量は次のように計算されます。

$$H^2 = 2\sum_{i=1}^{I}\sum_{j=1}^{J}H_{ij}^2$$

とします。ここで、

$$H_{ij}^2 = \begin{cases}\hat{\hat{m}}_{ij}\ln\left(\hat{\hat{m}}_{ij}/\hat{m}_{ij}\right) & \hat{\hat{m}}_{ij}/\hat{m}_{ij}>0\\ 0 \quad else\end{cases}$$

p 値は、$p = \Pr\left(\chi^2_{I-1} > H^2\right)$ によって算出されます。

推定期待セル度数（独立仮説）

分析の重みが指定された場合、独立に対する帰無仮説での期待セル度数の形式は、次のようになります。

$$m_{ij} = \overline{w}_{ij}^{-1}\alpha_i\beta_j$$

ここで、 α_i および β_j は推定されるパラメーターで、$N_{ij} > 0$ の場合は、$\overline{w}_{ij} = \frac{W_{ij}}{N_{ij}}$ となり、そうでない場合は、$\overline{w}_{ij} = 1$ となります。

パラメータは $\hat{\alpha}_i$, $\hat{\beta}_j$ を推定し、したがって、$\hat{m}_{ij}$ は次の反復手順から取得されます。

1. $k = 0$, $\alpha_i^{(0)} = \beta_j^{(0)} = 1$, $m_{ij}^{(0)} = \overline{w}_{ij}^{-1}$
2. $\alpha_i^{(k+1)} = \frac{N_{i.}}{\sum_j \overline{w}_{ij}^{-1}\beta_j^{(k)}} = \alpha_i^{(k)}\frac{N_{i.}}{\sum_j m_{ij}^{(k)}}$
3. $\beta_j^{(k+1)} = \frac{N_{.j}}{\sum_i \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}}$
4. $m_{ij}^{(k+1)} = \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}\beta_j^{(k+1)}$
5. $\max_{i,j}\left|m_{ij}^{(k+1)} - m_{ij}^{(k)}\right| < \epsilon$（デフォルトは 0.001）の場合または反復数が閾値より大きい場合（デフォルトは 100）、出力 $\alpha_i^{(k+1)}, \beta_j^{(k+1)}$ および $m_{ij}^{(k+1)}$ を最終推定 $\hat{\alpha}_i, \hat{\beta}_j, \hat{m}_{ij}$ として停止します。そうでない場合は、$k = k + 1$ となり、ステップ 2 に進みます。

推定期待セル度数（行効果モデル）

行効果モデルの場合、Y のクラスのスコアが必要になります。デフォルトでは、s_j^*（Y のクラスの順序）がクラス スコアとして使用されます。これらの順次は、次の線型変換を使用して標準化され、最大スコアは 100 となり、最小スコアは 0 となります。

$$s_j = 100\left(s_j^* - s_{\min}^*\right) / \left(s_{\max}^* - s_{\min}^*\right)$$

ここで、$s_{\min}^*$ および $s_{\max}^*$ はそれぞれ最小の順序、および最大の順序です。

行効果モデルでの期待セル度数は、次のように計算されます。

$$m_{ij} = \overline{w}_{ij}^{-1}\alpha_i\beta_j\gamma_i$$

ここで、$\overline{s} = \sum_{j=1}^{J} W_{.j}s_j / \sum_{j=1}^{J} W_{.j}$, となり、$W_{.j} = \Sigma_i W_{ij}$、$\alpha_i$, β_j、および γ_i は推定される不明なパラメータです。

パラメータは $\hat{\alpha}_i, \hat{\beta}_j, \hat{\gamma}_i$ を推定します。したがって、$\hat{m}_{ij}$ は次の反復手順から取得されます。

1. $k = 0$, $\alpha_i^{(0)} = \beta_j^{(0)} = \gamma_i^{(0)} = 1$, $m_{ij}^{(0)} = \overline{w}_{ij}^{-1}$

2. $\alpha_i^{(k+1)} = \frac{N_{\cdot j}}{\sum_j \overline{w}_{ij}^{-1}\beta_j^{(k)}\left(\gamma_i^{(k)}\right)^{(s_j-\overline{s})}} = \alpha_i^{(k)}\frac{N_{i\cdot}}{\sum_j m_{ij}^{(k)}}$

3. $\beta_j^{(k+1)} = \frac{N_{\cdot j}}{\sum_i \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}\left(\gamma_i^{(k)}\right)^{(s_j-\overline{s})}}$

4. $m_{ij}^* = \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}\beta_j^{(k+1)}\left(\gamma_i^{(k)}\right)^{(s_j-\overline{s})}$, $G_i = 1 + \frac{\sum_j (s_j-\overline{s})\left(N_{ij}-m_{ij}^*\right)}{\sum_j (s_j-\overline{s})^2 m_{ij}^*}$

5. $\gamma_i^{(k+1)} = \begin{cases} \gamma_i^{(k)}G_i & G_i > 0 \\ \gamma_i^{(k)} & \text{そうでない場合} \end{cases}$

6. $m_{ij}^{(k+1)} = \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}\beta_j^{(k+1)}\left(\gamma_i^{(k+1)}\right)^{(s_j-\overline{s})}$

7. $\max_{i,j}\left|m_{ij}^{(k+1)} - m_{ij}^{(k)}\right| < \epsilon$（デフォルトは 0.001）の場合、または反復数が閾値より大きい場合（デフォルトは 100)、出力 $\alpha_i^{(k+1)}, \beta_j^{(k+1)}, \gamma_i^{(k+1)}$ および $m_{ij}^{(k+1)}$ を最終推定 $\hat{\alpha}_i, \hat{\beta}_j, \hat{\gamma}_i, \hat{m}_{ij}$ として停止します。そうでない場合は、$k = k+1$ となり、ステップ 2 に進みます。

非監視結合

対象がない場合、度数に基づいてカテゴリを結合します。X に、昇順でソートされたカテゴリが I 件あります。順序型予測値の場合、その値にしたがってソートし、名義型予測値の場合は、カテゴリを最小度数から最大度数の順に再配置します。データ値の昇順ソートまたは文字順ソートによってタイを解決します。c_i は、i 番目のカテゴリのケース数、N_X は X のケース総数となります。等しい同数の方法を使用して、まばらなカテゴリを結合します。

1. まず、$j_1 = j_2 = 1$、g=1となります。

2. $j_1 > I$ の場合は、ステップ 5 へ進みます。

3. $\sum_{i=j_1}^{j_2} c_i < [b\% \times N_X]$ の場合、$j_2 = j_2 + 1$となります。そうでない場合、元のカテゴリ $j_1, j_1+1, \cdots, j_2$ は新しいカテゴリ g に結合され、$j_1 = j_2 + 1$、$j_2 = j_1$ 、$g = g+1$ とし、ステップ 2 に進みます。

4. $j_2 \geq I$ の場合、次のルールのいずれかを使用してカテゴリを結合します。

   i）$g = 1$ の場合、カテゴリ $1, 2, \cdots, I-1$ はカテゴリ g 二結合され、I は結合されません。

   ii）g =2 の場合、$j_1, j_1+1, \cdots, I$ がカテゴリ g=2 に結合されます。

   iii）g> >2 の場合、$j_1, j_1+1, \cdots, I$ はカテゴリ $g-1$ に結合されます。

   $j_2 < I$ の場合は、ステップ 3 へ進みます。

5. 結合ルールと結合された予測値を出力します。

   結合後、次のルールいずれかを保持します。

- 元のカテゴリまたは結合時に作成されたカテゴリのいずれも $[b\% \times N_X]$ ケースより多くなります。b は、$1 < b < 100$（デフォルトは 10）を満たすユーザー指定のパラメータで、[x] は、x の最も近い整数を示します。
- 結合された予測値のカテゴリは 2 つだけです。

単変量統計の更新： 元のカテゴリ $j_1, j_1+1, \cdots, j_2$ がある新しいカテゴリに結合される場合、この新しいカテゴリのケース数は、$\sum_{i=j_1}^{j_2} c_j$ となります。結合手順の終わりに、各カテゴリの新規カテゴリとケース数を取得します。「単変量統計収集 」（ p. 15 ）の式に従ってそのほかの統計量を更新しますが、統計量は、新規カテゴリおよびこれらのカテゴリの数値型ケースの基づいて更新する必要があります。

連続型予測値の処理

連続型変数の処理には、対象がカテゴリの場合の監視分割、対象が連続型の場合の予測値選択、対象が連続型またはデータセットに対象がない場合の予測値構築が含まれます。

連続型予測値の処理後、「単変量統計収集 」（ p. 15 ）の指揮に従って取得されたまたは構築された予測値の単変量統計を収集します。定数である派生した予測値、またはすべての欠損値をもつ派生した予測値は、高度な分析から除外されます。

監視されたデータ分割

カテゴリ対象がある場合、監視されたデータ分割を使用して、連続型予測値を順序型予測値に変換します。カテゴリ対象と連続型予測値間の 2 変量統計をすでに収集しているものとします。「2 変量統計収集 」（ p. 22 ）で説明された表記を使用して、等質サブセットを次のように Scheffe 方法で識別します。

$|\overline{x}_{\cdot i} - \overline{x}_{\cdot j}| \le c_{critical}$ の場合、$\overline{x}_{\cdot i}$ および $\overline{x}_{\cdot j}$ が等質サブセットとなります。ここでは、$c_{critical} = \max(\overline{x}_{\cdot i}) - \min(\overline{x}_{\cdot i})$ で、そうでない場合は $N_{XY} = J$ となります。その場合 $c_{critical} = R * C$ および $R = \sqrt{2(J-1)F_{1-\alpha}(J-1, N_{XY}-J)}$ and $C = MS \times \sqrt{\frac{\sum_{i=1}^{J} 1/W_{\cdot i}}{J}}$, $MS = \sqrt{\frac{\sum_{i=1}^{J} M_{\cdot i}^2}{N_{XY}-J}}$.

監視アルゴリズムは次のようになります。

1. 平均値 $\overline{x}_{\cdot i}$ を昇順でソートし、$\overline{x}_{\cdot(1)} \le \overline{x}_{\cdot(2)} \le \cdots \le \overline{x}_{\cdot(J)}$ と表します。
2. まず、i=1 および q=J となります。
3. $\left|\overline{x}_{\cdot(q)} - \overline{x}_{\cdot(i)}\right| \le c_{critical}$ の場合、$\{\overline{x}_{\cdot(i)}, \cdots, \overline{x}_{\cdot(q)}\}$ を等質サブセットとみなすことができます。同時に、このサブセットの平均値および標準偏差を計算します。$\overline{x}_{\cdot(i,q)} = \frac{\sum_{k=i}^{q} W_{\cdot(k)} \overline{x}_{\cdot(k)}}{\sum_{k=i}^{q} W_{\cdot(k)}}$ および $sd_{\cdot(i,q)} = \sqrt{\frac{M_{(i,q)}^2}{\sum_{k=i}^{q} N_{\cdot(k)} - 1}}$、この場合 $M_{(i,q)}^2 = \sum_{k=i}^{q} A_{\cdot(k)}$ そして $A_{\cdot(k)} = M_{\cdot(k)}^2 + W_{\cdot(k)}\left(\overline{x}_{\cdot(i,q)} - \overline{x}_{\cdot(k)}\right)^2$となり、$i = q+1$ および $q = J$ を設定します。そうでない場合は $q = q-1$ となります。
4. $i \le J$ の場合、ステップ 3 に進みます。

5. ビンの分割点を計算します。$r \leq J$ の等質サブセットがあり、これらのサブセットの平均値が $\overline{x}^*_{\cdot(1)}, \overline{x}^*_{\cdot(2)}, \cdots, \overline{x}^*_{\cdot(r)}$ で標準偏差が $sd^*_{\cdot(1)}, sd^*_{\cdot(2)}, \cdots, sd^*_{\cdot(r)}$ と仮定すると、i 番目および（i+1）番目の等質サブセット間の分割点は、$cut_i = \overline{x}^*_{(i)} + \frac{sd^*_{(i)}+\epsilon}{\left(sd^*_{(i)}+sd^*_{(i+1)}+2\epsilon\right)}\left(\overline{x}^*_{(i+1)} - \overline{x}^*_{(i)}\right)$ によって算出されます。

6. データ分割ルールの出力カテゴリ ：$X \leq cut_1$、カテゴリ 2 ：$cut_1 < X \leq cut_2$；…、カテゴリ ：$cut_{r-1} < X$

フィールド選択と構築

連続型対象がある場合、予測値および対象間の相関または偏相関から派生した p 値を使用して予測値選択を実行します。そう感度が高い場合、選択された予測値はグループ化されます。各グループにでは、主成分分析を使用して新しい予測値を指定します。ただし、対象がない場合、予測フィールド選択は実行しません。

相関する予測値を識別するには、次のように、スケールおよびグループ間の相関を計算します。X は予測値で、予測値 $X_1, X_2, \cdots, X_m$ がグループ G を形成するものとします。X とグループ G との相関は次のように定義されます。

$$r_{XG} = \min\left\{\left|r_{XX_i}\right|, X_i \in G\right\}$$

この場合、r_{XX_i} は、X および X_i 間の相関です。

α_{group} は予測値がグループとして識別されるレベルの相関となります。予測フィールド選択と予測フィールド構築アルゴリズムは次のようになります。

1. （対象が連続型で予測フィールド選択が有効である）連続型予測値と対象間の p 値が閾値より大きい場合（デフォルトは 0.05）、この予測値を相関行列および共分散行列から削除します。これらの p 値の計算の詳細については、「相関と偏相関 」（ p. 34 ）を参照してください。

2. まず、$\alpha_{group} = 0.9$ および i =1 とします。

3. $\alpha_{group} \leq 0.1$ の場合、すべての派生した予測値、入力予測値、各入力予測値の相関を停止して出力します。また、相関行列内の残りの予測値を出力します。

4. 2 つの最も相関する予測値を検索すると、絶対値の相関は、α_{group} より大きくなり、グループ i に含められます。選択する予測値がない場合、ステップ 9 へ進みます。

5. ある予測値をグループ i に追加すると、予測値はグループ i と最も相関し、相関は α_{group} より大きくなります。グループ i の予測値の数が閾値より大きくなるまで（デフォルトは 5）、または選択する予測値がなくなるまでこの手順を繰り返します。

6. 新しい予測値を、主成分分析を使用してグループ i から派生させます。 詳細は、p. 34 主成分分析 を参照してください。

7. （予測フィールド選択および予測フィールド構築がどちらも有効）新しい予測値の値を制御し、そのほかの連続型予測値と対象間の偏相関を計算します。また、偏相関に基づいて、p 値を計算します。これらの p 値の計算の詳細については、「相関と偏相関 」（ p. 34 ）を参照してください。連続型予測変数と連続型対象間の偏相関に

基づいた p 値が閾値より大きい場合（デフォルトは 0.05)、この予測値を相関行列および共分散行列から削除します。

8. グループ内にある予測値も相関行列から削除します。i =i+1 とし、ステップ 4 に進みます。

9. $\alpha_{group} = \alpha_{group} - 0.1$ の場合、ステップ 3 に進みます。

注:

- 予測フィールド選択のみが必要な場合、ステップ 1 のみが実行されます。予測フィールド構築のみが必要な場合、ステップ 1 およびステップ 7 以外のすべてのステップを実行します。予測フィールド選択および予測フィールド構築のいずれも必要な場合は、すべてのステップを実行します。
- 高度に相関する予測値を識別する場合、相関にタイがある場合、データセットの最小インデックスを持つ予測値を選択してタイを解決します。

主成分分析

$X_1, X_2, \cdots, X_m$ を、m 個の連続型予測値とします。主成分分析は、次のように説明されます。

1. 入力 $C_{m\times m}$、$X_1, X_2, \cdots, X_m$ の共分散行列。
2. 共分散行列の固有ベクトルと固有値を計算します。固有値（および対応する固有ベクトル）を降順にソートして $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ とします。
3. 新しい予測値を指定します。最初の成分 v_1 の要素が $v_{11}, v_{12}, \cdots, v_{1m}$ であると仮定すると、新しく派生した予測値は $\frac{v_{11}}{\sqrt{\lambda_1}}X_1 + \frac{v_{12}}{\sqrt{\lambda_1}}X_2 + \cdots + \frac{v_{1m}}{\sqrt{\lambda_1}}X_m$ となります。

相関と偏相関

相関と P 値

r_{XY} を連続型予測値 X および連続型対象 Y 間の相関とすると、p 値は、t 検定から次のように派生します。

$$p = \Pr\left(\left|t\left(N_{XY} - 2\right)\right| > t\right)$$

この場合、$t(N_{XY} - 2)$ は、自由度が $N_{XY} - 2$ の t 分布の無作為変数で、$t = r_{XY}\sqrt{\frac{N_{XY}-2}{1-r_{XY}^2}}$ となります。$r_{XY}^2 = 1$ の場合、p =0 を設定し、$N_{XY} \leq 2$ の場合、p=1 と設定します。

偏相関と P 値

2 つの連続型変数、X および Y について、新しい連続型変数 Z の値を制御して、変数間の偏相関を次のように計算します。

$$r_{XY|Z} = \frac{r_{XY} - r_{XZ}r_{YZ}}{\sqrt{1 - r_{XZ}^2}\sqrt{1 - r_{YZ}^2}}$$

新しい変数 Z は常に、複数の連続型変数の線型結合であるため、元のデータセットではなく共分散のプロパティを使用して、Z および連続型変数の相関を計算します。新しく派生した予測値 Z は元の予測値 $X_1, X_2, \cdots, X_m$ の線型結合です。

$$Z = a_1X_1 + a_2X_2 + \cdots + a_mX_m$$

連続型変数 X（連続型予測値または連続型対象）の場合、X と Z の相関は次のようになります。

$$r_{ZX} = \frac{c_{ZX}}{\sqrt{c_{ZZ}c_{XX}}}$$

この場合、$c_{ZX} = \sum_{i=1}^{m} a_i c_{X_iX}$、および $c_{ZZ} = \sum_{i=1}^{m} a_i^2 c_{X_iX_i} + 2\sum_{i \neq j} a_i a_j c_{X_iX_j}$ となります。

$1 - r_{XZ}^2$ or $1 - r_{YZ}^2$ が 10^{-10} より小さい場合、$r_{XY|Z} = 0$ となります。$r_{XY|Z}$ が 1 より大きい場合、1 に設定します。$r_{XY|Z}$ が −1 より小さい場合、−1 に設定します(ペアごとの削除で発生する場合があります)。偏相関に基づいて、p 値 はt 検定から次のように派生します。

$$p = \Pr\left(|t\left(N_{XY} - 3\right)| > t\right)$$

この場合、$t(N_{XY} - 3)$ は、自由度が $N_{XY} - 3$ の t 分布の無作為変数で、$t = r_{XY|Z}\sqrt{\frac{N_{XY}-3}{1-r_{XY|Z}^2}}$ となります。$r_{XY|Z}^2 = 1$ の場合、p=0 を設定し、$N_{XY} \leq 3$ の場合、p=1 を設定します。

連続型予測フィールドの離散化

離散化を使用して、予測の精度を計算し、ヒストグラムを作成します。

予測精度を計算するための離散化

変換された対象がカテゴリの場合、等幅分割方法を使用して、連続型予測変数を対象のカテゴリ数に等しい分割に離散化します。離散化を検討される変数には次のようなものがあります。

- 推奨されたスケール予測フィールド。
- 推奨された予測フィールドの元の連続変数。

ヒストグラムを作成するための離散化

等幅分割方法を使用して、連続型予測フィールドを最大 400 ビンに離散化します。離散化を検討される変数には次のようなものがあります。

- 推奨された連続型変数。
- 新しい変数の派生に使用されていない除外された連続型変数。
- 推奨された変数の元の連続変数。
- 新しい変数の派生に使用されていない除外された変数の元の連続型変数。
- 新しい変数の構築に使用されるスケール変数。元の変数も連続型である場合、元の変数は離散化されます。
- 日付/時間変数。

離散化後、各ビンのケースおよび平均値の数を収集してヒストグラムを作成します。

注 ： 元の予測フィールドを変更した場合、この変更されたバージョンは「元の」予測フィールドとみなされます。

予測精度

予測精度の 2 変量統計の収集

推奨された予測変数と（変換された）対象間の 2 変量統計を収集します。推奨された予測フィールドに元の予測フィールドがある場合、元の予測フィールドと対象との間の 2 変量統計も収集します。元の予測フィールドに変更されたバージョンがある場合、変更されたバージョンを使用します。

対象がカテゴリであるにもかかわらず、推奨された予測フィールドまたは元の予測フィールド/変更バージョンが連続型である場合、「連続型予測フィールドの離散化 」（ p. 35 ）の方法で連続型予測フィールドを離散化し、カテゴリ対象とカテゴリ予測フィールド間の 2 変量統計を収集します。

予測フィールドと対象との 2 変量統計は、「2 変量統計収集 」（ p. 22 ）で説明されている 2 変量統計と同じです。

予測精度の計算

予測精度を使用して、予測の有用性を促成し、（変換された）対象について計算されます。推奨された予測フィールドに元の予測フィールドがある場合、元の予測フィールドの予測精度も計算します。元の予測フィールドに変更されたバージョンがある場合、変更されたバージョンを使用します。

尺度対象： 対象が連続型の場合、線型回帰モデルに適合し、予測精度は次のように計算されます。

- 尺度予測$r_{XY}^2 = \left(\frac{c_{XY}}{\sqrt{c_{XX}}\sqrt{c_{YY}}}\right)^2$
- カテゴリ予測 ：$1 - \frac{S_e}{S_T}$で、この場合、$S_e = \sum_{i=1}^{I} M_{i.}^2$および $S_T = \sum_{i=1}^{n} f_i w_i (y_i - \overline{y}_x)^2$ となります。

カテゴリ対象：（変換された）対象がカテゴリである場合、Naïve Bayes モデルを適合し、分類制度が予測精度として機能します。連続型予測フィールドを 「連続型予測フィールドの離散化 」（ p.35 ）で説明しているように離散化すると、カテゴリ予測フィールドの予測精度のみを考慮します。

N_{ij} が $X = i$ および $Y = j$、$N_{i.} = \sum_{j=1}^{J} N_{ij}$、および $N_{.j} = \sum_{i=1}^{I} N_{ij}$ であるケース数である場合、カイ 2 乗統計は、次のように算出されます。

$$\chi^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{\left(N_{ij} - \hat{N}_{ij}\right)^2}{\hat{N}_{ij}}$$

ここで、$\hat{N}_{ij} = \frac{N_{i.} N_{.J}}{N_{XY}}$

および Cramer が V 次のように定義されます。

$$V = \left(\frac{\chi^2}{N_{XY} \left(\min\left(I, J\right) - 1\right)} \right)^{1/2}$$

参照

Box, G. E. P., および D. R. Cox. 1964. An analysis of transformations. Journal of the Royal Statistical Society, Series B, 26, 211-246.

Goodman, L. A. 1979. Simple models for the analysis of association in cross-classifications having ordered categories. Journal of the American Statistical Association, 74, 537-552.

章 5

Bayesian Network のアルゴリズム

Bayesian Network のアルゴリズムの概要

Bayesian Network は、与えられた一連のランダム変数関する結合確率分布を簡潔に説明する方法を提供します。

V を一連のカテゴリ ランダム変数とし、G = (V, E) をノード V と一式の有向辺 E のある有向非巡回グラフとします。Bayesian Network モデルは、グラフ G と、親ノードの値が与えられた各ノードの条件付き確率表からなります。親の値を与えられた各ノードは、その子孫でないすべてのノードからは独立しているとみなされます。変数 V の結合確率分布は、各ノードの親の値を与えられたすべてのノードの条件付き確率の積として計算されます。

与えられた一連の変数 V と対応するサンプル データセットを使用して、適切な Bayesian Network モデルをあてはめるタスクが示されます。グラフ G 内の適切な辺を決定するタスクは **構造学習**と呼ばれます。一方、各ノードに親の値が与えられた条件付き確率表を推定するタスクは**パラメータ学習**と呼ばれます。

一次計算

IBM® SPSS® Modeler には、Bayesian Network モデルを構築する方法が 2 つ用意されています。

- **Tree Augmented Naïve Bayes :** このアルゴリズムは、主に分類に使用されます。このアルゴリズムでは、単純な Bayesian Network モデルが効果的に作成されます。このモデルは Naïve Bayes モデルの改良型で、各予測値は対象変数に加えて別の予測値に依存することもできます。主な利点は、その分類の精度で、一般的な Bayesian Network モデルと比較してパフォーマンスに秀でています。またその単純さも利点の 1 つで、ノード間で見つかった依存構造に対してより大きい制限を課します。
- **Markov Blanket 推定 :** Bayesian Network での対象変数ノードの Markov Blanket は、対象の親、子、および子の親を含む一連のノードです。Markov Blanket は対象変数を予測するために必要なネットワークのすべての変数を識別します。これによって、より複雑なネットワークが生成される可能性がありますが、同時に、生成するのにより長い時間がかかります。フィールド選択処理を使用すると、このアルゴリズムのパフォーマンスをかなり向上させることができます。

表記

次の表記がこのアルゴリズムの説明で使用されます。

G	Bayesian Network モデルを表す有向非巡回グラフ
D	データセット
Y	カテゴリの対象変数
x_i	i 番目の予測値
π_i	対象 Y のほかの i 番目の予測値の親セット。TAN モデルの場合、このサイズは ≤1 です。
N	D 内のケース数
n	予測値の数
N_{ijk}	(π_i, Y) がその j 番目の値をとり、 X_i がその k 番目の値をとる D 内のレコード数を表します。
N_{ij}	(π_i, Y) がその j 番目の値をとる D 内のレコード数を表します。
θ_{ijk}	$\Pr\left(X_i = x_i^k \mid (\pi_i, Y) = (\pi_i, Y)^j\right)$
θ_{Y_i}	$\Pr(Y = Y_i)$
K	TAN の非冗長パラメータの数
MB	対象 Y に関する Markov Blanket 境界
S	X のサブセット
$S_{X_i X_j}$	変数 X_i と X_j が、$S_{X_i X_j}$ に関して条件的に独立するような、$X \setminus X_i, X_j$ のサブセット
$X_i - X_j$	G 内の変数 X_i, X_j 間の無向アーク。X_i と X_j は互いに隣接しています。
$X_i \rightarrow X_j$	G 内の X_i から X_j への有向アーク。X_i は X_j の親で、X_j は X_i の子です。
ADJ_{X_i}	辺の方向を無視した、G 内の変数 X_i のすべての隣接変数を表した変数セット。
$I(\cdot)$	検定の p 値を返す条件付き独立性（CI）検定の関数。
α	2 変数間の CI 検定の有意水準。検定の p 値が α より大きい場合、それらは独立しています。また、その逆も真です。
r_i	X_i のカーディナリティ。$r_i = \lvert X_i \rvert$
q_i	X_i の親セット π_i のカーディナリティ。

連続型予測値の処理

IBM® SPSS® Modeler の BN モデルでは、離散型の変数のみを使用できます。対象変数は離散型でなければなりません（フラグ型またはセット方）。BN モデルが構築される前に、数値型予測値は 5 つの等幅のビンに離散化されます。構築されたビンのいずれかが空（ビンの範囲内に値を持ったレコードがない）の場合、そのビンは隣接する空でないビンと結合されます。

幅優先検索を使用したフィールド選択

フィールド選択処理は次のように行われます。

- まず、独立性の統計検定を基に、与えられた対象Y に直接隣接する変数を検索します。詳細は、 p. 45 Markov Blanket 条件付き独立性検定 を参照してください。 これらの変数は Y の親または子とされるもので、$PC(Y)$ で表されます。
- 各 $X \in PC(Y)$ について、$PC(X)$、つまり X の親および子を検索します。
- 各 $Z \in PC(X)$ について、それが Y から独立していなければ、それを MB_Y に追加します。

明確なアルゴリズムを次に示します。

```
RecognizeMB
(
 D :Dataset, eps :threshold
)
{
  // Recognize Y's parents/children
  CanADJ_Y = X \ {Y};
  PC = RecognizePC(Y,CanADJ_Y,D,eps);
  MB = PC;

  // Collect spouse candidates, and remove false
  // positives from PC
  for (each X_i in PC){
    CanADJ_X_i = X \ X_i;
    PC = RecognizePC(Y,CanADJ_Y,D,eps);
    if (Y notin CanSP_X_i)   // Filter out false positive
      MB = MB \ X_i;
  }
  // Discover true positives among candidates
  for (each X_i in MB)
    for (each Z_i in CanSP_X_i and Z_i notin MB)
      if (I(Y,Z_i|{S_Y,Z_i + X_i}) ≤ eps) then
        MB = MB + Z_i;
  return MB;
}
```

```
RecognizePC (
 T       :target to scan,
 ADJ_T    :Candidate adjacency set to search,
 D :Dataset,
 eps      :threshold,
 maxSetSize :)
{
   NonPC = {empty set};
   cutSetSize = 0;
   repeat
      for (each X_i in ADJ_T){
         for (each subset S of {ADJ_T \ X_i} with |S| = cutSetSize){
            if (I(X_i,T|S) > eps){
               NonPC = NonPC + X_i;
               S_T,X_i = S;
               break;
            }
         }
      }
      if (|NonPC| > 0){
         ADJ_T = ADJ_T \ NonPC;
         cutSetSize +=1;
         NonPC = {empty set};
      } else
         break;
   until (|ADJ_T| ≤ cutSetSize) or (cutSetSize > maxSetSize)
   return ADJ_T;
}
```

Tree Augmented Naïve Bayes 手法

Bayesian Network 分類器は、単純な分類の手法で、ケース $d_j = \left(x_1^j, x_2^j, \ldots, x_n^j\right)$ を、それが i 番目の対象カテゴリ Y_i に属する確率を決定することによって分類します。これらの確率は、次のように計算されます。

$$\begin{aligned}
&\Pr\left(Y_i | X_1 = x_1^j, X_2 = x_2^j, \ldots, X_n = x_n^j\right) \\
&= \frac{\Pr(Y_i)\Pr\left(X_1=x_1^j, X_2=x_2^j, \ldots, X_n=x_n^j | Y_i\right)}{\Pr\left(X_1=x_1^j, X_2=x_2^j, \ldots, X_n=x_n^j\right)} \\
&\propto \Pr\left(Y_i\right) \prod_{k=1}^{n} \Pr\left(X_k = x_k^j | \pi_k^j, Y_i\right)
\end{aligned}$$

ここで、π_k は、Y のほかの X_k の親セットで、空の場合があります。$\Pr\left(X_k | \pi_k, Y\right)$ は、各ノード X_k に関連付けられた条件付き確率表（CPT）です。n 個の独立した予測値がある場合、確率は次に比例します。

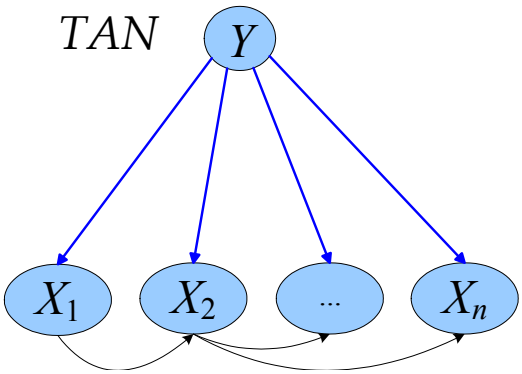
$$\Pr(Y_i)\prod_{k=1}^{n}\Pr\left(X_k = x_k^j | Y_i\right)$$

この依存の仮定（クラスを与えられた予測値間の条件付き独立性)がなされた場合、この分類器は Naïve Bayes (NB) と呼ばれます。Naïve Bayes は、より複雑な最新の分類器にも匹敵することを示してきました。ここ数年、Naïve Bayes 分類器の改良に焦点を合わせた研究がさかんです。1 つの重要な手法に、独立性の仮定を緩和する方法があります。Tree Augmented Naïve Bayesian (TAN) 分類器 (Friedman(F), Geiger, および Goldszmidt, 1997)を使用した場合、次の条件によって定義されます。

- 各予測値には親としての対象がある。
- 予測値には親としてのもう 1 つの別の予測値がある場合がある。

この構造の例を次に示します。

図 5-1
単純な Tree Augmented Naïve Bayes (TAN) モデルの構造



TAN 分類器の学習プロシージャ

$\mathbf{X} = (X_1, X_2, \ldots, X_n)$ をカテゴリ予測値ベクトルとします。TAN 分類器のアルゴリズムはまず、Y に対して条件とされている相互情報量を使用して、$\mathbf{X}$ に関するツリー構造を学習します。その後、対象ノードから各予測値ノードにリンク（またはアーク）を追加します。

TAN 学習プロシージャは次のとおりです。

1. 学習データ D、$\mathbf{X}$、および Y を入力としてとります。
2. 次のような構造学習のアルゴリズムを使用することによって、$\mathbf{X}$ に関するツリーのようなネットワーク構造を学習します。
3. $1 \le i \le n$ である各 X_i の親として Y を追加します。
4. TAN ネットワークのパラメータの学習

TAN 構造学習

MWST (maximum weighted spanning tree) 手法を使用して、データからツリー Bayesian Network を構築します(Chow および Liu, 1968)。この手法は、2 つの変数間の相互情報量に対応する重みを各辺に関連付けます。重み行列が作成されたら、MWST のアルゴリズム (Prim, 1957) によって、任意のルートから開始できる無向ツリーが得られます。

2 つのノード X_i, X_j の相互情報量は、次のように定義されます。

$$I(X_i, X_j) = \sum_{x_i, x_j} \Pr(x_i, x_j) \log\left(\frac{\Pr(x_i, x_j)}{\Pr(x_i)\Pr(x_j)}\right)$$

2 つの予測値間の相互情報量を、対象を与えられた 2 つの予測値間の条件付き相互情報量と置き換えます(Friedman(F) など, 1997)。これは、次のように定義されます。

$$I(X_i, X_j|Y) = \sum_{x_i, x_j, y_k} \Pr(x_i, x_j, y_k) \log\left(\frac{\Pr(x_i, x_j|y_k)}{\Pr(x_i|y_k)\Pr(x_j|y_k)}\right)$$

ネットワークは次の手順に従って構築できます。

1. 変数の各ペア間で $I(X_i, X_j|Y), i = 1, \ldots, n, j = 1, \ldots, n, i \neq j$ を計算します。
2. Prim のアルゴリズム(Prim など, 1957)を使用して、$I(X_i, X_j|Y)$ によって、X_i と X_j を接続する辺の重みを持つ MWST を構築します。

   このアルゴリズムは、次のように機能します。辺のないツリーから開始し、変数を、入力として無作為にマークします。マークした変数の 1 つによる重みが最大であるマークされていない変数を見つけると、この変数にマークをつけ、ツリーに辺を追加します。このプロセスは、すべての変数にマークが付けられるまで繰り返されます。
3. X_1 をルート ノードとして選択し、すべての辺の方向をそこから外側に向くよう設定することによって、結果の無向ツリーを有向ツリーに変換します。

TAN パラメータ学習

r_i を X_i のカーディナリティとします。q_i で、X_i の親セット (π_i, Y) のカーディナリティ、つまり、X_i の親がインスタンス化されるさまざまな値の数を表すこととします。これは、$q_i = r_{\pi_i} \times |Y|$ のように計算されます。$\pi_i = \emptyset$ は $q_i = |Y|$ を示すことに注意してください。N_{ij} を使用して、(π_i, Y) が j 番目の値をとる場合の D 内のレコード数を表します。N_{ijk} を使用して、(π_i, Y) が j 番目の値をとり、X_i が k 番目の値をとる場合の D 内のレコード数を表します。

最尤推定

対数尤度スコアを最大にするような、パラメータ $\theta_{Y_i}\,(1 \leq i \leq |Y|)$ と $\theta_{ijk}\,(1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i)$ の閉形式解は次のとおりです。

$\hat{\theta}_{Y_i} = \frac{N_{Y_i}}{N}$
$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}}$

ここで N_{Y_i} は、学習データ内の $Y = Y_i$ のケース数を表します。

$N_{ij} = 0$ であれば、$\hat{\theta}_{ijk} = 0$ となることに注意してください。

パラメータ K の数は、次のとおりです。

$$K = \sum_{i=1}^{n} (r_i - 1) \cdot q_i + |Y| - 1$$

TAN 事後評価

パラメータ セット $\theta_{Y_i} (1 \le i \le |Y|)$、および $\theta_{ijk} (1 \le k \le r_i)$、$1 \le i \le n$、$1 \le j \le q_i$ の各セットに、Dirichlet 事前分布が指定されたと仮定します (Heckerman, 1999)。$N^0_{Y_i}$ と N^0_{ijk} で、$N^0 = \sum_i N^0_{Y_i}$ および $N^0_{ij} = \sum_k N^0_{ijk}$ のような対応する Dirichlet 分布パラメータを表すこととします。データセット D を観測するにあたって、次のパラメータ セットを使用して Dirichlet 事前分布を取得します。

$\hat{\theta}^P_{Y_i} = \frac{N_{Y_i} + N^0_{Y_i}}{N + N^0}$
$\hat{\theta}^P_{ijk} = \frac{N_{ijk} + N^0_{ijk}}{N_{ij} + N^0_{ij}}$

事後評価は、モデルの更新に常に使用されます。

小さいセルの度数の調整

ゼロまたは非常に小さいセルの度数によって起こる問題を解決するために、非情報的な Dirichlet 事前分布 $N^0_{Y_i} = \frac{2}{|Y|}$ と $N^0_{ijk} = \frac{2}{r_i \cdot q_i}$ を使用して、パラメータを事後パラメータ $\hat{\theta}^P_{Y_i} (1 \le i \le |Y|)$ および $\hat{\theta}^P_{ijk} (1 \le k \le r_i), 1 \le i \le n, 1 \le j \le q_i$ として推定できます。

Markov Blanket のアルゴリズム

Markov Blanket のアルゴリズムは、変数間の条件付き独立性の関係を識別することによって BN 構造を学習します。また、統計検定（カイ 2 乗検定や G 検定など）を使用して、ノード間の条件付き独立性の関係を見つけ出し、これらの関係を制約として使用して BN 構造を構築します。このアルゴリズムは、独立性分析ベースまたは制約ベースのアルゴリズムの一種です。

Markov Blanket 条件付き独立性検定

条件付き独立性（CI）検定は、2 つの変数が条件変数セットに対して条件的に独立しているかどうかを検定します。CI 検定の計算法には、2 つの似通った手法があります。χ^2（Pearson カイ 2 乗）検定と G^2（対数尤度比）検定です。

X,Y を検定する 2 つの変数とし、S を $X,Y \notin S$ である条件変数セットと仮定します。$O(x_i,y_j)$ を、$X=x_i$ と $Y=y_j$ を持つケースの観測された度数とし、$E(x_i,y_j)$ を、$X=x_i$ と $Y=y_j$ を持つケースの正確な数とします。このとき、X,Y は独立していると仮定します。

カイ 2 乗検定

帰無仮説を X,Y は独立しているとします。この仮説での χ^2 検定の統計量は次のようになります。

$$\chi^2(X,Y)=\sum_{i,j}\frac{(O(x_i,y_j)-E(x_i,y_j))^2}{E(x_i,y_j)}$$

N を D 内のケースの総数、$N(x_i)$ を X_i がその i 番目のカテゴリをとる D 内のケースの数、そして、$N(y_j)$ と $N(s_k)$ を Y と S に対応する数とします。つまり、$N(x_i,y_j)$ は、X_i がその i 番目のカテゴリをとり、Y_j がその j 番目のカテゴリをとる D 内のケース数となります。$N(x_i,s_k)$、$N(y_j,s_k)$、および $N(x_i,y_j,s_k)$ は同様に定義されます。次のようになります。

$$\chi^2(X,Y)=\sum_{i,j}\frac{(N(x_i,y_j)-N(x_i)\,N(y_j)/N)^2}{N(x_i)\,N(y_j)/N}=\sum_{i,j}\frac{(N\cdot N(x_i,y_j)-N(x_i)\,N(y_j))^2}{N(x_i)\,N(y_j)\cdot N}$$

$\chi^2(X,Y)\sim\chi^2_v$（ここで $v=(|X|-1)(|Y|-1)$）は χ^2 分布の自由度であるため、$\chi^2(X,Y)$ の p 値は次のように得られます。

$$P\left(U>\chi^2(X,Y)\right)$$

すでに確認したように、p 値が大きいほど、帰無仮説が否定される可能性は少なくなります。与えられた有意水準 α に対して、p 値が α より大きい場合、X,Y が独立しているという仮説を否定することはできません。

この独立性検定を、次のように簡単に条件付き独立性検定に一般化することができます。

$$\chi^2(X,Y|S)=\sum_k\chi^2(X,Y|S=s_k)$$

$$=\sum_{i,j,k}\frac{(N(x_i,y_j,s_k)\,N(s_k)-N(x_i,s_k)\,N(y_j,s_k))^2}{N(x_i,s_k)\,N(y_j,s_k)\,N(s_k)}$$

$\chi^2 \sim \chi^2_v$ の自由度は次のとおりです。

$$\nu = (|X| - 1)(|Y| - 1) \cdot |S|$$

尤度比検定

帰無仮説を X,Y は独立しているとします。この仮説での G^2 検定の統計量は次のようになります。

$$G^2(X,Y) = 2\sum_{i,j} O(x_i, y_j) \ln\left(\frac{O(x_i, y_j)}{E(x_i, y_j)}\right)$$

または

$$G^2(X,Y) = 2\sum_{i,j} N(x_i, y_j) \ln\left(\frac{N(x_i, y_j) N}{N(x_i) N(y_j)}\right)$$

条件付きの G^2 独立性検定の場合は、次のようになります。

$$G^2(X,Y|S) = 2\sum_{i,j,k} O(x_i, y_j|S = s_k) \ln\left(\frac{O(x_i, y_j|S = s_k)}{E(x_i, y_j|S = s_k)}\right)$$

$$= 2\sum_{i,j,k} N(x_i, y_j, s_k) \ln\left(\frac{N(x_i, y_j, s_k) N(s_k)}{N(x_i, s_k) N(y_j, s_k)}\right)$$

この G^2 検定は χ^2_v 分布として漸近的に分布されます。このとき、自由度は χ^2 検定の場合と同じです。したがって、G^2 検定の p 値は次のようになります。

$$P\left(U > G^2(X,Y)\right)$$

このドキュメントでは以降、$I(\cdot)$ を使用して、適用されるすべての検定における p 値を一様に示します。$I(X,Y) > \alpha$ の場合、変数 X と Y は独立していることを示します。$I(X,Y|S) > \alpha$ の場合、変数 X と Y は与えられた変数セット S に対して条件的に独立していることを示します。

Markov Blanket 構造学習

このアルゴリズムは、データセットから Bayesian Network 構造を学習することを目的としています。完全なグラフ G から開始します。$X_i, X_j \in \mathbf{X}$ とし、G 内の各変数ペアについて $I(X_i, X_j)$ を計算します。$I(X_i, X_j) > \alpha$ の場合、X_i, X_j 間のアークを削除します。そして、各アークについて、$X_i - X_j$ で $ADJ_{X_i} \setminus \{X_j\}$ 内の全数検索を行って、$I(X_i, X_j|S) > \alpha$ となる最も小さい条件変数セット S を探します。そのような S が存在すれば、アーク $X_i - X_j$ を削除します。その後、方向のルールが適用されて G 内のアークは方向づけられます。

Markov Blanket アークの方向のルール

作成された構造内のアークは、次のルールに基づいて方向付けられます。

1. $X_i - X_j - X_k$ 形式または $X_i \to X_j - X_k$ 形式のすべてのパターンは、$X_j \notin S_{X_i X_j}$ の場合、$X_i \to X_j \leftarrow X_k$ に更新されます。
2. $X_i \to X_j - X_k$ 形式のパターンは、$X_j \to X_k$ となるように更新されます。
3. $X_i - X_j$ 形式のパターンは、$X_i \to X_j$ に更新されます。
4. 形式のパターン

$$\begin{array}{ccccc} X_i & - & X_j & - & X_k \\ & \searrow & \downarrow & \swarrow & \\ & & X_l & & \end{array}$$

は、$X_j \to X_l$ となるように更新されます。

最後のステップのあとにグラフ内にまだ無向アークが残っている場合、ステップ 2 に戻り、すべてのアークに方向がつくまで繰り返します。

Markov Blanket 構造の作成

Markov Blanket は、Bayesian Network のローカル構造の 1 つです。Bayesian Network G と対象変数 Y が与えられ、Y の Markov Blanket を導き出すには、π_Y で表される G 内の Y のすべての有向の親、X_{Ch} で表される G 内の Y のすべての有向の子、および π で表される G 内のX_{Ch} のすべての有向の親を選択する必要があります。$\pi_Y \cup Y \cup X_{Ch} \cup \pi$ と、G から継承されたそれらのアークが、Markov Blanket MB_Y を定義します。

Markov Blanket パラメータ学習

最尤推定

対数尤度スコアを最大化する、パラメータ$\theta_{ijk}\,(1 \le i \le n, 1 \le j \le q_i, 1 \le k \le r_i)$ の閉形式解は、次のとおりです。

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}}$$

$\pi_i = \emptyset$ の場合、$\hat{\theta}_{ijk} = \frac{N_k}{N}$ となることに注意してください。

パラメータ K の数は、次のとおりです。

$$K = \sum_{i=1}^{n} (r_i - 1) \cdot q_i$$

事後評価

$\theta_{ijk}\,(1 \le k \le r_i), 1 \le i \le n, 1 \le j \le q_i$ の各セットに Dirichlet 事前分布が指定されたと仮定します (Heckerman など, 1999)。N^0_{ijk} で、$N^0_{ij} = \sum_k N^0_{ijk}$ のような対応する Dirichlet 分布のパラメータを表すこととします。データセット D を観測するにあたって、次のパラメータ セットを使用して Dirichlet 事前分布を取得します。

$$\hat{\theta}^P_{ijk} = \frac{N_{ijk} + N^0_{ijk}}{N_{ij} + N^0_{ij}}$$

事後評価は、モデルの更新に常に使用されます。

小さいセルの度数の調整

ゼロまたは非常に小さいセルの度数によって起こる問題を解決するために、パラメータは、$N^0_{ijk} = \frac{2}{r_i \cdot q_i}$ で指定される非情報的な Dirichlet 事前分布を使用して、事前パラメータ$\theta_{ijk}\,(1 \le k \le r_i), 1 \le i \le n, 1 \le j \le q_i$ として推定できます。

空白の処理

デフォルトでは、出力フィールドまたは出力フィールドに欠損値があるレコードは、モデルの作成から除外されます。[**完全なレコードのみ使用**] オプションの選択が解除されている場合、フィールド間の各ペアごとの比較で、問題の 2 つのフィールドに有効な値が含まれているレコードはすべて使用されます。

モデル ナゲット／スコアリング

Bayesian Network モデル ナゲットは、スコアリングされたレコードの予測値および確率を生成します。

Tree Augmented Naïve Bayes モデル

学習データから推定されたモデルを使用すると、新しいケース $\mathbf{x} = (x_1, \ldots, x_n)$ に関して、それが i 番目の対象カテゴリ Y_i に属する確率は、$\Pr(Y = Y_i | \mathbf{X} = \mathbf{x})$ で計算されます。事後確率が最も大きい対象カテゴリは、このケース $Y(\mathbf{x})$ の予測カテゴリで、次のように予測されます。

$$\begin{aligned}
&\hat{Y}(\mathbf{x}) = arg\max_i \{\Pr(Y = Y_i | \mathbf{X} = \mathbf{x})\} \\
&= arg\max_i \{\Pr(\mathbf{X} = \mathbf{x} | Y = Y_i) \Pr(Y = Y_i)\} \\
&= arg\max_i \left\{ \Pr(Y = Y_i) \prod_{i=1}^{n} \Pr(X_i = x_i | \pi_i = \pi_i, Y = Y_i) \right\}
\end{aligned}$$

Markov Blanket モデル

スコアリング関数は、推定されたモデルを使用して、新しいケース X_P の各カテゴリに属する Y の確率を計算します。π_Y を Y の親セットとし、$\pi_{Y|P}$ で与えられたケース X_P のπ_Y の構成を表すこととし、$X_{Ch} = (X_1, \ldots, X_m)$ で Y の有向子セットを表すこととし、π_i で X_{Ch} の i 番目の変数の親セット（Y を除く）を表すこととします。Y の各カテゴリのスコアは、次のように計算されます。

$$\Pr\left(Y = y_l | X_P = x_P\right) = \frac{Pr\left(Y = y_l, X_P = x_P\right)}{\sum_{y_l} Pr\left(Y = y_l, X_P = x_P\right)}$$

ここで、$Y = y_l$ と $X_P = x_P$ の結合確率は、次のようになります。

$$\Pr\left(Y = y_l, X_P = x_P\right) = c \cdot Pr\left(Y = y_l | \pi_Y = \pi_{y|P}\right) \prod_{i=1}^{m} Pr\left(X_i = x_i | \pi_i = \pi_{i|P}, Y = y_l\right)$$

とします。ここで、

$$c = \Pr\left(\pi_Y = \pi_{y|P}\right) \prod_{i=1}^{m} Pr\left(\pi_i = \pi_{i|P}\right)$$

値が上記で指定されたスコアリング方程式の分子および分母から除外されるため、c はスコアリング時に自動的に計算されません。

章 6

2 値の分類の比較メトリック

2 値の分類ノードで、フラグ型出力フィールド用の複数のモデルが生成されます。各モデル タイプがどのように構築されるかの詳細は、それぞれのタイプに応じた適切なアルゴリズム ドキュメントを参照してください。

ノードも、アプリケーションに適したモデルを選択する上で役立つように、各モデルについてのいくつかの比較メトリックを報告します。次のメトリックを利用できます。

最大プロフィット

これで、モデルおよびプロフィットとコストの設定に基づいたプロフィットの最大量が定められます。次のように計算されます。

$$\text{プロフィット}_{\max} = \sum_{i=1}^{j} (h(x_i) \cdot r - c)$$

ここで $h(x_i)$ は、次のように定義されます。

$$h(x_i) = \begin{cases} 1 & x_i \text{ がヒット} \\ 0 & \text{の場合} \end{cases}$$

r はユーザー定義のヒット当たりの収益量であり、c はレコード当たりのユーザー定義のコストです。最高の $\hat{p}_i$（たとえば $(\hat{p}_{j+1} \cdot (r - c)) - \left((1 - \hat{p}_{j+1}) \cdot c\right) \leq 0$）を含む j に対して合計が計算されます。

最大プロフィット発生のパーセンテージ

これで、モデルの予測に基づく正のプロフィットを提供する、学習レコードのパーセンテージが与えられます。

$$\text{プロフィット}_{\%} = \frac{j}{n} \cdot 100\%$$

n は、構築中のモデルに含まれるレコードの全体数です。

リフト

これで、全体のレスポンス率に相対的な比率として、上位 q% のレコード（予測確率でソート済み）のレスポンス率を示します。

$$\text{リフト} = \frac{\sum_{i=1}^{k} \hat{p}_i / k}{\sum_{i=1}^{n} h(x_i) / n}$$

ここで、k は n（モデル構築に使用された学習レコード数）の q% です。q のデフォルト値は 30 ですが、この値は 2 値の分類ノードのオプションで変更できます。

全体の精度

これは、結果が正しく予測されたレコードのパーセンテージです。

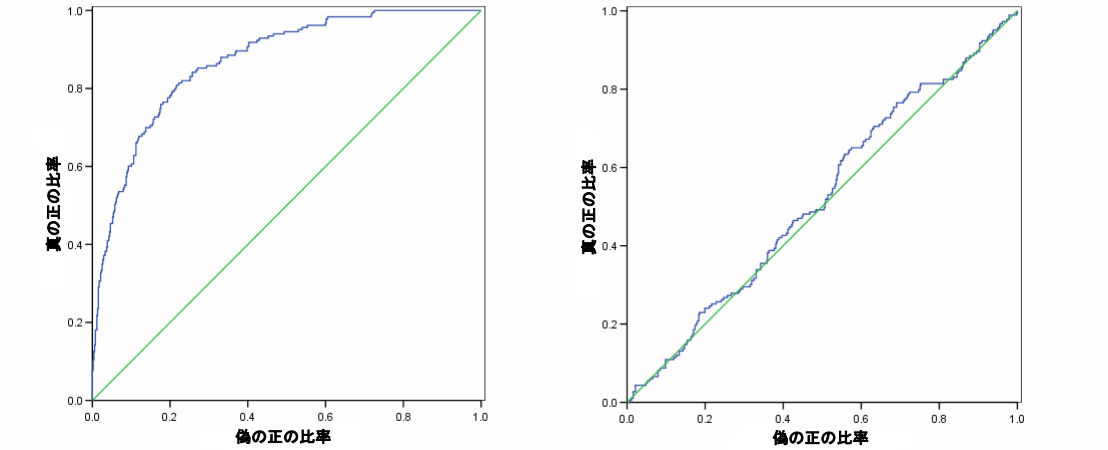
$$a = \frac{\sum_{i=1}^{n} m(i)}{n} \cdot 100\%, m(i) = \begin{cases} 1 & \text{if } (\hat{x}_i = x_i) \\ 0 & \text{otherwise} \end{cases}$$

ここでは、$\hat{x}_i$ はレコード i の予測された結果値であり、 x_i が観察値です。

曲線下の領域 (AUC)

これは、モデルの受信者動作特性（ROC）曲線下の領域を表します。ROC 曲線は、偽の正の比率（モデルが対象のレスポンスを予測しても実際のレスポンスが観察されない）に対する真の正の比率（モデルが対象のレスポンスを予測し、実際のレスポンスが観察される）をプロットします。良好なモデルの場合、曲線は左の軸近くを急上昇して頂上近くを横切るので、単位正方形内のほとんどすべての領域が曲線の下に収まります。役に立たないモデルの場合、曲線は、グラフの左下から右上隅までのほぼ対角線になります。したがって、AUC が 1.0 に近づくほど、良いモデルです。

図 6-1
良いモデル（左）と役に立たないモデル（右）の ROC 曲線



AUC は、すべてが対象の値についての同じ予測確率を持つレコード サブセットを決定する予測フィールド値の固有な組み合わせとして、セグメントを識別することで計算されます。モデルの予測フィールドで定義される s 個のセグメントが予測確率の降順でソートされ、AUC は次のようにして計算されます。

$$AUC = \sum_{i=1}^{s} |f_i - f_{i-1}| \cdot \frac{t_i + t_{i-1}}{2}$$

ここで、 f_i はセグメント i の偽陽性の累積数、つまりセグメント i の偽陽性と先行するすべての$j < i$，t_i は真性要請の累積数で、$f_0 = t_0 = 0$ の関係となります。

章

7

C5.0 アルゴリズム

C5.0 モデルの学習用コードは、RuleQuest Research Ltd Pty からライセンスを受けており、アルゴリズムはこの会社により独自に開発されています。詳細は RuleQuest Web サイト http://www.rulequest.com/ を参照してください。

注 :Modeler 13 では、C5.0 のバージョンを 2.04 から 2.06 にアップグレードしました。詳細は、RuleQuest の Web サイトを参照してください。

スコアリング

レコードは、そのレコードに該当するルールのクラスおよび確信度によってスコアリングされます。

ルール セットが C5.0 ノードから直接生成される場合、そのルールの確信度は次のように計算されます。

$$\frac{(\text{葉内の正しい数} + 1)}{(\text{葉内の合計レコード数} + 2)}$$

ルール セットが C5.0 ノードから生成されたディシジョン ツリーから生成される場合、その確信度は次のように計算されます。

$$\frac{(\text{葉内の正しい数} + 1)}{(\text{葉内の合計レコード数} + \text{対象内のカテゴリ数})}$$

ルール セット票決によるスコア

ルール セット内のルール間で票決が行われる場合、レコードに割り当てられる最終スコアは、次のように計算されます。各レコードのすべてのルールを調べ、レコードに適用される各ルールを使用して予測および関連付けられた確信度を生成します。各出力値の確信度値の合計を計算し、最も大きい確信度合計を持つ値を最終的な予測として選択します。最終的な予測の確信度は、その値の確信度合計をそのレコードに該当するルールの数で割ったものになります。

ブーストされた C5.0 分類辞を使用したスコア (ディシジョン ツリーおよびルール セット)

ブーストされた C5.0 ルール セットを使用してスコアリングする場合、ブーストされたルール セットを構成する n 件のルール セット (ブースティングのトライアルごとに 1 つのルール セット) は各スコアを使用して (上記で取得したように) 票決し、ブーストされたルール セットによってケースに割り当てられる最終スコアに到達します。

ブーストされた C5 分類辞の票決は次のとおりです。レコードごとに、複合分類辞（ルール セットまたはディシジョン ツリー）が予測値および確信度を割り当てます。各出力値の確信度値の合計を計算し、最も大きい確信度合計を持つ値を最終的な予測として選択します。ブーストされた分類辞による最終的な予測の確信度は、その値の確信度合計をすべての値の確信度の合計で割ったものになります。

章

8

Carma アルゴリズム

概要

連続アソシエーション ルール マイニング アルゴリズム (continuous association rule mining algorithm : Carma) は Apriori に代わるもので、I/O コスト、時間、およびスペースの要件を低くします(Hidber, 1999)。2 つのデータ パスだけを使用し、Apriori よりもずっと低いサポート レベルについても結果を生成できます。また、実行中にサポート レベルを変更できます。

Carma は、トランザクションを構成するアイテムおよびアイテムセットを処理できます。**アイテム**は、あるトランザクション中の特定の事柄の存在または非存在を示すフラグ型の条件です。**アイテムセット**は、トランザクション内で同時に発生する傾向がある、または発生する傾向がないアイテムのグループです。

ルールの派生

Carma では、2 つのステージに分けて処理が行われます。まず、データ中の多頻度アイテムセットを識別し、次に多頻度アイテムセットの格子からルールを生成します。

多頻度アイテムセット

Carma は、多頻度アイテムセットを識別するための 2 フェーズの手法を採用します。

フェーズ I : 推定

推定フェーズでは、単一のデータ パスを使って頻出アイテムセットの候補を識別します。**格子**は、アイテムセットに関する情報を保持するために使用されます。格子中の各ノードは、アイテムセットを構成するアイテムと、関連づけられたアイテムセットの 3 つの値を保持します。

- count : アイテムセットが格子に追加された以降に当該アイテムセットを含むトランザクションの数
- firstTrans : アイテムセットが格子に追加されたトランザクションのレコード インデックス
- maxMissed : 格子に追加される前のアイテムセットの頻度数の上限

格子はアイテムセット間の関係についての情報をコード化します。この関係情報は、アイテムセット中のアイテムによって決まります。アイテムセット X が アイテムセット Y 中のすべてのアイテムを保持する場合、Y は X の**祖先**になります。より特殊なケースとして、X が Y 中のすべてのアイテムに加えて 1 つのアイテムを保持する場合、Y は X の**親**になります。逆に、Y が X 中のすべてのアイテムを保持する場合に Y は X の**子孫**となり、Y が X 中のすべてのアイテムに加えて 1 つのアイテムを保持する場合、Y は X の**子**になります。

たとえば、X = {milk, cheese, bread} の場合、Y = {milk, cheese} は X の親であり、Z = {milk, cheese, bread, sugar} は X の子です。

格子は、最初は何のアイテムセットも保持しません。各トランザクションが読み込まれるに伴い、格子は 3 つのステップで更新されます。

- **統計量をインクリメント(増分)する。**現行トランザクション中に存在する格子内の各アイテムセットについて、count 値をインクリメント（増分）します。
- **新規アイテムセットを挿入する。**トランザクション中にあって格子中にない各アイテムセット v について、そのアイテムセットのすべてのサブセットを格子中でチェックします。アイテムセットのすべてのサブセット候補が $maxSupport \geq \sigma_i$ を持つ格子中にある場合、そのアイテムセットを格子に追加して値を設定します。
  - count は 1 に設定されます。
  - firstTrans は、現行トランザクションのレコード インデックスに設定されます。
  - maxMissed は、次のように定義されます。

$$maxMissed(v) = \min_{w \subset v} \left\{ \left(\lfloor (i-1) avg\left(\lceil \sigma \rceil_{i-1} \right) \rfloor + |v| - 1 \right), (maxMissed(w) + count(w) - 1) \right\}$$

ここで、w はアイテムセット v のサブセット、$\lceil \sigma \rceil_{i-1}$ は可変サポートについてはトランザクション i までの σ の最高値（または固定サポートについては単にσ）、|v| はアイテムセット v 中のアイテム数を表します。

- **格子を剪定する。**k 個のトランザクション（ここで k は**剪定値**で、デフォルトで 500 に設定されます）ごとに格子が検証され、小さなアイテムセットは削除されます。小さなアイテムセットは、maxSupport < σ_i であるアイテムセットとして定義されます。ここで、maxSupport = (maxMissed + count)/i です。

フェーズ Ⅱ : 検証

頻出アイテムセットの候補が識別されると、それらの候補の正確な度数を計算するための第 2 のデータ パスが作成され、この度数に基づいて頻出アイテムリストの最終リストが決定されます。

フェーズ II における最初のステップは、潜在的アイテムセットを格子から除去することです。この格子は、フェーズ I で説明したのと同じ方法で剪定されます。σ_n は、当該モデルに対してユーザが指定したサポート レベルです。

初期剪定の後で、学習データが再度処理されます。格子中の各アイテムセット v がチェックされ、インデックスが i の各トランザクション レコードについて次のように更新されます。

- firstTrans(v) < i であれば、v は正確であると見なされ、それ以上は何の更新も検討されません(格子中のすべてのノードが正確であると期された時点でフェーズ II は終了します)。
- 現行トランザクション中に v が出現すれば、次のようにして v が更新されます。
  - count(v) をインクリメント（増分）します。
  - maxMissed(v) をデクリメント（減分）します。

- firstTrans(v) = i であれば、maxMissed(v) = 0 に設定し、maxSupport(w) > maxSupport(v) である格子中の v のすべてのスーパーセット w について maxMissed を調整します。こうしたスーパーセットについて、maxMissed(w) = count(v) - count(w) に設定します。
- maxSupport(v) < σ_n であれば、格子から v を除去します。

ルールの生成

Carma は、冗長なルールを排除する傾向にあるアイテムセットの格子からルールを抽出するための一般的なルール生成アルゴリズムを採用します(Aggarwal および Yu, 1998)。ルールは、次のようにアイテムセットの格子（「多頻度アイテムセット」（ p.55 ）参照）から生成されます。

1. 格子中の各アイテムセットについて、最大祖先アイテムセットの集合を取得します。$\frac{support(Y)}{support(X)} \leq \frac{1}{c}$ である場合、アイテムセット Y はアイテムセット X の最大祖先になります。ここで c は、ルールの指定確信度閾値です。
2. X のすべての子アイテムセットの最大祖先を除去することにより、最大祖先のリストを剪定します。
3. 剪定された最大祖先リスト中の各アイテムセットについて、ルール $Y \Rightarrow X - Y$ を生成します。ここで X-Y は、アイテムセット Y 中のアイテムを除去されたアイテムセット X です。

   たとえば、X がアイテムセット {milk, cheese, bread} で、Y がアイテムセット {milk, bread} である場合、結果としてのルールは milk, bread ⇒ cheese となります。

空白の処理

Carma アルゴリズムでは、空白は無視されます。このアルゴリズムでは、入力フィールドに対して空白を含むレコードは処理されます。ただし、そのようなレコードは、1 つまたは複数のフィールドに対して空白値を持つ任意のルールと一致するとはみなされません。

オプションの効果

ルールの最小範囲／最小確信値：これらの値は、どのルールがテーブルに取り込まれるかに関する制約を定義します。範囲および確信度値が、指定された値を超えたルールだけが、ルール テーブルに取り込まれます。

最大ルール サイズ：1 つのアイテムセットとして見なされるアイテム数に関する制限を設定します。

複数の結果を持つルールを除外：このオプションは、最終ルール リスト中のルールを、単一アイテムを結果として持つルールだけに制限します。

剪定値の設定：剪定のパス間で処理すべきトランザクションの数を設定します。 詳細は、 p.55 多頻度アイテムセット を参照してください。

可変サポート：学習データ中の初期トランザクションの間に学習を促進するために、サポートを変えることを許可します。詳細は、後述の「可変サポート」を参照してください。

前提条件を持たないルールを許可：結果だけのルールを許可します。従来の if-then ルールに加えて、同時発生するアイテムの単純なステートメントであるルールを許可します。

可変サポート

可変サポートのオプションを選択すると、より効率的な学習のために、トランザクション処理の間に目標サポート値が変動します。サポート値は大きな値から始まり、トランザクションが処理されるに伴って 4 ステップで減少します。第 1 のサポート値 s_1 は最初の 9 トランザクションに、第 2 のサポート値 s_2 は次の 90 トランザクションに、第 3 のサポート値 s_3 は 100 番目のトランザクションから 4999 番目のトランザクションに、そして第 4 のサポート値 s_4 は残りのすべてのトランザクションに適用されます。最終的なサポート値を s、推定トランザクション数を t とすると、次の制約によってサポート値が決定されます。

- $s \geq 0.2$ または $t < 19$ であれば、$s_1 = s_2 = s_3 = s_4$ に設定されます。
- $19 \leq t < 190$ であれば、$s_1 = 5s_2, s_3 = s_4 = s_2$ に設定されます。これにより $\frac{(9s_1+(t-9)s_2)}{t} = s$ になります。
- $190 \leq t < 7000$ であれば、$s_1 = 5s_2, s_2 = 2s_3, s_4 = s_3$ に設定されます。これにより $\frac{(9s_1+90s_2+(t-99)s_3)}{t} = s$ となります。
- $t \geq 7000$ であれば、$s_1 = 5s_2, s_2 = 2s_3, s_3 = 5s_4$ に設定されます。これにより $\frac{(9s_1+90s_2+4900s_3+(t-4999)s_4)}{t} = s$ となります。

いずれの場合でも、方程式の解を求めることによって $s_1 > 0.5$ となる場合、s_1 は 0.5 に設定され、その他の値は関係 $\frac{\sum_{i=1}^{n} s(i)}{t} = s$ を維持するように調整されます。ここで、s(i) は i 番目のトランザクションの対象サポート（s_1、s_2、s_3、s_4 の値の 1 つ）です。

生成されたモデル／スコアリング

Carma ノードから生成されたモデルと新規データのスコアリングは、GRI 生成モデルと同様に処理されます。

章

9

C&RT アルゴリズム

C&RT の概要

C&RT& は、**分類と回帰ツリー**を意味しています。この名前は、元々(Breiman, Friedman(F), Olshen, および Stone, 1984)で使用されています。C&RT では、各サブセット中のレコードが前のサブセットよりも等質になるように、データが 2 つのサブセットに分割されます。これは**帰納的**な処理のため、分割された 2 つのサブセットは次に再び分割されます。この処理は、等質基準に達するか、または他の停止基準を満たすまで（他のツリー成長法のように）継続されます。ツリー中の異なるレベルにおいて、同じ予測フィールドが複数回使用されることもあります。代理変数の分岐を使用して、欠損値のあるデータができる限り有効活用されます。

C&RT は非常に柔軟性があります。C&RT では、ツリー成長処理において等しくない誤分類コストを考慮することができます。また、ある分類の問題中の事前確率分布を指定することもできます。C&RT ツリーにコストが複雑な自動剪定を適用して、より一般化しやすいツリーを取得することができます。

一次計算

ここでは、モデル構築時に直接行われる計算を説明していきます。

度数およびケースの重みフィールド

度数およびケースの重みフィールドは、データセットのサイズを減らすために役立ちます。それぞれには別個の機能があります。誤ってケースの重みフィールドを度数フィールドとして指定した場合、またはその逆の指定を行った場合、結果は不正なものとなってしまいます。

後述する計算では、度数またはケースの重みフィールドが指定されていない場合、すべてのレコードに対する度数およびケースの重みが 1.0 であると仮定します。

度数フィールド

度数フィールドは、各レコードが示す総観測数を表しています。レコードが 1 つ以上を表す集計データの分析に役立ちます。度数フィールドの値の合計は、常にサンプル中の総観測数と等しくなければなりません。度数フィールドを使っても、ケースごとのデータを使っても、出力と統計量は同じであることに注意してください。以下のテーブルに、予測フィールドの［性別］と［雇用］、および対象フィールド［ 回答］を持つ仮説の例を示します。たとえば、この例の度数フィールドからは、職のある 10 人の男性が対象の質問に対して「はい」と回答しており、職を持たない 19 人の女性が「いいえ」と回答していることがわかります。

テーブル 9-1
度数フィールドのあるデータ セット

性別	雇用	回答	度数
M	Y	Y	10
M	Y	N	17
M	N	Y	12
M	N	N	21
F	Y	Y	11
F	Y	N	15
F	N	Y	15
F	N	N	19

このケースで度数フィールドを使用すると、8 レコードのテーブルを処理することになります。一方ケースごとのデータを使用すると、120 レコードを処理しなければなりません。

ケースの重み

ケースの重みフィールドを使用すると、データセット中のレコードに対して、異なる取り扱いを行うことができます。**ケースの重みフィールド**を使用する場合、分析におけるレコードの寄与率が、サンプル中のレコードが表す母集団に比例して重み付けられます。たとえば、ダイレクトメールによる販促活動において、10,000 世帯から回答があり、1,000,000 世帯からは回答がなかった場合を考えてみましょう。ここで、データ ファイルのサイズを減らすために、すべての回答を利用するけれども、非回答者については 1 % のサンプル（10,000）だけを利用します。この場合、回答者に対してはケースの重みに 1 を、非回答者に対しては 100 を定義します。

モデル パラメータ

C&RT は、分岐により作成されるそれぞれの子ノードが、親ノードよりもより純粋になるような分岐点を選択することにより機能します。ここで**純粋（純度）**は、対象フィールドの値の類似性を表しています。完全に純粋なノードでは、すべてのレコードの対象フィールドに同じ値が存在しています。C&RT では、**不純度測定法**を定義することにより、ノード分岐点の不純度を測定します。 詳細は、 p. 63 不純度を参照してください。

C&RT ツリー（すべてのレコードを含むルート ノードから開始）を構築するために使用する手順を次に示します。

各予測フィールドの最良の分岐点を探す：各予測フィールドに対して、次のようにそのフィールドに対する最良の分岐点を探します。

- **範囲型（数値）フィールド**：ノード中のレコードのフィールド値を、最小の値から最大の値へとソートします。分岐点となるポイントを選択し、その分岐により子ノードとなる不純度統計量を算出します。フィールドに対する最良の分岐点

を選択します。最良の分岐点は、分岐対象ノードの不純度と相対的に、もっとも不純度が減少する分岐点です。

- **シンボル値（カテゴリ）フィールド**：有効な各値の組み合わせを、2 つのサブセットとして調査します。各組み合わせに対して、その組み合わせに基づいて分岐に対する子ノードの不純度を計算します。フィールドに対する最良の分岐点を選択します。最良の分岐点は、分岐対象ノードの不純度と相対的に、もっとも不純度が減少する分岐点です。

ノードの最良の分岐点を探す：最良の分岐によりノードの不純度がもっとも減少するフィールドを判断し、そのフィールドの最良の分岐点を、ノードに対するすべての分岐の中で最良の分岐として選択します。

停止基準を確認して、処理を繰り返します：分岐や親ノードにより停止基準が満たされていない場合は、分岐を行って 2 つの子ノードを作成します(詳細は、 p.65 停止基準 を参照してください。)。各子ノードに対して、アルゴリズムをもう一度適用します。

空白の処理

ツリー モデルの構築時に、対象フィールドに欠損値があるレコードは無視されます。

代理変数の分岐は、予測フィールドの空白値を処理するために用いられます。ある特定のノードに対して、分岐に使用する最良の予測フィールドに空白値または欠損値がある場合、そのノードに関連する予測フィールドと類似した分岐を生成する他のフィールドが、予測フィールドの代理変数として使用され、その値はいずれかの子ノードにレコードを割り当てるために用いられます。

たとえば、ノード t における最良の分岐点 s* を定義する予測フィールド X* を考えてみましょう。代理変数の分岐処理では、分岐がノード t における s* に最も類似するように、他の予測フィールド X をベースに他の分岐点 s を探します。新規レコードが予測対象で、ノード t における X* 上に欠損値がある場合、代わりに代理変数の分岐 s が適用されます(このレコードが、X 上にも欠損値を持つ場合を除きます。このような状況下では、次に最良の代理変数が、指定された代理変数の数の上限まで順次使用されます。)

処理速度およびメモリー保持の観点から、ツリー中の各分岐に対して、一定数の代理変数しか使用されません。レコードの分岐フィールドおよびすべての代理変数フィールドに欠損値がある場合、それは重みが大きい確率を持つ子ノードに割り当てられ、次のように計算されます。

$$\frac{N_{f,j}(t)}{N_f(t)}$$

ここで $N_{f,j}(t)$ はノード t のカテゴリ j 中にあるレコードの度数の重みの合計を、$N_f(t)$ は、ノード t 中のすべてのレコードの度数の重みの合計を表します。

均等化またはユーザーが指定した事前確率を使って構築されたモデルの場合は、計算に事前確率が取り入れられます。

$$\frac{\pi(j)}{p_f(t)} \times \frac{N_{f,j}(t)}{N_f(t)}$$

ここで π(j) はカテゴリ j の事前確率を、$p_f(t)$ はノードに割り当てられたレコードの重み付けられた確率を表します。

$$p_f(t) = \sum_j \frac{\pi(j) N_{f,j}(t)}{N_{f,j}}$$

ここで $N_{f,j}(t)$ はカテゴリ j に所属するノード t 中の度数の重みの合計（度数の重みが定義されていない場合はレコード数）を、$N_{f,j}$ は学習サンプル全体のカテゴリに所属するレコードの度数の重みの合計を表します。

予測関連度

$\hbar_{X^* \cap X}$（または $\hbar_{X^* \cap X}(t)$）を、X* および X の両方が欠損値でない学習ケース（ノード t の学習ケース）であるとします。$p(s^* \approx s_X|t)$ を、$\hbar_{X^* \cap X}(t)$ のケースを s^* および s_X によって同一の子に送る確率とし、$\tilde{s}_X$ を最大確率 $p(s^* \approx \tilde{s}_X|t) = \max_{s_X}(p(s^* \approx s_X|t))$ による分割とします。

ノード t の s* および $\tilde{s}_X$ 間の $\lambda(s^* \approx \tilde{s}_X|t)$ 予測関連度は、次のようになります。

$$\lambda(s^* \approx \tilde{s}_X|t) = \frac{\min(p_L, p_R) - (1 - p(s^* \approx \tilde{s}_X|t))}{\min(p_L, p_R)}$$

ここで、p_L（または p_R）は相対確率で、ノード t の最適な確率 s* at は、X* の値が欠損値でないケースを左側（または右側）の子ノードに送ります。そして、ここで

$$p(s^* \approx s_X|t) = \begin{cases} \sum_j \frac{\pi(j) N_{w,j}(s^* \approx s_X, t)}{N_{w,j}(X^* \cap X)} & \text{if} Y \text{is categorical} \\ \frac{N_w(s^* \approx s_X, t)}{N_w(X^* \cap X)} & \text{if} Y \text{is continuous} \end{cases}$$

を

$$N_w(X^* \cap X) = \sum_{n \in \hbar_{X^* \cap X}} w_n f_n, \quad N_w(X^* \cap X, t) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n$$

$$N_w(s^* \approx s_X, t) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n I(n : s^* \approx s_X)$$

$$N_{w,j}(X^* \cap X) = \sum_{n \in \hbar_{X^* \cap X}} w_n f_n I(y_n = j), \quad N_{w,j}(X^* \cap X) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n I(y_n = j)$$

$$N_{w,j}(s^* \approx s_X, t) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n I(y_n = j) I(n : s^* \approx s_X)$$

そして $I(n : s^* \approx s_X)$ は分割 s* および s_X がケース n を同一の子に送る場合に値 1 をとり、そうでない場合は 0 をとる指標関数となります。

オプションの効果

不純度

C&RT モデルの分岐点を探すために、対象フィールドの種類に応じて 3 種類の不純度が使用されます。シンボル値の対象フィールドの場合、Gini または Twoing を使用することができます。連続型の対象フィールドの場合、最小 2 乗偏差 (LSD) 法が自動的に選択されます。

Gini

C&RT ツリーのノード t における Gini インデックス g(t) は、次のように定義されます。

$$g(t) = \sum_{j \neq i} p(j|t)p(i|t)$$

ここで、 i および j は対象フィールドのカテゴリを表します。また、

$$p(j|t) = \frac{p(j,t)}{p(t)}$$

$$p(j,t) = \frac{\pi(j)N_j(t)}{N_j}$$

$$p(t) = \sum_j p(j,t)$$

ここで $\pi(j)$ はカテゴリ j の事前確率値、$N_j(t)$ はノード t のカテゴリ j 中のレコード数、そして N_j はルート ノードのカテゴリ j 中のレコード数を表します。ツリー成長時に Gini インデックスを使って分岐の改善を探す場合、分岐予測フィールドに対する有効な値を持つノード t およびルート ノード中のレコードだけが、それぞれ $N_j(t)$ および N_j の計算に用いられます。

Gini インデックスの式は、次のように記述することもできます。

$$g(t) = 1 - \sum_j p^2(j|t)$$

このように、ノード中のレコードがカテゴリ間に渡って均等に分散している場合、Gini インデックスは $1 - 1/k$ の最大値をとります。ここで、k は対象フィールドのカテゴリ数を表します。ノード中のすべてのレコードが同じカテゴリに所属している場合、Gini インデックスは 0 になります。

ノード t における分岐 s の Gini 基準関数 Φ(s, t) は、次のように定義されます。

$$\Phi(s,t) = g(t) - p_L g(t_L) - p_R g(t_R)$$

ここで p_L は左側の子ノードに送られた t 中のレコードの比率を、p_R は右側の子ノードに送られたレコードの比率を表します。比率 p_L および p_R は、次のように定義されます。

$$p_L = \frac{p(t_L)}{p(t)}$$

および

$$p_R = \frac{p(t_R)}{p(t)}$$

分岐 s は、Φ(s, t) の値が最大になるように選択されます。

Twoing

Twoing インデックスでは、対象カテゴリを 2 つのスーパークラスに分岐し、それらのスーパークラスに基づいて予測フィールドの最良の分岐点を探します。スーパークラス C_1 および C_2 は、次のように定義されます。

$$C_1 = \{j : p(j|t_L) \geq p(j|t_R)\}$$

および

$$C_2 = C - C_1$$

ここで C は対象フィールドのカテゴリ セットを、$p(j|t_R)$ および $p(j|t_L)$ はそれぞれ右側および左側の子ノードに対して Gini 式で定義されている p(j|t) を表します。 詳細は、 p.63 Gini を参照してください。

ノード t における分岐 s の Twoing 基準関数は、次のように定義されます。

$$\Phi(s,t) = p_L p_R \left[\sum_j |p(j|t_L) - p(j|t_R)| \right]^2$$

ここで t_L および t_R は、分岐 s により作成されるノードを表します。分岐点は、この基準関数を最大化する分岐点が選択されます。

最小 2 乗偏差法

連続する対象フィールドの場合、不純度として**最小 2 乗偏差**（LSD）法が使用されます。LSD 測度 R(t) は、単純に重みが付けられたノード t に対するノード内分散で、ノードに対するリスク推定の結果と等しくなります。これは、次のように定義されます。

$$R(t) = \frac{1}{N_W(t)} \sum_{i \in t} w_i f_i (y_i - \overline{y}(t))^2$$

ここで $N_W(t)$ はノード t 中の重み付けられたレコード数、w_i はレコード i に対する重み付けフィールドの値（ある場合）、f_i は度数フィールドの値（ある場合）、y_i は対象フィールドの値、そして y(t) は（重み付けられた）ノード t の平均を表します。ノード t における分岐 S の LSD 基準関数は、次のように定義されます。

$$\Phi(s,t) = R(t) - p_L R(t_L) - p_R R(t_R)$$

分岐 s は、Φ(s, t) の値が最大になるように選択されます。

停止基準

停止基準は、ツリー中のノードの分岐をいつ停止するかを判断するために用いられます。ツリーの成長は、ツリー中の各枝葉ノードが最低 1 つの停止基準を満たすまで継続します。ノードの分岐を停止する条件を次に示します。

- ノードが純粋の場合（すべてのレコードの対象フィールドが同じ値を持つ）
- モデルが使用するすべての予測フィールドに対して、ノード中のすべてのレコードが同じ値を持つ場合
- 現在のノードのツリーの深さ（現在のノードを定義する帰納的ノード分岐数）が最大ツリー深さ（デフォルト値またはユーザー指定）の場合
- ノード中のレコード数が最小親ノード サイズ（デフォルト値またはユーザー指定）未満の場合
- ノードの最良の分岐点の結果となる任意の子ノード中のレコード数が、最小子ノード サイズ（デフォルト値またはユーザー指定）未満の場合
- ノードの最良の分岐点により不純度の最小変化（デフォルト値またはユーザー指定）より小さい不純度の減少が生じる場合

プロフィット

プロフィットは、（シンボル値）対象フィールドのカテゴリに関連する数値で、セグメントに関連するゲインまたはロスを推定するために用いられます。プロフィットは、対象フィールドの各値の相対値を定義しています。値は、ゲインを算出するために使われますが、ツリーの成長には使用されません。

ツリー中の各ノードのプロフィットは、次のように計算されます。

$$\sum_j f_j(t) P_j$$

ここで j は対象フィールドのカテゴリ、$f_j(t)$ は対象フィールドに対するカテゴリ j と、ノード t 中のすべてのレコードの度数フィールド値の合計、そして P_j はカテゴリ j に対するユーザーが定義したプロフィットの値を表します。

事前確率

事前確率は、対象フィールドのカテゴリの誤分類率に影響する数値です。事前確率は、分析前に対象フィールドの各カテゴリに所属すると推定されるレコードの比率を示します。この値はツリーの成長とリスク推定の両方に使用されます。

事前確率を取得するには、3 種類の方法があります。

経験的事前確率

デフォルトでは、事前確率は学習データに基づいて算出されます。各対象カテゴリに割り当てられる事前確率は、そのカテゴリに所属する学習データ中の重み付けられたレコードの比率になります。

$$\pi(j) = \frac{N_{w,j}}{N_w}$$

ツリー成長およびクラス割り当てにおいては、N はケースの重みと度数の重みの両方を考慮します（定義されている場合）。リスク推定においては、度数の重みだけが経験的事前確率の計算に含まれます。

等事前確率

等事前確率（すべてのクラスで同じ）では、それぞれの J カテゴリの事前確率に同じ値が選択されます。

$$\pi(j) = \frac{1}{J}$$

ユーザー定義事前確率

ユーザー定義（設定）の事前確率を使用する場合、事前確率を含む計算に指定された値が使用されます。事前確率に指定された値は、確率の制約に従っている必要があります。全カテゴリの事前確率の合計は、 1.0 である必要があります。ユーザーが定義した事前確率がこの条件を満たしていない場合、次の式を使って事前確率が調整されます。この事前確率は、元の事前確率の比率を維持しながら、制約に準拠するように調整されます。

$$\pi'(j) = \frac{\pi(j)}{\sum_J \pi(j)}$$

ここで π'(j) はカテゴリ j の調整された事前確率を、π(j) はユーザーが定義したカテゴリ j に対する元の事前確率を表します。

コスト

Gini: コストを指定した場合、Gini インデックスは次のように計算されます。

$$g(t) = \sum_{j \neq i} C(i|j)p(j|t)p(i|t)$$

ここで C(i|j) はカテゴリ j をカテゴリ i として誤分類するコストを表します。

Twoing: コストを指定した場合、Twoing を使ったノードの分岐時にそのコストは考慮されません。ただし、ノードの割り当てとリスクの推定時にはコストが考慮されます。後述する「予測値」および「リスク推定」を参照してください。

LSD: 回帰ツリーには、コストは適用されません。

剪定

剪定は、完全に成長したツリーの調査、およびツリーの精度にさほど寄与しない下位レベルの分岐の削除に関する処理です。ツリーの剪定では、可能な最大ツリーの誤分類リスクと比べてさほど誤分類リスクが大きくならない、最小のツリーが作成されます。このために、より複雑なツリーを持つためのコストが、他のレベルのノード（枝葉）を持つためのゲインを超える場合、ツリーの枝葉は削除されます。

誤分類リスクとツリーの複雑さの両方を最低限に抑えるために、これらを測定するインデックスが使用されます。このコストと複雑さの測度は、次のように定義されます。

$$R_\alpha(T) = R(T) + \alpha\left|\tilde{T}\right|$$

R(T) はツリー T の誤分類リスクを、$\left|\tilde{T}\right|$ はツリー T のターミナル ノード数を表します。項目 α は、「ターミナル ノードごと」のツリーの複雑性コストを表します。（α の値は、剪定中のアルゴリズムによって算出されることに注意してください。）

生成するツリーには最大サイズ（T_{max}）があり、各ターミナル ノードには 1 つのレコードしか含まれません。複雑さのコストがない場合（α = 0）、すべてのレコードが完全に予測されるため、最大のツリーのリスクが最も低くなります。そのため、α の値が大きくなるほど、T(α) 中のターミナル ノード数は少なくなります。ここで T(α) は、与えられた α に対して、複雑さのコストが最も低いツリーを表します。α が 0 から増加するにつれて、サブツリーの有限数列（T_1, T_2, T_3）が生成されます。それぞれ、順次ターミナル ノード数が少なくなっていきます。コストと複雑さによる剪定は、最も弱い分岐を削除することにより行われます。

任意の単一ノード {t} および {t} の子枝葉 T_t のコストの複雑さを表す式を次に示します。

$$R_\alpha(\{t\}) = R(t) + \alpha$$

$$R_\alpha(T_t) = R(T_t) + \alpha\left|\tilde{T}_t\right|$$

$R_\alpha(T_t)$ が $R_\alpha(\{t\})$ 未満の場合、枝葉 T_t は単一ノード {t} よりもコストの複雑さが小さくなります。

ツリー成長の過程では、($\alpha = 0$) に対する $R_\alpha(\{t\}) \geq R_\alpha(T_t)$ が保証されます。α が 0 から増加するにつれて、$R_\alpha(\{t\})$ と $R_\alpha(T_t)$ の両方が線型的に成長していきます。この場合、後者の成長率が速くなります。最終的には、すべての $\alpha > \alpha'$ に対して $R_\alpha(\{t\}) < R_\alpha(T_t)$ となるような閾値 α' に達します。このことは、α が α' よりも大きく成長した場合、{t} 下の子枝葉 T_t を削除すると、ツリーのコストの複雑さを減らせることを意味しています。閾値は単純な計算により決められます。この最初の不等式 $R_\alpha(\{t\}) \geq R_\alpha(T_t)$ を解いて、不等式が保持する α の最大値を探すことができます。これは、g(t) として表すこともできます。最終的には次のようになります。

$$\alpha \leq g(t) = \frac{R(t) - R(T_t)}{\left|\tilde{T}_t\right| - 1}$$

g(t) の最も小さい値を持つノードとして、ツリー T 中の最も弱いリンク (t) を定義することができます。

$$g(\bar{t}) = \min_{t \in T} g(t)$$

このため、α が増加するにつれて、$\bar{t}$ が $R_\alpha(\{t\}) = R_\alpha(T_t)$ である最初のノードになります。この時点で $\{\bar{t}\}$ が $T_{\bar{t}}$ に対してより好ましくなり、子枝葉が剪定されます。

このようなバックグラウンド情報を判断した後、剪定アルゴリズムにより次の処理が行われます。

- $\alpha_1 = 0$ を設定し、完全に成長したツリー $T_1 = T(0)$ から開始します。
- 枝葉が剪定されるまで、α の値を増やします。ツリーから枝葉を剪定し、剪定されたツリーのリスク推定を計算します。
- ルート ノードだけが残るまで、直前のステップを繰り返し、一連のツリー T_1、T_2、... を残します。T_k とします。
- 標準エラー ルール オプションが選択されている場合、最小のツリー T_{opt} を選択します。

$$R(T_{opt}) \leq \min_k R(T_k) + m \times SE(R(T))$$

- 標準エラー ルール オプションが選択されていない場合は、リスク推定が最も小さいツリー R(T) を選択します。

二次計算

二次計算はモデルの構築に直接には関係していませんが、モデルとそのパフォーマンスに関する情報を得ることができます。

リスク推定

リスク推定は、ツリーの特定ノードおよびツリー全体の予測フィールド中の誤差リスクを表しています。

シンボル値フィールドに対するリスク推定

分類ツリー（シンボル値対象フィールドを持つ）の場合、ノード t のリスク推定値 r(t) は次のように算出されます。

$$r(t) = \frac{1}{N_f} \sum_j N_{f,j}(t) C(j^*(t)|j)$$

ここで C(j*(t)|j) は対象値 j を j*(t) として持つレコードを分類する際の誤分類コストを、$N_{f,j}(t)$ はカテゴリ j の ノード t 中のレコードの度数の重みの合計（または度数の重みが定義されていない場合はレコードの数）、N_f は学習データ中のすべてのレコードの度数の重みの合計を表します。

ユーザーが指定した事前確率を使って構築されたモデルの場合は、リスク推定値は次のように計算されます。

$$\sum_j \frac{\pi(j) N_{f,j}(t)}{N_{f,j}} C(j^*(t)|j)$$

ケースの重みはリスク推定値の計算では考慮されません。

数値型対象フィールドのリスク推定

回帰ツリー（数値型対象フィールドを持つ）の場合、ノード t のリスク推定値 r(t) は次のように算出されます。

$$r(t) = \frac{1}{N_f(t)} \sum_{i \in t} f_i (y_i - \overline{y}(t))^2$$

ここで f_i はレコード i（ノード t に割り当てられたレコード）の度数の重み、y_i はレコード i の対象フィールドの値、そして $\overline{y}(t)$ はノード t 中のすべてのレコードに対する、重み付けられた対象フィールドの平均を表します。

ツリーのリスク推定

分類ツリーと回帰ツリーの両方で、ツリー（T）のリスク推定値 R(T) は、ターミナル ノード r(t) のリスク推定値の合計を取得することにより算出されます。

$$R(T) = \sum_{t \in T'} r(t)$$

ここで T’ はツリー中のターミナル ノードのセットを表します。

ゲインの要約

ゲインの要約は、ツリーのターミナル ノードの記述統計量を提供しています。
対象フィールドが連続型（スケール）の場合、ゲインの要約は各ターミナル ノードの対象値の重み付けられた平均を表します。

$$g(t) = \sum_{i \in t} w_i f_i x_i$$

対象フィールドがシンボル値（カテゴリ）の場合は、選択した対象カテゴリ中のレコードの重み付けられた割合を表します。

$$g(t, j) = \frac{\sum_{i \in t} f_i x_i(j)}{\sum_{i \in t} f_i}$$

ここでレコード x_i が対象カテゴリ j 中にある場合 $x_i(j) = 1$ に、それ以外の場合は 0 になります。ツリーのプロフィットが定義されている場合、各ターミナル ノードのプロフィット値の平均がゲインになります。

$$g(t) = \sum_{i \in t} f_i P(x_i)$$

ここで $P(x_i)$ はレコード x_i 中に観測される対象値に割り当てられたプロフィット値を表します。

生成されたモデル／スコアリング

C&RT 生成モデルにより行われる計算は、後述します。

予測値

新しいレコードは、ツリーのターミナル ノードへのツリー分割にしたがってスコアリングされます。各ターミナル ノードには、それに対応する予測フィールドがあります。予測フィールドは、次のように決定されます。

分類ツリー

シンボル値対象フィールドを持つツリーの場合、各ターミナル ノードの予測カテゴリは、ノードの重み付けられたコストが最も低いカテゴリになります。この重み付けられたコストは、次のように算出されます。

$$\min_i \sum_j C(i|j)p(j|t)$$

ここで C(i|j) はレコードが実際にはカテゴリ j にある場合に、レコードをカテゴリ i に分類する際の、ユーザーが定義した誤分類コストを、p(j|t) はカテゴリ j がノード t にある場合の、カテゴリ中のレコードの重み付けられた条件確率を表し、次のように定義されます。

$$p(j|t) = \frac{p(j,t)}{\sum_j p(j,t)}, p(j,t) = \pi(j)\frac{N_{w,j}(t)}{N_{w,j}}$$

ここで π(j) はカテゴリ j, N_w の事前確率、$_j$(t) はカテゴリ j を持つノード t 中の重み付けられたレコード数（度数またはケースの重みが指定されていない場合はレコード数）を表します。

$$N_{w,j}(t) = \sum_{i \in t} w_i f_i j(i)$$

$N_{w,j}$ は、カテゴリ j（ノードは任意）中の重み付けられたレコード数を表します。

$$N_{w,j} = \sum_{i \in T} w_i f_i j(i)$$

回帰ツリー

数値型対象フィールドを持つツリーの場合、各ターミナル ノードの予測カテゴリは、ノード中のレコードに対する対象値の重み付けられた平均になります。この重み付けられた平均は、次のように算出されます。

$$\overline{y}(t) = \frac{1}{N_w(t)} \sum_{i \in t} w_i f_i y_i$$

ここで N_w(t) は、次のように定義されます。

$$N_w(t) = \sum_{i \in t} w_i f_i$$

確信度

分類ツリーに関しては、生成モデルを通過するレコードに対する確信度値は、次のように計算されます。回帰ツリーには、確信度値は割り当てられません。

分類ツリー

スコア付きレコードの確信度とは、予測カテゴリに所属するターミナル ノードに割り当てられたスコア付きレコードに対する、学習データ内にある重み付けされたレコードの比率です。ラプラス補正による次のような変更も加えられます。

$$\frac{N_{f,j}(t)+1}{N_f(t)+k}$$

空白の処理

新規レコードの分類において、空白はツリーの成長時のように処理されます。可能な場合には代理変数が使用され、必要に応じて重み付けられた確率に基づいて分岐が行われます。 詳細は、 p. 61 空白の処理 を参照してください。

章
10

CHAID アルゴリズム

CHAID の概要

CHAID は、Chi-squared Automatic Interaction Detector（カイ 2 乗による相互作用の自動検出）の略称で、(Kass, 1980) が開発した非常に効率的な分岐またはツリー成長のための統計技法です。CHAID では、検証の有意度を基準として使用して、すべての潜在的な予測フィールドの値を評価し、対象の変数に関して統計的に等質（類似）と判定された値を結合して、同質ではない（非類似）と判断された他の値すべてを保持します。

次に、最良の予測フィールドを選択し、選択したフィールドの同質値で各子ノードが構成されるように決定ツリーの最初の枝葉を形成します。ツリーが完全に成長するまでこの処理が繰り返されます。使用する統計検定は、対象フィールドの測定レベルによって決まります。対象フィールドが連続する場合には F 検定が使用され、対象フィールドがカテゴリの場合にはカイ2 乗検定が使用されます。

CHAID は 2 進木法ではありません。つまり、ツリーの特定のレベルで 3 つ以上のカテゴリを生成できます。したがって、2 進成長法を使用した場合よりも幅の広いツリーが形成される傾向があります。CHAID ではあらゆる種類の変数を使用でき、ケースの重み付け変数と度数変数の両方も使用できます。また、欠損値は単一の有効カテゴリとして扱われます。

拡張 CHAID

Exhaustive CHAID は、CHAID 法のいくつかの欠点を克服するために開発された修正版です (Biggs, de Ville, および Suen, 1991)。特に、CHAID では残りのカテゴリが統計的に異質と判断されるとカテゴリの結合が停止されるため、変数の最適な分岐を発見できない場合があります。Exhaustive CHAID では、最後の 2 つのスーパーカテゴリが残るまで予測変数のカテゴリの結合を続行することで、この欠点を克服しています。次に、一連の結合から予測を調べ、対象変数ともっとも密接なアソシエーションを持つカテゴリのセットを特定して、そのアソシエーションの調整された p 値を計算します。このように、Exhaustive CHAID では予測変数ごとに最適な分岐を見つけて、調整された p 値を比較することで、分岐点となる予測変数を選択できます。

Exhaustive CHAID と CHAID では、使用する統計検定と欠損値の扱い方は同じです。Exhaustive CHAID では CHAID の場合より徹底的に変数のカテゴリを結合していくため、計算に時間がかかります。ただし、時間に余裕があれば、一般的には CHAID よりも Exhaustive CHAID の方が安全です。通常は CHAID よりも有用な分岐が見つかることが多いとは言え、データによっては Exhaustive CHAID でも CHAID でも結果に変わりがない場合もあります。

一次計算

ここでは、モデル構築時に直接行われる計算を説明していきます。

度数およびケースの重みフィールド

度数およびケースの重みフィールドは、データセットのサイズを減らすために役立ちます。それぞれには別個の機能があります。誤ってケースの重みフィールドを度数フィールドとして指定した場合、またはその逆の指定を行った場合、結果は不正なものとなってしまいます。

後述する計算では、度数またはケースの重みフィールドが指定されていない場合、すべてのレコードに対する度数およびケースの重みが 1.0 であると仮定します。

度数フィールド

度数フィールドは、各レコードが示す総観測数を表しています。レコードが 1 つ以上を表す集計データの分析に役立ちます。度数フィールドの値の合計は、常にサンプル中の総観測数と等しくなければなりません。度数フィールドを使っても、ケースごとのデータを使っても、出力と統計量は同じであることに注意してください。以下のテーブルに、予測フィールドの［性別］と［雇用］、および対象フィールド［ 回答］を持つ仮説の例を示します。たとえば、この例の度数フィールドからは、職のある 10 人の男性が対象の質問に対して「はい」と回答しており、職を持たない 19 人の女性が「いいえ」と回答していることがわかります。

テーブル 10-1
度数フィールドのあるデータ セット

性別	雇用	回答	度数
M	Y	Y	10
M	Y	N	17
M	N	Y	12
M	N	N	21
F	Y	Y	11
F	Y	N	15
F	N	Y	15
F	N	N	19

このケースで度数フィールドを使用すると、8 レコードのテーブルを処理することになります。一方ケースごとのデータを使用すると、120 レコードを処理しなければなりません。

ケースの重み

ケースの重みフィールドを使用すると、データセット中のレコードに対して、異なる取り扱いを行うことができます。**ケースの重みフィールド**を使用する場合、分析におけるレコードの寄与率が、サンプル中のレコードが表す母集団に比例して重み付けられます。たとえば、ダイレクトメールによる販促活動において、10,000 世帯から回答があり、1,000,000 世帯からは回答がなかった場合を考えてみましょう。ここで、データ ファイルのサイズを減らすために、すべての回答を利用するけれども、非回答者については 1 % のサンプル（10,000）だけを利用します。この場合、回答者に対してはケースの重みに 1 を、非回答者に対しては 100 を定義します。

尺度レベルの予測フィールドのビン化

スケール レベル（連続）「予測」フィールドは、自動的に順序カテゴリのセットに離散化（**ビン化**）されます。このプロセスがモデル内のスケール レベルの予測フィールドごとに実行され、その後で CHAID（または Exhaustive CHAID）アルゴリズムが適用されます。ビン化されたカテゴリは次のように決定されます。

1. データ値 y_i がソートされます。
2. 一意の値ごとに、もっとも小さな値から始めて、現在の値 y_i 以下の値の相対（重み付けられた）度数を計算します。

$$cf_i = \sum_{y_k < y_i} w_k$$

ここで、w_k は重み付けられたレコード k（重みが定義されていない場合は 1.0）です。

3. 相対的な度数と最適なビン百分率のカットポイント（0.10、0.20、0.30 など）を比較して値が属するビンを決定します。

$$binindex = \frac{g}{W+1} \times 10$$

ここで、W は学習データ $\sum_i w_i$ のすべてのレコードの重み付けられた度数の合計です。この場合、次のようにして計算されます。

$$g = \begin{cases} cf_{i-1} + \frac{w_i+1}{2}, & w_i \geq 1 \\ cf_{i-1} + \frac{w_i}{2}, & w_i < 1 \end{cases}$$

- この値のビン インデックスが前のデータ値のビン インデックスと異なる場合には、ビン リストに新しいビンを追加して、そのカットポイントを現在のデータ値に設定します。
- ビン インデックスが前のデータ値のビン インデックスと等しい場合には、そのビンのカットポイントを現在のデータ値に更新します。

通常、CHAID ではデフォルトで k = 10 個のビンの作成が試みられますが。ただし、単一の値を持つレコードの数が多い場合（または同じ値を持つレコードのセットに大きな重み付けられた度数が結合されている場合）、作成されるビンの数はもっと少なくなることがあります。これは、ビン中の同じ値を持つレコードの重み付けられた度数が重み付けられた期待度数（重み付けられた度数の合計の 1/k）より大きい場合に起こります。また、学習データ中のレコードのビン化されたフィールドの異なる値が k より少ない場合にも起こります。

モデル パラメータ

CHAID はあらゆる種類の連続フィールドおよびカテゴリ フィールドに適用できます。ただし、連続する予測フィールドは、分析用に自動的にカテゴリに分類されます。詳細は、 p.75 尺度レベルの予測フィールドのビン化 を参照してください。

CHAID のエキスパート オプションを使用して、以下のオプションのいくつかを設定できます。これらのオプションには、Pearson カイ二乗検定、尤度比検定、結合のレベル分割のレベル、スコア値、停止規則の詳細があります。

CHAID アルゴリズムは次のように実行されます。

予測フィールドのカテゴリ結合 (CHAID)

すべての予測フィールドを結合し、対象フィールドに関して統計的に異質でないカテゴリをまとめて、それぞれの分岐点を探します。最終的な予測フィールドの各カテゴリ X は、X がノードの分岐に使用された場合、子ノードを表します。各予測フィールド X に対して、下記の手順が適用されます。

1. X に 1 つまたは 2 つのカテゴリがある場合、これ以上カテゴリの結合は行われません。この場合には、以下のノードの分岐に進んでください。
2. 対象フィールドに関連した該当する統計検定の p 値によって決まる、もっとも異質でない（もっとも類似する）X の適したカテゴリのペアを見つけます。 詳細は、p.77 使用される統計検定 を参照してください。

   順序型フィールドの場合、結合できるのは隣接するカテゴリのみです。名義フィールドの場合は、すべてのペアを結合できます。
3. もっとも大きな p 値を持つペアの場合、p 値が α_{merge} より大きければ、そのカテゴリのペアを 1 つのカテゴリに結合します。それ以外の場合は、ステップ 6 に進みます。
4. ユーザーが［**結合されたカテゴリの分岐を許可**］オプションを選択し、新しく形成された結合カテゴリに 3 つ以上のカテゴリが含まれる場合には、結合カテゴリ内の最良の 2 進木分岐点（統計検定の p 値がもっとも小さいカテゴリ）を見つけます。その p 値が $\alpha_{split-merge}$ に等しいか、またはこれより小さい場合には、分岐を行って結合されたカテゴリから 2 つのカテゴリを作成します。
5. この期待値フィールドについて、ステップ 1 から手順を繰り返してカテゴリを結合します。
6. ユーザーが指定した最小セグメント サイズのレコード数に満たないカテゴリはすべて、もっとも類似する他のカテゴリ（つまり、もっとも小さなカテゴリと比較した場合にもっとも大きなp 値 が得られるカテゴリ）と結合されます。

予測フィールドのカテゴリの結合 (Exhaustive CHAID)

Exhaustive CHAID と CHAID の違いは、予測フィールドごとに最適なカテゴリのセットを見つけるためにカテゴリの結合がより徹底的に検証される点です。通常の CHAID の場合と同様に、最終的な予測フィールドの各カテゴリ X は、X がノードの分岐に使用された場合、子ノードを表します。各予測フィールド X に対して、下記の手順が適用されます。

1. 予測変数 X ごとに、対象変数 Y に関してもっとも異質でない（つまり、もっとも p 値が大きい）X のカテゴリのペアを見つけます。ここで使用する p 値の計算方法は、Y の測定レベルによって決まります。 詳細は、 p.77 使用される統計検定 を参照してください。

2. p 値がもっとも大きいペアを結合カテゴリに結合します。
3. 次に、X の新しいカテゴリのセットに基づいて p 値を計算します。これは、X のカテゴリの 1 セットを表します。p 値と対応するカテゴリのセットを覚えておきます。
4. 最後に 2 つのカテゴリが残るまで、ステップ 1、2、3 を繰り返します。次に、一連の結合処理の各ステップで生成された X のカテゴリのセットを比較して、ステップ 3 の p 値がもっとも小さいカテゴリのセットを見つけます。このセットが、現在のノードの分岐点を決定するために使用する X の結合されたカテゴリのセットになります。

ノードの分岐

すべての予測フィールドについてカテゴリを結合したら、以下で説明するように、アソシエーションの統計テストの調整済みの p 値に基づいて各フィールドの対象フィールドとのアソシエーションが評価されます。

もっとも密接なアソシエーションを持つ（調整済みの p 値がもっとも小さい）予測フィールドと、分岐の閾値 α_{split} が比較され、その p 値が α_{split} に等しいか、またはこれより小さい場合には、そのフィールドが現在のノードの分岐フィールドとして選択されます。分岐フィールドの結合されたカテゴリはそれぞれ、分岐の子ノードを定義します。

現在のノードに分岐を適用した後、順に結合/分岐プロセスを子ノードに適用して、分岐が可能かどうかが確認されます。分岐されていないすべてのノードについて 1 つまたは複数の停止基準が満たされ、かつそれ以上の分岐が不可能になるまで、この処理が再帰的に実行されます。

使用される統計検定

未調整の p 値の計算は、対象フィールドのデータ型によって異なります。結合ステップでは、カテゴリはペア単位で比較されます。つまり、1 つの（結合）カテゴリが別の（結合）カテゴリと比較されます。このような比較では、現在のノード中のいずれかの比較カテゴリに所属しているレコードのみが比較の対象になります。分岐ステップでは、すべてのカテゴリが p 値の計算対象になります。したがって、現在のノードのすべてのレコードが使用されます。

尺度対象フィールド（F 検定）

スケール レベル対象フィールドがあるモデルの場合、p 値は標準の ANOVA F 検定に基づいて、考慮対象の予測フィールドのカテゴリ全体の対象値平均を比較して計算されます。F 統計は、次のように算出されます。

$$F = \frac{\sum_{i=1}^{I}\sum_{n\in D} w_n f_n I\left(x_n = i\right)\left(\overline{y}_i - \overline{y}\right)^2/(I-1)}{\sum_{i=1}^{I}\sum_{n\in D} w_n f_n I\left(x_n = i\right)\left(y_n - \overline{y}_i\right)^2/\left(N_f - I\right)}$$

p 値は次のようになります。

$$p = \Pr\left(F\left(I-1, N_f - I\right) > F\right)$$

ここで、

$$\overline{y}_i = \frac{\sum_{n\in D} w_n f_n y_n I(x_n = i)}{\sum_{n\in D} w_n f_n I(x_n = i)}, \overline{y} = \frac{\sum_{n\in D} w_n f_n y_n}{\sum_{n\in D} w_n f_n}, N_f = \sum_{n\in D} f_n$$

F(I − 1, N_f − I) は、自由度 (I − 1) と (N_f − I) を持つ F 分布に従うランダム変数です。

名義対象フィールド (カイ 2 乗検定)

対象フィールド Y がセット型 (カテゴリ) フィールドの場合、X と Y の独立に対する帰無仮説が検定されます。検定を行うために、Y のクラスを列、予測フィールド X のカテゴリを行とする分割 (カウント) 表が作成されます。独立に対する帰無仮説での期待セル度数が推定されます。観測セル度数と期待セル度数を使用してカイ 2 乗統計が計算され、p 値は算出された統計に基づいて計算されます。

Pearson のカイ 2 乗検定

Pearson のカイ 2 乗統計は、次のように算出されます。

$$X^2 = \sum_{j=1}^{J}\sum_{i=1}^{I} \frac{\left(n_{ij} - \hat{m}_{ij}\right)^2}{\hat{m}_{ij}}$$

ここで、 $n_{ij} = \sum_n f_n I(x_n = i \wedge y_n = j)$ は観測セル度数を表し、(x_n = i, y_n = j) に対し、下記の独立モデルから $\hat{m}_{ij}$ は期待セル度数を表しています。対応する p 値は、$p = \Pr\left(\chi_d^2 > X^2\right)$ のように計算されます。ここで、χ_d^2 は、自由度 d = (J − 1)(I − 1) のカイ 2 乗分布に従います。

カイ 2 乗テストの期待頻度数

尤度比のカイ 2 乗検定

尤度比のカイ 2 乗は、上記のように期待度数と観測度数に基づいて算出されます。尤度比のカイ 2 乗統計は、次のように算出されます。

$$G^2 = 2\sum_{j=1}^{J}\sum_{i=1}^{I} n_{ij} \ln\left(n_{ij}/\hat{m}_{ij}\right)$$

p 値は、$p = \Pr\left(\chi_d^2 > G^2\right)$ のように算出されます。

カイ 2 乗の期待度数の検定

ケースの重みのないモデルの場合、期待度数は次のように計算されます。

$$\hat{m}_{ij} = \frac{n_{i.}n_{.j}}{n_{..}}$$

ここで、

$$n_{i.} = \sum_{j=1}^{J} n_{ij},\ n_{.j} = \sum_{i=1}^{I} n_{ij},\ n_{..} = \sum_{j=1}^{J}\sum_{i=1}^{I} n_{ij}.$$

重みが指定された場合、独立に対する帰無仮説での期待セル度数の形式は、次のようになります。

$$m_{ij} = \overline{w}_{ij}^{-1}\alpha_i\beta_j$$

ここで、α_i および βj は推定対象のパラメータを表します。また、

$$\overline{w}_{ij} = \frac{w_{ij}}{n_{ij}},\ w_{ij} = \sum_{n\in D} w_n f_n I(x = i \wedge y_n = j).$$

パラメータは $\hat{\alpha}_i$, $\hat{\beta}_j$ を推定します。したがって、$\hat{m}_{ij}$ は次のような反復処理に基づいて計算されます。

1. 最初は k = 0 であるため、$\alpha_i^{(0)} = \beta_j^{(0)} = 1$、$m_{ij}^{(0)} = \overline{w}_{ij}^{-1}$
2. $\alpha_i^{(k+1)} = \frac{n_{i.}}{\sum_J \overline{w}_{ij}^{-1}\beta_j^{(k)}} = \alpha_i^{(k)}\frac{n_{i.}}{\sum_j m_{ij}^{(k)}}$.
3. $\beta_j^{(k+1)} = \frac{n_{.j}}{\sum_i \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}}$
4. $m_{ij}^{(k+1)} = \tilde{w}_{ij}^{-1}\alpha_i^{(k+1)}\beta_j^{(k+1)}$.
5. $\max_{i,j}\left|m_{ij}^{(k+1)} - m_{ij}^{(k)}\right| < \epsilon$ である場合、停止して、$\alpha_i^{(k+1)}$、$\beta_j^{(k+1)}$、および $m_{ij}^{(k+1)}$ を、$\hat{\alpha}_i$、$\hat{\beta}_j$、および $\hat{m}_{ij}$ の最終推定値として出力します。それ以外の場合は、k を増分してステップ 2 を繰り返します。

順序型対象フィールド（行効果モデル）

対象フィールド Y が順序型の場合、X のカテゴリを行、Y のカテゴリを列として、X と Y の独立に対する帰無仮説が行効果モデルに対して検定されます(Goodman, 1979)。$\hat{m}_{ij}$（独立に対する帰無仮説）と $\hat{\hat{m}}_{ij}$（データが行効果モデルに従うという仮説）という 2 つの期待セル度数が推定されます。尤度比統計量は次のように計算されます。

$$H^2 = 2\sum_{i=1}^{I}\sum_{j=1}^{J}\hat{m}_{ij}\ln\left(\hat{\hat{m}}_{ij}/\hat{m}_{ij}\right)$$

また、p 値は次のように算出されます。

$$p = \Pr\left(\chi^2_{I-1} > H^2\right)$$

行効果モデルの期待セル度数

行効果モデルの場合、Y のカテゴリのスコアが必要になります。デフォルトでは、カテゴリ スコアとして各カテゴリの順序が使用されます。ユーザーは独自のスコアのセットを指定できます。行効果モデルでの期待セル度数は、次のように計算されます。

$$m_{ij} = \overline{w}_{ij}^{-1}\alpha_i\beta_j\gamma_i^{(s_j-\overline{s})}$$

ここで、s_j は Y の カテゴリ j のスコアです。また、

$$\overline{s} = \frac{\sum_{j=1}^{J} w_{\cdot j}s_j}{\sum_{j=1}^{J} w_{\cdot j}}$$

ここで、$w_{\cdot j} = \sum_i w_{ij}$、$\alpha_i$、$\gamma_j$、および γ_i は推定対象の不明なパラメータです。

パラメータは、$\hat{\alpha}_i$、$\hat{\beta}_j$、$\hat{\gamma}_i$ を推定するため、$\hat{\hat{m}}_{ij}$ は次のような反復処理を用いて計算されます。

1. $k = 0, \alpha_i^{(0)} = \beta_j^{(0)} = \gamma_i^{(0)} = 1, m_{ij}^{(0)} = \overline{w}_{ij}^{-1}$
2. $\alpha_i^{(k+1)} = \frac{n_{\cdot j}}{\sum_j \overline{w}_{ij}^{-1}\beta_j^{(k)}\left(\gamma_i^{(k)}\right)^{(s_j-\overline{s})}} = \alpha_i^{(k)}\frac{n_{i\cdot}}{\sum_j m_{ij}^{(k)}}$
3. $\beta_j^{(k+1)} = \frac{n_{\cdot j}}{\sum_i \overline{w}_{ij}^{-1}\alpha_i^{(k)}\left(\gamma_i^{(k)}\right)^{(s_j-\overline{s})}}$
4. $m_{ij}^{*} = \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}\beta_j^{(k+1)}\left(\gamma_i^{(k)}\right)^{(s_j-\overline{s})}$, $G_i = 1 + \frac{\sum_j (s_j-\overline{s})\left(n_{ij}-m_{ij}^{*}\right)}{\sum_j (s_j-\overline{s})^2 m_{ij}^{*}}$
5. $\gamma_i^{(k+1)} = \begin{cases} \gamma_i^{(k)}G_i & G_i > 0 \\ \gamma_i^{(k)} & \text{そうでない場合} \end{cases}$
6. $m_{ij}^{(k+1)} = \overline{w}_{ij}^{-1}\alpha_i^{(k+1)}\beta_j^{(k+1)}\left(\gamma_i^{(k+1)}\right)^{(s_j-\overline{s})}$
7. $\max_{i,j}\left|m_{ij}^{(k+1)} - m_{ij}^{(k)}\right| < \epsilon$ である場合、停止し、$\alpha_i^{(k+1)}$、$\beta_j^{(k+1)}$、$\gamma_i^{(k+1)}$、および $m_{ij}^{(k+1)}$ を、$\hat{\alpha}_i$、$\hat{\beta}_j$、$\hat{\gamma}_i$、および $\hat{\hat{m}}_{ij}$ の最終推定値として設定します。それ以外の場合は、k を増分してステップ 2 を繰り返します。

Bonferroni の調整

調整済み p 値は、p 値と Bonferroni の乗数を乗じた値として計算されます。Bonferroni の乗数は、複数の統計検定に渡って全体的な p 値を制御します。

たとえば、元の予測フィールドのカテゴリ数が I で、結合処理後にカテゴリ数が r に減ったとします。Bonferroni B は、I 個のカテゴリを r 個のカテゴリに結合するための有効な方法です。r = I の場合、B = 1 です。2 ≤ r < I の場合は次のようになります。

$$B = \begin{cases} \binom{I-1}{r-1} & \text{順序型予測フィールド} \\ \sum_{v=0}^{r-1}(-1)^v \frac{(r-v)^I}{v!(r-v)!} & \text{名義型予測フィールド} \\ \binom{I-2}{r-2} + r\binom{I-2}{r-1} & \text{欠損値のある序数} \end{cases}$$

空白の処理

レコードの対象フィールドが空白の場合、または予測フィールドがすべて空白の場合、モデル構築時にレコードは無視されます。ケースの重みが指定されていて、レコードのケースの重みが空白、ゼロ、または負の場合、レコードは無視され、度数の重みについてもレコードは無視されます。

その他のレコードの場合、予測フィールドの空白はそのフィールドの追加のカテゴリとして扱われます。

順序型予測フィールド

アルゴリズムはまず、すべて非空白の情報を用いてカテゴリの最良のセットを生成し、次に、空白のカテゴリにもっとも類似するカテゴリを識別します。最後に、2 つの p 値が算出されます。空白のカテゴリをそれにもっとも類似するカテゴリと結合して生成されたカテゴリのセットについての p 値と、空白のカテゴリを別個のカテゴリとして追加することで生成されたカテゴリのセットについての p 値です。もっとも小さな p 値を持つカテゴリのセットが使用されます。

名義型予測フィールド

分析では、欠損カテゴリは他のカテゴリと同様に扱われます。

オプションの効果

停止規則

停止基準は、ツリー中のノードの分岐をいつ停止するかを判断するために用いられます。ツリーの成長は、ツリー中の各枝葉ノードが最低 1 つの停止基準を満たすまで継続します。ノードの分岐を停止する条件を次に示します。

- ノードが純粋の場合（すべてのレコードの対象フィールドが同じ値を持つ）

- モデルが使用するすべての予測フィールドに対して、ノード中のすべてのレコードが同じ値を持つ場合
- 現在のノードのツリーの深さ（現在のノードを定義する帰納的ノード分岐数）が最大ツリー深さ（デフォルト値またはユーザー指定）の場合
- ノード中のレコード数が最小親ノード サイズ（デフォルト値またはユーザー指定）未満の場合
- ノードの最良の分岐点の結果となる任意の子ノード中のレコード数が、最小子ノード サイズ（デフォルト値またはユーザー指定）未満の場合
- ノードの最良の分岐点により α_{split}（デフォルト値またはユーザー指定）より大きい p 値が生じる場合。

プロフィット

プロフィットは、（シンボル値）対象フィールドのカテゴリに関連する数値で、セグメントに関連するゲインまたはロスを推定するために用いられます。プロフィットは、対象フィールドの各値の相対値を定義しています。値は、ゲインを算出するために使われますが、ツリーの成長には使用されません。

ツリー中の各ノードのプロフィットは、次のように計算されます。

$$\sum_j f_j(t) P_j$$

ここで j は対象フィールドのカテゴリ、$f_j(t)$ は対象フィールドに対するカテゴリ j と、ノード t 中のすべてのレコードの度数フィールド値の合計、そして P_j はカテゴリ j に対するユーザーが定義したプロフィットの値を表します。

スコア値

CHAID と Exhaustive CHAID では**スコア**を使用できます。スコアは、順序型のカテゴリ対象フィールドのカテゴリの順序とカテゴリ間の距離を定義します。言い換えれば、スコアはフィールドの尺度を定義するものです。スコアの値はツリーの成長に含まれます。

ユーザー定義のスコアが指定された場合、上記で説明するように、そのスコアが期待セル度数の計算に使用されます。

コスト

コストを指定した場合、CHAID ツリーの成長にはそのコストは考慮されません。ただし、ノードの割り当てとリスクの推定時にはコストが考慮されます。後述する「予測値」および「リスク推定」を参照してください。

二次計算

二次計算はモデルの構築に直接には関係していませんが、モデルとそのパフォーマンスに関する情報を得ることができます。

リスク推定

リスク推定は、ツリーの特定ノードおよびツリー全体の予測フィールド中の誤差リスクを表しています。

シンボル値フィールドに対するリスク推定

分類ツリー（シンボル値対象フィールドを持つ）の場合、ノード t のリスク推定値 r(t) は次のように算出されます。

$$r(t) = \frac{1}{N_f} \sum_j N_{f,j}(t) C(j^*(t)|j)$$

ここで C(j*(t)|j) は対象値 j を j*（t）として持つレコードを分類する際の誤分類コストを、$N_{f,j}(t)$ はカテゴリ j の ノード t 中のレコードの度数の重みの合計（または度数の重みが定義されていない場合はレコードの数）、N_f は学習データ中のすべてのレコードの度数の重みの合計を表します。

ケースの重みはリスク推定値の計算では考慮されません。

数値型対象フィールドのリスク推定

回帰ツリー（数値型対象フィールドを持つ）の場合、ノード t のリスク推定値 r(t) は次のように算出されます。

$$r(t) = \frac{1}{N_f(t)} \sum_{i \in t} f_i (y_i - \overline{y}(t))^2$$

ここで f_i はレコード i（ノード t に割り当てられたレコード）の度数の重み、y_i はレコード i の対象フィールドの値、そして $\overline{y}(t)$ はノード t 中のすべてのレコードに対する、重み付けられた対象フィールドの平均を表します。

ツリーのリスク推定

分類ツリーと回帰ツリーの両方で、ツリー（T）のリスク推定値 R(T) は、ターミナルノード r(t) のリスク推定値の合計を取得することにより算出されます。

$$R(T) = \sum_{t \in T'} r(t)$$

ここで T’ はツリー中のターミナル ノードのセットを表します。

ゲインの要約

ゲインの要約は、ツリーのターミナル ノードの記述統計量を提供しています。

対象フィールドが連続型（スケール）の場合、ゲインの要約は各ターミナル ノードの対象値の重み付けられた平均を表します。

$$g(t) = \sum_{i \in t} w_i f_i x_i$$

対象フィールドがシンボル値（カテゴリ）の場合は、選択した対象カテゴリ中のレコードの重み付けられた割合を表します。

$$g\left(t, j\right) = \frac{\sum_{i \in t} f_i x_i(j)}{\sum_{i \in t} f_i}$$

ここでレコード x_i が対象カテゴリ j 中にある場合 $x_i(j) = 1$ に、それ以外の場合は 0 になります。ツリーのプロフィットが定義されている場合、各ターミナル ノードのプロフィット値の平均がゲインになります。

$$g(t) = \sum_{i \in t} f_i P(x_i)$$

ここで $P(x_i)$ はレコード x_i 中に観測される対象値に割り当てられたプロフィット値を表します。

生成されたモデル／スコアリング

CHAID 生成モデルにより行われる計算は、後述します。

予測値

新しいレコードは、ツリーのターミナル ノードへのツリー分割にしたがってスコアリングされます。各ターミナル ノードには、それに対応する予測フィールドがあります。予測フィールドは、次のように決定されます。

分類ツリー

シンボル値対象フィールドを持つツリーの場合、各ターミナル ノードの予測カテゴリは、ノードの重み付けられたコストが最も低いカテゴリになります。この重み付けられたコストは、次のように算出されます。

$$\min_i \sum_j C(i|j) p(j|t)$$

ここで C(i|j) はレコードが実際にはカテゴリ j にある場合に、レコードをカテゴリ i に分類する際の、ユーザーが定義した誤分類コストを、p(j|t) はカテゴリ j がノード t にある場合の、カテゴリ中のレコードの重み付けられた条件確率を表し、次のように定義されます。

$$p(j|t) = \frac{p(j,t)}{\sum_j p(j,t)}, p(j,t) = \pi(j)\frac{N_{w,j}(t)}{N_{w,j}}$$

ここで $\pi(j)$ はカテゴリ j, N_w の事前確率、$_j(t)$ はカテゴリ j を持つノード t 中の重み付けられたレコード数（度数またはケースの重みが指定されていない場合はレコード数）を表します。

$$N_{w,j}(t) = \sum_{i \in t} w_i f_i j(i)$$

$N_{w,j}$ は、カテゴリ j（ノードは任意）中の重み付けられたレコード数を表します。

$$N_{w,j} = \sum_{i \in T} w_i f_i j(i)$$

回帰ツリー

数値型対象フィールドを持つツリーの場合、各ターミナル ノードの予測カテゴリは、ノード中のレコードに対する対象値の重み付けられた平均になります。この重み付けられた平均は、次のように算出されます。

$$\overline{y}(t) = \frac{1}{N_w(t)} \sum_{i \in t} w_i f_i y_i$$

ここで $N_w(t)$ は、次のように定義されます。

$$N_w(t) = \sum_{i \in t} w_i f_i$$

Confidence

分類ツリーに関しては、生成モデルを通過するレコードに対する確信度値は、次のように計算されます。回帰ツリーには、確信度値は割り当てられません。

分類ツリー

スコア付きレコードの確信度とは、予測カテゴリに所属するターミナル ノードに割り当てられたスコア付きレコードに対する、学習データ内にある重み付けされたレコードの比率です。ラプラス補正による次のような変更も加えられます。

$$\frac{N_{f,j}(t) + 1}{N_f(t) + k}$$

空白の処理

新規レコードの分類において、空白はツリーの成長時のように処理されます。可能な場合には空白ではないカテゴリと結合され、追加のカテゴリとして扱われます。 詳細は、 p. 81 空白の処理 を参照してください。

学習データに空白がなかったノードの場合、そのノードの分岐には空白のカテゴリは存在しません。この場合、重み付けられた確率に基づいて分岐フィールドに空白値を持つレコードが割り当てられます。

$$j^*(t) = \max_j p(j|t)$$

ここで、

$$p(j|t) = \frac{N_{w,j}(t)}{N_w(t)}, N_{w,j}(t) = \sum_{i \in t} w_i f_i j(i), N_w(t) = \sum_{i \in t} w_i f_i$$

章
11

クラスタ評価アルゴリズム

ここでは、クラスタリング モデルを評価するために使用する指標について説明します。

- **シルエット係数は**、クラスタ結合の概念（密に結合するクラスタを含むモデルを選択）とクラスタ分割の概念（分割されたクラスタを含むモデルを選択）を結合します。シルエット係数を使用して、各オブジェクト、クラスタ、モデルを評価します。
- **平方和の誤差（SSE）**は、プロトタイプベースの結合の指標であり、**群間平方和（SSB）**は、プロトタイプ ベースの分割の指標です。
- **予測値の重要度**は、変数がさまざまなクラスタをどれだけ区別するかを示します。範囲型（数値）変数と離散型変数の両方で、重要度が高いほど、クラスタ間の変数の変動は偶然によるものではなく、何らかの潜在的な差異によるものである可能性が高くなります。

表記

この章では特に明記しない限り、次の表記を使用します。

x_{ik}	ケース i の連続型変数 k（標準化）。
x_{iks}	ケース i の変数 k の s 番目のカテゴリ（one-of-c コーディング）。
N	有効なケースの総数。
N_j	クラスタ j 内のケース数。
Y	J クラスタ ラベルの付いた変数。
μ_{jk}	変数 k の クラスタ j の重心。
D_{ij}	ケース i とクラスタ j の重心の距離。
D_j	全体の平均値 u とクラスタ j との距離。

適合度

平均シルエット係数は、各ケースに対する次の計算のすべてのケースの平均です。

$$(B - A) / \max(A, B)$$

ここで、A は、同じくラスタに割り当てられたケースからケースへの平均距離で、B は、すべてのクラスタ間においてケースから別のクラスタのケースへの平均最短距離です。

この係数は、計算に時間がかかります。この負担を軽くするには、A および B についての次の定義を使用します。

- A は、ケースからケースが属するクラスタの重心への距離です。
- B は、ケースから他のクラスタの重心への最短距離です。

距離は、ユークリッド距離を使用して計算できます。シルエット係数（およびその平均）は、-1（非常に悪いモデルを示す）から 1（非常に良いモデルを示す）です。Kaufman および Rousseeuw によって発見され（1990）たように、0.5 を超える平均シルエットはデータの適切な区分を示し、0.2 を下回る平均シルエットは、データがクラスタ構造を表さないことを示します。

データ準備

シルエット係数を計算する前に、次のようにケースを変換する必要があります。

1. **one-of-c コーディングを使用してカテゴリ変数を再コード化する**:変数に c 個のカテゴリがある場合、c 個のベクトルとして保存し、最初のカテゴリは (1,0,...,0)、次のカテゴリは (0,1,0,...,0)、最後のカテゴリは (0,0,...,0,1) と表記します。カテゴリの順序は、データ値の昇順または文字順です。
2. **連続型変数の再スケール**: 連続型変数は、変換 [2*(x-min)/(max-min)]-1 を使用して区間 [-1, 1] に正規化されます。この正規化によって、連続型フィールドおよびカテゴリ型フィールドの貢献度を均一にします。

基本統計量

次の統計を収集して、適合度を計算します。クラスタ j の変数 k の重心 μ_{jk}、ケースと重心間の距離、全体の平均値 u。

順序型および連続型変数 k の μ_{jk} の場合、クラスタ j 内の変数 k の標準化されたすべての値を平均化します。名義変数の場合、μ_{jk} は、クラスタ j の変数 k の各ステート s の発生する確率のベクトル $\{\varphi_{jks}\}$ です。カウント時、変数 k に欠損地を持つケースは考慮されません。変数 k の値は、クラスタ j ないのすべてのケースにおいて欠損しており、 μ_{jk} は欠損値としてマークされます。

ケース i とクラスタ j の重心間の距離 D_{ij}^2 は、すべての変数間の距離コンポーネント d_{ijk}^2 の重み付き合計によって計算されます。つまり、次のようになります。

$$D_{ij}^2 = \frac{\Sigma_k w_{ijk} d_{ijk}^2}{\Sigma_k w_{ijk}}$$

ここで、w_{ijk} は重みを示します。この時点で、差分の重みを考慮しないため、ケース i の変数 k が有効である場合、w_{ijk} は 1 となり、そうでない場合は 0 となります。すべての w_{ijk} は 0 になる場合、$D_{ij}^2 = 0$ を設定します。

順序型変数および連続型変数の距離コンポーネント d_{ijk}^2 は、次のように計算されます。

$$d_{ijk}^2 = \left(x_{ik} - \mu_{jk}\right)^2$$

バイナリ変数または名義変数の場合は次のようになります。

$$d_{ijk}^2 = \frac{1}{S_k}\sum_{s=1}^{S_k}\left(x_{iks} - \varphi_{jks}\right)^2$$

この場合、変数 k は、one-of-c コーディングを使用し、S_k はステートの数となります。

D_j の計算は D_{ij} の計算と同じですが、全体の平均値 u は μ_{jk} の代わりに、μ_{jk} が x_{ik} の代わりに使用されます。

シルエット係数

ケース i のシルエット係数は次のようになります。

$$\frac{\min\left\{D_{ij}, j \in C_{-i}\right\} - D_{ic_i}}{\max\left(\min\left\{D_{ij}, j \in C_{-i}\right\}, D_{ic_i}\right)}$$

この場合、C_{-i} は、ケース i をメンバーとして含まないクラスタ ラベルを示し、c_i はケース i を含むクラスタ ラベルとなります。$\max\left(\min\left\{D_{ij}, j \in C_{-i}\right\}, D_{ic_i}\right)$ が 0 となる場合、ケース i のシルエットは平均操作で使用されます。

これらの各データに基づき、全体平均のシルエット係数は次のようになります。

$$SC = \frac{1}{N}\sum_{i=1}^{N}\frac{\min\left\{D_{ij}, j \in C_{-i}\right\} - D_{ic_i}}{\max\left(\min\left\{D_{ij}, j \in C_{-i}\right\}, D_{ic_i}\right)}$$

平方和の誤差 (SSE)

SSE は、平方されたユークリッド距離が使用される、プロトタイプベースの結合指標です。モデル間で比較するために、次のように定義された平均化形式を使用します。

$$\text{Average SSE} = \frac{1}{N}\sum_{j\in C}\sum_{i\in j} D_{ij}^2$$

群間平方和 (SSB)

SSB は、平方されたユークリッド距離が使用される、プロトタイプベースの分割指標です。モデル間で比較するために、次のように定義された平均化形式を使用します。

$$\text{Average SSB} = \frac{1}{N}\sum_{j \in C} N_j D_j^2$$

予測値の重要度

フィールド i の重要度は、次のように定義されます。

$$VI_i = \frac{-\log_{10}(sig_i)}{\max_{j \in \Omega}(-\log_{10}(sig_j))}$$

ここで Ω は予測フィールドと評価フィールドのセットを示し、sig_i は、有意度または以下で説明するように特定の検定を適用して算出する p 値になります。sig_i が 0 の場合、$sig_i = MinDouble$ を設定します。MinDouble 倍精度の最小値です。

クラスタ間

カテゴリ フィールドの p 値は、Pearson のカイ 2 乗に基づきます。これは次のように算出されます。

p-value = Prob($\chi_d^2 > X^2$),

ここで、

$$X^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(N_{ij} - \hat{N}_{ij}\right)^2 / \hat{N}_{ij}$$

ここで、$\hat{N}_{ij} = N_{i.}N_{.j}/N(X)$ となります。

- $N(X) = 0$ の場合、重要度は未定義または不明と設定されます。
- $N_{i.} = 0$ の場合、I から 1 を引くと各カテゴリは I' を取得します。
- $N_{.j} = 0$ の場合、J から 1 を引くと各クラスタは J' を取得します。
- $J' \leq 1$ または $I' \leq 1$ の場合、重要度は未定義または不明と設定されます。

自由度は、$\left(I'-1\right)\left(J'-1\right)$ となります。

連続型フィールドの p 値は、F 検定に基づきます。これは次のように算出されます。

p-value = Prob{$F(J-1, N-J) > F$},

ここで、

$$F = \frac{\sum_{j=1}^{J} N_j \left(\overline{x}_j - \overline{\overline{x}}\right)^2 / (J-1)}{\sum_{j=1}^{J} (N_j - 1)\, s_j / (N-J)}$$

- N=0 の場合、重要度は未定義または不明と設定されます。
- $N_j = 0$ の場合、J から 1 を引くと各クラスタは J' を取得します。
- $J' \leq 1$ または $N \leq J'$ の場合、重要度は未定義または不明と設定されます。
- F 統計の式の分母が 0 の場合、重要度は未定義または不明と設定されます。
- F 統計の式の分子が 0 の場合、p 値が 1 となります。

自由度は $\left(J' - 1, N - J'\right)$ となります。

クラスタ内

カテゴリ フィールドの帰無仮説では、クラスタ j のカテゴリのケースの割合が全体の割合と同じとなります。

クラスタ j のカイ 2 乗統計量は次のように計算されます。

$$X^2 = \sum_{i=1}^{I} \frac{(N_{ij} - N_j p_i)^2}{N_j p_i}$$

$N_j = 0$ の場合、重要度は未定義または不明と設定されます。

$p_i = 0$ の場合、I から 1 を引くと各クラスタは I' を取得します。

$I' \leq 1$ の場合、重要度は未定義または不明と設定されます。

自由度は $d = I' - 1$ となります。

連続型フィールドの帰無仮説では、クラスタ j の平均値が全体の平均値と同じとなります。

Student 版のクラスタ j の t 統計量は次のように計算されます。

$$t = \frac{\left(\overline{x}_j - \overline{\overline{x}}\right)}{s_j / \sqrt{N_j}}$$

ここで自由度は $d = N_j - 1$ となります。

$N_j \leq 1$ または $s_j = 0$ の場合、重要度は未定義または不明と設定されます。

分子が 0 の場合、p 値が 1 となります。

ここで、Student 版の t 分布に基づく p 値は次のように計算されます。

p-value = 1 − Prob{$|T(d)| \leq |t|$}.

参照

Kaufman, L., および P. J. Rousseeuw. 1990. Finding groups in data:An introduction to cluster analysis. New York: John Wiley and Sons.

Tan, P., M. Steinbach, および V. Kumar. 2006. Introduction to Data Mining. : Addison-Wesley.

章
12

COXREG アルゴリズム

Cox 回帰アルゴリズム

Cox（1972年）は、ライフタイムに関連した要素がハザード関数に乗法効果のあるモデルを最初に提案しました。これらのモデルは比例ハザードモデルと呼ばれます。比例ハザードを想定して、X とした t の h ハザード関数は次の形式となります。

$$h(t|\mathbf{x}) = h_0(t)\, e^{\mathbf{x}'\beta}$$

ここでは、 $\mathbf{x}$ は個体に関連したリグレッサー変数の既知ベクトル、β は未知のパラメーターのベクトル、そして$h_0(t)$は$\mathbf{x} = 0$ の個体のベースライン ハザード関数です。よって $\mathbf{x}_1$ と$\mathbf{x}_2$ をセットするどの2つの共変量に対しても、 ログ ハザード関数 $h(t|\mathbf{x}_1)$ と $h(t|\mathbf{x}_2)$ は時間と並行しなくてはいけません。

因子が乗法でハザード関数に影響しない時は、階層化がモデル構築に役立つ場合があります。個体が 1 つ以上の因子レベルで定義された m の異なる階層の 1 つに割り当てられるとします。j 番目の階層の個体に対するハザード関数は次のように定義されます。

$$h_j(t|\mathbf{x}) = h_{0j}(t)\, e^{\mathbf{x}'\beta}$$

モデルには回帰パラメーターβ とベースライン ハザード関数 $h_{0j}(t)$、2 つの未知のコンポーネントがあります。パラメーターの推定は下記で説明します。

推定

いくつかの母集団における個体のライフタイムを表す非負のランダム可変 T を検討することから始めます。$f(t|\mathbf{x})$ でリグレッサー x を仮定した T の確率密度関数（PDF）を示し、また $S(t|\mathbf{x})$ を生存関数（t まで個体が生存する確率)とします。したがって

$$S(t|\mathbf{x}) = \int_t^\infty f(u|\mathbf{x})\, du$$

ハザード $h(t|\mathbf{x})$ は次によって定義されます。

$$h(t|\mathbf{x}) = \frac{f(t|\mathbf{x})}{S(t|\mathbf{x})}$$

$h(t|\mathbf{x})$ に関して $S(t|\mathbf{x})$ に役立つ他の式は次のとおりです。

$$S(t|\mathbf{x}) = \exp\left(-\int_0^t h(u|\mathbf{x})\, du\right)$$

したがって、

$$\ln S(t|\mathbf{x}) = -\int_0^t h(u|\mathbf{x})\,du$$

いくつかの目的に対しては、累積ハザード関数を定義することも有用です。

$$H(t|\mathbf{x}) = \int_0^t h(u|\mathbf{x})\,du = -\ln S(t|\mathbf{x})$$

比例ハザードを想定して、生存関数は次のように書き込めます。

$$S(t|\mathbf{x}) = [S_0(t)]^{\exp\left(\mathbf{x}'\beta\right)}$$

ここでは、$S_0(t)$ は次によって定義されるベースライン生存関数です。

$$S_0(t) = \exp\left(-H_0(t)\right)$$

および

$$H_0(t) = \int_0^t h_0(u)\,du$$

後ほど使用する $S(t|\mathbf{x})$, $H(t|\mathbf{x})$ と $H_0(t)$、 $S_0(t)$ と $h_0(t)$ の関係は次のとおりです。

$$\ln S(t|\mathbf{x}) = -H(t|\mathbf{x}) = -\exp\left(\mathbf{x}'\beta\right)H_0(t)$$

$$\ln\left(-\ln S(t|\mathbf{x})\right) = \mathbf{x}'\beta + \ln H_0(t)$$

生存関数 $S(t|\mathbf{x})$ を推定するには、推定する必要のある β と $S_0(t)$、2 つのコンポーネントがある生存関数の式から確認します。ここで用いるアプローチは部分的尤度関数から β を推定し、次に $S_0(t)$ の完全尤度を最大にします。

ベータの推定

以下を想定します

- 階層変数に m レベルが存在する。
- 同じ階層の個体には比例ハザード関数がある。
- リグレッサー変数の相対効果が各階層で同じである。

$t_{j1} < \cdots < t_{jk_j}$を j 番目の階層における n_j 個体の観察された無検閲な失敗時間とし、また $x_{j1}?\ldots?x_{jk_j}$ を対応する共変量とします。次に部分尤度関数が次によって定義れます。

$$L(\beta) = \prod_{j=1}^{m}\prod_{i=1}^{k_j}\frac{e^{\mathbf{s}'_{ji}\beta}}{\left(\sum_{l\in R_{ji}} w_l e^{\mathbf{x}'_l\beta}\right)^{d_{ji}}}$$

d_{ji} が t_{ji} と等しいライフタイムの個体のケースの重み合計で、$\mathbf{S}_{ji}$ がこれら d_{ji} 個体の回帰ベクトル $\mathbf{x}$ の重みづけされた合計の時、 w_l は個体 1 のケースの重みで、R_{ji} は j 番目の階層における t_{ji} 直前の生きた無検閲の個体セットです。したがって部分尤度関数から生じる対数尤度は次のとおりです。

$$l = \ln L(\beta) = \sum_{j=1}^{m}\sum_{i=1}^{k_j} \mathbf{S}'_{ji}\beta - \sum_{j=1}^{m}\sum_{i=1}^{k_j} d_{ji} \ln\left(\sum_{l\in R_{ji}} w_l e^{\mathbf{x}'_l\beta}\right)$$

そして 1 の最初の導関数は 次のとおりです。

$$D_{\beta_r} = \frac{\partial l}{\partial \beta_r} = \sum_{j=1}^{m}\sum_{i=1}^{k_j}\left(S_{ji}^{(r)} - d_{ji}\frac{\sum_{l\in R_{ji}} w_l x_{lr} e^{\mathbf{x}'_l\beta}}{\sum_{l\in R_{ji}} w_l e^{\mathbf{x}'_l\beta}}\right), \quad r = 1,\ldots,p$$

$S_{ji}^{(r)}$ は $\mathbf{S}_{ji} = \left(S_{ji}^{(1)},\ldots,S_{ji}^{(p)}\right)'$ の r 番目のコンポーネントです。β の最大部分尤度推定（MPLE）は、 p がモデル内の独立変数の数である時、$r = 1,\ldots,p$ に関して $\frac{\partial l}{\partial \beta_r}$ をゼロに等しく設定することで得られます。$\frac{\partial l}{\partial \beta_r} = 0 \quad (r = 1,\ldots,p)$ の方程式は通常ニュートンラプソン法を用いて解くことができます。

その方程式から部分尤度関数 $L(\beta)$ は翻訳で不変です。全ての共変量は対応する全体平均で集められます。共変量の全体平均は、各階層における検閲および無検閲のケースの重みや共変量の積合計として定義されます。単純表記のため、推定セクションで使用する $\mathbf{x}_l$ は集められた共変量を示します。

ニュートンラプソン法には 3 つの収束基準が利用できます。

- 以前の反復のパラメータ推定の値で割った反復作業間 (δ) のパラメータ推測における最大差の絶対値

  $$\text{BCON} = \left|\frac{\delta}{\text{前の 反復 の パラメータ推定}}\right|$$

- 以前の反復作業の対数尤度関数で割った反復作業間の対数尤度関数の絶対差。
- 反復作業の最大数

MPLE $\hat{\beta} = \left(\hat{\beta}_1,...,\hat{\beta}_p\right)'$ の漸近分散共分散行列は、I が $\ln L$ の 2 番目の部分導関数を引いた情報行列で $\mathbf{I}^{-1}$ によって推定されます。.I の（r, s）番目のエレメントは次によって定義されます。

$$\begin{aligned}\mathbf{I}_{rs} &= -E\frac{\partial^2}{\partial\beta_r\partial\beta_s}\ln L \\ &= \sum_{j=1}^{m}\sum_{i=1}^{k_j} d_{ji}\left[\frac{\sum_{l\in R_{ji}} w_l x_{ls} x_{lr} e^{\mathbf{x}'_l\beta}}{\sum_{l\in R_{ji}} w_l e^{\mathbf{x}'_l\beta}} - \frac{\left(\sum_{l\in R_{ji}} w_l x_{lr} e^{\mathbf{x}'_l\beta}\right)\left(\sum_{l\in R_{ji}} w_l x_{ls} e^{\mathbf{x}'_l\beta}\right)}{\left(\sum_{l\in R_{ji}} w_l e^{\mathbf{x}'_l\beta}\right)^2}\right]\end{aligned}$$

次のように行列形式で I を書き込むこともできます。

$$I_{rs} = \sum_{j=1}^{m} \sum_{i=1}^{k_j} d_{ji} \left(x^{'}(t_{ji}) \right) V(t_{ji}) \left(x(t_{ji}) \right)$$

ここでは $\mathbf{x}(t_{ji})$ が、t_{ji} の時に評価されたモデル内の p 共変量変数を表す $n_{ji} \times p$で、n_{ji} は R_{ji} における異なる個体の数、そして $\mathbf{V}(t_{ji})$ は次によって定義される l 番目の対角エレメント $v_{ll}(t_{ji})$ を持つ $n_{ji} \times n_{ji}$ 行列です。

$$v_{ll}(t_{ji}) = p_l(t_{ji}) w_l - \left(w_l p_l(t_{ji}) \right)^2$$

$$p_l(t_{ji}) = \frac{\exp\left(\mathbf{x}^{'}{}_l \hat{\beta} \right)}{\sum_{h \in R_{ji}} w_h \exp\left(\mathbf{x}^{'}{}_h \hat{\beta} \right)}$$

および（l, k）エレメント $v_{lk}(t_{ji})$ は次によって定義されます。

$$v_{lk}(t_{ji}) = w_l p_l(t_{ji}) \times w_k p_k(t_{ji})$$

ベースライン関数の推定

β の MPLE $\hat{\beta}$ が検出されると、ベースライン生存関数の $S_{0j}(t)$ は各階層ごとに別々に推定されます。1 つの階層には、サンプル内で $t_1 < \cdots < t_k$ がライフタイムに観察されたと仮定します。t_i では、リスクにある n_i と 死亡した d_i があり、また $[t_{i-1}, t_i)$ 区間では λ_i 検閲時間があります。$S_0(t)$ は生存関数で増加せずに継続されるため、$\hat{S}_0(t)$ は観測ライフタイムの$t_1, \ldots, t_k$ においてジャンプ以外は一定である必要があります。

さらに以下が続きます

$$\hat{S}_0(t_1) = 1$$

そして

$$\hat{S}_0(t_i+) = \hat{S}_0(t_{i+1})$$

$\hat{S}_0(t_i+) = p_i (i = 1, \ldots, k)$ の書き込みで、観測尤度関数は次の形式です。

$$L_1 = \prod_{i=1}^{k} \left\{ \prod_{l \in D_i} \left(p_{i-1}^{\exp\left(\mathbf{x}^{'}{}_l \beta \right)} - p_i^{\exp\left(\mathbf{x}^{'}{}_l \beta \right)} \right)^{w_l} \prod_{l \in C_i} \left(p_{i-1}^{\exp\left(\mathbf{x}^{'}{}_l \beta \right)} \right)^{w_l} \right\} \prod_{l \in C_{k+1}} \left(p_k^{\exp\left(\mathbf{x}^{'}{}_l \beta \right)} \right)^{w_l}$$

ここでは D_i は t_i で死亡している個体のセット、そして C_i は $[t_{i-1}, t_i)$ において検閲時間のある個体のセットです。（最後の観測が無検閲の場合、C_{k+1} は空で $p_k = 0$ です。）

$\alpha_i = p_i / p_{i-1} (i = 1, \ldots, k)$ とした場合、L_1 は次のように書き込むことができます。

$$L_1 = \prod_{i=1}^{k} \prod_{l \in D_i} \left(1 - \alpha_i^{\exp\left(\mathbf{x}^{'}{}_l \beta \right)} \right)^{w_l} \prod_{l \in R_i - D_i} \alpha_i^{w_l \exp\left(\mathbf{x}^{'}{}_l \beta \right)}$$

$\alpha_1,\ldots,\alpha_k$ に関して $\ln L_1$ を微分し、ゼロに等しい方程式を設定すると、次が得られます。

$$\sum_{l\in D_i}\frac{w_l\exp\left(\mathbf{x}'_l\beta\right)}{1-\alpha_i^{\exp\left(\mathbf{x}'_l\beta\right)}}=\sum_{l\in R_i}w_l\exp\left(\mathbf{x}'_l\beta\right)\quad i=1,\ldots,k$$

次に β の MPLE $\hat{\beta}$ をこの方程式に当てはめ、これらの k 方程式を別々に解きます。

次の 2 点は意味がありません。

- $|D_i|=1$, $\hat{\alpha}_i$ のいずれかが明確に解ける場合。

  $$\hat{\alpha}_i=\left[1-\frac{w_i\exp\left(\mathbf{x}'_i\hat{\beta}\right)}{\sum_{l\in R_i}w_l\exp\left(\mathbf{x}'_l\hat{\beta}\right)}\right]^{\exp\left(-\mathbf{x}'_l\overline{\beta}\right)}$$

- $|D_i|>1$、 累積統計ハザード関数の方程式を $\hat{\alpha}_i$ に対して反復して解く必要がある場合。$\hat{\alpha}_i$ の良い初期値は以下のとおりです。

  $$\hat{\alpha}_i=\exp\left(\frac{-d_i}{\sum_{l\in R_i}w_l\exp\left(\mathbf{x}'_l\beta\right)}\right)$$

  ここでは、$d_i=\sum_{l\in D_i}w_l$ は D_i セットの重さ合計です。(Lawless, 1982年, p. 361 を参照。)

$\hat{\alpha}_i$, $i=1,\ldots,k$ が検出されると、$S_0(t)$ は以下によって推定されます。

$$\hat{S}_0(t)=\prod_{i:(t_i<t)}\hat{\alpha}_i$$

同じ値が存在すると $S_0(t)$ の上記推定は反復計算を要するため、Breslow (1974年) は $|D_i|>1$ が推定値の時に α_i の方程式を用いることを提案しています。しかし私たちはこれを初期推定値として使用します。

$-\ln\hat{S}_0(t)$ の漸近変数は Kalbfleisch and Prentice (1980年) の第 4 章に記載されています。特定の時間 t に、常時次のように推定されています。

$$var\left(-\ln\hat{S}_0(t)\right)=\sum_{t_i<t}|D_i|\left(\sum_{l\in R_i}w_l\exp\left(\mathbf{x}'_l\hat{\beta}\right)\right)^{-2}+\mathbf{a}'\mathbf{I}^{-1}a$$

ここでは、次のように定義される j 番目のエレメントで a は p×1 ベクトルです。

$$\sum_{t_i<t}|D_i|\frac{\sum_{l\in R_i}w_lx_{lj}\exp\left(\mathbf{x}'_l\hat{\beta}\right)}{\left(\sum_{l\in R_i}w_l\exp\left(\mathbf{x}'_l\hat{\beta}\right)\right)^2}$$

そして I は情報クロス集計です。$\hat{S}(t|x)$ の漸近分散は以下によって推測されます。

$$e^{2\mathbf{x}'\hat{\beta}}\left(\hat{S}(t|\mathbf{x})\right)^2 var\left(-\ln\hat{S}_0(t)\right)$$

ステップワイズ法の選択統計値

可変選択の同じ方法は、 2 分岐ロジスティック回帰でも提供されています。詳細は、22 章 p. 207 ステップワイズ変数選択 を参照してください。 ここではワルド、LR、および条件の 3 つの除去統計およびスコア入力統計だけを定義します。

スコア統計

スコア統計は、どの変数をモデルに加えるかを判断するために、モデル内にない各変数に対して計算されます。まずモデル内の変数のパラメータ推定値およびモデル内にない変数のゼロパラメータ推定値に基づいた、全ての的確な変数の情報クロス集計 I を計算します。次に結果 I を次のように 4 つのサブ列に分割します。

$$\begin{bmatrix}\mathbf{A}_{11} & \mathbf{A}_{12}\\ \mathbf{A}_{21} & \mathbf{A}_{22}\end{bmatrix}$$

ここでは、$\mathbf{A}_{11}$ と $\mathbf{A}_{22}$ はモデル内の変数およびモデル内にない変数それぞれの平方行列で、$\mathbf{A}_{12}$ は内外の変数に対するクロス積行列です。変数 $\mathbf{x}_i$ のスコア統計は次のように定義されます。

$$\mathbf{D}'_{x_i}\mathbf{B}_{22,i}\mathbf{D}_{x_i}$$

ここでは、$\mathbf{D}_{x_i}$ が $\mathbf{x}_i$ および $\mathbf{B}_{22,i}$ に関連した全てのパラメータに関する対数尤度の最初の導関数で、は $\left(\mathbf{A}_{22,i}-\mathbf{A}_{21,i}\mathbf{A}_{11}^{-1}\mathbf{A}_{12,i}\right)^{-1}$ に等しく、そして $\mathbf{A}_{22?i}$ そして $\mathbf{A}_{12,i}$ は関数 $\mathbf{x}_i$に関連した $\mathbf{A}_{22}$ および $\mathbf{A}_{12}$ におけるサブ行列です。

ワルド統計

ワルド統計はモデル内の変数を計算して除去する変数を選択します。変数 $\mathbf{x}_i$ のワルド統計は次のように定義されます。

$$\hat{\beta}_j'\mathbf{B}_{11,j}\hat{\beta}_j$$

ここでは、$\hat{\beta}_j$ が $\mathbf{x}_j$ および $\mathbf{B}_{11??????????????j}$ は $\mathbf{x}_j$ に関連した $\mathbf{A}_{11}^{-1}$ のサブ行列です。

LR（尤度比）統計

LR 統計は、そのMPLES で評価された 2 つのモデルの尤度関数の率の対数の 2 倍として定義されます。r 変数が現在のモデルにあると想定して、現在のモデルを完全モデルと呼ぶことにします。完全モデル用パラメータの MPLES に基づいて、l(full) は ベータの推定 に定義されます。完全モデルから削除した各 r 変数に対し、MPLESが検出され、縮小した数尤度関数 l(reduced) が計算されます。このとき LR 統計は、次のように定義されます。

-2(l(reduced) - l(full))

条件統計

条件統計も、モデル内のすべての変数に対して計算されます。条件統計の公式は LR 統計のものと同じですが、ただし、それぞれの縮小モデルのためのパラメータ推定値が MPLE ではなく条件推定になります。条件推定は次のように定義されます。$\hat{\beta} = \left(\hat{\beta}_1, \ldots, \hat{\beta}_r\right)'$ を r変数 （ブロック）のMPLESとし、C を $\hat{\beta}_i$ が次の場合のモデルに残されたパラメータの漸近分散共分散とします。

$$\tilde{\beta}_{(i)} = \hat{\beta}_{(i)} - \mathbf{C}_{12}^{(i)}\left(\mathbf{C}_{22}^{(i)}\right)^{-1}\hat{\beta}_i$$

$\hat{\beta}_i$ が $\mathbf{x}_i$ に関連するパラメータのMPLEで、$\hat{\beta}_{(i)}$ が $\hat{\beta}_i$ を伴わない $\hat{\beta}$ の時、 $\mathbf{C}_{12}^{(i)}$ は $\hat{\beta}_{(i)}$ および $\hat{\beta}_i$ に残されたパラメータ推定値間の分散共分散で、$\mathbf{C}_{22}^{(i)}$ は $\hat{\beta}_i$ の分散共分散です。そして変数 $\mathbf{x}_i$ の条件統計は次のように定義されます。

$$-2\left(l\left(\mathsf{b}_{(i)}\right) - l(full)\right)$$

ここでは、$l\left(\tilde{\beta}_{(i)}\right)$ は$\tilde{\beta}_{(i)}$ で評価される対数尤度関数です。

これら４つの統計には全て、対応モデルが持つパラメータの数と等しい自由度のカイ 2 乗分布があります。

統計

次の出力統計が利用できます。

初期モデル情報

最初の方法の初期モデルは共変量を含まないモデルを対象にしています。対数尤度関数 1 は次と同等です。

$$l(0) = -\sum_{j=1}^{m}\sum_{i=1}^{k_j} d_{ji}\ln\left(n_{ji}^{*}\right)$$

ここでは、n_{ji}^{*} はセット R_{ji} における個体の重み合計です。

モデル情報

ステップワイズ法が要求される時は、その各ステップにおいて、- 2 対数尤度関数と3つのカイ 2 乗統計（モデル カイ 2 乗、改善カイ 2 乗、および全体カイ 2 乗）、およびそれらに対応する自由度と有意確率が印刷されます。

-2 対数尤度

$$-2\sum_{j=1}^{m}\sum_{i=1}^{k_j}\left(\mathbf{s}'_{ji}\hat{\beta}-d_{ji}\ln\left(\sum_{l\in R_{ji}}w_l\exp\left(\mathbf{x}'_l\hat{\beta}\right)\right)\right)$$

ここで $\hat{\beta}$ は現在モデルのβ のMPLEです。

改善カイ 2 乗

(-以前のモデルの 2 対数尤度関数) - (現在のモデルの -2 対数尤度関数)

以前のモデルは前回のステップからのモデルです。自由度はこれら 2 つのモデル内で推定されたパラメータ数の差の絶対値と同等です。

モデル カイ 2 乗

(-初期モデルの 2 対数尤度関数) - (現在のモデルの -2 対数尤度関数)

初期モデルは前回の方法からの最終モデルです。自由度はこれら 2 つのモデルで推定されたパラメータ数の差の絶対値と同等です。

注 :モデル カイ 2 乗と改善カイ 2 乗の値は 0 またはそれ以下です。自由度が 0 に等しい場合、カイ 2 乗は印刷されません。

全体カイ 2 乗

全体カイ 2 乗統計は、モデル内の変数の全ての回帰係数が同様に 0 であるという仮説を検定します。この統計は次のように定義されます。

$$\mathbf{u}'(0)\mathbf{I}^{-1}\mathbf{u}(0)$$

ここでは、$\mathbf{u}(0)$ は $\beta=0$ で評価される部分的な対数尤度関数の最初の導関数のベクトルを表します。$\mathbf{u}$ と $\mathbf{I}$ のエレメントは ベータの推定 で定義されます。

方程式の変数に関する情報

方程式の単一変数それぞれに対し、MPLE、MPLE の SE、ワルド統計、それに対応する df、有意確率および偏相関 R が与えられます。単一変数に関しては、R は次のように定義されます。

$$R=\left[\frac{\text{Wald}-2}{-2\text{ log-likelihood for the intial model}}\right]^{1/2}\times\text{sign of MPLE}$$

Wald >2の場合。それ以外の場合 R は 0 に設定されます。複数カテゴリの変数に関しては、R が次のように定義される時、ワルド統計、df、有意確率、偏相関 R のみが印刷されます。

$$R=\left[\frac{\text{Wald}-2*\text{df}}{-2\text{ log-likelihood for the intial model}}\right]^{1/2}$$

Wald > 2 dfの場合。それ以外の場合 R は 0 に設定されます。

方程式にない変数に関する情報

方程式の中にないそれぞれの変数について、スコア統計が計算され、それに対応するた自由度、有意確率、および偏相関 R が印刷されます。方程式にない関数の偏相関 R は、ワルド統計からスコア統計へ変更することで方程式の変数のRと同様に定義されます。

残差カイ 2 乗と呼ばれる全体統計が1つあります。この統計は、方程式にない変数の全ての回帰係数が 0 の場合に検定を行います。それは次のように定義されます。

$$\mathbf{u}'\left(\hat{\beta}\right)\mathbf{B}_{22}\,\mathbf{u}\left(\hat{\beta}\right)$$

ここでは、$\mathbf{u}\left(\hat{\beta}\right)$ は、MPLE $\hat{\beta}$で評価される方程式にない全てのパラメータに関する部分的対数尤度関数の最初の導関数のベクトルで、 $\mathbf{B}_{22}$ は $\left(\mathbf{A}_{22}-\mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\right)^{-1}$に等しく、そしてA は スコア統計 で定義されています。

生存テーブル

各階層に対し、ベースライン累積統計生存 (S_0) およびハザード (H_0) 関数の推定値およびそれらの標準誤差が計算されます。$H_0(t)$ は次のように推定されます。

$$\hat{H}_0(t) = -\ln \hat{S}_0(t)$$

また $\hat{H}_0(t)$ の漸近変数はベースライン関数の推定 で定義されます。最後に、累積統計ハザード関数 $H(t|\mathbf{x})$ および 生存関数 $S(t|\mathbf{x})$ は次のように推定されます。

$$\hat{H}(t|\mathbf{x}) = \exp\left(\mathbf{x}'\hat{\beta}\right)\hat{H}_0(t)$$

および $\mathbf{x}$ の仮定に対してです。

$$\hat{S}(t|\mathbf{x}) = \left[\hat{S}_0(t)\right]^{\exp\left(\mathbf{x}'\hat{\beta}\right)}$$

漸近変数は次のとおりです。

$$var\left(\hat{H}(t|\mathbf{x})\right) = \exp\left(2\mathbf{x}'\hat{\beta}\right)var\left(\hat{H}_0(t)\right)$$

そして

$$var\left(\hat{S}(t|\mathbf{x})\right) = \exp\left(2\mathbf{x}'\hat{\beta}\right)\left(\hat{S}(t|\mathbf{x})\right)^2 var\left(\hat{H}_0(t)\right)$$

プロット

特定のパターンに対し、共変量値 $\mathbf{x}_c$ が確定され、また $\mathbf{x}_c$ が計算されます。Cox 回帰 には 3 つのプロットが利用できます。

生存プロット

階層 j に関しては、$\left(t_i, \hat{S}_0(t_i \,|\, \mathbf{x}_c)\right)$, $i = 1, \ldots, k_j$ が次でプロットされます。

$$\hat{S}(t_i \,|\, \mathbf{x}_c) = \left(\hat{S}_0(t_i)\right)^{\exp\left(\mathbf{x}'_c \hat{\beta}\right)}$$

ハザード プロット

階層 j に対しては、ここでは$\left(t_i, \hat{H}(t_i \,|\, \mathbf{x}_c)\right)$, $i = 1, \ldots, k_j$ がプロットされます。

$$\hat{H}(t_i \,|\, \mathbf{x}_c) = \exp\left(\mathbf{x}'_c \hat{\beta}\right) \hat{H}_0(t_i)$$

LML プロット

対数-対数関数プロットは階層変数を共変量として含めるべきかを確認するために使用します。階層 j に対して, $\left(t_i, \mathbf{x}'_c \hat{\beta} + \ln \hat{H}_0(t_i)\right)$, $i = 1, \ldots, k_j$ がプロットされます。プロットで階層に並列がある場合は、階層変数は共変量でなくてはいけません。

空白処理

入力フィールドまたは出力フィールドに欠損値があるすべてのレコードは、モデルの推定から除外されます。

スコアリング

生存および累積統計ハザードの推定値は 生存テーブル p. 101 で与えられています。

t_0 まで生存する条件で、 t まで生存する確率は次のとおりです。

$$\hat{S}\left(t + t_0 | t_0\right) = \frac{\hat{S}\left(t + t_0\right)}{\hat{S}\left(t_0\right)}$$

空白処理

最終モデルにおいて、入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールド $null$ が割り当てられます。

さらに、最長観測無検閲生存時間の記録を上回る「合計」生存時間（過去+未来）の記録もまた、$null$ の予測値が割り当てられます。

参照

Breslow, N. E. 1974. Covariance analysis of censored survival data. Biometrics, 30, 89-99.

Cain, K. C., および N. T. Lange. 1984. Approximate case influence for the proportional hazards regression model with censored data. Biometrics, 40, 493-499.

Cox, D. R. 1972. Regression models and life tables (with discussion). Journal of the Royal Statistical Society, Series B, 34, 187-220.

Kalbfleisch, J. D., および R. L. Prentice. 2002. The statistical analysis of failure time data, 2 ed. New York: John Wiley & Sons, Inc.

Lawless, R. F. 1982. Statistical models and methods for lifetime data. New York: John Wiley & Sons, Inc..

Storer, B. E., および J. Crowley. 1985. A diagnostic for Cox regression and general conditional likelihoods. Journal of the American Statistical Association, 80, 139-147.

章
13
ディシジョン リストのアルゴリズム

ディシジョン リストの目的は、たとえば高い確率での製品購入など、固有の行動パターンを持つ人々のグループを探し出すことです。ディシジョン リスト モデルは、ディシジョン ルールのセットから構成されます。ディシジョン ルールは if-then ルールであり、前提条件と結果の 2 つの部分からなります。**前提条件**は予測フィールドのブール式であり、**結果**は、前提条件が真の場合の対象フィールドの予測値です。ディシジョン リストの最も単純な構造は、1 つの予測フィールドに基づく 1 セグメントです。たとえば、Gender = 'Male' or $10 < Age \leq 20$ となります。

ルールの前提条件が真の場合、ルールによってレコードが**カバー**されます。ケースがディシジョン リスト内のルールの 1 つでカバーされると、リストによってカバーされると見なされます。

ディシジョン リスト内ではルールの順序が重要であり、ケースがあるルールでカバーされると、以後のルールは無視されます。

アルゴリズムの概要

ディシジョン リストのアルゴリズムは、以下のように要約できます。

- 候補のルールは、元のデータ セットから検出されます。
- 最適なルールがディシジョン リストに付加されます。
- ディシジョン リストにカバーされるレコードは、データ セットから削除されます。
- 新しいルールは、その減らされたデータ セットに基づいて検出されます。

この処理は、いずれかの停止基準を満たすまで繰り返されます。

ディシジョン リストのアルゴリズムの用語集

ディシジョン リストのアルゴリズムの説明に、次の用語が使用されます。

モデル: ディシジョン リスト モデル。

サイクル: 候補となるルールのセットが検出される、各ルール検出サイクル。その後、ルールが構築中のモデルに追加されます。結果のモデルが次のサイクルに投入されます。

属性: データ セット内の変数またはフィールドの別の呼び方。

入力属性: 予測フィールドの別の呼び方。

モデルの拡張: ディシジョン ルールをディシジョン リストへ追加すること、またはセグメントをディシジョン ルールへ追加すること。

グループ: データ セット内のレコードのサブセット。

セグメント: グループの別の呼び方。

基本的な計算

表記

ディシジョン リストのアルゴリズムの説明に、次の表記が使用されます。

X	データ行列:列はフィールド（属性）、行はレコード（ケース）です。
L	リスト モデルの集まり
L_i	L の i 番目のリスト モデル
L_{null}	ルールを含まないリスト モデル
$\hat{P}_{L_i}$	リスト L_i の推定レスポンス確率
N	母集団の合計サイズ
$X_{m,n}$	X の n 番目のレコード（行）の m 番目のフィールド（列）の値
X_{L_i}	リスト モデル L_i にカバーされる X 内のレコードのサブセット
Y	X 内の対象フィールド
Y_n	n 番目のレコードの対象フィールドの値
A	X のすべての属性（フィールド）の集まり
A_j	X の j 番目の属性
R	先行するルール リストを拡張するルールの集まり
R_k	ルール集合 R 内の k 番目のルール
T	候補リスト モデルのセット
ResultSet	ディシジョン リスト モデルの集まり

基本のアルゴリズム

ディシジョン リスト モデルを作成する基本のアルゴリズムは、次のとおりです。

1. モデルを初期化します。

- d = 検索の深さ、w = 検索の幅、とします。
- L = ∅ の場合、L_{null} を L に追加します。
- T = ∅ とします。

2. L のすべての要素 L_i。

- L_i のルールにカバーされていないレコード $X_{\overline{L_i}}$ を選択します。

$$X_{\overline{L_i}} = X - X_{L_i}$$

- ディシジョン ルールのアルゴリズムを呼び出して、代替ルール セット R を$X_{\overline{L_i}}$ に対して作成します。 詳細は、 p. 106 ディシジョン ルールのアルゴリズム を参照してください。
- R 内の各ルールを L_i に付加して、新しい候補モデルのセットを構築します。
- 拡張されたリストを T に保存します。

3. T からリスト モデルを選択します。

- T の各モデルの推定レスポンス確率 $\hat{P}_{L_i}$ は次のように計算されます。

$$\hat{P}_{L_i} = \frac{N(Y_n = 1, X_n \in X_{L_i})}{N(X_n \in X_{L_i})}$$

- w リストを T から選択します。ここでは、L^* として $\hat{P}_{L_i}$ が最も高くなります。

4. L^* を ResultSet に追加します。
5. d = 1 or $L^* = \varnothing$ の場合は、ResultSet を返して終了します。そうでない場合は d を 1 減らし、ステップ 2 を繰り返します。

ディシジョン ルールのアルゴリズム

各ルールは、ディシジョン ルール サイクル内で拡張されます。ディシジョン ルールにより、著しく出現頻度数が増えた対象値を探して、グループが検索されます。必要に応じて確率を上げたり下げたりして、ディシジョン ルールによりグループが検索されます。

表記

ディシジョン リストのアルゴリズムの説明に、次の表記が使用されます。

X	データ行列:列はフィールド（属性）、行はレコード（ケース）です。
R	先行するルール リストを拡張するルールの集まり
R_i	ルール集合 R 内の i 番目のルール
R_{all}	X 内のすべてのケースをカバーする特別ルール
$\hat{P}_{R_i}$	R_i の推定レスポンス確率
N	母集団の合計サイズ
$X_{m,n}$	X の n 番目のレコード（行）の m 番目のフィールド（列）の値
X_{R_i}	ルール リスト R_i にカバーされる X 内のレコードのサブセット
Y	X 内の対象フィールド
Y_n	n 番目のレコードの対象フィールドの値
A	X のすべての属性（フィールド）の集まり

A_j	X の j 番目の属性。[**属性の再使用の許可**] が偽の場合、A は先行ルールに存在する属性を除外します。
SplitRule(X, A_j)	A_j および X 内のレコードについてのルールを導くルール分割のアルゴリズム。 詳細は、 p. 108 ディシジョン ルールの分割アルゴリズム を参照してください。
T	候補リスト モデルのセット
ResultSet	ディシジョン リスト モデルの集まり

アルゴリズムのステップ

ディシジョン ルール アルゴリズムは次のように実行されます。

1. ルール セットを初期化します。

▶ d = 検索の深さ、w = 検索の幅、とします。

▶ R = ⌀ の場合 R_{all} を R に追加します。

▶ T = ⌀ とします。

2. R のすべてのルール R_i。

▶ ルール R_i にカバーされた X_{R_i} を選択します。

▶ 新しいセグメントの空のセット S を作成します。

▶ A 内の 属性A_j

■ 属性 A_j に基づいて新しいセグメントを生成します。

SplitRule(X_{R_i}, A_j)

■ セグメントを S に追加します。

▶ S 内の各セグメントで R_i を拡張して新しい候補のルールのセットを構築します。

▶ 拡張されたルールを T に保存します。S = ⌀ の場合、R_i を ResultSet に追加します。

3. T からルールを選択します。

▶ T の拡張されたルールを使用して、推定レスポンス確率 $\hat{P}_{R_i}$ を次のように計算します。

$$\hat{P}_{R_i} = \frac{N(Y_n = 1, X_n \in X_{R_i})}{N(X_n \in X_{R_i})}$$

▶ w ルールを選択します。ここでは、R^* として $\hat{P}_{R_i}$ が最も高くなります。

L^* を ResultSet に追加します。

▶ d = 1 の場合は ResultSet を返して終了します。そうでない場合は R = R^*, T = ⌀ を設定し、d を 1 つ減らしてステップ 2 を繰り返します。

ディシジョン ルールの分割アルゴリズム

ディシジョン ルールの分割アルゴリズムは、単一属性（フィールド）から高いレスポンス セグメントを生成するのに使用されます。レコードおよびそこからセグメントを生成する属性が提供される必要があります。このアルゴリズムはすべての通常の属性に適用可能であり、通常の属性は必ず順序付けられている値を持っている必要があります。このアルゴリズムによって生成されるセグメントは、n 次元のルールを（n1）次元に拡張するのに使用できます。このディシジョン ルール分割アルゴリズムは、海面方法と呼ばれることもあります。

表記

ディシジョン ルール分割アルゴリズムの説明に、次の表記が使用されます。

X	データ行列:列はフィールド（属性）、行はレコード（ケース）です。
C	分割する属性値 （カテゴリ） のソート済みリスト。値は昇順でソートされています。
C_i	カテゴリ C のリスト内の i 番目のカテゴリ。
$X_{n,c}$	X の n 番目のレコード（行）の分割フィールド（属性）の値
Y	X 内の対象フィールド
Y_n	n 番目のレコードの対象フィールドの値
N	母集団の合計サイズ
M	C 内のカテゴリ数
P_i	カテゴリ C_i の観察されたレスポンスの確率
$S_{L,R}$	カテゴリのセグメント、$S_{L,R} = \{C_i \vert C_i \in C, 1 \le L \le i \le R \le M\}$。
$\left(p^-_{S_{L,R}}, p^+_{S_{L,R}}\right)$	$S_{L,R}$ のレスポンス確率のための信頼区間（CI）。
$\max_p(C_i, C_j)$	$\{C_i, C_j\}$ の高いレスポンス確率を持つカテゴリ。
$\max_n(C_i, C_j)$	$\{C_i, C_j\}$ の大きなレコード数を持つカテゴリ。

アルゴリズムのステップ

ディシジョン ルールの分割アルゴリズムは次のように実行されます。

1. 各カテゴリ C_i の P_i を計算します。

$$P_i = \frac{N(Y_n = 1, X_{n,c} \in C_i)}{N(X_{n,c} \in C_i)}, P_0 = P_{(M+1)} = 0$$

$N(X_{n,c} \in C_i) = 0$ の場合、C_i はスキップされます。

2. P_i の極大値を検出してセグメント セットを作成します。

$$PeakSet = \{C_i | C_i \in C, 0 \leq i = I \leq M\}$$

ここで、I は次の条件を満たす正の整数です。

$$P_I > P_{(I-1)}$$

$$P_I = P_{(I+l)}, 0 \leq l \leq L - I$$

$$P_L > P_{(L+1)}$$

セグメント セットは、P_{S_i} に基づいた順序付けられたセグメントです。

$$SegmentSet = \left\{S_{L,R} | C_i \in PeakSet, L = R = i, P_{S_i} \geq P_{S_{i+1}}\right\}$$

3. SegmentSet 内のセグメントを選択します。

▶ SegmentSet が空の場合は、ResultSet を返して終了します。

▶ レスポンス確率 $P_{S_{L,R}}$ が最も高い $S_{L,R}$ を選択します。

▶ $R - L + 1 = M$ or $P_{S_{L,R}} \leq P_{S_{1,M}}$ の場合、 SegmentSet からセグメントを削除して別のセグメントを選択します。

4. セグメントを検証します。

▶ 次の条件が満たされるかどうかを検証します。

■ セグメントのサイズが最小セグメント サイズの基準を超えているか

$$Size\left(S_{L,R}\right) > Max(gs_{\min}, d, Max(g \cdot Size(parent)))$$

ここで、

$$parent \in ResultSet, L_{parent} \geq L, R_{parent} \leq R$$

■ セグメントのレスポンス確率は、重なり合わない複数の信頼区間で示されるように、標本全体より著しく高くなっています。

$$p^{-}_{S_{L,R}} > p^{+}_{Pop}$$

詳細は、 p. 110 信頼区間 を参照してください。

■ セグメントの拡張がレスポンス確率を下げるかどうか

$$P_{S_{L-1,R}} < P_{S_{L,R}} \text{ and } P_{S_{L,R+1}} < P_{S_{L,R}}$$

次にセグメント $S_{L,R}$ を ResultSet に追加し、親として $S_{L,R}$ を持ち、$Size\left(S'_{L,R}\right) \leq g \cdot Size(S_{L,R})$ となる $S'_{L,R}$ を ResultSet から削除します。

5. セグメントを拡張します。

▶ $C_{adjacent}$ を $S_{L,R}$ に追加します。ここで、

$$C_{adjacent} = \begin{cases} Max_p(C_{L-1}, C_{R+1}) & \text{if } P_{L-1} \neq P_{R+1} \\ Max_n(C_{L-1}, C_{R+1}) & \text{if } P_{L-1} = P_{R+1} \text{ and } N\left(C_{L-1}\right) \neq N(C_{R+1}) \\ C_{R+1} & \text{otherwise} \end{cases}$$

▶ R または L を適宜調節します。つまり、$C_{adjacent} = C_{L-1}$ の場合、$L = L - 1$ を設定し、$C_{adjacent} = C_{R+1}$ の場合、$R = R + 1$ を設定します。

▶ $S_{L,R}$ を SegmentSet に返し、ステップ 3 から繰り返します。

信頼区間

確信度の限度 (p^-, p^+) for $\hat{p}$ は次のように計算されます。

$$p^- = \begin{cases} \frac{x}{x+(n-x+1)F_{2(n-x+1),2x;1-\alpha/2}} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

$$p^+ = \begin{cases} \frac{(x+1)F_{2(x+1),2(n-x);1-\alpha/2}}{n-x+(x+1)F_{2(x+1),2(n-x);1-\alpha/2}} & \text{if } x \neq n \\ 1 & \text{if } x = n \end{cases}$$

ここで、n はルールまたはリストの範囲、x はルールまたはリストのレスポンスの頻度、α は希望する信頼区間であり、$F_{a,b;c}$ は、自由度 a および b の F用の逆数累積分布関数です。パーセンタイルは $100c$ となります。

二次指標

各セグメントについて、次の指標が報告されます。

範囲: セグメント内のレコードの数、$N(S)$。

度数: レスポンスが真の、セグメント内のレコードの数、$N(Y_n = 1, X_n \in S)$。

確率: レスポンスが真の、セグメント内のレコードの数、$\frac{N(Y_n=1, X_n \in S)}{N(S)}$、または $\frac{\text{Frequency}}{\text{Coverage}}$。

空白の処理

ディシジョン リスト モデルの入力フィールドの空白値は、エキスパート モデルのオプションに応じて、セグメントを定義するのに使用できる独立したカテゴリとして取り扱うことも、モデルから除外することもできます。デフォルトは、セグメントを定義するカテゴリとして空白を使用することです。対象フィールドのための空白値のあるレコードは、モデルの構築から除外されます。

生成されたモデル/スコアリング

ディシジョン リスト生成モデルは、セグメントのセットから構成されます。新しいデータのスコアリング時に、各モデルは各セグメント内の所属メンバーから順番に評価されます。予測フィールドに基づいてレコードを記述するモデル順で最初のセグメントが、レコードを申請し、予測値と確率を決定します。予測値がレスポンス値でないレコードは、値が $null となります。確率は、前述の で説明されている通りに計算されます。

空白の処理

ディシジョン リスト生成モデルでデータをスコアリングする際に、空白はセグメント定義に有効な値と見なされます。エキスパート オプションの **[条件内に欠損値を許可]** を無効にしてモデルが作成された場合、入力フィールドの 1 つが欠損値のレコードは、定義でそのフィールドに依存するどのセグメントにも一致しません。

章
14

DISCRIMINANT アルゴリズム

空でないグループの数が 2 より少ないか、ケース数または重みの合計が空でないグループの数を超えない場合のサブファイル グループに対しては、何の分析も行われません。変数の選択中に何の変数も選択されなかったり固有分析が失敗したりする場合は、分析が停止されることがあります。

表記

この章では特に明記しない限り、次の表記を使用します。

g	グループの数
p	変数の数
q	選択された変数の数
X_{ijk}	グループ j 内のケース k に対する変数 i の値
f_{jk}	グループ j 内のケース k に対するケースの重み
m_j	グループ j 内のケースの数
n_j	グループ j 内のケース重みの合計
n	重みの合計

基本統計量

プロシージャにより、次の基本統計量が計算されます。

平均

$$\overline{X}_{ij} = \left(\sum_{k=1}^{m_j} f_{jk} X_{ijk} \right) / n_j \qquad \text{(variable } i \text{ in group } j\text{)}$$

$$\overline{X}_{i\bullet} = \left(\sum_{j=1}^{g} \sum_{k=1}^{m_j} f_{jk} X_{ijk} \right) / n \qquad \text{(variable } i\text{)}$$

分散

$$S_{ij}^2 = \frac{\left(\sum_{k=1}^{m_j} f_{jk} X_{ijk}^2 - n_j \overline{X}_{ij}^2\right)}{(n_j - 1)} \qquad \text{(variable } i \text{ in group } j\text{)}$$

$$S_{i\bullet}^2 = \frac{\left(\sum_{j=1}^{g}\sum_{k=1}^{m_j} f_{jk} X_{ijk}^2 - n \overline{X}_i^2\right)}{(n-1)} \qquad \text{(variable } i\text{)}$$

2 乗およびクロス乗積行列のグループ内合計 (W)

$$w_{il} = \sum_{j=1}^{g}\sum_{k=1}^{m_j} f_{jk} X_{ijk} X_{ljk} - \sum_{j=1}^{g}\left(\sum_{k=1}^{m_j} f_{jk} X_{ijk}\right)\left(\sum_{k=1}^{m_j} f_{jk} X_{ljk}\right)/n_j \quad i, l = 1, \ldots, p$$

2 乗およびクロス乗積行列の合計 (W)

$$t_{il} = \sum_{j=1}^{g}\sum_{k=1}^{m_j} f_{jk} X_{ijk} X_{ljk} - \left(\sum_{j=1}^{g}\sum_{k=1}^{m_j} f_{jk} X_{ijk}\right)\left(\sum_{j=1}^{g}\sum_{k=1}^{m_j} f_{jk} X_{ljk}\right)/n$$

グループ内の共分散行列

$$\mathbf{C} = \frac{\mathbf{W}}{(n-g)} \quad n > g$$

各グループの共分散行列

$$c_{il}^{(j)} = \frac{\left(\sum_{k=1}^{m_j} f_{jk} X_{ijk} X_{ljk} - \overline{X}_{ij}\overline{X}_{lj} n_j\right)}{(n_j - 1)}$$

グループ内の相関行列 (R)

$$r_{il} = \begin{cases} \frac{w_{il}}{\sqrt{w_{ii} w_{ll}}} & \text{if } w_{ii} w_{ll} > 0 \\ \text{SYSMIS} & \text{otherwise} \end{cases}$$

総共分散行列

$$\mathbf{T}' = \frac{\mathbf{T}}{n-1}$$

1 変量 F および Λ 変数 I のため

$$F_i = \frac{(t_{ii} - w_{ii})(n-g)}{w_{ii}(g-1)}$$

これには g−1 および n−g の自由度付きです。

$$\Lambda_i = \frac{w_{ii}}{t_{ii}}$$

これには 1、g-1 および n-g の自由度付きです。

変数選択のルール

直接およびステップワイズの両方の変数投入が可能です。複数の包含レベルも指定することができます。

Method = Direct

直接変数選択の場合、変数は上流ノードから渡される順に包含されると見なされます。変数が分析に含まれている場合は、指定された許容限度（デフォルトは 0.001）よりも低い許容レベルの変数は分析中にありません。

ステップワイズ変数選択

各ステップで、次のルールにより変数選択が制御されます。

- 低い包含レベルの変数より前に、高い包含レベルの的確な変数が投入されます。
- 同じ包含レベルの的確な変数の投入順序は、上流のノード内で決定されます。
- 同じ包含奇数レベルの的確な変数の投入順序は、投入基準の値で決定されます。基準統計量に「最適な」値を持つ変数が最初に投入されます。
- レベル 1 の処理に達すると、すべての的確な変数の包含に先立って、レベル 1 包含番号のすでに投入された変数が、削除されるかどうか検査されます。変数の削除する F が変数削除の F 値より小さい場合、その変数は削除に適していると見なされます。または、確率基準が使用される場合、削除する F の有意性が指定された確率レベルを超えた場合も、その変数は削除に適していると見なされます。複数の変数が削除に適している場合は、残りの変数の統計基準に「最適」な値を残してその変数が削除されます。変数の削除は、削除に適した変数がなくなるまで続きます。その後変数の順次投入は前に説明したように、進行します。ただし各ステップの後、1 の包含番号付きの変数が、前に述べたように排除の対象になるかが検討されます。
- 何らかの統計量が表示されることがあっても、包含レベルが 0 の変数が投入されることはありません。

除外ための不適格性

奇数の包含番号を持つ変数は、次の場合に包含に不適格であると見なされます。

- 分析中の任意の変数（それ自体も含めて）の許容度が指定された許容限度を下回ったとき、または
- その投入 F が値を投入する変数の F 値より小さいとき、または
- 確率基準が使用される場合、投入 F に関連付けられた有意水準が入力する確率を超えているとき。

偶数の包含番号を持つ変数は、上の最初の条件が満たされる場合、包含には不適格です。

変数選択中の計算

変数選択中に、行列 W は、1969 年 12 月に公表された対称スイープ操作を使用して、新規行列 $\mathbf{W}^*$ でステップごとに置き換えられます。最初の q 個の変数が分析に含まれた場合、W は次のようにデータ区分されます。

$$\begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix}$$

ここで、$\mathbf{W}_{11}$ は q×q となります。この段階で、行列 $\mathbf{W}^*$ は、次のように定義されます。

$$\mathbf{W}^* = \begin{bmatrix} -\mathbf{W}_{11}^{-1} & \mathbf{W}_{11}^{-1}\mathbf{W}_{12} \\ \mathbf{W}_{21}\mathbf{W}_{11}^{-1} & \mathbf{W}_{22} - \mathbf{W}_{21}\mathbf{W}_{11}^{-1}\mathbf{W}_{12} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{11}^* & \mathbf{W}_{12}^* \\ \mathbf{W}_{21}^* & \mathbf{W}_{22}^* \end{bmatrix}$$

さらに、ステップワイズ変数選択が使用される場合は、T が行列 T^* に置き換えられ、同様に定義されます。

次の統計量が計算されます。

許容度

$$\text{TOL}_i = \begin{cases} 0 & \text{if } w_{ii} = 0 \\ w_{ii}^*/w_{ii} & \text{if variable } i \text{ is not in the analysis and } w_{ii} \neq 0 \\ -1/(w_{ii}^* w_{ii}) & \text{if variable } i \text{ is in the analysis and } w_{ii} \neq 0. \end{cases}$$

変数の許容度が指定された許容限度以下であったり、分析内の包含が別の変数の許容度を下限まで減らすことになる場合は、以下の統計量がその変数またはそれを含むセットに対して計算されません。

F-to-Remove

$$F_i = \frac{(w_{ii}^* - t_{ii}^*)(n-q-g+1)}{t_{ii}^*(g-1)}$$

これには g-1 および n-q-g+1 の自由度付きです。

F-to-Enter

$$F_i = \frac{(t_{ii}^* - w_{ii}^*)(n-q-g)}{w_{ii}^*(g-1)}$$

これには g-1 および n-q-g の自由度付きです。

グループ平均の同等性検定のための Wilk のラムダ

$$\Lambda = |\mathbf{W}_{11}|/|\mathbf{T}_{11}|$$

これには q, g-1 および n-g の自由度付きです。

Rao の R として知られるラムダの近似 F 検定 (1971 年、タツオカ)

$$F = \frac{(1-\Lambda^s)(r/s+1-qh/2)}{\Lambda^s qh}$$

ここで、

$$s = \begin{cases} \sqrt{\frac{q^2+h^2-5}{q^2h^2-4}} & \text{if } q^2+h^2 \neq 5 \\ 1 & \text{otherwise} \end{cases}$$
$$r = n-1-(q+g)/2$$
$$h = g-1$$

自由度は qh および r/s+1-qh/2 となります。近似は、q または h が 1 または 2 の場合正確になります。

Rao の V (Lawley-Hotelling トレース) (Rao, 1952; Morrison, 1976)

$$V = -(n-g)\sum_{i=1}^{q}\sum_{l=1}^{q} w_{il}^{*}(t_{il}-w_{il})$$

n-g が大きい場合、帰無仮説のもと V は自由度q(g-1) の χ^2 として近似的に分布します。追加の変数が入力された場合、V の変更が正であれば、自由度 g-1 の χ^2 の近似的な分布となります。

グループ a および b 間の共有マハラノビスの距離

$$D_{ab}^2 = -(n-g)\sum_{i=1}^{q}\sum_{l=1}^{q} w_{il}^{*}\left(\overline{X}_{ia}-\overline{X}_{ib}\right)\left(\overline{X}_{la}-\overline{X}_{lb}\right)$$

グループ a および b の平均の同等性を検定する F 値(最小 F 比率)

$$F_{ab} = \frac{(n-q-g+1)n_a n_b}{q(n-g)(n_a+n_b)} D_{ab}^2$$

不明な変動の合計 (Dixon, 1973)

$$R = \sum_{a=1}^{g-1}\sum_{b=a+1}^{g} 4/\left(4+D_{ab}^2\right)$$

分類関数

q 個の変数のセットが選択されると、分類関数 (Fisher の線型判別分析関数とも呼ばれる) が、各係数に対して次を使用して計算できます。

$$b_{ij} = (n-g)\sum_{l=1}^{q} w_{il}^{*}\overline{X}_{lj} \quad i = 1, 2, \ldots, q, j = 1, 2, \ldots, g$$

および、定数に関しては

$$a_j = \log p_j - \tfrac{1}{2}\sum_{i=1}^{q} b_{ij}\overline{X}_{ij} \quad j = 1, 2, \ldots, q$$

ここで、p_j はグループ j の前の確率です。

標準判別分析関数

標準分析判別関数の係数は、一般的な固有値問題を解決することで決定されます。

$$(\mathbf{T} - \mathbf{W})\mathbf{V} = \lambda\mathbf{W}\mathbf{V}$$

ここで、V は判別分析関数係数の計測されない行列であり、λ は固有値の対角線行列です。固有システムは次のようにして解決されます。

Cholesky の分解

$$\mathbf{W} = \mathbf{L}\mathbf{U}$$

が形成されます。L は下三角行列であり、$\mathbf{U} = \mathbf{L}'$ となります。

対象行列 $\mathbf{L}^{-1}\mathbf{B}\mathbf{U}^{-1}$ が形成され、システムは次のようになります。

$$\left(\mathbf{L}^{-1}(\mathbf{T} - \mathbf{W})\mathbf{U}^{-1} - \lambda\mathbf{I}\right)(\mathbf{U}\mathbf{V}) = 0$$

が三角対角化および QL 方法を使用して解決されます。結果は m 固有値で、$m = \min(q, g-1)$ となり、 対応する正規直行型固有値 UV です。元のシステムの固有値は、次のようにして取得されます。

$$\mathbf{V} = \mathbf{U}^{-1}(\mathbf{U}\mathbf{V})$$

重要度の高い順に並べられた固有値のそれぞれに対して、次の統計量が計算されます。

グループ間変数の割合

$$\frac{100\lambda_k}{\sum_{k=1}^{m}\lambda_k}$$

正準相関

$$\sqrt{\lambda_k/(1+\lambda_k)}$$

Wilks のラムダ

最初の k の後のすべての判別分析関数の有意性の検定:

$$\Lambda_k = \prod_{i=k+1}^{m} 1/(1+\lambda_i) \quad k = 0, 1, \ldots, m-1$$

有意性のレベルは次に基づいています。

$\chi^2 = -(n-(q+g)/2-1)\ln\Lambda_k$

これは、自由度（q-k)(g-k-1）の χ^2 として分布します。

標準正準判別係数マトリックス D

標準化された標準判別分析係数行列 D は、次のようにして計算されます。

$\mathbf{D} = \mathbf{S}_{11}\mathbf{V}$

ここで、

S=diag$(\sqrt{w_{11}}, \sqrt{w_{22}}, \ldots, \sqrt{w_{pp}})$

$\mathbf{S}_{11}$ は、最初の q 行と S 列を含むデータ区分です。

V ,Í $\mathbf{V}'\mathbf{W}_{11}\mathbf{V}$=I のような固有ベクトルの行列です。

標準判別分析関数と変数の判別分析との間の相関

標準判別分析関数と変数の判別分析との間の相関は、次によって定められます。

$\mathbf{R} = \mathbf{S}_{11}^{-11}\mathbf{W}_{11}\mathbf{V}$

変数の中には分析への包含に選択されないものもあり（q<p)、選択されなかった変数を相関行列に含めるために、固有ベクトルは暗にゼロで拡張されます。この計算では、$W_{ii}=0$ は S および W から除外されます。 その後 p はグループ内分散のゼロ以外の値の変数の数を表します。

標準化されていない係数

標準化されない係数は、標準化された係数から次を使用して計算されます。

$\mathbf{B} = \sqrt{(n-g)}\mathbf{S}_{11}^{-1}\mathbf{D}$

関連付けられた定数は次のとおりです。

$a_k = -\sum_{i=1}^{q} b_{ik}\overline{X}_{i\bullet}$

グループの重心は、次のグループ平均で評価される標準的判別分析関数です。

$\overline{f}_{kj} = a_k + \sum_{i=1}^{q} b_{ik}\overline{X}_{ij}$

変数の同等性の検定

Box の M は、グループ共分散行列の同質性の検査に使用されます。

$$M = (n-g)\log\left|\mathbf{C}'\right| - \sum_{j=1}^{g}(n_j-1)\log\left|\mathbf{C}^{(j)}\right|$$

ここで、

$\mathbf{C}'$ は、異質の共分散行列でグループを排除する、プールされたグループ内共分散行列です。

$\mathbf{C}^{(j)}$ はグループ j 番目のグループのサンプル共分散行列

$\mathbf{C}'$ および $\mathbf{C}^{(j)}$ の行列式は、Cholesky の分解から得られます。分解の対角要素が 10^{-11} より少ない場合、行列式は異質と見なされ、分析から除外されます。

$$\log\left|\mathbf{C}^{(j)}\right| = 2\sum_{i=1}^{p}\log l_{ii} - p\log(n_j-1)$$

ここで、l_{ii} は、$(n_j-1)\mathbf{C}^{(j)} = \mathbf{L}'\mathbf{L}$ のような i 番目の $\mathbf{L}$ の対角入力となります。同様に、

$$\log\left|\mathbf{C}'\right| = 2\sum_{i=1}^{p}\log l_{ii} - p\log\left(n' - g\right)$$

ここで、

$$\left(n' - g\right)\mathbf{C}' = \mathbf{L}'\mathbf{L}$$

n' は、異質でない共分散行列によるすべてのグループ内のケースの重み合計です。

有意水準は、次（Cooley and Lohnes, 1971）を使用した、自由度 t_1 および t_2 の F 分布から得られます。

$$F = \begin{cases} M/b & \text{if } e_2 > e_1^2 \\ \frac{t_2 M}{t_1(b-M)} & \text{if } e_2 < e_1^2 \end{cases}$$

ここで、

$$e_1 = \left(\sum_{j=1}^{g} \frac{1}{n_j - 1} - \frac{1}{n - g} \right) \frac{2p^2+3p-1}{6(g-1)(p+1)}$$

$$e_2 = \left(\sum_{j=1}^{g} \frac{1}{(n_j - 1)^2} - \frac{1}{(n - g)^2} \right) \frac{(p-1)(p+2)}{6(g-1)}$$

$$t_1 = (g-1)p(p+1)/2$$

$$t_2 = (t_1 + 2)/\left|e_2 - e_1^2\right|$$

$$b = \begin{cases} \frac{t_1}{1-e_1-t_1/t_2} & \text{if } e_2 > e_1^2 \\ \frac{t_2}{1-e_1-2/t_2} & \text{if } e_2 < e_1^2 \end{cases}$$

$e_1^2 - e_2$ がゼロの場合、または e_2 より大幅に小さい場合、t_2 を計算できない、あるいは計算できたとしても正確ではない場合があります。次のような場合

$$e_2 = e_2 + 0.0001\left(e_2 - e_1^2\right)$$

プログラムでは、次のように F 統計よりも Bartlett の χ^2 統計を使用します。

$$\chi^2 = M(1 - e_1)$$

自由度は t_1 です。

標準的判別分析関数のグループ共分散行列の検定の場合、プロシージャは同じです。共分散行列 $\mathbf{C}'$ および $\mathbf{C}^{(j)}$ は $\mathbf{D}_j$ および $\mathbf{D}'$ に置き換えられます。

$$\mathbf{D}_j = \mathbf{B}'\mathbf{C}^{(j)}\mathbf{B}$$

は、判別分析関数のグループ共分散行列です。

このケースにプールされた共分散行列は恒等式なので、次のようになります。

$$\mathbf{D}' = (n-g)\mathbf{I}_m - \sum_j (n_j - 1)\mathbf{D}_j$$

ここで、要約は単一グループ $\mathbf{D}_j$ に対してのみです。

空白の処理

入力フィールドまたは出力フィールドに欠損値があるすべてのレコードは、モデルの推定から除外されます。

生成されたモデル/スコアリング

ケースを分類するための基本的な手続きは、次のとおりです。

- X がケースのために判別している変数の 1×q ベクトルの場合、標準的判別分析関数値の 1×m ベクトルは、次のとおりです。

$\mathbf{f} = \mathbf{XB} + \mathbf{a}$

- 各重心からのカイ 2 乗距離が計算されます。

  $\chi_j^2 = (\mathbf{f} - \bar{\mathbf{f}}_j)\mathbf{D}_j^{-1}(\mathbf{f} - \bar{\mathbf{f}}_j)'$

  ここで、$\mathbf{D}_j$ はグループ j の判別分析関数のグループ共分散行列であり、$\bar{\mathbf{f}}_j$ はグループ重心ベクトルです。ケースがグループ j のメンバーの場合、χ_j^2 は自由度 m の χ^2 分布となります。出力で P(D>d|G=g) とラベルづけされた P(X|G) は、χ_j^2 のような有意確率です。
- 分類または事後確率は、次のとおりです。

  $$P(\mathbf{G}_j|\mathbf{X}) = \frac{P_j|\mathbf{D}_j|^{-1/2}e^{-\chi_j^2/2}}{\sum_{j=1}^{g} P_j|\mathbf{D}_j|^{-1/2}e^{-\chi_j^2/2}}$$

  ここで、p_j はグループ j の前の確率です。. ケースは、$P(\mathbf{G}_j|\mathbf{X})$ が最も高いグループに分類されます。

$P(\mathbf{G}_j|\mathbf{X})$ の実際の計算は、次のとおりです。

$$g_j = \log P_j - \tfrac{1}{2}\left(\log|\mathbf{D}_j| + \chi_j^2\right)$$

$$P(\mathbf{G}_j|\mathbf{X}) = \begin{cases} \dfrac{\exp(g_j - \max_j g_j)}{\sum_{j=1}^{g} \exp\left(g_j - \max_j g_j\right)} & \text{if } g_j - \max_j g_j > -46 \\ 0 & \text{otherwise} \end{cases}$$

個々のグループ共分散が分類で使用されない場合、判別分析関数のプールされたグループ内共分散行列（恒等式行列）は、上記の計算中に $\mathbf{D}_j$ に置き換えられ、注目に値する単純さが生まれます。

$\mathbf{D}_j$ のいずれかが単数の場合、疑似逆数の形式

$\begin{bmatrix} \mathbf{D}_{j11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}$ が、

$\mathbf{D}_j^{-1}$ を置き換え、$|\mathbf{D}_{j11}|$ が $|\mathbf{D}_j|$ を置き換えます。 $\mathbf{D}_{j11}$ は $\mathbf{D}_j$ の副行列であり、その行と列は先行する関数には依存しない関数に対応します。つまり、関数 1 は $\mathbf{D}_j = 0$ のランクの場合にのみ除外され、関数 2 は関数 1 に依存している場合にのみ除外される、という順になっていきます。疑似逆数のこの選択は、$\mathbf{D}_{j11}^{-1}$ の数値的安定性に対しては最適ではありませんが、残りの関数の判別分析力を最大限にします。

交差検証(リーブワンアウト分類)

次の表記がこのセクションで使用されます。

$\underset{\sim}{X_{jk}}$	$(X_{1jk},\ldots,X_{qjk})^T$
$\underset{\sim}{M_j}$	j 番目のグループのサンプル平均
	$\underset{\sim}{M_j}=\frac{1}{n_j}\sum_{k=1}^{m_j} f_{jk}\underset{\sim}{X_{jk}}$
$\underset{\sim}{M_{jk}}$	j 番目のグループ除外ポイント $\underset{\sim}{X_{jk}}$ のサンプル平均
	$\underset{\sim}{M_{jk}}=\frac{1}{n_j-f_{jk}}\sum_{\substack{l=1\\ l\neq k}}^{m_j} f_{jl}\underset{\sim}{X_{jl}}$
Σ	調査対象のサンプル共分散行列
Σ_j	j 番目のグループのサンプル共分散行列
Σ_{jk}	ポイント $\underset{\sim}{X_{jk}}$ のないポールされたサンプル共分散行列
Σ_{jk}^{-1}	$=\frac{n-g-f_{jk}}{n-g}\left(\Sigma^{-1}+\frac{n_j\Sigma_j^{-1}\left(\underset{\sim}{X_{jk}}-\underset{\sim}{M_j}\right)\left(\underset{\sim}{X_{jk}}-\underset{\sim}{M_j}\right)^T\Sigma_j^{-1}}{(n_j-f_{jk})(n_j-g)-n_j\left(\underset{\sim}{X_{jk}}-\underset{\sim}{M_j}\right)^T\Sigma_j^{-1}\left(\underset{\sim}{X_{jk}}-\underset{\sim}{M_j}\right)}\right)$
$d_0^2\left(\underset{\sim}{a},\underset{\sim}{b}\right)$	$\left(\underset{\sim}{a}-\underset{\sim}{b}\right)^T\Sigma_{jk}^{-1}\left(\underset{\sim}{a}-\underset{\sim}{b}\right)^T$

交差検証は、線型判別分析にのみ適用されます（二次式分析ではなく）。交差検証中に、データ セット内のすべてのケースが繰り返し検証されます。各ケース、たとえば $\underset{\sim}{X_{jk}}$ は、1 回抽出され、検定データとして取り扱われます。残りのケースは新しいデータ セットとして取り扱われます。

ここでは、$d_0^2\left(\underset{\sim}{X_{jk}},\underset{\sim}{M_{jk}}\right)$ および $d_0^2\left(\underset{\sim}{X_{jk}},\underset{\sim}{M_i}\right)(i=1,...,g.i\neq j)$ を計算します。i$(i\neq j)$ があり、$\left(\log(P_i)-d_0^2\left(\underset{\sim}{X_{jk}},\underset{\sim}{M_i}\right)/2>\log(P_j)-d_0^2\left(\underset{\sim}{X_{jk}},\underset{\sim}{M_{jk}}\right)/2\right)$ を満たす場合、抽出ポイント $\underset{\sim}{X_{jk}}$ が誤分類されます。予測誤差率の推定値は、誤分類されたケースの重みの合計とすべてのケースの重みの合計の比率です。

計算時間を減らすために、標準的判別分類方法の代わりに線型判別分類方法が使用されます。理論的ソリューションは、両方の方法でまったく同じです。

空白の処理（判別分析アルゴリズムのスコアリング）

最終モデルにおいて、入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールド $null$ が割り当てられます。

参照

Anderson, T. W. 1958. Introduction to multivariate statistical analysis. New York: John Wiley & Sons, Inc..

Cooley, W. W., および P. R. Lohnes. 1971. Multivariate data analysis. New York: John Wiley & Sons, Inc..

Dempster, A. P. 1969. Elements of Continuous Multivariate Analysis. Reading, MA: Addison-Wesley.

Dixon, W. J. 1973. BMD Biomedical computer programs. Los Angeles: University of California Press.

Tatsuoka, M. M. 1971. Multivariate analysis. New York: John Wiley & Sons, Inc..

章
15

アンサンブル アルゴリズム

アンサンブルを使用して、モデルの制度を拡張（ブースティング）、モデルの安定性を拡張（バギング）、そして非常に大きいデータセットのモデルを構築（パス、ストリーム、結合）します。

- 詳細は、 p. 130 パス、ストリーム、結合（PSM）アルゴリズム を参照してください。
- 詳細は、 p. 124 バギング アルゴリズムとブースティング アルゴリズム を参照してください。

バギング アルゴリズムとブースティング アルゴリズム

ブートストラップ集計（バギング）とブースティングは、モデルの安定性および制度の向上に使用されるアルゴリズムです。バギングは不安定な基本モデルに有効で、予測値の分散を少なくすることができます。ブースティングはどんな種類のモデルにも使用でき、予測値の分散およびバイアスを少なくすることができます。

表記

この章では特に明記しない限り、バギングとブースティングに次の表記を使用します。

K	学習セット内のレコードの数。
X_k	k 番目のレコードの予測値。
y_k	k 番目のレコードの対象値。
f_k	k 番目のレコードの度数の重み。
w_k	k 番目のレコードの分析の重み。
N	合計レコード数（$N = \Sigma_{k=1}^{K} f_k$）。
M	構築する基本モデルの数。バギングの場合、ブートストラップ サンプルの数。
$T^m(\cdot)$	m 番目のサンプルに構築されたモデル。
f_k^m	m 番目のブートストラップ サンプルの k 番目のレコードのシミュレートされた度数の重み。
w_k^m	m 番目のブートストラップ サンプルの k 番目のレコードの更新された分析の重み。
$\hat{y}_k^m = T^m(X_k)$	m 番目のモデルによって予測された k 番目のレコードの対象値。
$P_{l_i}^m(X_k)$	カテゴリ型対象の場合、モデル m の k 番目のレコードがカテゴリ l_i, i=1, ..., C に属する確率。
$II(\pi)$	条件 π について、π がホールドする場合、$II(\pi)$ が 1 となり、ホールドしない場合は 0 となります。

ブートストラップ集計

ブートストラップ集計（バギング）では、元のデータセットから置換してサンプリングすることによって、学習データセットの複製を作成します。これにより、元のデータセットとサイズが同じブートストラップ サンプルが作成されます。k=1,..,K および m=1,...,M でアルゴリズムがインタラクティブに実行され、次のような度数の重みを生成します。

$$f_{mk}^{*}=\begin{cases} rv.binom\left(N,\frac{f_k}{N}\right) & k=1 \\ rv.binom\left(N-\Sigma_{i=1}^{k-1}f_{mi}^{*},\frac{f_k}{N-\Sigma_{i=1}^{k-1}f_i}\right) & \text{otherwise} \end{cases}$$

モデルが繰り返しごとに構築されます。同時にこれらのモデルがアンサンブル モデルを形成します。アンサンブル モデルは、次のいずれかの方法を使用して新規レコードをスコアリングします。使用できる方法は対象の測定レベルによって異なります。

連続型対象のスコアリング

- 平均

  $\hat{y}_k=\frac{1}{M}\sum_{m=1}^{M}\hat{y}_k^m$

- 中央値

  並べ替えを行い（$\hat{y}_k^m$）再びラベルを付けます（$\hat{y}_{(1)}\le ...\le \hat{y}_{(M)}$）。

  $$\hat{y}_k=\begin{Bmatrix} \hat{y}_{\left(\frac{M+1}{2}\right)} & (\ M\ \text{が奇数の場合}) \\ \frac{1}{2}\left(\hat{y}_{\left(\frac{M}{2}\right)}+\hat{y}_{\left(\frac{M}{2}+1\right)}\right) & (\ M\ \text{が偶数の場合}) \end{Bmatrix}$$

カテゴリ型対象のスコアリング

- 票決

  $\hat{y}_k=arg\max_{l_i\in\Omega}\frac{1}{\left|M_{l_i}\right|}\sum_{m\in M_{l_i}}P_{l_i}^{m}\left(X_k\right)$

  $\hat{p}_{\hat{y}_k}=\frac{1}{\left|M_{\hat{y}_k}\right|}\sum_{m\in M_{\hat{y}_k}}P_{\hat{y}_k}^{m}\left(X_k\right)$

  この場合 $\Omega=\{arg\max_{l_i}|M_{l_i}|\}$

- 最も高い確率

  $\hat{y}_k=arg\max_{l_i}\left(\max_m\left(P_{l_i}^{m}\left(X_k\right)\right)\right)$

  $\hat{p}_{\hat{y}_k}=\max_m\left(P_{\hat{y}_k}^{m}\left(X_k\right)\right)$

- 最も高い平均値の確率

  $\hat{y}_k=arg\max_{l_i}\frac{1}{M}\sum_{m=1}^{M}P_{l_i}^{m}\left(X_k\right)$

$$\hat{p}_{\hat{y}_k} = \frac{1}{M}\sum_{m=1}^{M} P_{\hat{y}_k}^{m}\left(X_k\right)$$

モデル指標のバギング

精度

精度は、Naive モデル、リファレンス（シンプル）モデル、アンサンブル モデル（各アンサンブル方法と関連)、および基本モデルについて計算されます。

カテゴリ型対象の場合、分類の精度は次のようになります。

$$\frac{1}{N}\sum_{k=1}^{K} f_k II\left(y_k == \hat{y}_k\right)$$

連続型対象の場合は、次のとおりです。

$$R^2 = 1 - \frac{\Sigma_{k=1}^{K} f_k (y_k - \hat{y}_k)^2}{\Sigma_{k=1}^{K} f_k (y_k - \overline{y})^2}$$

この場合 $\overline{y} = \frac{1}{N}\Sigma_{k=1}^{K} f_k y_k$

R^2 は 1 より大きくなることはありませんが、0 より小さくなる場合があります。

Naïve モデルの場合、$\hat{y}_k$ はカテゴリ型対象ではモーダルなカテゴリ、連続型対象では平均値となります。

多様性

多様性は 0 ～ 1 の範囲の測定で、数値が大きいほどより多様になります。基本モデルで予測値がどれほど異なるかを示します。

カテゴリ型対象の場合、多様性は次のようになります。

$$\frac{1}{N \cdot M^2}\sum_{k=1}^{K} f_k L\left(y_k\right)\left[M - L\left(y_k\right)\right]$$

この場合 $L\left(y_k\right) = \sum_{m=1}^{M} II\left(y_k = \hat{y}_k^m\right)$ となります。

連続型対象の場合、多様性は次のようになります。

$$\mathsf{D} = \frac{\sum_{k=1}^{K} f_k \left[\frac{1}{M\left(M-1\right)}\sum_{m=1}^{M}\sum_{n=1,n\neq m}^{M}\left(y_k - \hat{y}_k^m\right)\left(\hat{y}_k^n - y_k\right)\right]}{\Sigma_{k=1}^{K} f_k (y_k - \overline{y}_k)^2}$$

Adaptive Boosting

Adaptive Boosting (AdaBoost) は連続型対象でモデルをブースティングする場合に使用するアルゴリズムです (Freund and Schapire 1996, Drucker 1997)。

1. 値を初期化します。

$w_k = \begin{cases} \frac{w_k}{\Sigma_{i=1}^{K} w_i f_i} & \text{(分析の重みが指定されている場合)} \\ 1/N & \text{(指定されていない場合)} \end{cases}$ を設定します。

m=1、$w_k^m = w_k$、および $f_k^m = f_k$ を設定します。基本モデルの構築に使用する方法が分析の重みをサポートしない場合でも、分析の重みが初期化されます。

2. 学習セットを使用して、基本モデル m を構築し、学習セットをスコアリングします。$T^m(\cdot)$

基本モデル m のモデルの重みを設定します。$\omega^m = \log\left(\frac{1-\sum_{k=1}^{K} L_k w_k^m f_k}{\sum_{k=1}^{K} L_k w_k^m f_k}\right)$

この場合、$L_k = \frac{abs(\hat{y}_k^m - y_k)}{\max_k\left(abs(\hat{y}_k^m - y_k)\right)}$ となります。

3. 次の基本モデルの重みを設定します。

$w_k^{m+1} = \frac{a_k^{m+1}}{\Sigma_{i=1}^{K} a_i^{m+1} f_i}$

この場合 $a_k^{m+1} = w_k^m \left(\frac{\sum_{k=1}^{K} L_k w_k^m f_k}{1-\sum_{k=1}^{K} L_k w_k^m f_k}\right)^{1-L_k}$ となります。分析の重みは常に更新されますので注意してください。基本モデルの構築に使用される方法で分析の重みがサポートされていない場合、次の基本モデルの度数の重みが次のように更新されます。

$f_k^{m+1} = \begin{cases} rv.binom\left(N, w_k^{m+1} f_k\right) & k = 1 \\ rv.binom\left(N - \Sigma_{i=1}^{k-1} f_k^{m+1}, \frac{w_k^{m+1} f_k}{1 - \Sigma_{i=1}^{k-1} w_k^{m+1} f_i}\right) & \text{otherwise} \end{cases}$

m<M の場合、m=m+1 を設定して ステップ 2 に進みます。そうでない場合、アンサンブル モデルが完了します。

注 : $\sum_{k=1}^{K} L_k w_k^m f_k \geq 0.5$ または $\max_k\left(abs(\hat{y}_k^m - y_k)\right)$ の場合、基本モデルはアンサンブルから除外されます。

スコアリング

AdaBoost では重み付き中央値の方法を使用してアンサンブル モデルをスコアリングします。

並べ替えを行い（$\hat{y}_k^m$）再びラベルを付けます（$\hat{y}_{(1)} \le ... \le \hat{y}_{(M)}$）。 モデルの重みのアソシエーションを保持し（$\omega^m$）再びラベルを付けます（$\omega_{(1)}, ..., \omega_{(M)}$）。

アンサンブル予測値は $\hat{y}_k = \hat{y}_{(i)}$ となります。この場合、i は $\sum_{m=1}^{i-1} \omega^m < \frac{1}{2}\sum_{m=1}^{M} \omega^m \le \sum_{m=1}^{i} \omega^m$ となる値です。

多重クラス指数損失を使用した段階ごとの加法的モデリング

多重クラス指数損失関数を使用した段階ごとの加法的モデリング（SAMME）は、元の AdaBoost アルゴリズムをカテゴリ型対象に拡張するアルゴリズムです。

1. 値を初期化します。

   $w_k = \begin{cases} \frac{w_k}{\Sigma_{i=1}^{K} w_i f_i} & \text{(分析の重みが指定されている場合)} \\ 1/N & \text{(指定されていない場合)} \end{cases}$ を設定します。

   m=1、$w_k^m = w_k$、および $f_k^m = f_k$ を設定します。基本モデルの構築に使用する方法が分析の重みをサポートしない場合でも、分析の重みが初期化されます。

2. 学習セットを使用して、基本モデル m を構築し、学習セットをスコアリングします。$T^m(\cdot)$

   基本モデル m のモデルの重みを設定します。$\omega^m = \log \frac{1-err_m}{err^m} + \log(C-1)$

   この場合、$err^m = \sum_{k=1}^{K} w_k^m f_k II(y_k \ne \hat{y}_k^m)$ となります。

3. 次の基本モデルの重みを設定します。

   $w_k^{m+1} = \frac{a_k^{m+1}}{\Sigma_{i=1}^{K} a_i^{m+1} f_i}$

   この場合、$a_k^{m+1} = w_k^m \exp(\omega^m II(y_k \ne \hat{y}_k^m))$ となります。分析の重みは常に更新されますので注意してください。基本モデルの構築に使用される方法で分析の重みがサポートされていない場合、次の基本モデルの度数の重みが次のように更新されます。

   $f_k^{m+1} = \begin{cases} rv.binom\left(N, w_k^{m+1} f_k\right) & k = 1 \\ rv.binom\left(N - \Sigma_{i=1}^{k-1} f_k^{m+1}, \frac{w_k^{m+1} f_k}{1 - \Sigma_{i=1}^{k-1} w_k^{m+1} f_i}\right) & \text{otherwise} \end{cases}$

   m<M の場合、m=m+1 を設定して ステップ 2 に進みます。そうでない場合、アンサンブル モデルが完了します。

   注 ：$err_m = 0$ または $\omega^m <= 0$ の場合、基本モデルはアンサンブルから除外されます。

スコアリング

SAMME では重み付き多数決の方法を使用してアンサンブル モデルをスコアリングします。

m 番目の基本モデルの k 番目のレコードに対する予測値は、$\hat{y}_k^m = arg\max_{l_i} P_{l_i}^m(X_k)$ となります。

アンサンブルの予測値は $\hat{y}_k = arg\max_{l_i} \sum_{m=1}^{M} \omega^m II(\hat{y}_k^m == l_i)$ となります。タイは無作為に解決されます。

アンサンブルの予測確率は $\hat{p}_{\hat{y}_k} = \sum_{m \in M_{\hat{y}_k}} \frac{\omega^m}{\sum_{i \in M_{\hat{y}_k}} \omega^i} P_{\hat{y}_k}^m(X_k)$ となります。

モデル指標のブースティング

精度

精度は、Naive モデル、リファレンス（シンプル）モデル、アンサンブル モデル（各アンサンブル方法と関連）、および基本モデルについて計算されます。

カテゴリ型対象の場合、分類の精度は次のようになります。

$$\frac{1}{N}\sum_{k=1}^{K} f_k II(y_k == \hat{y}_k)$$

連続型対象の場合は、次のとおりです。

$$R^2 = 1 - \frac{\Sigma_{k=1}^{K} f_k (y_k - \hat{y}_k)^2}{\Sigma_{k=1}^{K} f_k (y_k - \overline{y})^2}$$

この場合 $\overline{y} = \frac{1}{N}\Sigma_{k=1}^{K} f_k y_k$ となります。

R^2 は 1 より大きくなることはありませんが、0 より小さくなる場合があります。

Naïve モデルの場合、$\hat{y}_k$ はカテゴリ型対象ではモーダルなカテゴリ、連続型対象では平均値となります。

リファレンス

Drucker, H. 1997. Improving regressor using boosting techniques. In: Proceedings of the 14th International Conferences on Machine Learning, D. H. Fisher, Jr., ed. San Mateo, CA: Morgan Kaufmann, 107-115.

Freund, Y., および R. E. Schapire. 1995. A decision theoretic generalization of on-line learning and an application to boosting. In: Computational Learning Theory:7 Second European Conference, EuroCOLT ’95, , 23-37.

パス、ストリーム、結合 (PSM) アルゴリズム

アンサンブル モデル作成で PSM 機能の PASS、STREAM、MERGE を実行します。PASS は、データを 1 つだけ渡して非常に大きなデータセットにモデルを構築します。STREAM は、古い学習データを保存または呼び出さずに新しいケースで既存のモデルを更新します。MERGE は、指定された環境にモデルを構築し、構築されたモデルを 1 つのモデルに結合します。

アンサンブル モデルでは、学習セットはブロックと呼ばれるサブセットに分割され、各ブロックにモデルが構築されます。ブロックはさまざまなスレッド（ここでは 1 つのプロセスに 1 つのスレッド）やさまざまなマシンに送信される場合があるため、さまざまなプロセスのモデルを同時に構築することができます。新しいデータ ブロックを受信すると、アルゴリズムはこの手順を繰り返します。そのため、データ ストリームを容易に処理し、アンサンブル モデル作成の増分学習を実行することができます。

パス

PASS 操作には、次のステップがあります。

1. データを学習ブロック、検定セット、ホールドアウト セットに分割します。指定されている場合、学習セットをブロックに分割する場合、度数の重みは無視されますが、検定セットとホールドアウト セット作成時は考慮されます。
2. 学習ブロックに基本モデルを構築し、検定セットに参照モデルを構築します。単一のモデルが検定セットと各学習ブロックに構築されます。
3. 検定セットに基づいて精度を計算し、各基本モデルを評価します。基本モデルのサブセットを、精度に従ってアンサンブル要素として選択します。
4. ホールドアウト セットに基づいて精度を計算し、アンサンブル モデルと参照モデルを評価します。アンサンブル モデルのパフォーマンスが、ホールドアウト セットの参照モデルのパフォーマンスよりよくない場合、参照モデルを使用して新しいケースのスコアリングを行います。

モデルの精度の計算

基本モデルの精度は、検定セットで評価されます。検定セット T で観測される予測値 x_i の各ベクトルと、対応するラベル c_i について、$\hat{c}(x_i)$ を指定されたモデルで予測されたラベルとします。検定エラーが次のように推定されます。

カテゴリ対象 :$E = \frac{1}{\sum_{i=1}^{|T|} f_i} \sum_{i=1}^{|T|} \left(f_i \cdot I\left(c_i \neq \hat{c}\left(x_i \right) \right) \right)$

連続型対象 :$E = \frac{1}{\sum_{i=1}^{|T|} f_i} \sum_{i=1}^{|T|} \left(f_i \cdot |y_i - \hat{y}_i| \right)$

ここで、$c_i \neq \hat{c}(x_i)$ の場合、$I(c_i \neq \hat{c}(x_i))$ は 1 となり、そうでない場合は 0 となります。

指定されたモデルの精度は A=1-E で計算されます。アンサンブル モデル全体と参照モデルの精度は、ホールドアウト セットで評価されます。

ストリーム

新しいケースを入手し、これらのケースで既存のアンサンブル モデルを更新する場合、アルゴリズムは次のようになります。

1. PASS 操作を開始して、新しいデータにアンサンブル モデルを構築します。
2. 新しく作成されたアンサンブル モデルと既存のアンサンブル モデルを結合します。

結合

MERGE 操作は、次のステップに従って行われます。

1. ホールドアウト セットを単一のホールドアウト セットに結合し、必要に応じて、このセットを適切なサイズに縮小します。
2. 検定セットを単一の検定セットに結合し、必要に応じて、このセットを適切なサイズに縮小します。
3. 結合された検定セットに結合された参照モデルを構築します。
4. 結合された検定セットに基づいて精度を計算し、各基本モデルを評価します。基本モデルのサブセットを、精度に従って結合されたアンサンブル要素として選択します。
5. 結合されたホールドアウト セットに基づいて精度を計算し、結合されたアンサンブル モデルと結合された参照モデルを評価します。

適応予測選択

基本モデルの構築に使用する方法に内部予測選択アルゴリズムがあるかどうかによって、2 つの方法があります。

予測選択アルゴリズムがある場合

最初の基本モデルは、その方法の予測選択アルゴリズムに使用できるすべての予測値で構築されます。基本モデル j（j > 1）では i 番目に予測値を次の確率で使用できます。

$$p_i = \max\left(\frac{n'_i + C}{n_i + C}, \beta\right)$$

この場合 n'_i は、以前の j-1 の基本モデルで i 番目の予測値がその方法の予測選択アルゴリズムで選択された回数、n_i は以前の j-1 基本モデルで i 番目の予測値がその方法の予測選択アルゴリズムに使用できるようになった回数、C は p_i の値を平滑にする定数、そして β は p_i の下限となります。

予測選択アルゴリズムがない場合

各基本モデルでは、i 番目の予測値を次の確率で使用できるようになります。

$$p_i = \begin{cases} (1-\rho_i)^2 & \text{if } \rho_i < 0.05 \\ \beta & \text{otherwise} \end{cases}$$

この場合、ρ_i は次で定義するように i 番目の検定の p 値となります。

- カテゴリ型対象およびカテゴリ型予測値の場合、ρ は $G^2 = 2\sum_{i=1}^{I}\sum_{j=1}^{J} G_{ij}^2$ のカイ 2 乗検定となります。この場合、$G_{ij}^2 = \begin{cases} N_{ij}\ln\left(N_{ij}/\hat{N}_{ij}\right) & N_{ij} > 0 \\ 0 & \text{else} \end{cases}$ となり、自由度は $(I-1)(J-1)$ となります。N_{ij} は、X=i のケース数で、Y=j、$N_{i\cdot} = \sum_{j=1}^{J} N_{ij}$、$N_{\cdot j} = \sum_{i=1}^{I} N_{ij}$、および $\hat{N}_{ij} = N_{i\cdot}N_{\cdot j}/N$ となります。
- カテゴリ型対象および連続型予測値の場合、ρ は$F = \frac{\sum_{j=1}^{J} N_j\left(\overline{x}_j - \overline{\overline{x}}\right)^2/(J-1)}{\sum_{j=1}^{J}(N_j-1)s_j/(N-J)}$ の F 検定で、自由度は $J-1, N-J$ となります。N_j は Y=j のケース数で、Y=j の場合、$\overline{x}_j$ および s_j^2 はサンプル平均値、X のサンプル分散となり、そして $\overline{\overline{x}} = \sum_{j=1}^{J} N_j\overline{x}_j/N$ となります。
- 連続型対象およびカテゴリ型予測値の場合、ρ は F test of $F = \frac{\sum_{i=1}^{I} N_i\left(\overline{y}_i - \overline{\overline{y}}\right)^2/(I-1)}{\sum_{i=1}^{I}(N_i-1)s(y)_i/(N-I)}$ で自由度は $I-1, N-I$ となります。N_i は X=i のケース数で、X=i の場合、$\overline{y}_i$ および $s(y)_i^2$ はサンプル平均値および Y のサンプル分散、そして $\overline{\overline{y}} = \Sigma_{i=1}^{I} N_i\overline{y}_i/N$ となります。
- 連続型対象および連続型予測値の場合、ρ は $t = r\sqrt{\frac{N-2}{1-r^2}}$ の t 検定となり、この場合 $r = \frac{\Sigma_{i=1}^{N}(x_i-\overline{x})(y_i-\overline{y})/(N-1)}{\sqrt{s(x)^2 s(y)^2}}$ となり、自由度は $N-2$ となります。$s(x)^2$ は X のサンプル分散で、$s(y)^2$ は Y のサンプル分散となります。

自動カテゴリ バランシング

比較的めったに発生しない対象カテゴリがある場合、モデルの全体の予測率が比較的良い場合でも、多くのモデルはそのめったに発生しないカテゴリの予測メンバーのジョブが悪くなります。自動カテゴリ バランシングで、まれに出現する値を予測する場合のモデルの精度を向上させる必要があります。

レコードが到達すると、いっぱいになるまで学習ブロックに追加されます。各カテゴリのレコードの割合は、$C_i = \frac{w_i}{w}$ のように計算され、w_i はカテゴリ i のレコードの重み付き数で、w はレコード数の合計（重み付き）です。

▶ $C_i < \alpha/(10\cdot|C|)$ となるカテゴリがあり、$|C|$ が対象カテゴリ数で α= 0.3 となる場合、各レコードを学習ブロックから無作為に、次の確率で削除します。

$Min\left\{(1-Min(C)/C_i), \left(1-\frac{\alpha}{|C|}\right)\right\}$

この操作では、頻繁に出現するカテゴリからレコードを削除する傾向があります。新しいレコードを再びいっぱいになるまで学習ブロックに追加し、条件が満たされなくなるまでこの手順を繰り返します。

- $C_i < \alpha/|C|$ のようなカテゴリがある場合、レコード k の度数の重みを $f_k = f_k \max\left(10, \alpha \max(C)/C_{i(k)}\right)$ で計算します。この場合 $i(k)$ は k 番目のレコードのカテゴリです。この操作で、出現頻度の低いカテゴリにより大きな重みを割り当てます。

モデル指標

次の表記法が適用されます。

N	合計レコード数
M	合計基本モデル数
f_k	レコード k の度数の重み
y_k	レコード k の観測対象値
$\hat{y}_k$	アンサンブル モデルによって予測された k の対象値
$\hat{y}_k^m$	基本モデル m によって予測されたレコード k の対象値

精度

精度は、Naive モデル、リファレンス（シンプル）モデル、アンサンブル モデル（各アンサンブル方法と関連）、および基本モデルについて計算されます。

カテゴリ型対象の場合、分類の精度は次のようになります。

$$\frac{1}{N}\sum_{k=1}^{K} f_k II\left(y_k == \hat{y}_k\right)$$

とします。ここで、

$$II\left(y_k = \hat{y}_k\right) = \begin{cases} 1, \text{if}\left(y_k = \hat{y}_k\right) \\ 0, \text{otherwise} \end{cases}$$

連続型対象の場合は、次のとおりです。

$$R^2 = 1 - \frac{\Sigma_{k=1}^{K} f_k (y_k - \hat{y}_k)^2}{\Sigma_{k=1}^{K} f_k (y_k - \overline{y})^2}$$

この場合 $\overline{y} = \frac{1}{N}\Sigma_{k=1}^{K} f_k y_k$ となります。

R^2 は 1 より大きくなることはありませんが、0 より小さくなる場合があります。

Naïve モデルの場合、$\hat{y}_k$ はカテゴリ型対象ではモーダルなカテゴリ、連続型対象では平均値となります。

多様性

多様性は 0 ～ 1 の範囲の測定で、数値が大きいほどより多様になります。基本モデルで予測値がどれほど異なるかを示します。

カテゴリ型対象の場合、多様性は次のようになります。

$$\frac{1}{N \cdot M^2} \sum_{k=1}^{K} f_k L\left(y_k\right)\left[M - L\left(y_k\right)\right]$$

この場合 $L\left(y_k\right) = \sum_{m=1}^{M} II\left(y_k = \hat{y}_k^m\right)$ および $II\left(y_k = \hat{y}_k^m\right)$ は上記で定義されたとおりです。

多様性は、連続型対象には使用できません。

スコアリング

アンサンブル モデルを使用してスコアリングする方法がいくつかあります。

連続型対象

平均値: $\hat{y}_{i,PSM} = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_{i,m}$

中央値: $\hat{y}_{i,PSM} = Median_1^M\left(\hat{y}_{i,m}\right)$

この場合、$\hat{y}_{i,PSM}$ は、ケースi の最終予測値で、$\hat{y}_{i,m}$ は、ケース i の m 番目の基本モデルの予測値です。

カテゴリ対象

票決 : $d_{m,k}$ が予測値の指定されたベクトルの m 番目の基本モデルに対するラベル出力を示すとします。m 番目の基本モデルに割り当てられたラベルがk 番目の対象カテゴリである場合 $d_{m,k} = 1$ となり、そうでない場合は 0 となります。M 個のベース モデルと K 個のベース モデルの合計があります。多数の基本モデルに割り当てられている場合、多数決の方法で、j 番目のカテゴリが選択されます。次の方程式を満たします。

$$\sum_{m=1}^{M} d_{m,j} = \max_{k=1}^{K}\left(\sum_{m=1}^{M} d_{m,k}\right)$$

E_m が、m 番目の基本モデルに推定された検定エラーとなります。重み付き多数決の重みが、次の式にしたがって計算されます。

$$w_m = \max\left(\log\frac{1-E_m}{E_m}, 0\right) / \sum_{i=1}^{M} \max\left(\log\frac{1-E_i}{E_i}, 0\right)$$

確率票決 :$p_{m,k}$ が、予測値の指定されたベクトルの m 番目の基本モデルで k 番目の対象カテゴリに推定された事後確率であると仮定します。次のルールにより、基本モデルで計算された確率を結合します。対応する方程式を満たすよう、j 番目のカテゴリが選択されます。

- 最も高い確率： $\max_{m=1}^{M}(p_{m,j}) = \max_{k=1}^{K}\left(\max_{m=1}^{M}(p_{m,k})\right)$
- 最も高い平均値の確率 ：$\frac{1}{M}\sum_{m=1}^{M} p_{m,j} = \max_{k=1}^{K}\left(\frac{1}{M}\sum_{m=1}^{M} p_{m,k}\right)$

タイは無作為に解決されます。

Softmax 平滑化： Softmax 関数を使用して、確率を平滑化できます。

$$p_i^S = \frac{Exp(p_i)}{\sum_{i=1}^{K} Exp(p_i)}$$

この場合、 p_i は、カテゴリ i のルールに基づいた確信度で、p_i^S は平滑化値です。

章
16

因子／主成分分析アルゴリズム

概要

因子分析ノードは、主成分分析および 6 種類の因子分析を行います。

一次計算

因子の抽出

主成分分析

因子 m に基づいた因子負荷の行列は次のようになります。

$$\Lambda_m = \Omega_m \Gamma_m^{\frac{1}{2}}$$

ここで、

$$\Omega_m = (\omega_1, \omega_2, ..., \omega_m)$$

$$\Gamma_m = diag\left(|\gamma_1|, |\gamma_2|, ..., |\gamma_m|\right)$$

変数 i の共通性は、次の式により与えられます。

$$h_i = \sum_{j=1}^{m} |\gamma_j| \omega_{ij}^2$$

相関行列の分析

$\gamma_1 \geq \gamma_2 \geq ... \geq \gamma_m$ は固有値で、ω_i は対応する $\mathbf{R}$ の固有ベクトルになります。ここで、$\mathbf{R}$ は相関行列です。

共分散行列の分析

$\gamma_1 \geq \gamma_2 \geq ... \geq \gamma_m$ は固有値で ω_i は対応する $\mathbf{\Sigma}$ の固有ベクトルになります。ここで、$\mathbf{\Sigma} = (\sigma_{ij})_{n \times n}$ は相関行列になります。

再スケールされた負荷行列は $\Lambda_{mR} = [diag(\Sigma)]^{-\frac{1}{2}} \Lambda_m$ になります。

再スケールされた変数 i の共通性は $h_{iR} = \sigma_{ii}^{-1} h_i$ になります。

主因子法

相関行列の分析

共通性および因子負荷の反復解が探されます。反復 i において、前の反復からの共通性は $\mathbf{R}$ の対角に配置され、その結果となる $\mathbf{R}$ は $\mathbf{R}_i$ と表されます。$\mathbf{R}_i$ に対して固有分析が実施され、新しく変数 j の共通性が次の式により推定されます。

$$h_{j(i)} = \sum_{j=1}^{m} \left|\gamma_{k(i)}\right| \omega_{jk(i)}^2$$

因子負荷は次の式により算出されます。

$$\Lambda_{m(i)} = \Omega_{m(i)} \Gamma_{m(i)}^{\frac{1}{2}}$$

反復は最大数（デフォルトは 25）に達するまで、または共通性の推定値の最大変化が収束基準（デフォルトは 0.001）未満になるまで繰り返されます。

共分散行列の分析

この分析は相関行列の分析とほぼ同じですが、共分散行列 $\mathbf{R}$ の代わりに Σ が使用されます。収束は、再スケールされた共通性の推定値の最大変化に依存しています。

反復 i で、再スケールされた負荷行列は $\Lambda_{m(i)R} = [diag(\Sigma)]^{-\frac{1}{2}} \Lambda_{m(i)}$ となります。再スケールされた変数 i の共通性は $h_{j(i)R} = \sigma_{ii}^{-1} h_{j(i)}$ になります。

最尤法

Λ および ψ^2 の最尤法による解は、次の式

$$F = tr\left[\left(\Lambda\Lambda' + \psi^2\right)^{-1}\mathbf{R}\right] - \log\left|\left(\Lambda\Lambda' + \psi^2\right)^{-1}\mathbf{R}\right| - p$$

を Λ と ψ に関して極小化することにより得られます。ここで p は変数の数、Λ は因子負荷行列、そして ψ^2 は一意の分散の対角行列を表します。

F の極小化は、TwoStep アルゴリズムを利用して実施されます。まず、与えられた y に対して、F の条件極小化が検索されます。これにより、Newton-Raphson の手続きを使って数値的に極小化する関数 $f(\psi)$ が得られます。$\mathbf{x}^{(s)}$ を s 番目の反復における y の対角線要素の対数を含む列ベクトルとします。すると、次のようになります。

$$\mathbf{x}^{(s+1)} = \mathbf{x}^{(s)} - \mathbf{d}^{(s)}$$

ここで $\mathbf{d}^{(s)}$ は、次の連立一次方程式の解を表します。

$$\mathbf{H}^{(s)}\mathbf{d}^{(s)} = \mathbf{h}^{(s)}$$

さらに、ここで

$$\mathbf{H}^{(s)} = \frac{\partial^2 f(\psi)}{\partial x_i \partial x_j}$$

$\mathbf{h}^{(s)}$ は、$\frac{\partial f(\psi)}{\partial x_i}$ を含む列ベクトルを表します。開始点 $\mathbf{x}^{(1)}$ は

$$\mathbf{x}_i^1 = \begin{cases} \log\left[\left(1 - \frac{m}{2p}\right)/r^{ii}\right] & \text{(ML および GLS の場合)} \\ \left[\left(1 - \frac{m}{2p}\right)/r^{ii}\right]^{\frac{1}{2}} & \text{(ULS の場合)} \end{cases}$$

ここで m は因子数を、r^{ii} は $\mathbf{R}^{-1}$ の i 番目の対角線要素を表します。

$f(\psi)$, $\frac{\partial f}{\partial x_i}$、および $\frac{\partial^2 f}{\partial x_i \partial x_j}$ の値は、固有値 $\gamma_1 \leq \gamma_2 \leq ... \leq \gamma_p$ および 対応する行列 $\psi\mathbf{R}^{-1}\psi$ の固有ベクトル ω_1, $\omega_2, \ldots, \omega_p$ として表すことができます。つまり、次のようになります。

$$f(\psi) = \sum_{k=m+1}^{p} \left(\log \gamma_k + \gamma_k^{-1} - 1\right)$$

$$\frac{\partial f}{\partial x_i} = \sum_{k=m+1}^{p} \left(1 - \gamma_k^{-1}\right)\omega_{ik}^2$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = -\delta_{ij}\frac{\partial f}{\partial x_i} + \sum_{k=m+1}^{p} \omega_{ik}\omega_{jk}\left(\sum_{n=1}^{m} \frac{\gamma_k + \gamma_n - 2}{\gamma_k - \gamma_n}\omega_{in}\omega_{jn} + \delta_{ij}\right)$$

ここで、

$$\delta_{ij} = \begin{cases} 1 & if\ i = j \\ 0 & if\ i \neq j \end{cases}$$

2 階（2 次）導関数の近似値

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \cong \left(\sum_{k=m+1}^{p} \omega_{ik}\omega_{jk}\right)^2$$

は、初期ステップで、正確な 2 階（2 次）導関数の行列が正値確定でない場合、またはベクトル $\mathbf{d}$ のすべての要素が 0.1 を超える場合に使用されます。$\frac{\partial^2 f}{\partial x_i^2} < 0.05$（Heywood 変数の場合）、対角線要素は 1 と置換され、その列と行の残りの要素は 0 に設定されます。ステップ $\mathbf{d}$ により $f(\psi)$ の値が減少しない場合、そのステップは $f(\psi)$ の値が減少するまで繰り返し半減されます。また、25 回半減しても値が減少しない場合は計算が終了します。（この場合、計算は終了しません。）このステップは、$\mathbf{d}$ の要素の最大絶対値が基準値（デフォルトは 0.001）未満になるまで、または最大反復数（デフォルトは 25）に達するまで繰り返されます。収束値 ψ（$\hat{\psi}$ により示される）を使用して、行列 $\hat{\psi}\mathbf{R}^{-1}\hat{\psi}$ に対して固有分析が実施されます。因子負荷は次のように計算されます。

$$\hat{\Lambda}_m = \hat{\psi}\Omega_m\left(\Gamma_m^{-1} - \mathbf{I}_m\right)^{\frac{1}{2}}$$

ここで、

$$\Gamma_m = diag\left(\gamma_1, \gamma_2, ..., \gamma_m\right)$$

$$\Omega_m = \left(\omega_1, \omega_2, ..., \omega_m\right)$$

重みなしおよび一般化最小 2 乗法

ULS（重みなし最小 2 乗法）および GLS（一般化最小 2 乗法）では、最尤法と同じ基本アルゴリズムが使用されますが、次の点が異なっています。

$$f(\psi) = \begin{cases} \sum_{k=m+1}^{p} \frac{\gamma_k^2}{2} & \text{(ULS の場合)} \\ \sum_{k=m+1}^{p} \frac{(\gamma_k - 1)^2}{2} & \text{(GLS の場合)} \end{cases}$$

ULS の場合、固有分析は行列 $\mathbf{R} - \psi^2$ に対して実施されます。ここで $\gamma_1 \geq \gamma_2 \geq ... \geq \gamma_p$ は固有値を表します。導関数の観点から、ULS については次のようになります。

$$\frac{\partial f}{\partial x_i} = 2x_i \sum_{k=m+1}^{p} \gamma_k \omega_{ik}^2$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = 4\left[x_i x_j \sum_{k=m+1}^{p} \omega_{ik}\omega_{jk} \sum_{n=1}^{m} \frac{\gamma_k + \gamma_n}{\gamma_k - \gamma_n}\omega_{ik}\omega_{jk} + \delta_{ij} \sum_{k=m+1}^{p} \left(x_i^2 - \frac{\gamma_k}{2}\right)\omega_{ik}^2\right]$$

そして

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \cong 4x_i x_j \left(\sum_{k=m+1}^{p} \omega_{ik}\omega_{jk}\right)^2$$

GLS の場合は次のようになります。

$$\frac{\partial f}{\partial x_i} = \sum_{k=m+1}^{p} \left(\gamma_k^2 - \gamma_k\right)\omega_{ik}^2$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \delta_{ij}\frac{\partial f}{\partial x_i} + \sum_{k=m+1}^{p} \gamma_k \omega_{ik}\omega_{jk}\left(\sum_{n=1}^{m} \gamma_n \frac{\gamma_k + \gamma_n - 2}{\gamma_k - \gamma_n}\omega_{in}\omega_{jn} + r^{ii}\exp\left(\frac{x_i + x_j}{2}\right)\right)$$

そして

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \cong \left(\sum_{k=m+1}^{p} \omega_{ik}\omega_{jk}\right)^2$$

また、ULS の因子負荷は、次のように取得されます。

$$\hat{\Lambda}_m = \Omega_m \Gamma_m^{\frac{1}{2}}$$

ML および GLS の場合の m 因子のカイ 2 乗統計量は、次の式により得られます。

$$\chi_m^2 = \left(W - 1 - \frac{2p+5}{6} - \frac{2m}{3} \right) f\left(\hat{\psi}\right)$$

自由度は $\left((p-m)^2 - p - m\right)/2$ になります。

α 因子分析

α 因子分析には反復処理が含まれており、各反復 i に対して次の処理が行われます。

次の式に対する固有値（$\gamma_{(i)}$）および固有ベクトル $\omega_{(i)}$）が算出されます。

$$\mathbf{H}_{(i-1)}^{\frac{1}{2}} (\mathbf{R} - \mathbf{I}) \mathbf{H}_{(i-1)}^{\frac{1}{2}} + \mathbf{I}$$

が計算されます。

新しい共通性は次のようになります。

$$h_{k(i)} \left(\sum_{j=1}^{m} \left|\gamma_{j(i)}\right| \omega_{kj(i)}^2 \right) h_{k(i-1)}$$

共通性の初期値 $\mathbf{H}_0$ は、次のようになります。

$$h_{io} = \begin{cases} 1 - \frac{1}{r^{ii}} & |\mathbf{R}| \geq 10^{-8} \text{となり、それ以外は} 0 \leq h_{io} \leq 1 \\ \max_j |r_{ij}| & \text{となります。} \end{cases}$$

ここで r^{ii} は $\mathbf{R}^{-1}$ の i 番目の対角線エントリを表します。

$|R| \geq 10^{-8}$ で、すべての r^{ii} が 1 と等しい場合、処理は終了します。一部の i に対して $\max_j |r_{ij}| > 1$ の場合、処理は終了します。

反復は、次のいずれかが真（true）の場合に停止します。

$$\max_k \left| h_{k(i)} - h_{k(i-1)} \right| < \epsilon$$

$$i = MAX$$

$$h_{k(i)} = 0 \text{ (任意の} k$$

共通性は反復が停止した時の値になります。ただし、最後の停止基準が真（true）の場合は除きます（この場合処理が終了する）。因子パターン行列は次のようになります。

$$F_m = H_{(f)}^{\frac{1}{2}} \Omega_{m(f)} \Gamma_{m(f)}^{\frac{1}{2}}$$

ここで f は最後の反復を表します。

イメージ因子法

相関行列の分析

$\mathbf{S}^{-1}\mathbf{R}\mathbf{S}^{-1}$ の固有値と固有ベクトルが見つかりました。

$$S^2 = diag\left(\frac{1}{r^{11}}, ..., \frac{1}{r^{nn}}\right)$$

ここで r^{11} は、$\mathbf{R}^{-1}$ の i 番目の対角線要素を表します。

因子パターン行列は次のようになります。

$$\mathbf{F}_m = \mathbf{S}\Omega_m(\Lambda_m - \mathbf{I}_m)\Lambda_m^{-\frac{1}{2}}$$

ここで Λ_m および Ω_m は、1 より大きい m 個の固有値に対応しています（および関連する固有ベクトル）。$m = 0$ の場合、処理は終了します。

共通性は次のようになります。

$$h_i = \sum_{j=1}^{m} \frac{(\gamma_j - 1)^2 \omega_{ij}^2}{(\gamma_j r^{ii})}$$

イメージ共分散行列は次のようになります。

$$\mathbf{R} + \mathbf{S}^2\mathbf{R}^{-1}\mathbf{S}^2 - 2\mathbf{S}^2$$

反イメージ共分散行列は次のようになります。

$$\mathbf{S}^2\mathbf{R}^{-1}\mathbf{S}^2$$

共分散行列の分析

共分散行列の分析時には、相関行列 $\mathbf{R}$ の代わりに共分散行列 Σ が使用されます。計算は、相関行列の場合と同じように行われます。

再スケールされた因子パターン行列は次のようになります。

$$\mathbf{F}_{mR} = [diag(\Sigma)]^{\frac{1}{2}}\mathbf{F}_m$$

また、再スケールされた変数 i の共通性は $h_{iR} = \sigma_{ii}^{-1}h_i$ になります。

因子の回転

直交回転法

回転は、反復の最大数に達するまで、または収束基準を満たすまで、因子のペアに対して循環的に行われます。すべての直交回転法に対してアルゴリズムは同じで、回転角度のタンジェント値の計算だけが異なります。

因子パターン行列は、共通性の平方根により正規化されます。

$$\Lambda_m^* = \mathbf{H}^{\frac{1}{2}}\Lambda_m$$

ここで、

$\Lambda_m = (\underline{\lambda}_1, ..., \underline{\lambda}_m)$ は因子パターン行列です。

$\mathbf{H} = diag\,(h_1, ..., h_n)$

変換行列 $\mathbf{T}$ は $\mathbf{I}_m$ に初期化されます。

各反復 i において、

■ 収束基準は次のようになります。

$$SV_{(i)} = \sum_{j=1}^{m}\left(n\sum_{k=1}^{n}\lambda_{kj(i)}^{*4} - \left(\sum_{k=1}^{n}\lambda_{kj(i)}^{*2}\right)^2\right)/n^2$$

ここで $\Lambda_{m(1)}^*$ の初期値は、元の因子パターン行列を表します。以降の反復の初期値は、すべての因子ペアが回転された時点の $\Lambda_{m(i-1)}^*$ の最終値になります。

すべての因子ペア（λ_j, λ_k、ここでは $k > j$）に対して、次の計算が行われます。

■ 回転の角度は次のようになります。

$$P = \frac{1}{4}\tan^{-1}\left(\frac{X}{Y}\right)$$

ここで、

$$X = \begin{cases} D - \frac{2AB}{n} & \text{Varimax} \\ D - \frac{mAB}{n} & \text{Equamax} \\ D & \text{Quartimax} \end{cases}$$

$$Y = \begin{cases} C - \left(\frac{A^2 - B^2}{n}\right) & \text{Varimax} \\ C - \frac{m(A^2 - B^2)}{C^{2n}} & \text{Equamax} \\ & \text{Quartimax} \end{cases}$$

$$u_{p(i)} = f_{pj(i)}^{*2} - f_{pk(i)}^{*2} \quad v_{p(i)} = 2f_{pj(i)}^{*}f_{pk(i)}^{*} \quad p = 1, ..., n$$

$$A = \sum_{p=1}^{n} u_{p(i)} \qquad B = \sum_{p=1}^{n} v_{p(i)}$$
$$C = \sum_{p=1}^{n} \left[u_{p(i)}^2 - v_{p(i)}^2 \right] \quad D = \sum_{p=1}^{n} 2u_{p(i)} v_{p(i)}$$

$|\sin(P)| \leq 10^{-15}$ の場合、因子のペアに対して回転は行われません。

- 新しい回転因子は次のようになります。

$$\left(\underline{\tilde{\lambda}}_{j(i)}, \underline{\tilde{\lambda}}_{k(i)} \right) = \left(\underline{\lambda}^*_{j(i)}, \underline{\lambda}^*_{k(i)} \right) \begin{vmatrix} \cos(P) & -\sin(P) \\ \sin(P) & \cos(P) \end{vmatrix}$$

  ここで $\underline{\lambda}^*_{j(i)}$ は、この反復で計算された因子 j の最後の値になります。

- 得られた回転変換行列は次のようになります。

$$\left(\tilde{t}_j, \tilde{t}_k \right) = \left(t_j, t_k \right) \begin{vmatrix} \cos(P) & -\sin(P) \\ \sin(P) & \cos(P) \end{vmatrix}$$

  ここで、t_j と t_k は、$\mathbf{T}$ の j 番目と k 番目の列の計算された最後の値になります。

- 反復は、次の場合に終了します。

$$\left| SV_{(i)} - SV_{(i-1)} \right| \leq 10^{-5}$$

  または、反復の最大数に達した場合に終了します。

最終的な回転された因子パターン行列は次のようになります。

$$\hat{\Lambda}_m = H^{\frac{1}{2}} \Lambda^*_{m(f)}$$

ここで $\Lambda^*_{m(f)}$ は、最後の反復の値になります。

因子を負の合計に反映します。次のような場合

$$\sum_{i=1}^{n} \tilde{\lambda}_{ij(f)} < 0$$

すると、以下のようになります。

$$\underline{\tilde{\lambda}}_j = -\underline{\tilde{\lambda}}_{j(f)}$$

回転された因子を次のようになるように再配置します。

$$\sum_{j=1}^{n} \tilde{\lambda}_{j1}^2 \geq \ldots \geq \sum_{j=1}^{n} \tilde{\lambda}_{jm}^2$$

共通性は次のようになります。

$$h_j = \sum_{i=1}^{m} \tilde{\lambda}_{ji}^2$$

直接オブリミン回転法

直接オブリミン回転法(Jennrich および Sampson, 1966)は、斜交回転に用いられます。ユーザーは、パラメータ δ を選択することができます。デフォルト値は $\delta = 0$ です。

因子パターン行列は、共通性の平方根により正規化されます。

$$\Omega_m^* = H^{\frac{1}{2}} \Lambda_m$$

ここで、

$$h_j = \sum_{k=1}^{m} \lambda_{jk}^2$$

Kaiser が指定されていない場合、正規化は行われません。

初期化

因子相関行列 $\mathbf{C}$ は、$\mathbf{I}_m$ に初期化されます。また、次の計算も行われます。

$$s_k = \begin{cases} 1 & \text{if Kaiser} \\ h_k & \text{if no Kaiser} \end{cases} \quad k = 1, ..., n$$

$$u_i = \sum_{j=1}^{n} \lambda_{ji}^{*2} \; i = 1, ..., m$$

$$v_i = \sum_{j=1}^{n} \lambda_{ji}^{*4}$$

$$x_i = v_i - \left(\frac{\delta}{n}\right) u_i^2$$

$$D = \sum_{i=1}^{m} u_i$$

$$G = \sum_{i=1}^{m} x_i$$

$$H = \sum_{k=1}^{n} s_i^2 - \left(\frac{\delta}{n}\right) D^2$$

$$FO = H - G$$

各反復において、すべての有効な因子ペアが回転されます。因子ペア $\underline{\lambda}_p^*$ および $\underline{\lambda}_p^*$（$p \neq q$）に対して、次の計算が行われます。

$$D_{pq} = D - u_p - u_q$$

$$G_{pq} = G - x_p - x_q$$

$$s_{pq,i} = s_i - \lambda_{ip}^{*2} - \lambda_{iq}^{*2}$$

$$y_{pq} = \sum_{i=1}^{n} \lambda_{ip}^* \lambda_{iq}^*$$

$$z_{pq} = \sum_{i=1}^{n} \lambda_{ip}^{*2} \lambda_{iq}^{*2}$$

$$T = \sum_{i=1}^{n} s_{pq,i} \lambda_{ip}^{*2} - \left(\frac{\delta}{n}\right) u_p D_{pq}$$

$$Z = \sum_{i=1}^{n} s_{pq,i} \lambda_{ip}^* \lambda_{iq}^* - \left(\frac{\delta}{n}\right) y_{pq} D_{pq}$$

$$P = \sum_{i=1}^{n} \lambda_{ip}^{*3} \lambda_{iq}^* - \left(\frac{\delta}{n}\right) u_p y_{pq}$$

$$R = z_{pq} - \left(\frac{\delta}{n}\right) u_p u_q$$

$$P' = \frac{3}{2}\left(c_{pq} - \frac{P}{x_p}\right)$$

$$Q' = \frac{1}{2}(x_p - 4c_{pq}P + R + 2T)/x_p$$

$$R' = \frac{1}{2}(c_{pq}(T + R) - P - Z)/x_p$$

式 $b^3 + P'b^2 + Q'b + R = 0$ の累乗根（ルート）a が計算されます。また、次の計算が行われます。

$$A = 1 + 2c_{pq}a + a^2$$

$$t_1 = |A|^{\frac{1}{2}}$$

$$t_2 = \frac{a}{t_1}$$

回転した因子ペアは次のようになります。

$$\left(\tilde{\underline{\lambda}}_p^*, \tilde{\underline{\lambda}}_q^*\right) = \left(\underline{\lambda}_p^*, \underline{\lambda}_q^*\right) \begin{vmatrix} t_1 & -a \\ 0 & 1 \end{vmatrix}$$

これらの値が、前の因子値と置換されます。

新しい値は、次のように算出されます。

$$\tilde{u}_p = |A| u_p$$

$$\tilde{x}_p = A^2 x_p$$

$$\tilde{v}_q = \sum_{i=1}^{n} \tilde{\lambda}_{iq}^{*4}$$

$$\tilde{u}_q = \sum_{i=1}^{n} \tilde{\lambda}_{iq}^{*2}$$

$$\tilde{x}_q = \tilde{v}_q - \left(\frac{\delta}{n}\right) \tilde{u}_q^2$$

$$\tilde{S}_k = S_{pq,k} + \tilde{\lambda}_{kp}^{*2} + \tilde{\lambda}_{kq}^{*2}$$

$$\tilde{D} = D_{pq} + \tilde{u}_p + \tilde{u}_q$$

$$\tilde{G} = G_{pq} + \tilde{x}_p + \tilde{x}_q$$

波型記号（~）が付けられている値はすべて、元の値と置き換えられ、以降の計算で使用されます。

因子 p を持つ新しい因子相関は次のようになります。

$$\tilde{c}_{ip} = t_1^{-1} c_{ip} + t_2 c_{iq} \; (i \neq p)$$

$$\tilde{c}_{pi} = \tilde{c}_{ip}$$

$$\tilde{c}_{pp} = 1$$

すべての因子ペアの回転後、次の場合に反復は終了します。

最大反復数に達した場合、または次の場合

$$\left|F1_{(i)} - F1_{(i-1)}\right| < (FO)(EPS)$$

ここで、

$$F1_{(i)} = \tilde{H} - \tilde{G}$$

$$\tilde{H} = \sum_{i=1}^{n} \tilde{s}_k^2 - \left(\frac{\delta}{n}\right)\tilde{D}^2$$

$$F1_{(0)} = FO$$

それ以外の場合は、もう一度因子ペアが回転されます。

回転された因子パターン行列は最終的に次のようになります。

$$\tilde{\lambda}_m = \mathbf{H}^{\frac{1}{2}}\tilde{\lambda}_m^*$$

ここで $\tilde{\lambda}_m$ は最後の反復における値を表します。

因子構造行列は次のようになります。

$$\mathbf{S} = \tilde{\Lambda}_m\tilde{\mathbf{C}}_m$$

ここで $\tilde{\mathbf{C}}_m$ は最後の反復における因子相関行列を表します。

プロマックス回転法

プロマックス回転法は、計算が高速な回転法です(Hendrickson および 白, 1964)。この速度は、最初に直交バリマックス解に回転し、次に単純構造に適合させるために因子の直交性を緩和することによって達成されます。

バリマックス回転法を使って直交回転行列 $\Lambda_R = \{\lambda_{ij}\}$ を取得します。

行列 $\mathbf{P} = (p_{ij})_{p\times m}$ が算出されます。ここで、

$$p_{ij} = \left|\frac{\lambda_{ij}}{\left(\sum_{j=1}^{m}\lambda_{ij}^2\right)^{\frac{1}{2}}}\right|^{k+1}\frac{\left(\sum_{j=1}^{m}\lambda_{ij}^2\right)^{\frac{1}{2}}}{\lambda_{ij}}$$

ここで k はプロマックス回転 $(k>1)$ のべき乗です。

行列 $\mathbf{L}$ が算出されます。

$$\mathbf{L} = \left(\Lambda'_R\Lambda_R\right)^{-1}\Lambda'_R\mathbf{P}$$

行列 $\mathbf{L}$ が列により、変換行列に正規化されます。

$$\mathbf{Q} = \mathbf{LD}$$

ここで $\mathbf{D} = (diag(\mathbf{L'L}))^{\frac{1}{2}}$ は $\mathbf{L}$ の列を正規化する対角行列を表します。

この時点で、回転因子は次のようになります。

$$f_{promax\_temp} = \mathbf{Q}^{-1} f_{varimax}$$

$var(f_{promax\_temp}) = (\mathbf{Q}'\mathbf{Q})^{-1}$ で、対角要素が 1 と等しくないため、回転因子を次のように修正する必要があります。

$$f_{promax} = \mathbf{C} f_{promax\_temp}$$

ここで、$\mathbf{C} = \left\{ diag\left((\mathbf{Q}'\mathbf{Q})^{-1} \right) \right\}^{-\frac{1}{2}}$ となります。

回転因子パターンは次のようになります。

$$\Lambda_{promax} = \Lambda_{varimax} \mathbf{Q}\mathbf{C}^{-1}$$

因子の相関行列は次のようになります。

$$\mathbf{R}_{ff} = \mathbf{C}\left(\mathbf{Q}'\mathbf{Q}\right)^{-1}\mathbf{C}'$$

因子構造行列は次のようになります。

$$\Lambda_S = \Lambda_{promax} \mathbf{R}_{ff}$$

因子スコア係数

IBM® SPSS® Modeler は、因子スコア係数演算の回帰法を使用しています (Harman, 1976)。

$$\mathbf{W} = \begin{cases} \Lambda_m \Gamma_m^{-1} & \text{回転なしの PCA} \\ \Lambda_m (\Lambda'_m \Lambda_m)^{-1} & \text{回転ありの PCA} \\ \mathbf{R}^{-1}\mathbf{S}_m & \text{その他の場合} \end{cases}$$

ここで $\mathbf{S}_m$ は因子構造行列を表します。直交回転の場合、$\mathbf{S}_m = \Lambda_m$ になります。

回転なしの主成分分析の場合、$|\gamma_i| \leq 10^{-8}$ ならば、因子スコア係数は計算されません。回転ありの主成分分析の場合、$\Lambda'_m \Lambda_m$ の決定因子が 10^{-8} 未満ならば、係数は計算されません。それ以外の場合、$\mathbf{R}$ が単数ならば、因子スコア係数は計算されません。

空白の処理

デフォルトでは、入力または出力フィールドに欠損値があるケースは、すべての結果演算の基となる相関行列の演算からは削除されます。[**完全なレコードのみ使用**] オプションの選択を解除している場合は、他のフィールドに欠損値があるかどうかに関わらず、相関に関連する 2 つのフィールドに完全なデータを持つレコードに基づいて、相関行列 $\mathbf{R}$ 中の各相関が計算されます。一部のデータ セットでは、この方法により非正値定 $\mathbf{R}$ 行列になる可能性があるため、モデルを推定することはできません。

二次計算

フィールド統計量および他の計算

回帰式ノードの詳細出力に表示されている統計量は、IBM® SPSS® Statistics の FACTOR プロシージャと同じ方法で計算されます。詳細は、『SPSS Statistics Factor algorithm』を参照してください。このドキュメントは、http://www.ibm.com/support から入手できます。

生成されたモデル／スコアリング

因子スコア

因子スコアは、因子スコア係数をレコードの入力フィールド値に適用することにより、スコアリングされたレコードに割り当てられます。

$$fs_k = \sum_{i=1}^{n} w_{ki} f_i$$

ここで fs_k は k 番目の因子の因子スコア、w_{ki} は i 番目の入力フィールド（**W** 行列から）および k 番目の因子のの因子スコア係数 、そして f_i はスコアリング対象レコードの i 番目の入力フィールドの値を表します。詳細は、 p. 148 因子スコア係数 を参照してください。

空白の処理

最終モデルにおいて、入力フィールドに欠損値があるレコードはスコアリングできません。そのため、$null$ の因子／コンポーネント スコア値が割り当てられます。

章
17

フィールド選択のアルゴリズム

概要

データ マイニングのトラブルは時に数百、または数千の変数が伴うことです。最終的にモデリング プロセスでは、どの変数をモデルに使用するかをチェックする時間と努力が必要になります。ニューラル ネットワークまたはディシジョン ツリーを変数にフィットするには、時間がかかります。

フィールド選択をすることによりサイズを縮小でき、モデルの属性の管理を容易にできるようになります。予測分析プロセスにフィールド選択を加えるといくつかの次の利点が生じます。

- 予測モデルを構築するのに必要な選択のスコープを狭くして簡略します。
- 本質的に予測フィールドに直接働きかけることができるので予測モデルの作成に要するメモリと計算時間を最小化できます。
- 正確で、費用のままらないモデルを作成できます。
- 予測モデルが予測フィールドに基づいているのでスコアを生成する時間を短縮できます。

一次計算

フィールド選択は、次の 3 つの段階から成り立っています。

- **スクリーニング**: 重要でなく問題を含んだ予測値またはケースを削除します。
- **ランク付け**:残りの予測値をソートし、ランクを割り当てます。
- **選択**:機能の重要なサブセットを識別し、後続のモデルで使用します。

ここで記述されたアルゴリズムは、一連の予測値がターゲット変数を予測するために使用される監視学習の状況に制限されます。分析におけるすべての変数は、カテゴリ変数または連続変数のいずれかになります。一般的なターゲット変数には、顧客が解約するか否か、購入するか否か、または病気が存在するか否かなどがあります。

機能、**変数**および**属性**の用語は、互換性を持って用いられることがよくあります。このドキュメントでは、機能選択アルゴリズムへの入力に関して、後続のモデル作成プロセスで使用するアルゴリズムに実際選択される予測値を参照する機能によって、変数と予測値を使用します。

スクリーニング

このステップで、予測に有用な情報を提供しない変数とケースを削除し、そのような変数についての警告を発行します。

次の変数が削除されます。

- すべて欠損値の変数。

- すべて定数の変数。
- ケース ID を表す変数。

次のケースが削除されます。

- 対象値が欠けているケース。
- すべての予測フィールドが欠損値のケース。

ユーザーの設定に基づいて、次の変数が削除されます。

- 欠損値が m_1% を超える変数。
- m_2% を超えるケースの単一カテゴリ カウントがあるカテゴリ型変数。
- 標準偏差が < m_3% の連続型変数。
- 変動係数が |CV| < m_4% の連続型変数。CV = 標準偏差/平均。
- m_5% を超えるケースが多数のカテゴリを持つカテゴリ型変数。

m_1、m_2、m_3、m_4、および m_5 は、ユーザー制御のパラメータです。

予測フィールドのランク付け

このステップでは一度に 1 つの予測フィールドを検討して、各予測フィールドが単独で対象変数をどのくらい適切に予測するかを調べます。予測フィールドは、ユーザー定義の基準に従ってランク付けされます。利用可能な基準は、対象および予測フィールドの測定レベルによって異なります。

各変数の**重要な**値は、次のようにして計算されます。$(1-p)$。ここで、p は、次に述べるように、候補予測フィールドと対象変数間のアソシエーションの、適切な統計検定の p 値です。

カテゴリ対象

このセクションでは、次のシナリオに沿ったカテゴリ対象の予測フィールドのランク付けを説明します。

- すべての予測フィールドがカテゴリ型
- すべての予測フィールドが連続型
- カテゴリ型と連続型の予測フィールドが混在

すべての予測フィールドがカテゴリ型

次の表記法が適用されます。

X	I カテゴリ数で検討中の予測フィールド。
Y	J カテゴリの対象変数。
N	ケース総数。
N_{ij}	X = i および Y = j のケース数。

$N_{i\cdot}$ X = i のケース数。$N_{i\cdot} = \sum_{j=1}^{J} N_{ij}$

$N_{\cdot j}$ Y = j のケース数。$N_{\cdot j} = \sum_{i=1}^{I} N_{ij}$

上の表記法は、(X, Y) の非欠損ペアに基づいています。したがって、J, N と $N_{\cdot j}$ は、予測フィールドごとに異なる可能性があります。

Pearson カイ 2 乗に基づいた P 値

Pearson カイ 2 乗は、観察結果と予測された度数の間の差分に関連する、X と Y の間の独立性の検定です。独立に対する帰無仮説での期待セル度数が、次の式によって推定されます。$\hat{N}_{ij} = N_{i\cdot}N_{\cdot j}/N$。帰無仮説のもとで、Pearson カイ 2 乗は漸近的にカイ 2 乗分布へ収束します。 χ^2_d自由度は d = (I-1)(J-1) です。

Pearson カイ 2 乗 X^2 の p 値は、次のようにして計算されます。p 値 = Prob(χ^2_d > X^2)。ここで、

$X^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(N_{ij} - \hat{N}_{ij}\right)^2/\hat{N}_{ij}$ です。

予測フィールドは、次のルールによりランク付けされます。

1. 予測フィールドを p 値の昇順でソートします
2. タイ（同順位）が発生した場合は、カイ 2 乗の降順でソートされます。
3. タイ（同順位）がそれでも発生する場合は、自由度 d の降順でソートされます。
4. それでもまだタイ（同順位）が発生する場合は、データ ファイルの順にソートされます。

尤度比カイ 2 乗に基づいた P 値

尤度比カイ 2 乗は、観察結果と予測された度数の間の比率に関連する、X と Y の間の独立性の検定です。独立に対する帰無仮説での期待セル度数が、次の式によって推定されます。$\hat{N}_{ij} = N_{i\cdot}N_{\cdot j}/N$。帰無仮説のもとで、Pearson カイ 2 乗は漸近的にカイ 2 乗分布へ収束します。 χ^2_d自由度は d = (I-1)(J-1) です。

尤度比 カイ 2 乗 G^2 の p 値は、次のようにして計算されます。p 値 = Prob(χ^2_d> G^2)。ここで、

$G^2 = 2\sum_{i=1}^{I}\sum_{j=1}^{J} G^2_{ij}$, with $G^2_{ij} = \begin{cases} N_{ij}\ln\left(N_{ij}/\hat{N}_{ij}\right) & N_{ij} > 0, \\ 0 & \text{else.} \end{cases}$

予想フィールドは、Pearson カイ 2 乗に基づいた p 値に対するルールと同じルールに従ってランク付けされます。

Cramer の V

Cramer の V は、Pearson カイ 2 乗に基づく 0 と 1 の間のアソシエーションの測定結果です。これは、次のように定義されます。

$V = \left(\frac{X^2}{N(\min\{I,J\}-1)}\right)^{1/2}$。

予測フィールドは、次のルールによりランク付けされます。

1. 予想フィールドが Cramer の V の降順にソートされます。
2. タイ（同順位）が発生した場合は、カイ 2 乗の降順でソートされます。
3. それでもまだタイ（同順位）が発生する場合は、データ ファイルの順にソートされます。

ラムダ

ラムダは、独立変数の値が従属変数の値を予測するために使用されるときに誤差内の比例減力を反映する、アソシエーションの測定値です。1 の値は、独立変数が従属変数を完全に予測することを示します。1 の値は、独立変数が従属変数の予測に何の役にも立たないことを示します。これは、次のように計算されます。

$\lambda(Y|X) = \frac{\sum_i \max_j(N_{ij}) - \max_j(N_{\cdot j})}{N - \max_j(N_{\cdot j})}$。

予測フィールドは、次のルールによりランク付けされます。

1. 予想フィールドがラムダの降順にソートされます。
2. タイ（同順位）が発生した場合は、I の昇順でソートされます。
3. それでもまだタイ（同順位）が発生する場合は、データ ファイルの順にソートされます。

すべて連続型の予測フィールド

すべての予測フィールドが連続型の場合は、F 統計量に基づく p 値が使用されます。この考えは各連続型予測フィールドに一方向の ANOVA F 検定を実行することであり、これで、Y のすべての異なるクラスが X と同じ平均値を持っているかどうかを検定します。

次の表記法が適用されます。

N_j	Y = j のケース数。
$\bar{x}_j$	対象クラス Y = j に対する予測フィールド X のサンプル平均。

s_j^2 対象クラス Y = j に対する予測フィールド X のサンプル分散。$s_j^2 = \sum_{i=1}^{N_j} (x_{ij} - \overline{x}_j)^2/(N_j - 1)$

$\overline{\overline{x}}$ 予測フィールド X の総平均。$\overline{\overline{x}} = \sum_{j=1}^{J} N_j \overline{x}_j / N$

上の表記法は、(X, Y) の非欠損ペアに基づいています。

F 統計量に基づく P 値

F 統計量に基づく p 値は、p 値 = Prob{F(J-1, N-J) > F} により計算されます。ここで、

$$F = \frac{\sum_{j=1}^{J} N_j \left(\overline{x}_j - \overline{\overline{x}}\right)^2 / (J-1)}{\sum_{j=1}^{J} (N_j - 1)\, s_j^2 / (N-J)},$$

さらに、F(J-1, N-J) は自由度 J-1 および N-J で F 分布に従うランダム変数です。予測フィールドの分母がゼロの場合、予測フィールドには p 値 = 0 を設定します。

予測フィールドは、次のルールによりランク付けされます。

1. 予測フィールドが p 値の昇順でソートされます。
2. タイ（同順位）が発生した場合は、F の降順でソートされます。
3. タイ（同順位）が引き続き発生した場合は、N の降順でソートされます。
4. それでもまだタイ（同順位）が発生する場合は、データ ファイルの順にソートされます。

型混合の予測フィールド

連続型の予測フィールドとカテゴリ型の予測フィールドの両方がある場合、連続型予測フィールドの基準は引き続き F 統計量に基づいた p 値であるのに対し、カテゴリ型予測フィールドに利用できる基準は、Pearson のカイ 2 乗に基づいた p 値または尤度比率カイ 2 乗に基づいた p 値に制限されます。これら p 値は比較可能であり、そのため予測フィールドのランク付けに使用できます。

予測フィールドは、次のルールによりランク付けされます。

1. 予測フィールドが p 値の昇順でソートされます。
2. タイ（同じ順位）になった場合は、すべてのカテゴリ型およびすべての連続型予測フィールドの中で別々にタイを解決するためのルールに従い、その後最初の予測フィールドのデータ ファイルの順に 2 つのグループ （カテゴリ型予測フィールドのグループと連続型予測フィールドのグループ） をソートします。

連続型対象

このセクションでは、次のシナリオに沿った連続型対象の予測フィールドのランク付けを説明します。

- すべての予測フィールドがカテゴリ型
- すべての予測フィールドが連続型
- カテゴリ型と連続型の予測フィールドが混在

すべての予測フィールドがカテゴリ型

すべての予測フィールドがカテゴリ型で対象が連続型の場合は、F 統計量に基づく p 値が使用されます。この考えは各カテゴリ型予測フィールドを因子として使用して連続型対象に一方向の ANOVA F 検定を実行することであり、これで、X のすべての異なるクラスが Y と同じ平均値を持っているかどうかを検定します。

次の表記法が適用されます。

X	I カテゴリ数で検討中のカテゴリ型予測フィールド。
Y	連続型対象変数。y_{ij} は、X = i の j 番目のケースに対する連続型対象の値を表します。
Ni	X = i のケース数。
$\overline{y}_i$	対象フィールドのカテゴリが X = i の対象 Y のサンプル平均。
$s(y)_i^2$	予測フィールドのカテゴリが X = i の対象 Y のサンプル分散。$s(y)i2 = \sum_{j=1}^{N_i} (y_{ij} - \overline{y}_i)^2/(N_i - 1)$
$\overline{\overline{y}}$	対象 Y の総平均。$\overline{\overline{y}} = \Sigma_{i=1}^{I} N_i \overline{y}_i / N$

上の表記法は、(X, Y) の非欠損ペアに基づいています。

F 統計量に基づく p 値は、p 値 = Prob{F(I-1, N-I) > F} により計算されます。ここで、

$$F = \frac{\sum_{i=1}^{I} N_i \left(\overline{y}_i - \overline{\overline{y}}\right)^2 / (I-1)}{\sum_{i=1}^{I} (N_i - 1)\, s(y)_i^2 / (N-I)}$$、

さらに、F(I-1, N-I) は自由度 I-1 および N-I で F 分布に従うランダム変数です。上記公式の分母が指定されたカテゴリ型予測フィールド X に対してゼロの場合、予測フィールドには p 値 = 0 を設定します。

予測フィールドは、次のルールによりランク付けされます。

1. 予測フィールドが p 値の昇順でソートされます。
2. タイ（同順位）が発生した場合は、F の降順でソートされます。

3. タイ（同順位）が引き続き発生した場合は、N の降順でソートされます。
4. それでもまだタイ（同順位）が発生する場合は、データ ファイルの順にソートされます。

すべて連続型の予測フィールド

すべての予測フィールドが連続型で対象も連続型の場合、p 値は、Pearson 相関係数 r に対する変換 t の、漸近 t 分布に基づきます。

次の表記法が適用されます。

X	検討中の連続型予測フィールド。
Y	連続型対象変数。
$\overline{x} = \Sigma_{i=1}^{N} x_i / N$	予測値変数 X のサンプル平均。
$\overline{y} = \Sigma_{i=1}^{N} y_i / N$	対象 Y のサンプル平均。
$s(x)^2$	予測値変数 X のサンプル分散。
$s(y)^2$	対象変数 Y のサンプル分散。

上の表記法は、(X, Y) の非欠損ペアに基づいています。

Pearson 相関係数 r は、次のとおりです。

$$r = \frac{\Sigma_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})/(N-1)}{\sqrt{s(x)^2 s(y)^2}}.$$

r に対する変換 t は、次のようにして指定されます。

$$t = r\sqrt{\frac{N-2}{1-r^2}}.$$

母集団の Pearson 相関係数 ρ が 0 という帰無仮説のもとで、p 値が次のように計算されます。

$$p\ ?= \begin{cases} 0 & \text{if } r^2 = 1, \\ 2\ \text{Prob}\{T > |t|\} & \text{else.} \end{cases}$$

T は、自由度が N-2 の t 分布に従うランダム変数です。Pearson 相関係数に基づいた p 値は、X と Y 間の線形関係の検定です。X と Y に線形関係がない場合、検定がその捕捉に失敗する可能性があります。

予測フィールドは、次のルールによりランク付けされます。

1. 予測フィールドが p 値の昇順でソートされます。
2. タイ（同順位）が発生した場合は、r^2 の降順でソートされます。
3. タイ（同順位）が引き続き発生した場合は、N の降順でソートされます。

4. それでもまだタイ（同順位）が発生する場合は、データ ファイルの順にソートされます。

型混合の予測フィールド

データセット内に連続型とカテゴリ型の予測フィールドが混在している場合、連続型予測フィールドの基準は引き続き変換からの p 値に基づき、カテゴリ型予測フィールドの基準は F 統計量からに基づきます。

予測フィールドは、次のルールによりランク付けされます。

1. 予測フィールドが p 値の昇順でソートされます。
2. タイ（同じ順位）になった場合は、すべてのカテゴリ型およびすべての連続型予測フィールドの中で別々にタイを解決するためのルールに従い、その後最初の予測フィールドのデータ ファイルの順に 2 つのグループ （カテゴリ型予測フィールドのグループと連続型予測フィールドのグループ） をソートします。

予測値の選択

予測フィールド リストの長さが事前に指定されなかった場合は、以下の公式で、リストの長さを決める自動アプローチが提供されます。

L_0 は、学習中の予測フィールドの総数とします。リスト L の長さは、次の式で決定できます。

$L = \left[\min\left(\max\left(30, 2\sqrt{L_0}\right), L_0\right)\right]$、

[x] は、x に最も近い整数です。次の表に、予測フィールド L_0 の総数の異なる値のリストの長さ L を示します。

L_0	L	L/L_0 (%)
10	10	100.00%
15	15	100.00%
20	20	100.00%
25	25	100.00%
30	30	100.00%
40	30	75.00%
50	30	60.00%
60	30	50.00%
100	30	30.00%
500	45	9.00%
1000	63	6.30%
1500	77	5.13%
2000	89	4.45%
5000	141	2.82%
10,000	200	2.00%

L_0	L	L/L_0(%)
20,000	283	1.42%
50,000	447	0.89%

生成されたモデル

フィールド選択の生成モデルは、予測フィールドやその他の派生フィールドが追加されないという点で、他のほとんどの生成されたモデルと異なります。その代わり、生成されたモデルの設定に基づいてデータ ストリームから望ましくないフィールドを取り除くためのフィルタとして動作します。

ストリームからフィルタリングされるフィールドのセットは、次の基準のいずれかで制御されます。

- フィールド重要度のカテゴリ（**［重要度 高］**、**［境界］**、または **［重要度 低］**）。選択したカテゴリのいずれかが割り当てられたフィールドは保存され、その他は除去されます。
- 上位 k 個のフィールド： 最高の重要度の k 個のフィールドが保存され、その他は除去されます。
- 重要度の値： 指定した値よりも高い重要度のフィールドが保存され、その他は除去されます。
- 手動による選択： ユーザーが、保存または除去する特定のフィールドを選択できます。

章
18

GENLIN アルゴリズム

一般化線型モデル (GZLM) は通常、異なるタイプのデータの分析ツールに使用されます。一般化線型モデルは、正規分布したレスポンスの線型回帰、バイナリ データのためのロジスティック モデル、計数データのための対数線型モデルなどの広く使用される統計モデルだけではなく、非常に一般的なモデルの定式を通じて多くの有用な統計モデルも対象とします。

[一般化線型モデル]

一般化線型モデルは、最初に Nelder および Wedderburn によって紹介され (1972 年)、後に McCullagh および Nelder によって拡張されました (1989 年)。以下の考察は、彼らの業績に基づいています。

表記

このセクションでは、特に明記しない限り次の表記を使用します。

n	データセット内の完結したケースの数。これは整数であり、$n \geq 1$ です。
p	モデル内のパラメータの数 (もしあれば定数項を含む)。これは整数であり、$p \geq 1$ です。
p_x	デザイン行列内の非冗長列の数。これは整数であり、$p_x \geq 1$ です。
y	n × 1 の従属変数ベクトル。行がケースです。
r	二項分布のためのイベントの n × 1 ベクトルで、通常は「成功」数を表します。すべての要素は負でない整数です。
m	二項分布のための繰り返し回数の n × 1 ベクトル。すべての要素は正の整数で、$m_i \geq r_i$, i=1,...,n です。
μ	従属変数の期待値の n × 1 ベクトル。
η	線型予測フィールドの n × 1 のベクトル。
X	n × p のデザイン行列。行はケース、列はパラメータを表します。i 番目の行は $x_i = (x_{i1}, \ldots, x_{ip})^T$, i=1,...,n であり、モデルに定数項があれば $x_{i1} = 1$ となります。
0	スケール オフセットの n × 1 のベクトル。この変数は、従属変数 (**y**) にも、予想値変数 (**X**) の 1 つにもなり得ません。
β	不明パラメータの p × 1 ベクトル。β 内の最初のエレメント、ある場合には定数項です。
ω	スケールの重みの n × 1 ベクトル。エレメントが 0 以下または欠損している場合、対応するケースは使用されません。

f	度数カウントの n × 1 ベクトル。整数でないエレメントは、値を最も近い整数へ丸められて処理されます。値が 0.5 より小さいか欠損値の場合、対応するケースは使用されません。
N	効果的なサンプル サイズ。$N = \sum_{i=1}^{n} f_i$.度数カウント変数 f が使用されない場合は、N = n です。

モデル

予想値変数 X 付きの y の GZLM の形式は次のとおりです。

$$\eta = g(\mathsf{E}(\mathbf{y})) = \mathbf{X}\beta + \mathbf{O}, \quad \mathbf{y} \sim F$$

ここで、η は線型予想値、O は各観測に対して 1 の一定係数を使用するオフセット変数、g(.) は y、$E(\mathbf{y}) = \mu$ の平均がどのようにして線型予測値の η に関係付けられるかを述べる単調で微分可能なリンク関数、F はレスポンス確率の分布です。適切な確率分布とリンク関数のさまざまな組み合わせを選択することで、異なるモデルが作成されます。

さらにGZLM もまた、y_i が i=1、…、n に対して独立していると想定します。そこで各観察結果に対し、モデルは次のようになります。

$$\eta_i = g(\mu_i) = x_i^{\mathsf{T}}\beta + o_i, \quad y_i \sim F$$

メモ

- X は、スケール変数（共変量）、カテゴリ変数（因子）、および交互作用のどのような組み合わせでもかまいません。X のパラメータ化は、GLM プロシージャ内と同じです。データ内で発生する各因子効果レベルに対する別々のパラメータがあるようなパラメータ化が多すぎるモデルの使用が原因で、デザイン行列 X の列はしばしば従属的です。データ内のスケール変数間で共線性も発生する可能性があります。デザイン行列内で従属関係を確立するために、$\mathbf{X}^{\mathsf{T}}\mathbf{\Psi}\mathbf{X}$ の列はスイープ演算子を使用して検査されます。ここで、$\mathbf{\Psi} = \mathsf{diag}(f_1\omega_1, \ldots f_n\omega_n)?$ です。ある列が前の列に従属していることが判明した場合、対応するパラメータは冗長であるとして処理されます。冗長パラメータのソリューションは、ゼロで修正されます。
- y が male/female（男性/女性）または 1/2 のようなキャラクタまたは数値のバイナリ従属変数の場合、その値は 0 と 1 へ変換され、通常は成功またはその他の正の結果を表します。このドキュメントでは、成功の確率をモデリングしているものと想定します。またこのドキュメントでは、y が 0/1 の値に変換されており、常に成功の確率をモデル化する、つまり Prob(y = 1) であると想定しています。当初のどの値が 0 または 1 に変換されるかは、使用する参照カテゴリに左右されます。参照カテゴリが最後の値の場合、最初のカテゴリが成功を表し、その確率をモデリングします。たとえば、参照カテゴリが最後の値の場合、male/female の “male”、1/2 の 2 が最後の値（辞書では male が female より後にある）であり、0 に変換されます。“female” および 1 は、その確率をモデル化するのでそれぞれ 1 に変換されます。ただし、代わりに “male” および 2 の確率のモデル化を変更する 1 つの方法は、最初の値として参照カテゴリを指定

することです。最初の 2 進数形式が 0/1 で参照カテゴリが最後の値である場合は、0 が 1 に、1 が 0 に変換されることに注意してください。

- 成功の回数（または 1 の数）を表す **r** と試行回数を表す **m** が二項分布に使用されている場合、レスポンスは二項比率の **y = r/m** です。

確率分布

GZLM は通常、分布の中の指数ファミリーのフレームワーク内で公式化されます。指数系のレスポンス Y の確率密度関数は、次のように提示できます。

$$f(y) = \exp\left\{\frac{y\theta - b(\theta)}{\phi/\omega} + c(y, \phi/\omega)\right\}$$

ここで、θ は標準的な（自然の）パラメータ、ϕ は y の分散に関連するスケール パラメータ、ω はケースごとに変化する既知の事前の重みです。b(θ) および c(y, ϕ/ω) の異なる形式で、特定の分散を指定します。実際、指数系には、連続型と離散型（度数、2 進数および比率）の両方の結果をモデリングできるようにする表記法が提供されています。連続したものを含めていくつかは利用可能です。正規、逆ガウス、ガンマなど。不連続なもは負の 2 項、ポアゾン、2 項式 です。

y の平均と分散は、次のように表現できます。

$$E(y) = b'(\theta) = \mu$$

$$Var(y) = b''(\theta)\frac{\phi}{\omega} = V(\mu)\frac{\phi}{\omega}$$

ここで、$b'(\theta)$ および $b''(\theta)$ は θ に関するそれぞれ最初と 2 番目の b の導関数を指し、$V(\mu)$ は μ の関数である分散関数です。

GZLM では、y の分布は、標準的なパラメータ（θ）の代わりに平均（μ）およびスケール パラメータ（ϕ）の観点からパラメータ化されます。次の表に、y の分布、y の対応する範囲、分散関数（V(μ)）、y の分散（Var(y)）および分散関数の最初の導関数（$V'(\mu)$）（これは後に使用）を一覧表示します。

テーブル 18-1
分布、レスポンスの範囲と分散、分散関数と最初の導関数

分布	y の範囲	V(μ)	Var(y)	V'(μ)
Normal	$(-\infty, \infty)$	1	ϕ	0
逆ガウス分布	$(0, \infty)$	μ^3	$\phi\mu^3$	$3\mu^2$
Gamma	$(0, \infty)$	μ^2	$\phi\mu^2$	2μ
負の二項分布	0(1)∞	$\mu+k\mu^2$	$\mu+k\mu^2$	$1+2k\mu$
Poisson	0(1)∞	μ	μ	1
二項 (m)	0(1)m/m	$\mu(1-\mu)$	$\mu(1-\mu)/m$	$1-2\mu$

メモ

- 0(1)z は、範囲が 1 ずつ増加する 0 から z である、つまり 0、1、2、…、z ということです。
- 二項分布の場合、二項繰り返し回数変数の m は重み変数 ω の一部であると見なされます。
- 重み変数 ω が存在する場合は、ϕ が ϕ/ω. に置き換えられます。
- 負の二項分布の場合、補助パラメータ (k) はユーザー指定。k = 0 の場合、負の二項はポアソン分布に縮小されます。k = 1 の場合、負の二項は幾何分布です。

尺度パラメータの取り扱い :連続型の分布 のための V(μ) および Var(y) の式には、分散と平均 (Var(y)および μ) の関係を測定するのに使用できる尺度パラメータ ϕ が含まれています。通常は不明であるため、尺度パラメータを適合させるのに 3 つの方法があります。

1. 最大尤度方法により結合された β と共に推定できます。
2. 固定長の正の値に設定できます。
3. 導関数または Pearson カイ 2 乗で指定できます。 詳細は、 p.175 適合度統計 を参照してください。

一方、離散型分布にはこの特別なパラメータがありません (理論的に 1 に等しい)。そのため、y の分散が実際は名目上の分散と等しくならない可能性があります (負の二項分布には補助パラメータ k があるため、特にポアソン分布と二項分布の場合)。この状況に対処する単純な方法は、離散型分布の y の分散も同様に尺度パラメータを持てるようにすることですが、連続型分布と異なり、ML 方法で推定できません。そのため離散型分布の場合は、ϕ の値の取得のために 2 つの方法があります。

1. 導関数または Pearson カイ 2 乗で指定できます。
2. 固定長の正の値に設定できます。

指定された分布のレスポンスの範囲にデータが必ず適合するように、次のルールに従います。

- ガンマまたは逆ガウス分布の場合、**y** の値が実数で 0 より大きい必要があります。**y** の値が 0 以下または欠損している場合は、対応するケースが使用されません。
- 負の二項分布およびポアソン分布の場合、**y** の値が整数で非負数である必要があります。**y** の値が負数であったり、 0 より小さいか欠損している場合は、対応するケースが使用されません。
- 二項分布でレスポンスが単一変数の形式の場合、**y** はただ 2 つの異なる値である必要があります。**y** が 2 つより多い離散型の値を持っている場合、アルゴリズムはエラーとなって終了します。
- 二項分布でレスポンスがイベント数/試行数として示される 2 つの変数の比率の形式の場合は、**r** の値 (イベント数) が非負数の整数でなければならず、**m** の値 (試行数) は正の整数で $m_i \geq r_i$, $\forall$ i である必要があります。**r** の値が整数でなかったり、0 より小さいか欠損している場合は、対応するケースが使用されません。**m** の値が整数でなかったり、0 以下か、対応する **r** の値より小さいか、または欠損している場合は、対応するケースが使用されません。

ML方法は 継続型分布および Tweedie 分布のための β およびおそらく ϕ。パラメータ推定のための目標の関数として使用される対数尤度関数（ℓ_k）のカーネルと、完結した対数尤度関数（ℓ）は、次の表に分布ごとに表示されています。ℓ または ℓ_k を使用してもパラメータの推定には影響ありませんが、何を選択するかによって情報基準の計算に影響があります。 詳細は、 p. 175 適合度統計 を参照してください。

テーブル 18-2
確率分布のための対数尤度関数

分布	ℓ_k および ℓ
Normal	$\ell_k = \sum_{i=1}^{n} -\frac{f_i}{2}\left\{\frac{\omega_i(y_i-\mu_i)^2}{\phi} + \ln\left(\frac{\phi}{\omega_i}\right)\right\}$ $\ell = \ell_k + \sum_{i=1}^{n} -\frac{f_i}{2}\{\ln(2\pi)\}$
逆ガウス分布	$\ell_k = \sum_{i=1}^{n} -\frac{f_i}{2}\left\{\frac{\omega_i(y_i-\mu_i)^2}{\phi y_i \mu_i^2} + \ln\left(\frac{\phi y_i^3}{\omega_i}\right)\right\}$ $\ell = \ell_k + \sum_{i=1}^{n} -\frac{f_i}{2}\{\ln(2\pi)\}$
Gamma	$\ell_k = \sum_{i=1}^{n} f_i\left\{\frac{\omega_i}{\phi}\ln\left(\frac{\omega_i y_i}{\phi\mu_i}\right) - \frac{\omega_i y_i}{\phi\mu_i} - \ln\left(\Gamma\left(\frac{\omega_i}{\phi}\right)\right)\right\}$ $\ell = \ell_k + \sum_{i=1}^{n} f_i\{-\ln(y_i)\}$
負の二項分布	$\ell_k = \sum_{i=1}^{n} f_i\frac{\omega_i}{\phi}\{y_i\ln(k\mu_i) - (y_i+1/k)\ln(1+k\mu_i) + \ln(\Gamma(y_i+1/k)) - \ln(\Gamma(1/k))\}$ $\ell = \ell_k + \sum_{i=1}^{n} f_i\frac{\omega_i}{\phi}\{-\ln(\Gamma(y_i+1))\}$
Poisson	$\ell_k = \sum_{i=1}^{n} f_i\frac{\omega_i}{\phi}\{y_i\ln(\mu_i) - \mu_i\}$ $\ell = \ell_k + \sum_{i=1}^{n} f_i\frac{\omega_i}{\phi}\{-\ln(y_i!)\}$
二項（m）	$\ell_k = \sum_{i=1}^{n} f_i\frac{\omega_i^*}{\phi}\{y_i\ln(\mu_i) + (1-y_i)\ln(1-\mu_i)\}$ $\ell = \ell_k + \sum_{i=1}^{n} f_i\frac{\omega_i}{\phi}\left\{\ln\binom{m_i}{r_i}\right\}$、ここで、$\binom{m_i}{r_i} = \frac{m_i!}{r_i!(m_i-r_i)!}$

負数二項分布またはポアソン分布および y = 0 または 1に対して 二項分布の時は、別々の値の対数尤度が指定されます。負の二項およびポアソン に対して $y_i = 0$、また二項に対して 0/1 の時に、$\ell_{k,i}$ を個々のケース i の対数尤度とします。i の完全な対数尤度は i の対数尤度のカーネルと等しく、$\ell_i=\ell_{k,i}$ となります。

分布	$\ell_{k,i}$
負の二項分布	$\ell_{k,i} = -f_i \frac{\omega_i}{\phi} \frac{\ln(1+k\mu_i)}{k}$ if $y_i = 0$
Poisson	$\ell_{k,i} = -f_i \frac{\omega_i}{\phi} \mu_i$ if $y_i = 0$
二項 (m)	$\ell_{k,i} = \begin{cases} f_i \frac{\omega_i}{\phi} \ln(1-\mu_i) & \text{if } y_i = 0 \\ f_i \frac{\omega_i}{\phi} \ln(\mu_i) & \text{if } y_i = 1 \end{cases}$

- Γ(z) はガンマ関数であり、ln(Γ(z)) は z で評価される対数ガンマ関数（ガンマ関数の対数）です。
- 負の二項分布については、尺度パラメータが引き続き ℓ_k に含まれています。これは柔軟性を持たせるためですが、通常は 1 に設定されます。
- 二項分布（r/m）については、尺度重み変数が ℓ_k 内で $\omega_i^* = \omega_i m_i$ になります。つまり、二項試行変数 m が重みの一部とみなされます。ただし、ℓ の特別な意味での尺度の重みは、引き続き ω_i です。

リンク関数

次の表に、形式、逆の形式、$\hat{\mu}$ の範囲、および各リンク関数の最初および 2 番目の導関数を一覧表示します。

テーブル 18-3
リンク関数名、形式、リンク関数の逆数、および予測平均値の範囲

リンク関数	$\eta=g(\mu)$	逆 $\mu=g^{-1}(\eta)$	$\hat{\mu}$ の範囲
恒等式	μ	η	$\hat{\mu} \in R$
ログ	$\ln(\mu)$	$\exp(\eta)$	$\hat{\mu} \geq 0$
Logit	$\ln\left(\frac{\mu}{1-\mu}\right)$	$\frac{\exp(\eta)}{1+\exp(\eta)}$	$\hat{\mu} \in [0, 1]$
Probit	$\Phi^{-1}(\mu)$, where $\Phi(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} e^{-z^2/2} dz$	$\Phi(\eta)$	$\hat{\mu} \in [0, 1]$
補ログ・マイナス・ログ	$\ln(-(\ln(1-\mu))$	$1-\exp(-\exp(\eta))$	$\hat{\mu} \in [0, 1]$
Power(α) $\begin{cases} \alpha \neq 0 \\ \alpha = 0 \end{cases}$	$\begin{cases} \mu^\alpha \\ \ln(\mu) \end{cases}$	$\begin{cases} \eta^{1/\alpha} \\ \exp(\eta) \end{cases}$	$\begin{cases} \hat{\mu} \in R & \alpha \text{ or} 1/\alpha \text{ が 奇数の 整数の場合} \\ \hat{\mu} \geq 0 & \text{そうでない場合} \end{cases}$
対数-補数	$\ln(1-\mu)$	$1-\exp(\eta)$	$\hat{\mu} \leq 1$
負ログ・マイナス・ログ	$-\ln(-\ln(\mu))$	$\exp(-\exp(-\eta))$	$\hat{\mu} \in [0, 1]$
負の二項分布	$\ln\left(\frac{\mu}{\mu+\frac{1}{k}}\right)$	$\frac{\exp(\eta)}{k(1-\exp(\eta))}$	$\hat{\mu} \geq 0$
オッズ power(α) $\begin{cases} \alpha \neq 0 \\ \alpha = 0 \end{cases}$	$\begin{cases} \frac{(\mu/(1-\mu))^\alpha - 1}{\alpha} \\ \ln\left(\frac{\mu}{1-\mu}\right) \end{cases}$	$\begin{cases} \frac{(1+\alpha\eta)^{1/\alpha}}{1+(1+\alpha\eta)^{1/\alpha}} \\ \frac{\exp(\eta)}{1+\exp(\eta)} \end{cases}$	$\hat{\mu} \in [0, 1]$

注 : べき乗のリンク関数で、$|\alpha| < 2.2e\text{-}16$ の場合、α は 0 として処理されます。

テーブル 18-4
リンク関数の最初と 2 番目の導関数

リンク関数	最初の導関数 $g'(\mu) = \frac{\partial \eta}{\partial \mu} = \Delta$	2 番目の導関数 $g''(\mu) = \frac{\partial^2 \eta}{\partial \mu^2}$
恒等式	1	0
ログ	$\frac{1}{\mu}$	$-\Delta^2$
Logit	$\frac{1}{\mu(1-\mu)}$	$\Delta^2(2\mu - 1)$
Probit	$\frac{1}{\phi\left(\Phi^{-1}(\mu)\right)}$, where $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$	$\Delta^2 \Phi^{-1}(\mu)$
補ログ・マイナス・ログ	$\frac{1}{(\mu-1)\ln(1-\mu)}$	$-\Delta^2(1 + \ln(1-\mu))$
Power(α) $\begin{cases} \alpha \neq 0 \\ \alpha = 0 \end{cases}$	$\begin{cases} \alpha\mu^{\alpha-1} \\ \frac{1}{\mu} \end{cases}$	$\begin{cases} \Delta\frac{\alpha-1}{\mu} \\ -\Delta^2 \end{cases}$
対数-補数	$\frac{-1}{1-\mu}$	$-\Delta^2$
負ログ・マイナス・ログ	$\frac{-1}{\mu \ln(\mu)}$	$\Delta^2(1 + \ln(\mu))$
負の二項分布	$\frac{1}{\mu + k\mu^2}$	$-\Delta^2(1 + 2k\mu)$
オッズ power(α) $\begin{cases} \alpha \neq 0 \\ \alpha = 0 \end{cases}$	$\begin{cases} \frac{\mu^{\alpha-1}}{(1-\mu)^{\alpha+1}} \\ \frac{1}{\mu(1-\mu)} \end{cases}$	$\begin{cases} \Delta\left(\frac{\alpha-1}{\mu} + \frac{\alpha+1}{1-\mu}\right) \\ \Delta^2(2\mu - 1) \end{cases}$

標準的なパラメータが線型予測値と等しい、$\theta = \eta$ の場合、リンク関数は**標準的リンク関数** と呼ばれます。標準的なリンクからモデルの望ましい統計的プロパティが導かれたとしても、特に小さなサンプルの場合、通常は、モデル内のシステマティックな効果がそのリンクによって定められたスケールに何故付加されるかの先見的な理由がありません。確率分布のための標準リンク関数を次の表に示します。

テーブル 18-5
確率分布のための標準およびデフォルトのリンク関数

分布	標準適なリンク関数
Normal	恒等式
逆ガウス分布	Power(-2)
Gamma	Power(-1)
負の二項分布	負の二項分布
Poisson	ログ
Binomial	Logit

推定

特定のモデルを選択すると、パラメータを推定し、推定値の精度を評価することが要求されます。

パラメータの推定

パラメータは、観察されたデータから対数の尤度関数（または対数尤度関数のカーネル）を最大にすることで、推定されます。各パラメータに関して **s** を対数尤度の最初の導関数（変化）ベクトルとし、次を解決しようとします。

$$\mathbf{s} = \left[\frac{\partial \ell}{\partial \beta}\right]_{p\times 1} = 0$$

通常は、恒等式リンク関数付きの正規分布を除いて閉形式のソリューションがないので、推定は反復するプロセスを経て数値的に取得されます。Newton-Raphson と Fisher の両方または片方のスコアリング アルゴリズムが使用され、それは、対数尤度の最初の導関数の線型 Taylor 系列近似値に基づいています。

最初の導関数

尺度パラメータ ϕ が ML 方法によって推定されない場合、**s** は次の形式の p×1 ベクトルです。

$$\mathbf{s} = \sum_{i=1}^{n} \frac{f_i \omega_i (y_i - \mu_i)}{\phi V(\mu_i) g'(\mu_i)} \cdot x_i = \frac{1}{\phi} \sum_{i=1}^{n} \frac{f_i \omega_i (y_i - \mu_i)}{V(\mu_i) g'(\mu_i)} \cdot x_i$$

ここで $\mu_i, V(\mu_i)$ and $g'(\mu_i)$ は、次に定義されます。 テーブル 18-3“リンク関数名、形式、リンク関数の逆数、および予測平均値の範囲” p. 164 ， テーブル 18-1“分布、レスポンスの範囲と分散、分散関数と最初の導関数” p. 161 そして テーブル 18-4“リンク関数の最初と 2 番目の導関数” p. 165 となります。

尺度パラメータ ϕ が ML 方法で推定される場合は、ϕ が 0 より大きいことが必要なため、ln(ϕ) を検索することで処理されます。

τ = ln(ϕ) so ϕ = exp(τ) にすると、**s** は次の形式で (p+1)×1 ベクトルとなります。

$$\mathbf{s} = \begin{bmatrix} \frac{\partial \ell}{\partial \beta} \\ \frac{\partial \ell}{\partial \tau} \end{bmatrix}_{(p+1)\times 1} = \begin{bmatrix} \frac{1}{\exp(\tau)} \sum_{i=1}^{n} \frac{f_i \omega_i (y_i - \mu_i)}{V(\mu_i) g'(\mu_i)} \cdot x_i \\ \frac{\partial \ell}{\partial \tau} \end{bmatrix}$$

ここで $\partial\ell/\partial\beta$ は ϕ が(τ) と置き換えられるだけで、上記と同じです。$\partial\ell/\partial\tau$ は次のように分布の種類に応じて異なります。

テーブル 18-6
確率分布の最初の導関数 w.r.t. 尺度パラメータ

分布	$\frac{\partial \ell}{\partial \tau}$
Normal	$\sum_{i=1}^{n} \frac{f_i}{2}\left\{\frac{\omega_i(y_i-\mu_i)^2}{\exp(\tau)}-1\right\}$
逆ガウス分布	$\sum_{i=1}^{n} \frac{f_i}{2}\left\{\frac{\omega_i(y_i-\mu_i)^2}{\exp(\tau)\, y_i\mu_i^2}-1\right\}$
Gamma	$\sum_{i=1}^{n} -\frac{f_i\omega_i}{\exp(\tau)}\left\{\ln\left(\frac{\omega_i y_i}{\exp(\tau)\,\mu_i}\right)+\left(1-\frac{y_i}{\mu_i}\right)-\psi\left(\frac{\omega_i}{\exp(\tau)}\right)\right\}$

注 ： $\psi(z)$ はディガンンマ関数であり、z で評価されるガンマ関数の対数の導関数です。つまり、$\psi(z)=\frac{\partial \ln(\Gamma(z))}{\partial z}=\frac{\Gamma'(z)}{\Gamma(z)}$ です。

これまでに説明したように、古典的な線型回帰モデルである恒等式リンク関数による正規分布については、β と τ の両方に対して閉形式のソリューションがあるので、反復プロセスは必要ありません。GLM プロシージャ内で SWEEP 操作を適用した後の β のソリューションは、次のとおりです。

$$\hat{\beta}=\left(\sum_{i=1}^{n} f_i\omega_i \mathbf{x}_i^{\mathsf{T}}\mathbf{x}_i\right)^{-}\left(\sum_{i=1}^{n} f_i\omega_i \mathbf{x}_i^{\mathsf{T}}(y_i-o_i)\right)=\left(\mathbf{X}^{\mathsf{T}}\Psi\mathbf{X}\right)^{-}\left(\mathbf{X}^{\mathsf{T}}\Psi(\mathbf{y}-\mathbf{o})\right),$$

ここで、$\Psi=\mathbf{diag}(f_1\omega_1,\ldots f_n\omega_n)$ および $(\mathbf{Z})^{-}$ は、行列 $\mathbf{Z}$ の一般化された逆数です。尺度パラメータ ϕ も ML 方法で推定される場合、τ の推定は次のようになります。

$$\hat{\tau}=\ln\left(\hat{\phi}\right)=\ln\left(\frac{1}{N}\sum_{i=1}^{n} f_i\omega_i\left(y_i-\mathbf{x}_i^{\mathsf{T}}\hat{\beta}-o_i\right)^2\right).$$

2 番目の導関数

H を 2 番目の導関数 (Hessian) 行列とします。尺度パラメータ が ML 方法によって推定されない場合、H は次の形式の p×p 行列です。

$$\mathbf{H}=\left[\frac{\partial^2\ell}{\partial\beta\partial\beta^{\mathsf{T}}}\right]_{p\times p}=-\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}$$

ここで、W は n×n の対角行列です。どのアルゴリズムを使用するかによって、W の定義は2つあります。Fisher スコアリングの $\mathbf{W}_{\mathrm{e}}$ と Newton-Raphson の $\mathbf{W}_{\mathrm{o}}$ です。$\mathbf{W}_{\mathrm{e}}$ の i 番目の対角エレメントは、次のとおりです。

$$w_{e,i}=\frac{f_i\omega_i}{\phi}\cdot\frac{1}{V(\mu_i)\left(g'(\mu_i)\right)^2},$$

また、$\mathbf{W}_{\mathrm{o}}$ の i 番目の対角要素は、次のとおりです。

$$w_{o,i} = w_{e,i} + \frac{f_i \omega_i}{\phi}(y_i - \mu_i) \cdot \frac{V(\mu_i)\, g''(\mu_i) + V'(\mu_i)\, g'(\mu_i)}{(V(\mu_i))^2 (g'(\mu_i))^3},$$

ここで、$V'(\mu_i)$ および $g''(\mu_i)$ は次のように定義されています。テーブル 18-1 “分布、レスポンスの範囲と分散、分散関数と最初の導関数” p. 161 そして テーブル 18-4 “リンク関数の最初と 2 番目の導関数” p. 165 となります。$\mathbf{W}_o$ の期待値は $\mathbf{W}_e$ であり、標準的なリンクが指定された分布に使用されると $\mathbf{W}_o = \mathbf{W}_e$ であることに注意してください。

尺度パラメータが ML 方法により推定されない場合、H は次の形式の (p+1)×(p+1) 行列になります。

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \ell}{\partial \beta \partial \beta^\mathsf{T}} & \frac{\partial^2 \ell}{\partial \beta \partial \tau} \\ \frac{\partial^2 \ell}{\partial \tau \partial \beta^\mathsf{T}} & \frac{\partial^2 \ell}{\partial \tau^2} \end{bmatrix}_{(p+1)\times(p+1)}$$

ここで、$\partial^2\ell/\partial\beta\partial\tau$ は p×1 ベクトル、$\partial^2\ell/\partial\tau\partial\beta^\mathsf{T}$ は 1×p ベクトルであり、$\partial^2\ell/\partial\beta\partial\tau$ の行列入替です。3 つのすべての連続型分布に対しては、次のようになります。

$$\frac{\partial^2 \ell}{\partial \beta \partial \tau} = \sum_{i=1}^{n} -\frac{f_i \omega_i (y_i - \mu_i)}{\exp(\tau)\, V(\mu_i)\, g'(\mu_i)} \cdot \mathbf{x}_i = -\frac{\partial \ell}{\partial \beta}.$$

$\partial^2\ell/\partial\tau^2$ の形式は、次の表にあります。

テーブル 18-7
確率分布の 2 番目の導関数 w.r.t. スケール パラメータ

分布	$\frac{\partial^2 \ell}{\partial \tau^2}$
Normal	$\sum_{i=1}^{n} -\frac{f_i \omega_i}{2\exp(\tau)}(y_i - \mu_i)^2$
逆ガウス分布	$\sum_{i=1}^{n} -\frac{f_i \omega_i}{2\exp(\tau)\, y_i \mu_i^2}(y_i - \mu_i)^2$
Gamma	$\sum_{i=1}^{n} \frac{f_i \omega_i}{\exp(\tau)} \left\{ \ln\left(\frac{\omega_i y_i}{\exp(\tau)\,\mu_i}\right) + \left(2 - \frac{y_i}{\mu_i}\right) - \psi\left(\frac{\omega_i}{\exp(\tau)}\right) - \frac{\omega_i}{\exp(\tau)} \psi'\left(\frac{\omega_i}{\exp(\tau)}\right) \right\}$

注 : $\psi'(z)$ はトリガンマ関数であり、z で評価される $\psi(z)$ の導関数です。

反復回数

β (ϕ を含んでいる可能性があります) のためのソリューションを検出する反復プロセスは、Newton-Raphson (すべての反復に対して)、Fisher スコアリング (すべての反復に対して)、またはその混合型の方法に基づいています。混合型の方法は、指定された反復回数だけ Fisher スコアリングを適用してから、Newton-Raphson ステップ

へ切り換わります。Newton-Raphson は、初期値がソリューションに近いとうまくいきますが、混合型は、初期値が適切でない場合にアルゴリズムの堅牢さを向上させるために使用できます。向上した堅牢さは別として、Fisher スコアリングのほうが、Hessian 行列が単純な形式であるため高速です。

次の標記を反復プロセスに適用します。

I	完全乖離と疑似完全乖離を確認するための反復の開始。0 または正の整数でなければなりません。値が 0 の場合、この基準は使用されません。
J	段階 2 分法の最大の段階数。正の整数でなければなりません。
K	Fisher スコアリングを使用する最初の反復数で、その後 Newton-Raphson へ切り換わります。0 または正の整数でなければなりません。値が 0 の場合はすべての反復に Newton-Raphson を使用し、M 以上の場合はすべての反復に Fisher スコアリングを使用します。
M	反復の最大回数。負でない整数でなければなりません。値が 0 の場合は、最初のパラメータ値が最終的な推定値になります。
$\epsilon_\ell, \epsilon_\beta, \epsilon_{\mathbf{H}}$	3 種類の収束基準のための許容レベル。
Abs	0/1 の 2 進数変数。絶対的な変更が収束基準に使用される場合は Abs = 1、相対的な変更が使用される場合は Abs = 0。

また、反復プロセスの概略は次のとおりです。

1. 3 つの収束基準のそれぞれに対して、I、J、K、M、ϵ_ℓ?ϵ_β、$\epsilon_{\mathbf{H}}$ および Abs の値を入力します。
2. $\beta^{(0)}$ について初期値を計算し（下を参照）、その後対数尤度 $\ell^{(0)}$、傾斜ベクトル $\mathbf{s}^{(0)}$、および Hessian 行列 $\mathbf{H}^{(0)}$ を $\beta^{(0)}$ に基づいて計算します。
3. ξ=1 とします。
4. i 番目の反復の推定値を計算します。

   $\beta^{(i)} = \beta^{(i-1)} - \xi\left(\mathbf{H}^{(i-1)}\right)^{-}\mathbf{s}^{(i-1)}$、ここで $(\mathbf{H})^{-}$ は H の一般化された逆数です。その後、$\beta^{(i)}$ に基づいて対数-尤度を計算します。
5. $\ell^{(i)} < \ell^{(i-1)}$ の場合は、段階 2 分方法を使用します。ξ を半分減らして、ステップ (4) を繰り返します。.ξ の値のセットは以下のとおりです。 {0.5 j :j = 0, …, J - 1}.J に到達されても対数尤度が改善されない場合は、警告メッセージを発行し、停止します。
6. $\beta^{(i)}$ に基づいて、傾斜ベクトル $\mathbf{s}^{(i)}$ と Hessian 行列 $\mathbf{H}^{(i)}$ を計算します。i ≤ K の場合は $\mathbf{H}^{(i)}$ を計算するために $\mathbf{W}_e$ が使用され、i>K の場合は $\mathbf{H}^{(i)}$ を計算するために $\mathbf{W}_o$ が使用されることに注意してください。
7. データの完全乖離または疑似完全乖離が確立されているかどうか（下を参照）、分布が二項分布で現在の反復が i ≥ I かどうかを確認します。完全乖離または疑似完全乖離のどちらかが検出された場合は、警告メッセージを発行し、停止します。
8. 3 つのすべての収束基準（下を参照）が満たされたかどうかを確認します。満たされないのに M に達した場合は、警告メッセージを発行し、停止します。

9. 3 つのすべての収束基準が満たされている場合は、分布が二項分布で i<I の場合にデータの完全乖離または疑似完全乖離が確立されたかどうかを確認します（完全乖離または疑似完全乖離の確認がまだ開始されていなかったため）。完全乖離または疑似完全乖離のどちらかが検出された場合は、警告メッセージを発行し、停止します。そうでない場合は、停止します（二項分布のプロセスは正常に収束）。二項分布以外の分布のための 3 つの収束基準すべてが満たされた場合も、停止します （二項分布以外の分布プロセスは正常に収束）。推定値の最終ベクトルは、$\hat{\beta}$（および$\hat{\tau}$の$\hat{\Psi}$）で示されます。そうでない場合は、ステップ (3). に戻ります。

初期値

初期値は次のように計算されます。

1. 二項分布の場合は $\tilde{\mu}_i = (y_i m_i + 0.5)/(m_i + 1)$（$y_i$ は比例値または 0/1 の値も可）、二項分布以外の分布の場合は $\tilde{\mu}_i = y_i$ に初期適合値を設定します。これらから $\tilde{\eta}_i = g(\tilde{\mu}_i), g'(\tilde{\mu}_i)$ および $V(\tilde{\mu}_i)$ を導きます。$\tilde{\eta}_i$ が未定義になる場合は、$\tilde{\eta}_i = 1$ を設定します。

2. 重み行列の $\tilde{W}_e$ を対角要素 $\tilde{w}_{ei} = \frac{f_i \omega_i}{\phi} \cdot \frac{1}{V(\tilde{\mu}_i)(g'(\tilde{\mu}_i))^2}$ で計算します。ここで、ϕ は 1 に設定されるか固定正数値です。$\tilde{w}_{ei}$ の分母が 0 になる場合は、$\tilde{w}_{ei}$= 0 を設定します。

3. 調整済みの従属変数 z を、二項分布には i 番目の観察 $z_i = (\tilde{\eta}_i - o_i) + (y_i - \tilde{\mu}_i) g'(\tilde{\mu}_i)$ で、二項分布以外には $z_i = (\tilde{\eta}_i - o_i)$ で割り当てます。

4. 最初のパラメータ値を計算します。

$$\beta^{(0)} = \left(\mathsf{X}^{\mathsf{T}} \tilde{W}_e \mathsf{X}\right)^{-1} \mathsf{X}^{\mathsf{T}} \tilde{W}_e \mathsf{z}$$

そして

$$\phi^{(0)} = \left(\mathsf{z} - \mathsf{X}\beta^{(0)}\right)^{\mathsf{T}} \tilde{W}_e \left(\mathsf{z} - \mathsf{X}\beta^{(0)}\right)$$

尺度パラメータが ML 方法で推定される場合です。

尺度パラメータの取り扱い

1. 正規分布、逆ガウス分布、およびガンマ分布、 スケール パラメータが ML 方法で推定される場合、回帰パラメータと結合して推定されます。つまり、傾斜ベクトル **s** の最後のエレメントは τ に関してです。

2. 尺度パラメータが固定正数値に設定されている場合は、上記プロセスの各反復でその値が固定されたまま保持されます。

3. 尺度パラメータが逸脱または自由度で除算された Pearson カイ 2 乗で指定されている場合は、反復プロセス全体を通じて、回帰推定値を得るために 1 に固定されます。回帰推定値に基づいて、逸脱度および Pearson カイ 2 乗の値を計算し、尺度パラメータの推定値を取得します。

乖離の確認

ユーザー定義の反復数の後に各反復に対し、つまり i > I の場合に、以下を計算します（ここでは v がデータセット内のケースを参照します)。

$$p_{\min} = \min_v p_v$$

$$p_{\max} = \max_v p_v,$$

$$p^*_{\min} = \min_v \left(\min\left(\mu_v, 1-\mu_v\right)\right),$$

ここで、

$$p_v = \begin{cases} \mu_v & \text{if } y_v = \text{success } (=1) \\ 1-\mu_v & \text{if } y_v = \text{ failure } (=0) \end{cases}$$

（p_v はケース v の観察されたレスポンスの確立）および $\mu_v = g^{-1}\left(\mathbf{x}_v^{\mathsf{T}}\,\beta + o_v\right)$?

$\min\left(p_{\min}, p_{\max}\right) = p_{\min} > 0.99$ の場合、完全乖離があるとみなします。$p_{\max} > 0.99$ または $p^*_{\min} < 0.001$ の場合、そして非常に小さい対角成分（絶対値 $< \sqrt{10^{-7}} \approx 3.16 \times 10^{-4}$）が -H の Cholesky の分解における下三角行列の非冗長パラメータの位置にある場合（H は Hessian 行列)、疑似完全乖離があります。

収束基準

次の収束基準が考慮されます。

$$\text{対数尤度収束 }: \begin{cases} \dfrac{\left|\ell^{(i)} - \ell^{(i-1)}\right|}{\left|\ell^{(i-1)}\right| + 10^{-6}} < \epsilon_\ell & \text{相対的変更の場合} \\ \left|\ell^{(i)} - \ell^{(i-1)}\right| < \epsilon_\ell & \text{絶対的変更の場合} \end{cases}.$$

$$\text{パラメータ収束} \begin{cases} \max_j \left(\dfrac{\left|\beta_j^{(i)} - \beta_j^{(i-1)}\right|}{\left|\beta_j^{(i-1)}\right| + 10^{-6}} \right) < \epsilon\mathsf{p} & \text{相対的変更の場合} \\ \max_j \left(\left|\beta_j^{(i)} - \beta_j^{(i-1)}\right| \right) < \epsilon\mathsf{p} & \text{絶対的変更の場合} \end{cases}.$$

$$\text{Hessian 収束} \begin{cases} \dfrac{\left(\mathbf{s}^{(i)}\right)^{\mathsf{T}} \left(\mathbf{H}^{(i)}\right)^{-} \left(\mathbf{s}^{(i)}\right)}{\left|\ell^{(v)}\right| + 10^{-6}} < \epsilon_{\mathbf{H}} & \text{相対的変更の場合} \\ \left(\mathbf{s}^{(i)}\right)^{\mathsf{T}} \left(\mathbf{H}^{(i)}\right)^{-} \left(\mathbf{s}^{(i)}\right) < \epsilon_{\mathbf{H}} & \text{絶対的変更の場合} \end{cases}$$

ここで、$\epsilon_\ell, \epsilon\mathsf{p}$ と $\epsilon_{\mathbf{H}}$ は、各タイプに指定された許容レベルです。

Hessian 収束の基準がユーザー指定でない場合、対数-尤度またはパラメータ収束基準が満たされた後、ϵ_{H} = 1E-4 による絶対変化に基づいてチェックされます。Hessian 収束基準が満たされない場合、警告メッセージが表示されます。

パラメータ推定の共分散行列、相関行列、および標準誤差

パラメータ推定の共分散行列、相関行列および標準誤差は、パラメータ推定値で簡単に取得できます。尺度パラメータが ML によって推定されるかどうかにかかわらず、パラメータ推定の共分散および相関行列は、$\hat{\beta}$ および $\hat{\tau}$ の間の共分散が 0 である必要があるため、$\hat{\beta}$ にのみ一覧表示されます。

モデルベースのパラメータ推定値の共分散

モデルベースのパラメータ推定値の共分散行列は、次のように指定されます。

$$\Sigma_{\mathsf{m}} = -\mathrm{H}^{-} = -(-\mathbf{X}\,\mathbf{W}\,\mathbf{X})^{-}$$

ここで、$\mathbf{H}^{-}$ は、パラメータ推測値で評価される Hessian 行列の一般化された逆数です。冗長パラメータ推定値の対応する行と列は、0 に設定する必要があります。

堅牢なパラメータ推定値の共分散

Hessian に基づくパラメータ推定共分散行列の有効性は、レスポンスの平均回帰関数の正しい指定に加えて、レスポンスの分散関数の正しい指定に左右されます。堅牢なパラメータ推定値の共分散は、レスポンスの分散関数の仕様が誤っている場合でさえも、一貫した推定値を提供します。この堅牢な推定法は Huber の推定法と呼ばれます。Huber が 1967 年に最初にこの分散推定法を発表したためです。White の推定法または HCCM (heteroskedasticity consistent covariance matrix) 推定法は、1980 年に White がこの分散推定値が一貫してヘテロスケダスシティを含んだ線型回帰モデル下にあることを独自に示しました。または、3 つの項を含んでいるためにサンドイッチ推定方法とも呼ばれます。堅牢な (または Huber/White/サンドイッチ) 推定法は、次のように定義されます。

$$\Sigma_{\mathsf{r}} = \Sigma_{\mathsf{m}}\left(\sum_{i=1}^{n}\left[\frac{\partial \ell_i}{\partial \beta}\right]\left[\frac{\partial \ell_i}{\partial \beta}\right]^{\mathsf{T}}\right)\Sigma_{\mathsf{m}} = \Sigma_{\mathsf{m}}\left(\sum_{i=1}^{n} f_i\left(\frac{\omega_i(y_i-\mu_i)}{\phi V(\mu_i)\,g'(\mu_i)}\right)^{2}\cdot x_i\cdot x_i^{\mathsf{T}}\right)\Sigma_{\mathsf{m}}$$

パラメータ推定値の相関

相関行列は、通常どおりに共分散から計算されます。σ_{ij} を Σ_{m} または Σ_{r} のエレメントとし、相関行列の対応するエレメントは、$\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}}}$ となります。冗長パラメータ推定値の対応する行と列は、システム欠損値に設定する必要があります。

パラメータ推定値の標準誤差

$\hat{\beta}_i$ が、非冗長パラメータ推定値を指すようにします。標準誤差が Σ_{m} または Σ_{r} の i 番目の対角線要素の平方根の場合は、次のようになります。

$$\hat{\sigma}_{\beta_i} = \sqrt{\sigma_{ii}}$$

冗長パラメータ推定値の標準誤差は、システム欠損値に設定されます。尺度パラメータが ML メソッドで推定される場合、$\hat{\tau}$ および標準誤差推定 $\hat{\sigma}_\tau = \sqrt{-\frac{1}{\left(\frac{\partial^2 \ell}{\partial \tau^2}\right)}}$ を取得します。$\frac{\partial^2 \ell}{\partial \tau^2}$ は、次のリンクを参照してください。テーブル 18-7 “確率分布の 2 番目の導関数 w.r.t. スケール パラメータ” p. 168 . したがって、尺度パラメータの推定値は $\exp(\hat{\tau})$ であり、標準誤差推定値は $(\exp(\hat{\tau}) \cdot \hat{\sigma}_\tau)$ です。

ワルド信頼区間

ワルド信頼区間は、パラメータ推定値の漸近正規分布に基づいています。β_j の 100(1 - α)% ワルド信頼区間は、次により与えられます。

$$\left(\hat{\beta}_j - z_{1-\alpha/2}\hat{\sigma}_{\beta_j}, \hat{\beta}_j + z_{1-\alpha/2}\hat{\sigma}_{\beta_j}\right),$$

ここで、z_p は標準的な正規分布の 100p 番目の 100 分位です。

指数パラメータ推定値がロジスティック回帰または対数-線型モデルに対して要求された場合、デルタ方法を使用すると、$\exp(\beta_j)$ の推定値は $\exp\left(\hat{\beta}_j\right)$ であり、$\exp\left(\hat{\beta}_j\right)$ の標準誤差推定値は $\left(\exp\left(\hat{\beta}_j\right) \cdot \hat{\sigma}_{\beta_j}\right)$ です。また、$\exp(\beta_j)$ に対応する 100(1 - α)% ワルド信頼区間は次のとおりです。

$$\left(\exp\left(\hat{\beta}_j - z_{1-\alpha/2}\hat{\sigma}_{\beta_j}\right), \exp\left(\hat{\beta}_j + z_{1-\alpha/2}\hat{\sigma}_{\beta_j}\right)\right)$$

冗長パラメータ推定値のワルド信頼区間は、システム欠損値に設定されます。

同様に、ϕ は次のようになります。

$$\left(\exp\left(\hat{\tau} - z_{1-\alpha/2}\hat{\sigma}_\tau\right), \exp\left(\hat{\tau} + z_{1-\alpha/2}\hat{\sigma}_\tau\right)\right)$$

カイ 2 乗統計量

仮説 $H_{0i} : \beta_i = 0$ が、次のカイ 2 乗統計量を使用して各非冗長パラメータについて検定されます。

$$c_i = \left(\frac{\hat{\beta}_i}{\hat{\sigma}_{\beta_j}}\right)^2$$

これには、自由度 1 で漸近カイ 2 乗分布があります。

カイ 2 乗統計量と対応する p 値は、冗長パラメータ推定値のシステム欠損値に設定されます。

カイ 2 乗統計量は、ML 方法によって推定される場合であっても、尺度パラメータについては計算されません。

P 値

検定統計量の T と上に指定された対応する累積分布関数 G が指定されると、p 値は $p = 1 - G(T)$ として定義されます。たとえば、$H_{0i} : \beta_i = 0$ のカイ 2 乗検定の p 値は、$p_i = 1 - prob\left(\chi_1^2 \leq c_i\right)$ です。

モデル検定

パラメータを推定し関連する統計量を計算した後に、指定されたモデルにいくつかの検定が実行されます。

ラグランジュの未定係数法検定

正規分布、逆ガウス分布、ガンマ分布、分布のスケール パラメータが固定値に設定されるか、逸脱度または自由度に除算された Pearson カイ 2 乗によって指定されるか （スケール パラメータが逸脱度または自由度で除算された Pearson カイ 2 乗により指定されると、固定値とみなされることがあります）、または、負の二項分布の補助パラメータ k が 0 以外の固定値に設定された場合、ラグランジュの未定係数法 (LM) 検定により、その値の有効性が査定されます。固定された ϕ または k に対して、検定の統計値は次のように定義されます。

$$T_{LM} = \frac{s^2}{A}$$

ここで、$s = \partial\ell/\partial\tau$ および $A = -\left(\frac{\partial^2\ell}{\partial\tau^2}\right) - \left(-\frac{\partial^2\ell}{\partial\tau\partial\beta^{\mathsf{T}}}\right)\left(-\frac{\partial^2\ell}{\partial\beta\partial\beta^{\mathsf{T}}}\right)^{-}\left(-\frac{\partial^2\ell}{\partial\beta\partial\tau}\right)$ はパラメータ推定値で評価し、ϕ または k の値で固定しました。T_{LM} には自由度 1 の漸近カイ 2 乗分布があり、p 値はそれに応じて計算されます。

ϕ の検定については、 次を参照してください。 テーブル 18-6“確率分布の最初の導関数 w.r.t. 尺度パラメータ” p. 167 そして 次を参照してください。テーブル 18-7“確率分布の 2 番目の導関数 w.r.t. スケール パラメータ” p. 168 s および A のエレメントについては、それぞれ上記で説明されています。

k が 0 に設定されている場合は、上記の統計が適用できません。1998 年の Cameron および Trivedi の発表によれば、LM 検定は現在、次の補助 OLS 回帰（定数なし）に基づいて実行される必要があります。

$$\frac{(y_i - \hat{\mu}_i)^2 - y_i}{\hat{\mu}_i} = \alpha\hat{\mu}_i + \epsilon_i$$

ここで、$\hat{\mu}_i = g^{-1}\left(x_i^T\hat{\beta}\right)$ and ϵ_i はエラー項です。上記 OLS 回帰の $\left[(y_i - \hat{\mu}_i)^2 - y_i\right]/\hat{\mu}_i$ を z_i にし、説明変数 $\hat{\mu}_i$ を w_i にします。上記の回帰パラメータ α の予測値と α の予測の標準誤差は、次のとおりです。

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} f_i w_i z_i}{\sum_{i=1}^{n} f_i w_i^2} \quad \text{and} \quad \hat{\sigma}_\alpha = \sqrt{\frac{\frac{s_e^2}{n}}{\sum_{i=1}^{n} f_i w_i^2}},$$

ここで、$s_e^2 = \frac{1}{N-1}\sum_{i=1}^{n} f_i e_i^2$ および $e_i = z_i - \hat{\alpha} w_i$ です。これで、LM 検定統計は z 統計の

$$z = \frac{\hat{\alpha}}{\hat{\sigma}_\alpha},$$

さらに、ポアソン モデル内で過分散の仮説なしで漸近的標準正規分布（$H_0 : k = 0$）を持ちます。3 つの p-値が与えられます。別の仮説としては、片側過分散（$H_a : k > 0$）、過小分散（$H_a : k < 0$）または両側非方向（$H_a : k \neq 0$）に $V(\mu) = \mu + k\mu^2$ の分散関数を伴います。p 値の計算は選択肢によります。$H_a : k > 0$,p-value $= 1 - \Phi(z)$, で $\Phi(\cdot)$ が標準正規分布の累積確率の場合、$H_a : k < 0$,p-value $= \Phi(z)$; および $H_a : k \neq 0$,p-value $= 2(1 - \Phi(|z|))$. の場合

適合度統計

任意の一般化線型モデルの適合度を査定するために、統計量がいくつか計算されます。

Deviance

逸脱度の理論的定義は次のとおりです。

$$D = 2\phi(\ell(\mathsf{y}; \mathsf{y}) - \ell(\hat{\mu}; \mathsf{y})),$$

ここで、$\ell(\hat{\mu}; \mathsf{y})$ は、反応変数に与えられた予測された平均値 $\hat{\mu}$（パラメータ推定値に基づいて計算済み）の関数として表現された対数尤度関数であり、$\ell(\mathsf{y}; \mathsf{y})$ は、$\hat{\mu}$ を y で置き換えることで計算される対数尤度関数です。逸脱に使用される公式は $\sum_{i=1}^{n} f_i d_i$ です。各分布の di の形式は、次の表に示します。

分布	di
Normal	$\omega_i(y_i - \mu_i)^2$
逆ガウス分布	$\frac{\omega_i}{y_i \mu_i^2}(y_i - \mu_i)^2$
Gamma	$2\omega_i\left\{-\ln\left(\frac{y_i}{\mu_i}\right) + \frac{y_i - \mu_i}{\mu_i}\right\}$
負の二項	$2\omega_i\left\{y_i \ln\left(\frac{y_i}{\mu_i}\right) - (y_i + 1/k)\ln\left(\frac{y_i + 1/k}{\mu_i + 1/k}\right)\right\}$
Poisson	$2\omega_i\left\{y_i \ln\left(\frac{y_i}{\mu_i}\right) - (y_i - \mu_i)\right\}$
二項（m）	$2\omega_i^*\left\{y_i \ln\left(\frac{y_i}{\mu_i}\right) + (1 - y_i)\ln\left(\frac{1 - y_i}{1 - \mu_i}\right)\right\}$

注：

- **y** が 0/1 の値の 2 進数従属変数の場合（二項分布）、逸脱と Pearson カイ 2 乗は、部分母集団に基づいて計算されます。下記を参照してください。
- 負の二項分布およびポアソン分布で y = 0 か、**r**/**m** 形式の二項分布で y = 0（r = 0 に対して）または 1（r = m に対して）の場合は、別々の値が逸脱に対して与えられます。負の二項とポアソンについては $y_i = 0$、二項については 0/1 のときに、d_i を個々のケース i の逸脱値とします。

分布	d_i
負の二項	$2\omega_i \frac{\ln(1+k\mu_i)}{k}$ if $y_i = 0$
Poisson	$2\omega_i\mu_i$ if $y_i = 0$
二項（m）	$\begin{cases} -2\omega_i^* \ln(1-\mu_i) & \text{if } y_i = 0 \text{ または } r_i = 0 \\ -2\omega_i^* \ln(\mu_i) & \text{if } y_i = 1 \text{ または } r_i = m_i \end{cases}$

Pearson カイ 2 乗

$$\chi^2 = \sum_{i=1}^{n} f_i \gamma_i$$

ここで、二項分布の場合は $\gamma_i = \frac{\omega_i^*(y_i-\mu_i)^2}{V(\mu_i)}$ およびそのほかの分布の場合は $\gamma_i = \frac{\omega_i(y_i-\mu_i)^2}{V(\mu_i)}$ となります。

計測された逸脱度および計測された Pearson カイ 2 乗

計測された逸脱は $D^* = D/\phi$ であり、計測された Pearson カイ 2 乗は $\chi^{2*} = \chi^2/\phi$ です。

計測された逸脱と Pearson カイ 2 乗統計量には自由度が N - p_x の制限カイ 2 乗分布があるので、自由度で除算された逸脱とカイ 2 乗は、連続型と離散型の両方の分布の尺度パラメータの推定値として使用できます。

$\hat{\phi} = \frac{D}{N-p_x}$ または $\hat{\phi} = \frac{\chi^2}{N-p_x}$

尺度パラメータが逸脱または Pearson カイ 2乗により計測され、最初に $\phi = 1$ とみなす場合は、回帰パラメータを推定し、逸脱と Pearson カイ 2乗の値を計算して、上記公式から尺度パラメータ推定値を取得します。次に、両方の統計量の測定されたバージョンは、$\hat{\phi}$ によって逸脱と Pearson カイ 2 乗を除算することで得られます。その一方で、ある種の統計量には修正が必要です。傾斜ベクトルと Hessian 行列は $\hat{\phi}$ で除算され、共分散行列は $\hat{\phi}$ で乗算されます。推定された標準誤差もそれに応じて調整され、ワルド信頼区間と有意度検定は、パラメータ推定値が $\hat{\phi}$ に影響を受けない場合でさえも、影響されます。

対数尤度は修正されないことに注意してください。つまり対数尤度は、情報基準およびモデル適合オムニバス検定で公正な比較が行われるように尺度パラメータが対数尤度で同じであるように保たれる必要があるため、$\phi = 1$ に基づいています。

過剰分散

ポアソン分布と二項分布に対して、推定された尺度パラメータが想定された値に近似していない場合は、その値が 1 より大きいときはデータが過剰分散となり、1 より小さい場合は分散不足の可能性があります。過剰分散のほうが、実際は一般的です。過剰分散にかかわる問題は、それが推定中のパラメータの標準誤差の原因となる可能性があることです。変数は有意な予測フィールドであるように見えることがありますが、実際はそうではありません。

0/1 の 2 進数レスポンス変数付きの二項分布の逸脱度および Pearson カイ 2 乗

r および **m**（イベント/試行）変数が二項分布に使用されると、各ケースが m ベルヌーイ試行を表します。**y** が 0/1 の値の 2 進数従属変数の場合は、各ケースが単独の繰り返し回数を表します。この繰り返し回数は、同じ設定（すなわち、すべての予測値変数に同じ値を設定など）で複数回繰り返すことができます。たとえば、最初の 10 個の y 値が 2 個の 1 と 8 個の 0 で、**x** 値が同じであるとすると（イベント/繰り返し回数形式で記録されていると、10 ケースは r = 2 および m = 10 で 1 ケースとして記録される）、これら 10 ケースは同じ部分母集団からとみなされる必要があります。すべての予測値変数が含まれた変数リスト内で共通する値のケースは、同じ部分母集団からのものと見なされます。2 進数のレスポンス付きの二項分布が使用される場合は、この部分母集団に基づいて逸脱度と Pearson カイ 2 乗を計算する必要があります。ケースに基づいて計算すると、結果が役に立たないおそれがあります。

部分母集団が 0/1 反応変数付きの二項分布に指定された場合、データは、単一試行形式からイベント/試行形式へ再構築する必要があります。形式化されたデータには、次の表記が想定されます。

n_s	部分母集団数。
r_{j1}	j 部分母集団内で度数の積と y = 1 に関連付けられた尺度の重みの合計。したがって、r_{j0} は j 番目の部分母集団内の y = 0 の合計です。
m_j	重み付けされた観察値の合計で、$m_j = r_{j1} + r_{j0}$。
y_{j1}	j 番目の部分母集団内の 1s の比率で、$y_{j1} = r_{j1} / m_j$。
μ_j	j 番目の部分母集団内に組み込まれた確立（すべての予測値変数が各ケースに対して同じであるため、j 番目の部分母集団内の各ケースに対して、$\hat{\mu}_j$ は同じです）。

逸脱度および Pearson カイ 2 乗は、次のように定義されます。

$$D = 2\sum_{j=1}^{n_s} m_j \left\{ y_{j1} \ln\left(\frac{y_{j1}}{\mu_j}\right) + (1 - y_{j1}) \ln\left(\frac{1 - y_{j1}}{1 - \mu_j}\right) \right\} \text{ および } \chi^2 = \sum_{j=1}^{n_s} \frac{m_j (y_{j1} - \mu_j)^2}{\mu_j (1 - \mu_j)}、$$

そして、対応する尺度パラメータの推定値は次のようになります。

$\hat{\phi} = \frac{D}{n_s - p_x}$ および $\hat{\phi} = \frac{\chi^2}{n_s - p_x}$。

部分母集団に基づいて、完全対数尤度関数が次のように定義されます。

$$\ell = \ell_k + \sum_{j=1}^{n_s} \frac{1}{\phi}\left\{\ln\left(\begin{matrix} m_j \\ r_{j1} \end{matrix}\right)\right\} = \ell_k + \sum_{j=1}^{n_s} \frac{1}{\phi}\left\{\ln \frac{m_j!}{r_{j1}!r_{j0}!}\right\},$$

ここで、ℓ_k はカーネル対数尤度となります。これは以前のケースに基づき計算されたカーなる対数尤度と同じである必要があり、もう一度計算する必要はありません。

情報基準

情報量基準は、同じデータで異なるモデルを比較するときに使用されます。さまざまな基準の公式は次のとおりです。

赤池情報量基準（AIC）	$-2\ell + 2d$
修正された有限サンプル（AICC）	$-2\ell + \frac{2d \cdot N}{(N-d-1)}$
ベイズ情報量基準（BIC）	$-2\ell + d\ln(N)$
一致 AIC（CAIC）	$-2\ell + d(\ln(N)+1)$

ここで、ℓ はパラメータ推定値で評価される対数尤度です。β が含まれる場合のみ、 d = p_X です。正規、逆ガウス、ガンマの尺度パラメータが含まれている場合は、d = p_X + 1 です。負の二項分布には

メモ

- ℓ（完全な対数尤度）は、ユーザーの選択に応じて ℓ_k（対数尤度のカーネル）で置き換えることができます。
- **r** および **m**（イベント/試行）変数が二項分布に使用されると、ここで使用される N が試行度数の合計、つまり $N = \sum_{i=1}^{n} f_i m_i$ になります。この方法では、データが生の 2 進数形式または要約された二項形式のどちらでも、同じ結果になります。

モデル適合度の検定

モデル適合度のオムニバス検定は、検討中のモデルおよび初期モデルについての -2 対数尤度値に基づいています。検討中のモデルの場合、-2 対数尤度の値は次のようになります。

$$-2\ell\left(\hat{\beta}\right)$$

検討中のモデル内に切片がある場合は初期モデルを切片のみのモデルとし、そうでない場合は空のモデルとします。定数項のみのモデルの場合、-2 対数尤度の値は次のようになります。

$$-2\ell\left(\hat{\beta}_0\right)$$

空のモデルの場合、-2 対数尤度の値は次のようになります。

$$-2\ell(0)$$

そのため、オムニバス（またはグローバル）検定の統計量は、次のようになります。

切片のみのモデルの場合は $S = 2\left(\ell\left(\hat{\beta}\right) - \ell(\beta_0)\right)$

空のモデルの場合は $S = 2\left(\ell\left(\hat{\beta}\right) - \ell(0)\right)$

S は検討中のモデルと初期モデルの間の有効なパラメータ数の差異に等しい、r 自由度のある漸近カイ 2 乗分布を持っています。 切片だけのモデルは r = $p_x - 1$、また空のモデルは r = p_x です。p 値はそれに応じて算出できます。

尺度パラメータ が検討中のモデル内で ML 方法により推定される場合は、初期モデル内でも ML 方法で推定されます。

モデル効果のデフォルトの検定

モデル内で指定された各回帰効果に対し、タイプ I およびタイプ III の分析を実施できます。

タイプ I の分析

タイプ I 分析は、モデルのシーケンスの適合、切片のみのモデル（ある場合）から始め、各ステップでモデルの共変量、係数と交互作用になり得る追加の効果を加算します。そのため、モデル内に指定される効果の順番に影響を受けます。一方、タイプ III 分析は効果の順序に左右されません。

ワルド統計量： モデル内に指定された各効果について、タイプ I の検定行列 $\mathbf{L}_i$ が作成され、H_0:$\mathbf{L}_i\beta$ = **0** が検定されます。行列 $\mathbf{L}_i$ の作成は、生成行列の $\mathbf{H}_\omega = \left(\mathbf{X}^\mathsf{T}\mathbf{\Omega}\mathbf{X}\right)^{-}\mathbf{X}^\mathsf{T}\mathbf{\Omega}\mathbf{X}$, に基づいています。 **Ω** は、is the scale weight matrix with i 番目の傾斜要素 ω_i 付きの尺度と重みの行列であり $\mathbf{L}_i\beta$ のようなものが推定可能です。これには、ある一定の効果およびその効果を含む複数の効果に対してのみのパラメータが関与しています。このような行列が作成できない場合、効果は検定不能です。

ワルド統計はタイプ I と III 分析そしてユーザー設定検定に適用できるため、ワルド統計をもっと一般的な形式で表します。$\mathbf{L}_i\beta = \mathbf{K}$ を検定するワルド統計に関しては、$\mathbf{L}_i$ は、r×p 完全行ランク仮説行列であり、 **K** は r×1 結果ベクトルであり、次によって定義されます。

$$S = \left(\mathbf{L}_i\hat{\beta} - \mathbf{K}\right)^\mathsf{T}\left(\mathbf{L}_i\mathbf{\Sigma}\mathbf{L}_i{}^\mathsf{T}\right)^{-}\left(\mathbf{L}_i\hat{\beta} - \mathbf{K}\right)$$

ここで、$\hat{\beta}$ は最大尤度推定値であり、$\mathbf{\Sigma}$ はパラメータ推定値共分散行列です。S には、自由度が r_C の漸近的カイ 2 乗分布があります。ここで、$r_C = rank\left(\mathsf{L}\,\Sigma\mathsf{L}^{\mathsf{T}}\right)$ です。$r_C < r$ の場合、$\left(\mathsf{L}\,\Sigma\mathsf{L}^{\mathsf{T}}\right)^{-}$ は一般化された逆数で、ワルド検定は H_0 から独立した行の特定のサブセット C を含む制限付きの仮説セット $\mathbf{L}_{iC}\beta - \mathbf{K}_C$ に対して効果的です。

タイプ I および III 分析に対しては、対応する仮説行列 $\mathrm{L_i}$ および K=0 に従って各効果 i のワルド統計値を計算します。

タイプ III の分析

ワルド統計量： 前に説明したタイプ I の「ワルド統計量」を参照してください。L_i が、i 番目の効果のタイプ III 検定行列です。

空白の処理

入力フィールドまたは出力フィールドに欠損値があるすべてのレコードは、モデルの推定から除外されます。

スコアリング

スコアリングは、データセット内の 1 ケースへの 1 つ以上の値の割り当てとして定義されます。

予測値

非線型リンク関数が原因で、予測値は線型予測フィールドと反応の平均に対して別々に計算されます。また、線型予測フィールドの予測値の推定標準誤差が計算されるので、平均の信頼区間を簡単に取得できます。

予測値は、指定されたモデル内ですべての予測値変数に非欠損値があるかぎり、引き続き計算されます。

線型予測フィールドの予測値

$$\hat{\eta}_i = x_i^{\mathsf{T}}\hat{\beta} + \mathsf{o}_i$$

線型予測フィールドの予測値の推定標準誤差

$$\hat{\sigma}_\eta = \sqrt{x_i^{\mathsf{T}}\,\mathbf{\Sigma}\,x_i}$$

平均の予測値

$$\hat{\mu}_i = g^{-1}\left(x_i^{\mathsf{T}}\hat{\beta} + o_i\right)$$

ここで、g^{-1} はリンク関数の逆数です。0/1 レスポンス変数付きの二項レスポンスにとって、これはカテゴリ 1 の予測確率です。

平均の信頼区間

平均の約 100(1−α)% の信頼区間が次のようにして計算できます。

$$g^{-1}\left(x_i^{\mathsf{T}}\hat{\beta} + o_i \pm z_{1-\alpha/2}\hat{\sigma}_\eta\right)$$

引数のどちらかの終点が逆リンク関数の有効な範囲の外側の場合、対応する信頼区間の終点はシステム欠損値に設定されます。

空白の処理

最終モデルにおいて、入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールド $null$ が割り当てられます。

参照

Aitkin, M., D. Anderson, B. Francis, および J. Hinde. 1989. Statistical Modelling in GLIM. Oxford: Oxford Science Publications.

Albert, A., および J. A. Anderson. 1984. On the Existence of Maximum Likelihood Estimates in Logistic Regression Models. Biometrika, 71, 1-10.

Cameron, A. C., および P. K. Trivedi. 1998. Regression Analysis of Count Data. Cambridge: Cambridge University Press.

Diggle, P. J., P. Heagerty, K. Y. Liang, および S. L. Zeger. 2002. The analysis of Longitudinal Data, 2 ed. Oxford: Oxford University Press.

Dobson, A. J. 2002. An Introduction to Generalized Linear Models, 2 ed. Boca Raton, FL: Chapman & Hall/CRC.

Dunn, P. K., および G. K. Smyth. 2005. Series Evaluation of Tweedie Exponential Dispersion Model Densities. Statistics and Computing, 15, 267-280.

Dunn, P. K., および G. K. Smyth. 2001. Tweedie Family Densities:Methods of Evaluation. In: Proceedings of the 16th International Workshop on Statistical Modelling, Odense, Denmark: .

Gill, J. 2000. Generalized Linear Models:A Unified Approach. Thousand Oaks, CA: Sage Publications.

Hardin, J. W., および J. M. Hilbe. 2001. Generalized Estimating Equations. Boca Raton, FL: Chapman & Hall/CRC.

Hardin, J. W., および J. M. Hilbe. 2003. Generalized Linear Models and Extension. Station, TX: Stata Press.

Horton, N. J., および S. R. Lipsitz. 1999. Review of Software to Fit Generalized Estimating Equation Regression Models. The American Statistician, 53, 160-169.

Huber, P. J. 1967. The Behavior of Maximum Likelihood Estimates under Nonstandard Conditions. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA: University of California Press, 221-233.

Lane, P. W., および J. A. Nelder. 1982. Analysis of Covariance and Standardization as Instances of Prediction. Biometrics, 38, 613-621.

Lawless, J. E. 1984. Negative Binomial and Mixed Poisson Regression. The Canadian Journal of Statistics, 15, 209-225.

Liang, K. Y., および S. L. Zeger. 1986. Longitudinal Data Analysis Using Generalized Linear Models. Biometrika, 73, 13-22.

Lipsitz, S. H., K. Kim, および L. Zhao. 1994. Analysis of Repeated Categorical Data Using Generalized Estimating Equations. Statistics in Medicine, 13, 1149-1163.

McCullagh, P. 1983. Quasi-Likelihood Functions. Annals of Statistics, 11, 59-67.

McCullagh, P., および J. A. Nelder. 1989. Generalized Linear Models, 2nd ed. London: Chapman & Hall.

Miller, M. E., C. S. Davis, および J. R. Landis. 1993. The Analysis of Longitudinal Polytomous Data:Generalized Estimating Equations and Connections with Weighted Least Squares. Biometrics, 49, 1033-1044.

Nelder, J. A., および R. W. M. Wedderburn. 1972. Generalized Linear Models. Journal of the Royal Statistical Society Series A, 135, 370-384.

Pan, W. 2001. Akaike's Information Criterion in Generalized Estimating Equations. Biometrics, 57, 120-125.

Pregibon, D. 1981. Logistic Regression Diagnostics. Annals of Statistics, 9, 705-724.

Smyth, G. K., および B. Jorgensen. 2002. Fitting Tweedie's Compound Poisson Model to Insurance Claims Data:Dispersion Modelling. ASTIN Bulletin, 32, 143-157.

白, H. 1980. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. Econometrica, 48, 817-836.

Williams, D. A. 1987. Generalized Linear Models Diagnostics Using the Deviance and Single Case Deletions. Applied Statistics, 36, 181-191.

Zeger, S. L., および K. Y. Liang. 1986. Longitudinal Data Analysis for Discrete and Continuous Outcomes. Biometrics, 42, 121-130.

章
19

欠損値の代入

次の方法で、欠損値の代入ができます。

固定： 固定値で置き換えます（指定のフィールド計測、範囲の中間または一定数）。

無作為： 正常または均一分布に基づいたランダム値で置き換えます。

式：ユーザー設定の式を指定することができます。たとえば、値をグローバル値設定ノードで作成されたグローバル変数と置き換えることができます。

アルゴリズム： C&RT アルゴリズムの基づいたモデルによって予測された値で置き換えます。この方法で代入された各フィールドに対し、空白値やヌル値をモデルで予測された値と置き換える置換ノードとともに、個別の C&RT モデルが作成されます。フィルタ ノードを使用して、モデルが生成した予測値を削除します。

各代入方法を以下に詳しく説明します。

固定値の代入

固定値を代入する場合は、次の 3 つのオプションが利用できます。

平均値：次のようにして、代入されるフィールドの有効な学習データ値の平均で置き換えます。

$$\frac{\sum_{i=1}^{n_{valid}} x_i}{n_{valid}}$$

ここで、 x_i は レコード i に対するフィールド x の値で欠損値は含まれず、n_{valid} は、フィールド x に対し有効な値を持つレコードの数です。

中間値：次のようにして、代入されるフィールドの最小値と最大値の間の中間の値で置き換えます。

$$x_{\min} + \frac{x_{\max} - x_{\min}}{2} = \frac{x_{\max} + x_{\min}}{2}$$

ここで、 $x_{\min}$ および $x_{\max}$ はそれぞれ、フィールド x に対する有効な観察値の最小値および最大値です。

定数：ユーザー定義の定数値で置き換えます。

セット型またはフラグ型フィールド内に固定欠損値を代入する場合は、[**定数**] オプションのみが利用できます。

注 ： スケール フィールドに固定された代入値を使用すると、そのフィールドの分散が人工的に減らされ、そのフィールドを使用するモデル構築が妨害されることがあります。固定値を使用して代入し、フィールドにはもうモデル内の期待効果がないことが判明した場合は、フィールドの分散への影響がより少ない別の方法での代入を検討します。

乱数値の代入

乱数値の代入の場合、代入されるフィールドの型に応じてオプションが異なります。

範囲型フィールド

範囲型のフィールドの場合、一様分布または正規分布から選択できます。

一様分布 : 値は、$[x_{\min}, x_{\max}]$ の間隔で無作為に生成され、その間隔内の各値は同じように生成されます。

正規分布 : 値は、平均が $\overline{x}_{valid}$ で分散が s^2_{valid} の正規分布から生成され、$\overline{x}_{valid}$ および s^2_{valid} は、学習データ内の x の有効な観察値から得られます。

$$\overline{x}_{valid} = \frac{\sum_{i=1}^{n_{valid}} x_i}{n_{valid}}$$

$$s^2_{valid} = \frac{\sum_{i=1}^{n_{valid}} \left(x_i - \overline{x}_{valid}\right)^2}{n_{valid} - 1}$$

セット型フィールド

セット型フィールドについては、乱数代入値が観察結果の値のリストから選択されます。デフォルトでは、次のようにすべての値の確率が同じです。

$$p(k) = \frac{1}{j}$$

これは、k の j 個の可能な値の場合です。[**均等化**] ボタンを押すと、変更された値がデフォルトの等しい確率に戻ります。

[**検査に基づく**] を選択すると、確率は、学習データ内の値の相対的な度数へ比例して割り当てられます。

$$p(k) = \frac{n_k}{n_{valid}}$$

ここで、n_k は、$x_i = k$ の場合のレコード数です。

[**正規化**] を選択した場合、同じ相対的な比率を保ちながら値が 1.0 への合計へ調整されます。

$$p_{normalized}(k) = \frac{p(k)}{\sum_k p(k)}$$

生成された乱数値に独自の重みを入力したいのにかかわらずそれが確率として表現されていない場合に、これは有用です。たとえば、Yes 値の 2 倍の No 値が必要であることがわかっている場合は、No に「2」、Yes に「1」を入力して、**[正規化]** をクリックします。正規化により、相対的な重みを保存してしかもそれを確率として表現しながら、値が 0.667 および 0.333 へ調整されます。

式から作成された代入値

式基準の代入では、代入値がユーザー定義の CLEM 式に基づきます。式は、置換ノードであるかのように評価されます。 この方法による代入後も欠損値が存在する可能性があるので、式の中には $null または他の欠損値を返すことがあります。

アルゴリズムから作成された代入値

アルゴリズム方法の場合は、予測フィールドとして他のすべての入力フィールドを使用して、C&RT モデルが代入される各フィールドに構築されます。代入される各レコードに対し、代入されるフィールドのモデルが予測を作成するためにレコードに適用され、これが代入された値として使用されます。 詳細は、 9 章 p.59 C&RT の概要 を参照してください。

章
20

K-Means アルゴリズム

概要

K-Means はクラスタリング手法で、一連の入力フィールドの値の類似性に基づいてレコードをグループ化するために用いられます。基本的には、それぞれのクラスタ中のレコードが互いに類似しており、他のクラスタ中のレコードとは異なっているような k クラスタを探します。K-Means は反復アルゴリズムで、クラスタの初期セットが定義された後、改善が不可能になるまで（または指定した反復数を超えるまで）繰り返し更新されます。

一次計算

K-Means モデルの構築では、測定値の尺度と種類の差異を明らかにするために入力フィールドがコード化された後、クラスタが定義、更新されて、最終モデルが生成されます。これらの計算の詳細を以下に示します。

フィールドのコード化

後述するように、値がアルゴリズムに入力される前に入力フィールドが記録されます。

範囲型フィールドの尺度

ほとんどのデータセットでは、範囲型フィールドの尺度が大幅に異なっています。たとえば、年齢と世帯あたりの自家用車保有数を考えてみましょう。注目する母集団に応じて、年齢は 80 まで、またはそれ以上の値をとります。ただし、世帯あたりの自家用車保有数は、ほとんどのケースで 3 または 4 を超えることはありません。

これらの両方のフィールドをモデルの入力と同じ自然尺度を使用した場合、モデル中で年齢フィールドは、単に「年齢」の値（つまりレコード間の差異）が「世帯あたりの自家用車保有数」より大幅に大きいために、世帯あたりの自家用車保有数よりも大きな重みが与えられることになります。

この尺度による影響を補正するために、範囲型フィールドは同じ尺度を持つように変換されます。IBM® SPSS® Modeler の場合、範囲型フィールドは 0〜1 の範囲の値を持つように再スケールされます。変換には次の式が使用されます。

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}?$$

ここで x'_i はレコード i に対する入力フィールド x の再スケールされた値、x_i はレコード i の x の元の値、x_{min} はすべてのレコードに対する x の最小値、そして x_{max} はすべてのレコードに対する x の最大値を表します。

シンボル値フィールドの数値コード化

レコード間の数値的差異に基づいて計算を行うモデリング アルゴリズムの場合、シンボル値フィールドの処理には特別な労力を費やす必要があります。2 つのカテゴリ間の数値的差異をどのように算出すれば良いでしょうか？

この問題に対する一般的なアプローチ、そして IBM® SPSS® Modeler で使用されているアプローチとして、シンボル値フィールドを数値型フィールドのグループとして記録する方法が挙げられます（元のフィールドの各カテゴリまたは値に対して 1 つの数値型フィールド）。各レコードに対して、レコードのカテゴリに対応する派生フィールドの値は 1.0 に設定されます。また、他のすべての派生フィールドの値は 0.0 に設定されます。このような派生フィールドは、**指標フィールド**と呼ばれることもあります。また、この記録は、**指標のコード化**と呼ばれます。

たとえば、次のデータを例に考えてみましょう。ここで、x は A、B、および C の値をとる可能性があるシンボル値フィールドを表しています。

レコード番号	X	X_1'	X_2'	X_3'
1	B	0	1	0
2	A	1	0	0
3	C	0	0	1

このデータで、元のセット型フィールド x は 3 つの派生フィールド x_1'、x_2'、および x_3' に記録されます。x_1' はカテゴリ A の指標、x_2' はカテゴリ B の指標、および x_3' はカテゴリ C の指標になります。

ダミー変数の調整値の適用

セット型フィールドを記録した後、k 個の派生フィールド（k は元のセット中のカテゴリ数）の差異を取得することにより、セット型フィールドの数値的差異を算出することができます。ただし、ここには問題があります。ユークリッド距離を使ってレコード間の差異を測定するアルゴリズムの場合、異なる値 i と j を持つ 2 つのレコード間の差異は次のようになります。

$$\sqrt{\sum_{k=1}^{J}\left(x_{k1}-x_{k2}\right)^2}$$

ここで J はカテゴリ数を、x_{kn} はレコード n に対するカテゴリ k の派生する指標の値を表します。ただし、派生する 2 つの指標 x_i と x_j の値は異なります。そのため、合計は$\sqrt{(1-0)^2+(0-1)^2}=\sqrt{2}\approx 1.414$ となり、1.0 よりも大きな値になってしまいます。つまり、このコーディングに基づくと、モデルにおいて 0-1 の範囲に再スケールされる範囲型フィールドに比べて、セット型フィールドはより大きい重みを持つことになります。

K-Means ではこのバイアスを調整するため、セット型フィールド上の値の差異がユークリッド距離 1.0 になるように、派生したセット型フィールドに調整値を適用します。デフォルトの調整値は$\sqrt{\frac{1}{2}}\approx 0.707$ です。この値で適切な結果が得られるかどうかを判断するには、次の距離式に値を挿入します。

$$\sqrt{\left(\sqrt{\frac{1}{2}}-0\right)^2+\left(0-\sqrt{\frac{1}{2}}\right)^2}=\sqrt{\frac{1}{2}+\frac{1}{2}}=1$$

調整値の値を変更するには、K-Means ノードの［エキスパート］オプションにある［**ダミー変数の調整値**］パラメータを変更します。

フラグ型フィールドのコード化

フラグ型フィールドはシンボル値フィールドの特殊なケースです。ただし、このフィールドはセット中に 2 つの値しか持っていないため、他のセット型フィールドと比べて多少効率的な方法で処理することができます。フラグ型フィールドは単一の数値型フィールドで表されます。値 1.0 は真（true）の値を、値 0.0 は偽（false）の値を表します。フラグ型フィールドの空白値には、値として 0.5 が割り当てられます。

モデル パラメータ

K-Means の一次計算は、クラスタ中心の計算とクラスタへのレコード割り当ての反復処理です。この処理の主な作業を次に示します。

1. 初期のクラスタ中心を選択します。
2. 各レコードをもっとも近いクラスタに割り当てます。
3. 各クラスタに割り当てられたレコードに基づいてクラスタ中心を更新します。
4. ステップ 2 およびステップ 3 を、次のいずれかの条件を満たすまで繰り返します。
   - ステップ 3 で、前の反復からクラスタ中心の変更がない場合
   - 反復回数が指定された最大反復数を超えた場合

クラスタは中心により定義されます。**クラスタ中心**は、（コード化された）入力フィールドの値のベクトルです。ベクトル値は、クラスタに割り当てられたレコードの値の平均に基づいています。

初期クラスタ中心の選択

ユーザーは、モデル中のクラスタ数を表す k を指定します。初期クラスタ中心は、マキシミン アルゴリズムを使って選択されます。

1. 最初のクラスタ中心を、最初のデータ レコードの入力フィールドの値として初期化します。
2. 各データ レコードに対して、レコードと定義された各クラスタ中心間の最小（ユークリッド）距離を計算します。
3. 定義したクラスタ中心から最大の最小距離を持つレコードを選択します。選択したレコードの入力フィールドの値を持つ、新しいクラスタ中心を追加します。

4. k 個のクラスタ中心がモデルに追加されるまで、ステップ 2 と 3 を繰り返します。

初期クラスタ中心が選択されると、割り当て、更新の反復処理が開始されます。

レコードのクラスタへの割り当て

アルゴリズムの各反復において、それぞれのレコードが、中心がもっとも近いクラスタに割り当てられます。距離は、一般ユークリッド平方距離を用いて計測されます。

$$d_{ij} = ||X_i - C_j||^2 = \sum_{q=1}^{Q} (x_{qi} - c_{qj})^2$$

ここで X_i はレコード i に対するコード化された入力フィールドのベクトル、C_j はクラスタ j のクラスタ中心ベクトル、Q はコード化された入力フィールド数、x_{qi} は i 番目のレコードに対する q 番目のコード化された入力フィールドの値、そして c_{qj} は j 番目のレコードに対する q 番目のコード化された入力フィールドの値を表します。

それぞれのレコードに対して、レコードと各クラスタ中心間の距離が算出され、レコードからの距離がもっとも小さいクラスタ中心を持つクラスタが、そのレコードのクラスタとして新しく割り当てられます。すべてのレコードが割り当てられたら、クラスタ中心が更新されます。

クラスタ中心の更新

レコードをもっとも近いクラスタに割り当て（再割り当て）終わったら、クラスタ中心が更新されます。クラスタ中心は、そのクラスタに割り当てられたレコードのベクトルの平均として算出されます。

$$C_j = \overline{X}_j$$

ここで、ベクトルの平均の構成要素 $\overline{X}_j$ は、通常のように計算されます。

$$\overline{x}_{qj} = \frac{\sum_{i=1}^{n_j} x_{qi}(j)}{n_j}$$

ここで n_j はクラスタ j 中のレコード数、$x_{qi}(j)$ はクラスタ j に割り当てられたレコード i の q 番目のコード化フィールド値を表します。

空白の処理

K-Means で空白は、欠損値に対する「中間の」値で代用することにより処理されます。欠損値（空白およびヌル）を持つ範囲型およびフラグ型フィールドの場合、欠損値は 0.5 で置換されます。セット型フィールドの場合、派生する指標フィールドの値はすべて 0.0 に設定されます。

オプションの効果

モデル計算に影響するさまざまなオプションがあります。

最大反復数

最大反復数は、いつまで安定したクラスタ ソリューションを探し続けるかを定義します。ここに指定された値まで、分類／更新サイクルが繰り返されます。この回数に達した場合、サイクルは終了し、現在のクラスタ セットが最終モデルになります。

誤差許容度（収束基準）

誤差許容度（収束基準）は、別の意味でいつまで安定したクラスタ ソリューションを探し続けるかを定義します。反復 t に対するクラスタ平均中の最大変化は、次のように計算されます。

$$\max_J || C_j(t) - C_j(t-1) ||$$

ここで $C_j(t)$ は反復 t における j 番目のクラスタのクラスタ中心ベクトルを、$C_j(t-1)$ は前の反復におけるクラスタ中心ベクトルを表します。最大変化が現在の反復に指定された許容度未満の場合、処理が終了して、現在のクラスタ セットが最終モデルとして生成されます。

ダミー変数の調整値

ダミー変数の調整値は、K-Means アルゴリズムにおけるセット型フィールドの相対的な重みを定義しています。デフォルト値の $\sqrt{0.5} \approx 0.707$ では、範囲型フィールドとセット型フィールド間の重みが等しくなります。セット型フィールドの重みを強くするには、調整値を 1.0 に近い値に設定します。範囲型フィールドの重みを強くするには、調整値を 0.0 に近い値に設定します。 詳細は、 p. 188 シンボル値フィールドの数値コード化 を参照してください。

モデル要約統計量

クラスタ近接は、クラスタ中心間のユークリッド距離として算出されます。

$$d_{ij} = ||C_i - C_j|| = \sqrt{\sum_{q=1}^{Q} (c_{qi} - c_{qj})^2}$$

生成されたモデル/スコアリング

生成された K-Means モデルは、予測された所属クラスタと、各レコードのクラスタ中心からの距離を提供しています。

予測された所属クラスタ

予測された所属クラスタに新規レコードを割り当てる場合、レコードと各クラスタ中心間のユークリッド距離が計算され（モデル構築フェーズにおけるレコードの割り当てと同じ方法で)、そのレコードにもっとも近いクラスタ中心を持つクラスタが、レコードの予測されたクラスタとして割り当てられます。

距離

各レコードの**距離フィールド**の値が要求された場合、レコードと割り当てられたクラスタ中心間のユークリッド距離として値が算出されます。

$$d_{ij} = ||X_i - C_j|| = \sqrt{\sum_{q=1}^{Q} (x_{qi} - c_{qj})^2}$$

空白の処理

K-Means では、生成されたモデルでのレコードのスコアリングにおける空白の処理は、モデル構築時と同じ方法で処理されます。 詳細は、 p. 190 空白の処理 を参照してください。

章
21

Kohonen アルゴリズム

概要

Kohonen モデル(Kohonen, 2001)は、**非監視学習**を実行する特殊な種類のニューラル ネットワーク モデルです。このアルゴリズムは、入力ベクトルを取得して、空間的に編成されたクラスタリングまたは機能マッピングを実施し、類似レコードをグループ化して、クラスタ間の多次元近接関係に近似した入力空間を 2 次元空間に折りたたみます。

Kohonen ネットワーク モデルは、入力層と出力層の 2 つのニューロンまたはユニットの層から成り立っています。入力層は、完全に出力層に接続されており、それぞれの接続には対応する重みがあります。他のネットワーク構造を把握する方法として、関連する中心を持つ各出力層ユニットを考慮することが考えられます。この中心は、それがもっとも強く応答する入力のベクトルとして表されます（中心ベクトルの各エレメントは、出力ユニットから対応する入力ユニットへの重みになります)。

一次計算

フィールドのコード化

範囲型フィールドの尺度

ほとんどのデータセットでは、範囲型フィールドの尺度が大幅に異なっています。たとえば、年齢と世帯あたりの自家用車保有数を考えてみましょう。注目する母集団に応じて、年齢は 80 まで、またはそれ以上の値をとります。ただし、世帯あたりの自家用車保有数は、ほとんどのケースで 3 または 4 を超えることはありません。

これらの両方のフィールドをモデルの入力と同じ自然尺度を使用した場合、モデル中で年齢フィールドは、単に「年齢」の値（つまりレコード間の差異）が「世帯あたりの自家用車保有数」より大幅に大きいために、世帯あたりの自家用車保有数よりも大きな重みが与えられることになります。

この尺度による影響を補正するために、範囲型フィールドは同じ尺度を持つように変換されます。IBM® SPSS® Modeler の場合、範囲型フィールドは 0〜1 の範囲の値を持つように再スケールされます。変換には次の式が使用されます。

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}?$$

ここで x'_i はレコード i に対する入力フィールド x の再スケールされた値、x_i はレコード i の x の元の値、x_{min} はすべてのレコードに対する x の最小値、そして x_{max} はすべてのレコードに対する x の最大値を表します。

シンボル値フィールドの数値コード化

レコード間の数値的差異に基づいて計算を行うモデリング アルゴリズムの場合、シンボル値フィールドの処理には特別な労力を費やす必要があります。2 つのカテゴリ間の数値的差異をどのように算出すれば良いでしょうか？

この問題に対する一般的なアプローチ、そして IBM® SPSS® Modeler で使用されているアプローチとして、シンボル値フィールドを数値型フィールドのグループとして記録する方法が挙げられます（元のフィールドの各カテゴリまたは値に対して 1 つの数値型フィールド）。各レコードに対して、レコードのカテゴリに対応する派生フィールドの値は 1.0 に設定されます。また、他のすべての派生フィールドの値は 0.0 に設定されます。このような派生フィールドは、**指標フィールド**と呼ばれることもあります。また、この記録は、**指標のコード化**と呼ばれます。

たとえば、次のデータを例に考えてみましょう。ここで、x は A、B、および C の値をとる可能性があるシンボル値フィールドを表しています。

レコード番号	X	X_1'	X_2'	X_3'
1	B	0	1	0
2	A	1	0	0
3	C	0	0	1

このデータで、元のセット型フィールド x は 3 つの派生フィールド x_1'、x_2'、および x_3' に記録されます。x_1' はカテゴリ A の指標、x_2' はカテゴリ B の指標、および x_3' はカテゴリ C の指標になります。

フラグ型フィールドのコード化

フラグ型フィールドはシンボル値フィールドの特殊なケースです。ただし、このフィールドはセット中に 2 つの値しか持っていないため、他のセット型フィールドと比べて多少効率的な方法で処理することができます。フラグ型フィールドは単一の数値型フィールドで表されます。値 1.0 は真（true）の値を、値 0.0 は偽（false）の値を表します。フラグ型フィールドの空白値には、値として 0.5 が割り当てられます。

モデル パラメータ

Kohonen モデルでは、入力ユニットと出力ユニット間の**重み**として、または代わりに各出力ユニットに関連する**クラスタ中心**として、パラメータが表されます。入力レコードはネットワークに渡され、K-Means モデルの構築と同じような方法で、クラスタ中心が更新されます。ただし、クラスタは2次元グリッドに空間的に配置され、各レコードは割り当てられているユニット（クラスタ）に影響するだけでなく、勝ち取ったユニットに関する**隣接**内のユニットにも影響する点で異なっています。 詳細は、 p. 195 隣接 を参照してください。

Kohonen ネットワークの学習処理は、次のように行われます。

- ネットワークが、無作為の小さな重みで初期化されます。

- 入力レコードが、無作為の順序でネットワークに渡されます。各レコードが渡されると、入力ベクトルに対してもっとも近い中心を持つ出力ユニットが、勝ち取ったユニットと判断されます。詳細は、 p. 195 距離 を参照してください。
- クラスタ中心が入力ベクトルに近づくように、勝ち取ったユニットの重みが調整されます。 詳細は、 p. 196 重みの更新 を参照してください。
- 隣接サイズが 0 より大きい場合は、勝ち取ったユニットの隣接内に存在する他の出力ユニットも、中心が入力ベクトルに近づくように更新されます。
- 各サイクルの終わりには、学習率パラメータη（イータ）が更新されます。
- この処理が、いずれかの停止基準を満たすまで繰り返されます。学習は、全体構造フェーズと微調整フェーズの 2 つのフェーズに分かれています。通常、最初のフェーズでは相対的に大きい隣接サイズおよび大きいηを持ち、データの全体的な構造を学習します。2 番目のフェーズでは、小さい隣接と小さいηを使用して、クラスタ中心を微調整します。

距離

Kohonen ネットワーク中の距離は、コード化された入力ベクトルと出力ユニットのクラスタ中心間のユークリッド距離として算出されます。

$$d_{ij} = \sqrt{\sum_k \left(x_{ik} - w_{jk}\right)^2}$$

ここで x_{ik} は ith 番目のレコードの k 番目の入力フィールドの値を、w_{jk} は j 番目の出力ユニット上の k 番目の入力フィールドの重みを表します。

出力ユニットの活性度は、単に出力ユニットの重みベクトル（中心）と入力ベクトル間のユークリッド距離になります。Kohonen ネットワークでは、もっとも小さい活性度を持つ出力ユニットが勝ち取ったユニットになることに注意してください。これは、大きい活性度が強い応答を表す他の種類のニューラル ネットワークとは異なっています。 詳細は、 25 章 p. 238 概要 を参照してください。

隣接

隣接関数は、チェビシェフ距離に基づいています。チェビシェフ距離では、任意の単一次元上の最大距離だけが考慮されます。

$$d_c\left(x, y\right) = \max_i \left|x_i - y_i\right|$$

ここで x_i は出力グリッドの次元 i にあるユニット x の場所を、y_i は同じ次元上にある他のユニット y の場所を表します。

$d_c\left(o_i, o_j\right) < n$ の場合、出力ユニット o_j は他の出力ユニット o_i に隣接しているとみなされます。ここで n は隣接サイズを表します。

各フェーズ中、隣接サイズは一定です。ただし、一般的に別のフェーズでは、別の隣接サイズが使用されます。デフォルトでは、フェーズ 1 の場合 $n = 2$、フェーズ 2 の場合 $n = 1$ になります。

重みの更新

勝ち取った出力ノードでは（および、隣接が > 0 の場合はその隣接も）、入力ベクトルと現在の重みベクトル間の差異の部分を追加することにより、重みが調整されます。変更の絶対値は、学習率パラメータη（イータ）により判断されます。重みの変更は、次のように算出されます。

$$\Delta W = \eta \cdot (W - I)$$

ここで、W は更新対象出力ユニットの重みベクトル、I は入力ベクトル、そして η は学習率パラメータを表します。各ユニットの項は、次のようになります。

$$\Delta w_j = \eta \cdot (w_j - i_j)$$

ここで w_j は更新対象出力ユニットに対する入力ユニット j に対応する重みを、i_j は j 番目の入力ユニットを表します。

η の減衰

各サイクルの最後には、ηの値が更新されます。通常は、学習サイクルに渡ってηの値は減少していきます。ユーザーは、線型減衰または指数減衰を選択することにより、この減少率を制御することができます。

線型減衰。デフォルトの減衰率です。このオプションを選択した場合、ηの値は線型に減少します。次の式により、各サイクルごとに一定量が減少します。

$$\eta(t+1) = \eta(t) - \left(\frac{\eta(0) - \eta_{low}}{c} \right)$$

ここで $\eta(0)$ は現在のフェーズにおける初期 η の値を、η_{low} は現在の学習フェーズにおける最小 η を表します。最小 η は、現在のフェーズと次のフェーズにおける初期 η 値の小さい方として計算されます。また、c は現在のフェーズに設定されているサイクル数を示します。

指数減衰。このオプションを選択した場合、ηの値は指数的に減少します。次の式により、各サイクルごとに一定比率が減少します。

$$\eta(t+1) = \eta(t) \cdot \exp\left(\frac{\log\left(\frac{\eta_{low}}{\eta(0)}\right)}{c} \right)$$

η_{low} の値は、対数取得時の数学的誤差を防ぐために、最小値 0.0001 となっています。

空白の処理

Kohonen ネットワークで空白は、欠損値に対する「中間の」値で代用することにより処理されます。欠損値（空白およびヌル）を持つ範囲型およびフラグ型フィールドの場合、欠損値は 0.5 で置換されます。範囲型フィールドの場合、フィールド

のデータ型情報中に見つかった範囲制限外の数値は、データ型に定義された範囲に強制変換されます。セット型フィールドの場合、派生する指標フィールドの値はすべて 0.0 に設定されます。

オプションの効果

停止条件。デフォルトでは、各フェーズに対して指定されたサイクル数の学習が実行されます。[**時間**] オプションが選択されている場合、指定された時間が経過した場合（または、指定された時間が経過する前に、両方のフェーズにおいて指定されたサイクル数が完了した場合）に学習が停止されます。

ランダム シード。乱数ジェネレータが新しいネットワークの重みや、学習レコードを渡す順序を初期化するために使用するシードを設定します。再現可能なネットワークを作成する場合は、シードに対して固定値を指定します。

生成されたモデル/スコアリング

所属クラスタ

新規レコードの所属クラスタは、レコードの入力ベクトルをネットワークに渡して、もっとも近い重みベクトルを持つ出力ニューロンを識別することにより決められます（前述の「距離」を参照）。予測フィールドは、出力グリッド中の勝ち取ったニューロンの x および y 座標として返されます。

空白の処理

スコアリングの空白処理は、モデル構築時の空白処理と同じです。 詳細は、p. 196 空白の処理 を参照してください。

章
22

ロジスティック回帰アルゴリズム

ロジスティック回帰モデル

ロジスティック回帰は、二項または多項結果を予測するための確立された統計手法です。IBM® SPSS® Modeler では、次のようなロジスティック回帰モデリングの 2 つのアルゴリズムを提供しています。

多項ロジスティック: これは SPSS Modeler で使用されている元からのロジスティック回帰のアルゴリズムで、バージョン 6.0 で導入されました。このアルゴリズムは、対象フィールドが 2 つ以上の値をとるセット型フィールドの場合にモデルを作成します。詳細は以下を参照してください。さらに、新しい二項ロジスティックアルゴリズムのようなモデルと同じレベルの詳細な統計値は作成しませんが、フラグ型つまり二者択一の結果のモデルも作成できます。

二項ロジスティック :SPSS Modeler 11 で導入されたこのアルゴリズムは、対象フィールドがフラグ型、つまり 2 進数フィールドのモデルに限定されます。このアルゴリズムは、多項ロジスティックのアルゴリズムの出力と比較して統計量の出力が強化され、セルの数（予測フィールド値の一意の組み合わせ）がレコード数に比べて多い場合に、問題の影響を受けにくくなります。 詳細は、 p. 206 二項ロジスティック回帰 を参照してください。

フラグ型出力フィールドのあるモデルの場合、ロジスティック アルゴリズムの選択は [**プロシージャ**] オプションによりモデル ノード内で制御されます。

多項ロジスティック回帰

多項ロジスティック回帰の目的は、一連のシンボル値フィールドや数値予測（入力）フィールドの名義（シンボル値）出力フィールドの依存関係（関連性）をモデル化することにあります。

一次計算

フィールドのコード化

ロジスティック回帰では、各シンボル値（セット型）フィールドが数値フィールドのグループとして記録されます（元のフィールドの各カテゴリまたは値に対して 1 つの数値フィールド、ただし最後のカテゴリは除く）。最後のカテゴリは参照カテゴリとして定義されます。各レコードに対して、レコードのカテゴリに対応する派生フィールドの値は 1.0 に設定されます。また、他のすべての派生フィールドの値は 0.0 に設定されます。参照カテゴリの値を持つレコードでは、すべての派生フィールドに 0.0 が設定されます。このような派生フィールドは、**ダミー フィールド**と呼ばれることもあります。また、この記録は、**ダミー コーディング**と呼ばれます。

たとえば、次のデータを例に考えてみましょう。ここで、x は A、B、および C の値をとる可能性があるシンボル値フィールドを表しています。

レコード番号	X	X_1'	X_2'
1	B	0	1
2	A	1	0
3	C	0	0

このデータで、元のセット型フィールド x は、2 つの派生フィールド x_1' および x_2' に記録されます。x_1' はカテゴリ A の指標で、x_2' はカテゴリ B の指標です。最後のカテゴリとなるカテゴリ C は参照カテゴリになります。このカテゴリに所属するレコードは、x_1' とx_2' の両方に 0.0 が設定されます。

表記

この章では特に明記しない限り、次の表記を使用します。

Y	出力フィールドで、1 ～ J の値をとります。
J	出力フィールドのカテゴリ数。
m	部分母集団数。
$\mathbf{X}^A$	ベクトル要素 x_i^A を持つ $m \times p^A$ 行列、ベクトル要素は部分母集団の i 番目の観測値で、コマンドに指定された入力フィールドにより決まります。
$\mathbf{X}$	ベクトル要素 x_i^A を持つ $m \times p$ 行列、ベクトル要素は i 番目の部分母集団における配置モデルの入力フィールドの観測値です。
n_{ij}	部分母集団 i の、$Y = j$ に対応するセルに所属する観測の度数の重みの合計。
N	すべての n_{ij}' の合計。
π_{ij}	部分母集団 i の $Y = j$ に対応するセルの確率。
$\log(\pi_{ij}/\pi_{ik})$	応答カテゴリ k と相対的な応答カテゴリ j のロジット。
$\beta_j = (\beta_{j1}, ..., \beta_{jp})'$	j 番目のロジット中の不明なパラメータの $p \times 1$ ベクトル（つまり、応答カテゴリ j から応答カテゴリ J へのロジット）。
p	各ロジット中のパラメータ数。$p \geq 1$。
p_j^{nr}	最尤推定後の、ロジット j 中の非冗長パラメータ数。$p \geq p_j^{nr} \geq 0$。
p^{nr}	最尤推定後の、非冗長パラメータの総数。$p^{nr} = \sum_{j=1}^{k-1} p_j^{nr}$。
$\mathbf{B} = (\beta'_1, ..., \beta'_{J-1})'$	モデル中の不明なパラメータの $(k-1)p \times 1$ ベクトル。
$\hat{\mathbf{B}} = \left(\hat{\beta}'_1, ..., \hat{\beta}'_{J-1}\right)'$	$\mathbf{B}$ の最尤推定。
$\hat{\pi}_{ij}$	π_{ij} の最尤推定。

データの集計

観測値は、部分母集団の定義により集計されます。部分母集団は、入力フィールドのセットの交差分類により定義されます。

n_i が部分母集団 i の限界度数とします。

$$n_i = \sum_{j=1}^{k} n_{ij}$$

部分母集団 i において、$Y = j$ のセルに対して観測値がない場合、$n_i \neq 0$ ならば $n_{ij} = 0$ と仮定されます。限界度数 n_i がゼロ以外の場合、任意のゼロのセル（つまり $n_{ij} = 0$ のセル）に非負スカラー $\delta \in [0, 1)$ が追加されることがあります。デフォルトでは δ の値はゼロになります。

一般化ロジット モデル

一般化ロジット モデルでは、部分母集団 i における応答カテゴリ j の確率 π_{ij} は次のようになります。

$$\pi_{ij} = \frac{\exp\left(\mathbf{x'}_i \beta_j\right)}{1 + \sum_{k=1}^{J-1} \exp\left(\mathbf{x'}_i \beta_k\right)}$$

ここで最後のカテゴリ J は、参照カテゴリと仮定されます。

ロジットの観点から、モデルは次のように表現することができます。

$$\log\left(\frac{\pi_{ij}}{\pi_{iJ}}\right) = \mathbf{x'}_i \beta_j$$

j = 1、…、J-1 になります。

J = 2 の場合、このモデルは 2 項ロジスティック回帰モデルと同じになります。そのため、上記のモデルは 2 項のレスポンスから多項名義レスポンスへの、2 項ロジスティック回帰モデルの延長と考えられます。

対数尤度

モデルの対数尤度は次の式により生じます。

$$\begin{aligned} l(\mathbf{B}) &= \sum_{i=1}^{m} \sum_{j=1}^{J} n_{ij} \log\left(\pi_{ij}\right) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{J} n_{ij} \log\left(\frac{\exp\left(\mathbf{x'}_i \beta_j\right)}{1 + \sum_{k=1}^{J-1} \exp\left(\mathbf{x'}_i \beta_l\right)}\right) \end{aligned}$$

ここでは、パラメータに依存しない定数は除外されています。定数の値は $c = \sum_{i=1}^{m} \log\left(n_i!/(n_{i1}!...n_{iJ}!)\right)$ となります。

モデル パラメータ

対数尤度の導関数

任意の j = 1、…、J-1、s = 1、…、p に対して、β_{js} に関する l の一次導関数は次のようになります。

$$\frac{\partial l}{\partial \beta_{js}} = \sum_{i=1}^{m} x_{is}(n_{ij} - n_i \pi_{ij}).$$

任意の j、j' = 1、…、J-1 および s、t = 1、…、p に対して、β_{js} および $\beta_{j't}$ に関する l の二次導関数は次のようになります。

$$\frac{\partial^2 l}{\partial \beta_{js} \partial \beta_{j't}} = -\sum_{i=1}^{m} n_i x_{is} x_{it} \pi_{ij}(\delta_{jj'} - \pi_{ij'})$$

ここで $\delta_{jj'} = 1$ if $j = j'$、それ以外の場合は 0 になります。

最尤推定

B の最尤推定を取得するために、Newton-Raphson 反復推定法が用いられます。この方法は、 **B** に関する l の二次導関数の期待値が観測対象の期待値と同じため、このモデルの Fisher-Scoring 反復推定法と同じになります。

$\partial l/\partial \mathbf{B}$ が、**B** に関する l の一次導関数の $(J-1)p \times 1$ ベクトルとします。さらに、$\left[\partial^2 l/\partial \mathbf{B} \partial \mathbf{B}\right]$ が、**B** に関する二次導関数 l の $(J-1)p \times (J-1)p$ 行列とします。$-\left[\partial^2 l/\partial \mathbf{B} \partial \mathbf{B}\right] = \sum_{i=1}^{m} \mathbf{X}_i^* \Delta_i \mathbf{X}_i^{*\prime}$ ここで、Δ_i は次のような $(J-1) \times (J-1)$ 行列になります。

$$\Delta_i = n_i\left(Diag\left(\pi_i^{(-J)}\right) - \pi_i^{(-J)} \pi_i^{(-J)\prime}\right)$$

$\pi_i^{(-J)} = (\pi_{i1}, ..., \pi_{iJ-1})'$ および $Diag\left(\pi_i^{(-J)}\right)$ は $\pi_i^{(-J)}$ の β_{js} 対角行列を表します。$B^{(\nu)}$ が反復 ν におけるパラメータ推定値とすると、反復 $\nu+1$ におけるパラメータ推定値 $B^{(\nu+1)}$ は次のように更新されます。

$$\mathbf{B}^{(\nu+1)} = \mathbf{B}^{(\nu)} + \xi\left(\sum_{i=1}^{m} \mathbf{X}_i^* \Delta_i^{(\nu)} \mathbf{X}_i^{*\prime}\right) \frac{\partial l}{\partial \mathbf{B}^{(\nu)}}$$

また $\xi > 0$ は $l\left(\mathbf{B}^{(\nu+1)}\right) - l\left(\mathbf{B}^{(\nu)}\right) \geq 0$ となるようなステップ基準スカラーで、$\mathbf{X}^*$ は独立ベクトルの $(J-1)p \times (J-1)$ 行列になります。

$$\mathbf{X}_i^* = \begin{pmatrix} \mathbf{x}_i & 0 & ... & 0 \\ 0 & \mathbf{x}i & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & ... & 0 & \mathbf{x}i \end{pmatrix}$$

また $\Delta_i^{(\nu)}$ は Δ_i、そして $\partial l/\partial \mathbf{B}^{(\nu)}$ は $\partial l/\partial \mathbf{B}$ となり、両方とも $\mathbf{B} = \mathbf{B}^{(\nu)}$ で評価されます。

ステップ基準

$l\left(\mathbf{B}^{(\nu+1)}\right) - l\left(\mathbf{B}^{(\nu)}\right) < 0$ の場合、段階 2 分法を使用します。V が段階 2 分法におけるステップの最大数とすると、ξ の値のセットは $\{1/2^\nu : \nu = 0, ..., V-1\}$ となります。

パラメータの開始値

モデルに定数項が含まれている場合は、$\beta_j^{(0)} = \left(\beta_{j1}^{(0)}, 0, ..., 0\right)'$ を設定します。この場合、以下のようになります。

$$\beta_{j1}^{(0)} = \log\left(\frac{\tilde{\pi}_{ij}}{\tilde{\pi}_{iJ}}\right) = \log\left(\frac{\sum_{i=1}^{m} n_{ij}}{\sum_{i=1}^{m} n_{iJ}}\right)$$

j = 1、…、J-1 になります。

モデルに定数項が含まれない場合は、次のように設定します。

$$\beta_j^{(0)} = (0, ..., 0)'$$

j = 1、…、J-1 になります。

収束基準

2 つの収束基準 $\epsilon_k > 0$ および $\epsilon_p > 0$ が与えられると、次のいずれかの基準を満たした場合に反復は収束したとみなされます。

1. $\left|l\left(\mathbf{B}^{(\nu+1)}\right) - l\left(\mathbf{B}^{(\nu)}\right)\right| < \epsilon_k$
2. $\max_i \left|\mathbf{B}_i^{(\nu+1)} - \mathbf{B}_i^{\nu}\right| < \epsilon_p$
3. $\partial l/\partial \mathbf{B}^{(\nu+1)}$ 中の上記の要素の最大値が $\min(\epsilon_k, \epsilon_p)$ 未満。

乖離の確認

ロジスティック回帰では、反復 ν^{chksep}（デフォルトは 20）から始まるデータの乖離が確認されます。乖離を確認するには

1. 各部分母集団 i に対して、$j^*:\hat{\pi}_{ij^*}=\max_j(\hat{\pi}_{ij})$ を検出します。
2. $n_{ij^*}=n_i$ の場合は、部分母集団 i に対する完全な予測があります。
3. すべての部分母集団に完全な予測がある場合、完全な乖離があることになります。一部のパターンに完全な予測があり、$\hat{\mathbf{B}}$ の Hessian が単数の場合、疑似完全乖離があることになります。

空白の処理

入力フィールドまたは出力フィールドに欠損値があるすべてのレコードは、モデルの推定から除外されます。

二次計算

モデル要約統計量

対数尤度

切片のある初期モデル。モデルに切片が含まれている場合、初期モデル（切片だけのモデル）の予測確率は次のようになります。

$$\tilde{\pi}_{ij}=\frac{\sum_{i=1}^{m}n_{ij}}{N}$$

また、初期モデルの -2 対数尤度の値は次のようになります。

$$-2l(\tilde{\pi})=-2\sum_{i=1}^{m}\sum_{j=1}^{J}n_{ij}\log(\tilde{\pi}_{ij}).$$

切片のない初期モデル。モデルに切片が含まれていない場合、初期モデルの予測確率は次のようになります。

$$\tilde{\pi}_{ij}=\frac{1}{J}$$

また、初期モデルの -2 対数尤度の値は次のようになります。

$$-2l(\tilde{\pi})=-2N\log\left(\frac{1}{J}\right).$$

最終モデル。最終モデルの 2 対数尤度の値は次のようになります。

$$-2l(\tilde{\pi}) = -2\sum_{i=1}^{m}\sum_{j=1}^{J} n_{ij}\log\left(\hat{\pi}_{ij}\right)$$

カイ 2 乗モデル

カイ 2 乗モデルは次の式により与えられます。

$$-2l(\tilde{\pi}) - \{-2l(\hat{\pi})\}$$

最終モデルに切片が含まれる場合、初期モデルは切片だけのモデルになります。帰無仮説 $H_0 : \beta^{intercepts} = \mathbf{0}$ とすると、カイ 2 乗モデルは $p^{nr} - (J-1)$ の自由度を持つ漸近カイ 2 乗分布になります。

モデルに切片が含まれない場合、初期モデルは空のモデルになります。帰無仮説 $H_0 : \beta = \mathbf{0}$ とすると、カイ 2 乗モデルは p^{nr} の自由度を持つ漸近カイ 2 乗分布になります。

擬似 R2 乗法

Cox と Snell。Cox と Snell の R^2 乗は、次のように計算されます。

$$R^2_{CS} = 1 - \left(\frac{L(\tilde{\pi})}{L(\hat{\pi})}\right)^{\frac{2}{n}}.$$

Nagelkerke。Nagelkerke の R^2 乗は次のように計算されます。

$$R^2_N = \frac{R^2_{CS}}{1 - L(\tilde{\pi})^{2/n}}.$$

McFadden。McFadden の R^2 乗は次のように計算されます。

$$R^2_M = 1 - \left(\frac{l(\hat{\pi})}{l(\tilde{\pi})}\right).$$

適合度

Pearsonの相関係数。Pearson の適合度は次のようになります。

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{J}\frac{\left(n_{ij} - n_i\hat{\pi}_{ij}\right)^2}{n_i\hat{\pi}_{ij}}.$$

帰無仮説では、Pearson の適合度統計は $m(J-1) - p^{nr}$ の自由度を持つ漸近カイ 2 乗分布になります。

逸脱。逸脱適合度は次のようになります。

$$D = 2\sum_{i=1}^{m}\sum_{j=1}^{J} n_{ij}\log\left(\frac{n_{ij}}{n_i\hat{\pi}_{ij}}\right).$$

帰無仮説では、逸脱適合度統計は $m(J-1)-p^{nr}$ の自由度を持つ漸近カイ 2 乗分布になります。

フィールド統計量および他の計算

ロジスティック（式）ノードの詳細出力に表示されている統計量は、IBM® SPSS® Statistics の NOMREG プロシージャと同じ方法で計算されます。詳細は、『SPSS Statistics Nomreg algorithm』を参照してください。このドキュメントは、http://www.ibm.com/support から入手できます。

生成されたモデル／スコアリング

予測値

レコード i の予測フィールドは、最大のロジット値 r_{ij} を持つ出力フィールド カテゴリ j になります。

$$r_{ij} = \log\left(\frac{\pi_{ij}}{\pi_{iJ}}\right) = \mathbf{x'}_i\beta_j$$

j = 1, ..., J-1 になります。参照カテゴリ J, r_{iJ} のロジットは 1.0 です。

予測確率

スコアリングされたレコード i の予測カテゴリ j^* は、カテゴリ j^* のロジットから生成されます。

$$\hat{\pi}_{ij} = \frac{\exp\left(r_{ij'}\right)}{1+\sum_{k=1}^{J-1}\exp\left(r_{ij'}\right)} = \frac{\exp\left(\mathbf{x'}_i\beta_j\right)}{1+\sum_{k=1}^{J-1}\exp\left(\mathbf{x'}_i\beta_k\right)}$$

[すべての確率を追加] オプションが選択されている場合、すべての J カテゴリに対して同じ方法で確率が算出されます。

空白の処理

入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールドおよび確率値に $null$ が割り当てられます。

二項ロジスティック回帰

二項モデル（対象としてフラグ型フィールドのあるモデル）の場合、IBM® SPSS® Modeler は、ここで説明したようにそのようなモデルに対して最適化されたアルゴリズムを使用します。

表記

この章では特に明記しない限り、次の表記を使用します。

n	観察されるケースの数
p	パラメータの数
y	要素 y_i（二分従属変数の i 番目の観察される値）がある $n \times 1$ ベクトル
X	要素 x_{ij}（j 番目のパラメータの i 番目のケースの観察される値）のある $n \times p$ クロス集計
β	要素 β_j（j 番目のパラメータの係数）のある $p \times 1$ ベクトル
w	要素 w_i（i 番目のケースの重み）のある $n \times 1$ ベクトル
l	尤度関数
L	対数-尤度関数
I	情報クロス集計

モデル

線形ロジスティック モデルでは、確率 π の二分従属変数 Y を仮定します。ここで i 番目のケースでは次の式が成り立ちます。

$$\pi_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$$

または

$$\ln\left(\frac{\pi_i}{1-\pi_i}\right) = \eta_i = \mathbf{X}_i'\beta$$

このために、n 個の観察 $y_1, \ldots, y_n$、確率 $\pi_1, \ldots, \pi_n$ およびケースの重み $w_1, \ldots, w_n$ である尤度関数 l は、次のように書くことができます。

$$l = \prod_{i=1}^{n} \pi_i^{w_i y_i} (1-\pi_i)^{w_i(1-y_i)}$$

これは、l のアルゴリズムが次のものであることに従います。

$$L = \ln(l) = \sum_{i=1}^{n} (w_i y_i \ln(\pi_i) + w_i(1-y_i)\ln(1-\pi_i))$$

そして、β_j に関して L の導関数は次のようになります。

$$L^*_{X_j} = \frac{\partial L}{\partial \beta_j} = \sum_{i=1}^{n} w_i(y_i - \pi_i)x_{ij}$$

最尤推定量 (MLE)

β の最尤推定量は、次の方程式を満たします。

j 番目のパラメータについて、$\sum_{i=1}^{n} w_i(y_i - \hat{\pi}_i)x_{ij} = 0$

ここで、$i = 1, \ldots, n$ の場合、$x_{i0} = 1$

次の事項に注意してください。

1. ニュートン-ラフスン型アルゴリズムが、MLE を得るために使用されます。収束は次のものに基づきます。
   - 反復間のパラメータ推定値の絶対差分
   - 連続反復間の対数尤度関数の割合差分
   - 指定される反復の最大数
2. 反復の中で、すべてのケースで $\hat{\pi}_i(1 - \hat{\pi}_i)$ が 10^{-8} 未満である場合は、対数尤度関数は、ゼロに非常に近づきます。この状況になると、反復が停止し、メッセージ「予想されるすべての値は、1 または 0 です」が発行されます。 “”

最大尤度推定量 $\hat{\beta}$ を得た後で、I^{-1}(情報クロス集計 I の逆数）によって、漸近分散共分散クロス集計を推定します。ここでは次のようになります。

$$I = -\left[E\left(\frac{\partial^2 L}{\partial \beta_i \partial \beta_j}\right)\right] = \mathbf{X}'\mathbf{W}\hat{\mathbf{V}}\mathbf{X},$$

$$\hat{\mathbf{V}} = Diag\{\hat{\pi}_1(1 - \hat{\pi}_1), \ldots, \hat{\pi}_n(1 - \hat{\pi}_n)\},$$

$$\mathbf{W} = Diag\{w_1, \ldots, w_n\},$$

$$\hat{\pi}_i = \frac{\exp(\hat{\eta}_i)}{1 + \exp(\hat{\eta}_i)},$$

および

$$\hat{\eta}_i = \mathbf{X}_i{}'\hat{\beta}$$

ステップワイズ変数選択

独立変数の選択のために、いくつもの方法が使用できます。強制投入法では、変数リストの中の任意の変数が、モデルに投入されます。ステップワイズ法には、変数増加法と変数減少法の 2 種類があります。ステップワイズ法は、ワルド統計、尤度比、または変数除去の条件アルゴリズムのいずれかを使用します。両方のステップワイズ法で、スコア統計がモデルに投入する変数選択に使用されます。

変数増加ステップワイズ法 (FSTEP)

1. FSTEP が最初に要求される方法である場合は、初期モデルのためにパラメータおよび尤度関数を推定します。こうしない場合は、以前の方法の最後のモデルが FSTEP の初期モデルになります。現在のモデルのパラメータの MLE、予測確率、現在のモデルの尤度関数などの必須情報を取得します。
2. 現在のモデルの MLE に基づいて、含有対象のすべての変数に関してスコア統計を計算し、その有意確率を見つけます。
3. 最小有意確率の変数を選択します。この有意確率が、投入する変数の確率未満である場合は、ステップ 4 に進みます。それ以外の場合は、FSTEP を中止します。
4. 新しい変数を追加して、現在のモデルを更新します。こうすることで、すでに評価されたモデルになった場合は、FSTEP を中止します。
5. 現在のモデルの中の各変数について、LR またはワルド統計または条件統計を計算します。それから、その相当する有意確率を計算します。
6. 最大有意確率の変数を選択します。この有意確率が変数除去の確率未満である場合は、ステップ 2 に戻ります。そうでない場合は、除去される変数がある現在のモデルが以前のモデルと同じである場合は、FSTEP を中止します。そうでない場合は、次のステップに進みます。
7. 以前のモデルから最大有意確率のある変数を除去することで、現在のモデルを修正します。修正したモデルについてパラメータを推定し、ステップ 5 に戻ります。

変数減少ステップワイズ法 (BSTEP)

1. 以前の方法からの最終モデルおよびすべての対象変数を含む完全なモデルについて、パラメータを推定します。BSTEP変数リストに一覧された変数だけが、投入および除去の対象です。現在のモデルを完全なモデルにします。
2. 現在のモデルの MLE に基づいて、モデルの中のすべての変数について LR またはワルド統計または条件統計を計算し、その有意確率を見つけます。
3. 最大有意確率の変数を選択します。この有意確率が変数除去の確率未満である場合は、ステップ 5 に進みます。そうでない場合は、最大有意確率のある変数を省いた現在のモデルが以前のモデルと同じである場合は、BSTEP を中止します。そうでない場合は、次のステップに進みます。
4. モデルから最大有意確率のある変数を除去することで、現在のモデルを修正します。修正したモデルについてパラメータを推定し、ステップ 2 に戻ります。
5. 何らかの対象の変数がモデルの中にないかを調べます。存在しない場合は、BSTEP を中止します。存在する場合は、次のステップに進みます。
6. 現在のモデルの MLE に基づいて、モデルの中にないすべての変数に関してスコア統計を計算し、その有意確率を見つけます。
7. 最小有意確率の変数を選択します。この有意確率が、変数投入の確率未満である場合は、次のステップに進みます。それ以外の場合は、FSTEP を中止します。

8. 現在のモデルに最小有意確率のある変数を追加します。モデルが以前のどのモデルとも同じでない場合は、新しいモデルのパラメータを推定し、ステップ 2 に戻ります。そうでない場合は、BSTEP を中止します。

ステップワイズ統計

ステップワイズ変数選択法で使用される統計は、次のように定義されます。

スコア統計

スコア統計は、変数がモデル中に入るべきかどうかを判断するために、モデルの中ではなく、各変数について計算されます。r_1 個の変数、すなわち、$\alpha_1, \ldots, \alpha_{r_1}$ がモデルの中にあり、r_2 個の変数、$\gamma_1, \ldots, \gamma_{r_2}$ がモデルの中にないと、仮定します。γ_i のスコア統計は、次のように定義されます。

$$\mathbf{S}_i = \left(\mathbf{L}^*_{\gamma_i}\right)^2 \mathbf{B}_{22,i}$$

これは、γ_i が、カテゴリ変数でない場合です。γ_i は、m カテゴリのあるカテゴリ変数である場合、$(m-1)$ 次元ダミー ベクトルに変換されます。これらの新しい $m-1$ 個の変数を $\tilde{\gamma}_i, \ldots, \tilde{\gamma}_{i+m-2}$ として表示します。γ_i のスコア統計は、このとき次のようになります。

$$\mathbf{S}_i = \left(\mathbf{L}^*_{\tilde{\gamma}}\right)' \mathbf{B}_{22,i} \mathbf{L}^*_{\tilde{\gamma}}$$

ここで $\left(\mathbf{L}^*_{\tilde{\gamma}}\right)' = \left(L^*_{\tilde{\gamma}_i}, \ldots, L^*_{\tilde{\gamma}_{i+m-2}}\right)$ および $(m-1) \times (m-1)$ クロス集計 $\mathbf{B}_{22,i}$ は次のとおりです。

$$\mathbf{B}_{22,i} = \left(\mathbf{A}_{22,i} - \mathbf{A}_{21,i}\mathbf{A}_{11}^{-1}\mathbf{A}_{12,i}\right)^{-1}$$

を

$$\begin{aligned}
\mathbf{A}_{11} &= \underset{\sim}{\alpha}' \hat{\mathbf{V}} \underset{\sim}{\alpha}, \\
\mathbf{A}_{12,i} &= \underset{\sim}{\alpha}' \hat{\mathbf{V}} \underset{\sim}{\gamma}_i, \\
\mathbf{A}_{22,i} &= \underset{\sim}{\gamma}_i' \hat{\mathbf{V}} \underset{\sim}{\gamma}_i
\end{aligned}$$

ここで$\underset{\sim}{\alpha}$ は、変数 $\alpha_1, \ldots, \alpha_{r_1}$ の計画行列であり、$\underset{\sim}{\gamma}_i$ は、ダミー変数 $\tilde{\gamma}_i, \ldots, \tilde{\gamma}_{i+m-2}$ の計画行列です。ただし、$\underset{\sim}{\alpha}$ は、定数項が η から除外されない限り、1 の列を含みます。モデルの中のパラメータの MLE に基づいて、$\hat{\mathbf{V}}$ は、$\hat{\mathbf{V}} = Diag\{\hat{\pi}_1(1-\hat{\pi}_1), \ldots, \hat{\pi}_n(1-\hat{\pi}_n)\}$ によって推定されます。スコア統計の漸近分布は、関係する変数の数に等しい自由度のあるカイ 2 乗検定です。

次の事項に注意してください。

1. モデルが原点を通り、モデルの中に変数がない場合は、$\mathbf{B}_{22,i}$ は、$\mathbf{A}_{22,i}^{-1}$ によって定義され、$\hat{\mathbf{V}}$ は$\frac{1}{4}\mathbf{I}_n$ に等しくなります。

2. $\mathbf{B}_{22,i}$ が正の定数でない場合は、スコア統計および残りのカイ 2 乗統計はゼロになるように設定されます。

ワルド統計

ワルド統計は、変数を除外するべきかを判断するために、モデルの中の変数について計算されます。i 番目の変数がカテゴリ変数でない場合は、ワルド統計は次のように定義されます。

$Wald_i = \frac{\hat{\beta}_i^2}{\hat{\sigma}^2_{\hat{\beta}_i}}$

それがカテゴリ変数である場合は、ワルド統計は次のように計算されます。

$\hat{\beta}_i$ を $m-1$ 個のダミー変数に関連付けられた最尤推定量にして、$\mathbf{C}$ を $\hat{\beta}_i$ の漸近分散共分散クロス集計にします。ワルド統計は次のとおりです。

$Wald_i = \hat{\beta}_i' \mathbf{C}^{-1} \hat{\beta}_i$

ワルド統計の漸近分布は、推定されるパラメータの数に等しい自由度のあるカイ 2 乗検定です。

尤度比 (LR) 統計

LR 統計は、MLE で評価された 2 つのモデルの尤度関数の率の対数の 2 倍として定義されます。LR 統計は、変数をモデルから除外するべきかどうかを判断するために使用されます。完全モデルと呼ばれる現在のモデルの中に r_1 個の変数があると仮定します。完全モデルのMLE に基づいて、l(full) を計算します。1 度に 1 つづつ完全モデルから除外する変数のそれぞれについて、MLE を計算し、尤度関数 l(reduced) を計算します。このとき LR 統計は、次のように定義されます。

$LR = -2\ln\left(\frac{l(reduced)}{l(full)}\right) = -2(L(reduced) - L(full))$

LR は、2 つのモデルの中で推定されるパラメータの数の間の差分に等しい自由度を持つ漸近カイ 2 乗分布になります。

条件統計

条件統計も、モデルの中のすべての変数について計算します。条件統計の公式は、LR 統計のものと同じですが、ただし、それぞれの縮小モデルのためのパラメータ推定値が MLE ではなく条件推定になります。条件推定は次のように定義されます。$\hat{\beta} = \left(\hat{\beta}_1, \ldots, \hat{\beta}_{r_1}\right)'$ をモデルの中の r_1 個の変数のための MLE として、C を $\hat{\beta}$ の漸近分散共分散クロス集計とします。変数 x_i がモデルから除外された場合は、与えられたモデル $\hat{\beta}$ の中に残されたパラメータの条件推定は、次のようになります。

$\tilde{\beta}_{(i)} = \hat{\beta}_{(i)} - \mathbf{c}_{12}^{(i)}\left(\mathbf{c}_{22}^{(i)}\right)^{-1}\hat{\beta}_i$

ここで、$\hat{\beta}_i$ は x_i に関連付けられたパラメータの MLE であり、$\hat{\beta_{(i)}}$ は削除された $\hat{\beta}_i$ のある $\hat{\beta}$ であり、$\mathbf{c}_{12}^{(i)}$ は $\hat{\beta_{(i)}}$ と $\hat{\beta}_i$ の間の共変量であり、$\mathbf{c}_{22}^{(i)}$ は $\hat{\beta}_i$ の共変量です。このとき、条件統計は次のように計算されます。

$$-2\left(L\left(\tilde{\beta}_{(i)}\right) - L(full)\right)$$

ここで、 $L\left(\tilde{\beta}_{(i)}\right)$ は、$\hat{\beta_{(i)}}$ で評価される対数尤度関数です。

統計

次の出力統計を利用できます。

初期モデル情報

β_0 がモデルに含まれていない場合は、予測確率は、すべてのケースについて 0.5 であると推定され、対数尤度関数 $L(0)$ は次のようになります。

$$L(0) = W\ln(0.5) = -0.6931472W$$

ここで $W = \sum_{i=1}^{n} w_i$ です。β_0 がモデルに含まれる場合は、予測確率は次のように推定されます。

$$\hat{\pi}_0 = \frac{\sum_{i=1}^{n} w_i y_i}{W}$$

そして、β_0 は次のように推定されます。

$$\hat{\beta}_0 = \ln\left(\frac{\hat{\pi}_0}{1-\hat{\pi}_0}\right)$$

ここで、漸近標準誤差は次のように推定されます。

$$\hat{\sigma}_{\hat{\beta}_0} = \frac{1}{\sqrt{W\hat{\pi}_0(1-\hat{\pi}_0)}}$$

対数尤度関数は次のようになります。

$$L(0) = W\left[\hat{\pi}_0 \ln\left(\frac{\hat{\pi}_0}{1-\hat{\pi}_0}\right) + \ln(1-\hat{\pi}_0).\right]$$

モデル情報

次の統計は、ステップワイズ法が指定された場合に計算します。

-2 対数尤度

$$-2\sum_{i=1}^{n}\left(w_i y_i \ln(\hat{\pi}_i) + w_i(1-y_i)\ln(1-\hat{\pi}_i)\right)$$

カイ 2 乗モデル

2(現在のモデルの対数尤度関数 – 初期モデルの対数尤度関数)

初期モデルには、定数がモデルの中にある場合は、その定数が含まれます。そうでない場合は、モデルには項がありません。カイ 2 乗モデル統計の自由度は、2 つのモデルのそれぞれで推定されるパラメータの数の間の差分に等しくなります。自由度がゼロの場合は、カイ 2 乗モデルは計算されません。

カイ 2 乗ブロック

2(現在のモデルの対数尤度関数 – 以前の方法からの最終モデルの対数尤度関数)

カイ 2 乗ブロック統計の自由度は、2 つのモデルのそれぞれで推定されるパラメータの数の間の差分に等しくなります。

カイ 2 乗改良度

2(現在のモデルの対数尤度関数 – 最後のステップからのモデルの対数尤度関数)

カイ 2 乗改良度統計の自由度は、2 つのモデルのそれぞれで推定されるパラメータの数の間の差分に等しくなります。

適合度

$$\sum_{i=1}^{n} \frac{w_i(y_i - \hat{\pi}_i)^2}{\hat{\pi}_i(1 - \hat{\pi}_i)}$$

Cox と Snell の R2 乗 (Cox and Snell, 1989; Nagelkerke, 1991)

$$R^2_{CS} = 1 - \left(\frac{l(0)}{l(\hat{\beta})}\right)^{\frac{2}{W}}$$

ここで、$l\left(\hat{\beta}\right)$ は現在のモデルの尤度であり、$1(0)$ は初期モデルの尤度です。つまり、定数がモデルに含まれていない場合は、$l(0) = W\log(0.5)$ になります。定数がモデルに含まれている場合は、$l(0) = W[\hat{\pi}_o \log\{\hat{\pi}_o/(1 - \hat{\pi}_o)\} + \log(1 - \hat{\pi}_o)]$ になります。ここでは、$\hat{\pi}_o = \Sigma_i^n w_i y_i / W$ です。

Nagelkerke の R2 乗 (Nagelkerke, 1981)'

$$R^2_N = R^2_{CS} / \max\left(R^2_{CS}\right)$$

ここでは、$\max\left(R^2_{CS}\right) = 1 - \{l(0)\}^{2/W}$

Hosmer-Lemeshow 適合度統計

検定統計量は、$2 \times g$ 分割表にカイ 2 乗検定を適用することで取得します。分割表は、予測事象確率を使用して、予測確率を区分することでグループが形成されるグループ分け変数（g による）を伴う二分従属変数をクロス分類することで、構築します。計算では、約 10 グループを使用します（g=10)。対応するグループは、多くの場合、「リスクの 10 分位」と呼ばれます。“”(Hosmer および Lemeshow, 2000)

観測 i および i’ の独立変数の値が同じである場合は、観測 i および i’ は同じブロックの中にあると言われます。同じ 10 分位の中に 1 つ以上のブロックが発生する場合は、ブロックがこの同じグループ割り当てられています。さらに、同じブロックの中の観測は、グループの中に配置されたときは分割されません。この攻略により 10 未満のグループができ（つまり、$g \leq 10$)、結果として、自由度が少なくなります。

Q 個のブロックが存在し、q 番目のブロックに m_q 数の観測があり、$q = 1, \ldots, Q$ であると仮定します。さらに、k 番目のグループ（$k = 1, \ldots, g$）が q_1 番目、…、q_k 番目のブロックの観測からなると仮定します。このとき、k 番目のグループの中の観測の総数は、$s_k = \Sigma_{q_1}^{q_k} m_j$ です。k 番目のグループ（これを 0_{1k} と呼びます）の中の事象の合計観測度数（つまり、Y=1）は、Y=1 である k 番目のグループの中の観測の合計数です。

$$\xi_k = \Sigma_{q_1}^{q_k} m_j \hat{\pi}_j / s_k$$

Hosmer-Lemeshow 適合度統計は、次のように計算されます。

$$\chi^2_{HL} = \sum_{k=1}^{g} \frac{(O_{1k} - E_{1k})^2}{E_{1k}(1 - \xi_k)}$$

p は、$\Pr(\chi^2 \geq \chi^2_{HL})$ によって与えられます。ここで χ^2 は、自由度（g-2）のカイ 2 乗統計量分布です。

方程式の中にない変数に関する情報

方程式の中にないそれぞれの変数について、関連付けられた自由度、有意確率、および偏相関 R とともに、スコア統計を計算します。X_i を現在モデルの中にない変数として、S_i をスコア統計とします。偏相関 R は次のように定義されます。

$$Partial\_R = \begin{cases} \sqrt{\frac{S_i - 2 \times df}{-2L(initial)}} & S_i > 2 \times df \\ 0 & \text{である場合。そうでなければ} \end{cases}$$

ここで、df は、S_i に関連付けられた自由度であり、$L(initial)$ は、初期モデルの対数尤度関数です。

方程式にない変数のために印刷された残りのカイ 2 乗検定は、次のように定義されます。

$$R_{CS} = \left(L^*_{\mathsf{g}}\right)' B_{22} L^*_{\mathsf{g}}$$

ここで、$L^*_{\mathsf{g}} = \left(L^*_{\gamma_1}, \ldots, L^*_{\gamma_{r_2}}\right)'$

方程式の中にある変数に関する情報

方程式の中にあるそれぞれの変数については、ベータ係数の MLE は、標準誤差、ワルド統計、自由度、有意確率、および偏相関 R とともに計算します。X_i が現在方程式の中にあるカテゴリ変数でない場合は、偏相関 R は次のように計算されます。

$$Partial\_R = \begin{cases} sign\left(\hat{\beta}_i\right)\sqrt{\frac{Wald_i - 2}{-2L(initial)}} & \text{if } Wald_i > 2 \\ 0 & \text{である場合。そうでない場合は} \end{cases}$$

X_i が m 個のカテゴリのあるカテゴリ変数である場合は、偏相関 R は次のようになります。

$$Partial\_R = \begin{cases} \sqrt{\frac{Wald_i - 2(m-1)}{-2L(initial)}} & \text{if } Wald_i > 2(m-1) \\ 0 & \text{である場合。そうでなければ} \end{cases}$$

ケースワイズ統計

次の統計はそれぞれのケースについて計算します。

個別逸脱度

i 番目のケースの逸脱度 G_iは、次のように定義されます。

$$G_i = \begin{cases} \sqrt{2(y_i \ln(\hat{\pi}_i) + (1-y_i)\ln(1-\hat{\pi}_i))} & \text{if } y_i > \hat{\pi}_i \\ -\sqrt{2(y_i \ln(\hat{\pi}_i) + (1-y_i)\ln(1-\hat{\pi}_i))} & \text{otherwise} \end{cases}$$

レバレッジ

i 番目のケースのレバレッジ h_i は、クロス集計の i 番目の対角線要素です。

$$\hat{\mathbf{V}}^{\frac{1}{2}}\mathbf{X}\left(\mathbf{X}'\mathbf{C}\hat{\mathbf{V}}\mathbf{X}\right)^{-1}\mathbf{X}'\hat{\mathbf{V}}^{\frac{1}{2}}$$

この場合、

$$\hat{\mathbf{V}} = Diag\{\hat{\pi}_1(1-\hat{\pi}_1),\ \ldots,\ \hat{\pi}_n(1-\hat{\pi}_n)\}$$

スチューデント化残差

$$\tilde{G}_i^* = \frac{G_i}{\sqrt{1-h_i}}$$

ロジット残差

$$\tilde{e}_i = \frac{e_i}{\hat{\pi}_i(1-\hat{\pi}_i)}$$

ここで、$e_i = y_i - \hat{\pi}_i$

標準化残差

$$z_i = \frac{e_i}{\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}}$$

クックの距離 ’

$$D_i = \frac{z_i^2 h_i}{1-h_i}$$

Df ベータ

$\Delta\beta_i$ を、ケース i を削除したことによる係数推定の変化とします。これは、次のように計算されます。

$$\Delta\beta_i = \frac{\left(\mathbf{x}'\mathbf{C}\hat{\mathbf{v}}\mathbf{x}\right)^{-1}\mathbf{x}'_i e_i}{1-h_i}$$

予測グループ

$\hat{\pi}_i \geq 0.5$ である場合、予測グループは y=1 であるグループです。

次の事項に注意してください。

分析の中の独立変数の非欠損値がある選択しなかったケースについて、レバレッジ $\left(\tilde{h}_i\right)$ は次のように計算されます。

$$\tilde{h}_i = h_i - \frac{\hat{V}_i h_i^2}{1+\hat{V}_i h_i}$$

この場合、

$$h_i = \hat{V}_i \mathbf{X}'_i \left(\mathbf{X}'\mathbf{C}\hat{\mathbf{V}}\mathbf{X}\right)^{-1}\mathbf{X}_i$$

選択しなかったケースでは、クックの距離およびDfベータは、$\tilde{h}_i$ に基づいて計算されます。

生成されたモデル/スコアリング

生成された 2 項ロジスティック回帰モデルを通じて渡されたそれぞれのレコードについて、予測値および確信度スコアは次のように計算します。

予測値

レコード i の値 y = 1 の確率は、次のように計算されます。

$$\hat{\pi}_i = \frac{\exp\left(\hat{\eta}_i\right)}{1+\exp\left(\hat{\eta}_i\right)}$$

この場合、

$$\hat{\eta}_i = \mathbf{X}_i{}' \hat{\beta}$$

$\hat{\pi} > 0.5$ である場合は、予測値は 1 になります。そうでない場合は、予測値は 0 になります。

遖コ董。蟒ヲ

y = 1 の予測値があるレコードについて、確信度の値は $\hat{\pi}$ になります。y = 0 の予測値があるレコードについて、確信度の値は $(1 - \hat{\pi})$ になります。

空白の処理（生成されたモデル）

最終モデルにおいて、入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールド $null$ が割り当てられます。

章

23

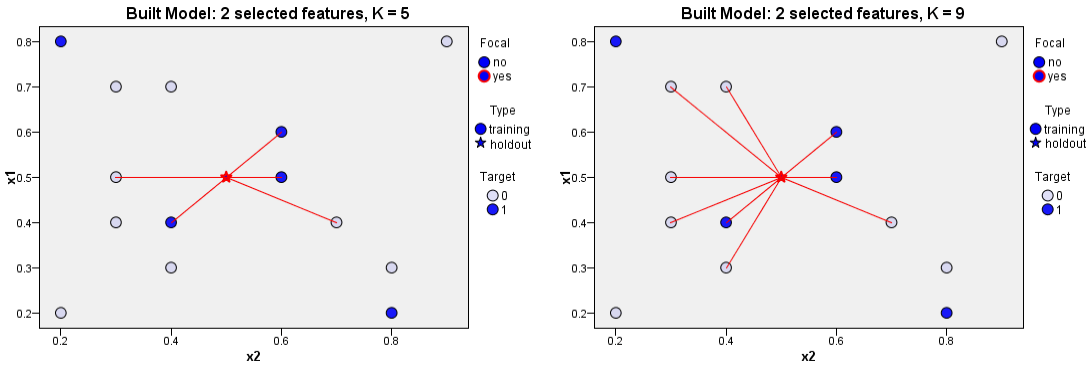
KNN アルゴリズム

最近隣分析は、そのほかのケースに対する類似性に基づいてケースを分類する方法です。マシン学習で、保存されたパターン、またはケースへに完全に一致する必要なくデータのパターンを認識する方法として開発されました。同様のケースはお互いに近く、異なるケースはお互いに離れています。つまり、2 つのケース間の距離は、それらの非類似度の尺度です。

お互いに近いケースは「近隣」と呼ばれます。新しいケース（ホールドアウト）が表示されている場合、モデルのケースからの距離が計算されます。最も類似した分類「最近隣」が集計され、新しいケースが、最大数の最近隣を含むカテゴリに投入されます。

検証する最近隣の数を指定できます。この値は k となります。図は、新しいケースが 2 つの異なる値の k を使用してどのように分類されるかを示します。k = 5 の場合、最近隣の大部分はカテゴリ 1 に属するため、新しいケースはカテゴリ 1 にあります。ただし k = 9 の場合、最近隣の大部分はカテゴリ 0 に属するため、新しいケースはカテゴリ 0 にあります。

図 23-1
分類時に k を変更した場合の効果



また、最近隣分析を使用して、連続型対象の値を計算することもできます。この場合、最近隣の平均または中央の対象値を使用して、新しいケースの予測値を取得します。

表記

この章では特に明記しない限り、次の表記を使用します。

Y	要素 y_n を持つレスポンスのオプションの 1×N ベクトルです。n=1,...,N は、ケースを指数化します。
X^0	要素 x^0_{pn} を持つフィールドの $P^0 \times N$ 行列で、p=1,...,P^0 はフィールドを、n=1,...,N はケースを指数化します。

X	要素 x_{pn} を持つエンコードされたフィールドの P×N 行列で、p=1,...,P はフィールドを、n=1,...,N はケースを指数化します。
P	特徴空間の次元数。連続型フィールドの数と、カテゴリ型フィールド全体のカテゴリ数。
N	ケース総数。
$N_j, j = 1, 2, \cdots, J$	Y = j のケース数。Y は、J 件のカテゴリを持つレスポンス変数です。
$\hat{N}_j$	クラス j に所属し、j として分類されるケース数。
$\hat{N}_j^*$	j として分類されるのケース数の合計。

事前処理

フィールドをコード化して、測定値の尺度の差異を説明します。

連続

連続型フィールドは、オプションで、調整済み正規化を使用してコード化されます。

$$x_{pn} = \frac{2\left(x_{pn}^0 - \min\left(x_p^0\right)\right)}{\max\left(x_p^0\right) - \min\left(x_p^0\right)} - 1$$

x_{pn} は、ケース n の 入力フィールド p の正規化された値で、x_p^0 は、ケース n のフィールドの元の値、$\min\left(x_p^0\right)$ は、すべての学習ケースのフィールドの最小値、そして、$\max\left(x_p^0\right)$ はすべての学習ケースの最大値です。

カテゴリ

カテゴリ フィールドは、必ず one-of-c コーディングを使用して一時的に再コード化されます。フィールドに c 個のカテゴリがある場合、c 個のベクトルとして保存し、最初のカテゴリは（$1, 0, \ldots, 0$）、次のカテゴリは（$0, 1, 0, \ldots, 0$）、最後のカテゴリは（$0, 0, \ldots, 0, 1$）と表記します。

学習

最近隣モデルの学習では、フィールド セットの値に基づいたケース間の距離の計算が行われます。指定されたケースと最近隣との距離は、最短距離です。距離メトリック、最近隣数の選択、フィールド セットの選択には、次のオプションがあります。

距離メトリック

次のいずれかのメトリックを使用して、クエリー ケースと最近隣の類似性を測定します。

ユークリッド距離： 2 つのケース間の距離は、すべての次元の、ケースの値の間の差の平方和の平方根です。

$$Euclidean_{ih} = \sqrt{\sum_{p=1}^{P} w_{(p)} \left(x_{(p)i} - x_{(p)h}\right)^2}$$

都市ブロック距離： 2 つのケースの間の距離は、すべての次元の、それらのケースの値の間の重み付けされた絶対差の合計です。

$$CityBlock_{ih} = \sum_{p=1}^{P} w_{(p)} \left|x_{(p)i} - x_{(p)h}\right|$$

フィールドの重要度を距離の重み付けに使用しない場合、フィールドの重み$w(p)$ は 1、距離の重み付けに使用する場合、フィールドの重みは正規化されたフィールドの重要度となります。

$$w_{(p)} = FI_{(p)} / \sum_{p=1}^{P} FI_{(p)}$$

フィールドの重要度の計算 $FI_{(p)}$ ついては、「出力統計 」を参照してください。

k の選択の交差検証

交差検証を使用して、最小値 $k_{\min}$ および最大値 $k_{\max}$ の間の最近隣数を自動的に選択します。学習セットに、整数値 $1, 2, \ldots,$ V の交差検証変数があるとします。交差検証アルゴリズムは、次のようになります。

- 各 $k \in [k_{\min}, k_{\max}]$ について、k の平均誤差率または誤差平方和を計算します。$CV_k = \sum_{v=1}^{V} e_v / V$ となり、最近隣モデルを適用して、$X = v$ でケースの予測を行う場合、つまり、他のケースを学習データセットとして使用する場合、e_v は誤差率または誤差平方和となります。
- 最適な k を選択すると、$\hat{k} = arg\{\min CV_k : k_{\min} \le k \le k_{\max}\}$ のようになります。

  注 ： 複数の値の k が最も低い平均誤差と同等である場合、同等である中から最も小さい k を選択します。

フィールド選択

フィールド選択は、Cunningham および Delany (2007) のラッパー アプローチに基づいており、モデルに投入される J_{Forced} フィールドから始まる変数増加法を使用します。それ以降のフィールドは順番に選択されます。各ステップで選択されるフィールドは、誤差率または誤差平方和で最も大きく減少するフィールドです。

S_J が、現在選択されている J フィールドのセットを示すとすると、 S_J^c は残りのフィールドのセットを示し、e_J は、S_J に基づくモデルと関連する誤差率または誤差平方和を示します。

アルゴリズムは次のようになります。

- $J = J_{Forced}$ フィールドから始まります。
- S_J^c の各フィールドについて、k の最近隣モデルとこのフィールド、そして S_J の既存フィールドを適合し、各モデルの誤差率および誤差平方和を算出します。モデルの誤差率または誤差平方和が最も小さい S_J^c のフィールドを追加して、S_{J+1} を作成します。
- 選択した停止基準をチェックします。基準を満たす場合、選択したフィールド サブセットを停止して報告します。そうでない場合は、J=J+1 となり、前のステップに戻ります。

注 : カテゴリ型予測の関連するエンコードされたフィールドのセットは、セットとしてみなされ、フィールド選択の目的で追加されます。

停止基準

2 つの停止基準の一方を、フィールド選択アルゴリズムに適用することができます。

固定された数のフィールド : アルゴリズムでは、モデルに投入されるフィールドに加え、固定された数のフィールド J_{add} を追加します。最後のフィールド サブセットには、$J_{add} + J_{Forced}$ フィールドがあります。J_{add} は、ユーザー指定または自動的に計算されます。自動的に計算される場合、値は次のようになります。

$$J_{add} = \max\left\{\min\left(20, P^0\right) - J_{Forced},\ \ 0\right\}$$

これが停止基準である場合、J_{add} フィールドがモデルに追加されると、フィールド選択アルゴリズムは停止します。つまり、$J_{add} = J + 1$ となった場合、S_{J+1} を選択したフィールドサブセットとして停止および報告します。

注 : $J_{add} = 0$ である場合、フィールドは追加されず、$J = J_{Forced}$ である S_J が選択されたフィールド サブセットとして報告されます。

誤差率または誤差平方和の変化 : アルゴリズムは、絶対誤差率の変化が、これ以上特徴を追加してもモデルが改善されないことを示す場合に停止します。具体的には、$e_{J+1} = 0$ または $e_J \geq e_{J+1}$ および

$$\frac{|e_J - e_{J+1}|}{e_J} \leq \Delta_{\min}$$

である場合、$\Delta_{\min}$ は、指定された変化の最小値であり、S_{J+1} を選択されたフィールド サブセットとして停止および報告します。

$e_J < e_{J+1}$ および

$$\frac{|e_J - e_{J+1}|}{e_J} > 2\Delta_{\min}$$

S_J を選択されたフィールド サブセットとして停止および報告します。

注 : $J = J_{Forced}$ で $e_J = 0$ である場合、フィールドは追加されず、$J = J_{Forced}$ である S_J が、選択されたフィールド サブセットとして報告されます。

k の結合およびフィールド選択

次の方法は、結合された近隣とフィールド選択に使用されます。

1. 各 k について、フィールド選択に変数増加法を使用します。
2. k と、誤差率が最も低い、または誤差平方和が最も低い、付随するフィールド セットを選択します。

空白の処理

入力フィールドまたは出力フィールドに欠損値があるすべてのレコードは、モデルの推定から除外されます。

出力統計

次の統計を利用できます。

クラス j の正確パーセント

$$\frac{\hat{N}_j}{N_j} \times 100\%$$

クラス j のすべてのパーセント

$$\frac{\hat{N}_j^*}{N} \times 100\%$$

すべてのパーセントと正確パーセントの交差

$$\left(\sum_{j=1}^{J} \hat{N}_j / N \right) \times 100\%$$

分類の誤差率

$$\left(1-\sum_{j=1}^{J}\hat{N}_j/N\right)\times 100\%$$

連続型レスポンスの誤差平方和

$$\sum_{n=1}^{N}(y_n-\hat{y}_n)^2$$

この場合、$\hat{y}_n$ は、y_n の推定値です。

フィールドの重要度

誤差率または誤差平方和が e である、変数増加法プロセスのモデルに $X_{(1)},X_{(2)}\cdots X_{(m)}\left(1\le m\le P^0\right)$ があるとします。そのモデルのフィールド $X_{(p)}$ の重要度は、次の方法で算出されます。

- モデルからフィールド $X_{(p)}$ を削除し、フィールド $X_{(1)},X_{(2)}\cdots X_{(p-1)},X_{(p+1)},\cdots,X_{(m)}$ に基づいて、誤差率または誤差平方和 $e_{(p)}$ を予測し、評価します。
- 誤差率 $e_{(p)}+\frac{1}{m}$ を算出します。

$X_{(p)}$ is $FI_{(p)}=e_{(p)}+\frac{1}{m}$ のフィールドの重要度。

スコアリング

ケースの k の最近隣を検出した後、最近隣を分類したり、その回答値を予測することができます。

カテゴリ型レスポンス

学習ケースにおいて k の最近隣の多数決を行って、各ケースを分類します。

- 複数のカテゴリが最大予測確率で同等である場合、学習セットのケース数が最も多いカテゴリを選択してタイ値を解決する必要があります。
- 複数のカテゴリが学習セットの最大ケース数で同等である場合、同等のカテゴリ間で最もデータ値が小さいカテゴリを選択します。この場合、カテゴリはデータ値の昇順または文字順であると想定されています。

また、各カテゴリの予測カテゴリを算出することもできます。k_j は、k の最近隣で j 番目のカテゴリのケース数であるとします。$\frac{k_j}{k}$ によって j 番目のカテゴリの予測確率を推定するのではなく、次のようにラプラス補正を適用します。

$$\frac{k_j + 1}{k + J}$$

ここで J は、学習セット中のカテゴリ数です。

ラプラス補正によって、最近隣数が小さい場合、確率推定が 1/J まで減少します。また、クエリー ケースに、回答値が同じ最近隣が k けんある場合、確率推定は、1 または 0 ではなく、0 より大きく 1 より小さくなります。

連続型レスポンス

平均関数または中央値関数を使用して各ケースを予測します。

平均関数 :$\hat{y}_n = \sum_{m \in Nearest(n)} y_m / k$。$Nearest(n)$ はケース n の最近隣であるこれらのケースのインデックス セットであり、y_m は、ケース m の連続型レスポンス変数の値です。

中央関数 :$y_m, m \in Nearest(n)$ が連続型レスポンス変数の値であるとし、$y_m, m \in Nearest(n)$ を最小値から最大値まで調整し、それらを $y_{(j_1)} \leq y_{(m_2)} \leq \cdots \leq y_{(m_k)}$ と表すと、中央値は次のようになります。

$$\hat{y}_n = \begin{cases} y_{\left(\frac{k+1}{2}\right)} & k \text{ is odd} \\ \frac{y_{\left(\frac{k}{2}\right)} + y_{\left(\frac{k}{2}\right)+1}}{2} & k \text{ is even} \end{cases}$$

空白の処理

入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールドおよび確率値に $null$ が割り当てられます。

参照

Arya, S., および D. M. Mount. 1993. Algorithms for fast vector quantization. In: Proceedings of the Data Compression Conference 1993, , 381-390.

Cunningham, P., および S. J. Delaney. 2007. k-Nearest Neighbor Classifiers. Technical Report UCD-CSI-2007-4, School of Computer Science and Informatics, University College Dublin, Ireland, , - .

Friedman(F), J. H., J. L. Bentley, および R. A. Finkel. 1977. An algorithm for finding best matches in logarithm expected time. ACM Transactions on Mathematical Software, 3, 209-226.

章
24

線型モデル作成アルゴリズム

線型モデルは、対象と 1 つまたは複数の予測値との線型の関係に基づいて連続型対象を予測します。

モデルの精度の拡張、モデルの安定性の拡張、非常に大きいデータセットを扱うアルゴリズムの詳細は、「アンサンブル アルゴリズム」（ p. 124 ）を参照してください。

表記

この章では特に明記しない限り、次の表記を使用します。

n	データセット内のレコード数。これは整数であり、$n \geq 1$ 1 です。
p	モデル内のパラメータの数（ダミー変数のパラメータを含むが定数項は除く）。これは整数であり、$p \geq 0$ 0 です。
p^*	モデル内に現在ある非冗長パラメータの数（定数項は除く）。これは整数であり、$0 \leq p^* \leq p$ です。
p^c	モデル内に現在ある非冗長パラメータの数。$p^c = p^* + 1$
p^e	定数項を除く効果の数。これは整数であり、$0 \leq p^e \leq p$ です。
y	要素 y_i を持つ $n \times 1$ 対象ベクトル。
f	$n \times 1$ の度数の重みベクトル。
g	$n \times 1$ の回帰の重みベクトル。
N	効果的なサンプル サイズ。これは整数であり、$N = \sum_{i=1}^{n} f_i$ です。 度数の重みベクトルがない場合、N=n となります。
X	要素 x_{ij} を持つ $n \times (p+1)$ の計画行列。行はレコード、列はパラメータを表します。
ϵ	潜在誤差の $n \times 1$ のベクトル。
β	不明なパラメータの $(p+1) \times 1$ のベクトル、$\beta = (\beta_0, \beta_1, \cdots \beta_p)$。$\beta_0$ は定数項です。
$\hat{\beta}$	パラメータ推定の $(p+1) \times 1$ のベクトルです。
b	標準化パラメータ推定の $(p+1) \times 1$ のベクトルです。行列 **R** のスイープ操作の結果です。b_0 は定数項の標準化された推定で、0 となります。
$\hat{\mathbf{y}}$	予測対象値の $n \times 1$ のベクトル。
$\overline{X}_j$	X_j, $j = 1, 2, \cdots p$ の重みつきサンプル平均。
$\overline{y}$	**y** の重み付きサンプル平均。
S_{ij}	X_i と X_j, $i, j = 1, 2, \cdots p$ の標本サンプル共分散。
S_{iy}	X_i および **y** の間の重み付き標本共分散。

S_{yy}	y の重み付き標本分散。
R	X（定数項がある場合は除く）と y の$(p+1)\times(p+1)$ の重み付き標本相関行列。
$\tilde{\mathbf{R}}$	要素が $\tilde{r}_{ij}$ であるスイープ操作後に作成される行列。

モデル

線型回帰の形式は次のとおりです。

$\mathsf{y} = \mathsf{X}\beta \ + \ \varepsilon$

この場合、ε は、平均が 0 で分散が $\sigma^2\mathsf{D}^{-1}$ の正規分布となり、$\mathsf{D}^{-1} = diag(1/g_1, \ldots, 1/g_n)$ となります。ε の要素はお互い独立しています。

注:

- X は、連続型効果およびカテゴリ型効果を組み合わせることができます。
- 計画行列の定数列は、モデル作成では使用されません。
- n=1 の場合、または対象が定数の場合、モデルは構築されません。

欠損値

欠損値のあるレコードがリストごとに削除されます。

最小 2 乗推定

係数は、最小 2 乗（LS）を使用して推定されます。まず、次のように $\mathsf{D}^{1/2}$ を事前に乗算してモデルを変換します。

$\mathsf{D}^{1/2}\mathsf{y} = \mathsf{D}^{1/2}\mathsf{X}\beta \ + \mathsf{D}^{1/2}\ \varepsilon$

すると、新しい潜在誤差 $\mathsf{D}^{1/2}\ \varepsilon$ は正規分布 $N_n\left(0, \sigma^2\mathbf{I}\right)$ となり、I は恒等行列で $\mathsf{D}^{1/2} = diag\left(\sqrt{g_1}, \ldots, \sqrt{g_n}\right)$ となります。β の最小 2 乗推定は、次の式から取得できます。

$$\hat{\beta} = arg\min_{\beta}\left(\mathsf{D}^{1/2}\mathsf{y} - \mathsf{D}^{1/2}\mathbf{X}\beta\right)^{\mathsf{T}} F\left(\mathsf{D}^{1/2}\mathbf{y} - \mathsf{D}^{1/2}\mathbf{X}\beta\right)$$

ここで、$\mathsf{F} = \mathsf{diag}(f_1, \ldots, f_n)$ となります。次の点に注意してください。

$$\begin{aligned}&\left(\mathsf{D}^{1/2}\mathbf{y} - \mathsf{D}^{1/2}\mathbf{X}\beta\right)^{\mathsf{T}} F\left(\mathsf{D}^{1/2}\mathbf{y} - \mathsf{D}^{1/2}\mathbf{X}\beta\right)\\ &\quad = (\mathbf{y} - \mathbf{X}\beta)^{\mathsf{T}}\mathsf{D}^{1/2}F\mathsf{D}^{1/2}(\mathbf{y} - \mathbf{X}\beta)\\ &\quad\quad = (\mathbf{y} - \mathbf{X}\beta)^{\mathsf{T}}W(\mathbf{y} - \mathbf{X}\beta)\end{aligned}$$

ここで、$W = \mathsf{diag}(w_1, \ldots, w_n) = \mathsf{diag}(g_1 f_1, \ldots, g_n f_n)$ となるため、$\hat{\beta}$ の閉形式解は、次のようになります。

$$\hat{\beta} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}\right)^{-}\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{y}$$

$\hat{\beta}$ は、上記の等式の代わりにスイープ操作を適用して計算されます。また、スイープ操作を X と y の変換スケールに適用して、数値の安定性を実現します。具体的には、重み付き標本相関行列 R を構築し、スイープ操作を適用します。R 行列は次のように構築されます。

まず、重み付き標本平均、分散、共分散を $\mathtt{X}_{\mathrm{i}}$、$\mathtt{X}_{\mathrm{j}}$、$i, j = 1, \ldots, p,$ および y で計算します。

$\mathtt{X}_{\mathrm{i}}$ および y の重み付き標本平均は、$\overline{X}_i = \frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k x_{ki}$ および $\overline{y} = \frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k y_k$ です。

$\mathtt{X}_{\mathrm{i}}$ および $\mathtt{X}_{\mathrm{j}}$ の重み付き標本共分散は $S_{ij} = \frac{1}{N-1} \sum_{k=1}^{n} w_k \left(x_{ki} - \overline{X}_i\right)\left(x_{kj} - \overline{X}_j\right)$ となります。

$\mathtt{X}_{\mathrm{i}}$ および y の重み付き標本共分散は、$S_{iy} = \frac{1}{N-1} \sum_{k=1}^{n} w_k \left(x_{ki} - \overline{X}_i\right)\left(y_k - \overline{y}\right)$ です。

y の重み付き標本分散は $S_{yy} = \frac{1}{N-1} \sum_{k=1}^{n} w_k (y_k - \overline{y})^2$ となります。

次に、重み付き標本相関 $r_{ij} = \frac{S_{ij}}{\sqrt{S_{ii} S_{jj}}}$, $i, j = 1, ..., p$ および y を計算します。

行列 R は次のようになります。

$$\mathsf{R} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} & r_{1y} \\ r_{21} & r_{22} & \cdots & r_{2p} & r_{2y} \\ \vdots & \vdots & \ddots & \vdots & \\ r_{p1} & r_{2p} & \cdots & r_{pp} & r_{py} \\ r_{y1} & r_{y2} & \cdots & r_{yp} & r_{yy} \end{bmatrix} = \begin{bmatrix} \mathsf{R}_{11} & \mathsf{R}_{12} \\ \mathsf{R}_{12}^{\mathsf{T}} & R_{22} \end{bmatrix}$$

スイープ操作が $\mathbf{R}_{11}$ の各行に繰り返し適用され、$\mathbf{R}_{11}$ が現在のステップでモデルの予測フィールドを含む場合、結果は次のようになります。

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R}_{11}^{-1} & \mathbf{R}_{11}^{-1}\mathbf{R}_{12} \\ -\mathbf{R}_{12}^{\mathsf{T}}\mathbf{R}_{11}^{-1} & R_{22} - \mathbf{R}_{12}^{\mathsf{T}}\mathbf{R}_{11}^{-1}\mathbf{R}_{12} \end{bmatrix}$$

最後の列 $\mathsf{R}_{11}^{-1}\mathsf{R}_{12}$ には標準化係数推定が含まれます。つまり、$\mathbf{b} = \mathbf{R}_{11}^{-1}\mathbf{R}_{12}$ となります。モデル内に定数項がある場合、定数項の推定を除き、係数推定は次のようになります。

$$\hat{\beta}_j = b_j \sqrt{\frac{S_{yy}}{S_{jj}}}$$

モデル選択

次のモデル選択方法がサポートされています。

- なし：選択方法は使用されず、効果がモデルに強制投入されます。この方法では、スイープ操作時に特異性許容度が 1e-12 に設定されます。
- 変数増加ステップワイズ法：モデルの効果がない状態から始まり、ステップワイズ法の基準に従ってこれ以上追加または削除できなくなるまで一度に 1 ステップずつ効果を追加および削除します。
- 最適サブセット：「使用できるすべての」モデル、または少なくとも変数増加ステップワイズ法より大きいサブセットの使用できるモデルをチェックし、最適サブセットの基準に従って最適サブセットを選択します。

変数増加ステップワイズ法

変数増加ステップワイズ法の基本的理念は、追加できる限り、一度に効果を 1 つずつ追加されます。効果が追加されると、現在のモデルのすべての効果がチェックされ、いずれかを削除する必要があるかどうかを確認します。停止基準を満たすまで、処理は続行します。従来の効果の投入および削除の基準は、指定された投入および削除の有意度と比較されます F 統計量および対応する p に基づいています。ただし、これらの統計量は実際 F 分布にはならないため、結果が疑わしくなります。そのため、効果投入および削除に関して次の追加基準があります。

- 最大調整済み R^2 乗
- 最小補正済み赤池情報基準（AICC）
- オーバーフィット防止データの最小平均平方誤差（ASE）

候補の統計量

連続型効果 X_j またはカテゴリ型効果 $\{X_{j_s}\}_{s=1}^{\ell}$ の追加または削除について説明するには、いくつかの追加表記が必要です。 ℓ はカテゴリ数を示します。

ℓ^*	的確な効果の非冗長パラメータ数 $\mathbf{X}_j$ or $\{X_{j_s}\}_{s=1}^{\ell}$。
p^c	モデル内に現在ある非冗長パラメータの数（定数項を含む）。
p^r	作成されたモデル内の非冗長パラメータの数（定数項を含む）。$p^r = \begin{cases} p^c + \ell^* & \text{for entering an effect} \\ p^c - \ell^* & \text{for removing an effect} \end{cases}$ となります。
SSe_p	現在のモデルの重み付き残差平方和。
$SSe_{p+\ell}$	効果投入後に作成されたモデルの重み付き残差平方和。
$SSe_{p-\ell}$	効果削除後に作成されたモデルの重み付き残差平方和。
r_{yy}	現在の $\mathbf{R}$ 行列の最後の対角線要素。
$\tilde{r}_{yy}$	作成された $\tilde{\mathbf{R}}$ 行列の最後の対角線要素。

F 統計量： 現在のモデルから効果を投入または削除する F 統計量は次のようになります。

$$F_{enter_j} = \frac{\left(SSe_p - SSe_{p+\ell}\right)/\ell^*}{SSe_{p+\ell}/\left(N-p^r\right)} = \frac{\left(r_{yy} - \tilde{r}_{yy}\right)\left(N-p^r\right)}{\tilde{r}_{yy} \times \ell^*}$$

$$F_{remove_j} = \frac{\left(SSe_{p-\ell} - SSe_p\right)/\ell^*}{SSe_p/(N-p^c)} = \frac{(\tilde{r}_{yy} - r_{yy})(N-p^c)}{r_{yy} \times \ell^*}$$

そして、対応する p 値は次のようになります。

$$p_{enter_j} = P\left(F_{\ell^*,N-p^r} \geq F_{enter_j}\right) = 1 - P\left(F_{\ell^*,N-p^r} \leq F_{enter_j}\right)$$

$$p_{remove_j} = P\left(F_{\ell^*,N-p^c} \geq F_{remove_j}\right) = 1 - P\left(F_{\ell^*,N-p^c} \leq F_{remove_j}\right)$$

調整済み R2 乗： 現在のモデルから効果を投入または削除する調整済み R^2 乗の値は次のようになります。

$$\text{adj.}R^2 = 1 - \frac{(N-1)\tilde{r}_{yy}}{N-p^r}$$

補正赤池情報量基準 (AICC) 現在のモデルから効果を投入または削除する AICC の値は次のようになります。

$$AICC = N \ln\left(\frac{(N-1)\,S_{yy} \times \tilde{r}_{yy}}{N}\right) + \frac{2p^r N}{N-p^r-1}$$

平均平方誤差 (ASE)： 現在のモデルから効果を投入または削除する ASE の値は次のようになります。

$$ASE = \frac{1}{\sum_{t=1}^{T} f_t} \sum_{t=1}^{T} w_t (y_t - \hat{y}_t)^2$$

ここで、$\hat{y}_t = \mathbf{x}_t\hat{\beta}$ は y_t の予測値となり、T は、オーバーフィット防止セットの検定ケースの数です。

選択プロセス

モデル選択基準に応じて、選択プロセスに若干の違いがあります。

- F 統計量基準は、最小（最大）p 値の投入（削除）の効果を選択し、投入（削除）するすべての p 他指定された有意レベル以上 （以下）になるまでこの作業を続行します。
- その他 3 つの基準は、現在のモデルの統計量と、効果を投入（削除）した後のモデルの統計量（調整済み R^2 乗、AICC または ASE）を比較することです。選択は、ローカル最適値（調整済み R^2 乗基準の最大値と AICC および ASE の最小値）で停止します。

選択プロセスに必要ないくつかの定義は、次のとおりです。

FLAG	各効果の状態を記録する $p^e \times 1$ のインデックスのベクトル。$FLAG_i = 1$ は、効果 i が現在のモデルにあることを示し、$FLAG_i = 0$ が現在のモデルにないことを示します。$\lvert\{i \lvert FLAG_i = 1\}\rvert$ は、$FLAG_i = 1$ の効果の数を示します。
MAXSTEP	反復ステップの最大回数。デフォルト値は $3 \times p^e$ です。
MAXEFFECT	効果の最大数（定数項がある場合は除外）。デフォルト値は p^e です。
P_{in}	F 統計量の基準が使用される場合、効果投入の有意水準。デフォルトは 0.05 です。
P_{out}	F 統計量の基準が使用される場合、効果削除の有意水準。デフォルトは 0.1 です。
ΔF	F 統計量の変化。効果 X_j を投入または削除する場合、F_{enter_j} または F_{remove_j} となります（この場合、単純な表記であるため、X_j は連続型かカテゴリ型かは示しません）。
$p_{\Delta F}$	対応する ΔF の p 値。
$MSC_{current}$	現在のモデルの調整済み R^2 乗、AICC、または ASE の値。

1. $\{FLAG_i\}_{i=1}^{p^e} = 0$ および iter = 0 を設定します。最初のモデルは $\hat{y} = \overline{y}$ となります。調整済み R^2 乗、 AICC、または ASE の基準が使用されている場合、初期モデルの統計量を計算し、$MSC_{current}$ として示します。

2. $\{i \lvert FLAG_i = 0\} \neq 0$、iter ≤ MAXSTEP および $\lvert\{i \lvert FLAG_i = 1\}\rvert < MAXEFFECT$ の場合、次のステップに進みます。そうでない場合は停止し、現在のモデルを出力します。

3. 現在のモデルに基づいて、投入に適したすべての効果 j の場合、次のようになります（下記の「条件」を参照してください）。

   FC（F 統計量基準）が使用される場合、F_{enter_j} および p_{enter_j} を計算します。

   MSC（調整済み R^2、AICC、または ASE 基準）が使用される場合、MSC_j 計算します。

4. FC を使用する場合、効果 $X_{j^*}, j^* = arg\min_j \{p_{enter_j}\}$ を選択し、 $p_{enter_{j^*}} < P_{in}$ の場合、X_{j^*} を現在のモデルに投入します。

   MSC を使用する場合、効果 $X_{j^*}, j^* = arg\min_j \{MSC_j\}$ を選択し、$MSC_{j^*} < MSC_{current}$ の場合、X_{j^*} を現在のモデルに投入します（調整済み R^2 基準の場合、最小値を最大値に置き換えて、不等式を逆にします）。

   不等式が満たされない場合、現在のモデルで停止して出力します。

5. 以前取得したモデルと新しい効果のモデルが同じ場合、停止して現在のモデルを出力します。そうでない場合、現在の **R** 行列の X_{j^*} と関連する対応行および列にスイープ操作を実行して現在のモデルを更新します。$FLAG_{j^*} = 1$ および iter = iter + 1 を設定します。

   FC を使用する場合、$\Delta F = F_{enter_{j^*}}$ および $p_{\Delta F} = p_{enter_{j^*}}$ となります。

   MSC を使用する場合、$MSC_{current} = MSC_{j^*}$ となります。

6. 現在のモデルのすべての効果 k、 つまり $FLAG_k = 1, \forall k$ の場合、次のようになります。

   FC を使用する場合、F_{remove_k} および p_{remove_k} を計算します。

   MSC を使用する場合、MSC_k を計算します。

7. FC を使用する場合、効果 $X_{k^*}, k^* = arg\max_k \{p_{remove_k}\}$ を選択し、$p_{remove_{k^*}} > \text{P}_\text{out}$ の場合、現在のモデルから X_{k^*} を削除します。

   MSC を使用する場合、効果 $X_{k^*}, k^* = arg\min_k \{MSC_k\}$ を選択し、$MSC_{j^*} < MSC_{current}$ である場合、現在のモデルから X_{k^*} を削除します(調整済み R^2 基準の場合、最小値を最大値に置き換えて、不等式を逆にします)。

   不等式が満たされない場合、次のステップに進みます。そうでない場合は、ステップ 2 に戻ります。

8. 以前取得したモデルと効果を削除されたモデルが同じ場合、停止して現在のモデルを出力します。そうでない場合、現在の R 行列の X_{j^*} と関連する対応行および列にスイープ操作を実行して現在のモデルを更新します。$FLAG_{j^*} = 0$ および iter = iter + 1 を設定します。

   FC を使用する場合、$\Delta F = F_{remove_{k^*}}$ および $p_{\Delta F} = p_{remove_{k^*}}$ となります。

   AC を使用する場合、$AICC_{current} = AICC_{k^*}$ となります。そしてステップ 6 に戻ります。

   条件： 効果 j がモデルへの投入に適切なものとなるには、次の条件を満たす必要があります。

   連続型効果 X_j の場合、$r_{jj} \geq t$ となります（t は、値が 1e−4 の特異性許容度です)。

   カテゴリ型の場合、効果は $\{X_{j_s}\}_{s=1}^{\ell}$, $max\{r_{j_1 j_1}, r_{j_2 j_2}, \ldots, r_{j_\ell j_\ell}\} \geq t$となります。

   この場合、t は特異性許容度となり、r_{jj} および $r_{j_s j_s}, s = 1, \ldots, \ell,$ は現在の R 行列（投入前）の対角線要素となります。

   現在のモデルにある連続型効果 X_k の場合、$\tilde{r}_{kk} t \leq 1$ となります。

   現在のモデルにあるカテゴリ型効果 $\{X_{k_s}\}_{s=1}^{\ell'}$ でレベルが ℓ' である場合、$max\left\{\tilde{r}_{k_1 k_1}, \tilde{r}_{k_2 k_2}, \ldots, \tilde{r}_{k_{\ell'} k_{\ell'}}\right\} t \leq 1$ となります。

   この場合、$\tilde{r}_{kk}$ と $\tilde{r}_{k_s k_s}, s = 1, \ldots, \ell',$ は R 行列の対角線要素となります。つまり、現在の R 行列の X_k または $\{X_{k_s}\}_{s=1}^{\ell'}$ に関連する該当行および列にスイープ操作を実行した後の結果となります。効果の投入によりモデル中の効果の許容度が許容できないレベルまで減少することがないようにするために、上記の条件が適用されます。

最適サブセット

ステップワイズ法では、サブモデルの組み合わせの検索数が少なく、最適なサブモデルを選択できることが少なくなります。そのため、もう 1 つのオプションによって、すべての可能なモデルをチェックし、いくつかの基準に基づいて「最適な」ものを選択します。使用できる基準は、最大調整済み R^2 乗、最小 AICC、およびオーバーフィット防止セットの最小 ASE です。

p^e のない効果があるため、2^{p^e} モデルに対し、定数項のみのモデル（$\hat{y} = \bar{y}$）を含む包括的な検索を実行します。計算数が p^e で指数的に増加するため、必要な計算を実行する効果的なアルゴリズムを使用することが重要です。ただし、p^e が大きすぎる場合、すべてのモデルをチェックすることは実践的ではない場合があります。

効果の数を単位として、問題を 2 段階に分割します。

- $p^e \leq 20$ の場合、すべてのサブセットを検索します。
- $p^e > 20$ の場合、変数増加ステップワイズ法とすべてのサブセット法を結合する混合型の方法を適用します。

すべてのサブセットの検索

R 行列のスイープ操作数を最小化する効果的な方法（Schatzoff 1968）を適用し、すべてのモデルを検証します。概要は次のとおりです。

効果に対するスイープのステップごとにモデルが作成されます。そのため、2^{p^e} のモデルが、効果に対する 2^{p^e} のスイープのシーケンスで取得できます。$p^e - 1$ の効果のすべてのモデルが、最初の 2^{p^e-1} の重要な効果の 2^{p^e-1} のスイープのシーケンス S_{p^e-1} で取得できます。そして最後のスイープによってシーケンス S_{p^e-1} で作成されたモデルに最後の効果を追加する新しいモデルを作成し、シーケンス S_{p^e-1} を繰り返すと、別の 2^{p^e-1} のモデルが作成されます（最後の効果を含む）。これはシーケンスを構築す再帰的アルゴリズムとなります。つまり、$S_{p^e} = \left(\underline{S_{p^e-1}}, k, \underline{S_{p^e-1}}\right) = \left(\underline{S_{p^e-2}, k-1, S_{p^e-2}}, k, \underline{S_{p^e-2}, k-1, S_{p^e-2}}\right) = \ldots,$ などです。

次のように、作成されるモデルのシーケンスを示します。

k	S_k	作成されるモデルのシーケンス
0	0	定数項のみ
1	1	(1)
2	121	(1), (12), (2)
3	1213121	(1), (12), (2), (23), (123), (13), (3)
4	121312141213121	(1), (12), (2), (23), (123), (13), (3), (34), (134), (1234), (234), (24), (124),
...	...	...
p^e	S_{p^e-1}, p^e, S_{p^e-1}	定数項モデルを含むすべての 2^{p^e} もでる。

2 番目の列は、重要な中枢となる効果のインデックスを示します。3 番目の列のカッコは、回帰モデルを示します。カッコ内の数は、そのモデルに含まれる効果を示します。

混合型の方法

p^e> > 20 の場合、変数増加ステップワイズ法とすべてのサブセット法を結合する混合型の方法を適用します。

最適サブセットに選択した基準と同じ変数増加ステップワイズ法を使用して効果を選択します。p^s は、変数増加ステップワイズ法で選択して効果の数です。

p^s の値に応じて、次のいずれかの方法を適用します。

- $p^s \leq 20$ の場合、上記で説明しているように選択した効果のすべてのサブセットについて包括的な検索を行います。
- $20 < p^s \leq 40$ の場合、すべての p^s の効果のタイプ III の平方和検定の p 値に基づいて p^s - 20 を選択(「モデル評価 」(p.232) の ANOVA を参照) してそれらをモデルに投入し、上記で示した方法で、残りの 20 の効果に包括的な検索を行います。
- $40 < p^s$ の場合、何も行わず、最適モデルは p^s の効果であるモデルであると想定します (選択したモデルが変数増加ステップワイズ法に基づく警告メッセージが表示)。

モデル評価

次の出力統計が利用できます。

ANOVA

重み付き平方和

$$SS_t = \sum_{i=1}^{n} w_i (y_i - \overline{y})^2 = (N-1)\, S_{yy} \text{ with d.f. } = df_t = N-1$$

d.f. は、自由度を示します。これは、「修正総和の平方和」といいます。

重み付き残差平方和

$$SS_e = \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2 = \tilde{r}_{yy}\, (N-1)\, S_{yy}$$

d.f. = df_e = N - p^c となります。これは「誤差の平方和」となります。

重み付き回帰平方和

$$SS_r = \sum_{i=1}^{n} w_i (\hat{y}_i - \overline{y})^2 = (1 - \tilde{r}_{yy})\, (N-1)\, S_{yy} = SS_t - SS_e$$

d.f. = $df_r = p^*$ となります。定数項がある場合、「修正モデルの平方和」といいます。

回帰平均平方誤差

SS_r/df_r

残差平均平方誤差

SS_e/df_e

修正モデルの F 統計量

$$F = \frac{SS_r/df_r}{SS_e/df_e} = \frac{SS_r \cdot df_e}{SS_e \cdot df_r}$$

自由度が df_r および df_e の F 分布となり、対応する p 値をそれに応じて計算できます。

各効果のタイプ III 平方和

効果 j のタイプ III SS（平方和）$j = 1, \ldots, p^e$, を計算する場合、タイプ III 検定行列 L_i をまず構築する必要があります。L_i の構造は、$H_\omega = \left(X^T D X\right)^- X^T D X$, の生成行列に基づき、この場合 $D = diag\left(g_1, \ldots, g_n\right)$ となり、$L_i \beta$ が推定可能となります。これには、ある一定の効果およびその効果を含む複数の効果に対してのみのパラメータが関与しています。タイプ III の分析の場合、L_i はモデルで指定された効果の順番には依存しません。このような行列が作成できない場合、効果は検定不能です。各効果 j の場合、タイプ III SS が次のように計算されます。

$$\mathbf{S}_j = \hat{\beta}^T \mathbf{L}_j^T \left(\mathbf{L}_j \mathbf{G} \mathbf{L}_j^T\right)^{-1} \mathbf{L}_j \hat{\beta}$$

この場合 $\mathbf{G} = \left(\mathbf{X}^T \mathbf{W} \mathbf{X}\right)^-$ となります。

各効果の F 統計量

効果 j の SS を使用して、次のように仮説検定 H_0: $L_i \beta = 0$ の F 統計量も計算します。

$$F_j = \frac{\mathbf{S}_j / r_j}{SS_e / df_e}$$

この場合、r_j は、$\mathbf{L}_i$ の完全な行の順位となります。自由度が r_i と df_e である F 分布となり、p 値をそれぞれ計算できます。

モデルの要約

調整済み R2 乗

$$\text{adj.}R^2 = 1 - \frac{SS_e/df_e}{SS_t/df_t} = R^2 - \frac{\left(1-R^2\right)p^*}{df_e} = 1 - \frac{df_t \times \tilde{r}_{yy}}{df_e}$$

ここで、

$$R^2 = \frac{SS_r}{SS_t} = 1 - \frac{SS_e}{SS_t} = 1 - \tilde{r}_{yy}.$$

モデル情報基準

補正赤池情報量基準 (AICC)

$$AICC = N\ln\left(\frac{SS_e}{N}\right) + \frac{2p^cN}{N-p^c-1}$$

係数と統計の推論

モデル選択プロセスの後、スイープされた相関行列から係数と関連する統計量を取得できます。次の統計は R 行列について計算します。

非標準化係数推定

$$\hat{\beta}_j = b_j\sqrt{\frac{S_{yy}}{S_{jj}}} = \tilde{r}_{jy}\sqrt{\frac{S_{yy}}{S_{jj}}}$$

この場合、$j = 1, \cdots, p^*$ となります。

回帰係数の標準誤差

$\hat{\beta}_j$ の標準誤差は次のようになります。

$$\hat{\sigma}_{\hat{\beta}_j} = \sqrt{var\left(\hat{\beta}_j\right)} = \sqrt{\frac{\tilde{r}_{jj}\tilde{r}_{yy}S_{yy}}{S_{jj}df_e}}$$

定数項の推定

定数項は、次のように、モデル内の他のすべてのパラメータによって推定されます。

$$\hat{\beta}_0 = \overline{y} - \sum_{j=1}^{p}\hat{\beta}_j\overline{X}_j$$

$\hat{\beta}_0$ の標準誤差は次のように推定されます。

$$\hat{\sigma}_{\hat{\beta}_0} = \sqrt{\hat{\sigma}^2_{\hat{\beta}_0}}$$

ここで、

$$\begin{aligned}\hat{\sigma}^2_{\hat{\beta}_0} &= \frac{(N-1)\tilde{r}_{yy}S_{yy}}{N(N-p^*-1)} + \sum_{j=1}^{p}\overline{X}_j^2\hat{\sigma}^2_{\hat{\beta}_j} + 2\sum_{j=1}^{p-1}\sum_{k=j+1}^{p}\overline{X}_k\overline{X}_j cov\left(\hat{\beta}_k,\hat{\beta}_j\right)\\ &= \frac{SS_e}{N\times df_e} + \sum_{j=1}^{p}\overline{X}_j^2\hat{\sigma}^2_{\hat{\beta}_j} + 2\sum_{j=1}^{p-1}\sum_{k=j+1}^{p}\overline{X}_k\overline{X}_j\frac{\tilde{r}_{kj}\times SS_e}{\sqrt{S_{kk}S_{jj}}\times(N-1)df_e}.\end{aligned}$$

$\hat{\sigma}^2_{\hat{\beta}_0} = \frac{(N-1)\tilde{r}_{yy}S_{yy}}{N(N-p^*-1)} + \sum_{j=1}^{p}\overline{X}_j^2\hat{\sigma}^2_{\hat{\beta}_j} + 2\sum_{j=1}^{p-1}\sum_{k=j+1}^{p}\overline{X}_k\overline{X}_j cov\left(\hat{\beta}_k,\hat{\beta}_j\right)$ となり
$cov\left(\hat{\beta}_k,\hat{\beta}_j\right)$ は、パラメータ推定共分散行列の k 番目の行と j 番目の列となります。

回帰係数の t 統計量。

$$t = \frac{\hat{\beta}_j}{\hat{\sigma}_{\hat{\beta}_j}} = \tilde{r}_{jy}\sqrt{\frac{df_e}{\tilde{r}_{yy}\tilde{r}_{jj}}}$$

この場合、$j = 1,\cdots,p^*$ で、自由度は df_e となり、p 値を計算できます。

100(1−α)% の信頼区間

$$\hat{\beta}_j \pm \hat{\sigma}_{\hat{\beta}_j} \times t_{\alpha/2,df_e}$$

注 :冗長パラメータの場合、係数の推定は 0 に設定され、標準誤差、t 統計量、および信頼区間を欠損値に設定します。

スコアリング

予測値

$$\hat{y}_k = \sum_{i=0}^{p} x_{ki}\hat{\beta}_i, k = 1,\ldots,n.$$

診断

次の値を計算して、さまざまな診断グラフやテーブルを作成します。

Residuals

$$e_k = y_k - \hat{y}_k$$

スチューデント化残差

標準誤差に対する残差の比率です。

$$SRES_k = \frac{e_k}{s\sqrt{\frac{(1-h_k)}{g_k}}}$$

この場合、s は平均平方誤差の平方根となります。つまり、$s = \sqrt{SS_e/df_e}$ となり、h_k は k 番目のレバレッジ値となります（下記参照）。

Cook の距離

$$COOK_k = \frac{e_k^2 h_k g_k}{s^2(1-h_k)^2 p^c}$$

ここで、「レバレッジ」

$$h_k = g_k \mathbf{x}_k \mathbf{G} \mathbf{x}_k^{\mathsf{T}}$$

は ハット行列の k 番目の対角要素です。

$$\mathsf{H} = \mathsf{W}^{1/2}\mathsf{X}\left(\mathsf{X}^{\mathsf{T}}\mathsf{W}\mathsf{X}\right)^{-}\mathsf{X}^{\mathsf{T}}\mathsf{W}^{1/2} = \mathsf{W}^{1/2}\mathsf{X}\mathbf{G}\mathsf{X}^{\mathsf{T}}\mathsf{W}^{1/2}$$

Cook の距離が $\frac{4}{N-p^c}$ より大きいレコードは、影響力があると見なされます（Fox, 1997）。

予測値の重要度

最終完全モデルから一度に予測値を 1 つ削除して、残差平方和（SSe）に基づき、リーブワンアウト法を使用して予測値の重要度を計算します。

最終完全モデルに p 予測値が含まれる場合 $X_1, X_2, \cdots, X_p$、予測値の重要度は次のように計算されます。

1. i=1
2. i>p の場合、ステップ 5 に移動します。
3. 最終完全モデルの R 行列の X_i に関連する該当行および列にスイープ操作を実行します。
4. 現在の $\tilde{\mathbf{R}}$ で最終対角要素を取得し、それを $\tilde{r}_{yy}^{(i)}$ のように示します。そして、X_i の予測値の重要度は、$VI_i = SSE_{(i)} = \tilde{r}_{yy}^{(i)}(N-1)SS_{yy}$ となります。i = i + 1 とし、ステップ 2 に進みます。
5. 次のように、X_i の正規化予測重要度を計算します。

   $NormVI_i = \frac{VI_i + 1/p}{\sum_{i=1}^{p}(VI_i + 1/p)}$

   注 :VI_i の総和が 0 となる場合があるため、正規化予測値の重要度に $1/p$ を導入します。

参照

Belsley, D. A., E. Kuh, および R. E. Welsch. 1980. Regression diagnostics:Identifying influential data and sources of collinearity. New York: John Wiley and Sons.

Dempster, A. P. 1969. Elements of Continuous Multivariate Analysis. Reading, MA: Addison-Wesley.

Fox, J. 1997. Applied Regression Analysis, Linear Models, and Related Methods. Thousand Oaks, CA: SAGE Publications, Inc..

Fox, J., および G. Monette. 1992. Generalized collinearity diagnostics. Journal of the American Statistical Association, 87, 178-183.

Schatzoff, M., R. Tsao, および S. Fienberg. 1968. Efficient computing of all possible regressions. Technometrics, 10, 769-779.

Velleman, P. F., および R. E. Welsch. 1981. Efficient computing of regression diagnostics. American Statistician, 35, 234-242.

章

25

ニューラル ネットワークのアルゴリズム

概要

注 ： 機能が拡張された新しいバージョンのニューラル ネットワーク ノードがこのリリースで使用できます。詳細は、 26 章 p.252 ニューラル ネットワーク アルゴリズム を参照してください。旧バージョンでモデルを作成およびスコアリングできますが、新しいバージョンを使用することをお勧めします。旧バージョンの詳細を参照用に記載しておりますが、それに対するサポートは今後のリリースで廃止されます。

ニューラル ネットワークの基本要素は**ニューロン**です。ニューロンは、多数の入力を取り込んで合計し、それを伝達関数に適用して（通常は非線形）、モデル予測または他のニューロンへの入力として結果を生成する簡単な仮想デバイスです。

ニューラル ネットワークは、このようなニューロンが多数集まって、体系的に接続された構造になります。IBM® SPSS® Modeler では、**フィードフォワード** ニューラル ネットワークが使用されています。これは、**多層パーセプトロン**とも呼ばれています。このようなネットワーク中のニューロン（**ユニット**と呼ばれることもあります）は、複数の層で編成されています。通常は、入力ニューロン用に 1 つの層（**入力層**）、内部処理ユニット用に 1 つまたは複数の層（**隠れ層**）、そして出力ニューロン用に 1 つの層（**出力層**）が存在しています。それぞれの層は、前の層および以降の層と完全に相互接続されています。たとえば、1 つの入力層、1 つの隠れ層、および 1 つの出力層からなるネットワークの場合、入力層の各ニューロンが隠れ層の各ニューロンと接続され、隠れ層の各ニューロンが出力層の各ニューロンと接続されています。

ニューロン間の接続にはそれに対応した重みが付けられます。これにより、あるニューロンが別のニューロンに対して及ぼす影響の強さが決まります。情報は、入力層から処理層を経由して出力層に流れ、予測が生成されます。学習時に接続の重み（接続強度）を調整し、予測フィールドを特定のレコードの目標値と一致させることにより、ネットワークは「学習」し、より良い予測を生成することができます。

一次計算

フィールドのコード化

範囲型フィールドの尺度

ほとんどのデータセットでは、範囲型フィールドの尺度が大幅に異なっています。たとえば、年齢と世帯あたりの自家用車保有数を考えてみましょう。注目する母集団に応じて、年齢は 80 まで、またはそれ以上の値をとります。ただし、世帯あたりの自家用車保有数は、ほとんどのケースで 3 または 4 を超えることはありません。

これらの両方のフィールドをモデルの入力と同じ自然尺度を使用した場合、モデル中で年齢フィールドは、単に「年齢」の値（つまりレコード間の差異）が「世帯あたりの自家用車保有数」より大幅に大きいために、世帯あたりの自家用車保有数よりも大きな重みが与えられることになります。

この尺度による影響を補正するために、範囲型フィールドは同じ尺度を持つように変換されます。IBM® SPSS® Modeler の場合、範囲型フィールドは 0〜1 の範囲の値を持つように再スケールされます。変換には次の式が使用されます。

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}?$$

ここで x'_i はレコード i に対する入力フィールド x の再スケールされた値、x_i はレコード i の x の元の値、x_{min} はすべてのレコードに対する x の最小値、そして x_{max} はすべてのレコードに対する x の最大値を表します。

シンボル値フィールドの数値コード化

レコード間の数値的差異に基づいて計算を行うモデリング アルゴリズムの場合、シンボル値フィールドの処理には特別な労力を費やす必要があります。2 つのカテゴリ間の数値的差異をどのように算出すれば良いでしょうか？

この問題に対する一般的なアプローチ、そして IBM® SPSS® Modeler で使用されているアプローチとして、シンボル値フィールドを数値型フィールドのグループとして記録する方法が挙げられます（元のフィールドの各カテゴリまたは値に対して 1 つの数値型フィールド）。各レコードに対して、レコードのカテゴリに対応する派生フィールドの値は 1.0 に設定されます。また、他のすべての派生フィールドの値は 0.0 に設定されます。このような派生フィールドは、**指標フィールド**と呼ばれることもあります。また、この記録は、**指標のコード化**と呼ばれます。

たとえば、次のデータを例に考えてみましょう。ここで、x は A、B、および C の値をとる可能性があるシンボル値フィールドを表しています。

レコード番号	X	X_1'	X_2'	X_3'
1	B	0	1	0
2	A	1	0	0
3	C	0	0	1

このデータで、元のセット型フィールド x は 3 つの派生フィールド x_1'、x_2'、および x_3' に記録されます。x_1' はカテゴリ A の指標、x_2' はカテゴリ B の指標、および x_3' はカテゴリ C の指標になります。

2 進法によるコード化

ニューラル ネットワークでは、大きなセット型フィールドを処理するために役立つ 2 進法によるコード化を利用できます。デフォルトのコード化では、セット型フィールドの有効な各値に対して、1 つの入力が作成されます。このため大きなセットでは多数の入力が作成され、ネットワークの処理性能が低下したり、大量のメモリーを消費してしまう可能性があります。

2 進法によるコード化では、2 進演算によるコード化を使用して、各セット型フィールドが、数値入力のグループとしてコード化されます。2 進法によるコード化では、セット型フィールドに対する k 個の入力ユニット（k はセットに対する有効な値数）の代わりに、$\log_2(k+1)$ 個の入力ユニット（切り上げ）が使用されます。コード化された値を取得するために、有効なセット値が昇順に列挙され、値の数がその 2 進表記（底 2）に変換されます。これにより、入力値を一意に表す一連の 1 と 0 のセットが得られます。2 進表記は、記録された入力の値を設定します。

たとえば、A、B、および C の 3 つの有効な値を持つセット型フィールドを考えてみましょう。このフィールドは次の表のように、$\log_2(3) \approx 1.58 \rightarrow 2$ 個の派生ユニットとして記録されます。

レコード番号	X	X_1	X_2
1	A	0	1
2	B	1	0
3	C	1	1

フラグ型フィールドのコード化

フラグ型フィールドはシンボル値フィールドの特殊なケースです。ただし、このフィールドはセット中に 2 つの値しか持っていないため、他のセット型フィールドと比べて多少効率的な方法で処理することができます。フラグ型フィールドは単一の数値型フィールドで表されます。値 1.0 は真（true）の値を、値 0.0 は偽（false）の値を表します。フラグ型フィールドの空白値には、値として 0.5 が割り当てられます。

多層パーセプトロン

多層パーセプトロンの学習には、一般化デルタ ルール(Rumelhart, McClelland, および The PDP Research Group, 1986)に基づいて、**誤差逆伝播法**と呼ばれる手法が用いられます。学習時にネットワークに送られる各レコードに対して、情報がネットワーク中を前方向に通過し、出力層から予測が生成されます。この予測は学習レコードの記録された出力値と比較されます。次に、類似パターンの予測を向上するために、予測フィールドと実際の出力値の誤差がネットワーク中を後方向に伝播されて、接続の重みが調整されます。

フィードフォワードの計算

情報は、ネットワーク中を次のように流れていきます。

各入力ニューロンは、その活性化をコード化された入力フィールドの値に設定します。隠れ層または出力層ににおける各ニューロンの活性化は、次のように計算されます。

$$a_i = \sigma\left(\Sigma_j w_{ij} o_j\right),$$

ここで a_i はニューロン i の活性化、j は前の層中のニューロンのセット、w_{ij} はニューロン i とニューロン j 間の接続の重み、o_j は出力ニューロン j、そして $\sigma(x)$ is the **sigmoid** または **logistic** 伝達関数を表します。

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

逆伝播の計算

学習の初めに、ネットワーク中の重みはすべて $-0.5 \le w_{ij} \le 0.5$ の範囲にある無作為の値に設定されます。

レコードは**サイクル**（**エポック**と呼ばれることもあります）に渡されます。各サイクルは、無作為に選択された n 個の学習レコードを包含しています。ここで、n は学習データ中のレコード数を表しています。選択処理は無作為に行われるため、特定のサイクルにおいてある学習レコードが複数回現れるのに、別の学習レコードはまったく現れないようなこともあります。

各レコードに対して、ネットワークに情報が流されて、上記の説明のように予測が生成されます。予測は現在のレコードの学習データ中にある目標値と比較され、その差異をネットワーク内に逆伝播して、重みが更新されます。より精度を向上するために、重みを更新する変更値 Δw は次のように算出されます。

$$\Delta w_{ij}(n+1) = \eta \delta_{pj} o_{pi} + \alpha \Delta w_{ij}(n),$$

ここで、η は学習率パラメータ、δ_{pj} は伝播誤差（後述）、o_{pi} はレコード p に対するニューロン i の出力、α は慣性パラメータ、そして $\Delta w_{ji}(n)$ は前のサイクルにおける w_{ji} の変更値を表します。

学習時に α の値は固定されますが、η の値は学習のサイクルによって異なります。η はユーザーが指定した初期 η から始まり、最小 η の値まで対数的に減少し、次に最大 η の値に戻ってから、もう一度最小 η の値まで減少します。η の値は次のように算出されます。

$$\eta(t) = \eta(t-1) \cdot \exp\left(\log\left(\frac{\eta_{low}}{\eta_{high}}\right)/d\right),$$

ここで d は、ユーザーが指定した η の減衰 サイクル数を表します。$\eta(t-1) < \eta_{low}$ ならば、$\eta(t)$ は η_{high} に設定されます。また、η は、このように学習が完了するまでサイクルが続行されます。

誤差逆伝播値 δ_{pj} は、接続がネットワーク中のどこにあるかに基づいて計算されます。出力ニューロンへの接続の場合、この値は次のように算出されます。

$$\delta_{pj} = (t_{pj} - o_{pj}) o_{pj} (1 - o_{pj})$$

ここで t_{pj} は、レコード p に対する出力 j の目標値を表します。

出力ニューロンに接続しない重みの場合、δ_{pj} は上流の誤り伝播を考慮して計算されます。

$$\delta_{pj} = o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{kj}$$

ここで k はニューロン j の出力が接続されているニューロン群、w_{kj} は現在のニューロンとニューロン k 間の重み、そして δ_{pk} は現在の入力レコードに対する重みの伝播誤差を表しています。

学習時にネットワークに各レコードが渡されると、それに応じて重みが即座に更新されます。

RBFN

RBFN (radial basis function network) は、特殊な種類のニューラル ネットワークです。RBFN は、入力層、隠れ層（**レセプター層**とも呼ばれます)、および出力層の 3 つの層から成り立っています。入力層と出力層は、多層パーセプトロンの場合と同様です。ただし、隠れ層またはレセプター層は、K-Means モデル中のクラスタと似ている、入力パターンのクラスタを表すニューロンから成り立っています。これらのクラスタは **RBF (Radial Basis Function)**、または RBF の中心と入力値のベクトル間の距離の関数に基づいています。入力ニューロンとレセプター ニューロン間の接続（レセプターの重み）は、基本的には K-Means モデルと同様の方法で学習されます(詳細は、 20 章 p.187 概要 を参照してください。)。特に、レセプターの重みは入力フィールドとしか学習されません。学習の第 1 段階では出力フィールドは無視されます。レセプターの重みが最適化され、入力データ中のクラスタが探された後にだけ、レセプターと出力ニューロン間の接続が学習され、予測が生成されます。

RBF の中心の推定

RBF（レセプター）の中心は、K-Means ノードに実装されている K-Means アルゴリズムを使って学習されます。詳細は、 20 章 p.187 概要 を参照してください。ただし、モデリング パラメータは固定されます。

- 反復数は 10 に設定されます
- 収束基準は 0.000001 に設定されます
- セット型フィールドに対して、コード化の値は使用されません（または、コード化値に 1.0 が設定されます）

基底関数の幅の割り当て

各レセプター ニューロンには、それに対応する RBF があります。IBM® SPSS® Modeler で使用される基底関数は、多次元ガウスです。

$$\exp\left(-\frac{{d_i}^2}{2\sigma_i^2}\right),$$

ここで d_i は、クラスタ中心 i からの距離、σ_i はクラスタ／ニューロン i に対する RBF のサイズを記述する尺度パラメータを表します。RBF のサイズをニューロンの**受容領域**として考慮したり、それが応答する入力範囲の幅として考慮することができます。

尺度パラメータ σ_i は、もっとも近くにある 2 つのクラスタ間の距離に基づいて算出されます。

$$\sqrt{\frac{d_1 + d_2}{2}},$$

ここで d_1 はクラスタ中心と、それにもっとも近い他のクラスタの中心間の距離を、d_2 は次に近いクラスタ中心との距離を表します。このように、他のクラスタと近接するクラスタは小さい受容領域を持ち、一方他のクラスタから遠いクラスタは大きい受容領域を持つことになります。

RBFN に対する出力の重みの学習

出力の重みの学習時に、レコードは多層パーセプトロンの場合と同様にネットワークに渡されます。レセプター ニューロンは、自己の活性化をその RBF サイズおよびユーザーが指定するオーバーラップ値 h の関数として計算します。レセプター ニューロン j の活性化は、次のように算出されます。

$$a_j = \exp\left(-\frac{||r - c||^2}{2\sigma_j^2 h}\right),$$

ここで、r はレコード入力のベクトルを、c はクラスタ中心ベクトルを表します。

出力ニューロンは、レセプターまたは隠れニューロンと完全に相互接続されます。レセプター ニューロンは活性化値を渡します。この活性化値は、出力ニューロンにより重み付けられて、合計されます。

$$o_k = \sum_j W_{jk} a_j.$$

出力の重み W_{jk} は、2 層逆伝播ネットワークと同様の方法で学習されます。重みは $-0.001 \le w_{ij} \le 0.001$ の範囲の小さい無作為の値で初期化された後、次の式により各サイクル t で更新されます。

$$w_{jk}(t) = w_{jk}(t-1) + \Delta w_{jk}(t).$$

変更値 $\Delta w_{jk}(t)$ は、次のように算出されます。

$$\Delta w_{jk}(t) = \eta(r_k - o_k)a_i + \alpha \Delta w_{jk}(t-1),$$

この式は、逆伝播方法で使用される式と類似しています(詳細は、 p. 241 逆伝播の計算 を参照してください。)。ただし 1 つ重要な違いがあり、RBFN の出力層の学習では、学習時に η の値は固定されます。

オプションの効果

過度な学習を予防

このオプションは、入力データを学習セットと検証セットの 2 つのセットに分割します。ネットワークは学習セットにより学習し、誤差は各サイクルにおいて検証セットにより評価されます。ユーザーはオプションで乱数ジェネレータのシードを設定し、

検定セットと学習セット間で再現可能な分割を作成できます(同じシードを使用する分割を作成すると、同じでコードを同じサブセットに割り当て、入力のレコードの順序は同じであると想定します)。

停止条件

学習サイクルの停止基準を決定します。

- **デフォルト**:後述する持続性パラメータを使用して、いつネットワークの学習を停止するかを判断します。
- **精度**: 各サイクルの終了時に、後述するようにネットワークの精度が評価されます。精度が指定された値を超えた場合に、学習が終了します。特定のモデリングの問題に合致しないように精度の基準を設定することもできます。この場合、ユーザーにより中断されるまで学習が継続されます。
- **サイクル**:指定したサイクル数の後に、学習が終了します。
- **時間**:指定した時間の経過後に学習が終了します。

持続性

持続性は、サイクル間で改善されない場合にネットワークがどの程度学習を継続するかを決定します。ネットワークは、ユーザーが指定した持続性の値だけサイクルを繰り返しても改善が見られない場合に、学習を終了します。

高速方法

高速方法を選択した場合、単一のニューラル ネットワークが学習されます。デフォルトでは、ネットワークには $\max(3,(n_i+n_o)/20)$ 個のニューロンを含む 1 つの隠れ層があります。ここで n_i は入力ニューロン数を、n_o は出力ニューロン数を表します。ネットワークは、前述した逆伝播法を使用して学習します。

動的方法

動的方法を選択した場合、学習時にネットワークのトポロジーが変更されます。ネットワークが目的の精度を達成するまでニューロンを追加して、パフォーマンスを改善します。動的学習には、トポロジーの発見と最終ネットワークの学習の 2 段階があります。

トポロジーの発見

トポロジーを発見するためには、次の手順を行います。

▶ 学習パラメータを設定します。
- 持続性 : 5
- α : 0.9
- 初期 η : 0.05
- 停止許容度 : 0.02

- それぞれが 2 つのニューロンを含む、2 つの隠れ層を持つネットワークを構築します。初期ネットワークを通常のように、1 サイクルで学習を行います。
- 初期ネットワークのコピーを 2 つ作成します（左および右ネットワーク）。右ネットワークの 2 番目の隠れ層に、1 つニューロンを追加します。
- 両方のネットワークを 1 サイクルで学習し、各ネットワークの全体誤差を判断します。全体誤差は、サイクル中の j 出力と p レコードに渡る、δ_{pj} の合計として算出されます。
- 左ネットワークの誤差が小さい場合はそれを保持して、右ネットワークの「最初の」隠れ層に 1 つのニューロンを追加します。
- 右ネットワークの誤差が小さい場合は、左ネットワークを右ネットワークのコピーと置換して、右ネットワークの「2 番目の」隠れ層にニューロンを追加します。
- 他のサイクルで両方のネットワークを学習し、停止条件を満たすまで学習／増加サイクルを繰り返します。 詳細は、 p. 243 オプションの効果 を参照してください。

η の調整

動的方法による学習では、η の変更によりそれまでのネットワークのパフォーマンスが考慮されます。各サイクルにおいて、2 つのベクトルが計算されます。1 つは、サイクルに渡る重みの変化に基づいた**動向**です。

$$M(t) = 2[W(t) - W(t-1)]$$

ここで $W(t)$ はサイクル t における重みのベクトルで、$W(t-1)$ は前のサイクルにおける重みのベクトルを表します。もう 1 つは、現在のサイクルのモーメントに基づいた **変化** です。

$$C(t) = 0.8 \cdot C(t-1) + M(t).$$

これらのベクトルの大きさの比率は次のようになります。

$$m(t) = \frac{||M(t)||}{||C(t)||}$$

これは、学習を加速するインデックスです。$1 + \frac{||C(t)||}{10}$ 未満のインデックスがある場合、学習は遅くなっているため、η は因子 1.2 で増加されます。インデックスが 5.0 を超える場合、学習は加速しているため、η は因子 $\frac{4}{m(t)}$ で減少されます。

最終ネットワークの学習

良好なトポロジーを発見したら、通常の逆伝播法により次の設定で最終ネットワークの学習が行われます。

- 持続性 ： 5
- α ： 0.9

- 初期 η ： 0.02
- 停止許容度 ： 0.005

複数方法

複数方法を選択した場合、疑似並行モードで複数のネットワークの学習が行われます。指定した各ネットワークが初期化され、すべてのネットワークの学習が行われます。すべてのネットワークに対して停止条件を満たした場合に、もっとも精度の高いモデルが最終モデルとして返されます。 詳細は、 p.243 オプションの効果 を参照してください。

ネットワーク トポロジーは、ユーザーの設定から取得されます。ネットワークの設定がない場合（シンプル オプションを使用する場合など）、IBM® SPSS® Modeler では次のアルゴリズムを使ってネットワーク トポロジーが生成されます。

- 3 ～入力ニューロン数までの、異なる数の隠れ層を持つ単一層ネットワークが生成されます。これらのネットワークは、常に 12 ニューロン以上になり（入力ニューロン数が 12 未満の場合でも）、60 ニューロンを超えることはありません。

  各入力ニューロン数に対して 3、4、7、12 のような順番でネットワークが生成されます。この場合、各ステップでの増分は、前の増分よりも 2 大きい値になります。
- それぞれの単一層ネットワークに対して、2 層ネットワークのセットも作成されます。最初の層は単一層ネットワークと同じ数の隠れニューロンを持ち、2 番目の層のニューロン数はネットワークによって異なります。

  各 2 番目の層のニューロン数に対して 2、5、10、17 のような順番で、最大で 1 番目の隠れ層中のニューロン数までのネットワークが生成されます。この場合、各ステップでの増分は、前の増分よりも 2 大きい値になります。

剪定方法

剪定方法 は、理論的には動的方法の反対です。小さいネットワークから開始してそれを大きくしていく代わりに、剪定方法では大きなネットワークから開始して、入力層および隠れ層から不要なニューロンを削除することにより、徐々にネットワークを**剪定**していきます。

剪定では、隠れニューロンの剪定と入力ニューロンの剪定の 2 つのステージに分けて処理が行われます。これらの各段階の詳細は、後述します。この 2 段階の処理は、すべての停止条件を満たすまで繰り返されます。 詳細は、 p.243 オプションの効果 を参照してください。

隠れニューロンの剪定

最初の段階で、隠れニューロンの剪定処理は次のように行われます。

▶ 学習データでネットワークの学習を行います。

▶ 隠れ層のいずれかの停止条件を満たした場合は、入力ニューロンの剪定処理に進みます。隠れ層の剪定は、次の場合に終了します。

- すべての停止条件を満たした場合

- 隠れ持続性の制限を超えた場合
- 現在のネットワークの誤差が、それまでの最良のネットワークの誤差の 3 倍を超えた場合

隠れ層のいずれの停止条件も満たしていない場合は、隠れニューロンに対して重要度分析が行われ、もっとも弱いニューロンが探されます(詳細は、 p. 249 重要度分析 を参照してください。)。ユーザーが指定した比率（隠れ率に指定）の隠れニューロンが削除され、他の隠れニューロン剪定処理の反復が行われます。

入力ニューロンの剪定

2 番目の段階で、入力ニューロンの剪定処理は次のように行われます。

▶ 学習データでネットワークの学習を行います。

▶ 入力層のいずれかの停止条件を満たした場合、全体の停止条件が確認され、必要に応じて 2 番目の段階の処理が繰り返されます。入力層の剪定は、次の場合に終了します。

- すべての停止条件を満たした場合
- 入力持続性の制限を超えた場合
- 現在のネットワークの誤差が、それまでの最良のネットワークの誤差の 3 倍を超えた場合

入力層のいずれの停止条件も満たしていない場合は、入力ニューロンに対して重要度分析が行われ、もっとも弱いニューロンが探されます。ユーザーが指定した比率（入力率に指定）の入力ニューロンが削除され、他の入力ニューロン剪定処理の反復が行われます。

全体の停止基準

全体の停止基準は、他の学習方法の停止基準と同様に解釈されます。剪定方法のデフォルト値を次に示します。

- 隠れ層の数 ： 1
- 隠れ層中のユニット数 ：$\min(50, round(\log(n_r)\log(k_i + k_o)))$、ここで、$n_r$ は学習データ内のレコードの数、k_i はネットワーク中の入力ユニット数、そして k_o は出力ユニット数を表します。
- α ： 0.9
- 初期 η ： 0.4
- 最大 η ： 0.15
- 最小 η ： 0.01
- 持続性 ： 100
- 全体の持続性 ： 4
- 隠れ持続性 ：$\min\left(10, \max\left(1, \frac{k_i+k_h}{10}\right)\right)$、ここで k_h は隠れユニット数を表します。
- 隠れ率 ： 0.15
- 入力持続性 ：$\min\left(10, \max\left(2, \frac{k_i-k_o}{5}\right)\right)$
- 入力率 ： 0.15

拡張剪定方法

拡張剪定方法は、剪定方法の特殊な方法で、次のパラメータ値が用いられます。

- 隠れ層の数 ： 2
- 隠れ層 1 中のユニット数 ： 30
- 隠れ層 2 中のユニット数 ： 20
- 持続性 ： 200
- 全体の持続性 ： 4
- 隠れ持続性 ： 100
- 隠れ率 ： 0.02
- 入力持続性 ： 100
- 入力率 ： 0.01

空白の処理

ニューラル ネットワークで空白は、欠損値に対する「中間の」値で代用することにより処理されます。欠損値（空白およびヌル）を持つ範囲型およびフラグ型フィールドの場合、欠損値は 0.5 で置換されます。範囲型フィールドの場合、フィールドのデータ型情報中に見つかった範囲制限外の数値は、データ型に定義された範囲に強制変換されます。セット型フィールドの場合、派生する指標フィールドの値はすべて 0.0 に設定されます。

二次計算

ネットワークの精度

［過度な学習を防止］オプションを選択した場合、ネットワークの精度は検証セットに基づきます。それ以外の場合は、学習データに基づきます。計算は、出力フィールドの種類によって異なります。

シンボル値出力フィールド:ネットワークの予測がデータ中の観測値に一致するレコードの割合。

数値型出力フィールド:精度は次のように計算されます。

$$accuracy = \sum_{R} \frac{1 - |t_r - o_r|}{\max(t) - \min(t)} / n_r,$$

ここで R はレコードのセット、t_r はレコード r の目標出力値、o_r はレコード r に対するネットワークにより生成された予測、そして n_r はレコード数を表します。

複数の出力を持つモデルの場合、精度は個別の出力に対する精度の単純平均になります。

重要度分析

重要度分析には、変化するネットワークの活性化と、その結果となる他のネットワーク ペア中の変化の観測（出力の活性化など）が含まれ、ネットワークのどの部分がもっとも重要で、どの部分の重要度が低いかが判断されます。

入力ユニット

ユーザーが重要度分析を選択した場合、および内部的に剪定学習方法を使用する場合、入力ユニットは重要度分析に従って、その重要度をランク付けします。入力の分析時に、入力フィールドは個別のニューロンではなく、分析ユニットとみなされます。そのため、セット型入力フィールドの場合、セット型フィールドを表す入力ニューロンのグループ全体が分析されます。

入力フィールドの重要度は、検証セット中の各レコードに対して、その入力フィールドの値を変化させることにより計算されます。値を変化させるに従って、出力の最大値と最小値が記憶され、出力中の最大差異が計算されます。この最大差異が各レコードに対して計算され、次に平均が算出されます。値は次のように変化します。

フラグ型フィールド:真の値（1.0）と偽の値（0.0）が使用されます。

セット型フィールド:セット型フィールドの有効な各値がテストされます（デフォルトのコード化の場合、セットの各入力ニューロンが順次オンになり、2 進法によるコード化の場合、許される入力値を表すすべての組み合わせが使用される）。

数値型フィールド:元の入力フィールドの範囲をカバーする 5 つの等間隔の値を表す、再スケールされた値 0.0、0.25、0.5、0.75、および 1.0 が使用されます。

隠れユニット

剪定学習方法を選択した場合、隠れユニットも重要度分析に従います。隠れユニットの分析は、次のように行われます。

- テスト データがネットワーク内を通過し、結果がベースラインとして記録されます。
- 最初の隠れユニットは、一時的に重みに 0 を設定することにより、「無効に」されます。テスト データが変更されたネットワークに渡され、結果がベースライン結果と比較されます。各レコードに対して、完全なネットワーク出力と変更されたネットワーク出力間の差異の絶対値が記録され、すべてのテスト レコードに渡ってこれらの値の標準偏差が計算されます。複数出力の場合、各出力に対して個別に値が計算され、次に出力ユニットに渡って値の平均が算出されます。
- この処理がそれぞれの隠れニューロンに対して繰り返され、次にこの値に応じてニューロンがランク付けされます。大きい値は重要なニューロンを、小さい値は重要でないニューロンを示しています。

生成されたモデル／スコアリング

予測値

予測フィールドは、入力としてスコアリングされるレコードをネットワークに渡して、次のように出力フィールドの種類に基づいて出力活性化を受け取ることにより生成されます。

フラグ型フィールド:出力活性化が $o < 0.5$ の場合、偽（0.0）が予測フィールドになります。$o \geq 0.5$ の場合は、真（1.0）が予測フィールドになります。

セット型フィールド:標準のコード化の場合、もっとも高い活性化を持つ入力フィールドを表すグループ中の出力ユニットにより、どの値を予測フィールドとして設定するかが決まります。2 進法によるコード化の場合、出力ユニット グループの出力活性化が、有効なそれぞれのコード化された値と比較され、グループに渡ってもっとも小さい誤差合計を持つコード化された値が、予測フィールドとして選択されます。

数値型フィールド:予測フィールドは、初期数値型フィールドのコード化式を逆にすることにより得られます。

$$p = o \cdot (x_{\max} - x_{\min}) + x_{\min}.$$

Confidence

ニューラル ネットワークの予測における確信度値は、予測対象出力フィールドの種類に基づいて算出されます。確信度の計算は、生成されたモデル ノードの設定に応じて異なります。数値型出力フィールドの場合、確信度値は生成されません。

Difference

差分法では、出力フィールドの種類と使用するコード化に応じて、もっとも近い一致と 2 番目に近い一致を比較することで予測の確信度を計算します。

- **フラグ型フィールド**:確信度は $c = 2 \cdot |0.5 - o|$ として計算されます。ここで o は出力ユニットの出力活性化を表します。
- **セット型フィールド**: 標準のコード化の場合、確信度は $c = o_1 - o_2$ として計算されます。ここで o_1 はもっとも高い活性化を持つユニットのフィールド グループ中の出力ユニットを、o_2 は 2 番目に高い活性化を持つユニットを表します。

  2 進法によるコード化の場合、もっとも近い一致および 2 番目に近い一致に対して、出力活性化を比較する誤差合計とコード化されたセット値が計算され、確信度が $c = e_2 - e_1$ として計算されます。ここで e_2 は 2 番目に近い一致に対する誤差を、e_1 はもっとも近い一致の誤差を表します。

Softmax

Softmax は、多項ロジスティック変換と同じで、確信度値に対する確率的解釈を行います。

- **セット型フィールド :**各出力カテゴリの確信度は次のように計算されます。

$$c_i = \frac{e^{o_i}}{\sum_{i=1}^{k} e^{o_i}},$$

ここで o_i はカテゴリ i に対応する出力ユニットの活性化、k は出力カテゴリの数を表します。

空白の処理

スコアリングの空白処理は、モデル構築時の空白処理と同じです。 詳細は、p. 248 空白の処理 を参照してください。

章

26

ニューラル ネットワーク アルゴリズム

ニューラル ネットワークは、データ内の不明で、複雑であると思われるパターンを検出ことによって、1 つまたは複数の予測値に基づいて、連続型またはカテゴリ型対象を予測します。

モデルの精度の拡張、モデルの安定性の拡張、非常に大きいデータセットを扱うアルゴリズムの詳細は、「アンサンブル アルゴリズム」（ p. 124 ）を参照してください。

多層パーセプトロン

多層パーセプトロン（MLP）は、最大 2 つの隠れ層を持つ、フィードフォワードの監視学習ネットワークです。MLP ネットワークは、1 つまたは複数の対象の予測エラーを最小限にする、1 つまたは複数の予測値の関数です。予測フィールドおよび対象フィールドは、カテゴリ型フィールドと連続型フィールドが混在している場合があります。

表記

この章では特に明記しない限り、多層パーセプトロンに次の表記を使用します。

$X^{(m)} = \left(x_1^{(m)}, ..., x_P^{(m)}\right)$	入力ベクトル、パターン m、m=1,...M です。
$Y^{(m)} = \left(y_1^{(m)}, ..., y_R^{(m)}\right)$	対象ベクトル、パターン m です。
I	入力層を差し引いた、層の数。
J_i	層 i の単位数。J_0 = P, J_i = R。バイアス単位は差し引き。
Γ^c	一連のカテゴリ型出力。
Γ	一連の連続型出力。
Γ_h	1-of-c コーディングされた h 番目のカテゴリ型フィールドを含む $Y^{(m)}$ のサブベクトル。
$a_{i:j}^m$	層 i の単位 j、パターン m で $j = 0, ..., J_i; i = 0, ..., I$ となります。
$w_{i:j,k}$	層 i-1, unit j から層 i に移行した重み、単位 k。$a_{i-1:j}^m$ およびバイアス $a_{i:0}^m$ に繋がる重みはありません。つまり、j の $w_{i:j,0}$ はありません。
$c_{i:k}^m$	$\sum_{j=0}^{J_{i-1}} w_{i:j,k} a_{i-1:j}^m$, i=1,...,I
$\gamma_i(c)$	層 i の活性化関数。
w	すべての重みを含む重みのベクトル $\left(w_{1:0,1}, w_{1:0,2}, ..., w_{I:J_{I-1},J_I}\right)$。

アーキテクチャ

MLP ネットワークの一般的なアーキテクチャは、次のようになります。

入力層:J_0=P units, $a_{0:1},\cdots,a_{0:J_0}$; with $a_{0:j}=x_j$

i 番目の隠れ層:J_i の単位、$a_{i:1},\cdots,a_{i:J_i}$、$a_{i:k}=\gamma_i\left(c_{i:k}\right)$ および $c_{i:k}=\sum_{j=0}^{J_{i-1}} w_{i:j,k}a_{i-1:j}$ で、この場合 $a_{i-1:0}=1$ となります。

出力層:J_I=R の単位で、$a_{I:1},\cdots,a_{I:J_I}$、で $a_{I:k}=\gamma_I\left(c_{I:k}\right)$ および $c_{I:k}=\sum_{j=0}^{J_1} w_{I:j,k}a_{i-1:j}$ の場合、$a_{i-1:0}=1$ となります。

各層のパターン インデックスとバイアス項目は、その層の単位の合計数にカウントされません。

活性化関数

双曲線タンジェント

$$\gamma\left(c\right)=\tanh\left(c\right)=\frac{e^{c}-e^{-c}}{e^{c}+e^{-c}}$$

この関数は、隠れ層に使用されます。

恒等式

$$\gamma(c)=c$$

この関数は、連続型対象がある場合、出力層に使用されます。

Softmax

$$\gamma\left(c_k\right)=\frac{\exp\left(c_k\right)}{\sum_{j\in\Gamma_h}\exp\left(c_j\right)}$$

この関数は、すべての対象がカテゴリ型である場合、出力層に使用されます。

誤差関数

平方和

$$E_T\left(w\right)=\sum_{m=1}^{M}E_m\left(w\right)$$

ここで、

$$E_m(w) = \frac{1}{2}\sum_{r=1}^{R}\left(y_r^{(m)} - a_{I:r}^m\right)^2$$

この関数は、連続型対象がある場合に使用されます。

クロスエントロピー

$$E_T(w) = \sum_{m=1}^{M} E_m(w)$$

ここで、

$$E_m(w) = -\sum_{r\in\Gamma^c} y_r^{(m)} \log\left(\frac{a_{I:r}^m}{y_r^{(m)}}\right)$$

この関数は、すべての対象がカテゴリ型である場合に使用されます。

エキスパート アーキテクチャの選択

エキスパート アーキテクチャの選択によって、1 つのの隠れ層に「最適な」隠れ層の単位を決定します。

無作為サンプルがデータ セット全体から取得され、学習サンプル（70%）と検定サンプル（30%）に分割されます。無作為サンプルのサイズは N = 1000 です。データセット全体のレコード数が N 未満の場合、それらのすべてのを使用します。学習データセットおよび検定データセットが別々に提供されている場合、学習および検定の無作為サンプルはそれぞれのデータセットから取得する必要があります。

K_{min} および K_{max} である場合、アルゴリズムは次のようになります。

1. k の隠れ単位の初期ネットワークから開始します。デフォルトは k=min(g(R,P),20,h(R,P)) です。この場合、

$$g(R,P) = \begin{cases} \lfloor 4.5 + \sqrt{P+R} \rfloor & R < 5, P \geq 8 \\ \lfloor 0.5 + 0.5(P+R) \rfloor & \text{otherwise} \end{cases}$$

この場合、$\lfloor x \rfloor$ は x 以下の最も大きい整数となります。$h(R,P) = \left[\frac{M-R}{P+R+1}\right]$ は隠れ単位の最大数で、学習セット全体のレコード数より重みが大きくなります。

$k < K_{min}$ の場合、$k = K_{min}$ を設定します。または $k > K_{max}$ の場合、$k = K_{max}$ を設定します。代替のシミュレーションされたアニーリングと学習プロシージャを介してこのネットワークを学習します（ステップ 1 ～ 5）。

2. $k>K_{min}$ の場合、DOWN=TRUE を設定します。学習誤差率が > 0.01 の場合、DOWN=FALSE を設定します。または停止して初期ネットワークを報告します。

3. DOWN=TRUE の場合、最も弱い隠れ単位を削除します（下記参照)k=k-1。そうでない場合、隠れ単位を追加します k=k+1。

4. 古い重みの初期の重みとして以前適合した重みと、新しい重みの無作為な重みを使用して、停止条件を満たすまで、代替のシミュレーションされた兄ーリングおよび学習プロシージャを使用して、ネットワークの古い重みと新しい重みを学習します（ステップ 3 ～ 5)。

5. 検定データの誤差が削除される場合、次のようになります。

   DOWN=FALSE の場合、k< K_{max} の場合、学習エラーは削除されますが誤差率が、0.01 を超える場合、ステップ 3 に戻ります。k> K_{min} の場合もステップ 3 に戻ります。あるいは、停止して検定エラーが最小のネットワークを報告します。

   DOWN=TRUE 、$|k-k_0|$>1 の場合、停止して検定エラーが最小のネットワークを報告します。または k=k_0 の学習エラーが 0.01 より大きい場合、DOWN=FALSE、k=k_0 を設定してステップ 3 に戻ります。または停止して、初期ネットワークを報告します。

   または停止して検定エラーが最小のネットワークを報告します。

   複数のネットワークの検定エラーが最小である場合、隠れ単位が最も少ないものを選択します。

   この手順で作成された学習誤差率（出力フィールドを予測する出力フィールドの平均値を使用して、学習エラーをモデルの誤差で割る）が 0.1 より大きい場合、誤差率が<=0.1 となる、またはプロシージャが 5 回繰り返されるまで、異なる所期の重みでアーキテクチャ選択を繰り返し、検定エラーが最も小さいものを選択します。

   その重みを初期値としてこのネットワークを使用し、学習セット全体のセットワークを再学習します。

最も弱い隠れ単位

各隠れ単位 j について、j がネットワークから削除される場合、検定データの誤差を計算します。最も弱い隠れ層は、削除時に合計の検定エラーが最も小さい隠れ層です。

学習

重みの推定に関する問題は、次のようなものがあります。

- 重みの初期化。無作為サンプル を取得し、無作為サンプルに代替のシミュレーションされたアニーリングと学習プロシージャを適用し、初期の重みを派生します。ステップ 3 の学習は、デフォルトの学習パラメータをすべて使用して実行します。
- 重みに関する誤差関数の二次導関数の計算。これは、誤差逆伝搬法アルゴリズムを使用して解決します。
- 推定重みの更新。これは、傾斜降下法または尺度共役勾配法によって解決されます。

代替シュミレーテッド アニーリングおよび学習

次のプロシージャでは、シミュレーテッド アニーリングと学習を最大 K_1 回使用します。シミュレーテッド アニーリングを使用して、ローカルの最小値を摂動し、学習で検出されたローカルの最小値を K_2 回分割します。分割が正常に行われた場合、シミュレーティッド アニーリングは、次の学習により適切な初期重みを設定します。このプロシージャを K_3 回繰り返して、グローバル最小値を検出することをお勧めします。このプロシージャは、大きいデータセットの場合に手間がかかるため、無作為サンプルにのみ使用して、初期の重みとアーキテクチャ選択を検索します。K_1=K_2=4、K_3=3 とします。

1. [a_0−a、a_0+a] の間で K_2 の重みベクトルを生成します。この場合、a_0=0 および a=0.5 となります。各重みベクトルの学習エラーを計算します。最小学習誤差を初期重みとして指定する重みを選択します。
2. k_1=0 を設定します。
3. 指定した初期重みでネットワークを学習します。学習した重み **w** を呼び出します。
4. 学習誤差率が <= 0.05 の場合、k_1 のループを停止して、**w** をループの結果として使用します。または k_1 = k_1+1 を設定します。
5. k_1<K_1 の場合、[a(k_1), a(k_1)] に K_2 の異なる無作為ノイズ $\mathbf{w}_n$ を追加して古い重みを摂動し、K_2 を新しい重み $\mathbf{w}' = \mathbf{w} + \mathbf{w}_n$ を形成します。この場合、$a(k_1) = (0.5)^{k_1} a$ となります。$\mathbf{w}_{\min}$ を、摂動したすべての重みの最小学習エラーを指定する重みとします。$E_T(\mathbf{w}_{\min}) < E_T(\mathbf{w})$ の場合、初期重みを $\mathbf{w}_{\min}$ に設定し、ステップ 3 に戻ります。または停止して **w** を最終結果として報告します。

   または k_1 のループを停止し、**w** をループの結果として使用します。

   重みにの学習誤差率が 0.1 を超える場合、学習誤差率が <=0.1 となるまでこのアルゴリズムを繰り返すか、プロシージャを K_3 回繰り返し、k_1 のループの結果で最も学習エラーが小さい重みを選択します。

誤差伝搬法

誤差伝搬法を使用して、重みに関する誤差関数の最初の変動関数を計算します。

まず、$\gamma'(c) = \begin{cases} 1 - [\gamma(c)]^2 & \text{tanh} \\ 1 & \text{identity} \end{cases}$ となります。

誤差伝搬法アルゴリズムは次のようになります。

それぞれの i、j、k について、$\frac{\partial E_T}{\partial w_{i:k,j}} = 0$ を設定します。

グループ T の m について、p=1,...,J_I について、次のようになります。

$$\delta^m_{I:p} = \frac{\partial E_m}{\partial c^m_{I:p}} = \begin{cases} a^m_{I:p} - y^{(m)}_p & \text{クロスエントロピーが使用される場合} \\ \gamma'_I\left(c^m_{I:p}\right)\left(a^m_{I:p} - y^{(m)}_p\right) & \text{層でない場合} \end{cases}$$

i=I,...,1（出力層から開始）の場合、j=1,...,J_i の場合、k=0,...,J_{i-1} の場合

- $\frac{\partial E_m}{\partial w_{i:k,j}} = \delta^m_{i:j} a^m_{i-1:k}$ とします。この場合、$\delta^m_{i:j} = \frac{\partial E_m}{\partial c^m_{i:j}}$ となります。
- $\frac{\partial E_T}{\partial w_{i:k,j}} = \frac{\partial E_T}{\partial w_{i:k,j}} + \frac{\partial E_m}{\partial w_{i:k,j}}$ を設定します。
- k > 0 および i > 1 の場合、$\delta^m_{i-1:k} = \gamma'_{i-1}\left(c^m_{i-1:k}\right)\sum_{j=1}^{J_i}\delta^m_{i:j} w_{i:k,j}$ とします。

この場合、$\sum_{i=0}^{I-1}(J_i+1)J_{i+1}$ 要素のベクトルとなり、$E_T(w_k)$ の傾斜を形成します。

傾斜降下法

学習率パラメータが η_0（0.4 に設定）、慣性率が α（0.9 に設定）の場合、傾斜降下法は次のようになります。

1. k=0 とします。重みベクトルを w_0 に初期化し、学習率を η_0 に初期化します。$\Delta w_0 = 0$ とします。
2. すべてのデータを読み込み、$E_T(w_k)$ を傾斜 $g_k = \nabla E_T(w_k)$ を検出します。$|g_k| < 10^{-6}$ の場合、停止して現在のネットワークを報告します。
3. $\eta_k|g_k| \le \alpha|\Delta w_k|$ の場合、$\alpha = 0.9\eta_k\frac{|g_k|}{|\Delta w_k|}$ となります。ここで、最も急な傾斜降下方向が、次のステップのもみの変化を示すことを確認します。このステップを実行しない場合、次のステップの重みが最も急な降下と反対方向に変化し、η_k がどれだけ小さくても、誤差は小さくなりません。
4. $v = w_k - \eta_k g_k + \alpha\Delta w_k$ とします。
5. $E_T(v) < E_T(w_k)$ の場合、$w_{k+1} = v$，$\Delta w_{k+1} = w_{k+1} - w_k$、そして $\eta_{k+1} = \eta_k$ とし、$\eta_k = .5\eta_k$ の場合、ステップ 3 に戻ります。
6. 停止規則を満たす場合、終了して、停止基準で指定されているとおりにネットワークを報告します。または k=k+1 の場合、ステップ 2 に戻ります。

モデルの更新

学習率パラメータが η_0（0.4 に設定）および η_{low}（0.001 に設定）、慣性率が α（0.9 に設定）、そして学習率減衰因子が β = (1/pK)*ln(η_0/η_{low}) の場合、オンラインおよび小バッチ学習の傾斜降下法は次のようになります。

1. k=0 とします。重みベクトルを w_0 に初期化し、学習率を η_0 に初期化します。$\Delta w_0 = 0$ とします。
2. T_k のレコードを読み込み（T_k は無作為に選択）、$E_{T_k}(w_k)$ とその傾斜 $g_k = \nabla E_{T_k}(w_k)$ を検出します。
3. $\eta_k|g_k| \le \alpha|\Delta w_k|$ の場合、$\alpha = 0.9\eta_k\frac{|g_k|}{|\Delta w_k|}$ となります。ここで、最も急な傾斜降下方向が、次のステップのもみの変化を示すことを確認します。このステップを実行しない場合、次のステップの重みが最も急な降下と反対方向に変化し、η_k がどれだけ小さくても、誤差は小さくなりません。

4. $v = w_k - \eta_k g_k + \alpha \Delta w_k$ とします。
5. $E_{T_k}(v) < E_{T_k}(w_k)$ の場合、$w_{k+1} = v$ および $\Delta w_{k+1} = w_{k+1} - w_k$ とし、または $w_{k+1} = w_k, \Delta w_{k+1} = \Delta w_k$
6. $\eta_{k+1} = e^{-\beta}\eta_k$ とします。$\eta_{k+1} < \eta_{low}$ の場合、$\eta_{k+1} = \eta_{low}$ とします。
7. 停止規則を満たす場合、終了して、停止基準で指定されているとおりにネットワークを報告します。または k=k+1 の場合、ステップ 2 に戻ります。

尺度共役勾配法

はじめに、重みベクトルを $\mathbf{w}_0$ に初期化し、N を重みの合計とします。

1. k=0。スカラーを $\lambda_0 = 5.0\mathsf{E}-7, \sigma = 5.0\mathsf{E}-5, \overline{\lambda_0} = 0$ とします。$\mathbf{r}_0 = \mathbf{p}_0 = -\nabla E_T(\mathbf{w}_0)$、success=true とします。
2. success=true の場合、次の二次導関数の情報を検出します。$\sigma_k = \frac{\sigma}{|\mathbf{p}_k|}$, $\mathbf{s}_k = \frac{\nabla E_T(\mathbf{w}_k + \sigma_k \mathbf{p}_k) - \nabla E_T(\mathbf{w}_k)}{\sigma_k}$, $\delta_k = \mathbf{p}_k^t \mathbf{s}_k$。この場合、スーパースクリプト t は入れ替えを示します。
3. $\delta_k = \delta_k + (\lambda_k - \overline{\lambda_k})|\mathbf{p}_k|^2$ とします。
4. $\delta_k \le 0$ の場合、Hessian 正値定が次のようになります。$\overline{\lambda_k} = 2\left(\lambda_k - \frac{\delta_k}{|\mathbf{p}_k|^2}\right)$, $\delta_k = -\delta_k + \lambda_k |\mathbf{p}_k|^2$, $\lambda_k = \overline{\lambda_k}$.
5. 次のようにステップ サイズを計算します。$\mu_k = \mathbf{p}_k^t \mathbf{r}_k$, $\alpha_k = \frac{\mu_k}{\delta_k}$.
6. 比較パラメータを計算します。$\Delta_k = 2\delta_k \frac{[E_T(\mathbf{w}_k) - E_T(\mathbf{w}_k + \alpha_k \mathbf{p}_k)]}{\mu_k}$.
7. $\Delta_k \ge 0$ の場合、誤差は少なくなります。$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{p}_k$、$\mathbf{r}_{k+1} = -\nabla E_T(\mathbf{w}_{k+1})$ とし、$|\mathbf{r}_{k+1}| < 10^{-6}$ の場合、$\mathbf{w}_{k+1}$ を最終重みベクトルとして返し、終了します。$\overline{\lambda_k} = 0$, success=true とします。k mod N=0 の場合、次のアルゴリズムを再開します。$\mathbf{p}_{k+1} = \mathbf{r}_{k+1}$、または $\beta_k = \frac{|\mathbf{r}_{k+1}|^2 - \mathbf{r}_{k+1}^t \mathbf{r}_k}{\mu_k}$、$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$ とします。$\Delta_k \ge .75$ の場合、次の尺度パラメータを小さくします。$\lambda_k = \frac{1}{4}\lambda_k$。または ($\Delta_k < 0$ の場合)、$\overline{\lambda_k} = \lambda_k$, success=false とします。
8. $\Delta_k < 0.25$ の場合、次の尺度パラメータを大きくします。$\lambda_k = \lambda_k + \frac{\delta_k(1-\Delta_k)}{|\mathbf{p}_k|^2}$
9. success=false の場合、ステップ 2 に戻ります。または停止規則を満たす場合、終了して停止基準に指定されているようにネットワークを報告します。または k=k+1、$\overline{\lambda_{k+1}} = \overline{\lambda}_k$, $\lambda_{k+1} = \lambda_k$ とし、ステップ 2 に戻ります。

注 : 各反復には最低 2 回のデータ パスが必要です。

停止規則

学習は、少なくとも 1 回の完全なデータ パスで行われます。次の基準に従って、検索が停止します。これらの停止基準は、表示順にチェックする必要があります。モデルを新規作成する場合、反復の完了後にチェックします。モデルの更新後、基準 1、

3、4、5 および 6 のチェックはデータ パスの完了後に行われ、基準 2 のチェックは反復後に行われます。以下の説明において、「ステップ」は、モデルを新規作成する場合の 1 回の反復と、モデル更新実行時の 1 回のデータ パスを意味します。E_1 を現在の最小誤差、K_1 を学習セットに出現する反復、E_2 および K_2 がオーバーフィット防止セットの反復、K3=min(K_1,K_2) となるようにします。

1. 各ステップの終わりに、オーバーフィット防止セットの合計誤差を計算します。ステップ K_2 から、検定エラーが次の n=1 のステップで E_2 より小さくならない場合、停止します。ステップ K_2 で重みを報告します。オーバーフィット防止セットがない場合、この基準はモデルの新規作成には使用されません。オーバーフィット防止セットがない場合のモデルの更新について、各ステップの終わりに学習データの合計誤差を計算します。ステップ K_1 から、学習エラーが次の n=1 のステップで E_1 より小さくならない場合、停止します。ステップ K_1 で重みを報告します。

2. 検索は、最大持ち時間を越えて継続しています。モデルを新規作成するために、ステップ K_3 で重みを報告します。モデル更新に対し、学習が現在のステップを完了する前に停止した場合でも、ステップを完了したものとして扱います。学習データセットおよび検定データセットの現在のエラーを計算し、それに応じて E_1、K_1、E_2、K_2 を更新します。ステップ K_3 で重みを報告します。

3. 検索は、最大データ パス数を超えて継続しています。ステップ K_3 で重みを報告します。

4. 学習エラーの相対変化が小さい場合、停止します。$\frac{|E_T(w_k)-E_T(w_{k-1})|}{\frac{1}{2}(E_T(w_k)+E_T(w_{k-1})+\delta)} < \epsilon_1$ for $\delta = 10^{-10}$ and $\epsilon_1 = 10^{-4}$、この場合 w_{k-1}, w_k は 2 つの継続ステップの重みベクトルです。ステップ K_3 で重みを報告します。

5. 初期エラーと比較して、現在の学習誤差率が小さい場合$\left|\frac{E_T(w_k)}{\overline{E}_T+\delta}\right| < \epsilon_2$ となり ($\delta = 10^{-10}$ および $\epsilon_2 = 10^{-3}$ の場合)、ここで $\overline{E}_T$ は出力フィールドの平均を使用してそのフィールドを予測するモデルの合計誤差です。$\overline{E}_T$ は、誤差関数で $a_{I:r}^m = \frac{1}{M}\sum_{m=1}^{M} y_r^{(m)}$ を使用して計算されます。w_k はあるステップの重みベクトルです。ステップ K_3 で重みを報告します。

6. 現在の制度は指定した閾値を満たします。精度は、オーバーフィット防止セットがある場合はそれに基づき、ない場合は麦秋セットに基づいて計算されます。

   注 : 基準 4 および 5 で、学習データ全体の合計誤差が必要となります。モデル更新に際し、オーバーフィット防止セットがある場合、これらの基準はチェックされません。

モデル更新

新しいレコードを使用できる場合、シナプスの重みを更新できます。新しいレコードはサイズ R = min(M,2N,1000) のグループに分割されます。この場合 M は学習レコード数、N はネットワーク内の重み数になります。1 回のデータ パスが新しいグループで行われ、重みを更新します。新しいグループの最後に通常のグループの 4 分の 1 を超えるレコード数がある場合、正常に処理されます。そうでない場合、内部バッ

ファとなり、これらのレコードを次の更新で使用できます。最後の更新の後、バッファ状態にある未使用のレコードがいくつか損失する場合があります。

放射基底関数

放射基底関数（RBF）ネットワークは、放射基底関数層という、隠れ層が 1 つだけあるフィードフォワードの監視学習ネットワークです。RBF ネットワークは、1 つまたは複数の対象の予測エラーを最小限にする、1 つまたは複数の予測値の関数です。予測フィールドおよび対象フィールドは、カテゴリ型フィールドと連続型フィールドが混在している場合があります。

表記

この章では特に明記しない限り、次の表記を使用します。

$X^{(m)} = \left(x_1^{(m)}, ..., x_P^{(m)}\right)$	入力ベクトル、パターン m、m=1,...M です。
$Y^{(m)} = \left(y_1^{(m)}, ..., y_R^{(m)}\right)$	対象ベクトル、パターン m です。
I	入力層を差し引いた、層の数。RBF ネットワークの場合、I=2 となります。
J_i	層 i の単位数。J_0 = P, J_i = R。バイアス単位は差し引き。J_1 は、RBF の単位数です。
$\phi_j\left(X^{(m)}\right)$	入力 $X^{(m)}$ の j 番目の RPF 関数、j=1, …, J_1。
μ_j	ϕ_j の中心、P 次元となります。
σ_j	ϕ_j の幅。P 次元です。
h	RBF オーバーラップ因子。
$a_{i:j}^m$	層 i の単位 j、パターン m で $j = 0, ..., J_i; i = 0, ..., I$ となります。
w_{rj}	RPF 層の r 番目の出力単位および j 番目の隠れ単位を接続する重み。

アーキテクチャ

RBF ネットワークには、次の 3 つの層があります。

入力層:J_0=P units, $a_{0:1}, \cdots, a_{0:J_0}$; with $a_{0:j} = x_j$

RBF 層:J_1 単位、$a_{1:1}, \cdots, a_{1:J_1}$ で、$a_{1:j} = \phi_j(X)$ and $\phi_j(X) = \exp\left(-\sum_{p=1}^{P} \frac{1}{2\sigma_{jp}^2}(x_p - \mu_{jp})^2\right) / \sum_{j=1}^{J_1} \exp\left(-\sum_{p=1}^{P} \frac{1}{2\sigma_{jp}^2}(x_p - \mu_{jp})^2\right)$ となります。

出力層:J_2=R の単位で、$a_{I:1}, \cdots, a_{I:J_2}$、そして $a_{I:r} = w_{r0} + \sum_{j=1}^{J_1} w_{rj}\phi_j(X)$ となります。

誤差関数

平方和誤差が使用されます。

$$E_T(w) = \sum_{m=1}^{M} E_m(w)$$

ここで、

$$E_m(w) = \frac{1}{2}\sum_{r=1}^{R}\left(y_r^{(m)} - a_{I:r}^m\right)^2$$

出力層の恒等式活性化関数による平方和誤差関数は、連続型対象およびカテゴリ型対象の両方に使用できます。連続型対象の場合、$a_{I:r}^m$ は、対象値 $E\left(y_r|X^{(m)}\right)$ の条件式の期待値を概算します。カテゴリ型対象の場合、$a_{I:r}^m$ は、次のようにクラス k の事後確率を概算します。$P\left(y_r = 1|X^{(m)}\right)$

注 :$\Sigma a_{I:r}^m = 1$（合計は同じカテゴリ型対象フィールドのすべてのクラス）で、 $a_{I:r}^m$ は範囲［0, 1］内にない場合があります。

学習

ネットワークは、次の 2 段階で学習されます。

1. **クラスタ化方法によって基本関数を決定する:** 各基本関数の中心および幅が計算されます。
2. **基本関数を指定された重みを決定する:** 指定された基本関数について、重みの通常最小 2 乗回帰推定を計算します。

これらの計算は単純なため、RBF ネットワークを非常に迅速に学習できます。

基本関数の決定

TwoStep クラスタ化アルゴリズムを使用して、RBF の中心と幅を検出します。各クラスタについて、各連続型フィールドの平均値と標準偏差、そして各カテゴリ化フィールドの各カテゴリの割合が派生します。クラスタリングの結果を使用して、j 番目の RBF の中心が次のように設定されます。

$$\mu_{jp} = \begin{cases} \overline{x}_{jp} & \text{if pth field is continuous} \\ \pi_{jp} & \text{if pth field is a dummy field of a categorical field} \end{cases}$$

この場合、$\overline{x}_{jp}$ は、p 番目の入力フィールドの j 番目のクラスタ平均（連続型）、π_{jp} は、p 番目の入力フィールドが対応するカテゴリ型フィールドのカテゴリの割合となります。j 番目の RBF は次のようになります。

$$\sigma_{jp} = h^{1/2}\begin{cases} s_{jp} & \text{if pth field is continuous} \\ \sqrt{p_{jp}(1-p_{jp})} & \text{if pth field is a dummy field of a categorical field} \end{cases}$$

この場合、s_{jp} は、p 番目のフィールドの j 番目のクラスタの標準偏差、h>0 は、RBF のオーバーラップの量を制御する RBF オーバーラップ因子です。一部の σ_{jp} が 0 になる場合があるため、球状のガウス バンプを使用します。つまり、すべての予測値の一般的な幅は

$$\sigma_j = \sqrt{\frac{1}{P}\sum_{p=1}^{P}\sigma_{jp}^2}$$

となります。一部の j の σ_j が 0 の場合、$\min\{\sigma_j : \sigma_j \neq 0,\}_{j=1}^{J_1}$ となるようにします。すべての σ_j が 0 になる場合、すべてを $\sqrt{h}$のように設定します.

多くの予測フィールドがある場合、$\sum_{p=1}^{P}(x_p - \mu_{jp})^2$ が容易に大きくなる場合があるため、σ_j が比較的小さい場合、すべてのレコードとすべての RBF 単位の $\exp\left(-\sum_{p=1}^{P}\frac{1}{2\sigma_j^2}(x_p - \mu_{jp})^2\right)$ が実際は 0 になります。この場合、モデルには定数工しかなくなる場合があるため、ORBF には特に不適切です。この問題を回避するには、入力数に比例するデフォルトのオーバーラップ因子 h を設定（h=1 + 0.1 P）して、σ_j を大きくします。

基本関数の数の自動選択

このアルゴリズムは、合理的な隠れ単位の数を使用し、「最適な」選択を行います。デフォルトでは、適切な範囲［K_1, K_2］は、まず TwoStep クラスタ化方法を使用して自動的にクラスタ数 K を検出することによって決定します。そして ORBF の場合は K_1 = min(K, R)、NRBF の場合は K_1 =max{2, min(K, R)}、そして K_2=max(10, 2K, R) とします。

検定データ セットが指定されると、「最適な」モデルは検定データでエラーが最も小さいモデルとなります。検定データがない場合、BIC（ベイズ情報基準）を使用して、「最適な」モデルを選択します。BIC は次のように定義されます。

$$BIC = MR\ln(MSE) + k\ln(M)$$

この場合、$MSE = \frac{1}{MR}\sum_{m=1}^{M}\sum_{r=1}^{R}\left(y_r^{(m)} - a_{I;r}^m\right)^2$ は平均平方誤差、k= (P+1+R)J_1 (NRBF) および (P+1+R)J_1+R (ORBF) は、モデル内のパラメータ数となります。

モデル更新

欠損値

欠損値の処理を行うオプションは、次のとおりです。

- 欠損値のあるレコードをリストごとに除外する。
- 欠損値を代入する。連続型フィールドは、観測値の最小値および最大値の平均を代入します。カテゴリ型フィールドでは、最も頻度の高いカテゴリを代入します。

出力統計

次の出力統計が利用できます。連続型フィールドの場合、フィールドの調整された値について、出力統計が報告されます。

Accuracy

各連続型対象 r について、次のようになります。

$$accuracy = \frac{1}{n}\sum_{m=1}^{M}\left(1 - \frac{\left|y_r^{(m)} - \hat{y}_r^{(m)}\right|}{\max_m\left(y_r^{(m)}\right) - \min_m\left(y_r^{(m)}\right)}\right)$$

カテゴリ型対象の場合は、予測値が観測値に一致するレコードの割合となります。

予測値の重要度

詳細は、 35 章 p. 342 予測値の重要度アルゴリズム を参照してください。

Confidence

ニューラル ネットワークの予測における確信度値は、予測対象出力フィールドの種類に基づいて算出されます。数値型出力フィールドの場合、確信度値は生成されません。

Difference

差分法では、出力フィールドの種類と使用するコード化に応じて、もっとも近い一致と 2 番目に近い一致を比較することで予測の確信度を計算します。

- **フラグ型フィールド**:確信度は $c = 2 \cdot |0.5 - o|$ として計算されます。ここで o は出力ユニットの出力活性化を表します。
- **セット型フィールド**: 標準のコード化の場合、確信度は $c = o_1 - o_2$ として計算されます。ここで o_1 はもっとも高い活性化を持つユニットのフィールド グループ中の出力ユニットを、o_2 は 2 番目に高い活性化を持つユニットを表します。

  2 進法によるコード化の場合、もっとも近い一致および 2 番目に近い一致に対して、出力活性化を比較する誤差合計とコード化されたセット値が計算され、確信度が $c = e_2 - e_1$ として計算されます。ここで e_2 は 2 番目に近い一致に対する誤差を、e_1 はもっとも近い一致の誤差を表します。

Simplemax

Simplemax は、最も高い予測確率を確信度として返します。

参照

Bishop, C. M. 1995. Neural Networks for Pattern Recognition, 3rd ed. Oxford: Oxford University Press.

Fine, T. L. 1999. Feedforward Neural Network Methodology, 3rd ed. New York: Springer-Verlag.

Haykin, S. 1998. Neural Networks:A Comprehensive Foundation, 2nd ed. New York: Macmillan College Publishing.

Ripley, B. D. 1996. Pattern Recognition and Neural Networks. Cambridge: Cambridge University Press.

Tao, K. K. 1993. A closer look at the radial basis function (RBF) networks. In: Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers, A. Singh, ed. Los Alamitos, Calif.: IEEE Comput. Soc. Press, 401-405.

Uykan, Z., C. Guzelis, M. E. Celebi, および H. N. Koivo. 2000. Analysis of input-output clustering for determining centers of RBFN. IEEE Transactions on Neural Networks, 11, 851-858.

章
27

最適データ分割アルゴリズム

最適データ分割プロシージャで、スケール変数の MDLP（最小記述長原理）離散化を実行します。この方法で、スケール変数を少数の間隔、つまりビンに分割します。その場合、各ビンは離散変数の独立したカテゴリへマッピングされます。

MDLP は、1 変量の監視付き離散方法です。このドキュメントで説明するアルゴリズムは、一般性を失うことなく、カテゴリ型ガイド変数との関係で 1 つの連続型の属性を考慮するのみです- 離散化は、カテゴリ型ガイドについては「最適」です。したがって、入力データ行列の S には、スケール変数 A とカテゴリ型ガイド C の 2 つの列が含まれます。

最適なデータ分割は、分割方法が［最適］に設定されている場合にデータ分割ノード内に適用されます。

表記

この章では特に明記しない限り、次の表記を使用します。

S	スケール変数 A の列とカテゴリ ガイド C の列を含む入力データ行列。各行は、独立した観察結果またはインスタンスです。
A	連続型属性ともいう、スケール変数。
S(i)	S 内の i 番目のインスタンスのための A の値。
N	S 内のインスタンスの数。
D	S 内のすべて異なる値のセット。
S_i	S のサブセット。
C	カテゴリ ガイドまたはクラス属性で、k 個のカテゴリ、またはカテゴリ クラスがあると仮定されます。
T	2 つのビンの間の境界を定義する分割点。
T_A	分割点のセット。
Ent(S)	S のクラス エントロピー。
E(A, T, S)	A 上の T に誘発されたデータ区分の、クラス エントロピー。
Gain(A, T, S)	A 上の分割点 T の情報の対応。
n	等しい頻度の方法の分割点の数を示すパラメータ。
W	各インスタンスの度数を示す重み属性。重み値が整数でない場合は、使用前に最も近い整数に丸められます。たとえば、0.5 は 1 に丸められ、2.4 は 2 に丸められます。重みがないまたは重みが 0.5 未満のインスタンスは使用されません。

Simple MDLP

このセクションでは、Fayyad および Irani により 1993 年に議論された監視付きデータ分割方法 （MDLP） を説明します。

クラス エントロピー

k クラスの C_1、...、C_k があり、$P(C_i, S)$ がクラス C_i のある S 内のインスタンスの比率だとします。クラス エントロピー Ent(S) は、次のように定義されます。

$$Ent(S) = -\sum_{i=1}^{k} P(C_i, S) \log_2 (P(C_i, S))$$

クラス情報エントロピー

インスタンス セット S、連続型属性 A、および分割点 T については、$S_1 \subset S$ は $A \leq T$ の値の S 内のインスタンスのサブセットとし、$S_2 = S - S_1$ とします。T, E(A, T; S) によって誘発されたデータ区分のクラス情報エントロピーは、次のように定義されます。

$$E(A, T; S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

情報の獲得

指定されたインスタンスのセット S、連続型属性 A、および A の分割点 T に対し、分割点 T の情報ゲインは次のようになります。

$$Gain(A, T; S) = Ent(S) - E(A, T; S)$$

MDLP 許容基準

N インスタンスのセット S に対して分割点 T により生じたデータ区分は、次の場合にのみ、許容されます。

$$Gain(A, T; S) > \frac{\log_2 (N-1)}{N} + \frac{\Delta(A, T; S)}{N}$$

上記以外の場合は、却下されます。

ここでは、$\Delta(A, T; S) = \log_2 (3^k - 2) - [k \cdot Ent(S) - k_1 Ent(S_1) - k_2 Ent(S_2)]$ となり、k_i は S のサブセット S_i のクラス数です。

注 : MDLP 許容基準が分割点を判断するのに A と C との間のアソシエーションを使用する一方で、作成されるビン数を少なく維持することも試行されます。したがって、A および C 間のアソシエーションが強ければ分割点が生じないという状況になります。たとえば、次のようなデータがあるとします。

D	クラス	
	2	3

1	1	0
2	0	6

次に、潜在的な分割点は T = 1 です。

$$Gain\,(A,T;S) = 0.5916728$$

$$\frac{\log_2(N-1)}{N} + \frac{\Delta\,(A,T;S)}{N} = 0.6530774$$

0.5916728 < 0.6530774 なので、A と C との間に明快な関係があったとしても、T は分割点として受け入れられません。

アルゴリズムBinaryDiscretization

1. d_i と d_{i+1} が同じクラスに属していないそれぞれ個別の値 $d_i \in D$ に対して、E(A, d_i; S) を計算します。この値のすべてのインスタンスが同じクラスを持っている場合、個別の値は 1 つのクラスに所属します。
2. E(A, T; S) がすべての分割点候補の中で最小になるような、つまり次のような分割点 T を選択します。

   $T = arg\min_{d_i} E\,(A,d_i;S)$

アルゴリズムMDLPCut

1. BinaryDiscretization(A, T; D, S)。
2. Gain(A, T; S) を計算します。
3. $Gain\,(A,T;S) > \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T;S)}{N}$ の場合、次のようになります。

   a) $T_A = T_A \cup T.$

   b) D を D_1 および D_2、S を S_1 および S_2 へ分割します。

   c) MDLPCut(A, T_A; D_1, S_1)。

   d) MDLPCut(A, T_A; D_2, S_2)。ここで $S_1 \subset S$ は A-values ≤ T を含む S のインスタンスのサブセット、そして S_2 = S-S_1 となります。D_1 および D_2 はそれぞれ、S_1 および S_2 内のすべて別々の値のセットです。

   さらに提示されているのは、MDLPCut(A, T_A; D, S) のインタラクティブ版です。インタラクティブ版の実装には、分割される D と S の残りを格納するスタックが必要です。

   最初に、D と S をstack へプッシュします。次に、while (stack≠∅) do を実行します。

1. D と S をstack をポップして取得します。
2. BinaryDiscretization(A, T; D, S)。

3. Gain(A, T; S) を計算します。
4. $Gain(A,T;S) > \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T;S)}{N}$ の場合、次のようになります。

   i) $T_A = T_A \cup T$

   ii) D を D_1 および D_2、S を S_1 および S_2 へ分割します。

   iii) D_1 と S_1 を stack へプッシュします。

   iv) D_2 と S_2 を stack へプッシュします。

   注 : 実際のところ、アルゴリズム内のすべての操作はグローバル行列 M に基づいています。その要素の m_{ij} は、値が $d_i \in D$ で S 内の j 番目のクラスに属するインスタンスの総数です。また、D が昇順でソートされます。したがって、D および S を stack へプッシュする必要はありませんが、D の境界を指す 2 つの整数値のみを stack へプッシュします。

アルゴリズムSimpleMDLP

1. 値 A の昇順で N インスタンスのセット S をソートします。
2. S 内ですべて異なる値のセット D を検索します。
3. $T_A = \varnothing$.
4. MDLPCut(A, T_A; D, S)
5. セット T_A を昇順でソートし、T_A を出力します。

Hybrid MDLP

S 内のそれぞれ異なる値のセット D が大きい場合は、各 $d_i \in D$ に対する E(A, d_i; S) の計算コストも大きくなります。計算コストを削減するために、監視なしの度数が等しいデータ分割方法を使用して、D のサイズを小さくしてサブセット $D_{ef} \in D$ を取得します。次に、MDLPCut(A, T_A; D_S, S) アルゴリズムが適用され、最終的な分割点のセット T_A が取得されます。

アルゴリズムEqualFrequency

連続型属性の A を n 個のビンに分割します。各ビンには N/n 個のインスタンスが含まれます。n はユーザー定義のパラメータで、$1 < n < N$ です。

1. 値 A の昇順で N インスタンスのセット S をソートします。
2. $D_{ef} = \varnothing$。
3. j1。
4. 非経験的なパーセンタイル方法を使用して、$\left(\frac{i \cdot N}{n} \times 100\right)$ 番目のパーセンタイルをさす $d_{p,i}$ を生成します。

5. $D_{ef} = D_{ef} \cup d_{p,i}$; i=i+1
6. i≤n の場合は、ステップ 4 へ進みます。
7. セット D_{ef} 内の重複している値を削除します。

   注 : たとえば、A の単一の値が多く存在する場合、等しい頻度基準が満たされない可能性があります。この場合、分割点は作成されません。

アルゴリズムHybridMDLP

1. D = ∅;
2. EqualFrequency(A, n, D; S).
3. T_A = ∅.
4. MDLPCut(A, T_A; D, S).
5. T_A を出力します。

モデル エントロピー

モデル エントロピーは、クラス変数 C でデータ分割された属性 A の予測精度の測定値です。インスタンスのセット S が指定されると、A が I 個のビンへ C を前提として離散します。ここで、i 番目のビンには値 A_i があります。$S_i \subset S$ を値 A_i の S 内のインスタンスのサブセットとし、モデル エントロピーは次のように定義されます。

$$E_m = \sum_{i=1}^{I} P(A_i) \left(-\sum_{j=1}^{J} P(C_j|A_i) \log_2 P(C_j|A_i) \right)$$

この場合 $P(A_i) = \frac{|S_i|}{|S|}$ and $P(C_j|A_i) = \frac{P(C_j,A_i)}{P(A_i)} = P(C_j,S_i)$ となります。

データがまばらに投入されたビンの結合

時には、非常に少ないケースのビンが作成されることがあります。以下の方針で、このような疑似的な分割点を削除します。

- 与えられた変数に対し、アルゴリズムにより n_{final} 個の分割点が検出されたので、ビン数は n_{final}+1 になります。ビンの i = 2, ..., n_{final} （2 番目に小さい値のビンから 2 番目に大きな値のビンまで） に対して、次のように計算します。

  $$\frac{sizeof(b_i)}{\min(sizeof(b_{i-1}),sizeof(b_{i+1}))}$$

  ここで sizeof(bin) は、ビン内のケースの数を表します。

- この値がユーザー定義の結合しきい値より小さい場合、b_i はデータがまばらに投入されていると見なされ、 b_{i-1} または b_{i+1} と結合されます。 この両方のうち、

クラス情報エントロピーの低いほうが使用されます。 詳細は、 p.266 クラス情報エントロピー を参照してください。

このプロシージャは、ビン全体を単一パスで実行します。

空白の処理

最適データ分析では、空白が対の方式で処理されます。つまり、フィールドの各ペア{データ分割フィールド, 対象フィールド} に対し、両フィールドに有効な値があるすべてのレコードは、分割されるほかのフィールドに存在する可能性があるどの空白も無視して、その特定のデータ分割フィールドを分割するのに使用されます。

参照

Fayyad, U., および K. Irani. 1993. Multi-interval discretization of continuous-value attributes for classification learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo, CA: Morgan Kaufmann, 1022-1027.

Dougherty, J., R. Kohavi, および M. Sahami. 1995. Supervised and unsupervised discretization of continuous features. In: Proceedings of the Twelfth International Conference on Machine Learning, Los Altos, CA: Morgan Kaufmann, 194-202.

Liu, H., F. Hussain, C. L. Tan, および M. Dash. 2002. Discretization:An Enabling Technique. Data Mining and Knowledge Discovery, 6, 393-423.

章
28

QUEST アルゴリズム

QUEST の概要

QUEST は、Quick, Unbiased, Efficient Statistical Tree（迅速で偏りがなく、効率的な統計樹木）の略称です。QUEST は比較的新しい 2 進木成長アルゴリズム(Loh および Shih, 1997)で、分岐フィールドの選択と分岐点の選択を個別に処理します。QUEST の単変量の分岐ではほぼ偏りのないフィールド選択が行われます。つまり、対象フィールドに関する情報の有益性がすべての予測フィールドで等しい場合、選択される確率はどの予測フィールドでも等しくなります。

QUEST には C&RT の数多くの利点がありますが、C&RT と同様に、ツリー（木）が扱いにくくなる可能性があります。コストと複雑さによる自動剪定（剪定 p.279 参照）を QUEST ツリーに適用して、そのサイズを縮小することができます。QUEST では、代理変数の分岐によって欠損値に対処します。 詳細は、 p.276 空白の処理を参照してください。

一次計算

ここでは、モデル構築時に直接行われる計算を説明していきます。

度数重みフィールド

度数フィールドは、各レコードが示す総観測数を表しています。レコードが 1 つ以上を表す集計データの分析に役立ちます。度数フィールドの値の合計は、常にサンプル中の総観測数と等しくなければなりません。度数フィールドを使っても、ケースごとのデータを使っても、出力と統計量は同じであることに注意してください。以下のテーブルに、予測フィールドの［性別］と［雇用］、および対象フィールド［ 回答］を持つ仮説の例を示します。たとえば、この例の度数フィールドからは、職のある 10 人の男性が対象の質問に対して「はい」と回答しており、職を持たない 19 人の女性が「いいえ」と回答していることがわかります。

テーブル 28-1
度数フィールドのあるデータ セット

性別	雇用	回答	度数
M	Y	Y	10
M	Y	N	17
M	N	Y	12
M	N	N	21
F	Y	Y	11
F	Y	N	15
F	N	Y	15
F	N	N	19

このケースで度数フィールドを使用すると、8 レコードのテーブルを処理することになります。一方ケースごとのデータを使用すると、120 レコードを処理しなければなりません。

QUEST は、ケースの重みをサポートしていません。

モデル パラメータ

QUESTは、フィールドの選択と分岐点の選択を個別に処理します。使用すべきαレベルは、QUESTの［エキスパート］オプションで指定できます。デフォルト値は、$\alpha_{nominal}$ = 0.05 です。

フィールドの選択

1. 各予測フィールド X について、X がシンボル値（カテゴリ）フィールドであれば、それが名義フィールドであっても序数フィールドであっても、X とその依存フィールドとの間の独立性に関する Pearson カイ 2 乗検定の p 値を算出します。X が尺度レベル（連続）であれば、F 検定を使って p 値を算出します。

2. 最小の p 値を、事前に指定された Bonferroni の調整されたαレベル α_B と比較します。
   - 最小の p 値が α_B よりも小さければ、対応する予測フィールドを選択してノードを分割します。ステップ 3 に進みます。
   - 最小の p 値が α_B よりも小さく「ない」場合には、尺度レベル（連続）である各 X について、等しくない分散についての Levene 検定によって p 値を算出します(つまり、X が等しくない分散をもつかどうかを対象フィールドのさまざまなレベルで検定します)。
   - この Levene 検定による最小の p 値を、Bonferroni の調整された新たなαレベル α_L と比較します。
   - この p 値が α_L よりも小さい場合には、対応する予測フィールドを Levene 検定による最小の p 値で選択し、ノードを分岐します。
   - p 値が α_L より大きい場合、ノードは分割されません。

分岐点の選択：尺度レベルの予測フィールド

1. Y のカテゴリが 2 つしかない場合には、次のステップにスキップしてください。それ以外の場合には、次のようにして Y のカテゴリを 2 つのスーパークラスにグループ化します。
   - Y の各カテゴリについて、X の平均値を算出します。
   - すべての平均値が同一であれば、度数の重みが最も大きいカテゴリをスーパークラスの 1 つとして選択します。もう 1 つのスーパークラスは、その他のすべてのカテゴリを結合して形成します(すべての平均値が同一で、度数の重みが最も大きいカテゴリが複数ある場合には、最もインデックスの小さいカテゴ

リをスーパークラスの 1 つとして選択し、それ以外のカテゴリを結合して別のカテゴリを形成します)。

- すべての平均値が同一というわけでない場合には、それらの平均値に対して 2 平均クラスタリング アルゴリズムを適用することにより、Y の 2 つのスーパークラスを取得します。初期クラスタ中心は、2 つの両極端のクラス平均に設定します(これは、K = 2 とする K-Means クラスタリングの特殊ケースです詳細は、 20 章 p.187 概要 を参照してください。)。

2. 分岐点を決定するには、二次判別分析 (Quadratic Discriminant Analysis : QDA) を適用します。QDA では一般に 2 つの分割点が作成されます。第 1 のスーパークラスの標本平均に近い方を選択してください。

分岐点の選択 : シンボル (カテゴリ) 予測フィールド

QUEST は、予測フィールドのカテゴリに判別座標を割り当てることにより、最初にシンボル値フィールドを連続フィールド ξ に変換します。そして派生フィールド ξ は、別の連続予測フィールドであったかのように、上述のようにして分割されます。

カイ 2 乗検定

Pearson のカイ 2 乗統計は、次のように算出されます。

$$X^2 = \sum_{j=1}^{J}\sum_{i=1}^{I}\frac{(n_{ij} - \hat{m}_{ij})^2}{\hat{m}_{ij}}$$

ここで、 $n_{ij} = \sum_n f_n I(x_n = i \wedge y_n = j)$ は観測セル度数を表し、(x_n = i, y_n = j) に対し、下記の独立モデルから $\hat{m}_{ij}$ は期待セル度数を表しています。対応する p 値は、$p = \Pr\left(\chi_d^2 > X^2\right)$ のように計算されます。ここで、χ_d^2 は、自由度 d = (J − 1)(I − 1) のカイ 2 乗分布に従います。

カイ 2 乗テストの期待頻度数

ケースの重みのないモデルの場合、期待度数は次のように計算されます。

$$\hat{m}_{ij} = \frac{n_{i.}n_{.j}}{n_{..}}$$

ここで、

$$n_{i.} = \sum_{j=1}^{J} n_{ij},\ n_{.j} = \sum_{i=1}^{I} n_{ij},\ n_{..} = \sum_{j=1}^{J}\sum_{i=1}^{I} n_{ij}.$$

F 検定

ノード t について、対象フィールド Y のクラスが J_t 個ある場合を考えます。連続予測フィールド X の F 統計量は、次のように計算されます。

$$F_X = \frac{\sum_{j=1}^{J_t} N_{f,j}(t)\left(\overline{x}^{(j)}(t) - \overline{x}(t)\right)^2/(J_t - 1)}{\sum_{i\in t} f_i\left(x_i - \overline{x}^{(y_n)}(t)\right)^2/\left(N_f(t) - J_t\right)}$$

ここで、

$$\overline{x}^{(j)}(t) = \frac{\sum_{i\in t} f_n x_n I(y_n = j)}{N_{f,j}(t)}, \overline{x}(t) = \frac{\sum_{i\in t} f_n x_n}{N_f(t)}$$

対応する p 値は、次の式により与えられます。

$$p_X = \Pr\left(F\left(J_t - 1, N_f(t) - J_t\right) > F_X\right)$$

ここで $F(J_t - 1, N_f(t) - J_t)$ は、自由度が $J_t - 1$ および $N_f(t) - J_t$ の F 分布に従います。

Levene 検定

連続予測フィールド X について、$z_n = \left|x_n - \overline{x}^{(y_n)}(t)\right|$ を計算します。ここで、 は、対象値 y_n を持つノード t 中のレコードについての X の平均です。予測フィールド X についての Levene F 統計量は、z_n についての ANOVA F 統計量です。

Bonferroni の調整

調整された α レベル α_B は、nominal a 値を比較の候補の数で除算したものになります。

QUEST では、初期予測フィールド選択のための Bonferroni の調整された α レベル α_B は、次のようになります。

$$\alpha_B = \frac{\alpha_{nominal}}{m}$$

ここで m は、モデル中の予測フィールドの数を表します。

Levene 検定では、Bonferroni の調整された α レベル α_L は、次のようになります。

$$\alpha_L = \frac{\alpha_{nominal}}{m + m_c}$$

ここで m_c は、連続予測フィールドの数を表します。

判別分析候補

値が {$b_1,...,b_I$} であるカテゴリ予測フィールド X について、QUEST は、連続変数 ξ からのスコア値を X の各カテゴリに割り当てます。割り当てられるスコアは、ξ のクラス内平方和に対するクラス間平方和の比率が対象フィールドクラスについて最大になるように選択されます。

各レコードについて、X をダミーフィールドのベクトル $\mathbf{g} = (g_1,...,g_I)'$ に変換します。ここで

$$g_i = \begin{cases} 1 & x = b_i \\ 0 & \text{otherwise} \end{cases}$$

ν の全体およびクラス j 平均を計算します。

$$\overline{\mathbf{g}} = \frac{\sum_n f_n \mathbf{g}_n}{N_f}, \overline{\mathbf{g}}^{(j)} = \frac{\sum_n f_n \mathbf{g}_n I(y_n = j)}{N_{f,j}}$$

ここで、f_n はレコード n の度数の重み、g_n はレコード n のダミーベクトル、N_f は学習データの度数の重みの総計、$N_{f,j}$ はカテゴリ j を持つレコードの度数の重みの合計を表します。

次の $I \times I$ 行列を計算します。

$$\mathbf{B} = \sum_{j=1}^{J} N_{f,j} \left(\overline{\mathbf{g}}^{(j)} - \overline{\mathbf{g}}\right)\left(\overline{\mathbf{g}}^{(j)} - \overline{\mathbf{g}}\right)'$$

$$\mathbf{T} = \sum_n f_n (\mathbf{g}_n - \overline{\mathbf{g}})(\mathbf{g}_n - \overline{\mathbf{g}})'$$

T に対して特異値分解を実行して $\mathbf{T} = \mathbf{QDQ}'$ を取得します。ここで、Q は $I \times I$ 直交行列、D = diag($d_1,...,d_I$)（ただし $d_1 \geq ... \geq d_I \geq 0$）です。$\mathbf{D}^{-\frac{1}{2}} = diag\,(d_1^*,...,d_I^*)$ とします。ここで、$d_i^* = d_i^{-\frac{1}{2}}$ if $d_i > 0$、それ以外の場合は 0 です。$\mathbf{D}^{-\frac{1}{2}}\mathbf{Q}'\mathbf{BQD}^{-\frac{1}{2}}$ に対して特異値分解を実行して、その最大固有値に関連づけられた固有ベクトル a を取得します。

g の最大判別座標は、次の射影になります。

$$\xi = \mathbf{a}'\mathbf{D}^{-\frac{1}{2}}\mathbf{Q}'\mathbf{g}$$

QDA（二次判別分析）

連続予測フィールドの分割点を決定するには、まず対象フィールド Y のカテゴリをグループ化して、前述したように 2 つのスーパークラス A および B を形成します。

$\min\left(s_A^2, s_B^2\right) = 0$ であれば、分散が小さい方が先になるように 2 つのスーパークラスを並べ替え、その分散を $s_1^2 \leq s_2^2$、平均値を $\overline{x}_1, \overline{x}_2$ として表します。ε を非常に小さい正の数とします。たとえば、ε = 10^{-12} といった具合です。$\overline{x}_1$ および ε に基づいて、次のようにカットポイント d を設定します。

$$d = \begin{cases} \overline{x}_1(1+\epsilon) & \text{if } \overline{x}_1 < \overline{x}_2 \\ \overline{x}_1(1-\epsilon) & \text{otherwise} \end{cases}$$

空白の処理

ツリー モデルの構築時に、対象フィールドに欠損値があるレコードは無視されます。

代理変数の分岐は、予測フィールドの空白値を処理するために用いられます。ある特定のノードに対して、分岐に使用する最良の予測フィールドに空白値または欠損値がある場合、そのノードに関連する予測フィールドと類似した分岐を生成する他のフィールドが、予測フィールドの代理変数として使用され、その値はいずれかの子ノードにレコードを割り当てるために用いられます。

たとえば、ノード t における最良の分岐点 s* を定義する予測フィールド X* を考えてみましょう。代理変数の分岐処理では、分岐がノード t における s* に最も類似するように、他の予測フィールド X をベースに他の分岐点 s を探します。新規レコードが予測対象で、ノード t における X* 上に欠損値がある場合、代わりに代理変数の分岐 s が適用されます(このレコードが、X 上にも欠損値を持つ場合を除きます。このような状況下では、次に最良の代理変数が、指定された代理変数の数の上限まで順次使用されます。)

処理速度およびメモリー保持の観点から、ツリー中の各分岐に対して、一定数の代理変数しか使用されません。レコードの分岐フィールドおよびすべての代理変数フィールドに欠損値がある場合、それは重みが大きい確率を持つ子ノードに割り当てられ、次のように計算されます。

$$\frac{N_{f,j}(t)}{N_f(t)}$$

ここで $N_{f,j}(t)$ はノード t のカテゴリ j 中にあるレコードの度数の重みの合計を、$N_f(t)$ は、ノード t 中のすべてのレコードの度数の重みの合計を表します。

均等化またはユーザーが指定した事前確率を使って構築されたモデルの場合は、計算に事前確率が取り入れられます。

$$\frac{\pi(j)}{p_f(t)} \times \frac{N_{f,j}(t)}{N_f(t)}$$

ここで $\pi(j)$ はカテゴリ j の事前確率を、$p_f(t)$ はノードに割り当てられたレコードの重み付けられた確率を表します。

$$p_f(t) = \sum_j \frac{\pi(j) N_{f,j}(t)}{N_{f,j}}$$

ここで $N_{f,j}(t)$ はカテゴリ j に所属するノード t 中の度数の重みの合計（度数の重みが定義されていない場合はレコード数）を、$N_{f,j}$ は学習サンプル全体のカテゴリに所属するレコードの度数の重みの合計を表します。

予測関連度

$\hbar_{X^*\cap X}$（または $\hbar_{X^*\cap X}(t)$）を、X* および X の両方が欠損値でない学習ケース（ノード t の学習ケース）であるとします。$p(s^* \approx s_X|t)$ を、$\hbar_{X^*\cap X}(t)$ のケースを s^* および s_X によって同一の子に送る確率とし、$\tilde{s}_X$ を最大確率 $p(s^* \approx \tilde{s}_X|t) = \max_{s_X}(p(s^* \approx s_X|t))$ による分割とします。

ノード t の s* および $\tilde{s}_X$ 間の $\lambda(s^* \approx \tilde{s}_X|t)$ 予測関連度は、次のようになります。

$$\lambda(s^* \approx \tilde{s}_X|t) = \frac{\min(p_L, p_R) - (1 - p(s^* \approx \tilde{s}_X|t))}{\min(p_L, p_R)}$$

ここで、p_L（または p_R）は相対確率で、ノード t の最適な確率 s* at は、X* の値が欠損値でないケースを左側（または右側）の子ノードに送ります。そして、ここで

$$p(s^* \approx s_X|t) = \begin{cases} \sum_j \dfrac{\pi(j) N_{w,j}(s^* \approx s_X, t)}{N_{w,j}(X^* \cap X)} & \text{if} Y \text{is categorical} \\ \dfrac{N_w(s^* \approx s_X, t)}{N_w(X^* \cap X)} & \text{if} Y \text{is continuous} \end{cases}$$

を

$$N_w(X^* \cap X) = \sum_{n \in \hbar_{X^*\cap X}} w_n f_n, \quad N_w(X^* \cap X, t) = \sum_{n \in \hbar_{X^*\cap X}(t)} w_n f_n$$

$$N_w(s^* \approx s_X, t) = \sum_{n \in \hbar_{X^*\cap X}(t)} w_n f_n I(n : s^* \approx s_X)$$

$$N_{w,j}(X^* \cap X) = \sum_{n \in \hbar_{X^*\cap X}} w_n f_n I(y_n = j), \quad N_{w,j}(X^* \cap X) = \sum_{n \in \hbar_{X^*\cap X}(t)} w_n f_n I(y_n = j)$$

$$N_{w,j}(s^* \approx s_X, t) = \sum_{n \in \hbar_{X^*\cap X}(t)} w_n f_n I(y_n = j) I(n : s^* \approx s_X)$$

そして $I(n : s^* \approx s_X)$ は分割 s* および s_X がケース n を同一の子に送る場合に値 1 をとり、そうでない場合は 0 をとる指標関数となります。

オプションの効果

停止規則

停止基準は、ツリー中のノードの分岐をいつ停止するかを判断するために用いられます。ツリーの成長は、ツリー中の各枝葉ノードが最低 1 つの停止基準を満たすまで継続します。ノードの分岐を停止する条件を次に示します。

- ノードが純粋の場合（すべてのレコードの対象フィールドが同じ値を持つ）
- モデルが使用するすべての予測フィールドに対して、ノード中のすべてのレコードが同じ値を持つ場合
- 現在のノードのツリーの深さ（現在のノードを定義する帰納的ノード分岐数）が最大ツリー深さ（デフォルト値またはユーザー指定）の場合
- ノード中のレコード数が最小親ノード サイズ（デフォルト値またはユーザー指定）未満の場合
- ノードの最良の分岐点の結果となる任意の子ノード中のレコード数が、最小子ノード サイズ（デフォルト値またはユーザー指定）未満の場合

プロフィット

プロフィットは、（シンボル値）対象フィールドのカテゴリに関連する数値で、セグメントに関連するゲインまたはロスを推定するために用いられます。プロフィットは、対象フィールドの各値の相対値を定義しています。値は、ゲインを算出するために使われますが、ツリーの成長には使用されません。

ツリー中の各ノードのプロフィットは、次のように計算されます。

$$\sum_j f_j(t) P_j$$

ここで j は対象フィールドのカテゴリ、$f_j(t)$ は対象フィールドに対するカテゴリ j と、ノード t 中のすべてのレコードの度数フィールド値の合計、そして P_j はカテゴリ j に対するユーザーが定義したプロフィットの値を表します。

事前確率

事前確率は、対象フィールドのカテゴリの誤分類率に影響する数値です。事前確率は、分析前に対象フィールドの各カテゴリに所属すると推定されるレコードの比率を示します。この値はツリーの成長とリスク推定の両方に使用されます。

事前確率を取得するには、3 種類の方法があります。

経験的事前確率

デフォルトでは、事前確率は学習データに基づいて算出されます。各対象カテゴリに割り当てられる事前確率は、そのカテゴリに所属する学習データ中の重み付けられたレコードの比率になります。

$$\pi(j) = \frac{N_{w,j}}{N_w}$$

ツリー成長およびクラス割り当てにおいては、N はケースの重みと度数の重みの両方を考慮します（定義されている場合）。リスク推定においては、度数の重みだけが経験的事前確率の計算に含まれます。

等事前確率

等事前確率（すべてのクラスで同じ）では、それぞれの J カテゴリの事前確率に同じ値が選択されます。

$$\pi(j) = \frac{1}{J}$$

ユーザー定義事前確率

ユーザー定義（設定）の事前確率を使用する場合、事前確率を含む計算に指定された値が使用されます。事前確率に指定された値は、確率の制約に従っている必要があります。全カテゴリの事前確率の合計は、 1.0 である必要があります。ユーザーが定義した事前確率がこの条件を満たしていない場合、次の式を使って事前確率が調整されます。この事前確率は、元の事前確率の比率を維持しながら、制約に準拠するように調整されます。

$$\pi'(j) = \frac{\pi(j)}{\sum_J \pi(j)}$$

ここで π’（j）はカテゴリ j の調整された事前確率を、π（j）はユーザーが定義したカテゴリ j に対する元の事前確率を表します。

コスト

誤分類コストを指定すると、変更された事前確率を使用することによって、それらが分岐計算に取り入れられます。変更された事前確率は、次のように定義されます。

$$\pi'(t) = \frac{C(j)\pi(j)}{\sum_J C(j)\pi(j)}$$

この場合、$C(j) = \sum_i C(i|j)$ です。

後述するように、誤分類コストはリスク推定と予測フィールド（それぞれ p. 281 と p. 282 ）にも影響します。

剪定

剪定は、完全に成長したツリーの調査、およびツリーの精度にさほど寄与しない下位レベルの分岐の削除に関する処理です。ツリーの剪定では、可能な最大ツリーの誤分類リスクと比べてさほど誤分類リスクが大きくならない、最小のツリーが作成されます。このために、より複雑なツリーを持つためのコストが、他のレベルのノード（枝葉）を持つためのゲインを超える場合、ツリーの枝葉は削除されます。

誤分類リスクとツリーの複雑さの両方を最低限に抑えるために、これらを測定するインデックスが使用されます。このコストと複雑さの測度は、次のように定義されます。

$$R_\alpha(T) = R(T) + \alpha\left|\tilde{T}\right|$$

R(T) はツリー T の誤分類リスクを、$\left|\tilde{T}\right|$ はツリー T のターミナル ノード数を表します。項目 α は、「ターミナル ノードごと」のツリーの複雑性コストを表します。(α の値は、剪定中のアルゴリズムによって算出されることに注意してください。)

生成するツリーには最大サイズ (T_{max}) があり、各ターミナル ノードには 1 つのレコードしか含まれません。複雑さのコストがない場合 (α = 0)、すべてのレコードが完全に予測されるため、最大のツリーのリスクが最も低くなります。そのため、α の値が大きくなるほど、T(α) 中のターミナル ノード数は少なくなります。ここで T(α) は、与えられた α に対して、複雑さのコストが最も低いツリーを表します。α が 0 から増加するにつれて、サブツリーの有限数列 (T_1, T_2, T_3) が生成されます。それぞれ、順次ターミナル ノード数が少なくなっていきます。コストと複雑さによる剪定は、最も弱い分岐を削除することにより行われます。

任意の単一ノード {t} および {t} の子枝葉 T_t のコストの複雑さを表す式を次に示します。

$$R_\alpha(\{t\}) = R(t) + \alpha$$

$$R_\alpha(T_t) = R(T_t) + \alpha\left|\tilde{T}_t\right|$$

$R_\alpha(T_t)$ が $R_\alpha(\{t\})$ 未満の場合、枝葉 T_t は単一ノード {t} よりもコストの複雑さが小さくなります。

ツリー成長の過程では、(α = 0) に対する $R_\alpha(\{t\}) \geq R_\alpha(T_t)$ が保証されます。α が 0 から増加するにつれて、$R_\alpha(\{t\})$ と $R_\alpha(T_t)$ の両方が線型的に成長していきます。この場合、後者の成長率が速くなります。最終的には、すべての $\alpha > \alpha'$ に対して $R_\alpha(\{t\}) < R_\alpha(T_t)$ となるような閾値 α' に達します。このことは、α が α' よりも大きく成長した場合、{t} 下の子枝葉 T_t を削除すると、ツリーのコストの複雑さを減らせることを意味しています。閾値は単純な計算により決められます。この最初の不等式 $R_\alpha(\{t\}) \geq R_\alpha(T_t)$ を解いて、不等式が保持する α の最大値を探すことができます。これは、g(t) として表すこともできます。最終的には次のようになります。

$$\alpha \leq g(t) = \frac{R(t) - R(T_t)}{\left|\tilde{T}_t\right| - 1}$$

g(t) の最も小さい値を持つノードとして、ツリー T 中の最も弱いリンク (t) を定義することができます。

$$g(\bar{t}) = \min_{t \in T} g(t)$$

このため、α が増加するにつれて、$\bar{t}$ が $R_\alpha(\{t\}) = R_\alpha(T_t)$ である最初のノードになります。この時点で $\{\bar{t}\}$ が $T_{\bar{t}}$ に対してより好ましくなり、子枝葉が剪定されます。

このようなバックグラウンド情報を判断した後、剪定アルゴリズムにより次の処理が行われます。

- $\alpha_1 = 0$ を設定し、完全に成長したツリー $T_1 = T(0)$ から開始します。
- 枝葉が剪定されるまで、α の値を増やします。ツリーから枝葉を剪定し、剪定されたツリーのリスク推定を計算します。
- ルート ノードだけが残るまで、直前のステップを繰り返し、一連のツリー T_1、T_2、... を残します。T_k とします。
- 標準エラー ルール オプションが選択されている場合、最小のツリー T_{opt} を選択します。

$$R(T_{opt}) \leq \min_k R(T_k) + m \times SE(R(T))$$

- 標準エラー ルール オプションが選択されていない場合は、リスク推定が最も小さいツリー R(T) を選択します。

二次計算

二次計算はモデルの構築に直接には関係していませんが、モデルとそのパフォーマンスに関する情報を得ることができます。

リスク推定

リスク推定は、ツリーの特定ノードおよびツリー全体の予測フィールド中の誤差リスクを表しています。

シンボル値フィールドに対するリスク推定

分類ツリー（シンボル値対象フィールドを持つ）の場合、ノード t のリスク推定値 r(t) は次のように算出されます。

$$r(t) = \frac{1}{N_f} \sum_j N_{f,j}(t) C(j^*(t)|j)$$

ここで C(j*(t)|j) は対象値 j を j*(t) として持つレコードを分類する際の誤分類コストを、$N_{f,j}(t)$ はカテゴリ j の ノード t 中のレコードの度数の重みの合計（または度数の重みが定義されていない場合はレコードの数）、N_f は学習データ中のすべてのレコードの度数の重みの合計を表します。

ユーザーが指定した事前確率を使って構築されたモデルの場合は、リスク推定値は次のように計算されます。

$$\sum_j \frac{\pi(j) N_{f,j}(t)}{N_{f,j}} C(j^*(t)|j)$$

ゲインの要約

ゲインの要約は、ツリーのターミナル ノードの記述統計量を提供しています。
対象フィールドが連続型（スケール）の場合、ゲインの要約は各ターミナル ノードの対象値の重み付けられた平均を表します。

$$g(t) = \sum_{i \in t} w_i f_i x_i$$

対象フィールドがシンボル値（カテゴリ）の場合は、選択した対象カテゴリ中のレコードの重み付けられた割合を表します。

$$g\left(t, j\right) = \frac{\sum_{i \in t} f_i x_i(j)}{\sum_{i \in t} f_i}$$

ここでレコード x_i が対象カテゴリ j 中にある場合 $x_i(j) = 1$ に、それ以外の場合は 0 になります。ツリーのプロフィットが定義されている場合、各ターミナル ノードのプロフィット値の平均がゲインになります。

$$g(t) = \sum_{i \in t} f_i P(x_i)$$

ここで $P(x_i)$ はレコード x_i 中に観測される対象値に割り当てられたプロフィット値を表します。

生成されたモデル/スコアリング

QUEST 生成モデルにより行われる計算は、後述します。

予測値

新しいレコードは、ツリーのターミナル ノードへのツリー分割にしたがってスコアリングされます。各ターミナル ノードには、それに対応する予測フィールドがあります。予測フィールドは、次のように決定されます。

シンボル値対象フィールドを持つツリーの場合、各ターミナル ノードの予測カテゴリは、ノードの重み付けられたコストが最も低いカテゴリになります。この重み付けられたコストは、次のように算出されます。

$$\min_i \sum_j C(i|j) p(j|t)$$

ここで $C(i|j)$ はレコードが実際にはカテゴリ j にある場合に、レコードをカテゴリ i に分類する際の、ユーザーが定義した誤分類コストを、$p(j|t)$ はカテゴリ j がノード t にある場合の、カテゴリ中のレコードの重み付けられた条件確率を表し、次のように定義されます。

$$p(j|t) = \frac{p(j,t)}{\sum_j p(j,t)}, p(j,t) = \pi(j)\frac{N_{w,j}(t)}{N_{w,j}}$$

ここで π(j) はカテゴリ j，N_w の事前確率、$_j$(t) はカテゴリ j を持つノード t 中の重み付けられたレコード数（度数またはケースの重みが指定されていない場合はレコード数）を表します。

$$N_{w,j}(t) = \sum_{i \in t} w_i f_i j(i)$$

$N_{w,j}$ は、カテゴリ j（ノードは任意）中の重み付けられたレコード数を表します。

$$N_{w,j} = \sum_{i \in T} w_i f_i j(i)$$

確信度

スコア付きレコードの確信度とは、予測カテゴリに所属するターミナル ノードに割り当てられたスコア付きレコードに対する、学習データ内にある重み付けされたレコードの比率です。ラプラス補正による次のような変更も加えられます。

$$\frac{N_{f,j}(t) + 1}{N_f(t) + k}$$

空白の処理

新規レコードの分類において、空白はツリーの成長時のように処理されます。可能な場合には代理変数が使用され、必要に応じて重み付けられた確率に基づいて分岐が行われます。 詳細は、 p. 276 空白の処理 を参照してください。

章

29

線型回帰アルゴリズム

概要

このアルゴリズムは、一般最小 2 乗重回帰法を実施し、変数の投入と削除に 4 種類の方法を使用します(Neter, Wasserman, および Kutner, 1990)。

一次計算

表記

この章では特に明記しない限り、次の表記を使用します。

y_i	分散 $\frac{\sigma^2}{g_i}$ のレコード i の出力フィールド
c_i	レコード i のケースの重み、IBM® SPSS® Modeler では $c_i \equiv 1$
g_i	レコード i の回帰の重み（回帰係数)、回帰の重みが指定されていない場合は $g_i = 1$
l	重複しないレコード数
w_i に置き換えます。	$c_i \cdot g_i$
W	レコードに渡る重みの合計、$\sum_{i=1}^{l} w_i$
p	入力フィールド数
C	ケースの重みの合計、$\sum_{i=1}^{l} c_i$
x_{ki}	レコード i の k 番目の入力フィールドの値
$\overline{X}_k$	k 番目の入力フィールドの標本平均、$\frac{\sum_{i=1}^{l} w_i x_{ki}}{W}$
$\overline{Y}$	出力フィールドの標本平均、$\frac{\sum_{i=1}^{l} w_i y_i}{W}$
S_{kj}	入力フィールド X_k および X_j の標本共分散
S_{yy}	出力フィールド Y の標本分散
S_{ky}	X_k および Y の標本共分散
p^*	モデル中の係数の数。切片が含まれていない場合 $p^* = p$、それ以外の場合は $p^* = p + 1$
$\mathbf{R}$	$X_1 \ldots X_p$ および Y の標本相関行列

モデル パラメータ

要約統計量 $\overline{X}_i$ および共分散 S_{ij} は暫定平均アルゴリズムを使って計算され、各レコードが読み込まれるにつれて値が更新されます。

$$\overline{X}_{i(k)} = \overline{X}_{i(k-1)} + \left(x_{ik} - \overline{X}_{i(k-1)}\right)\frac{w_k}{W_k}$$

そして

$$S_{ij} = \frac{C_{ij}}{C-1}$$

ここで、切片が含まれている場合、

$$C_{ij(k)} = C_{ij(k-1)} + \left(x_{ik} - \overline{X}_{i(k-1)}\right)\left(x_{jk} - \overline{X}_{j(k-1)}\right)\left(w_k - \frac{w_k^2}{W_k}\right)$$

切片が含まれていない場合は、

$$C_{ij(k)} = C_{ij(k-1)} + w_k x_{ik} x_{jk}$$

ここで W_k はレコード k までの累積重みを、$\overline{X}_{i(k)}$ はレコード k までの $\overline{X}_i$ の推定を表します。

次の形式の回帰モデルの場合

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_p X_{pi} + e_i$$

β の最小 2 乗推定値 **b** および関連する回帰統計量を計算するために、スイープ操作が使用されます(Dempster, 1969)。スイープ操作は相関行列 **R** から始まります。

$$\mathbf{R} = \begin{bmatrix} r_{11} & \cdots & r_{1p} & r_{1y} \\ r_{21} & \cdots & r_{2p} & r_{2y} \\ . & \cdots & . & . \\ r_{y1} & \cdots & r_{yp} & r_{yy} \end{bmatrix}$$

ここで、

$$r_{kj} = \frac{S_{kj}}{\sqrt{S_{kk}S_{jj}}}$$

そして

$$r_{yk} = r_{ky} = \frac{S_{ky}}{\sqrt{S_{kk}S_{yy}}}$$

$\tilde{\mathbf{R}}$ を k 番目の行および **R** の列に対してスイープ操作することにより生成される新規行列とします。$\tilde{\mathbf{R}}$ の要素は次のようになります。

$$\tilde{r}_{kk} = \frac{1}{r_{kk}}$$

$$\tilde{r}_{ik} = \frac{r_{ik}}{r_{kk}}, i \neq k$$

$$\tilde{r}_{kj} = \frac{r_{kj}}{r_{kk}}, j \neq k$$

そして

$$\tilde{r}_{ij} = \frac{r_{ij}r_{kk} - r_{ik}r_{kj}}{r_{kk}}, i \neq k, j \neq k$$

上記のスイープ操作が繰り返し以下の $\mathbf{R}_{11}$ の各行に対して適用されている場合、

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{pmatrix}$$

$\mathbf{R}_{11}$ に現在のステップにおける式中の入力フィールドが含まれているとすると、結果は次のようになります。

$$\tilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R}_{11}^{-1} & -\mathbf{R}_{11}^{-1}\mathbf{R}_{12} \\ \mathbf{R}_{21}\mathbf{R}_{11}^{-1} & \mathbf{R}_{22} - \mathbf{R}_{21}\mathbf{R}_{11}^{-1}\mathbf{R}_{12} \end{pmatrix}$$

ここで

$$\mathbf{R}_{21}\mathbf{R}_{11}^{-1}$$

の最後の行には標準化係数（ベータとも呼ばれます）が含まれ、

$$\mathbf{R}_{22} - \mathbf{R}_{21}\mathbf{R}_{11}^{-1}\mathbf{R}_{12}$$

を、すでに式中にある変数を制御する、式中にない変数の偏相関を取得するために使用できます。このルーチンはその逆ルーチンでもあり、入力フィールドの削除を行うためには、入力フィールドを投入する場合と完全に同じ操作が行われることに注意してください。

非標準化係数推定 $b_1 ... b_p$ は、次のように計算されます。

$$b_k = \frac{r_{yk}\sqrt{S_{yy}}}{\sqrt{S_{kk}}}$$

また、切片 b_0 がモデルに含まれている場合は、次のように切片が計算されます。

$$b_0 = \overline{y} - \sum_{k=1}^{p} b_k \overline{X}_k$$

自動フィールド選択

r_{ij} が X_i および X_j に関連する現在のスイープされた行列中の要素とします。変数は 1 回に 1 つずつ投入または削除されます。X_k が投入に適すると判断されるのは、それが現在モデル中にない入力フィールドで、次の条件を満たしている場合です。

$$r_{kk} \geq t$$

そして

$$\left(r_{jj} - \frac{r_{jk}r_{kj}}{r_{kk}}\right)t \leq 1$$

ここで t は許容度で、デフォルトは 0.0001 になります。

変数の投入によりモデル中の変数の許容度が許容できないレベルまで減少することがないようにするために、上記の 2 番目の条件が適用されます。

X_k の F-to-enter 値は次のように計算されます。

$$F - to - enter_k = \frac{(C - p^* - 1)V_k}{r_{yy} - V_k}$$

この場合 1 および $C - p^* - 1$ の自由度が用いられます。ここで p^* は現在モデル中にある係数の数を表します。また次のようになります。

$$V_k = \frac{r_{yk}r_{ky}}{r_{kk}}$$

X_k の F-to-remove 値は次のように計算されます。

$$F - to - remove_k = \frac{(C - p^*)\,|V_k|}{r_{yy}}$$

この場合 1 および $C - p^*$ の自由度が用いられます。

変数の投入および削除方法

変数の投入と削除には、4 種類の方法を利用することができます。選択処理は、投入または削除に適した独立変数がなくなるまで繰り返されます。次に、これらの 4 種類の方法のアルゴリズムを説明していきます。

Enter

選択された入力フィールドがすべてモデルに投入されます。フィールド選択は適用されません。

Stepwise

現在モデルに投入されている独立変数がある場合、$F - to - remove_k$ が最小になるような X_k を選択します。X_k は、$F - to - remove_k < F_{out}$（デフォルト = 2.71）の場合、また確率基準が使用されている時は $P(F - to - remove_k) > P_{out}$（デフォルト = 0.1）の場合に削除されます。不等式が成立しない場合、モデルから変数は削除されません。

現在モデルに投入されている独立変数がない場合、または削除する投入された変数がない場合は、$F - to - enter_k$ が最大になるように X_k を選択します。$F - to - enter_k > F_{in}$（デフォルト = 3.84）または、$P(F - to - enter_k) < P_{in}$（デフォルト = 0.05）の場合、$X_k$ が投入されます。不等式が成立しない場合は、変数は投入されません。

各ステップにおいて、すべての適した変数が削除および投入対象とみなされます。

変数増加法

この処理は、ステップワイズ法の入力フェーズです。

変数減少法

この処理は、モデル中のすべての入力フィールドから開始して、ステップワイズ法の削除フェーズを適用していきます。

空白の処理

デフォルトでは、入力または出力フィールドに欠損値があるケースは、すべての結果演算の基となる相関行列の演算からは削除されます。[**完全なレコードのみ使用**] オプションの選択を解除している場合は、他のフィールドに欠損値があるかどうかに関わらず、相関に関連する 2 つのフィールドに完全なデータを持つレコードに基づいて、相関行列 **R** 中の各相関が計算されます。一部のデータ セットでは、この方法により非正値定 **R** 行列になる可能性があるため、モデルを推定することはできません。

二次計算

モデル要約統計量

複数の相関係数 R は次のように計算されます。

$$R = \sqrt{1 - r_{yy}}$$

入力フィールドの結果となる出力フィールド中の分散の割合である R2 乗は、次のように計算されます。

$$R^2 = 1 - r_{yy}$$

学習データのサイズに相対的なモデルの複雑さを考慮に入れた調整済み R2 乗は、次のように計算されます。

$$R^2_{adj} = R^2 - \frac{(1 - R^2)p}{C - p^*}$$

フィールド統計量および他の計算

回帰（式）ノードの詳細出力に表示されている統計量は、IBM® SPSS® Statistics の REGRESSION（回帰）プロシージャと同じ方法で計算されます。詳細は、『SPSS Statistics Regression algorithm』を参照してください。このドキュメントは、http://www.ibm.com/support から入手できます。

生成されたモデル／スコアリング

予測値

新規レコードの予測フィールドは次のように計算されます。

$$\hat{y} = b_0 + \sum_{i=1}^{p} b_i X_i$$

空白の処理

最終モデルにおいて、入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールド $null$ が割り当てられます。

章
30

シーケンス アルゴリズム

シーケンス アルゴリズムの概要

IBM® SPSS® Modeler のシーケンス ノードは、時間の経過に応じた購入傾向などのシーケンシャル データ中のパターンを検出します。シーケンス ノード アルゴリズムでは、シーケンス パターンのマイニングを行うために、次の 2 段階の処理が行われます(Agrawal および Srikant, 1995)。

- **頻出シーケンスをマイニングする。**この段階では、パターンのクエリーに対して迅速に応答するために必要な情報を抽出し、頻出シーケンスの隣接格子を生成します。この構造により、第 2 段階で最適な設定を行うことができます。
- **シーケンス パターンをオンラインで生成する。**この段階では、あらかじめ算出された隣接格子を使用します。パターンは、サポート（範囲）や確信度の境界など指定した基準に基づいて抽出したり、前提条件シーケンスに制限を設けることにより抽出することができます。

一次計算

アイテムセット、トランザクション、およびシーケンス

ある単一の時点に関連付けられたアイテムのグループは、**アイテムセット** を構成します。ここではアイテムセットを、大括弧 “{ }” を使って表します。ここで、ある食料品店の販売状況を表す以下のサンプル データを検討してみましょう。

テーブル 30-1
サンプル データ

	購入商品			
顧客	**時間 1**	**時間 2**	**時間 3**	**時間 4**
1	チーズ & クラッカー	wine（ワイン）	ビール	-
2	wine（ワイン）	ビール	チーズ	-
3	bread（パン）	wine（ワイン）	チーズ & ビール	-
4	クラッカー	wine（ワイン）	ビール	チーズ
5	ビール	チーズ & クラッカー	bread（パン）	-
6	クラッカー	bread（パン）	-	-

たとえば、顧客 1 の項目セットは {チーズ & クラッカー}、{ワイン}、および {ビール} のようになります。アンパサンド (&) は、単一のアイテムセット中に現れる複数のアイテムを表しています。この場合、& で区切られた各アイテムは、同じ購入時に現れます。アイテムセットによっては、1 つのアイテムしかないものもあります。

単一のオブジェクト（この場合は顧客）に対する完全なアイテムセットのグループが、**トランザクション**となります。時間は、特定の顧客に対する購入の機会を表しており、すべての顧客に対する特定の時間を表している訳ではありません。たとえば、顧客 1 の最初の購入機会は 1 月 23 日で、顧客 4 の最初の購入機会は 2 月 12 日になります。日付はそれぞれの顧客で異なりますが、アイテムセットはその顧客に対する初めてのアイテムセットになります。この分析では、絶対的な時間ではなく、特定の顧客に対する相対的な時間に注目します。

時間を基準にアイテムセットを並べると、**シーケンス**となります。記号 “>” はアイテムセットの並びを表しています。この記号の右側のアイテムセットが、左側のアイテムセットよりも後に発生したことを示します。たとえば、顧客 6 のシーケンスは［{クラッカー} > {パン}］のようになります。

一般的に、シーケンスを記述するためには、**サイズ**と**長さ**の、2 つの特徴が用いられます。シーケンス中のアイテム数がシーケンス サイズになります。シーケンス中のアイテムセット数が長さになります。たとえば、顧客 5 の 3 つの時間項目から、シーケンスの長さは 3 でサイズは 4 と判断することができます。

あるシーケンスからアイテムセットを削除することにより、別のシーケンスが得られる場合、その別のシーケンスはあるシーケンス（最初のシーケンス）の**サブシーケンス**となります。次のシーケンスを例に考えてみましょう。

［{ワイン} > {ビール} > {チーズ}］

ここで、アイテムセット「チーズ」を削除すると、長さ 2 のシーケンス［{ワイン} > {ビール}］が得られます。この 2 つのアイテムセットからなるシーケンスは、元のシーケンスのサブシーケンスになります。同じように削除することにより、この 3 つのアイテムセットからなるシーケンスを、それぞれ独立した単一のアイテムセットを持つ 3 種類のサブシーケンス（{ワイン}、{ビール}、{チーズ}）に分解することができます。また、それぞれが 2 つのアイテムセットを持つ 3 種類のサブシーケンス（［{ワイン} > {ビール}］、［{ビール} > {チーズ}］、［{ワイン} > {チーズ}］）に分解することもできます。他のシーケンスのサブシーケンスにならないシーケンスは、**最大シーケンス**と呼ばれます。

Support

シーケンスの**サポート**は、そのシーケンスを含むトランザクションの比率です。下の表は、食料品店の販売データセットの 1 つ以上の取引に出現するシーケンスのサポート値を示します（前述のデータ セットとは異なるデータ セットですので注意してください）。

たとえば、シーケンス［{ワイン} > {ビール}］は6 つ中 4 つのトランザクションに発生してるため、サポートは 0.67 になります。同様に、シーケンス ルールのサポートは、ルールの前提条件と結果の両方をその順序で含むトランザクションの比率になります。たとえば、次のシーケンス ルールのサポートを考えてみましょう。

If [{チーズ} >
{ワイン}] then [{ビール}]

この順序でこれら 3 つのアイテムセットを含むトランザクションは、6 つ中 1 つだけのため、サポートは 0.17 になります。

どのトランザクション中にも現れないシーケンスのサポート値は 0 となるため、マイニング分析からは除外されます。

テーブル 30-2
ゼロ以外のサポート値

シーケンス	Support	シーケンス	Support
{チーズ}	0.83	{クラッカー} > {チーズ}	0.17
{クラッカー}	0.67	{ビール} > {チーズ & クラッカー}	0.17
{ワイン}	0.67	{チーズ & クラッカー} > {ワイン}	0.17
{ビール}	0.83	{チーズ & クラッカー} > {ビール}	0.17
{パン}	0.50	{パン} > {チーズ & ビール}	0.17
{チーズ & クラッカー}	0.33	{ワイン} > {チーズ & ビール}	0.17
{チーズ & ビール}	0.17	{チーズ & クラッカー} > {パン}	0.17
{チーズ} > {ワイン}	0.17	{チーズ} > {ワイン} > {ビール}	0.17
{チーズ} > {ビール}	0.17	{クラッカー} > {ワイン} > {ビール}	0.33
{ワイン} > {ビール}	0.67	{ワイン} > {ビール} > {チーズ}	0.33
{クラッカー} > {ワイン}	0.33	{パン} > {ワイン} > {ビール}	0.17
{クラッカー} > {ビール}	0.33	{パン} > {ワイン} > {チーズ}	0.17
{ワイン} > {チーズ}	0.50	{ビール} > {チーズ} > {パン}	0.17
{ビール} > {チーズ}	0.50	{ビール} > {クラッカー} > {パン}	0.17
{パン} > {ワイン}	0.17	{クラッカー} > {ワイン} > {チーズ}	0.17
{パン} > {ビール}	0.17	{クラッカー} > {ビール} > {チーズ}	0.17
{パン} > {チーズ}	0.17	{チーズ & クラッカー} > {ワイン} > {ビール}	0.17
{ビール} > {パン}	0.17	{パン} > {ワイン} > {チーズ & ビール}	0.17
{ビール} > {クラッカー}	0.17	{ビール} > {チーズ & クラッカー} > {パン}	0.17
{チーズ} > {パン}	0.17	{クラッカー} > {ワイン} > {ビール} > {チーズ}	0.17
{クラッカー} > {パン}	0.33		

通常、分析では最小の閾値（**サポート レベル**）よりも大きいサポート値を持つシーケンスが注目されます。ユーザーにより定義されるこの閾値により、保持されるシーケンスの最小レベルが決まります。この閾値を超えるサポート値を持つシーケンス（**頻出シーケンス**）が、隣接格子の基盤を形成します。たとえば、閾値が 0.40 に設定されている場合、シーケンス［{ワイン} > {ビール}］のサポート値は 0.67 となるた

め、このシーケンスは頻出シーケンスになります。閾値の基準を緩和することにより、より多くのシーケンスが頻出シーケンスとして分類されます。

[時間] の制約

イベントの発生時期を定義することは、シーケンスに多大な影響を与えます。たとえば、前述の購買データでそれぞれの購入の機会から、新しく時間付けられたアイテムセットを生成することができます。たとえば、顧客がワインを購入した後、車に戻った時にビールも必要だったことを思い出した場合を考えてみましょう。この顧客はすぐにお店に戻って買い忘れた商品ビールを購入します。この場合、これらの 2 回の購入内容は個別の購入とみなすべきでしょうか？

きわめて短時間に発生した複数のアイテムセットを調整する 1 つの方法として、**タイムスタンプ許容度**パラメータを利用することが考えられます。タイムスタンプ許容度は、単一のアイテムセットをカバーする時間の長さを定義します。タイムスタンプ許容度に、2 つの連続する事象（の時間）よりも大きい値を指定すると、それらの事象は同時に発生した 1 つのアイテムセットになります（先ほどの例では {ワイン & ビール}）。

そのほかに、シーケンスの分析時に考慮する必要がある時間に関する問題が、**隔たり**です。この統計量は、2 つのアイテム間の時間の差異を測定するもので、時間に基づいて将来の行動を予測するために用いられます。隔たりの統計量は、シーケンス中の最後のセットと最後から 2 番目のセット間の隔たり、またはシーケンス中の最後のセットと最初のセット間の隔たりに基づいています。

シーケンス パターン

シーケンス パターン、または**シーケンシャル アソシエーション ルール**は、トランザクション ベースのデータ中で、他のアイテムの後に頻出するアイテムを示します。シーケンス パターンは、単にアイテムセットを順に並べたリストです。最終のアイテムセットに向かうすべてのアイテムセットが**前提条件**シーケンスを形成し、最後のアイテムセットが**結果シーケンス**になります。これらのステートメントの形式を次に示します。

If [前提条件] then [結果]

たとえば、ワイン、ビール、およびチーズのシーケンス パターンは「顧客がワインを購入してビールを購入した場合、その後チーズを購入する」となります。ワインとビールは前提条件、チーズは結果となります。

シーケンス ルール中の前提条件を結果と分類するために、表記する際には記号 “=>” を使用します。この記号の左側にあるシーケンスが前提条件に、右側にあるシーケンスが結果になります。たとえば、前述のルールは次のように表記することができます。

[{ワイン} > {ビール} => {チーズ}]

シーケンスとシーケンス ルール間の唯一の表記的な違いは、サブシーケンスを結果として識別することにあります。

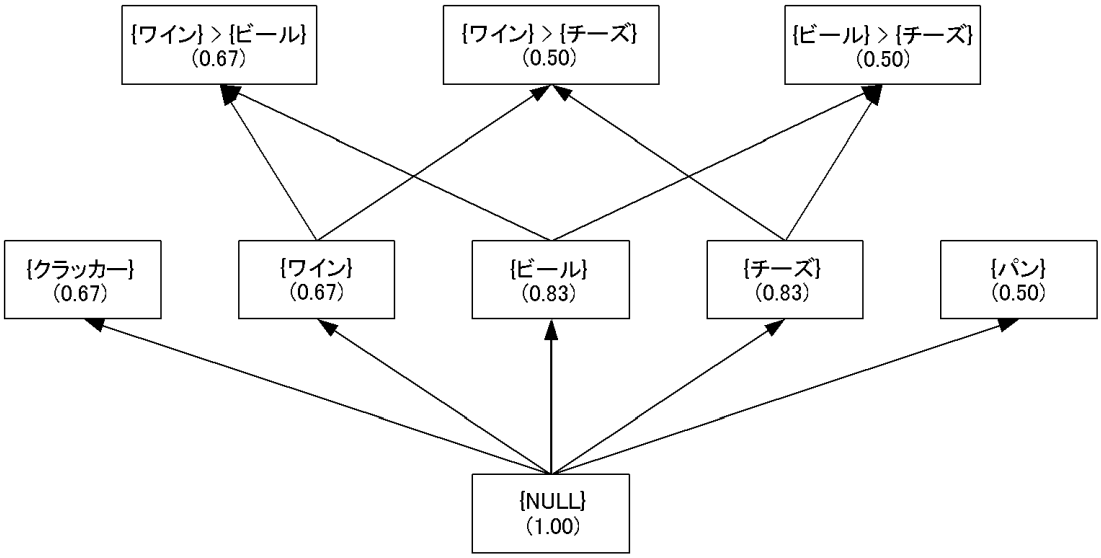
隣接格子

トランザクションの集合に対するアイテムセットおよびシーケンス数は、トランザクション中のアイテム数が増加するにつれて急速に大きくなります。通常、分析は多くのトランザクションに対して行われ、これらのトランザクションにはさまざまなアイテムセットが含まれています。大きいデータ セットの場合、特に急速なフィードバックが必要な場合に、シーケンス パターンを処理するために複雑な方法が必要になります。

隣接格子は、シーケンス パターンを素早く生成するための、シーケンスの編成構造を提供しています。あるシーケンスにアイテムを 1 つ加えるともう一方のシーケンスになる場合、それらの 2 つのシーケンスは**隣接**しており、どのシーケンスが他のシーケンスのサブシーケンスになるかを記述する階層構造を形成します。格子には、シーケンスの頻度や他の情報も含まれています。

すべての観測対象シーケンスの隣接格子は、一般的に大きすぎて実用的ではありません。構造を単純化するために、格子を剪定して頻出シーケンスを残す方が有用なこともあります。剪定された構造には、一定のサポート レベルに達するすべてのシーケンスが残されます。サンプル トランザクションによるサポート レベル 0.40 の隣接格子を次に示します。

図 30-1
閾値 0.40 の隣接格子（括弧内はサポート値）



頻出シーケンスのマイニング

IBM® SPSS® Modeler では、非シーケンシャル アソシエーション ルールによるマイニング方法が用いられています。この方法は、I/O コスト、時間、およびスペースの要件を最小化する観点から適しています。**連続アソシエーション ルール マイニング アルゴリズム** (Continuous Association Rule Mining Algorithm : Carma) では、2 つのデータ パスが使用され、実行時にサポート レベルを変更することができます。(Hidber, 1999)最終的に保証されるサポート レベルは、提供された一連のサポート値により異なります。

マイニング処理の最初の段階では、ある方法の Carma を使用して、その手法をシーケンシャル ケースに適用します。一般的な操作の順序は、次のようになります。

- トランザクション データを読み込みます。
- 頻出シーケンスを識別し、頻出しないシーケンスを破棄します。
- 頻出シーケンスの隣接格子を構築します。

Carma はトランザクションに基づいており、2 つのデータ パスしか必要ありません。フェーズ 1 と呼ばれる最初のデータ パスでは、頻出シーケンスの候補が生成されます。2 番目のデータ パス、フェーズ 2 では、フェーズ 1 で生成された候補シーケンスの正確な度数カウントが計算されます。

フェーズ 1

フェーズ 1 は推定フェーズに対応しています。このフェーズで Carma は、各トランザクションに対して候補シーケンスを連続的に生成します。候補シーケンスは、ある種類の「Apriori」の原理を満たしており、あるシーケンスに対して前のトランザクションからのすべてのサブシーケンスが候補である場合にだけ、そのシーケンスが候補になります。そのため、候補シーケンスのサイズは、各トランザクションにおいて増加する可能性があります。候補数が大きくなりすぎないように、Carma では度数の閾値に達しない候補シーケンスが定期的に剪定されます。剪定は、一定数のトランザクションを処理した後に行われます。剪定によりメモリー要件は低下しますが、演算コストは増加します。フェーズ 1 の最後には、算出されたサポート レベル（一連のサポートによって異なる）を超える度数を持つすべてのシーケンスが生成されます。Carma では、複数のサポート レベルを使用することができます。最高で、1 トランザクションあたり 1 つのサポート レベルを使用できます。

前述の購買データに対して剪定を行わない場合の、トランザクション処理中のサポート値を次の表に示します。トランザクションを処理するにつれて、そのトランザクション中に現れるアイテムおよび処理されたトランザクション総数に対応してサポート値が調整されます。たとえば、最初のトランザクション後に、格子にはチーズ、クラッカー、ワイン、およびビールが含まれており、それぞれが閾値レベルを超えるサポート値を持っています。2 番目のトランザクションの処理後に、クラッカーは 2 つのトランザクション中の 1 つにしか現れないため、このアイテムのサポート値は 1.0 から 0.50 に低下します。他のアイテムは、両方のトランザクション中に含まれているため、サポート値は変化しません。さらに、シーケンスを構成するサブシーケンスがすでに格子に存在しているため、シーケンス [{ワイン} > {ビール}] および [{ビール} > {チーズ}] が格子に入れられます。

テーブル 30-3
Carma のトランザクション処理

	トランザクション					
シ◆◆◆ケンス	1	2	3	4	5	6
{チーズ}	1	1	1	1	1	0.83
{クラッカー}	1	0.50	0.33	0.50	0.60	0.67
{ワイン}	1	1	1	1	0.80	0.67
{ビール}	1	1	1	1	1	0.83
{ワイン} > {ビール}		1	1	1	0.80	0.67

	トランザクション					
シ◆◆◆ケンス	1	2	3	4	5	6
{ビール} > {チーズ}		0.50	0.33	0.50	0.60	0.50
{パン}			0.33	0.25	0.40	0.50
{ワイン} > {チーズ}			0.67	0.75	0.60	0.50
{チーズ & ビール}			0.33	0.25	0.20	0.17
{クラッカー} > {ワイン}				0.50	0.40	0.33
{クラッカー} > {ビール}				0.50	0.40	0.33
{クラッカー} > {チーズ}				0.25	0.20	0.17
{ワイン} > {ビール} > {チーズ}				0.50	0.40	0.33
{チーズ & クラッカー}					0.40	0.33
{ビール} > {クラッカー}					0.20	0.17
{ビール} > {パン}					0.20	0.17
{チーズ} > {パン}					0.20	0.17
{クラッカー} > {パン}					0.20	0.33

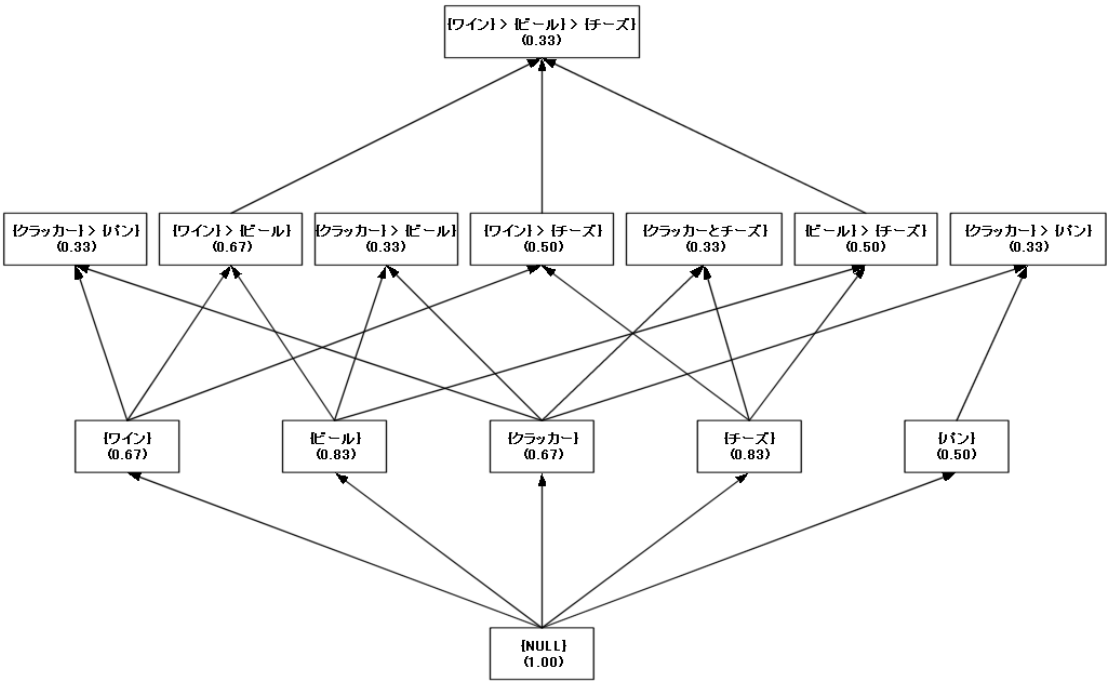
最初のデータ パスの完了後、格子には 1 つのアイテムを含む 5 つのシーケンス、2 つのアイテムを含む 12 のシーケンス、および 3 つのアイテムを含む 1 つのシーケンスが存在しています。

フェーズ 2

フェーズ 2 は検証フェーズで、候補シーケンスの正確な度数が判断されます。このフェーズでは候補シーケンスは生成されず、頻出しないシーケンスの剪定が 1 回だけ行われるため、フェーズ 1 よりもフェーズ 2 の方が速くなります。さらに、フェーズ 1 中の候補シーケンスの入力ポイントによっては、完全なデータ パスが必要ないこともあります。オンライン アプリケーションでは、Carma はフェーズ 2 全体を省略します。

格子の閾値レベルが 0.30 の場合を考えてみましょう。いくつかのシーケンスはこのレベルに達しないため、フェーズ 2 でこれらのシーケンスは剪定されます。剪定後の格子は次のようになります。

図 30-2
閾値 0.30 の隣接格子 (括弧内はサポート値)



[{クラッカー} > {ワイン} > {ビール}] のサポート値は閾値を超えているのに、格子に存在していないことに注目してください。シーケンス [{クラッカー} > {ワイン} > {ビール}] は 1/3 のトランザクションに発生していますが、それを構成するサブシーケンスがすべて格子内に取り込まれない限り、このシーケンスを格子に追加することはできません。最後の 2 つのサブシーケンスは 4 番目のトランザクションに発生していますが、その後完全な 3 つのアイテムセット シーケンスは観測されません。ただし、一般的にデータベースのトランザクションは、ここで取り上げられている例に比べて非常に大きいため、このような形で除外されることは滅多にありません。

シーケンス パターンの生成

シーケンス パターン マイニング処理の第 2 段階では、第 1 段階で生成された頻出シーケンスの隣接格子から、実際のパターンを照会します。Aggarwal および Yu (1998a) IBM® SPSS® Modeler は、隣接格子からオンラインにアソシエーション ルールを生成するために、効率的なアルゴリズムのセットを使用しています(Aggarwal および Yu, 1998)。これらのアルゴリズムをシーケンシャル ケースに適用すると、隣接格子データ構造により保持されたルール範囲と確信値の単調な特徴が活用されます。格子は、シーケンス パターンの生成に必要なすべての情報を効率的に保存し、生成できるすべてのパターンよりも極端に小さくなります。

クエリーには、結果として生成されるシーケンス パターンのセットが満たす必要がある制約が含まれています。これらの制約は、次の 2 つのカテゴリに分けられます。

- 統計インデックスの制約
- パターンの前提条件に含まれるアイテムの制約

統計インデックスの制約には、サポート、確信度、または原因などが含まれます。これらのクエリーでは、返されるパターンに指定した範囲内の統計量の値がなければなりません。通常は、確信度の下限が第 1 の基準になります。パターン サポート レベルの下限は、対応する隣接格子中のシーケンスのサポート レベルにより与えられます。ただし、パターン生成に指定されたサポート値が、格子作成に指定された値を超えてしまうことがしばしばあります。

前述の格子の場合、サポートの範囲を 0.30 ～ 1.00、確信度の範囲を 0.30 ～ 1.0、および原因の範囲を 0 ～ 1.0 に指定すると、次の 7 つのルールが生成されます。

- If [{クラッカー}] then [{ビール}].
- If [{クラッカー}] then [{ワイン}].
- If [{クラッカー}] then [{ビール}].
- If [{ワイン} > {ビール}] then [{チーズ}].
- If [{ワイン}] then [{ビール}].
- If [{ワイン}] then [{チーズ}].
- If [{ビール}] then [{チーズ}].

セットを最大シーケンスだけに制限すると、最後の 3 つのルールが省略されます（4 番目のルールのサブシーケンスのため）。

2 番目の種類のクエリーでは、シーケンス ルールの前提条件の指定が必要になります。この種類のクエリーでは、前提条件中の最終アイテムセットの後に新しい単一のアイテムセットを返します。たとえば、買い物かごに商品を入れたオンラインショッピングの顧客を例に考えてみましょう。将来のアイテム クエリーは、過去の購入履歴だけを参考にして、次に顧客がサイトを訪れた時のお奨め商品を作成します。

空白の処理

シーケンス ルール アルゴリズムでは、空白は無視されます。このアルゴリズムでは、入力フィールドに対して空白を含むレコードは処理されます。ただし、そのようなレコードは、1 つまたは複数のフィールドに対して空白値を持つ任意のルールと一致するとはみなされません。

二次計算

Confidence

確信度はシーケンス ルールの精度を表す測度で、ルールの前提条件と結果の両方を含むトランザクション数を、前提条件を含むトランザクション数で除算した値になります。つまり、確信度はルールのサポート値を、前提条件のサポート値で除算した値になります。たとえば、次のシーケンス ルール

If [{ワイン}] then
[{チーズ}]

の確信度は 3/4、または 0.75 になります。ワインを含むトランザクションの 3/4 は、その後チーズも含むことになります。逆に、次のシーケンス ルール ：

If [{チーズ}] then
[{ワイン}]

は、先ほどと同じアイテムセットを含んでいますが、確信度は 0.20 になってしまいます。チーズを含むトランザクションの 1/5 だけが、その後ワインも含むことになります。つまり、ワインがチーズにつながる可能性は、チーズからワインにつながる可能性よりも高いと判断することができます。

に、前述の購買データ中に観測される各シーケンス ルールの確信度を示します。空の前提条件を持つルールは、前の購入（トランザクション）履歴がないものに対応しています。

テーブル 30-4
ゼロ以外の確信度値

シ◆◆◆ケンス	Confidence	シ◆◆◆ケンス	Confidence
{チーズ}	1.00	{クラッカー} => {チーズ}	0.25
{クラッカー}	1.00	{ビール} => {チーズ & クラッカー}	0.20
{ワイン}	1.00	{チーズ & クラッカー} => {ワイン}	0.50
{ビール}	1.00	{チーズ & クラッカー} => {ビール}	0.50
{パン}	1.00	{パン} => {チーズ & ビール}	0.33
{チーズ & クラッカー}	1.00	{ワイン} => {チーズ & ビール}	0.25
{チーズ & ビール}	1.00	{チーズ & クラッカー} => {パン}	0.50
{チーズ} => {ワイン}	0.20	{チーズ} > {ワイン} => {ビール}	1.00
{チーズ} => {ビール}	0.20	{クラッカー} > {ワイン} => {ビール}	1.00
{ワイン} => {ビール}	1.00	{ワイン} > {ビール} => {チーズ}	0.50
{クラッカー} => {ワイン}	0.50	{パン} > {ワイン} => {ビール}	1.00
{クラッカー} => {ビール}	0.50	{パン} > {ワイン} => {チーズ}	1.00
{ワイン} => {チーズ}	0.75	{ビール} > {チーズ} => {パン}	0.33
{ビール} => {チーズ}	0.60	{ビール} > {クラッカー} => {パン}	1.00
{パン} => {ワイン}	0.33	{クラッカー} > {ワイン} => {チーズ}	0.50
{パン} => {ビール}	0.33	{クラッカー} > {ビール} => {チーズ}	0.50
{パン} => {チーズ}	0.33	{チーズ & クラッカー} > {ワイン} => {ビール}	1.00
{ビール} => {パン}	0.20	{パン} > {ワイン} => {チーズ & ビール}	1.00
{ビール} => {クラッカー}	0.20	{ビール} > {チーズ & クラッカー} => {パン}	1.00

シ◆◆◆ケンス	Confidence	シ◆◆◆ケンス	Confidence
{チーズ} => {パン}	0.20	{クラッカー} > {ワイン} > {ビール} => {チーズ}	0.50
{クラッカー} => {パン}	0.50		

生成されたモデル／スコアリング

予測値

シーケンス ルール モデルにデータ レコードを渡すと、モデルはレコードを時間に依存する方法で処理します（モデル構築にタイムスタンプ フィールドが使われていない場合は、順序に依存する方法で）。レコードは、ID フィールドとタイムスタンプ フィールド（ある場合）でソートされていなければなりません。

各レコードに対して、これまでに現在の ID に対して処理された一連のトランザクションとモデル中のルールが比較されます（現在のレコード、および同じ ID とより以前のタイムスタンプを持つ前のレコードを含む）。この一連のトランザクションに適用される、もっとも高い確信度値を持つ k ルールが、レコードの k 予測を生成するために用いられます。ここで、k はモデル構築時に指定された予測数を示します(複数のルールがトランザクション セットに対して同じ結果を予測した場合、もっとも確信度が高いルールだけが使用されます)。

各レコードに対する予測が必ずしもレコードのトランザクションに依存するわけではないことに注意してください。現在のレコードのトランザクションが特定のルールの要因とならない場合、ルールは現在の ID の以前のトランザクションに基づいて選択されます。つまり、現在のレコードにより有益な予測情報がシーケンスに追加されない場合は、この ID に対する前回の有益な予測が現在のレコードで使用されます。

たとえば、あるシーケンス ルール モデルに次のルールがある場合に、

ジャム -> パン (0.66)

これに次のレコードを渡す場合を考えてみましょう。

ID	購入品	予測
001	ジャム	bread (パン)
001	牛乳	bread (パン)

最初のレコードは、期待通りに予測「パン」を生成しています。2 番目のレコードの場合、「ジャム」の後に「ミルク」が続くルールはないため、「ミルク」トランザクションにより有益な情報は追加されません。そのため、ルール「**ジャム -> パン**」のルールが引き続き適用され、予測は「パン」になります。

Confidence

予測に関連する確信度は、予測を生成するルールの確信度になります。 詳細は、p.298 Confidence を参照してください。

空白の処理

シーケンス ルール アルゴリズムでは、空白は無視されます。このアルゴリズムでは、入力フィールドに対して空白を含むレコードは処理されます。ただし、そのようなレコードは、1 つまたは複数のフィールドに対して空白値を持つ任意のルールと一致するとはみなされません。

章
31
自己学習応答モデル アルゴリズム

自己学習応答モデル (SLRM) では、Naive Bayes 分類辞を使用してモデルを作成します。そのモデルは容易に更新され、モデル全体を再生成することなく新しいデータを組み込むことができます。ここで、SLRM によるモデルの作成、更新およびスコアリングに用いられる方法を説明します。

一次計算

SLRM に使用されるモデル構築アルゴリズムは Naive Bayes です。各ターゲットフィールドの Naive Bayes モデルで構成される Bayesian Network が生成されます。

Naive Bayes アルゴリズム

Naive Bayes モデルは、その単純さと安定性のため復興を享受している分類および予想値選択のための古い方法です。

表記

この章では特に明記しない限り、次の表記を使用します。

J_0	総予測数。
X	カテゴリ予測フィールド ベクトルは **X'** = (X_1, ..., X_J) となり、ここで J は検討された予測フィールドの数を表します。
M_j	予測フィールド X_j のカテゴリ数。
Y	カテゴリのターゲット変数。
K	Y のカテゴリ数。
N	学習データのケースまたはパターンの総数。
N_k	学習データ内の Y= k を含むケースの数。
N^j_{mk}	学習データ内の Y= k および X_j=m を含むケースの数。
π_k	Y= k の確率。
p^j_{mk}	Y= k を指定された X_j=m の確率。

Naive Bayes モデル

Naive Bayes モデルは、対象クラスを与えられた各予測フィールドの条件独立モデルに基づいています。Bayesian の原則は、最も大きな事後確率を持つクラスにケースを割り当てることです。Bayes の 定理では、, **X** を与えられた Y の確率は次のようになります。

$$P(Y=k|\mathbf{X}=\mathbf{x}) = \frac{P(\mathbf{X}=\mathbf{x}|Y=k)P(Y=k)}{\sum_{i=1}^{K} P(\mathbf{X}=\mathbf{x}|Y=i)\,P(Y=i)}$$

X_1, ..., X_J を、モデルで検討した J 予測フィールドとします。Naive Bayes もでるでは、X_1, ..., X_J は、条件的に独立してターゲットを与えられたいると仮定します。つまり、次のようになります。

$$P\left(\mathbf{X}=\mathbf{x}\,|Y=k) = \prod_{j=1}^{J} P\left(X_j=x_j|\,Y=k\right)\right.$$

これらの確率は、学習データから次の方程式で推定されます。

$$\pi_k = P(Y=k) = \frac{N_k+\lambda}{N+K\lambda}$$

$$p_{mk}^{j} = P(X_j=m|Y=k) = \frac{N_{mk}^{j}+f}{\Sigma_{l=1}^{M_j} N_{lk}^{j} + M_j f}$$

ここで N_k はすべての欠損していない Y に基づいて計算され、N^j_{mk} は欠損していないすべての X_J および Y に基づいています。また因子 λ および f が導入され、ゼロまたは非常に小さいセルの度数から生じる問題を解決します。これらの推定値は、Dirichlet 事前確率による多項式確率の Bayesian 推定に対応しています。経験的研究では、$\lambda = f = \frac{1}{N}$ (Kohavi, Becker, および Sommerfield, 1997) を提案しています。

単一データの受け渡しが、関連するすべての度数を収集するために必要です。

J = 0 となる特別な状況、つまり予測フィールドがひとつもない場合は、$P(Y=k|\mathbf{X}=\mathbf{x}) = P(Y=k)$ となります。ターゲット変数またはカテゴリ予測フィールドに空のカテゴリがある場合、これらのカテゴリを計算から削除する必要があります。

二次計算

モデルパラメータに加え、モデルの評価が計算されます。

モデルの評価

学習済みモデルの場合、その信頼度を評価する必要があります。こうした問題が与えられると、異なる解決方法を導く 2 つの条件に直面します。

- テスト データのサンプル(モデルの学習または更新には使用されない)が利用可能です。この場合、これらのデータをモデルに直接反映したり、結果を観察できます。
- 利用可能な追加のテスト データはありません。ユーザーは通常すべての利用可能なデータを適用してモデルを学習するため、このことはより一般的です。この場合、 π_k や p_{mk}^{j} など調整済みのモデル パラメータに基づいてデータをシミュレーションし、その後これらの擬似無作為データをスコアリングして、学習済みのモデルを評価します。

シミュレーションしたデータを使用したテスト

シミュレーションでは、 $n_{round} \times n_{sample\_per\_round}$ のデータが生成されます。各 round に対し、対応する精度を定義できます。すべての round に対しては、平均精度および分散を計算でき、それらは信頼性の統計として説明されます。

- 各 round に対し、n_{sample} のランダム ケースを次のように生成します。
  - y は、事前確率 π_k に基づいてランダム値が割り当てられます。
  - それぞれの X_i は、条件に関連する確率 $\hat{P}(X_j|Y=y)$ に基づいて、無作為に割り当てられます。
- 各 round の精度は、各ケースのモデルの予測された値をそのケースの生成された結果 y と比較して計算され、 $P_{accuracy} = \frac{n_{correct}}{n_{sample}}$ となります。
- 推定精度の平均、分散、最小および最大は、round 全体で計算されます。

空白の処理

ターゲットが存在しない場合、またはケースの J_0 予測フィールドが存在しない場合、そのケースは無視されます。予測フィールドのすべての値が存在しない場合、または予測フィールドの欠損値以外のすべての値が同じである場合、その予測フィールドは無視されます。

モデルの更新

「Naive Bayes モデル」 （ p. 302 ）の説明にあるとおり、新規レコードを説明するセルの度数 N_k、N_{mk}^j を更新し、確率 π_k および p_{mk}^j を再計算して、モデルを更新することができます。モデルの更新には、新規レコードのデータを渡す必要があります。

生成されたモデル／スコアリング

生成された SLRM モデルによるスコアリングについては、次に説明します。

予測された値および確信度

デフォルトでは、最大の予測された値を持つ最初のオファー M が戻されます。ただし、マーケティング戦略では低い可能性のオファーが求められる場合があります。モデル設定を行うと、特定のオファーに対する結果を与えたり、オファーにランダムなコンポーネントを含めることができます。

オファーのスコアリングの表記は次のとおりです。

N	すでにモデル化されているオファーの数
$P = \{P_1, P_2, ..., P_N\}$	各オファーのスコア
$P_r = \{P_{r1}, P_{r2}, ..., P_{rN}\}$	オファーに対し無作為に生成されたスコア

α	0.0（モデル予測のみに基づいたオファー）から 1.0 （完全にランダムなオファー）の範囲のランダム化ファクタ
$W = \{W_1, W_2, ..., W_N\}$	各オファーの学習に使用されるケース数
W_{emp}	信頼できるモデルおなる学習ケース数の経験的値（現在の実装では、500 の定数に設定します）
$S = \{S_1, S_2, ..., S_N\}$	オファーに対するユーザーの選択またはオファーの評価。負の数ではない値となる場合があり、より大きな値の場合は、対応するオファーに対するより強い購買アドバイスを意味します。デフォルトの設定は、$S = \{1, 1, ..., 1\}$ です。
$F = \{F_1, F_2, ..., F_N\}$	必須である選択または除外のフィルタ。$F_i \in \{0,1\}$ ここで、0 は除外されたオファーを表します。

各オファーの最終スコアリングは、次のように計算されます。

$$P_i = \left[\alpha P_{ri} + (1-\alpha)\left(\frac{W_i}{W_i + W_{emp}}P + \frac{W_{emp}}{W_i + W_{emp}}0.5\right)\right] \cdot \frac{S_i}{\max(S)} \cdot F_i$$

結果 P_i は降順または昇順の指定された順序に並んでおり、リストの最初の M のオファーが推奨されます。計算されたスコアリングは、スコアリングの確信度として報告されます。

変数の評価

モデル化されたすべての機能には、他の機能よりもモデルの精度に対して絶対的に重要なものがあります。ここで提案される重要度を測定するアプローチは、次の 2 通りです。予測値の重要度および情報の測定

予測値の重要度

予測可能なエラーの分散は、重要度の測定として使用することができます。この方法で、1 つの予測フィールド変数を 1 度に除外し、残りのモデルのパフォーマンスを観察します。変数が（すべての変数を持つ）完全なモデルの分散に比べてより多くの分散を追加した場合、その変数はほかの変数に比べてより重要とみなされます、

テスト データが利用可能な場合、それらのデータを使用して直接予測値の重要度を計算することができます。テストデータを利用できない場合、それらのデータはモデルパラメータ π_k および p^j_{mk} に基づいてシミュレーションされます。.

シミュレーションでは、 $n_{round} \times n_{sample\_per\_round}$ のデータが生成されます。各 round に対し、各サブモデルの対応する精度を、各 j 予測フィールドの X_j を除いて定義します。すべての round で、平均の精度および分散を計算することができます。

- 各 round に対し、n_{sample} のランダム ケースを次のように生成します。

- y は、事前確率 π_k に基づいてランダム値が割り当てられます。
- それぞれの X_i は、条件に関連する確率 $\hat{P}(X_j|Y=y)$ に基づいて、無作為に割り当てられます。

round 内では、それぞれの X_j 予測フィールドがモデルから削除され、各サブモデルに順番に生成されたテスト データに基づいて精度が計算されます。

- 各 round の精度は、各ケースのサブモデルの予測された値をそのケースの生成された結果 y と比較して計算され、それぞれの j モデルに対し $P_{accuracy\_without\_xj} = \frac{n_{correct\_without\_xj}}{n_{sample}}$ となります。
- 推定精度の平均および分散は、各サブモデルの round 全体で計算されます。各変数に対し、重要度は完全モデルの制度と変数を除いたサブモデルの平均精度との差として測定されます。

情報の測定

反応変数 Y に対する説明変数 X の重要度は、X の使用によって Y の結果を予測する際の不確実性を減少させる程度です。結果 Y の予測に関する不確実性は、分布のエントロピーによって次のように測定されます（Shannon 1948）。

$$H_Y \equiv -\sum_i P(Y=i)\log P(Y=i)$$

説明変数の値 x に基づいて、結果 Y の確率分布は条件分布 $f_{y|x}$ となります。予測フィールドの値 x を使用した情報値は、限界分布 f_y および条件分布 $f_{y|x}$ の集中度を比較して評価されます。条件分布エントロピーと限界分布エントロピーとの差は次のとおりです。

$$\Delta H(x_j) = H_Y - H_{Y|x_j}$$

ここで、$H_{Y|x_j}$ は条件分布 $f_{y|x_j}$ のエントロピーをあらわします。条件分布$f_{y|x_j}$ が f_y より集中している場合、値 x_j は Y に関してより有益となります。

Y を予測する場合のランダム値 X の重要度は、期待される不確実性の減少によって測定され、2 つの変数間の「相互情報」と呼ばれます。

$$\begin{aligned} M(Y,X) &\equiv \sum_i f_x(x_j)\,\Delta H(x_j) \\ &= H_Y - H_{Y|X} \\ &= H_Y + H_X - H_{Y,X} \end{aligned}$$

X による不確実性の減少の期待される割合は、次の式により生じる相互情報の指標です。

$$I_{y,x} = 1 - \frac{H_{Y|X}}{H_Y} = \frac{M_{Y,X}}{H_Y}$$

この指標は、0 〜 1 です。2 つの変数が独立している場合にのみ $I_{y,x} = 0$ となり、2 つの変数が線状または非線状の形式で、機能的に関連する場合にのみ $I_{y,x} = 100\%$ となります。

章

32

Support Vector Machine (SVM) アルゴリズム

Support Vector Machine アルゴリズムの概要

Support Vector Machine (SVM) は、ラベルの付いた学習データのセットから入出力マッピング関数を生成する監視学習方法です。マッピング関数は、分類関数または回帰関数のいずれかです。分類関数の場合、非線型カーネル関数を入力データが元の入力領域と比較してより分割できるようになる高次元の関数領域に転送された入力データに使用されます。最大余白の超平面が作成されます。作成されるモデルは、クラス境界の近くの学習データのサブセットによってのみ異なります。

同様に、Support Vector Regression で作成されたモデルは、モデル予測値に十分に近い学習データを無視します。(Support Vectors は、エラー チューブの境界またはチューブ外にのみ表示されます。)

SVM アルゴリズムの表記

x_i	i 番目の学習サンプル
y_i	i 番目の学習サンプルのクラス ラベル
l	学習サンプルの数
$K(x_i \cdot x_j)$	サンプル i、j のペアのカーネル関数値です。
$Q(x_i \cdot x_j) = y_i y_j K(x_i \cdot x_j)$	行 i および列 j のカーネル行列の要素
α_i	学習サンプルの係数 (サポート ベクトル以外の場合 0)
α_i^*	サポートベクトル回帰モデルの学習モデルの係数
$f(x)$	決定関数
m	学習サンプルのクラスの数
C	すべての変数の上限
e	すべての要素が 1 に等しいベクトル
$sgn(x)$	sign 関数: $\begin{cases} 1 & (x \geq 0 \text{の場合}) \\ -1 & (\text{それ以外の場合}) \end{cases}$

SVM の種類

この項では、LIBSVM テクニカル レポート(Chang および Lin, 2003)の説明に基づいて、使用できる SVM の種類を説明します。$K(x_i \cdot x_j)$ は、ユーザーが選択したカーネル関数です。 詳細は、 p.312 SMO アルゴリズム を参照してください。

C-Support Vector Classification (C-SVC)

2 つのクラスで、学習ベクトル $x_i \in R^l$, i = 1, ..., l、および $y_i \in \{-1,1\}$ のようなベクトル $y \in R^l$ の場合、C-SVC では次の 2 つの問題を解決します。

$$\min f(\alpha) = \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$

たとえば $0 \le \alpha_i \le C, i = 1,...,l$ and $y^T\alpha = 0$ で、

$$\alpha = (\alpha_1, \alpha_2, ..., \alpha_l)^T$$

および Q が $l \times l$ 行列の場合、$Q(x_i \cdot x_j) = y_i y_j K(x_i \cdot x_j)$ となります。

決定関数

$$sgn\left(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b\right)$$

の場合、b は、定数項となります。

ε -Support Vector Regression (ε -SVR)

回帰モデルでは、n 次元の入力ベクトル **x** の、従属（ターゲット）変数 $y \in R$ の関数依存関係を推定します。そのため、分類の問題とは異なり、実数の関数およびモデルを処理し、$R^n \to R^1$ のマッピングを行います。たとえば、$x_I \in R^n$ が入力で、$z_i \in R^1$ が対象出力となるような、データ $\{(x_1, z_1), ..., (x_l, z_l)\}$ のセットの場合、ε -Support Vector Regression の双対形式は、

$$\min f(\alpha, \alpha^*) = \frac{1}{2}(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + z_i\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)$$

となり、たとえば $i = 1,...,l$ の場合は $0 \le \alpha_i$ および $\alpha_i^* \le C$、

$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0$$

この場合、$\alpha = (\alpha_1, \alpha_2, ..., \alpha_l)^T$, $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_l^*)^T$, and Q is an $l \times l$ matrix, $Q_{ij} = K(x_i \cdot x_j)$ となります。

近似関数が

$$\sum_{i=1}^{l} (-\alpha_i + \alpha_i^*) K(x_i, x) + b$$

この場合、b は、定数項となります。

一次計算

SVM モデル構築の一次計算は、下記で説明しています。

二次問題の解決

決定関数または近似関数を見つけるために、二次問題を解決する必要があります。解決された後で、次のようにさまざまな α_i 係数を取得できます。

- $0 < \alpha_i < C$ の場合、対応する学習サンプルは**自由サポート ベクトル**です。
- $\alpha_i = C$ の場合、対応するサポート ベクトルは**境界サポートベクトル**です。
- $\alpha_i = 0$ の場合、対応する学習サンプルは、**非サポート ベクトル**で、分類結果または回帰結果には影響を与えません。

自由サポート ベクトルおよび境界サポート ベクトルは、**サポート ベクトル**と呼ばれています。

このマニュアルでは分解方法を適合させて、2 番目のオーダー情報（Fan, Chen, および Lin, 2005）を使用して二次問題を解決します。統一されたフレームワークですべての SVM の問題を解決するために、C-SVC および ε-SVR に一般的な形式を導入します。

ε-SVR の場合、双対形式 を次のようにリライトできます。

$$\min \quad f(\alpha, \alpha^*) = \frac{1}{2} \left(\alpha^T (\alpha^*)^T \right) \begin{pmatrix} Q & -Q \\ -Q & Q \end{pmatrix} \begin{pmatrix} \alpha \\ \alpha^* \end{pmatrix} + \begin{pmatrix} \varepsilon e^T + z^T \\ \varepsilon e^T - z^T \end{pmatrix}^T \begin{pmatrix} \alpha \\ \alpha^* \end{pmatrix}$$

たとえば $y^T \begin{pmatrix} \alpha \\ \alpha^* \end{pmatrix} = 0$ および $0 \le \alpha_i, \alpha_i^* \le C$ for i = 1, ... , l となり、ここで、i = 1, ..., l の場合、$y_i = 1$ の $2l \times 1$ のベクトルで、i = l + 1, ... , 2l の場合、$y_i = -1$ となります。

この場合、一般形式は

$$\min f(\alpha) = \frac{1}{2} \alpha^T Q \alpha + p^T \alpha$$

たとえば、i = 1, ... , l の場合 $0 \le \alpha_i \le C$ となり、$y^T\alpha = constant$ です。

	α W(α) の場合	p^T	$y^T\alpha = constant$
C-SVC	$(\alpha_1, \alpha_2, ..., \alpha_l)^T$	$-e^T$	$y = (y_1, ..., y_l)^T$ $y^T\alpha = 0$
ε -SVR	$(\alpha_1, \alpha_2, ..., \alpha_l, \alpha_1^*, \alpha_2^*, ..., \alpha_l^*)^T$	$\begin{pmatrix} \varepsilon e^T + z^T \\ \varepsilon e^T - z^T \end{pmatrix}^T$	$y = (1_1, ..., 1_l, -1_{l+1}, ..., -1_{2l})^T$ $y^T\alpha = 0$

決定関数の定数

二次プログラムの問題を解決した後、決定関数のサポート ベクトル係数を取得します。決定関数の定数項も計算する必要があります。2 つの従属変数 r_1 および r_2 を導入します。

- y_i = 1の場合、次のようになります。

  $0 < \alpha_i < C$ の場合、

  $$r_1 = \frac{\Sigma_{0<\alpha_i<C, y_i=1} \nabla f(\alpha_i)}{\Sigma_{0<\alpha_i<C, y_i=1} 1}$$

  それ以外の場合、

  $$r_1 = \frac{\max_{\alpha_i=C, y_i=1} \nabla f(\alpha_i) + \min_{\alpha_i=0, y_i=1} \nabla f(\alpha_i)}{2}$$

- y_i = −1 の場合、次のようになります。

  $0 < \alpha_i < C$ の場合、

  $$r_2 = \frac{\Sigma_{0<\alpha_i<C, y_i=-1} \nabla f(\alpha_i)}{\Sigma_{0<\alpha_i<C, y_i=-1} 1}$$

  それ以外の場合、

  $$r_2 = \frac{\max_{\alpha_i=C, y_i=-1} \nabla f(\alpha_i) + \min_{\alpha_i=0, y_i=-1} \nabla f(\alpha_i)}{2}$$

  r_1 および r_2 が取得されると、 $b = \frac{r_1 + r_2}{2}$ を計算します。

可変スケール

連続型入力変数の場合、次のように各属性を［-1, 1］または［0, 1］に線型に変化させます。

$$V^{new} = \frac{V - V_{\min}}{V_{\max} - V_{\min}} (newmax - newmin) + newmin$$

カテゴリ入力フィールドにおいて、m 個のカテゴリが存在する場合、カテゴリ数（0、1、2、...、m）を使用して、カテゴリ数を表し連続型入力変数に関して値を測定します。

モデル構築アルゴリズム

この項では、SVM を学習する高速アルゴリズムを提供します。変更された逐次最小最適化（SMO）アルゴリズムは、C-SVC バイナリおよび ε-SVR モデルに提供されます。分割統治に基づく SVM 高速学習アルゴリズムは、すべての SVM に使用されます。

SMO アルゴリズム

カーネル行列の密度により、従来の最適化方法を直接適用してベクトル α を解決することはできません。反復プロセスの各ステップでベクトル α 全体を更新する多くの最適化方法とは異なり、分解方法では反復ごとに α のサブセットを変更します。作業セット B と示されるこのサブセットの小さい下位問題は、各反復で最小化されます。逐次最小最適化（SMO）は、B に 2 つの要素しか存在しないように制限する手法の極端な例です。それぞれの反復には、2 つの変数の単純な問題を解決するために最適化アルゴリズムは必要ありません。SML の主要なステップは作業セット選択方法で、アルゴリズムの収束の速度を決定します。

カーネル関数

このアルゴリズムでは、次の 4 つのカーネル関数をサポートしています。

線型関数	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j$
多項式関数	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \left(\gamma \mathbf{x}_i^T \cdot \mathbf{x}_j + r\right)^d$
RBF 関数	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$ $= \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \gamma = \frac{1}{2\sigma^2}$
Sigmoid 関数	$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \tanh\left(\gamma \mathbf{x}_i^T \cdot \mathbf{x}_j + r\right)$ $\tanh(x) = \frac{e^x}{e^x + 1}$

基本作業セット選択アルゴリズム

基本選択アルゴリズムは、τ、C、対象ベクトル **y** および選択されたカーネル関数 K($\mathbf{x}_i$, $\mathbf{x}_j$) に基づいて選択セット B = {i, j} を算出します。

以下のように仮定します。

$$a_{ij} = K_{ii} + K_{jj} - 2K_{ij}, b_{ij} = -y_i \nabla f\left(\alpha^k\right)_i + y_j \nabla f\left(\alpha^k\right)_j$$

であり、

$$\overline{a}_{ts} = \begin{cases} a_{ts} & (a_{ts} > 0 \\ \tau & \text{の場合}) \end{cases}$$

ここで τ は小さい正の数です。

選択

$$i \in \arg\max_t \left\{ -y_t \nabla f\left(\alpha^k\right)_t | t \in I_{up}\left(\alpha^k\right) \right\},$$
$$j \in \arg\min_t \left\{ -\frac{b_{it}^2}{\overline{a}_{it}} | t \in I_{low}\left(\alpha^k\right), -y_t \nabla f\left(\alpha^k\right)_t < -y_i \nabla f\left(\alpha^k\right)_i \right\}$$

ここで、

$$I_{up}(\alpha) = \{t | \alpha_t < C, y_t = 1 \text{ or } \alpha_t > 0, y_t = -1\}$$

$$I_{low}(\alpha) = \{t | \alpha_t < C, y_t = -1 \text{ or } \alpha_t > 0, y_t = 1\}$$

B = {i, j} を戻し、この場合 $\nabla f(\alpha) = Q\alpha + \mathbf{p}$ です。

縮小アルゴリズム

反復プロセスが終わる前にアルゴリズムの収束を短縮するため、分割方法は最後の自由サポート ベクトルをすべて含む可能なセット A を識別します。そのため、問題全体を解決する代わりに、分割方法はより小さい問題を処理します。

$$\min_{\alpha_A} \quad \frac{1}{2}\alpha_A^T Q_{AA}\alpha_A - \left(\mathbf{p}_A - Q_{AN}\alpha_N\right)^T \alpha_A$$

s. t. $0 \le (\alpha_A)_t \le C, t = 1, \cdots, q$

$$\mathbf{y}_A^T \alpha_A = const - \mathbf{y}_N^T \alpha_N$$

ここでは、$N = \{1, 2, \cdots l\} \setminus A$ は、縮小された変数のセットです。

すべての min(l, 1000) 反復の後、一部の変数を縮小します。反復プロセス中は、$m\left(\alpha^k\right) > M\left(\alpha^k\right)$ です。$m\left(\alpha^k\right) - M\left(\alpha^k\right) \le \epsilon$ が満たされると、 次のセットの変数を縮小できます。

$$\left\{ t | -y_t \nabla f(\alpha_t) > m\left(\alpha^k\right), \alpha_t = C, y_t = 1 \text{ or } \alpha_t = 0, y_t = -1 \right\} \cup$$
$$\left\{ t | -y_t \nabla f(\alpha_t) < M\left(\alpha^k\right), \alpha_t = 0, y_t = 1 \text{ or } \alpha_t = C, y_t = -1 \right\}$$

そのため、有効化された変数のセット A は、min(l, 1000) 反復ごとに動的に減少します。

- 積極的すぎる縮小方法の傾向を説明するために、許容度が以下に達した場合に変化を再構築します。

$$m\left(\alpha^k\right) \le M\left(\alpha^k\right) + 10\epsilon$$

変化を再構築した後、以下の関係に基づいて以前縮小した変数の一部を復元します。

$$\left\{t| -y_t \nabla f\left(\alpha_t\right) \le m\left(\alpha^k\right), \alpha_t = C, y_t = 1 \text{ or } \alpha_t = 0, y_t = -1\right\} \cup$$
$$\left\{t| -y_t \nabla f\left(\alpha_t\right) \ge M\left(\alpha^k\right), \alpha_t = 0, y_t = 1 \text{ or } \alpha_t = C, y_t = -1\right\}$$

変化の再構築

変化 $\nabla f(\alpha)$ の再構築にかかるコストを縮小するため、 反復時は以下を常に保持します。

$$\overline{G}_i = C \sum_{\alpha_j = C} Q_{ij} \quad i = 1, \cdots l$$

そして変化 $\nabla f(\alpha_i)$, $i \notin A$ の場合、

$$\nabla f\left(\alpha_i\right) = \sum_{j=1}^{l} Q_{ij}\alpha_j + p_i = \overline{G}_i + \sum_{0<\alpha_j<C} Q_{ij}\alpha_j + p_i$$

変化 $\nabla f(\alpha_i)$, $i \in A$ の場合、

$$\nabla f\left(\alpha_i^{k+1}\right) = \nabla f\left(\alpha_i^k\right) + Q_{it}\Delta\alpha_t + Q_{is}\Delta\alpha_s$$

t および s は、作業セットの指標です。

不均衡データの戦略

分類の問題に対し、このアルゴリズムでは SVM 定式でさまざまなパラメータを使用します。α^k を更新する手順のみ異なります。さまざまな条件が次のように処理されます。

例

条件		パラメータの更新
$y_i \ne y_j$	$\alpha_i - \alpha_j > C_i - C_j$ および $\alpha_i > C_i$	$\alpha_i^{new} = C_i$ $\alpha_j^{new} = C_i - (\alpha_i - \alpha_j)$
	$\alpha_i - \alpha_j \le C_i - C_j$ および $\alpha_j > C_j$	$\alpha_j^{new} = C_j$ $\alpha_i^{new} = C_j + (\alpha_i - \alpha_j)$
	$\alpha_i - \alpha_j > 0$ および $\alpha_j < 0$	$\alpha_j^{new} = 0$ $\alpha_i^{new} = (\alpha_i - \alpha_j)$
	$\alpha_i - \alpha_j \le 0$ および $\alpha_i < 0$	$\alpha_i^{new} = 0$ $\alpha_j^{new} = -(\alpha_i - \alpha_j)$

$y_i = y_j$	$\alpha_i + \alpha_j > C_i$ および $\alpha_i > C_i$	$\alpha_i^{new} = C_i$ $\alpha_j^{new} = (\alpha_i + \alpha_j) - C_i$
	$\alpha_i + \alpha_j \le C_i$ および $\alpha_j < 0$	$\alpha_j^{new} = 0$ $\alpha_i^{new} = (\alpha_i + \alpha_j)$
	$\alpha_i + \alpha_j > C_j$ および $\alpha_j > C_j$	$\alpha_i^{new} = (\alpha_i + \alpha_j) - C_j$ $\alpha_j^{new} = C_j$
	$\alpha_i + \alpha_j \le C_j$ および $\alpha_i < 0$	$\alpha_i^{new} = 0$ $\alpha_j^{new} = (\alpha_i + \alpha_j)$

SMO 分解

SMO 分解の手順は、次のとおりです。

1. 初期の許容解の α^1 を検出して、k = 1 を設定します。
2. α^k が定常解の場合、停止します。

   $m(\alpha) - M(\alpha) \le \epsilon$ の場合許容解は定常解となり、

   $$m(\alpha) = \max_{i \in I_{up}} \{-y_i \nabla f(\alpha_i)\}$$

   $$M(\alpha) = \min_{i \in I_{low}} \{-y_i \nabla f(\alpha_i)\}$$

   $$I_{up}(\alpha) = \{t | \alpha_t < C, y_t = 1 \text{ or } \alpha_t > 0, y_t = -1\}$$

   $$I_{low}(\alpha) = \{t | \alpha_t < C, y_t = -1 \text{ or } \alpha_t > 0, y_t = 1\}$$

   作業セット選択アルゴリズムを使用して、2 つの要素のワーキングセット $B = \{i, j\}$ を検出します。（詳細は、 p. 312 基本作業セット選択アルゴリズム を参照してください。）
3. 縮小アルゴリズムを使用して収束を短縮する場合、SMO 分解アルゴリズムを適用します。（詳細は、 p. 313 縮小アルゴリズム を参照してください。）
4. α^{k+1} を以下のように算出します。

- $C_i \ne C_j$ の場合、または分類の問題を解決する場合、不均衡データ戦略を使用します。（詳細は、 p. 314 不均衡データの戦略 を参照してください。）
- $a_{ij} > 0$ の場合、次の下位問題を解決します。

$$\min_{\alpha_B} \quad \frac{1}{2}\begin{bmatrix}\alpha_i & \alpha_j\end{bmatrix}\begin{bmatrix}Q_{ii} & Q_{ij} \\ Q_{ji} & Q_{jj}\end{bmatrix}\begin{bmatrix}\alpha_i \\ \alpha_j\end{bmatrix} + \left(-\mathbf{p}_B + Q_{BN}\alpha_N^k\right)^T \begin{bmatrix}\alpha_i \\ \alpha_j\end{bmatrix} + \text{cont}$$

以下の制約

$$0 \le \alpha_i, \alpha_j \le C,$$
$$y_i\alpha_i + y_j\alpha_j = -\mathbf{y}_N^T\alpha_N^k$$

に従って、以下のとおりとします。

$$\alpha_i^{new} = \alpha_i^{old} + \frac{y_i b_{ij}}{a_{ij}}$$
$$\alpha_j^{new} = \alpha_j^{old} - \frac{y_j b_{ij}}{a_{ij}}$$

- そうでない場合、以下の下位問題を解決します。

$$\min_{\alpha_B} \quad \frac{1}{2}\begin{bmatrix}\alpha_i & \alpha_j\end{bmatrix}\begin{bmatrix}Q_{ii} & Q_{ij}\\ Q_{ji} & Q_{jj}\end{bmatrix}\begin{bmatrix}\alpha_i\\ \alpha_j\end{bmatrix} + \left(-\mathbf{p}_B + Q_{BN}\alpha_N^k\right)^T\begin{bmatrix}\alpha_i\\ \alpha_j\end{bmatrix} + \frac{\tau - a_{ij}}{4}\left(\left(\alpha_i - \alpha_i^k\right)^2 + \left(\alpha_j - \alpha_j^k\right)^2\right)$$

前述の制約に従い、 τ が小さい正の数で、$N = \{1,2,\cdots l\} \setminus B$ である場合、以下のようになります。

$$\alpha_i^{new} = \alpha_i^{old} + \frac{y_i b_{ij}}{\tau}$$
$$\alpha_j^{new} = \alpha_j^{old} - \frac{y_j b_{ij}}{\tau}$$

最後に α_B^{k+1} を下位問題の最適なポイントになるよう設定します。

$\alpha_N^{k+1} = \alpha_N^k$ を設定、$k = k + 1$ を設定し、ステップ 2 に進みます。

SVM 高速学習

SVM 二項モデルの場合、学習サンプル 1 の数が大きい場合、密度の高いカーネル行列をメモリーに保存することはできません。カーネル行列を計算するキャッシュ戦略に依存する標準分割アルゴリズムを使用せず、分割統治方法が使用され、元の問題を SMO アルゴリズム（Dong, Suen, および Krzyzak, 2005）で解決できる小さい下位問題に分割します。それぞれの下位問題に対し、カーネル行列を連続メモリーの一部として定義されたカーネルキャッシュに保存することができます。カーネル行列のサイズは、学習セット全体のサポート ベクトルをすべて保持するのに十分で、メモリー制約を満たすことができる大きさである必要があります。下位問題のカーネル行列が完全にキャッシュされているため、カーネル行列の各要素を 1 回のみ評価し、高速方法を使用して計算する必要があります。

SVM 高速学習方法には、次の 2 つの手順があります。

- 並列最適化
- 高速逐次最適化

これらの手順については、次で詳細に説明されています。

並列最適化

カーネル行列 Q が対象で半正定値性である場合、ブロック対角行列は半正定値性で、以下のように表されます。

$$Q_{diag} = \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_k \end{bmatrix}$$

この場合、$l_i \times l_i$ 行列の $Q_i, i = 1, \cdots, k, \sum_i^k l_i = l$ はブロック対角です。基本作業セット選択アルゴリズム p. 312 で説明されているように、最適化下位問題の k を取得します。すべての下位問題は、SMO 分解アルゴリズムを使用して、並行して最適化されます。並列最適化の後、ほとんどの非サポートベクトルは学習セットから削除されます。その後、下位問題のサポートベクトルを収集して新しい学習セットを取得できます。新しい学習セットのサイズは元の学習セットより非常に小さいものですが、特に大きいデータセットを処理する場合、メモリーはカーネル行列を保存できるほど大きくありません。このため、高速逐次最適化技術が使用されます。

高速逐次最適化

高速逐次最適化の技術は、問題のサブセットを反復して最適化することにより動作します。最初、学習セットがシャッフルされ、すべての $\alpha_i, i = 1, \cdots, l$ が 0 に設定され、サブセット $\mathrm{Sub} \subseteq S$ が学習セット S から選択されます。サブセット d のサイズが設定されます（$d \leq l$）。

最適化の手順は次のとおりです。

- SMO アルゴリズムを適用して、カーネル キャッシングを含む Sub の下位問題を最適化し、α_i およびカーネル行列を更新します。 詳細は、 p. 312 SMO アルゴリズム を参照してください。
- キュー サブセット方法を使用して、新しいサブセットを選択します。学習セットのすべてのサポート ベクトルを格納し、メモリーの制約を満たすのに十分なサブセットのサイズを選択します。 詳細は、 p. 318 サブセット選択のキュー方法 を参照してください。
- 次の停止条件がいずれも真でない場合、ステップ 1 に戻ります。
  - $|\Delta SV| < 20$ および（学習サンプル数）> 1
  - $SV \geq (d - 1)$
  - (学習サンプル数) $> T \cdot l$

この場合、$|\Delta SV|$ は 2 つの連続するサブセット間のサポート ベクトル数の変更で、l は新しい学習セットのサイズ、T（> 1.0）は有効なデータ全体におけるループのユーザー定義の最大数です。

サブセット選択のキュー方法

キュー方法を使用して、高速逐次最適化で学習できる学習セットのサブセットを選択します。 詳細は、 p. 317 高速逐次最適化 を参照してください。.

この方法は、学習データの最初の d レコードを含むサブセットおよび残りすべてのレコードを含むキュー $\mathbf{Q}_S$ を設定し、サブセットのカーネル行列を計算して初期化されます。

初期化されると、次のようにサブセット選択が行われます。サブセットの非サポート ベクトルがキューの終わりに追加され、キューの最初のレコードを持つサブセットに置き換えられます（結果的にはキューから削除されます)。すべての非サポート ベクトルが置き換えられると、サブセットが最適化に返されます。次の反復時、同じプロセスが適用され、最後の反復の終わりと同じ状態のサブセットおよびキューで始まります。

空白処理

入力フィールドまたは出力フィールドに欠損値があるすべてのレコードは、モデルの推定から除外されます。

モデルナゲット/スコアリング

SVM Model モデル ナゲットは、出力クラスの予測および予測確率を生成します。予測は、各レコードの最も高い予測確率を持つカテゴリに基づいています。

予測値を選択するために、事後確率が sigmoid 関数(Platt, 2000) を使用して推定されます。使用される近似式は次のとおりです。

$$P\left(y=1|x\right)\approx P_{A,B}\left(x\right)=\frac{1}{1+\exp\left(Af\left(x\right)+B\right)}.$$

最適化パラメータ A および B は、ラベルの付いた例 $(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_l,y_l),\mathbf{x}\in\mathbf{R}^n,y\in\{+1,-1\}$ を持つ次の正規化最大尤度問題を解決して推定され、N_+ は正の例の数で、N_- は負の例の数を表します。

$$\min_{z=(A,B)}\quad F\left(z\right)=-\sum_{i=1}^{l}\left(t_i\log\left(p_i\right)+\left(1-t_i\right)\log\left(1-p_i\right)\right)$$

$$p_i=P_{A,B}\left(x_i\right)\text{ および }t_i=\begin{cases}\frac{N_++1}{N_++2} & \text{if } y_i=+1\\ \frac{1}{N_-+2} & \text{if } y_i=-1\end{cases},i=1,\cdots,l$$

$$\nabla F(z) = \begin{bmatrix} \sum_{i=1}^{l} f_i (t_i - p_i) \\ \sum_{i=1}^{l} (t_i - p_i) \end{bmatrix}$$

$$H(z) = \nabla^2 F(z) = \begin{bmatrix} \sum_{i=1}^{l} f_i^2 p_i (1 - p_i) & \sum_{i=1}^{l} f_i p_i (1 - p_i) \\ \sum_{i=1}^{l} f_i p_i (1 - p_i) & \sum_{i=1}^{l} p_i (1 - p_i) \end{bmatrix}$$

空白処理

入力フィールドに欠損値があるレコードはスコアリングできません。そのため、予測フィールドおよび確率値に $null$ が割り当てられます。

章
33

時系列アルゴリズム

時系列ノードは、時系列のための単変量指数平滑法、ARIMA（自己回帰統合移動平均）分析、および転送関数（TF）モデルを構築し、予測を生成します。このプロシージャは、従属変数系列ごとに適切なモデルを特定および予測するExpert Modelerを含んでいます。また、カスタム モデルを指定することもできます。

このアルゴリズムは、University of Chicago の Ruey Tsay 教授の助言を想定して設計されています。

表記

この章では特に明記しない限り、次の表記を使用します。

Y_t (t=1, 2, ..., n)	調査中の単変量時系列
n	総観測数。
$\hat{Y}_t(k)$	系列 Y の時間 t におけるモデル推定 k ステップ先の予測。
S	季節性長さ。

モデル

The Time Series nodeは、指数平滑化モデルおよび ARIMA/TF モデルを予測します。

指数平滑化モデル

以下の表記は指数平滑化モデルに固有のものです。

α	レベル平滑化重み
γ	トレンド平滑化重み
ϕ	減衰トレンド平滑化重み
δ	季節平滑化重み

単純指数平滑化

単純指数平滑化には一つのレベル パラメータがあり、次の式で表すことができます。

$$L(t) = \alpha Y(t) + (1-\alpha) L(t-1)$$

$$\hat{Y}_t(k) = L(t)$$

これは機能的には ARIMA(0,1,1) プロセスと同じです。

Brown's 指数平滑化

Brown's 指数平滑化にはレベルとトレンドのパラメータがあり、次の式で表すことができます。

$$L(t) = \alpha Y(t) + (1-\alpha) L(t-1)$$

$$T(t) = \alpha(L(t) - L(t-1)) + (1-\alpha) T(t-1)$$

$$\hat{Y}_t(k) = L(t) + \left((k-1) + \alpha^{-1}\right) T(t)$$

これは、MA パラメータに制約があるものの、機能的には ARIMA(0,2,2) と同じです。

Holt's 指数平滑化

Holt's 指数平滑化にはレベルとトレンドのパラメータがあり、次の式で表すことができます。

$$L(t) = \alpha Y(t) + (1-\alpha)(L(t-1) + T(t-1))$$

$$T(t) = \gamma(L(t) - L(t-1)) + (1-\gamma) T(t-1)$$

$$\hat{Y}_t(k) = L(t) + kT(t)$$

これは機能的には ARIMA(0,2,2) と同じです。

減衰トレンド指数平滑化

減衰トレンド指数平滑化にはレベルと減衰トレンドのパラメータがあり、次の式で表すことができます。

$$L(t) = \alpha Y(t) + (1-\alpha)(L(t-1) + \phi T(t-1))$$

$$T(t) = \gamma(L(t) - L(t-1)) + (1-\gamma)\phi T(t-1)$$

$$\hat{Y}_t(k) = L(t) + \sum_{i=1}^{k} \phi^i T(t)$$

これは機能的には ARIMA(1,1,2) と同じです。

単純季節性指数平滑化

単純季節性指数平滑化にはレベルと季節のパラメータがあり、次の式で表すことができます。

$$L(t) = \alpha(Y(t) - S(t-s)) + (1-\alpha) L(t-1)$$

$$S(t) = \delta(Y(t) - L(t)) + (1-\delta) S(t-s)$$

$$\hat{Y}_t(k) = L(t) + S(t+k-s)$$

これは、MA パラメータに制約があるものの、機能的には ARIMA(0,1,(1,s,s+1))(0,1,0) と同じです。

Winters' 相加指数平滑化

Winders' 相加指数平滑化にはレベル、トレンドおよび季節のパラメータがあり、次の式で表すことができます。

$$L(t) = \alpha(Y(t) - S(t-s)) + (1-\alpha)(L(t-1) + T(t-1))$$

$$T(t) = \gamma(L(t) - L(t-1)) + (1-\gamma) T(t-1)$$

$$S(t) = \delta(Y(t) - L(t)) + (1-\delta) S(t-s)$$

$$\hat{Y}_t(k) = L(t) + kT(t) + S(t+k-s)$$

これは、MA パラメータに制約があるものの、機能的には ARIMA(0,1,s+1)(0,1,0) と同じです。

Winters' 相乗指数平滑化

Winders' 相乗指数平滑化にはレベル、トレンドおよび季節のパラメータがあり、次の式で表すことができます。

$$L(t) = \alpha(Y(t)/S(t-s)) + (1-\alpha)(L(t-1) + T(t-1))$$

$$T(t) = \gamma(L(t) - L(t-1)) + (1-\gamma) T(t-1)$$

$$S(t) = \delta(Y(t)/L(t)) + (1-\delta) S(t-s)$$

$$\hat{Y}_t(k) = (L(t) + kT(t)) S(t+k-s)$$

同等の ARIMA モデルはありません。

指数平滑化の推定と予測

1 ステップ先の予測の誤差の平方和、$\Sigma\left(Y_t - \hat{Y}_{t-1}(1)\right)^2$、は平滑化重みを最適化するために最小化されます。

指数平滑化の初期化

L はレベルを表し、T はトレンドを、S、すなわち長さsのベクトルは季節ステートを表すとします。初期平滑化ステートは、t=n から t=0 へと再構成することによって行います。ここでは、再構成の初期化について解説します。

すべてのモデルの場合、$L = y_n$.

トレンドによるすべての非季節性モデルの場合、T はリグレッサとしての時間でデータに適合された線の（切片を含む）勾配です。

単純季節モデルの場合、S の要素は標本平均を差し引いた季節平均であり、たとえば、月次データの場合、1月に対応する要素は標本平均を差し引いたサンプルにおけるすべての1月の値の平均です。

Winters の相加モデルについては、$y = \alpha t + \sum_{i=1}^{s} \beta_i I_i(t)$ をデータに適合させますが、この場合、t は時間であり、$I_i(t)$ は季節ダミーです。モデルには切片はありません。この場合、$T = \alpha$ であり、$S = \beta - mean(\beta)$ です。

Wintersの相乗モデルの場合、リグレッサとしての時間で季節ごとに個別の線（切片を含む）を適合させます。μ を切片のベクトル、β を勾配のベクトルと仮定します（これらのベクトルは長さ s です）。この場合、$T = mean(\beta)$ であり、$S = (\mu + \beta) / (\Sigma \mu_i + \beta_i)$ です。

ARIMA および転送関数のモデル

以下の表記はARIMA/TFモデルに固有のものです。

a_t (t = 1, 2, ..., n)	通常、平均値ゼロおよび分散 σ^2 で分布するホワイト ノイズ系列。
p	モデルの非季節性自己回帰部分の順序
q	モデルの非季節性移動平均部分の順序
d	非季節性差分の順序
P	モデルの季節性自己回帰部分の順序
Q	モデルの季節性移動平均部分の順序
D	季節性差分の順序
s	モデルの季節性または期間
$\phi_p(B)$	順序 p の B の AR 多項式、$\phi_p(B) = 1 - \varphi_1 B - \varphi_2 B^2 - ... - \varphi_p B^p$
$\theta_q(B)$	順序 q の B の MA 多項式、$\theta_q(B) = 1 - \vartheta_1 B - \vartheta_2 B^2 - ... - \vartheta_q B^q$
$\Phi_P(B^s)$	順序 P の BS の 季節性 AR 多項式、$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{s2} - ... - \Phi_P B^{sP}$
$\Theta_Q(B^s)$	順序 P の BS の 季節性 AR 多項式、$\Theta_Q(B^s) = 1 - \Theta_1 B^s - \Theta_2 B^{s2} - ... - \Theta_Q B^{sQ}$
Δ	差分演算子 $\Delta = (1 - B)^d (1 - B^s)^D$
B	$BY_t = Y_{t-1}$ および $Ba_t = a_{t-1}$ による変数減少法シフト演算子。
$Z\sigma_t^2$	Z_t の予測分散
$N\sigma_t^2$	ノイズ予測の予測分散

転送関数（TF）モデルはひじょうに大きなクラスのモデデルを形成し、これは特殊なケースとして単変量 ARIMA モデルを含んでいます。Y_t が従属系列である場合、およびオプションとして、$X_{1t}, X_{2t}, ..., X_{kt}$ をこのモデルで予測値系列として使用する場合を仮定します。従属系列と予測値系列との関係を説明する TF モデルは次のような形式になっています。

$$Z_t = f(Y_t),$$

$$\Delta Z_t = \mu + \sum_{i=1}^{k} \frac{Num_i}{Den_i} \Delta_i B^{b_i} f_i(X_{it}) + \frac{MA}{AR} a_t.$$

単変量 ARIMA モデルは TF モデルからの予測値を単に減少させるだけであり、したがって、以下のような形式になります。

$$\Delta Z_t = \mu + \frac{MA}{AR} a_t$$

このモデルの主な特徴は以下のとおりです。

- 従属系列と予測値系列の初期変換 f および f_i。この変換はオプションであり、従属系列の値が正である場合にのみ適用されます。許容される変換は対数と平方根です。これらの変換は分散安定化変換と呼ばれることもあります。
- 定数項 μ.
- 分散 σ^2の互いに独立で同一の分布に従うゼロ平均値、ガウス エラー処理 a_t。
- 移動平均ラグ多項式 MA=$\theta_q(B)\Theta_Q(B^s)$ および自己回帰ラグ多項式 AR=$\phi_p(B)\Phi_P(B^s)$。
- 差分/ラグ演算子 Δ and Δ_i。
- 遅延項、B^{b_i}。この場合、b_i は遅延の順序です。
- 予測値が与えられるものとします。分子と分母のラグ多項式は以下のような形式になります。Num_i=$(\omega_{i0} - \omega_{i1}B - \cdots - \omega_{iu}B^u)(1 - \Omega_{i1}B^s - \cdots - \Omega_{iv}B^{vs})B^b$ and Den_i=$(1 - \delta_{i1}B - \cdots - \delta_{ir}B^r)(1 - \Delta_{i1}B^s - \cdots)$.
- “ノイズ” 系列

  $$N_t = \Delta Z_t - \mu - \sum_{i=1}^{k} \frac{Num_i}{Den_i} \Delta_i B^{b_i} X_{it}$$

  これは平均値ゼロの固定 ARMA プロセスとみなされます。

ARIMA/TF の推定と予測

利用可能な 2 種類の予測アルゴリズムがあります。条件付き最小 2 乗（CLS）と正確最小 2 乗（ELS）または無条件最小 2 乗（ULS）予測。これらの 2 種類のアルゴリズムの違いは一つだけです。すなわち、ノイズ処理は別々に予測します。予測計算の一般的なステップは以下のとおりです。

1. 時系列期間におけるノイズ処理の計算 N_t。

2. 予測期間までのノイズ処理の予測 N_t。これは時系列期間中の 1 ステップ先の予測、およびその後の数ステップ先の予測です。CLS と ELS 予測手法での相違がこのステップで表面化します。ノイズ予測の予測分散もまたこのステップで計算します。

3. 最終的な予測は、まず定数項の寄与率と転送関数をノイズ予測に追加し、次にその結果を統合および逆変換することによって得られます。最終予測分散を得るために、ノイズ予測の予測分散を処理しなければならない場合もあります。

   $\hat{N}_t(k)$ と$\sigma_t^2(k)$ をそれぞれ k ステップ先の予測および予測分散とします。

条件付き最小 2 乗 (CLS) 法

$\hat{N}_t(k) = E(N_{t+k}|N_t, N_{t-1}, \cdots)$ assuming $N_t = 0$ for t<0.

$$\sigma_t^2(k) = \sigma^2 \sum_{j=0}^{k-1} \psi_j^2$$

この場合、ψ_j は $MA/(\Delta \times AR)$ のべき級数展開の係数です。

$S = \Sigma\left(N_t - \hat{N}_t(1)\right)^2$ を最小限に抑えます。

欠損値は N_t の予測値で代入します。

最尤 (ML) 法 (Brockwell と Davis、1991)

$\hat{N}_t(k) = E(N_{t+k}|N_t, N_{t-1}, \cdots, N_1)$

$\left\{N_t - \hat{N}_t(1)\right\}_{t=1}^{n}$ の最尤法。すなわち、

$$L = -\ln(S/n) - (1/n)\sum_{j=1}^{n} \ln(\eta_j)$$

この場合、$S = \Sigma\left(N_t - \hat{N}_t(1)\right)^2/\eta_t$、および $\sigma_t^2 = \sigma^2\eta_t$ は 1 ステップ先の予測分散です。

欠損値が存在する場合は、Kalman フィルタを使用して $\hat{N}_t(k)$ を計算します。

誤差分散

$\hat{\sigma}^2 = S/(n-k)$

どちらの方法においても上記のようになります。ここで、n はゼロ以外の残差の数であり、k はパラメータの数です（誤差分散を除く）。

ARIMA/TF の初期化

やや修正された Levenberg-Marquardt アルゴリズムは、目的関数を最適化するのに使用します。修正はパラメータに対する “許容性” 制約を考慮しています。許容制約の場合、AR と MA 多項式のルートは単位円の外側にある必要があり、分母多項式パラメータは各予測変数についてゼロ以外である必要があります。極小化アルゴリズムでは、反復検索を行うための開始値が必要です。分子と分母の多項式パラメータはすべてゼロに初期化されますが、ただし、対応する回帰係数に初期化される分母多項式の 0 乗の係数を除きます。

ARMA パラメータは以下のように初期化されます。

系列 Y_t は 平均値 0 の ARMA(p,q)(P,Q) モデルに従うものと仮定します。すなわち、

$$Y_t - \varphi_1 Y_{t-1} - \cdots - \varphi_p Y_{t-p} = a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

以下の場合、c_l と ρ_l は Y_t の 1 回目の自己共分散と自己相関をそれぞれ表し、$\hat{c}_l$ と $\hat{\rho}_l$ はそれらの推定を表します。

非季節性 AR パラメータ

AR パラメータの初期値の場合、推定される方法は (Box, Jenkins, および Reinsel, 1994) の付録A6.2の内容と同じです。$\hat{\varphi}'_1, \cdots, \hat{\varphi}'_{p+q}$ として推定を表します。

非季節性 MA パラメータ

次のように仮定します。

$$w_t = Y_t - \varphi_1 Y_{t-1} - \cdots - \varphi_p Y_{t-p} = a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

相互共分散

$$\lambda_l = E(w_{t+l} a_t) = E((a_{t+l} - \theta_1 a_{t+l-1} - \cdots - \theta_q a_{t+l-q}) a_t) = \begin{cases} \sigma_a^2 & l = 0 \\ -\theta_1 \sigma_a^2 & l = 1 \\ \cdots & \cdots \\ -\theta_q \sigma_a^2 & l = q \\ 0 & l > q \end{cases}$$

AR(p+q) が Y_t に近似すると仮定すると、以下のようになります。

$$Y_t - \varphi'_1 Y_{t-1} - \cdots - \varphi'_p Y_{t-p} - \varphi'_{p+1} Y_{t-p-1} - \cdots - \varphi'_{p+q} Y_{t-p-q} = a_t$$

このモデルの AR パラメータは上記のように推定され、$\hat{\varphi}'_1, \cdots, \hat{\varphi}'_{p+q}$ と表されます。

したがって、λ_l は以下により推定されます。

$$\begin{aligned} \lambda_l &\approx E\Big((Y_{t+l} - \varphi_1 Y_{t+l-1} - \cdots - \varphi_p Y_{t+l-p}) \left(Y_t - \varphi'_1 Y_{t-1} - \cdots - \varphi'_{p+q} Y_{t-p-q} \right) \Big) \\ &= \left(\rho_l - \sum_{j=1}^{p+q} \varphi_j \rho_{l+j} - \sum_{i=1}^{p} \varphi_i \rho_{l-i} + \sum_{i=1}^{p} \sum_{j=1}^{p+q} \varphi_i \varphi_j \rho_{l+j-i} \right) c_0 \end{aligned}$$

さらに、誤差分散 σ_a^2 は以下により近似されます。

$$\hat{\sigma}_a^2 = Var\left(-\sum_{j=0}^{p+q} \varphi^{'}{}_j Y_{t-j}\right) = \sum_{i=0}^{p+q}\sum_{j=0}^{p+q} \varphi^{'}{}_i \varphi^{'}{}_j c_{i-j} = c_0 \sum_{i=0}^{p+q}\sum_{j=0}^{p+q} \varphi^{'}{}_i \varphi^{'}{}_j \rho_{i-j}$$

ここでは、$\hat{\varphi}_0^{'} = -1$ とします。

次に、初期 MA パラメータは $\theta_l = -\lambda_l / \sigma_a^2$ によって近似され、以下により推定されます。

$$\hat{\theta}_l = -\hat{\lambda}_l / \hat{\sigma}_a^2 = \frac{\rho_l - \sum_{j=1}^{p+q} \hat{\varphi}_j \rho_{l+j} - \sum_{i=1}^{p} \hat{\varphi}_i \rho_{l-i} + \sum_{i=1}^{p}\sum_{j=1}^{p+q} \hat{\varphi}_i \hat{\varphi}_j \rho_{l+j-i}}{\sum_{i=0}^{p+q}\sum_{j=0}^{p+q} \hat{\varphi}_i^{'} \hat{\varphi}_j^{'} \rho_{i-j}}$$

したがって、$\hat{\theta}_l$ は $\hat{\varphi}_j^{'}, \hat{\varphi}_i$、および $\{\hat{\rho}_l\}_{l=1}^{p+2q}$ によって計算できます。この手順では、$\{\hat{\rho}_l\}_{l=1}^{p+q}$ だけが使用され、その他のすべてのパラメータは 0 に設定されます。

非季節性パラメータ

季節性 AR および MA コンポーネントについては、上記の式の季節ラグにおける自己相関を使用します。

診断統計

ARIMA/TF 診断統計は、ノイズ処理 $R(t) = N(t) - \hat{N}(t)$ の残差に基づきます。

Ljung-Box 統計

$$Q(K) = n(n+2)\sum_{k=1}^{K} r_k^2 / (n-k)$$

この場合、r_k は残差の k 回目のラグ ACF です。

Q(K) は $\chi^2(K-m)$ として近似的に分散しますが、この場合、m は定数項および予測値関連のパラメータ以外のパラメータの数です。

時系列分析における外れ値検出

観測系列は、いわゆる外れ値に悪影響を受けることがあります。これらの外れ値は、悪影響を受けていない系列の平均レベルを変化させる場合があります。外れ値検出の目的は、外れ値があるかどうか、外れ値の場所、タイプ、絶対値を判断することです。

時系列ノードは7種類の外れ値を考慮に入れています。それらは、相加的外れ値（AO）、技術革新的外れ値（IO）、レベル シフト（LS）、一時的（または過渡的）変化（TC）、季節性相加（SA）、ローカル トレンド（LT）、および AO パッチ（AOP）です。

表記

以下の表記は外れ値検出に固有のものです。

U(t) または U_t	外れ値なしの悪影響を受けない系列。これは単変量 ARIMA または転送関数のモデルとみなされます。

外れ値の定義

外れ値のタイプはここで個別に定義します。実際には、研究中の系列においてこれらのタイプの組み合わせが発生する場合があります。

AO（相加的外れ値）

t=T の時に AO 外れ値が発生すると仮定すると、観察系列は次のように表すことができます。

$$Y(t) = U(t) + wI_T(t)$$

この場合、$I_T(t) = \begin{cases} 0 & t \neq T \\ 1 & t = T \end{cases}$ はパルス関数であり、w は 外れ値によって生じる真の U(T) からの偏差です。

AO（技術革新的外れ値）

t=T の時に IO 外れ値が発生すると仮定すると、次のようになります。

$$Y(t) = \mu(t) + \frac{\theta(B)}{\Delta\varphi(B)}(a(t) + wI_T(t))$$

LS（レベル シフト）

t=T の時に LS 外れ値が発生すると仮定すると、次のようになります。

$$Y(t) = U(t) + wS_T(t)$$

この場合、$S_T(t) = \frac{1}{1-B}I_T(t) = \begin{cases} 0 & t < T \\ 1 & t \geq T \end{cases}$ はステップ関数です。

TC（一時的/過渡的変化）

t=T の時に TC 外れ値が発生すると仮定すると、次のようになります。

$$Y(t) = U(t) + wD_T(t)$$

この場合、$D_T(t) = \frac{1}{1-\delta B}I_T(t)$，$0 < \delta < 1$ は減衰関数です。

SA（季節性相加）

t=T の時に SA 外れ値が発生すると仮定すると、次のようになります。

$Y(t) = U(t) + wSS_T(t)$

この場合、$SS_T(t) = \frac{1}{1-B^s} I_T(t) = \begin{cases} 1 & t = T + ks, k \geq 0 \\ 0 & o.w. \end{cases}$ はステップ季節性パルス関数です。

LT（ローカル トレンド）

t=T の時に LT 外れ値が発生すると仮定すると、次のようになります。

$Y(t) = U(t) + wT_T(t)$

この場合、$T_T(t) = \frac{1}{(1-B)^2} I_T(t) = \begin{cases} t+1-T & t \geq T \\ 0 & o.w. \end{cases}$ はローカル トレンド関数です。

AOP（AO パッチ）

AO パッチは二つ以上の連続する AO 外れ値のグループです。AO パッチは開始時間と長さで記述できます。t=T の時に k の長さの AO 外れ値のパッチがあると仮定すると、観測系列は次のように表されます。

$$Y(t) = U(t) + \sum_{i=1}^{k} w_i I_{T-1+i}(t)$$

マスク効果により、外れ値を一つずつ検索している時は、AO 外れ値のパッチは検出するのが困難になります。このような理由により、AO パッチは個々の AO とは別個のタイプとみなされます。AO パッチのタイプについて、手順はパッチ全体をまとめて検索します。

要約

t=T の時のタイプ 0 の外れ値の場合（AO パッチを除く）:

$Y(t) = \mu(t) + wL_O(B) I_T(t) + \frac{\theta(B)}{\Delta\varphi(B)} a(t)$

この場合、

$$L_O(B) = \begin{cases} 1 & O = AO \\ 1/(\Delta\pi(B)) & O = IO \\ 1/(1-B) & O = LS \\ 1/(1-\delta B) & O = TC \\ 1/(1-B^s) & O = SA \\ 1/(1-B)^2 & O = LT \end{cases}$$

ここでは、$\pi(B) = \varphi(B)/\theta(B)$ とします。したがって、結合外れ値の一般的なモデルは以下のように記述できます。

$$Y(t) = \mu(t) + \sum_{k=1}^{M} w_k L_{O_k}(B) I_{T_{D,k}}(t) + \frac{\theta(B)}{\Delta\varphi(B)} a(t)$$

この場合、M は外れ値の数です。

外れ値の効果の推定

モデルおよびモデル パラメータはわかっていると仮定します。さらに、外れ値のタイプと場所もわかっていると仮定します。外れ値の絶対値の推定と検定統計量は以下のようになります。

このセクションの結果は、外れ値検出手順の中間ステップでのみ使用します。外れ値の最終推定は、すべてのパラメータが連係して推定されるすべての外れ値を組み込んでいるモデルから得られます。

非 AO パッチ決定論的外れ値

T の時の何らかのタイプの決定論的外れ値については（AO パッチを除く）、$e(t)$ を残差とし、および $x(t) = \pi(B) L(B) \Delta I_T(t)$ とします。したがって、

$$e(t) = wx(t) + a(t)$$

残差 e(t)から、T の時の外れ値のパラメータは x(t) に対する e(t) の単純な線型回帰で推定します。

j = 1 (AO)、2 (IO)、3 (LS)、4 (TC)、5 (SA)、6 (LT) の場合、検定統計量は以下のように定義します。

$$\lambda_j(\mathsf{T}) = \frac{w_j(T)}{\sqrt{\mathsf{Var}(w_j(T))}}$$

外れ値の帰無仮説では、モデルおよびモデル パラメータがわかっていると想定すると、$\lambda_j(\mathsf{T})$ は N(0,1) として分散します。

AO パッチ外れ値

時間 T に開始する長さ k のAO パッチの場合、i = 1 to k なので、$x_i(t;T) = \pi(B) \Delta I_{T+i-1}(t)$ とすると、以下のようになります。

$$e(t) = \sum_{i=1}^{k} w_i(T) x_i(t;T) + a(t)$$

多重線型回帰を使用してこのモデルを適合させます。検定統計量は以下のように定義されます。

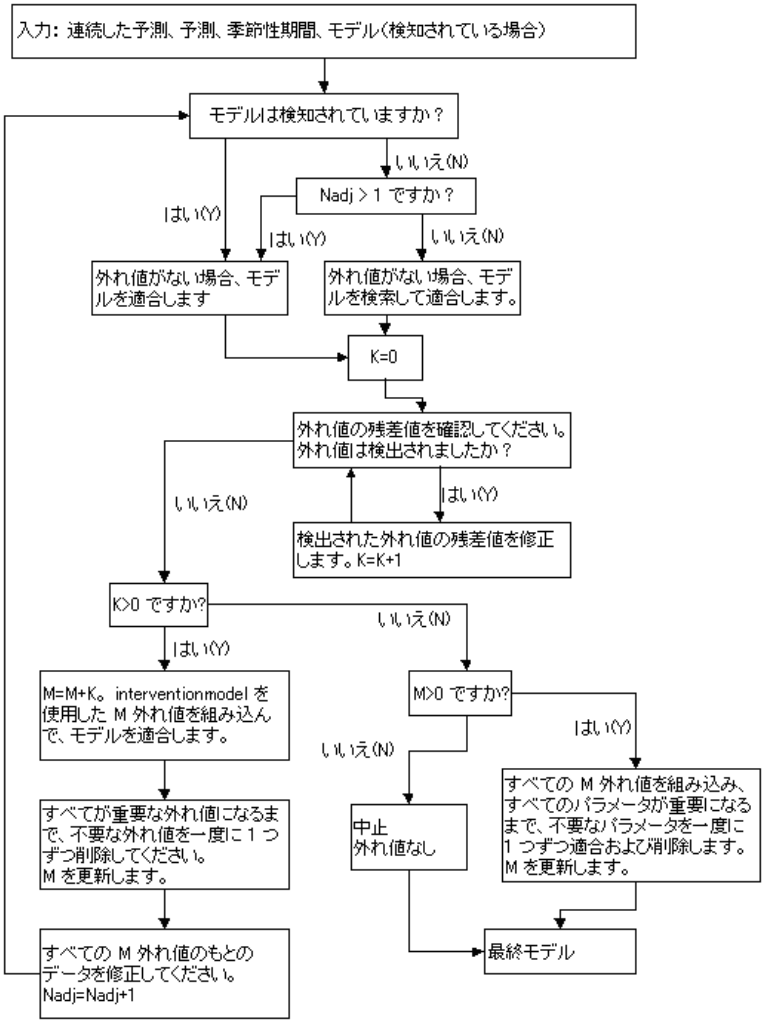
$$\chi^2(T) = \frac{\mathbf{w}'(T)(X_T X_T)\mathbf{w}(T)}{\sigma^2}$$

モデルおよびモデル パラメータがわかっていると仮定すると、帰無仮説 $w_1(T) = \cdots = w_k(T) = 0$ では、$\chi^2(T)$ は自由度 k のカイ 2 乗分散になります。

外れ値の検出

下記のフロー チャートは、自動外れ値検出の仕組みを図示しています。M を外れ値の総数、Nadj を外れ値について系列を調整する回数とします。手順の開始時は、M = 0 および Nadj = 0 です。

図 33-1



適合度統計

適合度統計は原系列 Y(t) に基づきます。k = モデルにおけるパラメータ数、n = 非欠損 残差数とします。

平方平均誤差

$$MSE = \frac{\Sigma\left(Y(t) - \hat{Y}(t)\right)^2}{n-k}$$

絶対平均誤差率

$$MAPE = \frac{100}{n}\Sigma\left|\left(Y(t) - \hat{Y}(t)\right)/Y(t)\right|$$

絶対最大誤差率

$$MaxAPE = 100\max\left(\left|\left(Y(t) - \hat{Y}(t)\right)/Y(t)\right|\right)$$

絶対平均誤差

$$MAE = \frac{1}{n}\Sigma\left|Y(t) - \hat{Y}(t)\right|$$

絶対最大誤差

$$MaxAE = \max\left(\left|Y(t) - \hat{Y}(t)\right|\right)$$

正規化ベイジアン情報量基準

正規化 $BIC = \ln(MSE) + k\frac{\ln(n)}{n}$

R-Squared

$$R^2 = 1 - \frac{\Sigma\left(Y(t) - \hat{Y}(t)\right)^2}{\Sigma\left(Y(t) - \overline{Y}\right)^2}$$

固定 R-Squared

同種の統計が Harvey (Harvey, 1989)によって使用されました。

$$R_S^2 = 1 - \frac{\sum_t \left(Z(t) - \hat{Z}(t)\right)^2}{\sum_t \left(\Delta Z(t) - \overline{\Delta Z}\right)^2}$$

この場合、

合計は、$Z(t) - \hat{Z}(t)$ および $\Delta Z(t) - \overline{\Delta Z}$ のどちらも欠損していない項を上回ります。

$\overline{\Delta Z}$ は差分変換系列に対する単純平均モデルであり、単変量ベースライン モデル ARIMA(0,d,0)(0,D,0) と同じです。

現在検討中の指数平滑化モデルについては、差分順序（もしあれば同等の ARIMA モデルに対応）を使用します。

$$d = \begin{cases} 2 & \text{Brown, Holt} \\ 1 & \text{other} \end{cases}, \quad D = \begin{cases} 0 & s = 1 \\ 1 & s > 1 \end{cases}$$

注：(−∞,1] の範囲では、固定 R-squared および通常 R-squared はどちらも負になることがあります。負の R-squared の値は、検討中のモデルがベースライン モデルよりも劣るということを意味します。ゼロの R-squared は、検討中のモデルがベースライン モデルと同程度であるということを意味します。正の R-squared の値は、検討中のモデルがベースライン モデルよりも優れているということを意味します。

エキスパート モデリング

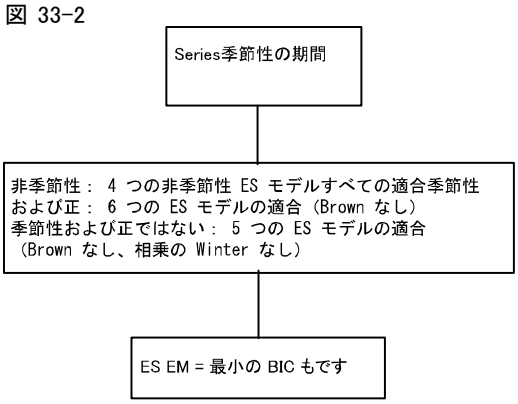
単変量系列

ユーザーは、エキスパート モデラーを使用して下記からモデルを選択できます。

- すべてのモデル（デフォルト）。
- 指数平滑化モデルのみ。
- ARIMA モデルのみ。

指数平滑化エキスパート モデル

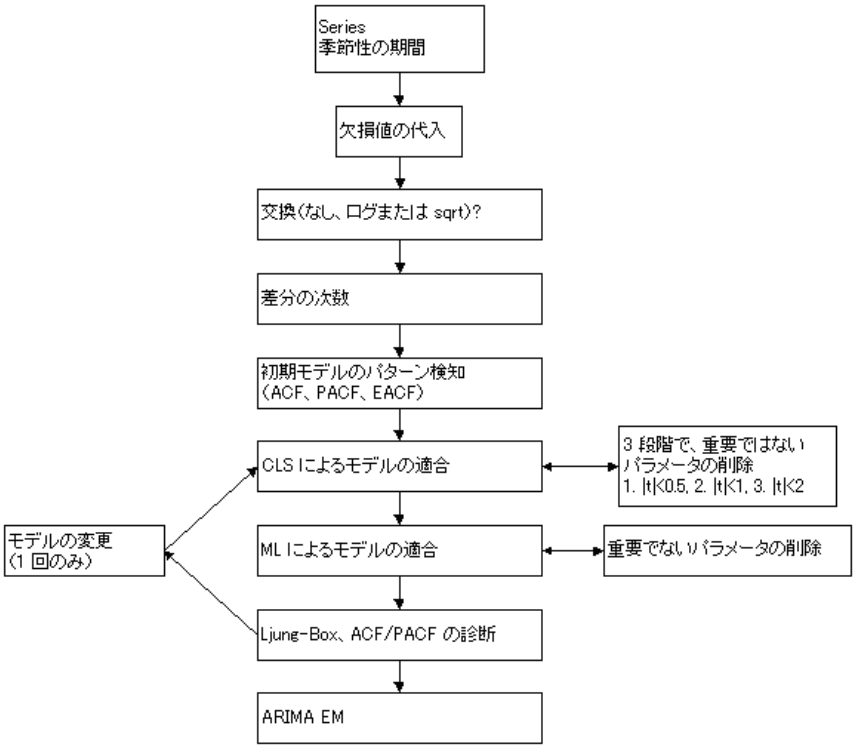
図 33-2



注：1<n≤10 という短い系列については、ESを適合させます。

ARIMA エキスパート モデル

図 33-3



注：短い系列については、以下のようにします。

- n<=10 ならば、定数項で AR(1) を適合させます。
- 10<n<3s の場合は、s=1 を設定して非季節性モデルを構築します。

すべての モデルのエキスパート モデル

この場合、指数平滑化モデルと ARIMA モデルが計算され、小さな正規化 BIC を備えているモデルが選択されます。

注：n<max(20,3s) の短い系列については、指数平滑化エキスパート モデル p. 333 を使用します。

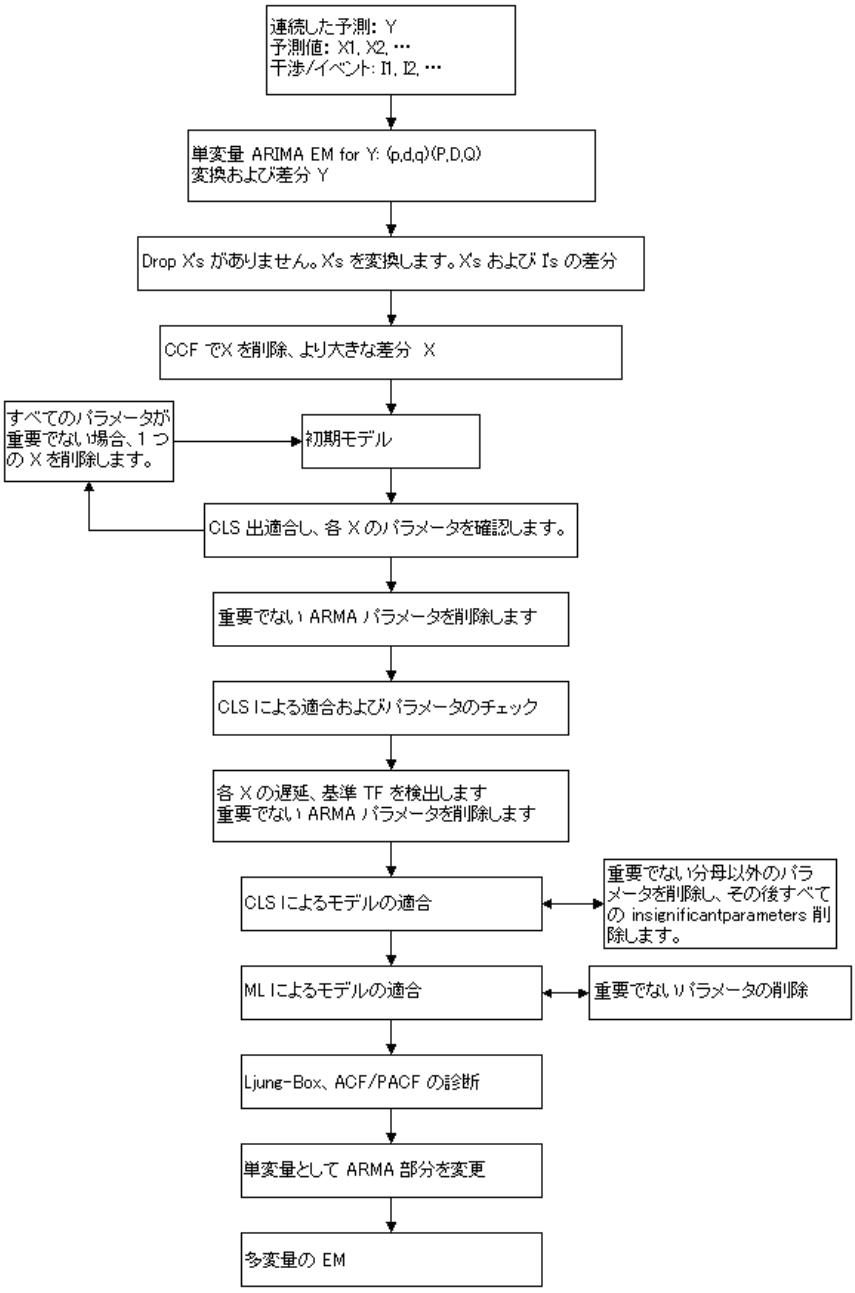
多変量系列

多変量状況では、ユーザーはエキスパート モデラーを使用して下記からモデルを選択できます。

- すべてのモデル（デフォルト）。多変量エキスパート ARIMA モデルがすべての予測値を削除して単変量エキスパート ARIMA モデルになってしまう場合、この単変量エキスパート ARIMA モデルはそれまでと同様にエキスパート指数平滑化モデルと比較され、エキスパート モデラーはどれが最良の包括モデルかを決定します。
- ARIMA モデルのみ。

転送関数エキスパート モデル

図 33-4



注：n<max(20,3s) の短い系列については、単変量エキスパート モデルを適合させます。

空白の処理

一般的には、系列データの欠損値は、時系列モデリング用のデータを作成するのに使用する時間間隔ノードにおいて代入されます。モデリング ノードに送信される系列データに空白が残っている場合、ARIMA/TF の推定と予測 p.324 で説明されているように、ARIMA モデルは値を代入しようと試みます。

予測値の欠損値は、時系列モデルから除外される欠損値を含んでいるフィールドに表示されます。

生成されたモデル／スコアリング

時系列モデルの予測値または予測は、モデリング処理そのものに複雑に関連しています。予測計算は対応するモデル タイプのアルゴリズムで記述されます。指数平滑化モデルでの予測は、指数平滑化モデル p.320 を参照してください。ARIMA モデルでの予測は、ARIMA/TF の推定と予測 p.324 を参照してください。

空白の処理

生成されるモデルの空白の処理は、モデリング ノードの空白の処理によく似ています。

予測期間中に予測値に欠損値が生じると、手続きによって警告が出され、可能な限り予測が行われます。

参照

Box, G. E. P., G. M. Jenkins, および G. C. Reinsel. 1994. Time series analysis:Forecasting and control, 3rd ed. Englewood Cliffs, N.J.: Prentice Hall.

Brockwell, P. J., および R. A. Davis. 1991. Time Series:Theory and Methods, 2 ed. : Springer-Verlag.

Gardner, E. S. 1985. Exponential smoothing:The state of the art. Journal of Forecasting, 4, 1-28.

Harvey, A. C. 1989. Forecasting, structural time series models and the Kalman filter. Cambridge: Cambridge University Press.

Makridakis, S. G., S. C. Wheelwright, および R. J. Hyndman. 1997. Forecasting:Methods and applications, 3rd ed. ed. New York: John Wiley and Sons.

Melard, G. 1984. A fast algorithm for the exact likelihood of autoregressive-moving average models. Applied Statistics, 33:1, 104-119.

Pena, D., G. C. Tiao, および R. S. Tsay、編集者. 2001. A course in time series analysis. New York: John Wiley and Sons.

章
34

TwoStep クラスタ アルゴリズム

概要

TwoStep クラスタ手法は、大規模なデータ セットを処理するために設計された、測定可能なクラスタ分析アルゴリズムです。このアルゴリズムでは、連続変数または属性データとカテゴリ変数または属性データの両方を取り扱うことができます。データ パスは 1 つしか必要ありません。このアルゴリズムは、次の 2 つのステップから成り立っています。1) ケース（またはレコード）を多数の小さいサブクラスタに予備クラスタ化する、および 2) 予備（事前）クラスタ化処理で作成されたサブクラスタを目的の数のクラスタにクラスタ化する。また、自動的にクラスタ数を選択することもできます。

モデル パラメータ

名前が示すように、TwoStep クラスタリング アルゴリズムには、予備クラスタリングとクラスタリングの 2 つのステップがあります。

予備クラスタリング

予備クラスタリング ステップでは、順次的なクラスタリング手法が用いられます。データ レコードが 1 つずつ順番に調べられ、現在のレコードを前に作成されたクラスタにマージするか、または距離基準（後述）に基づいて新しいクラスタを開始するかが判断されます。

このアルゴリズムの処理は、改良されたクラスタ特性（CF）ツリーを構築することにより行われます。CF ツリーは、ノードのレベルから成り立っており、ノードには複数のエントリが存在しています。葉エントリ（葉ノード中のエントリ）は、最終サブクラスタを表します。葉ノード以外のノード、およびそのエントリは、新しいレコードを正しい葉ノードに迅速に配置するためのガイドとして用いられます。各エントリは、エントリのレコード数、各範囲型フィールドの平均と分散、および各シンボル型フィールドのそれぞれのカテゴリの度数から成り立つ CF により特徴付けられています。ルート ノードから始まる連続した各レコードは、もっとも近い子ノードを探すために、ノード中のもっとも近いエントリにより再帰的にガイドされながら、CF ツリーを下っていきます。葉ノードに達したら、葉ノード中のもっとも近い葉エントリが探されます。レコードがもっとも近い葉エントリの閾値距離内にある場合、そのレコードは葉エントリに取り込まれ、その葉エントリの CF が更新されます。それ以外の場合は、葉ノード中に独自の葉エントリが作成されます。葉ノードに新しい葉エントリを作成する余地がない場合は、その葉ノードが 2 つに分割されます。元の葉ノード中のエントリは、もっとも遠い組み合わせをシードとして使用して、2 つのグループに分割されます。

CF ツリーが許可されている最大サイズを超えて成長する場合、既存の CF ツリーに基づいて閾値距離基準を増やし、その CF ツリーを再構築します。再構築される CF ツリーは小さくなるため、新しい入力レコードを取り込むスペースができます。完全なデータ パスの処理が完了するまで、この処理が続行されます。CF ツリー構築の詳細は、「BIRCH アルゴリズム(Zhang, Ramakrishnon, および Livny, 1996)」を参照してください。

すべてのレコードが同じエントリに該当する場合、総称してエントリの CF として表すことができます。エントリに新しいレコードが追加されると、この新規レコードおよび古い CF から、エントリ中の個別のレコードを確認せずに新しい CF を計算できます。これらの CF の性質により、個別のレコード セットを管理する代わりに、エントリ CF だけを管理することが可能になります。CF ツリーは元のデータより大幅に小さいため、メモリー中に効率的に格納することができます。

構築された CF ツリーの構造は、ケースまたはレコードの入力順序によって異なることに注意してください。順序による影響を最小限に抑えるには、モデル構築前にレコードを無作為に並べるようにしてください。

クラスタ

クラスタリング ステップでは、予備クラスタリング ステップで得られたサブクラスタ（外れ値の処理を使用している場合は非外れ値サブクラスタ）を入力として使用し、それらを目的の数のクラスタにグループ化します。サブクラスタ数は元のレコード数より大幅に少ないため、従来のクラスタリング手法を効果的に使用することができます。TwoStep は、自動クラスタリング手法（後述する「自動クラスタリング」を参照）と相性がいい、凝集型階層クラスタリング手法を使用しています。

階層クラスタリングは、最終的にすべてのレコードを含む 1 つのクラスタが残るまで、クラスタが再帰的に結合される処理です。この処理は、予備クラスタリング ステップで生成された各サブクラスタを、開始クラスタとして定義することから始まります。(詳細は、 p.337 予備クラスタリング を参照してください。) 次にすべてのクラスタが比較され、クラスタ間の距離がもっとも小さいクラスタのペアが結合されて、1 つのクラスタになります。結合が行われた後で、新たなクラスタ セットが比較され、最も近いペアが結合されます。すべてのクラスタが結合されるまで、この処理が繰り返されます。(ディシジョン ツリーが構築される方法を理解している場合、これは同様の処理で、元に戻ることはありません。)クラスタは再帰的に結合されるため、クラスタ数が異なる解との比較が容易です。たとえば、5 クラスタの解を取得するには、5 クラスタが残った時点で結合を停止します。4 クラスタの解を取得するには、5 クラスタの解を利用して、もう 1 回結合操作を実施します。

距離測度

TwoStep クラスタリングでは、シンボル値フィールドと範囲型フィールドの両方に対応するために、対数尤度距離測度を使用しています。これは、確率に基づいた距離です。2 つのクラスタ間の距離は、対数尤度の減少に関連しています（1 つのクラスタにまとめられるため)。対数尤度の計算時には、範囲型フィールドの場合正規分布が、シンボル値フィールドの場合多項分布が仮定されます。また、フィールドは互いに独立しており、またレコードも独立していると仮定されます。クラスタ i と j 間の距離は、次のように定義されます。

$$d(i,j) = \xi_i + \xi_j - \xi_{\langle i,j \rangle}$$

ここで、

$$\xi_v = -N_v \left(\sum_{k=1}^{K^A} \frac{1}{2} \log \left(\hat{\sigma}_k^2 + \hat{\sigma}_{vk}^2 \right) + \sum_{k=1}^{K^B} \hat{E}_{vk} \right)$$

および

$$\hat{E}_{vk} = -\sum_{l=1}^{L_k} \frac{N_{vkl}}{N_v} \log \frac{N_{vkl}}{N_v}$$

これらの式において、

K^A は範囲型入力フィールド数を示します。

K^B はシンボル値入力フィールド数を示します。

L_k は k 番目のシンボル値フィールドのカテゴリ数を示します。

N_v はクラスタ v のレコード数を示します。

N_{vkl} は k 番目のシンボル値フィールドの l 番目のカテゴリに所属するクラスタ v 中のレコード数を示します。

$\hat{\sigma}_k^2$ はすべてのレコードに対する k 番目の連続変数の推定された分散を表しています。

$\hat{\sigma}_{vk}^2$ は v 番目のクラスタ中のレコードに対する k 番目の連続変数の推定された分散を表します。

〈 i, j 〉は、結合するクラスタ i と j により生成されるクラスタを示すインデックスを表します。

ξ_v の式中で $\hat{\sigma}_k^2$ が無視された場合、クラスタ i と j 間の距離は、2 つのクラスタの結合時に正確に対数尤度で減少します。$\hat{\sigma}_{vk}^2 = 0$ により生じた問題を解決するために、$\hat{\sigma}_k^2$ 項が追加されます。これにより、自然対数が未定義になります（これは、たとえばクラスタが 1 つのケースしか持たない場合などに発生します）。

クラスタ数（自動クラスタリング）

TwoStep の 2 番目のステップで階層クラスタリング手法を使用して複数のクラスタの解を評価し、入力データに最適なクラスタ数を自動的に判断することができます。階層クラスタリングの特徴は、1 回の実行でパーティション（分割）の並びを生成できることにあります（1、2、3、… クラスタ）。対照的に K-Means アルゴリズムでは、この並びを生成するために複数回実行する必要があります（各クラスタ数ご

とに 1 回ずつ)。クラスタ数を自動的に決定するために、TwoStep では階層クラスタリング手法に適した 2 段階の処理を使用しています。最初の段階では、指定された範囲内の各クラスタ数の BIC が計算され、クラスタ数の初期推定値を探すために用いられます。BIC は次のように計算されます。

$$BIC(J) = -2\sum_{j=1}^{J}\xi_j + m_J \log(N)$$

この場合、

$$m_J = J\left\{2K^A + \sum_{k=1}^{K^B}(L_K - 1)\right\}$$

また、他の項は 距離測度 のように定義されます。最初の結合と相対的な後続の各結合における BIC の変化の比率が、初期推定値を決定します。$dBIC(J)$ を、J クラスタを持つモデルと (J + 1) クラスタを持つモデル間の差異とすると、$dBIC(J) = BIC(J) - BIC(J+1)$ となります。この場合、モデル J の変化率は次のようになります。

$$R_1(J) = \frac{dBIC(J)}{dBIC(1)}$$

$dBIC(1) < 0$ ならば、クラスタ数は 1 に設定されます (この場合、第 2 段階は省略されます)。それ以外の場合は、クラスタ数 k の初期推定値は、$R_1(J) < 0.04$ の最小値となります。

第 2 段階では、各階層クラスタリング ステージ中の 2 つの近接するクラスタ間の距離の、最大の相対増加を探すことにより、初期推定が調整されます。この処理は、次のように行われます。

- BIC 基準が示すモデル C_k から開始して、そのモデルおよび次に大きいモデル (階層クラスタリング処理における前のモデル) C_{k+1} のクラスタ間の最小距離の比率を取得します。

$$R_2(k) = \frac{d_{\min}(C_k)}{d_{\min}(C_{k+1})}$$

ここで C_k は k 個のクラスタを含むクラスタ モデルを、$d_{min}(C)$ はクラスタ モデル C の最小クラスタ間距離を表します。

- 次にモデル C_{k-1} から前述のように、次のモデル C_k との比率を計算します。比率が $R_2(2)$ になるまで、以降のモデルに対してこの処理を繰り返します。
- もっとも大きい 2 つの R_2 比率を比較します。最大の比率が 2 番目の比率の 1.15 倍を超えている場合、最大の R_2 比率を持つモデルを最適なクラスタ数として選択します。それ以外の場合は、最大の R_2 値を持つ 2 つのモデルから、クラスタ数の多いモデルを最適なモデルとして選択します。

空白の処理

TwoStep クラスタ ノードは、空白をサポートしていません。空白値、ヌル、または欠損値を持つレコードは、モデルの構築から除外されます。

オプションの効果

外れ値の処理

CF ツリーの構築処理アルゴリズムには、オプションの外れ値の処理が実装されています。外れ値は、どのクラスタにも適合しないデータ レコードとみなされます。エントリ中のレコード数が、CF ツリー中の最大の葉エントリのサイズに対して一定の割合（デフォルトは 25 %）未満の場合、その葉エントリ中のデータ レコードは外れ値とみなされます。CF ツリーを再構築する前に、外れ値候補がチェックされ、それらが除外されます。CF ツリーを再構築したら、ツリー サイズを増やさずにこれらの外れ値を適合させることができるかどうかがチェックされます。CF ツリー構築の最後に、適合できなかった小さいエントリが外れ値になります。

生成されたモデル/スコアリング

予測値

生成された TwoStep クラスタ モデルでレコードをスコアリングする場合、レコードはもっとも近いクラスタに割り当てられます。レコードと各クラスタ間の距離が計算され、もっとも小さい距離のクラスタがもっとも近いクラスタとして選択されます。このクラスタが、レコードの予測フィールドとして割り当てられます。距離は、モデル構築時と同じ方法で計算されます。レコードは 1 つのレコードだけを持つ「クラスタ」としてスコアリングされます。 詳細は、 p.338 距離測度 を参照してください。

モデル構築時に外れ値の処理が有効にされた場合、レコードとそれにもっとも近いクラスタ間の距離が閾値 C = log(V) と比較されます。

$$V = \prod_k R_k \prod_m L_m?$$

ここで R_k は k 番目の数値フィールドの範囲を、L_m は m 番目のシンボル値フィールドのカテゴリ数を表します。

もっとも近いクラスタからの距離が C より小さい場合は、そのクラスタをレコードの予測フィールドとして割り当てます。距離が C より大きい場合は、そのレコードを外れ値として割り当てます。

空白の処理

モデル構築と同様に、空白を持つレコードは処理されず、予測フィールド $null$ が割り当てられます。

章
35

予測値の重要度アルゴリズム

予測値の重要度は、重要度分析によって、各予測変数に起因する対象の分散の減少を計算することによって指定されます。この予測値の重要度の計算方法は、次のモデルで使用されます。

- ニューラル ネットワーク
- C5.0
- C&RT
- QUEST
- CHAID
- 回帰
- ロジスティック
- 判別分析
- 一般化線型
- SVM
- Bayesian Network

表記

この章では特に明記しない限り、次の表記を使用します。

Y	目標
X_j	j=1,...,k となる予測値
k	予測値の数
$Y = f(X_1, X_2, \ldots, X_k)$	X_k の予測値 X_1 に基づいた Y のモデル

分散ベースの方法

予測値は、次のように定義された重要度の測定にしたがってランク付けされます。

$$S_i = \frac{V_i}{V(Y)} = \frac{V(E(Y|X_i))}{V(Y)}$$

ここで、V(Y) は、無条件の出力分散となります。分子では、期待値の演算子 E はX_{-i} の積分、つまり X_i 以外のすべての因子を要求し、分散演算子 V は、X_i の高度な積分を意味します。

予測値の重要度は、正規化された重要度として計算されます。

$$VI_i = \frac{S_i}{\sum_{j=1}^{k} S_j}$$

Saltelli et al (2004) は、S_i が、予測値間の相互作用と非直行化の組み合わせの重要度の順番に予測値をランク付けするための、適切な重要度の測定方法であると示しています。

重要度の測定 S_i は、一次重要度測定で、一連の入力因子（X_1、X_2、…、X_k）が直交/独立（因子のプロパティ）でモデルが相加的である場合に正確です。つまり、モデルには入力因子間の交互作用（モデルのプロパティ）は含まれません。因子間で交互作用および非直交を組み合わせる場合、Saltelli (2004) は、S_i が重要度の順に入力因子の順位を付けるのに適切な重要度測定ですが、交互作用または/および非直交の有無により不正確な結果となるリスクがあることを指摘しています。S_i のより正確な推定を行うために、データセットのサイズを少なくとも数百にする必要があります。そうでない場合、S_i がかなり偏る場合があります。この場合、重要度測定をブートストラップで改善することができます。

計算

直交の場合、次のようにモンテカルロ法で入力因子の空間で多次元の積分を計算することによって、条件分散 V_i を推定することは簡単です。

まず、2 つのサンプル入力行列 $\mathbf{M}_1$ および $\mathbf{M}_2$ から始めましょう。それぞれの次元は N× k です。

$$\mathbf{M}_1 = \begin{matrix} x_1^{(1)} & x_2^{(1)} & \dots & x_k^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_k^{(2)} \\ \dots & \dots & \dots & \dots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_k^{(N)} \end{matrix}$$

そして

$$\mathbf{M}_2 = \begin{matrix} x_1^{(1')} & x_2^{(1')} & \dots & x_k^{(1')} \\ x_1^{(2')} & x_2^{(2')} & \dots & x_k^{(2')} \\ \dots & \dots & \dots & \dots \\ x_1^{(N')} & x_2^{(N')} & \dots & x_k^{(N')} \end{matrix}$$

ここで、N は、モンテカルロ推定のサンプル サイズで、そのサイズは数百から数千となります。各行は、入力のサンプルです。$\mathbf{M}_1$ および $\mathbf{M}_2$ から、3 番目の行列 $\mathbf{N}_j$ を構築できます。

$$\mathbf{N}_j = \begin{matrix} x_1^{(1')} & x_2^{(1')} & \dots & x_j^{(1)} & \dots & x_k^{(1')} \\ x_1^{(2')} & x_2^{(2')} & \dots & x_j^{(2)} & \dots & x_k^{(2')} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_1^{(N')} & x_2^{(N')} & \dots & x_j^{(N)} & \dots & x_k^{(N')} \end{matrix}$$

$\mathbf{M}_1$ を「サンプル」行列として、$\mathbf{M}_2$ を「再サンプル」行列として、$\mathbf{N}_j$ を行列としてみなし、X_j 以外のすべての因子が最サンプリングされます。次の方程式では、これらの分散を取得する方法を説明します (Saltelli 2002)。「hat」は、数値予測を示します。

$$\hat{V}(Y) = \frac{1}{N-1}\sum_{r=1}^{N} f^2\left(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}\right) - \hat{E}^2(Y)$$

とします。ここで、

$$\hat{E}^2(Y) = \left[\frac{1}{N}\sum_{r=1}^{N} f\left(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}\right)\right]^2$$

$$\hat{V}(E(Y|X_j)) = \hat{U}_j - \hat{E}^2(Y)$$

とします。ここで、

$$\hat{U}_j = \frac{1}{N-1}\sum_{r=1}^{N} f\left(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}\right) f\left(x_1^{(r')}, x_2^{(r')}, \dots, , x_{(j-1)}^{(r')}, x_j^{(r)}, x_{(j+1)}^{(r')}, \dots, x_k^{(r')}\right)$$

そして

$$\hat{E}^2(Y) = \frac{1}{N}\sum_{r=1}^{N} f\left(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}\right) f\left(x_1^{(r')}, x_2^{(r')}, \dots, x_k^{(r')}\right)$$

対象が連続型の場合、分散と期待値の累積手順に従います。カテゴリ方対象の場合、累積手順は各カテゴリの Y 向けとなります。各入力因子について、S_i は、各カテゴリの Y の要素を持つベクトルです。S_i の要素の平均を、Y の i 番目の入力因子の重要度の推定として使用します。

収束 :スケーラビリティを改善するために、収束を確認するときにレコードおよび予測変数のサブセットを使用します。具体的に、収束は次の基準によって判断されます。

$$\bigcap_{i \in I} \frac{1}{D} \sum_{j=t-D+1}^{t} \frac{\left|S_i(j) - \overline{S}_i\right|}{\overline{S}_i} < \epsilon$$

ここで $I = \{i|S_i(t) > 1/num\}$, D=100 となり、関心の幅 $\overline{S}_i = \frac{1}{D} \sum_{j=t-D+1}^{t} S_i(j)$ を示し、$\epsilon = 0.005$ は、必要な相対誤差の平均を定義します。

この指定は、重要度の値が平均より高い、「よい」予測変数に焦点を当てます。

レコードの順序:この予測値の重要度の計算方法は、大きいデータセットに対して正確に計算するため望ましい方法ですが、結果はデータセット内のレコードの順序によって異なります。ただし、大きくて順序が無作為のデータセットの場合、予測値の重要度の結果が一貫したものになると期待できます。

リファレンス

Saltelli, A., S. Tarantola, F. , F. Campolongo, および M. Ratto. 2004. Sensitivity Analysis in Practice - A Guide to Assessing Scientific Models. : John Wiley.

Saltelli, A. 2002. Making best use of model evaluations to compute sensitivity indices. Computer Physics Communications, 145:2, 280-297.

付録

A

注意事項

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

次の文は、条項が法律と一致しないイギリスなどの国には適用されません。本出版物は、SPSS INC., AN IBM COMPANY によって提供され、明示的および暗黙的なあらゆる保証、制限されていない場合を除く商品性や特定の目的への適合性、および無違反に関する暗黙的な保証を含む意思表示と保証を放棄します。特定の取引では明示的または暗黙的な保証の免責が許可されないため、この文が適用されない場合があります。

この情報には、技術的な誤りや誤植を含まれる場合があります。本文では変更が定期的に行われます。これらの変更は本書の次の版に組み込まれます。SPSS は、本文書に記載された製品やプログラムは予告なしに改善または変更される場合があります。

この情報内にある SPSS 以外または IBM 以外の Web サイトに対する参照は、便宜上提供されたものであり、これらのWebサイトを推奨するものではありませ提供されます。これらの Web サイトの資料は、この SPSS 社製品の使用の一部ではなく、これらの Web サイトの使用は個人の責任によるものです。

IBM または SPSS に情報を送信すると、あなたに対する義務を負うことなく、適切とする方法でその情報を使用または配布する非独占的権利と IBM および SPSS 付与するものとします。

SPSS 以外の製品に関する情報は、これらの製品、公開された通知、公表されているソースの供給者から得たものです。SPSS は、それらの製品をテストしていません。また、SPSS 以外の製品に関連するパフォーマンスの正確性、互換性、またはs

ポの他の要求を確認することはできません。SPSS 以外の製品の機能に関する質問は、これらの製品の供給者にお問い合わせください。

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

この情報には、日常の業務で使用されているデータおよびレポートの例が含まれています。それらを可能な限り詳細に説明するために、例には個人、企業、ブランド、製品の名前が含まれます。これらの名前はすべて架空のものであり、実際の名前や住所に似ているものでも、まったくの偶然によるものです。

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

商標

IBM、IBM ロゴ、ibm.com は世界各国の四方に基づく IBM 社の登録商標です。IBM の商標の現在のリストは Web サイト http://www.ibm.com/legal/copytrade.shmtl を参照してください。

SPSS Inc., an IBM Company の SPSS の商標 は、世界各国の司法に基づく登録商標です。

参考文献

Aggarwal, C. C., および P. S. Yu. 1998. Online generation of association rules. In: Proceedings of the 14th International Conference on Data Engineering, Los Alamitos, Calif: IEEE Computer Society Press, 402-411.

Agrawal, R., および R. Srikant. 1994. Fast Algorithms for Mining Association Rules. In: Proceedings of the 20th International Conference on Very Large Databases, J. B. Bocca, M. Jarke, および C. Zaniolo、編集者. San Francisco: Morgan Kaufmann, 487-499.

Agrawal, R., および R. Srikant. 1995. Mining Sequential Patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, Los Alamitos, Calif.: IEEE Computer Society Press, 3-14.

Aitkin, M., D. Anderson, B. Francis, および J. Hinde. 1989. Statistical Modelling in GLIM. Oxford: Oxford Science Publications.

Albert, A., および J. A. Anderson. 1984. On the Existence of Maximum Likelihood Estimates in Logistic Regression Models. Biometrika, 71, 1-10.

Anderson, T. W. 1958. Introduction to multivariate statistical analysis. New York: John Wiley & Sons, Inc..

Arya, S., および D. M. Mount. 1993. Algorithms for fast vector quantization. In: Proceedings of the Data Compression Conference 1993, , 381-390.

Belsley, D. A., E. Kuh, および R. E. Welsch. 1980. Regression diagnostics:Identifying influential data and sources of collinearity. New York: John Wiley and Sons.

Biggs, D., B. de Ville, および E. Suen. 1991. A method of choosing multiway partitions for classification and decision trees. Journal of Applied Statistics, 18, 49-62.

Bishop, C. M. 1995. Neural Networks for Pattern Recognition, 3rd ed. Oxford: Oxford University Press.

Box, G. E. P., および D. R. Cox. 1964. An analysis of transformations. Journal of the Royal Statistical Society, Series B, 26, 211-246.

Box, G. E. P., G. M. Jenkins, および G. C. Reinsel. 1994. Time series analysis:Forecasting and control, 3rd ed. Englewood Cliffs, N.J.: Prentice Hall.

Breiman, L., J. H. Friedman(F), R. A. Olshen, および C. J. Stone. 1984. Classification and Regression Trees. New York: Chapman & Hall/CRC.

Breslow, N. E. 1974. Covariance analysis of censored survival data. Biometrics, 30, 89-99.

Brockwell, P. J., および R. A. Davis. 1991. Time Series:Theory and Methods, 2 ed. : Springer-Verlag.

Cain, K. C., および N. T. Lange. 1984. Approximate case influence for the proportional hazards regression model with censored data. Biometrics, 40, 493-499.

Cameron, A. C., および P. K. Trivedi. 1998. Regression Analysis of Count Data. Cambridge: Cambridge University Press.

Chang, C. C., および C. J. Lin. 2003. LIBSVM:A library for support vector machines.Technical Report. Taipei, Taiwan: Department of Computer Science, National Taiwan University.

Chow, C. K., および C. N. Liu. 1968. Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, 14, 462-467.

Cooley, W. W., および P. R. Lohnes. 1971. Multivariate data analysis. New York: John Wiley & Sons, Inc..

Cox, D. R. 1972. Regression models and life tables (with discussion). Journal of the Royal Statistical Society, Series B, 34, 187-220.

Cunningham, P., および S. J. Delaney. 2007. k-Nearest Neighbor Classifiers. Technical Report UCD-CSI-2007-4, School of Computer Science and Informatics, University College Dublin, Ireland, , - .

Dempster, A. P. 1969. Elements of Continuous Multivariate Analysis. Reading, MA: Addison-Wesley.

Diggle, P. J., P. Heagerty, K. Y. Liang, および S. L. Zeger. 2002. The analysis of Longitudinal Data, 2 ed. Oxford: Oxford University Press.

Dixon, W. J. 1973. BMD Biomedical computer programs. Los Angeles: University of California Press.

Dobson, A. J. 2002. An Introduction to Generalized Linear Models, 2 ed. Boca Raton, FL: Chapman & Hall/CRC.

Dong, J., C. Y. Suen, および A. Krzyzak. 2005. Fast SVM training algorithm with decomposition on very large data sets. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27, 603-618.

Dougherty, J., R. Kohavi, および M. Sahami. 1995. Supervised and unsupervised discretization of continuous features. In: Proceedings of the Twelfth International Conference on Machine Learning, Los Altos, CA: Morgan Kaufmann, 194-202.

Drucker, H. 1997. Improving regressor using boosting techniques. In: Proceedings of the 14th International Conferences on Machine Learning, D. H. Fisher, Jr., ed. San Mateo, CA: Morgan Kaufmann, 107-115.

Dunn, P. K., および G. K. Smyth. 2005. Series Evaluation of Tweedie Exponential Dispersion Model Densities. Statistics and Computing, 15, 267-280.

Dunn, P. K., および G. K. Smyth. 2001. Tweedie Family Densities:Methods of Evaluation. In: Proceedings of the 16th International Workshop on Statistical Modelling, Odense, Denmark: .

Fan, R. E., P. H. Chen, および C. J. Lin. 2005. Working set selection using the second order information for training SVM.Technical Report. Taipei, Taiwan: Department of Computer Science, National Taiwan University.

Fayyad, U., および K. Irani. 1993. Multi-interval discretization of continuous-value attributes for classification learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo, CA: Morgan Kaufmann, 1022-1027.

Fine, T. L. 1999. Feedforward Neural Network Methodology, 3rd ed. New York: Springer-Verlag.

Fox, J., および G. Monette. 1992. Generalized collinearity diagnostics. Journal of the American Statistical Association, 87, 178-183.

Fox, J. 1997. Applied Regression Analysis, Linear Models, and Related Methods. Thousand Oaks, CA: SAGE Publications, Inc..

Freund, Y., および R. E. Schapire. 1995. A decision theoretic generalization of on-line learning and an application to boosting. In: Computational Learning Theory:7 Second European Conference, EuroCOLT ’95, , 23-37.

Friedman(F), J. H., J. L. Bentley, および R. A. Finkel. 1977. An algorithm for finding best matches in logarithm expected time. ACM Transactions on Mathematical Software, 3, 209-226.

Friedman(F), N., D. Geiger, および M. Goldszmidt. 1997. Bayesian network classifiers. Machine Learning, 29, 131-163.

Gardner, E. S. 1985. Exponential smoothing:The state of the art. Journal of Forecasting, 4, 1-28.

Gill, J. 2000. Generalized Linear Models:A Unified Approach. Thousand Oaks, CA: Sage Publications.

Goodman, L. A. 1979. Simple models for the analysis of association in cross-classifications having ordered categories. Journal of the American Statistical Association, 74, 537-552.

Hardin, J. W., および J. M. Hilbe. 2003. Generalized Linear Models and Extension. Station, TX: Stata Press.

Hardin, J. W., および J. M. Hilbe. 2001. Generalized Estimating Equations. Boca Raton, FL: Chapman & Hall/CRC.

Harman, H. H. 1976. Modern Factor Analysis, 3rd ed. Chicago: University of Chicago Press.

Harvey, A. C. 1989. Forecasting, structural time series models and the Kalman filter. Cambridge: Cambridge University Press.

Haykin, S. 1998. Neural Networks:A Comprehensive Foundation, 2nd ed. New York: Macmillan College Publishing.

Heckerman, D. 1999. A Tutorial on Learning with Bayesian Networks. In: Learning in Graphical Models, M. I. Jordan, ed. Cambridge, MA: MIT Press, 301-354.

Hendrickson, A. E., および P. O. 白. 1964. Promax:a quick method for rotation to oblique simple structure. British Journal of Statistical Psychology, 17, 65-70.

Hidber, C. 1999. Online Association Rule Mining. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, New York: ACM Press, 145-156.

Horton, N. J., および S. R. Lipsitz. 1999. Review of Software to Fit Generalized Estimating Equation Regression Models. The American Statistician, 53, 160-169.

Hosmer, D. W., および S. Lemeshow. 2000. Applied Logistic Regression, 2nd ed. New York: John Wiley and Sons.

Huber, P. J. 1967. The Behavior of Maximum Likelihood Estimates under Nonstandard Conditions. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA: University of California Press, 221-233.

Jennrich, R. I., および P. F. Sampson. 1966. Rotation for simple loadings. Psychometrika, 31, 313-323.

Kalbfleisch, J. D., および R. L. Prentice. 2002. The statistical analysis of failure time data, 2 ed. New York: John Wiley & Sons, Inc.

Kass, G. 1980. An exploratory technique for investigating large quantities of categorical data. Applied Statistics, 29:2, 119-127.

Kaufman, L., および P. J. Rousseeuw. 1990. Finding groups in data:An introduction to cluster analysis. New York: John Wiley and Sons.

Kohavi, R., B. Becker, および D. Sommerfield. 1997. Improving Simple Bayes. In: Proceedings of the European Conference on Machine Learning, , 78-87.

Kohonen, T. 2001. Self-Organizing Maps, 3rd ed. New York: Springer-Verlag.

Lane, P. W., および J. A. Nelder. 1982. Analysis of Covariance and Standardization as Instances of Prediction. Biometrics, 38, 613-621.

Lawless, R. F. 1982. Statistical models and methods for lifetime data. New York: John Wiley & Sons, Inc..

Lawless, J. E. 1984. Negative Binomial and Mixed Poisson Regression. The Canadian Journal of Statistics, 15, 209-225.

Liang, K. Y., および S. L. Zeger. 1986. Longitudinal Data Analysis Using Generalized Linear Models. Biometrika, 73, 13-22.

Lipsitz, S. H., K. Kim, および L. Zhao. 1994. Analysis of Repeated Categorical Data Using Generalized Estimating Equations. Statistics in Medicine, 13, 1149-1163.

Liu, H., F. Hussain, C. L. Tan, および M. Dash. 2002. Discretization:An Enabling Technique. Data Mining and Knowledge Discovery, 6, 393-423.

Loh, W. Y., および Y. S. Shih. 1997. Split selection methods for classification trees. Statistica Sinica, 7, 815-840.

Makridakis, S. G., S. C. Wheelwright, および R. J. Hyndman. 1997. Forecasting:Methods and applications, 3rd ed. ed. New York: John Wiley and Sons.

McCullagh, P. 1983. Quasi-Likelihood Functions. Annals of Statistics, 11, 59-67.

McCullagh, P., および J. A. Nelder. 1989. Generalized Linear Models, 2nd ed. London: Chapman & Hall.

Melard, G. 1984. A fast algorithm for the exact likelihood of autoregressive-moving average models. Applied Statistics, 33:1, 104-119.

Miller, M. E., C. S. Davis, および J. R. Landis. 1993. The Analysis of Longitudinal Polytomous Data:Generalized Estimating Equations and Connections with Weighted Least Squares. Biometrics, 49, 1033-1044.

Nelder, J. A., および R. W. M. Wedderburn. 1972. Generalized Linear Models. Journal of the Royal Statistical Society Series A, 135, 370-384.

Neter, J., W. Wasserman, および M. H. Kutner. 1990. Applied Linear Statistical Models, 3rd ed. Homewood, Ill.: Irwin.

Pan, W. 2001. Akaike's Information Criterion in Generalized Estimating Equations. Biometrics, 57, 120-125.

Pena, D., G. C. Tiao, および R. S. Tsay、編集者. 2001. A course in time series analysis. New York: John Wiley and Sons.

Platt, J. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Advances in Large Margin Classifiers, A. J. Smola, P. Bartlett, B. Scholkopf, および D. Schuumans、編集者. Cambridge, MA: MIT Press, 61-74.

Pregibon, D. 1981. Logistic Regression Diagnostics. Annals of Statistics, 9, 705-724.

Prim, R. C. 1957. Shortest connection networks and some generalisations. Bell System Technical Journal, 36, 1389-1401.

Ripley, B. D. 1996. Pattern Recognition and Neural Networks. Cambridge: Cambridge University Press.

Rumelhart, D. E., J. L. McClelland, および . The PDP Research Group. 1986. Parallel Distributed Processing:Explorations in the Microstructure of Cognition. Vol. 1:Foundations. Cambridge, MA: MIT Press.

Saltelli, A., S. Tarantola, F. , F. Campolongo, および M. Ratto. 2004. Sensitivity Analysis in Practice - A Guide to Assessing Scientific Models. : John Wiley.

Saltelli, A. 2002. Making best use of model evaluations to compute sensitivity indices. Computer Physics Communications, 145:2, 280-297.

Schatzoff, M., R. Tsao, および S. Fienberg. 1968. Efficient computing of all possible regressions. Technometrics, 10, 769-779.

Smyth, G. K., および B. Jorgensen. 2002. Fitting Tweedie's Compound Poisson Model to Insurance Claims Data:Dispersion Modelling. ASTIN Bulletin, 32, 143-157.

Storer, B. E., および J. Crowley. 1985. A diagnostic for Cox regression and general conditional likelihoods. Journal of the American Statistical Association, 80, 139-147.

Tan, P., M. Steinbach, および V. Kumar. 2006. Introduction to Data Mining. : Addison-Wesley.

Tao, K. K. 1993. A closer look at the radial basis function (RBF) networks. In: Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers, A. Singh, ed. Los Alamitos, Calif.: IEEE Comput. Soc. Press, 401-405.

Tatsuoka, M. M. 1971. Multivariate analysis. New York: John Wiley & Sons, Inc..

Uykan, Z., C. Guzelis, M. E. Celebi, および H. N. Koivo. 2000. Analysis of input-output clustering for determining centers of RBFN. IEEE Transactions on Neural Networks, 11, 851-858.

Velleman, P. F., および R. E. Welsch. 1981. Efficient computing of regression diagnostics. American Statistician, 35, 234-242.

Williams, D. A. 1987. Generalized Linear Models Diagnostics Using the Deviance and Single Case Deletions. Applied Statistics, 36, 181-191.

Zeger, S. L., および K. Y. Liang. 1986. Longitudinal Data Analysis for Discrete and Continuous Outcomes. Biometrics, 42, 121-130.

Zhang, T., R. Ramakrishnon, および M. Livny. 1996. BIRCH:An efficient data clustering method for very large databases. In: Proceedings of the ACM SIGMOD Conference on Management of Data, Montreal, Canada: ACM, 103-114.

白, H. 1980. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. Econometrica, 48, 817-836.

索引