

データサイエンス関連基礎調査 WG の活動報告

AKUR8 藤田 卓君

ミュンヘン再保 鈴木 理史君

アクサ生命 高橋 智嗣君

大同生命 上妻 玲兒君

T&D ホールディングス 浅芝 良一君

藤田 セッションA - 1にお越しいただき、まことにありがとうございます。本セッションの司会を務める、データサイエンス関連基礎調査ワーキンググループの藤田卓と申します。よろしくお願いたします。

本セッションは二つのプレゼンテーションからなります。当ワーキンググループがアクチュアリー実務を念頭に置いて独自で研究を行った内容について、ご紹介させていただきます。各プレゼンテーションの後に、質疑応答の時間も設けておりますので、積極的なご質問をお願いいたします。

まず一つ目のプレゼンテーションに関しては、鈴木理史さんと高橋智嗣さんから発表いただきます。二つ目は、上妻玲兒さんと浅芝良一さんから発表いただきます。

それでは、早速ですが、一つ目のプレゼンテーションに移りたいと思います。よろしくお願いたします。

鈴木 一つ目のプレゼンテーションを担当します、鈴木理史と申します。本日は、アクサ生命の高橋智嗣さんと、私、鈴木から、プレゼンテーションをさせていただきます。

こちらの研究成果ですけれども、9月にベルギーで行われましたJoCoという国際アクチュアリー会のジョイント・コロキウムで発表したものです。本日、ご登壇されていませんが、早稲田大学の岩沢先生との共著論文です。ちなみに、会場でいただいたASTINバッジを、今日のはつけて発表させていただきます。(注：ASTINは、国際アクチュアリー会の学術セクションの1つ)

本日のアジェンダですが、スライドに表示の1から5のような流れになっております。

アジェンダ

I. イントロダクション

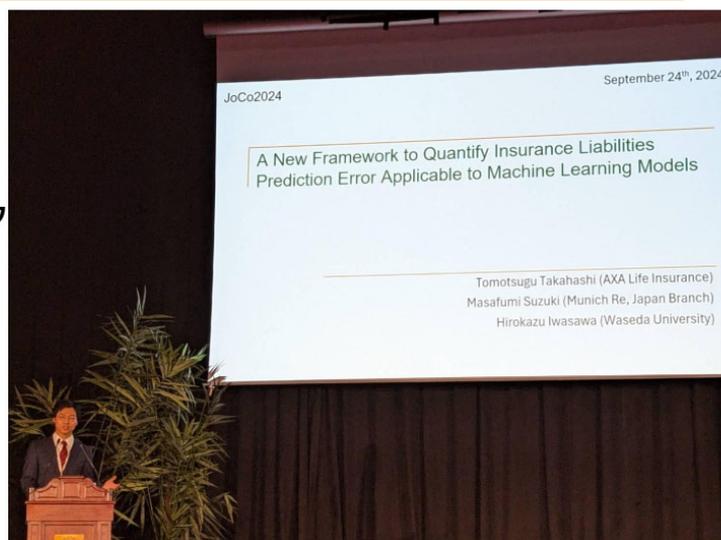
II. 予測誤差の定量化のフレームワーク

III. 推定方法

IV. 数値例

V. 結論

References



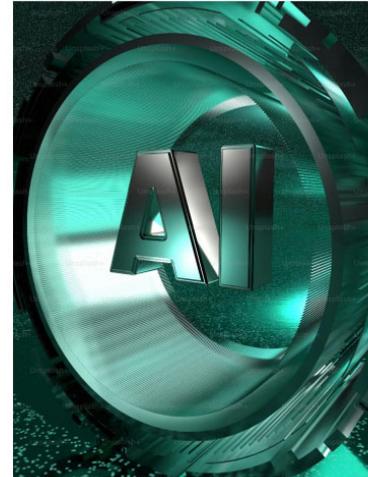
本研究成果は2024年9月にブリュッセル(ベルギー)で行われたJoCo2024で発表

まず、イントロダクションといたしまして、「アクチュアリーにとっての機械学習」ということで、お話しさせていただきます。

1. イントロダクション

アクチュアリーにとっての機械学習

- 機械学習の技術は絶えず進歩している。
- 生命保険の数理アサンプションは伝統的に、年齢や経過年数といった単純な説明変数に基づいていたが、機械学習手法により予測精度の向上が期待される。
- しかし、**機械学習の研究のほとんどは予測精度に焦点を当てており、予測誤差の分析に関する研究は十分に行われていない**（先行研究についてはAppendixを参照）。
- 本研究では、**機械学習手法に適用可能な、予測誤差を定量化する新しいフレームワークを提案**する。
- 予測誤差を定量化したうえでさらに、**プロセス誤差、パラメータ誤差、モデル誤差**に分解する。



3

ご存じのとおり、機械学習は絶えず進歩しております。一方、生命保険の数理アサンプションは、伝統的に年齢や経過年数といった単純な説明変数に基づいていましたが、機械学習手法の導入によって予測精度の向上が期待されております。

しかしながら、機械学習の研究のほとんどは予測精度に焦点を当てておりまして、われわれアクチュアリーが関心のある不確実性、すなわち予測誤差の分析に関する研究は十分に行われておりません。

本研究では、機械学習手法に適用可能な予測誤差を定量化する新しいフレームワークを提案します。また、その定量化した予測誤差を、プロセス誤差、パラメータ誤差、モデル誤差の三つに分解する方法を説明させていただきます。

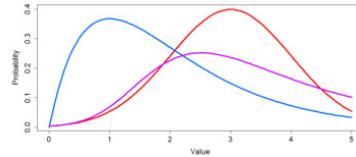
先ほど申し上げたモデル、パラメータ、プロセスについてですけれども、Actuarial Standard of Practice43 (A S O P 43) でも、不確実性の源泉として、モデルリスク、パラメータリスク、プロセスリスクの三つが挙げられておりまして、われわれアクチュアリーにとっても、なじみのある概念かと思いません。

I. イントロダクション

Actuarial Standards of Practice 43 (ASOP43)で述べられる不確実性の源泉（発表者訳）

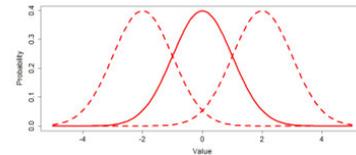
モデルリスク

- 手法が状況に適していない、またはモデルが特定の現象を表していないリスク



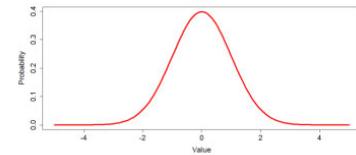
パラメータリスク

- 手法やモデルで使用されるパラメータが将来の結果を代表しないリスク



プロセスリスク

- パラメータが確実に分かっている場合でも、本質的に変動する将来の偶発事象の予測に関連するリスク



4

簡単に、これらの3つのリスクの概念をご説明させていただきます。まず、モデルリスクにつきましては、真の分布が赤い分布だったとして、それをパープルや青の分布で推定してしまうようなリスクです。続いて、パラメータリスクは、モデルは合っているけれども、そのパラメータ、例えば、この正規分布の平均などのパラメータが間違っていることによって、推定を誤るリスクです。プロセスリスクについては、モデルは正しいけれども、そのモデルの中に持つ確率的なぶれ、正規分布の中でのぶれなどによって生じるリスクです。この概念を、そのまま、機械学習手法に導入する方法について考えたのがわれわれの研究です。

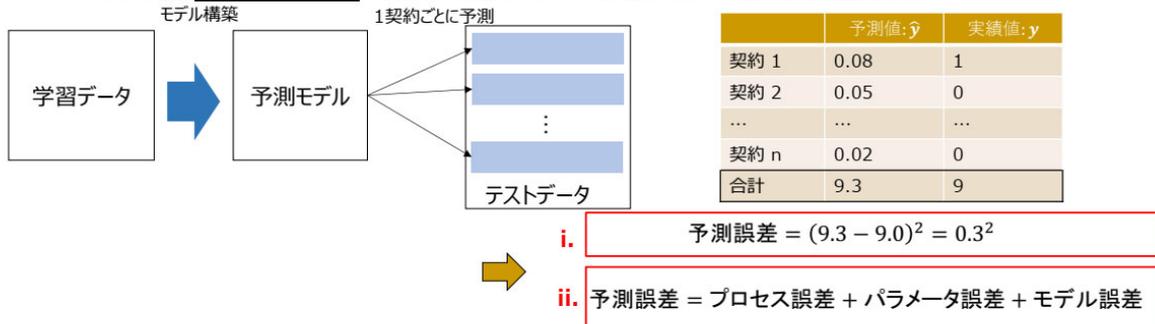
ちなみに、この分野に関する先行研究は、例えば、損害保険のロスリザービング（支払備金の見積もり）の分野で盛んに行われています。リスクをこの三つの要素に分解する手法も挙げられておりますが、機械学習にこの手法をそのまま適用することは、なかなか難しいです。

どういうことかと言いますと、機械学習の予測はある1点の値としての予測ですので、そもそも、確率分布は何なのかという話があり、不確実性を定量化することは難しいです。分布がわからない以上、この不確実性をプロセス・パラメータ・モデルに分解するのもなかなか難しいということになっております。われわれは、予測誤差の定量化およびプロセス・パラメータ・モデルへの分解を機械学習にも応用できるようなフレームワークを考えております。

II. 予測誤差の定量化のフレームワーク

本研究で提案するフレームワークの全体像

- 生命保険で扱う死亡/解約といった事象は、個別契約単位では0もしくは1の値しかとらない確率的な事象
- 生命保険ビジネスの管理は群団で行うため、1件単位ではなくポートフォリオ単位の予測誤差にこそ興味がある
- 我々のフレームワークの特徴は以下の2つ
 - 個別レコード単位ではなく**群団全体として誤差を捉え、分散効果を勘案**すること
 - その誤差を**3つの要素に分解**（プロセス誤差、パラメータ誤差、モデル誤差）して定量化すること



5

本フレームワークでの全体像を、ご説明させていただきます。生命保険で扱う死亡や解約などの事象は、個別契約単位では0もしくは1しか取らない確率的な事象です。例えば、スライド5ページの右側の方に表を載せておりますけれども、契約1の解約率が8%だと予測し、実績は1、すなわち解約したとします。この不確実性の源泉は何かと言うと、当然、確率的なぶれが主な要因ですので、われわれアクチュアリーがあまり興味あるものではありません。われわれが興味あるものは、むしろ、この契約群団を全部足して、全体として9.3件解約すると思っていたところ、実績は9件でしたという、このような群団単位の予測値にこそ興味があります。

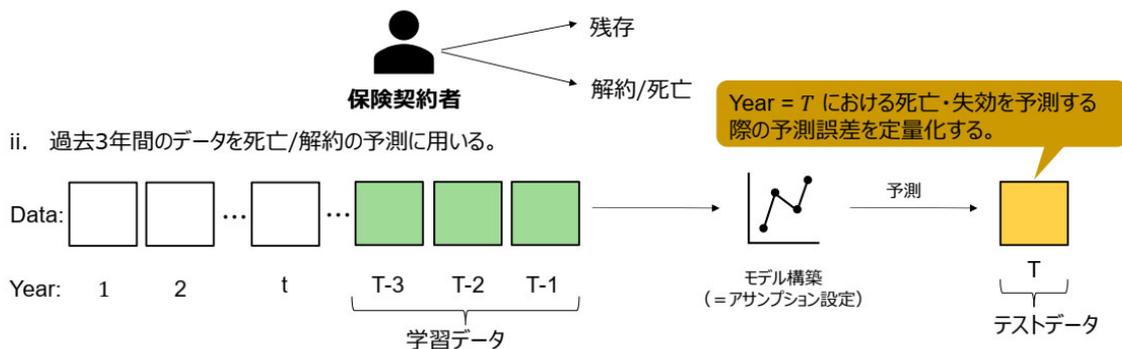
そこで、われわれのフレームワークでは、予測誤差を個別レコード単位ではなくて群団単位で捉えます。本研究は、群団単位で算出した予測誤差をプロセス誤差、パラメータ誤差、モデル誤差の三つに分解するというフレームワークです。

II. 予測誤差の定量化のフレームワーク

問題設定の詳細

本研究のフレームワークは、生命保険会社でよく使われるアサンプション実務に適用可能である。

- 目的変数は1（解約する/死亡する）もしくは0（解約しない/死亡しない）を取る。



6

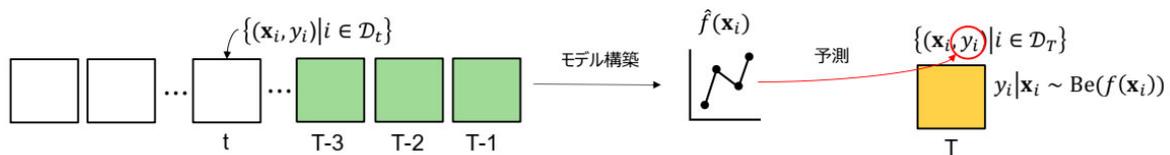
もう少し詳細に問題設定をご説明させていただきます。本研究でのフレームワークは生命保険会社によく使われるアサンプション実務に適用可能なものになっております。ここでの目的変数は、解約する、死亡するといった、1、もしくは、しないの0の二値の値を取ります。

また、アクチュアリーの実務に合わせまして、過去3年のデータを用います。データとしては、昔からビジネスをしていて何年も経験データがある中で、過去3年のデータを学習データとして用いて、モデルを構築、つまり、解約リスクや死亡リスクのアサンプションを設定します。予測するものは翌1年間の解約や死亡です。翌1年間の解約や死亡を予測する際の予測誤差を定量化します。

II. 予測誤差の定量化のフレームワーク

変数および関数の定義

記号	定義/説明
\mathbf{x}_i	契約者 <i>i</i> の特徴量ベクトル $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ 、既知の値
$f(\mathbf{x}_i)$	契約者 <i>i</i> の真の解約率
y_i	契約者 <i>i</i> の目的変数。1年後に残存するとき $y_i = 1$ 、解約するとき $y_i = 0$ y_i はパラメータ $f(\mathbf{x}_i)$ のベルヌーイ分布従う。 $y_i \mathbf{x}_i \sim \text{Be}(f(\mathbf{x}_i))$, $E[y_i \mathbf{x}_i] = f(\mathbf{x}_i)$
$\hat{f}(\mathbf{x}_i)$	契約者 <i>i</i> の解約率の予測値。この値は学習データに依存
$E[\hat{f}(\mathbf{x}_i)]$	学習データに対する解約率の予測値の期待値
w_i	契約者 <i>i</i> のウェイト、例えば年換算保険料や保険金額
D_t	Year = t におけるデータを表す集合



7

予測誤差の定量化のフレームワークにあたって、記号と関数の定義をご説明させていただきます。 x_i は、個別契約者の特徴量で、例えば、年齢や経過年数などの特徴量です。 f_{x_i} が真の解約率で、 y_i は、実際にその契約が解約したか、しないかであり、0か1の値を取るベルヌーイ分布に従います。

また、 \hat{f}_{x_i} は、解約率の予測、すなわちわれわれが作ったモデルでの予測値です。この予測値なのですが、学習データは有限ですので、学習データが少ないと予測がぶれます。そこで、 $E[\hat{f}_{x_i}]$ として予測値の期待値を導入します。これは、学習データに対する解約率の予測値の期待値でして、学習データが無限にある場合の予測値を意味しています。

w_i は、保険金額や年換算保険料のようなウェイトとお考えください。 D_t は、各年度のデータセットのことです。つまり、年度ごとにデータセット D_t がありまして、その中に契約者 i がいて、その個別の契約者が x_i の特徴量を持っていて、 y_i を予測するということです。

II. 予測誤差の定量化のフレームワーク

誤差の定義

予測誤差は、予測値と実際の値の差の二乗として定義される。

予測	1年間の解約量の予測	$\sum_{i \in \mathcal{D}_T} \hat{f}(\mathbf{x}_i) w_i$
実績	1年間の解約量の実績	$\sum_{i \in \mathcal{D}_T} y_i w_i$
誤差	予測と実績の差の二乗	$\left(\sum_{i \in \mathcal{D}_T} \hat{f}(\mathbf{x}_i) w_i - \sum_{i \in \mathcal{D}_T} y_i w_i \right)^2$

本研究においては、将来予測における**予想誤差を定量化**する。

- ✓ Year = T における将来のデータの条件付期待値
- ✓ $E[\dots]$ は y_i と学習データに対する期待値

$$\text{Err} := E \left[\left(\sum_{i \in \mathcal{D}_T} \hat{f}(\mathbf{x}_i) w_i - \sum_{i \in \mathcal{D}_T} y_i w_i \right)^2 \right]$$

8

ここで、われわれが定量化する誤差についてご説明させていただきます。予測するものは1年間の解約量です。つまり、個別契約の解約率×保険金額や解約率×年換算保険料を足した上げたものが解約量の予測値です。一方で、実績の解約量は、当然解約した契約の合計です。

誤差とは、この二つを引き算して、それを二乗したものを考えます。さらに、本研究においては、この二乗誤差がどのぐらい期待値として大きさを持つのか、すなわち予測誤差の期待値は大体どのぐらいあるのか。1年間でどれぐらい期待値としてぶれるのかを定量化します。

II. 予測誤差の定量化のフレームワーク

誤差分解

次のように変形することで、予測誤差 (Err) を3つの成分に分解できる。
本研究では、分解後の3つの項をパラメータ誤差、モデル誤差、プロセス誤差と呼ぶ。

$$\begin{aligned} \text{Err} &= E \left[\left(\sum_{i \in \mathcal{D}_T} (\hat{f}(\mathbf{x}_i) - y_i) w_i \right)^2 \middle| \{\mathbf{x}_i | i \in \mathcal{D}_T\} \right] \\ &= E \left[\left(\sum_{i \in \mathcal{D}_T} (\hat{f}(\mathbf{x}_i) - E[\hat{f}(\mathbf{x}_i)]) w_i + \sum_{i \in \mathcal{D}_T} (E[\hat{f}(\mathbf{x}_i)] - f(\mathbf{x}_i)) w_i + \sum_{i \in \mathcal{D}_T} (f(\mathbf{x}_i) - y_i) w_i \right)^2 \middle| \{\mathbf{x}_i | i \in \mathcal{D}_T\} \right] \\ &= E \left[\left(\sum_{i \in \mathcal{D}_T} (\hat{f}(\mathbf{x}_i) - E[\hat{f}(\mathbf{x}_i)]) w_i \right)^2 \right] + \left(\sum_{i \in \mathcal{D}_T} (E[\hat{f}(\mathbf{x}_i)] - f(\mathbf{x}_i)) w_i \right)^2 + E \left[\left(\sum_{i \in \mathcal{D}_T} (f(\mathbf{x}_i) - y_i) w_i \right)^2 \middle| \{\mathbf{x}_i | i \in \mathcal{D}_T\} \right] \end{aligned}$$

パラメータ誤差
モデル誤差
プロセス誤差

9

さらに、予測誤差を三つの要素に分解します。パラメータ誤差、モデル誤差、プロセス誤差の三つです。この式展開の詳細は、今日のご説明しないのですが、先ほど申し上げましたロスリザーピングの先行研究などでも使われているもので、簡単な仮定に基づいて分解することが可能となっております。

II. 予測誤差の定量化のフレームワーク 誤差分解とその解釈

予測誤差 = プロセス誤差 + パラメータ誤差 + モデル誤差

$y_i | x_i \sim \text{Be}(f(x_i))$ を用いることで簡単に得られる。

要素	式	別の表現	解釈
プロセス誤差	$E \left[\left(\sum_{i \in \mathcal{D}_T} (f(x_i) - y_i) w_i \right)^2 \middle \{x_i i \in \mathcal{D}_T\} \right]$	$= \sum_{i \in \mathcal{D}_T} f(x_i) (1 - f(x_i)) w_i^2$	ベルヌーイ分布に起因する内在誤差
パラメータ誤差	$E \left[\left(\sum_{i \in \mathcal{D}_T} (\hat{f}(x_i) - E[\hat{f}(x_i)]) w_i \right)^2 \right]$	$= V \left[\sum_{i \in \mathcal{D}_T} \hat{f}(x_i) w_i \right]$	モデルの予測値の分散
モデル誤差	$\left(\sum_{i \in \mathcal{D}_T} (E[\hat{f}(x_i)] - f(x_i)) w_i \right)^2$		モデルのバイアス

- 各誤差の分散効果の振る舞い（詳細はAppendix）
 - プロセス誤差：ポートフォリオの規模を増やせば減少
 - パラメータ誤差、モデル誤差：プロセス誤差ほどは減らない

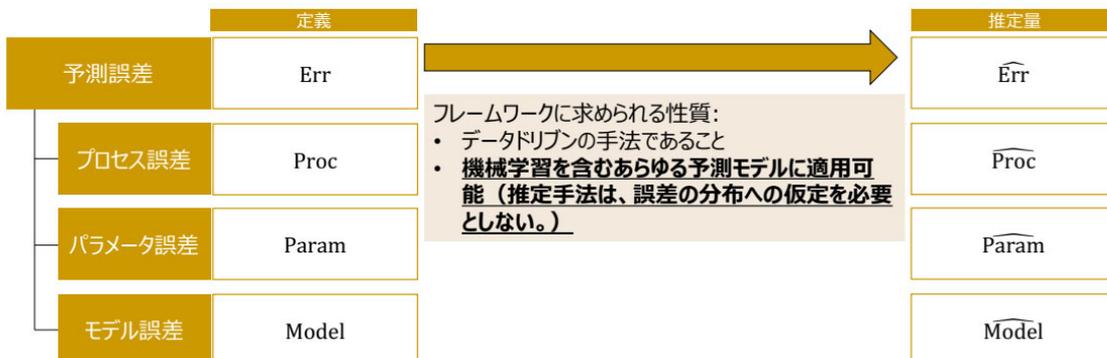
10

その分解した要素が何を表しているかというものが、このページのご説明です。まず、プロセス誤差はいわゆる確率的なぶれです。ベルヌーイ試行に従って、個別契約単位で解約するか、しないかが起こりますので、その確率的なぶれがプロセス誤差です。パラメータ誤差は、予測値とその予測値の期待値の差の二乗です。それはすなわち分散の式ですので、モデルの予測値の分散を表しています。モデル誤差とは、真の解約率との差なので、モデルのバイアスです。これらの誤差の要素をそれぞれ定量化するのが本研究のフレームワークです。

ちなみに、今回、契約単位ではなくてポートフォリオ単位で誤差を定量化していますので、分散効果があります。分散効果の振る舞いも、われわれの研究の成果の一つなのですが、本日は割愛させていただきます。興味がある方は、Appendix をご覧ください。

III. 推定方法

続いて予測誤差およびそれぞれの要素の大きさの推定を行う。



モデル誤差の推定 ($\widehat{\text{Model}}$) が我々の主要な成果であるため、ここに絞って説明する。
(プロセス誤差の推定 ($\widehat{\text{Proc}}$) とパラメータ誤差の推定 ($\widehat{\text{Param}}$) はAppendixを参照)

11

ここまでで分解できましたので、その分解した誤差を定量化していきます。われわれのフレームワーク

ですが、データドリブン手法で、機械学習を含む、あらゆるモデルに予測適用可能なものと考えております。

ここからは、具体的な手法について、高橋さんから、ご説明いただきたいと思います。

高橋 ここからは、アクサ生命の高橋が説明させていただきます。

ここからは、具体的な三つの要素について、どのようにデータドリブンで推定を行うか、実際に値を求めるかというところについて、ご説明いたします。

III. 推定方法

モデル誤差および予測誤差

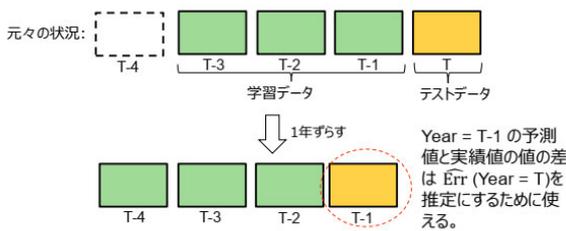
まずはポートフォリオの予測誤差の推定量 \widehat{Err} をどのように得るかを考える。 \widehat{Err} が得られれば、Model は以下のように得られる。

$$\widehat{Model} = \widehat{Err} - \widehat{Proc} - \widehat{Param}$$

Year=T-3より前のモデリングで使われていないデータを使って、 \widehat{Err} を推定できる。
しかしながら、**ポートフォリオ単位の予測誤差(Err)に起因する問題が生じる。**

定義
$Err = E \left[\left(\sum_{i \in D_T} \hat{f}(x_i) w_i - \sum_{i \in D_T} y_i w_i \right)^2 \mid \{x_i \mid i \in D_T\} \right]$
$Model = \left(\sum_{i \in D_T} (E[\hat{f}(x_i)] - f(x_i)) w_i \right)^2$

\widehat{Err} : の推定方法のベースアイデア



問題: 推定にあたってサンプルが1つしかない。

➤ これは統計的に信頼に足るものではない。

この問題は我々のフレームワークが、ポートフォリオ単位のデータを扱うから生じるものである。個別契約単位のデータを扱えばこのような問題は生じない。

もし、個別契約単位のデータを扱うならば、..



- ✓ Year = T-1には、複数の契約のデータがある。
- ✓ それぞれの個別契約に対して予測誤差が計算可能。
- ✓ 複数のサンプルに基づいて予測誤差の推定値が計算可能。

12

今日は、ちょっと時間がありませんので、プロセス誤差、パラメータ誤差については割愛させていただきます。こちらは既に求まっているものとして話を進めさせていただきます。

残りはモデル誤差ですけれども、一番の上の式にあるように、モデル誤差は先ほど分解した式がありますので、予測誤差からプロセス誤差、パラメータ誤差を控除することで求めることができます。ですので、ここは予測誤差をまず求めて、その差分でモデル誤差を出すというアプローチを考えます。

その予測誤差ですね。ここでは ERR ハットという記号にしていますが、こちらをどのように求めるかという基本的なアイデアにつきましては、先ほど、使っているデータより、ずっと昔のデータがあるという前提にしていたのですけれども、今使っているデータ、T-3年より前のデータを使って、このエラー、予測誤差を推定することが基本的な発想になります。

ただ、ナイーブにやると考えなければいけないところが幾つかあります。一番ナイーブなアイデアを左下の図に書いているのですけれども、図の上の状況ですね。「元々の状況」と書いてあるところが、先ほどまで申し上げている状況で、T-3からT-1年、3年分の学習データを使って、時点Tのテストデータについて予測を行うというものになっています。ナイーブな方法として、それを1年ずらしまして、T-4からT-2年のデータを学習データとして、T-1年をテストデータとして誤差を求めにいくということが考えられます。

T-1につきましては、実際の解約が起きたか、起きていないかというところは分かっているデータですので、予測値も知っているし、実績がどのようになったかも知っているので、誤差を具体的に求めるこ

とができます。ですので、 $T-1$ のデータで、予測と実績の二乗を取ることによって実際の誤差を知ることができて、それを推定値とするというところが、一つナイーブなアプローチになります。

ただ、問題があります。推定にあたってサンプルが一つしかないというところが、問題になるかと思えます。 $T-1$ の1時点でのみ評価するものになってしまいます。これは、当然、統計的に1サンプルでの推定になりますので、統計的に信頼に足りうるものではないかなと思っております。

一つのポートフォリオの中にはたくさん契約がありますが、ポートフォリオ単位の誤差を扱うために $T-1$ という一つのデータにまとめてしまうとサンプル数が1になってしまいます。つまり、この問題はわれわれがポートフォリオ単位の誤差を扱っていることによって生じる特有の問題と言えます。

III. 推定方法

モデル誤差および予測誤差

この問題を解決するため、以下のアプローチを取る：

✓ **利用可能な全ヒストリカルデータを使い、すべての組み合わせで予測誤差 (Err) を計算する。**

➤ 予測誤差を \hat{Err} をより多くのサンプルで計算可能。

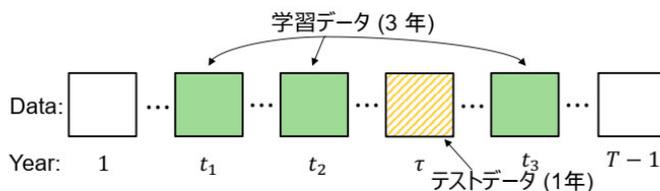
定義

$$Err = E \left[\left(\sum_{i \in \mathcal{D}_T} \hat{f}(x_i) w_i - \sum_{i \in \mathcal{D}_T} y_i w_i \right)^2 \middle| \{x_i | i \in \mathcal{D}_T\} \right]$$

$$Model = \left(\sum_{i \in \mathcal{D}_T} (E[\hat{f}(x_i)] - f(x_i)) w_i \right)^2$$

このアプローチを Step1 から Step3の流いで説明する。

Step 1. 学習データ(3年)とテストデータ(1年)を Year = 1 から Year = $T-1$ のデータから選ぶ。



13

それを解決するために、利用可能なヒストリカルデータを全て使って、サンプルの数を増やして求めにいくことが基本的な発想になります。

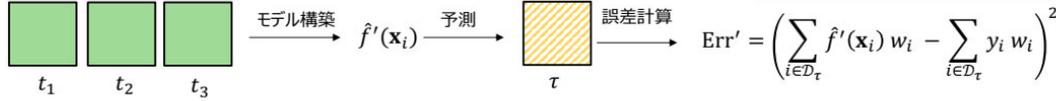
複雑になるので、ステップ1からステップ3に分けて、ご説明していきます。

まず、過去1年目から $T-1$ 年目までデータがあるのですが、こちらから任意の組み合わせで学習データ3年分とテストデータ1年分を選びます。

III. 推定方法

モデル誤差および予測誤差

Step 2. Step1で選んだ学習データからモデルを作成し、選んだテストデータの予測誤差を計算する。



Step 3. Step 1 と Step 2 を $(T - 1) \cdot T - 2 C_3$ 繰り返し、すべての学習データとテストデータの組み合わせで実施する。そして、その予測誤差の平均をポートフォリオ単位の予測誤差(Err)の推定値とする。

推定値

$$\widehat{Err} = \text{average of } \widehat{Err}'$$

$$\widehat{Model} = \widehat{Err} - \widehat{Proc} - \widehat{Param}$$

このアプローチの限界

- もしデータに強いトレンドがあれば、この推定方法は大きなバイアスを持ちうる。
- もし十分な年数のデータが無ければ、少ないサンプルでの推定となる。(データ年数(T-1)とデータ数の関係は下表のとおり)

$(T - 1)$	3	4	5	6	7	8	9	10
$(T - 1) \cdot T - 2 C_3$	N/A	4	20	60	140	280	504	840

ステップ2として、先ほど申し上げたナイーブなアイデアと全く同じように、3年分を学習データとしてモデルを構築して、テストデータについて予測を行って、誤差を実際に求める。ステップ3として、取りうる全ての組み合わせに対して同じ操作を実行して、得られたサンプルの誤差の単純平均を取って、われわれが求めたい予測誤差の推定値とするというのが、基本的なアイデアになります。

ちょっと、これは分かりにくいと思うので、次のページに、4年分データがある場合のパターンを具体的に書いております。

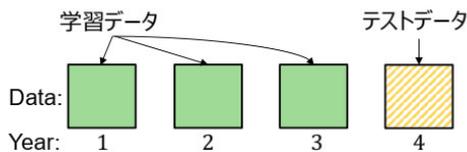
III. 推定方法

モデル誤差および予測誤差

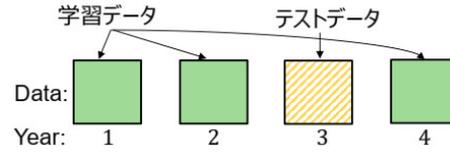
例: $T - 1 = 4$ の場合 (4年のデータが利用可能)

$(T - 1)$	3	4	5
$(T - 1) \cdot T - 2 C_3$	N/A	4	20

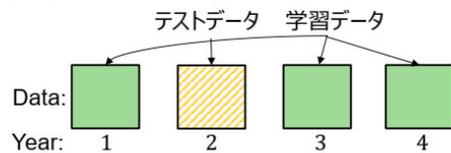
パターン 1:



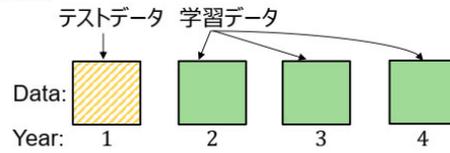
パターン 2:



パターン 3:



パターン 4:



$$\text{予測誤差の推定値 } (\widehat{Err}) = \{\widehat{Err}' (\text{パターン 1}) + \widehat{Err}' (\text{パターン 2}) + \widehat{Err}' (\text{パターン 3}) + \widehat{Err}' (\text{パターン 4})\} / 4.$$

$$\text{モデル誤差の推定値 } (\widehat{Model}) = \widehat{Err} - \widehat{Proc} - \widehat{Param}.$$

15

4年分のデータがあるので、学習データとテストデータの選び方は全部で4通り存在します。この書き下している4通りについて、全ての学習データ、テストデータそれぞれを使って、誤差を具体的に計算してあげて、それらの平均を取るということで、予測誤差の推定値を求めます。最初に申し上げたとおり、プロセス誤差、パラメータ誤差の推定値は既知としていますので、その差分でモデル誤差を求めることがで

きるという流れになります。

このアプローチについても、もろもろ限界はあるのではないかと考えております。一番下に書いてあるところですが、この求め方からして、過去のデータを使って、それを利用して予測誤差を推定するという方法になりますので、データに強いトレンドがある場合については、この推定方法は大きなバイアスを持ちうるのではないかと考えております。その場合は、トレンドを何らかの方法で除去するか、または、トレンド自体をモデルに取り込むような場合で適用できるのではないかと考えております。

また、データのサンプルを増やすと言ったのですが、例えば、下の表にあるように、7年のデータがあれば140個のサンプルを取れるのですが、実務上、7年は、なかなか長いのではないかと考えています。やはり、サンプルの数は、使えるデータとのかねあいで、大きな障害になるのではないかと考えております。

IV. 数値例

数値例で使用したデータの概要

- 統計数理ソフトRのパッケージCASdatasetsのデータセットuslapseagentを使用。
- uslapseagentは29,311件の米国の専業代理店チャネルで販売された契約日が1995年1月から2008年12月の終身保険のデータ。うち、本研究では2000年～2007年のデータを使用する(すなわち、T=8)
- 単純モデル(経過年数別解約率)、ロジスティック回帰、ランダムフォレストの3つの予測モデルを適用

変数	変数型	備考
polQ	数値型	経過年数。1～15年
acc.death.rider	カテゴリ型	特約あり/なし
gender	カテゴリ型	男性/女性
premium.frequency	カテゴリ型	年払/年払より短い期間/年払より長い期間
risk.state	カテゴリ型	喫煙/非喫煙
underwriting.age	カテゴリ型	若年(0～34歳), 中年(35歳～54歳), もしくは 熟年(55歳～84歳)
living.place	カテゴリ型	東海岸、西海岸、その他
annual.premium	数値型	標準化されており、-1.07～12.13の値を取る。 元のスケール(単位はドル)では、平均は560.88で、標準偏差は526.58
issue_year	カテゴリ型	1995年～2008年
surrender	0 or 1	この数値例での目的変数。1は解約、0は残存を表す。

16

ここまでの具体的なアイデアで、ここからは数値例を使って、イメージでお見せできればと考えております。

使用したデータについては、uslapseagentという、とあるオープンデータを使っております。こちらは、米国の終身保険の解約データになっております。われわれの研究では、2000年から2007年のデータを使っております。モデルとしましては、単純モデルと、われわれが呼んでいるものと、ロジスティクス回帰、ランダムフォレストという三つのモデルを使用しております。

単純モデルは何かと言うと、本当に一番ナイーブな、よくあるアサンプションの設定に近いもので、過去3年分のデータを使って、経過年数別に契約解約率のアサンプションを求めるというモデルになっております。

このデータセットは、変数がたくさんあるのですが、例えば後で登場するものとして、下から二つ目の赤く囲っているannual.premium、つまり年換算保険料が含まれています。他にもたくさんの変数が含まれています。

IV. 数値例

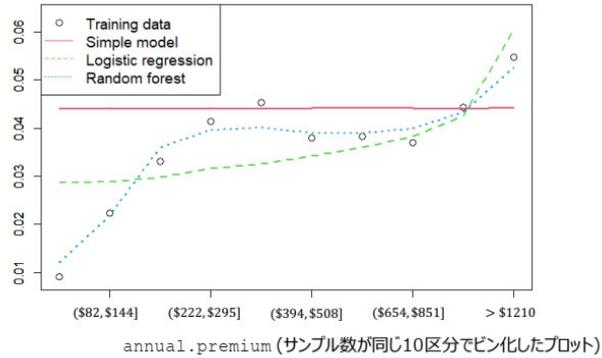
3つの予測モデルの比較

3つのモデルの特徴を理解するために、予測される解約率を`annual.premium*`でプロットする。

単純モデル：`annual.premium`は説明変数ではないため、解約率の予測値は一定。

ロジスティック回帰：ロジスティック曲線の効果を取り込んでいるが、フィッティングは良くない。

ランダムフォレスト：`annual.premium`との複雑な関係を捉えている。



* `annual.premium`は連続な数値型特徴量であるため、含まれるサンプル数が同じ10区分（ビン）に分けている。

17

モデルの違いをイメージしていただくために、1枚、スライドを書いております。こちらは、横軸を、先ほど言った `annual.premium` の水準ごとに取りまして、縦軸が解約率です。学習データと三つのモデルについて解約率をプロットしたものになります。学習データは丸印になっておりまして、`annual.premium` が低いほど解約率が低くなって、高くなってくると解約率も高くなってくる、という状況になっています。

単純モデルは、どのような感じになるかと言うと、赤い直線が一直線のものになっております。これは `annual.premium` を説明変数に取ってない、経過年数だけを取っていますので、`annual.premium` に依存せずに一直線のものになります。また、ロジスティック回帰は緑色の線になっていますけれども、ロジスティック回帰は基本的に線形モデルですので、線形で立ち上がっていくような形になっております。ランダムフォレストにつきましては、青い線ですけれども、一番表現力が高いモデルですので、学習データの複雑な構造をトレースしているような形になっております。

これが、三つのモデルがどのような違いになっているかというところを大まかに説明したものになっております。

IV. 数値例

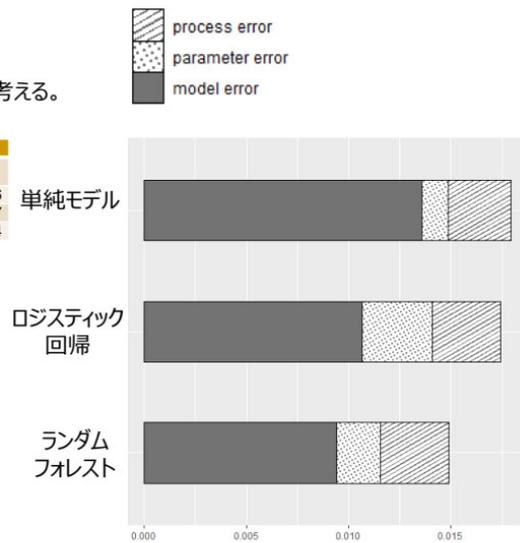
誤差の定量化と分解の結果

予測モデルごとの1件ごとの標準化予測誤差 ($\widehat{Err} / (\sum_{i \in D_T} y_i w_i)^2$) を考える。

予測モデル	予測誤差	ブレイクダウン		
		プロセス誤差	パラメータ誤差	モデル誤差
単純モデル	0.0179	0.0030	0.0013	0.0136
ロジスティック回帰	0.0174	0.0033	0.0034	0.0107
ランダムフォレスト	0.0149	0.0033	0.0021	0.0094

考察:

- **プロセス誤差**はすべてのモデルで同様である。
- **パラメータ誤差**は、単純モデルで最も小さい。これは1つの説明変数しか持たず、推定値が安定しているため。ロジスティック回帰のパラメータ誤差は大きいですが、正則化によって改善できるかもしれない。
- **モデル誤差**は単純モデルが最も大きく、当てはまりが悪いことを意味している。ランダムフォレストは、モデル誤差が最も小さく、解約率の構造を適切に捉えていることが示唆される。



$\sqrt{\text{Prediction Error}}$ = 0.134 (単純モデル), 0.132 (ロジスティック回帰), 0.122 (ランダムフォレスト).

これは解約量の予測値の1σの幅が±13.4% (単純モデル), ±13.2% (ロジスティック回帰), ±12.2% (ランダムフォレスト)であることを意味している。

18

ここで、誤差分解した結果について表示しております。ここで表示しているものは、契約1件単位に換算した予測誤差を書いております。右のグラフがその結果になっていまして、三つのモデルと予測誤差の結果を並べています。グラフをパターン分けしたところが、プロセス、パラメータ、モデル誤差を表していまして、三つを足したものが予測誤差全体の数字になっています。

まず、グラフの斜線のところがプロセス誤差を表しているのですが、こちらについては、どのモデルについても、大体ほぼ同じ値になっています。これは期待した結果になっていまして、プロセス誤差というものは、最初に申し上げたとおり、事象に内在しているような誤差だったので、モデルに依存しない結果が得られたということは、期待したとおりの結果になっています。また、一番規模が大きい黒く塗りつぶしているところがモデル誤差なのなのですが、こちらは単純モデルで一番モデル誤差が大きくて、ランダムフォレストで一番モデル誤差が小さいという形になっています。

先ほどのグラフから分かるように、単純モデルは、説明変数を一つしか取っていなくて、解約率の結果もうまくトレースできていなかったの、モデル自体にバイアスがあって、このようにモデル誤差が大きいという結果になっているかと思えます。

一方、ランダムフォレストについては、表現力が高いモデルで、先ほどの解約率のグラフでも、結果をうまくトレースできていたので、モデルのバイアスが比較的小さくて、モデル誤差も小さくなっているという結果が得られるかと思えます。

こちらが、誤差分解と定量化の主要な結果になります。

IV. 数値例

誤差分解によって得られる知見

- 伝統的なモデルと機械学習モデルの予測誤差を、同じデータドリブンな手法を用いて定量化し、比較することができる。
- 予測誤差の内訳を理解することで、誤差を低減する方法についての洞察が得られる。

	誤差を低減するための有効な方法
プロセス誤差	<ul style="list-style-type: none">➢ ポートフォリオサイズを大きくする。➢ 保険金額の大きい契約を出再して保有保険金額を引き下げる
パラメータ誤差	<ul style="list-style-type: none">➢ パラメータを決定する際の学習データを増やす。➢ 例えば正則化のような、予測値を安定させるテクニックを導入する。
モデル誤差	<ul style="list-style-type: none">➢ 予測モデリングそのものを見直す。

- それぞれの予測モデリング手法の特徴を理解し、予測モデルを選択する際の参考とすることができる。

19

その結果を得て、「何がうれしいの？」というところが、次のスライドになっております。メッセージとしては、真ん中の表が一番伝えたいところです。誤差の内訳を知ることができたので、モデルに誤差があったときに、どのようにして誤差を減らせばいいかという知見が得られるというところが、大きいのではないかと考えています。

例えば、誤差分解してプロセス誤差が大きかった場合については、プロセス誤差はモデルに依存しない事象自体に内在している誤差と言ったので、モデルをいじったから改善するものではないのではないかと考えています。ですので、プロセス誤差を検証させたいためには、ポートフォリオのサイズを大きくして分散効果を効かせるようにする。サイズを増やすことが難しければ、ビジネス上のアクションとして、再保険を活用して誤差を減らすということが考えられるのではないかと思います。

また、モデル誤差が大きかった場合については、モデルのバイアスになってきますので、モデル自体を見直す。アサンプションの文脈で言うと、アサンプションの説明変数を変えたり、全く別のモデルを試してみるということが分かるかと思えます。

このように、誤差の内訳を知ることによって取りうるアクション、どのようにすれば誤差を減らせるかということが分かるところが、一つ大きなうれしい知見ではないかと思えます。

V. 結論

- 機械学習法を含むあらゆる予測モデルに適用可能な、生命保険負債の評価における予測誤差を定量化するための新たなフレームワークを提案した。
- 我々の提案したフレームワークは…
 - 一般化されており、**データドリブン**である。
 - **分散効果後のポートフォリオベースの予測誤差**を扱うことができる。
 - **モデル選択**のための洞察を提供する。
 - **予測誤差を改善するための有効なアクション**を導く。
- ネクストステップは…
 - モデル誤差の推定手法にさらなる改善が必要かもしれない。
 - このフレームワークを、例えばベストエスティメイト負債のリスク評価など興味のある他の問題に拡張する。
 - 他のアサンプション、例えば死亡率などに応用する。

20

最後に結論を述べています。われわれの結論としましては、一般的な誤差分解を定量化できるフレームワークを提案しました。かつ、それはデータドリブンで決定することができます。また、保険会社が興味のある分散効果後のポートフォリオ単位の予測誤差を扱うことができるところが大きいところです。先ほど申し上げたとおり、予測誤差を改善するためのアクションを、その結果から導き出すことができることが、結論として挙げられます。

ただ、課題もありまして、途中で述べたのですけれども、モデル誤差の推定手法には、まだバイアスを用いる手法ではないかと思っているので、改善が必要というところ。今回は1年間の解約料をテーマとして扱ったのですけれども、他の興味ある題材、例えば、ベストエスティメイトの保険負債のリスクを直接扱ったり、今回は解約率を扱ったのですけれども、死亡率を扱うなど、他の興味ある題材にもチャレンジしていく必要があるのではないかと考えております。

一旦、私からの説明は以上になります。

藤田 ありがとうございます。それではここまでで、質問がありましたらお願いします。

A 発表、ありがとうございます。

質問です。今回の設定では、例えば男女などという属性から結果を予測するということだったのですが、将来的に、例えば、死亡率の予測などで時系列的なデータから予測したい。例えばですけれども、前年に収入が減ったから解約するような過去のデータを使いたいというモチベーションがあると思います。一方で、そうすると、パターンを四つ見せられていたと思うのですけれども、今のデータを使って過去を予測するという時系列的な動きができなくなってしまうと思うのです。その辺は、時系列の性質を使って予想したい場合に、どのように、この手法を使えるか、ということは考えられているのですか。

高橋 ご質問ありがとうございます。

まさに、おっしゃっていただいたとおり、時系列モデルにも適用したいということは、大きなモチベー

ションとして、あるのではないかと思います。

今、われわれの手法だと、パターンを四つ、先ほどの具体例で出していて、時系列に関係なく、時間関係を特に考慮せずに並べているものです。それは、やはり、時系列モデルや時系列的な要素を、今、見ないで使っているから並べていい、というところを考えられるのではないかと思います。

時系列モデルを使う場合には、やはり選び出すパターンとして、時系列の順序を守った、先ほどのパターン四つで言うと、もっとパターンが減るような組み合わせで抜き出してこななければいけないのではないかと思います。

その場合は、サンプルの数が減ってしまいますので、利用できるデータの年数を増やす、もしくは、今、年度単位で扱っているのですけれども、例えば四半期単位でデータを区切ると、データの数が4倍になります。時系列的にも、四半期単位という細かい粒度でデータを扱えるようになってくるので、そのようなやり方が、方向性としては考えられるのではないかと考えております。

A ありがとうございます。

鈴木 本研究では、高橋さんがご説明されたとおり、時系列のデータの順番を入れ替えてやっけてしまっているのですけれども、これを時系列の順番どおり、すなわち、時系列のデータを1年ずつ順番にずらしていくアプローチを取られている方はいます。時系列クロスバリデーションなどと言われる手法です。

われわれの手法で、最初はそれをやろうとしたのですけれども、先ほど話があったようにサンプルがすぐ減ってしまうので、全組み合わせを取ることでサンプルを増やしています。サンプルを増やすための工夫とご理解いただければと思います。

発表でもありましたとおり、時系列で何かが変わってしまう要素、発表の中ではトレンドと言いましたけれども、それがあある場合には、確かに、うまく機能しないので、そこはサンプル数とトレードオフというように考えております。

B かんぼ生命のチダです。大変興味深かったです。

質問です。今回の取り組みは、「よく分からないモデルを使って、よく分からない誤差なので、なるべく成分を分解したい」ということなので、私は強く関心があるのですけれども。今回、このように分離したモデル誤差が、とても賢いモデルを選んだら、このモデル誤差に、ある下限のようなものがあって、その下限が0まで行くのか、あるいは、ある値で落ち着くのかというところに、とても関心があるのですね。

質問は、幾つかモデルを、多分、検証されたと思うのですけれども、そのモデル誤差について、賢いモデルであっても、それなりに大きなモデル誤差があるのか、あるいは、モデル誤差がとても小さいような実験があったのか、ということを知りたいです。

高橋 ご質問ありがとうございます。

こちらについては、モデル誤差が、生命保険の問題については、必ず0になることは、なかなか難しい状況かと思っております。

といいますのも、やはり扱っている状況が、解約や死亡という不確実性が大きい状況になっておりまして、例えば、契約者の情報が全て分かっていたら、完璧に、その人が解約するのか、死亡するのか、とい

うことが分かるという問題ではないのかと思っています。

一定程度ポートフォリオで扱っているところで、群団単位で予測しているので、そのような要素を排除できていると思うのですが、なかなか、問題の性質上、モデル誤差を完璧に0にするのは難しい状況ではないかと思っております。実際、今回、扱ったデータでも、いろいろなモデルも試してみたのですが、モデル誤差を大きく減らすところまでは、できていない状況です。

鈴木 「モデル誤差」という言葉が、もしかしたら、よくないのかもしれないですが、モデルが持つバイアスだけではなくて、先ほどお話ししたトレンドの要素も、この中に混ざってしまっています。

今回のフレームワークだと、先ほどの数式のところで、 f_{xi} （真の解約率）がずっと変わらないという仮定を置いていて、その中でモデルのバイアスというもので考えています。だから、どれほど賢いモデルでも、何かトレンド的なもので解約率が変わってしまう場合は、モデル誤差はありうるのではないかと思っています。

あと、特徴量も時系列で変化しないもののみを使用しています。特徴量自体は、生命保険ではよくある特徴量ではあるのですが、加入したときからは変わらない特徴量を使っていて、このあたりも時系列で変わる特徴量を使うと影響が出てくるのではないかと思っています。

藤田 ありがとうございます。Slido から高橋さんへのご質問になります。

「モデル誤差について、単純にパラメータの多いモデルを採用した方が小さくなりやすいということはあるのでしょうか」

高橋 ありがとうございます。

傾向としては、パラメータが多い、表現力の高いほうが、モデル誤差は小さくなりやすいのではないかと思います。

ただ、一方で、よくデータサイエンスである状況ではないかと思っておりますけれども、学習データにオーバーフィットしやすいところもありますので、オーバーフィットしてしまうと、モデル誤差のバイアスが大きくなってしまったりはありうるのではないかと思います。

ですので、パラメータ誤差とモデル誤差に分解していくことが重要ではないかと考えております。

藤田 ありがとうございます。それではお時間になりましたので、鈴木さん、高橋さん、ありがとうございました。

続いて、二つ目のプレゼンテーションに移りたいと思います。では、上妻さん、浅芝さん、準備ができ次第、よろしくお願いいたします。

上妻 後半のプレゼンテーションは、大同生命の上妻と、T&Dホールディングスの浅芝さんから、ご報告させていただきます。



関数分解手法 MID の応用による 機械学習モデルの解釈手法の提案

データサイエンス関連基礎調査WG 活動報告

データサイエンス関連基礎調査WG

図の出典：Harry Steele Savage 'Pandora's Box'

From 'Stories of the Gods and Heroes' by Sally Benson, 1940, Dial Press. Reprinted in Colliers Junior Classics, 'Legends of Long Ago', 1962.

データサイエンス関連基礎調査WG | 0

二つ目のプレゼンテーションは、関数分解手法 MID、以降「ミッド」と読みますが、その応用による機械学習モデルの解釈手法の提案という内容になっています。

このプレゼンテーションで伝えたいこと

1

既存の代表的な IML 手法には課題がある

既存の IML 手法としてよく用いられる手法には次のとおり問題がある。

- ・特徴量間の相関が高い場合、PDP は主効果を捉えられない
- ・SHAP は（説明変数の値が同じでも）インスタンスごとに測定される効果量が異なる
- ・交互作用をわかりやすく可視化する方法が限られている

2

関数分解の考え方を応用した新たな IML 手法が 既存手法の課題のいくつかを解決する

今回提案する新しい IML 手法「MID」には次のような特徴があり、前述の問題点を解決。

- ・特徴量間の相関が高い場合でも主効果を捉えることが可能
 - ・特徴量の効果がグローバルな関数として得られる（説明変数の値が同じであれば測定される効果量も同じ）
 - ・交互作用をシンプルな仕組みでわかりやすく可視化できる
- また、実用的な R / パッケージも用意。

データサイエンス関連基礎調査WG | 1

最初にプレゼンの内容ですけれども、前半は、既存の代表的な IML 手法には課題があるという、課題を指摘するパートになっています。後半はそれに対して、MID という関数分解の考え方を応用した新たな IML 手法が、既存手法の課題のいくつかを解決する。そのような内容になっています。



既存の代表的なIML手法には課題がある

データサイエンス関連基礎調査WG | 2

まず、前半の問題点を指摘するパートです。

解釈可能な機械学習

- 予測モデルの解釈のしやすさと予測精度は、トレードオフの関係にあることがある。特に、さまざまな問題に対して高い予測精度を発揮するような機械学習モデルは往々にして非常に複雑な予測関数を持つため、理解することも説明することも難しい。
- 解釈しにくい予測モデルのことを**ブラックボックスモデル**という。
- 機械学習モデルの予測関数の入出力関係を可視化するなどの工夫を通じて**ブラックボックスモデルの解釈を行う手法**のことを、“Interpretable Machine Learning”（解釈可能な機械学習）の頭文字を取って「**IML手法**」という。
- 高精度なブラックボックスモデルの予測関数のふるまいを理解することで、データを生み出す現象の理解や、解釈のしやすいモデルの改良に役立てることができる。
- 予測の精度を高めるだけでなく、その予測の理由をわかりやすく説明することも要求される**アクチュアリーにとって、IML手法は有用なものだ**といえる。

データサイエンス関連基礎調査WG | 3

本題に入る前に、IMLという単語がいきなり出ておりますので、ここで1枚スライドを挟んで「IMLとは何ぞや」という話をします。時間の都合上あまり細かくは説明できませんので、簡潔に説明します。

まず、予測モデルのうち解釈しにくいもののことを、われわれはブラックボックスモデルと言ったりします。そのようなブラックボックスモデルに対して解釈を試みようというものを、Interpretable Machine Learning という英語のイニシャルをとって「IML手法」と言ったりします。このようなモデルに対し、予測の精度を高めるだけでなく、その予測の理由を分かりやすく説明することも、われわれアクチュアリーは要求されますので、アクチュアリーにとっては、IML手法は大変有用なものと考えております。

既存の代表的なIML手法 (PDP + ICE)

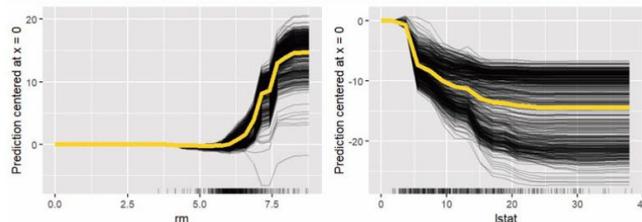
- **PDP + ICE**は特定の説明変数が予測値に与える**平均的な効果**を可視化する手法

- ICE : 各インスタンスについて、ほかの変数を固定したまま注目している説明変数の値だけを動かしたときの予測値の変化をプロットしたもの。インスタンスごとに形状が異なる場合、交互作用が存在する。
- PDP : 全インスタンスのICEを平均したもの。

$$f_{j,PDP}(x_j) = E_{X_C}[f(x_j, X_C)] = \int f(x_j, X_C)dP(X_C)$$

f : 解釈したい予測モデル・関数 | $j \in \{1, \dots, d\}$ 注目している説明変数の添え字 | x_j : 注目している説明変数 | x_C : x_j 以外の説明変数

イメージ: 各 $f(x_j, x_C^{(1)})$, $f(x_j, x_C^{(2)})$, $f(x_j, x_C^{(3)})$, ... について横軸を x_j にして描いたものがICE(黒)で、それを平均したものがPDP(黄色)



(*1) PDP = Partial Dependence Plot, ICE = Individual Conditional Expectation / (*2) インスタンス : データの1レコード (サンプル) のこと

データサイエンス関連基礎調査WG | 4

具体的にIML手法を幾つか紹介します。まず一つ目が、PDP + ICEといわれているものです。記号の使い方が前半のプレゼンテーションと違って申し訳ないのだけれども、まず「f」が予測関数で、先ほどのプレゼンだとfハットと書いていたものだけれども、こちらではfになっています。「X」が説明変数で、例えば、特徴量が8個あったら、8変数関数fのことを予測モデルと呼んでいます。

そのような状況なのだけれども、まず下のグラフがPDPやICEといわれているものです。例えば左側のグラフですと、横軸にrmという説明変数を取り、縦軸にその予測関数の予測値をとったプロットになっています。

実際にサンプルデータを何百件か用意して、それに対して一つrmという説明変数だけを例えば0から8まで横に動かし、それ以外の変数は元々のサンプルのまま。そのようなグラフを描くと、何百本かのグラフができますが、それが、このグラフの黒ひげのようなものです。このようなプロットをICEと呼びます。一方、この平均を取ったものが黄色の線にして、これをPDPと呼びます。このようなものが特定の説明変数、例えばrmなどが予測値に与える平均的な効果を可視化する手法として用いることができるということです。

既存の代表的なIML手法 (SHAP)

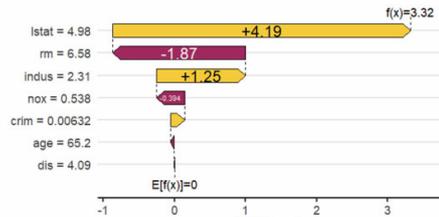
- SHAPは個別インスタンスの予測値を説明変数ごとの寄与に分解する手法

- 協力ゲーム理論における公平な配当分類の手法として定義される「シャプレー値」に基づく。
- 各説明変数についてデータを入力する場合としない場合の予測値の変動の加重平均として計算。

インスタンス $(x_1^{(i)}, \dots, x_d^{(i)})$ の予測値を、SHAP値 $\phi_1^{(i)}, \dots, \phi_d^{(i)}$ に加法的に分解

$$f(x_1^{(i)}, \dots, x_d^{(i)}) = E[f(X)] + \phi_1^{(i)} + \dots + \phi_d^{(i)}$$

$$\phi_j^{(i)} = \sum_{S \subseteq \{x_1^{(i)}, \dots, x_d^{(i)}\} \setminus \{x_j^{(i)}\}} \frac{|S|!(d - |S| - 1)!}{d!} [f(S \cup \{x_j^{(i)}\}) - f(S)]$$



(1*) SHAP = SHapley Additive exPlanations / (2) データを入力しない場合の予測値 (例えば $f(x_1, x_2)$ に x_2 を入力しない場合) は、期待値や条件付期待値で定義

データサイエンス関連基礎調査WG | 5

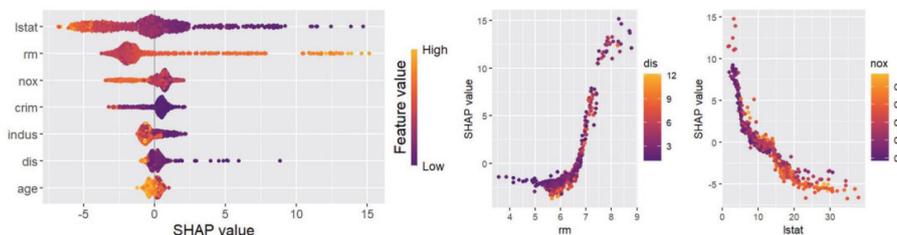
次に、また別の手法としてSHAPと呼ばれるものがあります。こちらは、個別のサンプルの予測値を説明変数ごとの寄与に分解する手法で、下の図を見ていただきたいのですが、下から上に予測値が出来上がっていく感じとなっています。

最初に、 $E[f(X)] = 0$ とあって、これは全体の予測値の平均が0とっている。次に、 $dis = 4.09$ という行があるかと思うのですが、これで予測値が、ちょっと右か左か分からないですが、行っている。次に、 age というものが左、 $crim$ が右。左、右、左、右と行って、最後の予測値が3.32です、というプロットになっております。

それぞれ、+4.19、-1.87 などと書いてあるものが、この説明変数によって予測値がどのように移動したという値になっていて、これらの値をわれわれはSHAP値と呼んだりします。ですので、一つの予測に対して説明変数の数だけSHAP値が計算されることになります。

既存の代表的なIML手法 (SHAP)

- SHAP値を多数のインスタンスについて計算することで、予測関数全体に対するグローバルな解釈を与えることも可能



説明変数ごとの効果の全体サマリー
(横軸にSHAP値、色分けを説明変数の値とした散布図を説明変数ごとに縦に並べる)

ある説明変数における効果の可視化
(縦軸にSHAP値、横軸をその説明変数の値、色分けを別の説明変数とした散布図)

データサイエンス関連基礎調査WG | 6

SHAP値とは個別の予測を分解するものですが、これをたくさんのサンプルで計算してやることによって、予測関数全体に対するグローバルな解釈を与えることもできます。左側のプロットは説明変数ごとのSHAP値を並べたものになっていて、例えば、一番上の lstat というものが一番横に広がっているのですが、要するに、lstat というもので予測値がたくさん動くということを示していて、説明変数の中でも一番重要であろうという感じで用いることができます。

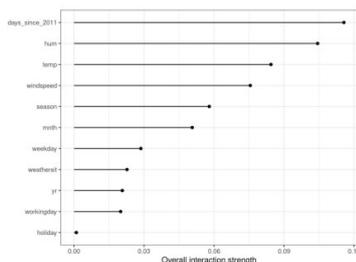
一方、右側に並んでいるグラフは、横軸に説明変数の値、縦軸にSHAP値をとったものです。例えば、rmを横軸にとっているものだと、7から8にかけてSHAP値が上がっている。これは要するに、7から8にかけて予測値が上がるということを示すことができるというものになっています。

既存の代表的なIML手法（H-統計量）

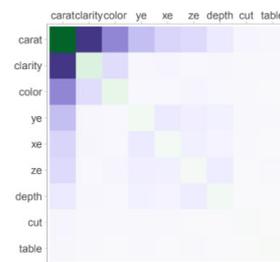
- **FriedmanのH-統計量**は、2つの説明変数間の**交互作用の大きさ**を表す指標で、“2次元PDPの2乗和のうち1次元PDPの和で説明できない部分の2乗和の割合”として定義

$$H_{jk}^2 = \frac{\sum_{i=1}^n [PD_{j,k}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)})]^2}{\sum_{i=1}^n [PD_{j,k}(x_j^{(i)}, x_k^{(i)})]^2}$$

PD_j は $f_{j,PDP}$ を中心化したもの。 $PD_{jk}(x_j, x_k) = f_{j,k,PDP}(x_j, x_k) - E[f_{j,k,PDP}(x_j, x_k)]$ 等



ある説明変数に対するH-統計量をソートして表示



全H-統計量をヒートマップとして表示

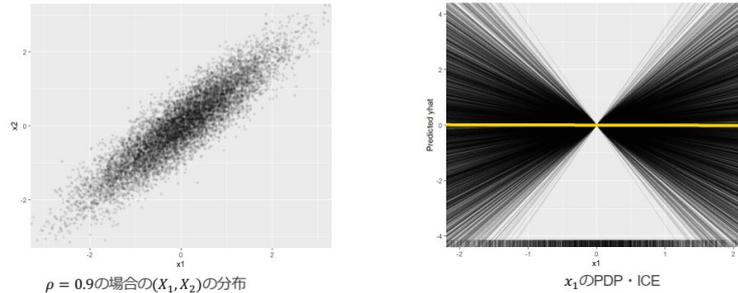
次は毛色が違うものなのですが、H-統計量といわれるものがあります。これは説明変数間の交互作用の大きさを表す指標で、2次元PDPの二乗和のうち、1次元PDPの和で説明できない部分を交互作用であるとみなし、その二乗和の割合として定義されます。

出来上がりとしては、右側のヒートマップが分かりやすいのではないかと思います。対角線は無視してそれ以外のところを見ていただくと、色が濃くなっているところは、H-統計量が大きいところを表しています。例えば、この図ですと、左から2番目、上から1行目のところで、carat と clarity というところが一番濃いので、「この二つの説明変数間の交互作用が強いですよ」ということを、この図は示しています。

PDPの問題点（独立性の仮定）

- PDPは説明変数間の相関が強い場合、主効果*をうまく捉えることが出来ない

- 例： (X_1, X_2) が平均 $(0,0)$ 、分散共分散行列 $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ の多次元正規分布に従うときの、 $f(x_1, x_2) = x_1 x_2$ という予測モデルの主効果を測定する場合
- X_1 と X_2 の相関係数 ρ が1に近い場合、予測モデルは x_1^2 に近いことから本来ならば主効果 $f_1(x_1)$ は放物線のような形になるべき。しかし PDP では $f_1(x_1) = 0$ となる。



(*) 主効果(main effect)とは、説明変数が予測値におよぼす影響を、1つの説明変数に注目して測定したもののこと。

たとえば、1次元のPDPやALE、SHAP Dependence Plot(縦軸にSHAP値、横軸をその説明変数の値をとった図)は主効果を測定する手法といえる。

なお、複数の説明変数間の交互作用を同時に測定する場合は、原則としてそれを取り除いた部分を指す。

データサイエンス関連基礎調査WG | 8

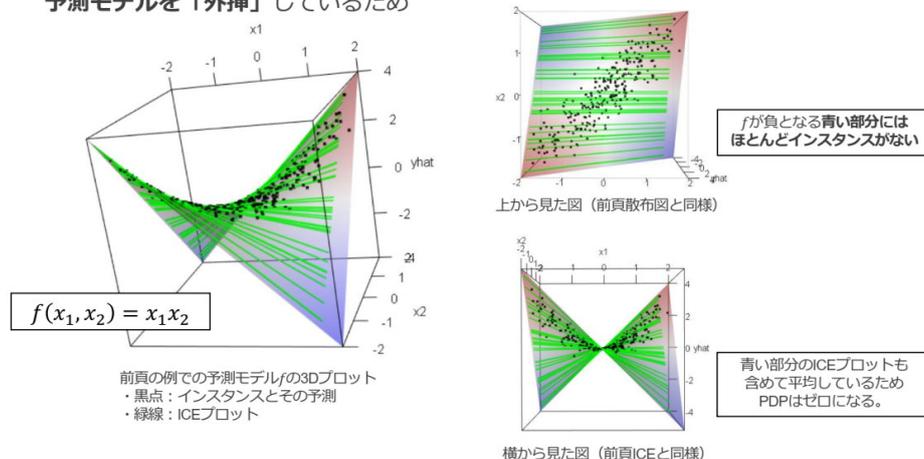
ここからは、問題点を指摘するパートに移ります。まず、PDPの問題点です。よく言われているのは、説明変数間の相関が強い場合に、主効果をうまく捉えることができないというものがあります。

具体的な例として、説明変数が二つの場合を考えます。X1とX2があって、それぞれは標準正規分布なのですが、それらに相関があるという状態を考えます。ここで解釈を試みる予測関数は、天下り式なのですが、 $X_1 \times X_2$ という積項と呼ばれるものを、ここでは与えます。X1とX2の分布の相関係数が1に近いような場合は、左側のプロットのような散布図が得られるのですが、これに対して、PDP、ICEを描いてみると、右の図のようになりまして、横一直線のPDPが得られる。

一方、予測関数は $X_1 \times X_2$ ですが、X1とX2は非常に強く相関しているので、実態としてはX1の二乗のようなものなのです。ですので、主効果としては放物線のような形になってほしいという気持ちがあるのですが、しかしながら、PDPは0一定になってしまう。

PDPの問題点（独立性の仮定）

- PDPが主効果を捉えられないことがあるのは、インスタンスが存在しない領域にまで予測モデルを「外挿」しているため



f が負となる青い部分にはほとんどインスタンスがない

青い部分のICEプロットも含めて平均しているためPDPはゼロになる。

データサイエンス関連基礎調査WG | 9

これをもう少し詳しく説明したものが、この図です。左側の図を見ていただきたいのですが、これは、3次元的に $X_1 \times X_2$ という関数をプロットしたものです。横軸に X_1 、奥行きは X_2 、縦軸に予測関数を置いているもので、黒い粒々は実際にサンプルがあるところ、緑のところはICEプロットを書くときのサンプルパスのようなものだと思ってください。

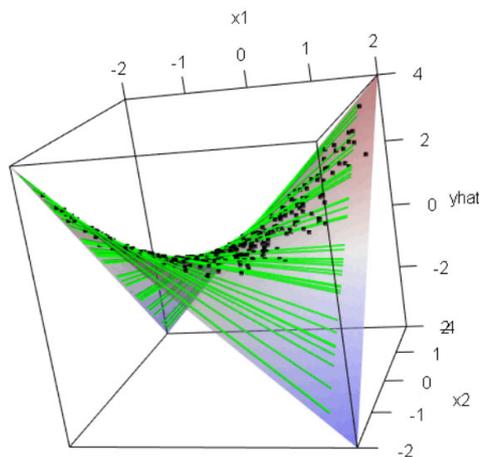
これを上から見ると、右上の図になりまして、黒い粒々は散布図と同じものが得られる。赤や青などがあるのですが、赤いところは予測関数が正のところ、青いところが負のところを表しています。このように見ると、負のところには、ほとんどサンプルがないことが分かります。

横から見たものが、右下の図になっております。黒い点を見てみると、ほとんどが正のところ集中しているということが分かり、また放物線のような形になっていることも分かります。

一方、緑色の線がICEプロットでして、このように見ると青いところも赤いところも平等に直線がある。結果として、青いほとんどサンプルがないところも、平等に平均に含んでしまっているので、PDPは横一直線になってしまうわけですが、このような現象のことを、「予測モデルを外挿しているからPDPはだめなのだ」と説明されることがあります。

「外挿」とは、右上の図で言うところのサンプルのないところまで予測値を出してしまっていることを表しています。今回の例ですと天下り式に $X_1 \times X_2$ を与えてしまっているのであまり問題に感じないかもしれませんが、実際に予測モデルを使ってみるときは、この青いところの予測値は、あまり本質的でないといえますか、意味がないことが多いので、そのようなところを平均に含めることは問題だろう、ということですね。

PDPの問題点 (独立性の仮定)



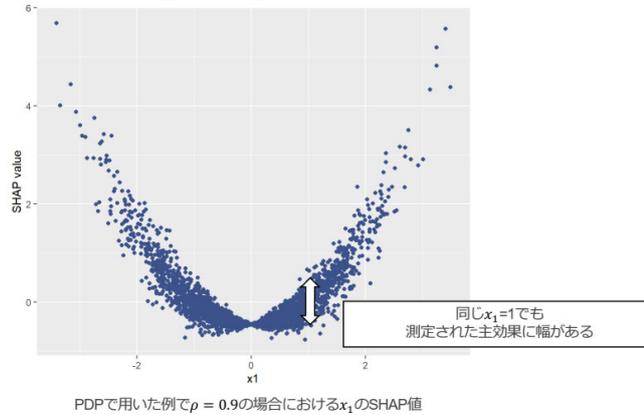
(おまけ) 前頁の図のアニメーション版

ちなみに、このような感じで図をアニメーションさせてみました。PDFには、これは当然反映されていないので、この場限りです。

SHAPの課題（説明変数と効果の非対応）

- SHAPの場合、注目している説明変数の値が同じインスタンスの間でも、他の説明変数の値が異なれば、**インスタンスごとに測定される主効果が異なってしまう**

➢ それがどうい原理によるものかを理解するのは容易ではない。



データサイエンス関連基礎調査WG | 11

次に、SHAPの課題です。下の図は、先ほどと同じように、横軸に説明変数の値、縦軸にSHAP値をとったプロットです。先ほどと同じ例なのですが、放物線の形にはなる一方で、縦幅のようなものが現れることが問題です。

SHAP値はサンプルごとに計算されるもののため、同じX1の値、例えば1のところにも上下の矢印を置いているのですが、それなのにSHAP値が上と下にぶれている。「このぶれが何なのか？」ということが、ぱっとは分かりにくいことがSHAP値の課題です。

交互作用の可視化手法の課題（解釈性・計算量）

- **交互作用の指標として用いられるH-統計量は、説明変数間の相関が強い場合、計算の基本となるPDが主効果をうまく捉えられていないために、交互作用の強さをうまく捉えることが出来ない**

➢ PDPで用いた2変数関数の例では、1次元PDPがゼロであることからH-統計量は1となる。つまり、予測値の変化のすべてがx1とx2の交互作用によるものと解釈される。

$$H_{jk}^2 = \frac{\sum_{i=1}^n [PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)})]^2}{\sum_{i=1}^n [PD_{jk}(x_j^{(i)}, x_k^{(i)})]^2}$$

↑ ゼロ ↑ ゼロ

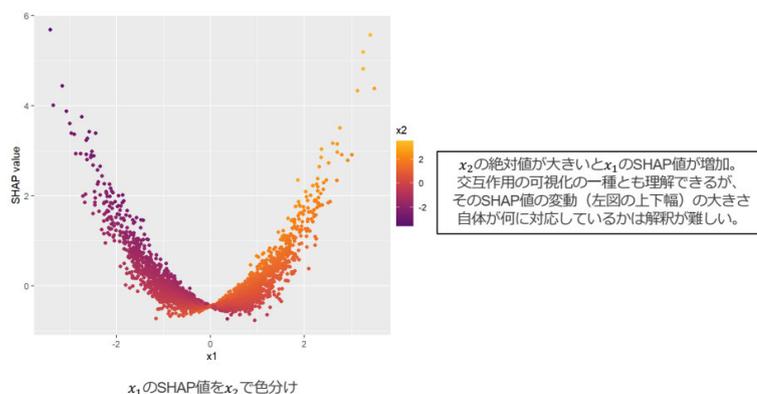
データサイエンス関連基礎調査WG | 12

次に、H-統計量を含めた交互作用の可視化手法に対する課題です。H-統計量はPDPを用いて計算されるものであるため、先ほどの例ですと、この算式のPDjやPDkと書いているものが1次元PDPなのですが、これがどちらも0になってしまい、1分の1と計算されて1になります。H-統計量が1ということは、「予測値の変化全てがX1とX2の交互作用です」と言っているわけです。このように説明変数間の

相関が強い場合は、PDPが微妙なためH-統計量も微妙だということになります。

交互作用の可視化手法の課題（解釈性・計算量）

- SHAP では、他の説明変数の値で色を変えることで2変数間の交互作用を可視化する方法があるものの、その結果を正確に理解することは簡単ではない



(*) 交互作用まで考慮した SHAP Interaction Value という手法もあるが、インスタンスによって効果の値が異なる点は共通している。

データサイエンス関連基礎調査WG | 13

実は、SHAPを使っても交互作用は可視化できます。この図は、先ほどの青かった図を色分けしたもので、X₂の値で色分けしております。X₂の値によってSHAP値が異なるという現象によって縦幅が生まれていたことが分かりました。

具体的には、「X₂が大きい場合にX₁のSHAP値が大きい」という状況になっております。X₁ × X₂という予測関数の持つ「X₂が大きいほどX₁が予測関数に与える影響が大きくなる」という特徴を表しているという点では非常に分かりやすいとは思いますが、「結局、この縦幅は何なのか？」というところは分からずじまいです。

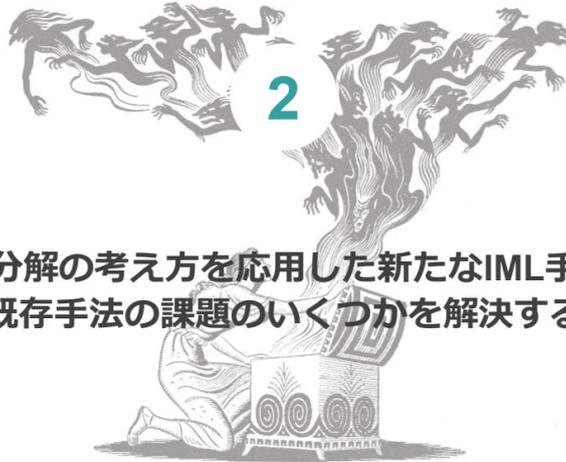
交互作用の可視化手法の課題（解釈性・計算量）

- さらに、H-統計量、SHAP いずれも計算量に課題がある
- H-統計量は2次元 PDP の計算を含むが、2変数の組み合わせ1組に対して2次元 PDP の計算だけでも“(インスタンス数) × (2次元のグリッド数)”個の予測値が必要であり、説明変数のすべての組み合わせで計算すると計算量が飛躍的に増大
- SHAPの計算は1インスタンスに対して (インスタンス数) × 2^(説明変数の数) 個の予測値が必要*であり、グローバルな手法として用いる場合は計算量が極めて大きい
 - ランダムフォレストやブースティング木に限り、Tree SHAPを用いることである程度高速に計算可能なものの、予測モデルが複雑な場合はそれでも計算に時間がかかる。

(*) 変数を入力しない場合の予測値として実データを用いた平均値をとる (Kernel SHAP) 場合の計算量。なお、SHAP Interaction Value の計算にはさらに多くの計算が必要になる。

データサイエンス関連基礎調査WG | 14

H-統計量、SHAPのどちらも、詳しい説明は割愛しますが、計算量が多い手法だといわれております。



関数分解の考え方を応用した新たなIML手法が 既存手法の課題のいくつかを解決する

データサイエンス関連基礎調査WG | 15

次に、「MIDという手法を使って、この課題を解決していこう」というパートに入ります。

関数分解の枠組み

- d 変数関数 $f(x_1, \dots, x_d)$ に対して関数の集合 $\{f_j\}_{j \in \{1, \dots, d\}}$ が以下の条件を満たすとき、後者を前者の**関数分解**という

<条件>

- $J \subseteq \{1, \dots, d\}$ に対して、 f_J は x_J 、つまり $\{x_j\}_{j \in J}$ の $|J|$ 変数関数
- $f(x_1, \dots, x_d) = \sum_{J \subseteq \{1, \dots, d\}} f_J(x_J)$ が成立

- 予測関数をこのように分解することができれば、機械学習モデルによる予測を、切片、各説明変数の**主効果**、説明変数間の**交互作用**などに分析して理解できる

$$\begin{aligned}
 f(x_1, \dots, x_d) &= f_{\emptyset} \quad \leftarrow \text{切片} \\
 &+ f_1(x_1) + f_2(x_2) + \dots + f_d(x_d) \quad \leftarrow \text{主効果} \\
 &+ f_{1,2}(x_1, x_2) + f_{1,3}(x_1, x_3) + \dots + f_{d-1,d}(x_{d-1}, x_d) \\
 &+ f_{1,2,3}(x_1, x_2, x_3) + f_{1,2,4}(x_1, x_2, x_4) + \dots \quad \uparrow \text{2次の交互作用} \\
 &\vdots \\
 &+ f_{1,\dots,d}(x_1, \dots, x_d)
 \end{aligned}$$

データサイエンス関連基礎調査WG | 16

まず、「関数分解」という用語について最初に説明します。下側の等式を見ていただきたいのですが、左辺が、われわれが解釈したい予測モデルです。ここでは d 変数関数ですね。これを右辺のように分解します。まず、切片という関数がない定数項があって、次に 1 変数関数の和として、 f_1 、 f_2 があって、次に、2 変数関数の和、3 変数関数の和……、 d 変数関数の和、といった形で分解する。こうすることによって、 f_1 や f_2 と書いている 1 変数関数のところが主効果である、次の $f_{1,2}$ というところが X_1 と X_2 の交互作用であるという感じで、分かりやすく効果を分解できます。

これだけだとあまり意味がありませんが、そこに幾つかの条件を加えることで、意味のある手法としたものが MID です。

MID (Maximum Interpretation Decomposition) の定義

- 岩沢・松森 (2024) が提案する MID (Maximum Interpretation Decomposition) は、次の条件によりブラックボックスモデルの予測関数 f を分解する手法

<条件>

- 予測関数 f に対する、 k 次以下の項と残差を表す $f_{1,\dots,d}$ からなる関数分解である。

$$f(x_1, \dots, x_d) = \sum_{J \subseteq \{1, \dots, d\}, |J| \leq k} f_J(x_J) + f_{1,\dots,d}(x_1, \dots, x_d)$$

- そのような関数分解で、平均2乗残差 $E \left[\left(f_{1,\dots,d}(X) \right)^2 \right]$ が最小。
- かつ各項に中心化されている。すなわち $J \subseteq \{1, \dots, d\}, |J| \leq k$ および $J' \subsetneq J$ に対し

$$E[f_J(x_J) | x_{J'}] = 0$$

- 上記で一意に定まらない場合は、そのうち $\sum_{J \subseteq \{1, \dots, d\}, |J| \leq k} E \left[\left(f_J(x_J) \right)^2 \right]$ が最小となるもの。

- k 次までの項の分解であることを明示するときは「 k 次 MID」と呼ぶ。
- “Maximum Interpretation” は「予測値と関数分解の平均2乗残差が最も小さい」という意味。

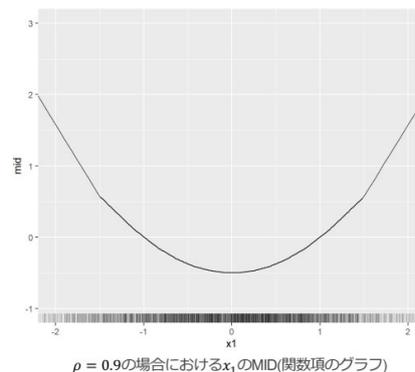
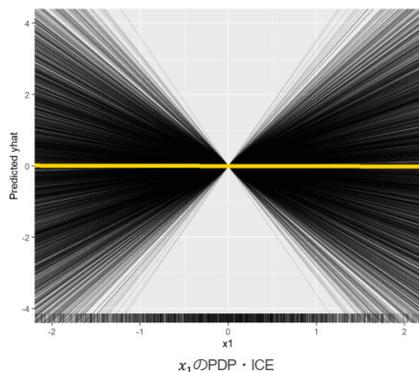
データサイエンス関連基礎調査WG | 17

条件を4点書いているのですが、まず1点目が、 k 変数以下の項で表して、最後に残差があるというような分解であるということです。2点目は、そのような中で平均二乗残差が最小になる。つまり、よくフィッティングしているということです。3点目は、中心化されている。つまり、ある条件付き期待値が0になるということです。これによって、例えば、交互作用の項に主効果が入るのを防ぐことができます。ここまでの条件では一意に定まらない場合があるので、そのような場合は二乗平均が最小になるもの。これをMIDといいます。 k 変数以下の項の分解であることを明示するときは、 k 次 MID といいます。

PDPの問題点 (独立性の仮定) の解決

- PDPは説明変数間の相関が強い場合に主効果を捉えることが出来なかったが、MIDではそのような場合でも主効果を捉えることができる

- PDPの頁の2変数関数の例では、以下のように放物線の形のグラフが得られる。



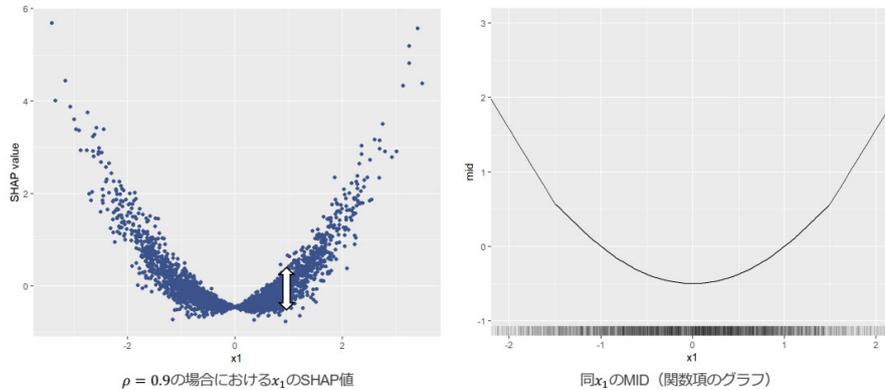
(*1) 説明変数が独立である場合、PDPとMIDの主効果は一致

データサイエンス関連基礎調査WG | 18

まず、このMIDを導入することで課題がどのように解決されるかを見ていきます。PDPでは、先ほどの例だと放物線の形にならず直線になっていましたが、MIDだと放物線の形になってくれます。

SHAPの課題（説明変数と効果の非対応）の解決

- SHAP では、説明変数の値が全く同じでもインスタンスによって測定される主効果が異なっていたが、MIDの場合は主効果がインスタンスに依存せず**理解しやすい**

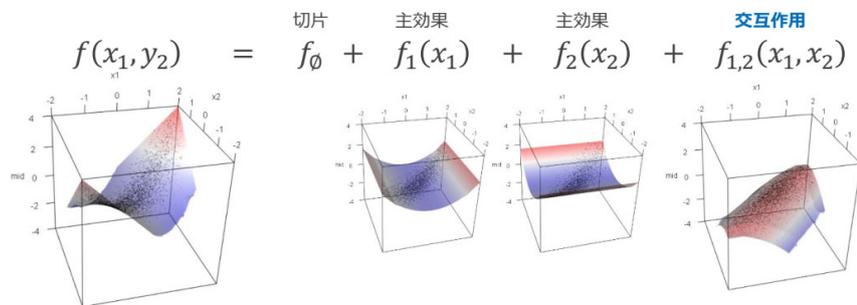


データサイエンス関連基礎調査WG | 19

次に、SHAPの上下幅があるという問題点も、MIDでは関数で分解するので、当然ながらありません。

交互作用の可視化手法の課題（計算量など）の解決

- 説明変数の相関が強い場合、PDPは主効果を捉えなかったが、MIDは主効果が捉えているため、MIDで計測される**交互作用はH-統計量よりも妥当なものになる**
- SHAPによる交互作用の解釈は難しいが、**MIDは関数分解に由来している**ので、交互作用が持つ意味合いが**わかりやすい**



(*) 中心化制約 (f_1, f_2 は期待値ゼロ、 $f_{1,2}$ は x_1, x_2 条件下の条件付期待値が常にゼロ) により関数を特定。

データサイエンス関連基礎調査WG | 20

最後に、交互作用の可視化の仕方です。H-統計量については、説明変数間の相関が強い場合にはPDPが主効果を捉えられないという問題がありましたが、MIDはそのような場合でも主効果が捉えられるので、計測される交互作用はH-統計量より妥当なものになっていると考えられます。

またSHAPの場合は交互作用の解釈の仕方が若干難しいところがあったのですが、MIDの場合は、「このように関数分解したら、残りがこれですよ」という感じで表示しているので、意味が分かりやすいです。

交互作用の可視化手法の課題（計算量など）の解決

- Rパッケージ **midr**（後述）の実装では、MIDは単純な線形回帰と同様の計算によって求められ、H-統計量や SHAP と比べて**計算量は抑えられる**傾向にある
 - ▶ 実行時間は利用するパッケージ、インスタンス数、特徴量の数、各手法のパラメータなどにも依存。

（参考）説明変数が9個のデータセットでrangerによるランダムフォレストモデルを解釈した場合

手法	R パッケージ	インスタンス数 (件)	計算時間 (秒)
2次のMID	midr	10,000	約17
(参考) 1次元ALE	iml	10,000	約14
(参考) 1次元PDP	iml	10,000	約420
H-統計量	vivid	300	約110
SHAP(TreeSHAP)	treeshap	300	約150
SHAP(KernelSHAP)	kernelshap	300	約610
SHAP Interaction Value	treeshap	300	約1600

(*)発表者のPCにおける実測値に基づく。詳細な条件はAppendix.1を参照。

データサイエンス関連基礎調査WG | 21

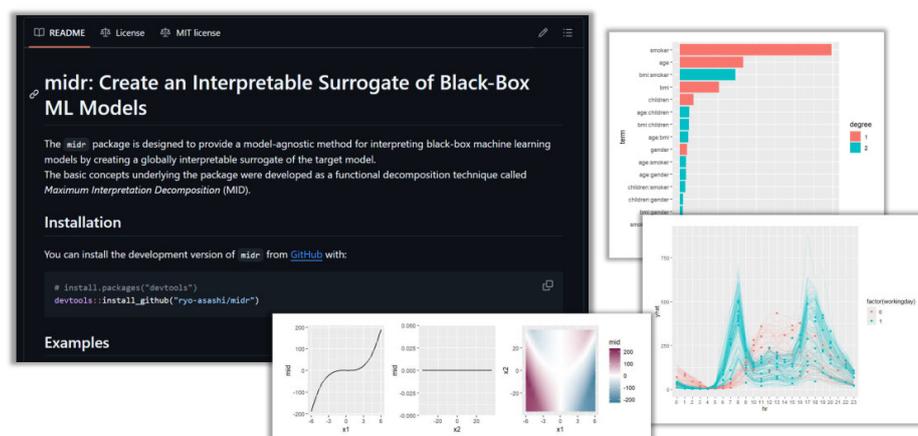
既存の交互作用の可視化手法には計算時間がかかるという課題もあったため、計算時間についても少し触れます。MIDは、解釈したい予測関数の予測値にフィッティングするような考え方で計算するため、下の表を見ていただきたいのですが、例えばH-統計量やSHAPで100秒ぐらいかかるようなデータであったとしても、2次のMIDの場合は20秒ぐらいで計算できたりします。

ここで、Rパッケージ「midr」というものが登場しました。今までMIDの理論的なところを中心に紹介しましたが、「実はRパッケージとして実装して、もう使えるような状態になっていますよ」ということが、本日WGの成果として報告したい内容になっております。

ここから、浅芝さんにバトンタッチいたします。

パッケージ midr の開発

- 任意の予測モデル・予測関数のMIDを計算するためのパッケージ **midr** を開発中
GitHub: <https://github.com/ryo-asashi/midr>



データサイエンス関連基礎調査WG | 22

浅芝 それでは、ここからは浅芝が伝えさせていただきます。

ここまでの発表で共有させていただいた関数分解手法MIDについて、私たちのチームでは、現在、任

意の予測モデルに対して関数分解を実行するためのRパッケージ、midr を開発しています。

※ パッケージ midr のウェブサイト：<https://ryo-asashi.github.io/midr/>

参考資料

数値計算による関数分解を実行する①

- 関数 `interpret()` を用いて制約付き最小二乗法による当てはめを実行する

- データ $\{(x^{(i)})\}_{i=1}^n$ のもとで目的変数 $\{y^{(i)}\}_{i=1}^n$ に対する残差平方和 $\sum_{i=1}^n (y^{(i)} - \sum_{j \in \Lambda} f_j(x_j^{(i)}))^2$ が最小になるように各関数項 $f_j(x_j)$, $j \in \Lambda$ を当てはめる。ただし、 Λ は説明変数集合の部分集合族。
- 「データの分布のもとで各関数項が中心化制約を満たす」という制約条件付きの最小二乗問題を解く。
- 定義域が離散的でない関数項は、データの存在範囲から標本点をとって折れ線関数を当てはめる。

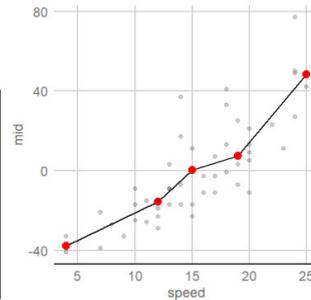
- データセット `datasets::cars` を用いた計算例

- 停止距離 (dist) と速度 (speed) の関係に関するデータセット。
当てはめた関数は $\hat{y} = 42.98 + f_{\{speed\}}(x_{\{speed\}})$ で $f_{\{speed\}}$ は右図。

```
Call:
interpret(formula = dist ~ speed, data = cars, k = 5)

Intercept: 42.98

Main Effects:
---
$speed
  4      12      15      19      25
-37.71627 -15.72081  0.28928  7.40400 48.47759
```



データサイエンス関連基礎調査WG | 23

Midr パッケージで中心的な役割を果たすのは、`interpret()` という関数です。この関数は、(1) 目的変数の値と、(2) 関数分解の関数値 (主効果と交互作用の和) の差をできるだけ小さくするように最小二乗法による当てはめを行います。ただし、関数分解の各項の平均が 0 になるという制約条件があるので、それを満たすように、制約条件付きの最小二乗法を解きます。また、変数の定義域が連続的な場合には、全部の点を評価することができないので、近似的な方法として折れ線関数を当てはめるようにしています。

たとえば、`datasets` パッケージに含まれている `cars` というデータを用いて、唯一の説明変数であるスピード (speed) を元に制動距離 (distance) を当てる回帰タスクを考えると、スピードの主効果として、23 ページの右下の図のような折れ線が当てはめられます。少し薄く載っているのですが、灰色の点はデータから制動距離の平均の分だけ上下にずらして表示したものです。図からわかるように、mid の平均が 0 になるように当てはめが行われています。

数値計算による関数分解を実行する②

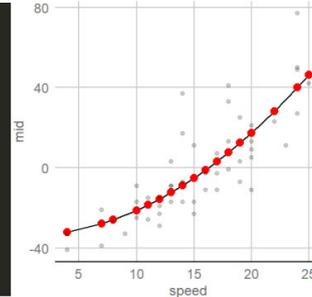
- 関数 `interpret()` を用いてMIDによる関数分解を実行する
 - 分解対象とするモデルを引数 `model` に渡すと、目的変数の代わりに対象モデルが出力する予測値 $\{\hat{y}_i\}_{i=1}^n$ との残差平方和を最小化するように学習を行う。
- 多項式回帰モデルを用いた計算例
 - $y = \beta \cdot x_{speed}^2 + \alpha$ という前提で学習させた多項式回帰モデルの予測関数を切片と速度の主効果に分解。なお、対象モデルの予測関数は $\hat{y} = 0.129 \cdot x_{speed}^2 + 8.86$ 。関数分解による主効果は下のグラフの通り。

```
Call:
interpret(formula = yhat ~ speed, data = cars, model = model,
k = 0)

Intercept: 42.98

Main Effects:
---
$speed
  4      7      8      9      10     11     12
-32.0565 -27.8005 -25.8660 -23.6735 -21.2231 -18.5147 -15.5485
 13     14     15     16     17     18     19
-12.3242 -8.8421 -5.1020 -1.1040  3.1520  7.6659 12.4377
 20     22     23     24     25
17.4675 28.3009 34.1045 40.1660 46.4855

Uninterpreted Rate: 0
```



データサイエンス関連基礎調査WG | 24

`Interpret()` 関数では、引数として予測モデルを渡すと、観測された目的変数の代わりに、予測モデルが出力する予測値を使って同様の当てはめを行います。

先ほどと同じ `cars` データセットに対して、たとえば、線形回帰を実行するための `lm()` 関数で2次の多項式回帰を行ったモデルを構築して、さらに、そのモデルに対して `interpret()` 関数を実行すると、データの目的変数ではなくて、多項式回帰モデルの予測値に対して当てはめが行われます。そして、その場合には、スピードの主効果は、24 ページの右下の図のように2次関数の形に変化します。

数値計算による関数分解を実行する③

- データセット `ISLR2::Bikeshare` に対する `ranger` モデルを対象とする実行例
 - 目的変数である「自転車レンタル数」を、月・時間・平日/休日・天気・気温・湿度・風速の7変数で説明するランダムフォレストモデルを構築し、目的変数の代わりにモデルの予測値を用いてMIDによる分解を実行する。
 - 実行結果は、切片、7個の主効果項、21個の交互作用項を持つ。未解釈割合*は約3.21%であり、2次以下の交互作用によって、`ranger` モデルの予測値の全分散の約96.79%が説明される。

```
Call:
interpret(formula = yhat ~ (mnth + factor(workingday) + hr +
weathersit + temp + hum + windspeed)^2, data = train, model = model,
lambda = 0.01, method = 2)

Intercept: 144.83

Main Effects:
7 main effect terms

Interactions:
21 interaction terms

Uninterpreted Rate: 0.0321
```

(*)未解釈割合 = 対象モデルの予測値と当てはめ値の残差の二乗和 ÷ 予測値の分散

データサイエンス関連基礎調査WG | 25

これは全然面白い例ではないですが、もう少し実践的な例として、IMLの文脈で使われることがよくある `Bikeshare` データセットというものに対して、ランダムフォレストモデルを構築して、その予測関数を対象として関数分解を実行してみます。この `Bikeshare` データセットというものは、アメリカの自動車シェアリングシステムにおける1時間ごとの自動車のレンタル数を、およそ2年間にわたって記録したも

のです。ここでは1年間分だけ使うのですが、他の変数には、平日か休日か、気温、湿度、風速などの情報が含まれています。

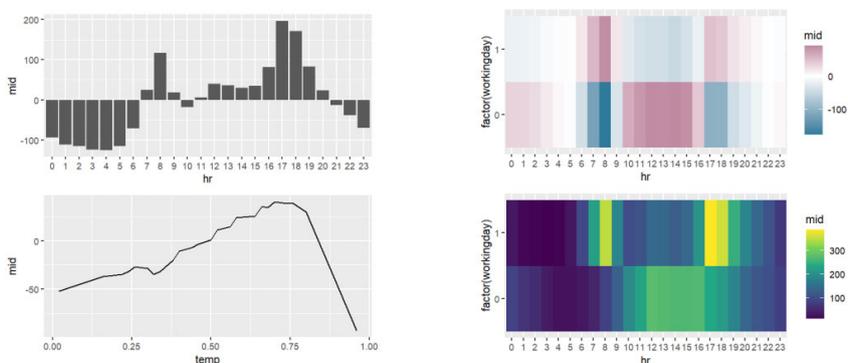
ランダムフォレストモデルは、よく使われるRのパッケージである `ranger` を使って構築します。関数分解を実行したときの出力には、出力画面の最終行に `Uninterpreted Rate` というものが出ております。これは、レンジャーモデルの予測値のばらつきの中で、関数分解における2次以下の項で説明されない部分が占める割合を示しています。ここでは3.21%となっているので、裏を返せば、2次までの交互作用までの関数分解によって、`ranger` モデルの予測値の分散の96.79%が説明されるということを示しています。

参考資料

説明変数の変化が予測値に与える影響を可視化する①

● 関数 `ggmid()` で説明変数ごとの主効果・交互作用を可視化する

- 分解後の関数項は `plot()` 関数や `ggmid()` 関数で可視化できる。下図は、左上が時間の主効果、左下が気温の主効果、右上が時間と平日/休日の交互作用をあらわす。
- なお、右下のように、切片・主効果・交互作用を合算したプロットを作成することもできる。



データサイエンス関連基礎調査WG | 26

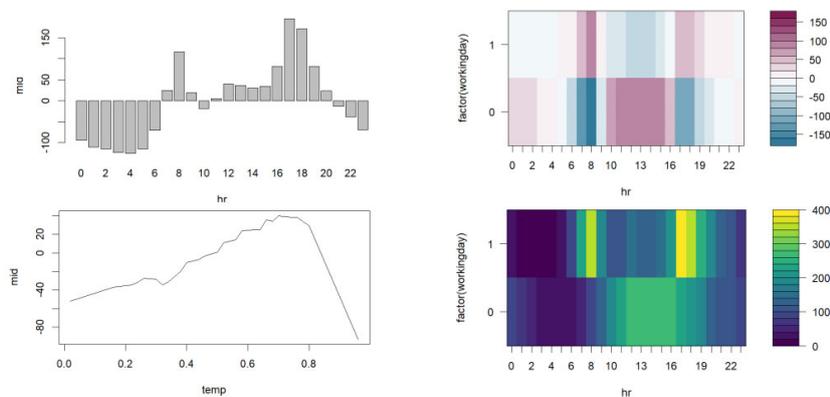
少し駆け足になりますが、`midr` パッケージでは、分解によって得られた主効果や交互作用を可視化するための機能を充実させています。たとえば、`ggmid()` という関数を使うと、関数分解モデルにおける主効果や交互作用を、`ggplot2` パッケージの機能を使って可視化することができます。

参考資料

説明変数の変化が予測値に与える影響を可視化する②

● 関数 `plot()` で説明変数ごとの主効果・交互作用を可視化する

- Rの基本的なパッケージ (`base` と `graphics`) を用いたグラフィックスを作成することもできる。



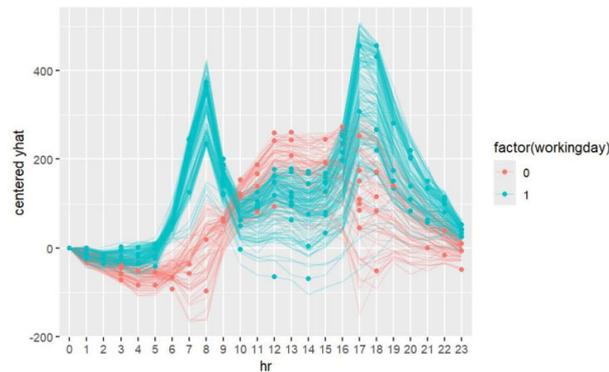
データサイエンス関連基礎調査WG | 27

同様に、plot() 関数を使うと、Rの標準的なパッケージである graphics パッケージを使ってプロットを作成することができます。

参考資料

説明変数の変化が予測値に与える影響を可視化する③

- 関数 mid.conditional() で説明変数の値を変化させたときの影響を計算・可視化する
 - 特定の説明変数の値を変化させたときの予測値の変化をインスタンスごとに計算する。
 - 計算結果に可視化関数 ggmid() や plot() を適用すれば、Individual Conditional Expectation (ICE) プロットを描くことができる。引数の指定によって、起点の中心化や他の変数による色分けなども可能。



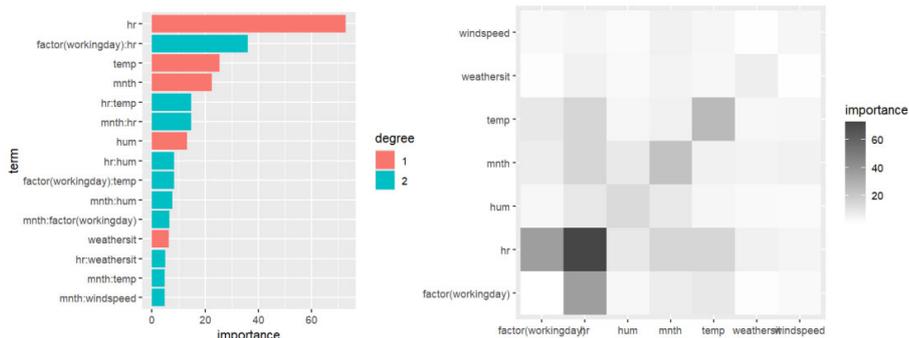
データサイエンス関連基礎調査WG | 28

そのほか、予測モデルの解釈をサポートするための機能として、まず、ICEプロットを作成する機能を実装しております。このプロットでは、説明変数のうちで時間だけを変えていったときに、自転車のレンタル数の予測値がどのように変化するというものが、個々のレコードごとにプロットされています。色は、平日か休日かで分けていて、このプロットから、平日か休日かによる自転車レンタルの利用数の変化のパターンの違いを見て取ることができます。

参考資料

説明変数の重要度を可視化する①

- 関数 mid.importance() で関数項ごとの重要度を計算・可視化する
 - 関数項 $f_j(x_j)$ の重要度を平均的な効果の大きさ $\sum_{i=1}^n |f(x_j^{(i)})| / n$ などの尺度で計算する。
 - 計算結果に可視化関数 ggmid() や plot() を適用すれば、棒グラフやヒートマップを描くことができる。



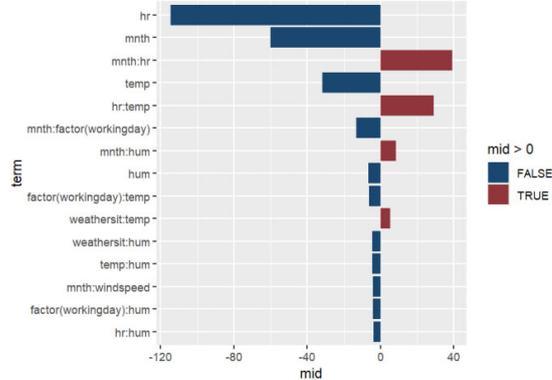
データサイエンス関連基礎調査WG | 29

さらに、主効果や交互作用ごとの項ごとの重要度を棒グラフやヒートマップで可視化する機能も実装しています。目的変数が多いときには、右のようなヒートマップが便利ではないかと思っています。

説明変数の重要度を可視化する②

● 関数 `mid.importance()` で個別のインスタンスの予測値を分析する

- 引数 `data` に1件のみのデータフレームを渡すと、特定のインスタンスにおける各関数項の効果 $f(x_j^{(i)})$ を用いて棒グラフを描くことができ、SHAP プロットのように用いることができる。



データサイエンス関連基礎調査WG | 30

また、これを応用して、個別の予測についても、各項の値を表示することで、主効果、交互作用ごとの予測値への寄与を可視化するSHAPと同じような使い方ができる機能も実装しています（注：バージョン0.4.7.900以降の `midr` パッケージでは、個別の予測を分析する機能は `mid.breakdown()` 関数に移されています）。

パッケージ `midr` の利用方法

● 開発中バージョンのパッケージの利用方法

- 開発中の `midr` パッケージ（当資料作成時点のバージョンは0.2.5）は、以下のコードでGitHubリポジトリからインストールすることができます。
`devtools::install_github("ryo-asashi/midr")`
- IMLの手法として（また、単体の予測モデル構築手法として）お試しいただき、改善のためのご意見等をいただけたら幸いです。なお、典型的な場合について理論値との一致を確認しているものの、不具合はまだまだ残されているかもしれません。また、計算速度や使い心地などは継続的な改良を予定しています。
- 基本的な使用方はGitHubリポジトリのトップページをご確認ください。また、個々の主要な関数の使い方は、`?関数名` または `help(関数名)` でご確認ください。

データサイエンス関連基礎調査WG | 31

最後になりますが、開発中のパッケージはGitHubリポジトリからインストールしていただくことが可能です [`devtools::install_github("ryo-asashi/midr")`]。もしよろしければ、`midr` パッケージを何かの際にお試しいただいて、改善のためのご意見をいただければ非常にうれしく思っております。

以上で、私からのご報告を終わります。ご清聴ありがとうございました。

藤田 上妻さん、浅芝さん、ありがとうございました。

それでは、質疑応答に移りたいと思います。Slido から一つご質問をいただきました。後半のお二方にご質問ということですが、「今回の手法、midr パッケージ等をアクチュアリー実務で活用できた事例があれば、ご紹介いただけますでしょうか」というご質問をいただいております。

浅芝 ありがとうございます。

率直にお答えしてしまうと、パッケージ自体ができたばかりで、なかなか実務のデータに対して実験的なことをしてみるということは、十分にはできていないというのが現状です。なので、ここからは、まずは、先ほどの発表でもあったようなオープンデータの保険関係のデータを使って、midr を使って分析し、実務にどのように役立てていけそうかということも、研究を進めていきたいと考えています。

藤田 ありがとうございます。

できたてほやほやのパッケージということで、ご質問いただいた方にも、ぜひ使っていただきたいと思っています。

他にご質問のある方は、いらっしゃいますか。

Slido からのご質問です。「上妻さんにご質問です。関数分解のそれぞれの関数は、多項式とは異なるのでしょうか。どのような分解になるイメージか教えていただきたく。また、関数分解の一意性は担保されるのでしょうか」というご質問をいただいております。

上妻 ご質問ありがとうございます。

まず理論的な話をしますと、各関数の形については特段条件がなく、その形がどうなるかは分からないということになります。一意性については、4点目の条件によって担保されると証明されています（注：各関数には可積分性に関する条件が課されています。また、ここでの一意性はいわゆる almost everywhere の意味です。一意性の証明を含む厳密な議論については、岩沢 宏和 & 松森 至宏 (2024)『ブラックボックスモデルの解釈を最大化する関数分解手法』（日本保険・年金リスク学会研究発表大会）を参照してください。）。

一方、midr パッケージでの実装は少しだけ事情が違っておりまして、連続変数に対しては、幾つかの値で区切ってビンニングしており、各関数は折れ線のような形になっています。

このような回答でよろしいでしょうか。

藤田 ありがとうございます。

Slido で新しい質問をいただきました。浅芝さんへのご質問になります。「25 ページで解釈できている割合が十分に高いように感じます。そのようになった要因はあるのでしょうか。多くの場合で、このような結果が期待されるのでしょうか」というご質問をいただいております。

浅芝 ありがとうございます。

97%となると、かなり高いのですが、実際、データに対して、どのように反応変数が関連しているかが一番重要で、2次の交互作用までを使えば十分説明ができてしまうようなタスクであれば、それに対して、ランダムフォレストや、あるいは、勾配ブースティングマシンなどで作った複雑なモデルであっても、結

局、予測値と説明変数の関係を、せいぜい2次の交互作用までで捉えることができ、midr で分析すると、大体、それがちょうど可視化できる形で出てきて、しかも、予測モデルの予測値と同じ値を、midr の関数分解によっても出力できる、という感じになります。ただ、3次以上の交互作用が本質的になっている場合には、精度の高い予測モデルに対する未解釈割合が小さくなっていかないということが考えられ、たとえば50%に満たないということも普通にあると思います。

藤田 ありがとうございます。お時間も近づいてまいりましたので、最後、Slido からいただいているご質問を、お願いいたします。こちら浅芝さんです。「midr パッケージに関する質問ですが、説明変数はカテゴリカル変数でも使用できるのでしょうか。その際、可能な場合には何か注意点等はございますか。」というご質問になります。

浅芝 ありがとうございます。

端的にお答えすると、可能です。注意すべき点としては、カテゴリ変数の場合には、実装上、すべての水準を評価の対象点に含めるということになっています。そうすると、たとえば、100 個水準があるときに、交互作用を考えると、他の変数のすべての水準×100 個の点が全部評価されるので、計算時間が膨大になる、ということが起こりやすいです。そのような、水準の多いカテゴリ変数については、自動的に水準を絞る機能も実装していますので、そのような対策をしていただく必要があるかと思います。

藤田 ありがとうございます。

それでは、お時間になりましたので、上妻さん、浅芝さん、ありがとうございました。

浅芝 ありがとうございました。

藤田 最後に、私から、ご案内させていただきます。

改めまして、本セッションは、いかがでしたでしょうか。

・ 講師陣

前編講師：野村 俊一 早稲田大学大学院 准教授

早稲田大学で統計科学を専門として、データサイエンス関連の授業も担当
 当会2017年度優秀論文賞受賞（統計数理研究所在籍時）
 2021年度早稲田大学ティーチングアワード受賞（受賞科目はデータサイエンスⅡ）

後編講師：岩沢 宏和 早稲田大学大学院 客員教授

当会基礎講座・追加演習講座（損保数理）講師
 2021年Hachemeister賞^(※)受賞（内容はデータサイエンス）
 ※ 損保分野の世界最高峰の賞

・ 充実した講義内容

初学者(若手からベテランまで)向けの体系的なカリキュラムで一通りの知識を習得可能
 独学で学んだ方の学び直しにも活用

・ 個人でも負担しやすい価格設定

受講料は5万円。公益社団法人として会員に提供している講座のため、
 受講料が安い

申込受付中！！
11月12日（火）正午 受付締切
kouza@actuaries.jpへご連絡下さい！

データサイエンス関連基礎調査ワーキンググループでは、このようにアクチュアリー実務を念頭に置いて、実務で使える手法を、独自で研究開発しております。聴講者の皆様におかれましては、このようなセッションを聞いて、「実際に実務で使ってみたい」、「データサイエンスについて、より理解を深めてみたい」という方も多くいらっしゃるかと思います。でも、まず、何から始めたらいいかわからないという方も、いらっしゃると思います。そのような中で、アクチュアリー会として、データサイエンスに関する専門講座を提供しておりますので、こちらに関して、最後、ご紹介させていただきます。

当講座は、早稲田大学の野村先生、岩沢先生を講師陣としてお迎えした、充実した講義内容になっています。具体的な内容としては、次のページです。

データサイエンス専門講座のご紹介 (内容・日程^(注1))



科目	講師	内容	日程
前編 (講義中心)	野村俊一 早稲田大学大学院 准教授	(第1回) Rの導入と簡単な回帰モデル (第2回) 線形回帰モデル (第3回) 主成分分析 (第4回) クラスタリング・時系列解析 (第5回) 決定木 (第6回) 一般化線形モデル (第7回) 正則化・一般化加法モデル	12/2~1/27 (原則月曜日 ^(注2)) ※一部火曜日あり
後編 (実習中心)	岩沢宏和 早稲田大学大学院 客員教授	(第8回) 予測モデリングの基本手順 (第9回) 探索的データ解析 (EDA) (第10回) 予測モデリング用のモデル例 (第11回) モデルの選択・評価の方法 (第12回) 回帰問題での実践 (第13回) 分類問題での実践	2/8~3/15 (原則土曜日 ^(注3))

(注1) 詳細は「2024年度アクチュアリー講座 専門講座(データサイエンス)実施要領」をご覧ください。
 (注2) 講義時間は、5時間。各回の講義の前に、事前オンデマンド配信を視聴いただきます。
 (注3) 講義時間は時間

前編、まず、基礎的な線形回帰モデルや主成分分析、クラスタリング、決定木など、基礎事項を、座学を中心に学んでいただいた後に、後半では、岩沢先生から予測モデリングの基本手順、探索的データ解析、

予測モデルの構築、モデルの選択・評価というような内容をハンズオンで、実際に手を動かして学んでいくという形式になっているので、最高品質の授業になっていると思います。

そして、受講料は5万円という破格の費用になっております。講座は12月から開始になるのですが、申込受付を11月12日まで行っているため、ぜひ、奮ってご応募いただければと思います。

以上、ご清聴ありがとうございました。本セッションは、こちらで終了させていただきます。