

## アクチュアリーとモデル選択 (Accurate GLM)

三井住友海上 小島 睦月君

共栄火災 佐野 誠一郎君

Guy Carpenter Japan, Inc. 藤田 卓君

小島 今、ご紹介にあずかりました、三井住友海上の小島です。どうぞよろしくお願いたします。

2018年度 日本アクチュアリー会 年次大会

### 機械学習手法の紹介と データへの適用

2018年11月9日

ASTIN関連研究会

小島睦月(三井住友海上)

1

私は、第1部として、機械学習の手法の紹介と、それをデータへ適用した場合の結果についてご紹介します。

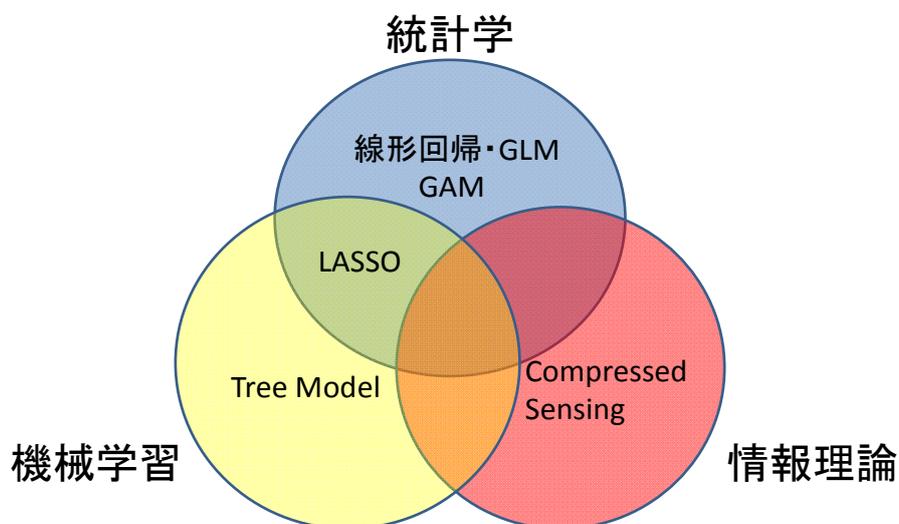
## 機械学習とアクチュアリー

- 近年注目されている機械学習
    - 実務に取り入れる
    - とにかくまずは何なのか知りたい
  - アクチュアリーとして何を重要視するか？
    - モデルの解釈可能性
    - 予測精度
    - 計算速度
- 予測精度とモデルの複雑さのトレードオフ

2

機械学習は近年注目されている分野でして、アクチュアリーの皆様におかれましても、実務に取り入れたいと考えていらっしゃる方や、とにかく何なのかを知りたいと思われる方が多いのではないかと思います。この第1部は、特に後者の何なのかを知りたいと思われている方に向けて、手法を幅広くおよびそれをデータへ当てはめた結果を紹介したいと思います。アクチュアリーとしてモデルに何を求めるかということは、幾つかあると思うのですが、第1部では、特に予測精度とモデルの複雑さ。この二つの間にトレードオフがあるということをご紹介したいと思います。

## 統計学・機械学習・情報理論



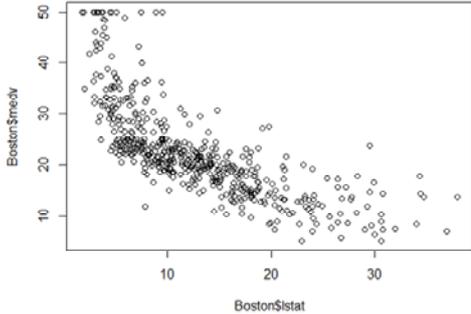
3

まず、機械学習がどのような立ち位置にあるかを示したものが、この図になります。各円はそれぞれの

領域を表していきまして、その中の文字列は、手法やモデルの名前になります。例えば、青色の枠は統計学のところとして、有名なものは線形回帰やGLM。これは、統計学の分野に入ると思います。一方、このあと紹介しますが、LASSOは、青と黄色、つまり統計学と機械学習の中間にあるような手法があります。これらの手法についてこのあと紹介しますが、一つなじみがないかもしれない分野が、情報理論です。この分野は、データの圧縮や通信に関する理論ですが、情報理論の分野も機械学習に大きな影響を与えていますので、一つの手法をこのあと触れたいと思います。

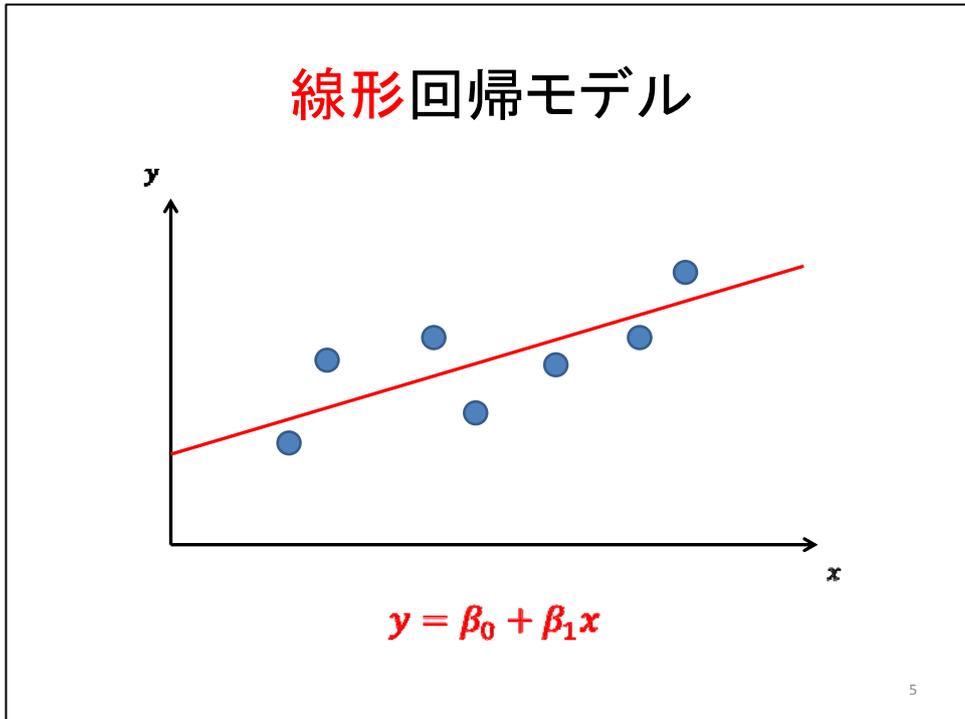
## 回帰モデル

- 二つの量(変数)の関係をデータから推定
  - 自動車保険料と走行距離
  - 経過年数とインカードロス(支払備金)
  - 家賃と低所得者層の割合



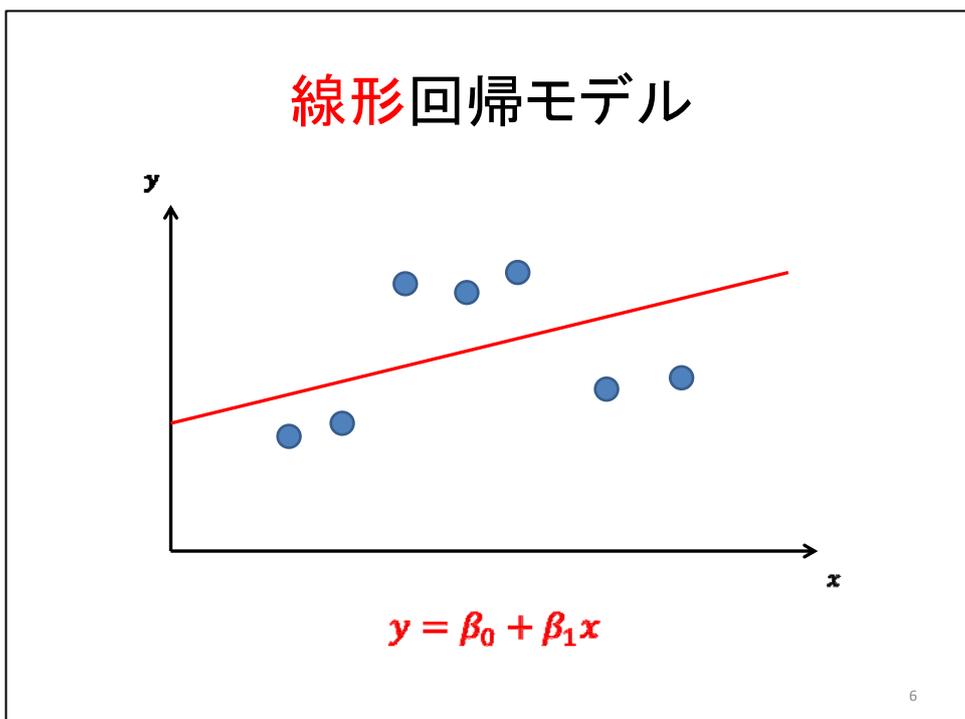
4

モデルや手法といっても、いろいろなデータを相手にしますので、今回の発表では、特に回帰モデルに限定して話をしたいと思います。回帰とは何かというと、二つの量の関係をデータから推定することです。例えば、自動車保険料と走行距離の関係、経過年数とインカードロスの関係。また、これは保険ではありませんが、家賃と低所得者層の割合の関係。これらの間の関係を推定するものになります。一番下の絵は、最後の例についてプロットしたもので、横軸が低所得者層の割合、縦軸が家賃の高さになりますが、ご覧いただけるとおり、低所得者層の割合が多い地域ほど、家賃は低い傾向にあると見ることができます。

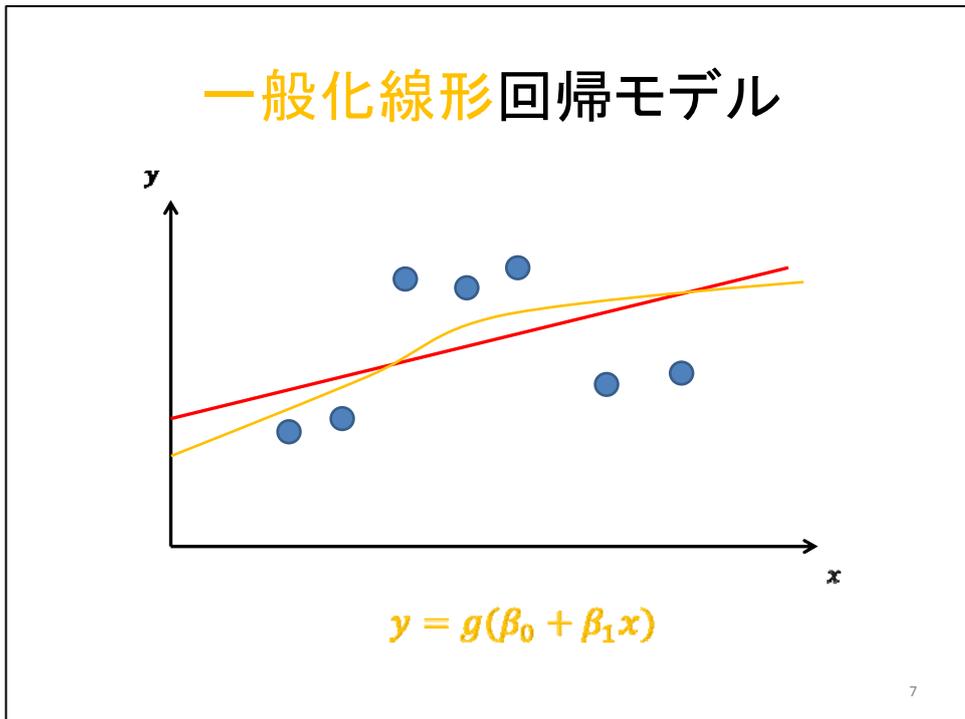


さて、回帰モデルで最も有名かつシンプルなのが、線形回帰モデルと呼ばれるものです。これは、青色のようなデータに対して直線を当てはめることで、モデルを推定するものです。この絵のように、データが線形の関係、つまり、 $x$ が増えると $y$ も比例して増える場合には、よく当てはまるモデルです。

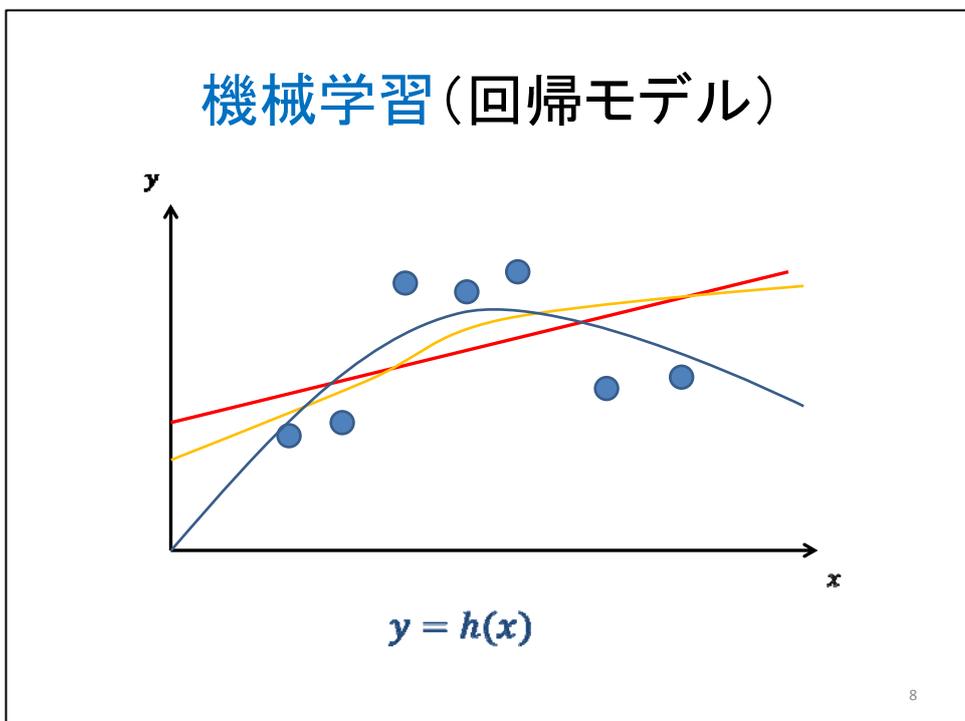
しかし、次のデータのように、 $x$ が増えると、最初は $y$ が増えているのですが、 $x$ がかなり大きいときには $y$ は必ずしも大きくなるというような、線形でない場合、線形回帰モデルを当てはめるにはやや難しい面があります。



そこで、アクチュアリー分野でよく知られている手法、一般化線形回帰モデル (GLM) を当てはめると、このような非線形的な関係を捉えることができます。



これは、 $g$  というリンク関数のおかげで、できていることになります。しかし、依然としてリンク関数は単調性を持つ必要がありますので、 $x$ が増えると  $y$ が増えるという制約は残ります。



一方、機械学習では、さまざまな手法が知られており、このような単調性を持たない、単調性の制約を外したようなモデルも、もちろんたくさんあります。そのため、機械学習を勉強して理解することで、線形回帰や GLM 以外の手法をアクチュアリーが使えるようになるといえます。この青色に分類されるもので

どのような手法があるのかを、このあと紹介します。

## GLMを超えて

1. Sparse Modelling
  - 1) LASSO
  - 2) Compressed Sensing
2. GAM
3. Tree Model
  - 1) Decision Tree
  - 2) Bagging
  - 3) Random Forest
  - 4) Boosting

9

大きく今回は三つ紹介したいと思います。一つめはスパースモデリングと呼ばれる手法で、二つめが GAM、三つめが決定木モデル。英語でツリーですね。ツリーモデルについて、紹介したいと思います。

## LASSO

- スパース(疎、0の多い)解を生成
- 回帰係数の計算が容易(凸最適化)

$$\text{Minimise } \beta \quad \|y - x\beta\|_2^2 + \lambda \|\beta\|_1$$

11

まず、一つめのスパースモデリングから紹介します。スパースは、日本語では「疎」という意味で、どのような意味かという、回帰係数  $\beta$  の一部または多くが 0 というようなものを、スパース、もしくはスパース解と呼びます。特にスパース手法として有名なものが、LASSO です。これは、真ん中の式で定義されるような回帰係数  $\beta$  のことを指していて、 $y$  と  $x$  の間の二乗誤差に対して回帰係数  $\beta$  の制約を加える、

この二つの和を最小化するようなものが LASSO と呼ばれます。LASSO は、スパースな解を生成しやすいということが、よく知られています。

これを説明したものが、下の絵になります。青色の部分が二乗誤差になっておりまして、中心に行くほど青色の部分が小さくなりますが、中心は、線形回帰モデル  $\beta$  の最尤推定量と等しくなります。全く赤色の部分がないと、線形回帰と同じになるのですが、LASSO は、この赤色の制約を課します。この制約は、このような立方体になっておりまして、尖っていますので、青と赤を両方とも小さくするような部分が LASSO 解となります。尖っているおかげで、必ずしも 100% ではないのですが、この例ですと、 $\beta_2$  が 0 のような角に当たることで、スパース解を生成しやすいと説明できます。

## データ適用例

- メジャーリーグ野球選手の年俸データ
- 年俸と選手のデータ(19変数)の計20変数
- RのISLRライブラリより

		線形回帰	LASSO
AtBat	打席数	-1.6	-1.5
Hits	ヒット数	7.0	5.7
HmRun	ホームラン数	4.1	0.0
Runs	得点	-2.4	0.0
RBI	打点	-1.0	0.0
Walks	四球	6.2	4.8
Years	年数	9.5	-9.4
CAtBat	累積打数	-0.2	0.0
CHits	累積ヒット数	1.0	0.0
CHmRuns	累積ホームラン数	-0.2	0.6
CRuns	累積得点数	1.5	0.7
CRBI	累積打点	0.8	0.4
CWalks	累積四球数	-0.8	-0.5
League	リーグ	87.0	32.0
Division	地区	-98.0	-118.0
PutOuts	刺殺	0.2	0.3
Assists	捕殺	0.3	0.2
Errors	エラー	-2.0	-2.0
NewLeague	次年度リーグ	-23.0	0.0

13

実際にデータに当てはめた例を紹介したいと思います。当てはめるデータは、野球のデータです。これはメジャーリーグのデータですが、各選手の年俸と、その選手の打率や打点のようなデータとの関係を調べようとしているデータです。これは、統計ソフトの R で利用できます。

当てはめた結果が、こちらになります。左側 2 列が、変数の名前ですね。その右側 2 列が、線形回帰および LASSO を当てはめたときの、各変数の回帰係数の値になります。ご覧いただけるように、LASSO につい

では、黄色の変数の回帰係数が0となっています。0とするのがスパースという意味です。また、解釈としては、この0という所は、効果を見なくてもいい。つまり、これらの変数は、打者の年俵の予測や推定には不要であると考えられます。従って、線形回帰と比べて、例えば変数が多い場合には、一部の変数の効果を見なくてもいいとすることができるので、線形回帰よりも、変数を見なくてもいいという意味で、分かりやすいということになります。これが、LASSO の特徴です。

## Compressed Sensing

- スパース(疎)な解を生成
- 計算(最適化)が困難(NP-hard)だが、種々の計算アルゴリズムが提案され、その理論的性質が調べられている

$$\hat{x} = \operatorname{argmin} \|x\|_0 \text{ s. t. } \|y - x\beta\|_2^2 = 0$$

14

また、スパースモデリングの別の手法として、compressed Sensing という手法があります。これは、先ほど紹介したとおり、情報理論の分野の手法として、よりシンプルに、回帰係数の非ゼロ要素を小さくする手法、最適化の問題です。深くは触れませんが、この分野で研究されている手法を機械学習に応用したものも幾つかあって、大きな影響を与えています。

## Generalised Additive Model

$$\text{GAM } y = f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

$$\text{線形回帰 } y = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

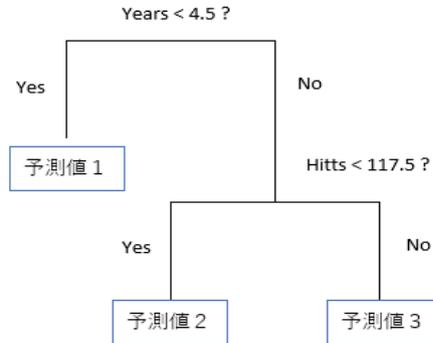
- データの非線形な関係を捉えられる
- 各変数の効果がわかり易い
- 説明変数が多い場合、推定が難しい

16

さて、三つのうち二つめですが、GAM について説明します。Generalised Additive Model と呼ばれるもので、ある意味で線形回帰を拡張したモデルとなっています。真ん中の 2 式目に線形回帰の式がありますが、線形回帰の場合は  $\beta \times x$  の線形結合になっていますが、GAM はそれを拡張して、 $x_1$  に対して  $f_1$ 、 $x_2$  に対して  $f_2$  というように、非線形な関数  $f_1$  や  $f_2$  をかませるという手法になっています。このため、データの非線形的な関係を捉えることができますし、線形回帰と同様、additive、つまり加法モデルですので、 $x_1$  の効果 +  $x_2$  の効果という形になっており、各変数の効果が分かりやすいということがいえます。ただし、 $f_1$  や  $f_2$  という関数を一つ一つ推定しますので、特に変数が多い場合は、推定が難しいということが知られています。ただ、このような線形回帰の拡張、線形性の拡張は、GAM の大きなメリットといえます。

# Tree Model

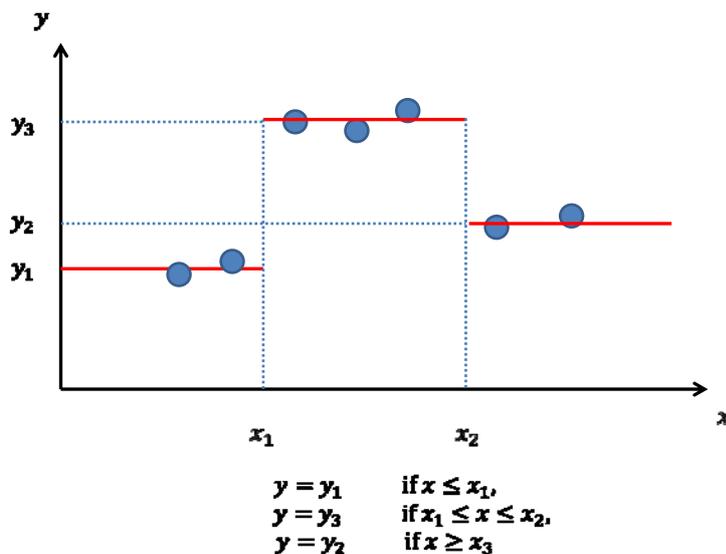
- 決定木モデルは説明変数の値に基づき木を構成



18

大きな三つめですが、決定木モデルを紹介したいと思います。この絵のように木を構成するのですが、どのように構成するかということ、野球のデータを例に説明します。まず、各打者の在籍年数が 4.5 年より大きい、小さいかで、イエスとノーにグループを分けます。イエスのグループは、一つのグループとして予測値を与えますが、ノーのグループは、ヒット数が 117.5 本より大きいかどうかで、更にイエスかノーの二つのグループに分けます。この例ですと、合計三つのグループに野球の選手を分けて、各グループの選手に予測や推定を当てはめるというモデルです。非常にシンプルで、分かりやすいかと思います。

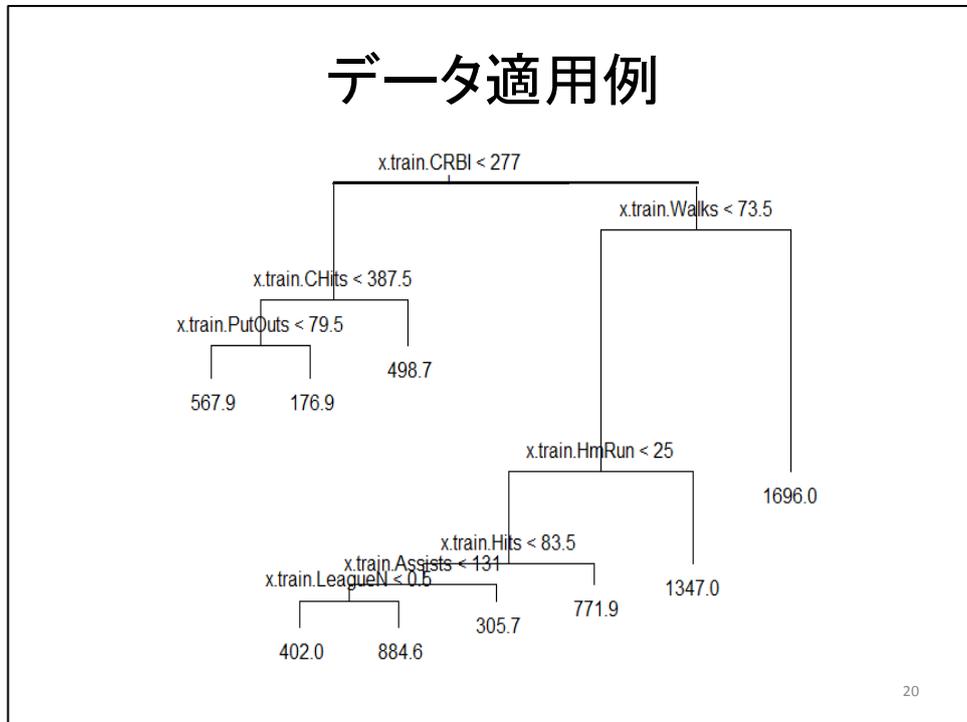
## Tree Model(1次元)



19

また、特に1次元の場合は、この絵のように階段の関数になります。つまり、 $x$ の値が、例えば  $x_1$ より

大きいのか、小さいのか。また、 $x_1$ より大きいとすると、 $x_2$ よりも大きいのか、小さいのかで、三つのグループに分かれています。そして、分かれた各グループ、データに対して、この例ですと、そのグループ内の平均、赤線を当てはめることでモデルを構成しています。この例ですと、タリフや等級制度に近いのかなと思っけていまして、 $x$ の値の大小によってグルーピングしているということになっています。

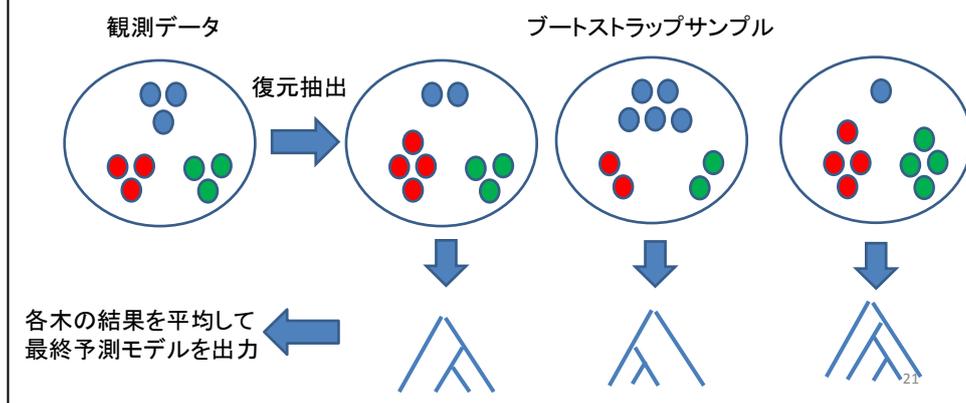


これを野球のデータに当てはめた結果を紹介します。やや変数が多いので複雑には見えますが、まず、一番上をご覧ください。CRBI と書いてありますが、これは累積打点ですね。累積打点が 277 より大きいのか、小さいのかで、大きく二つのグループに分かれています。例えば左のグループに行くと、さらに CHits、つまり累積ヒット数が 387.5 よりも大きいのか、小さいのかで分かれます。このように、各変数の値でどんどんグルーピングして行って、残った最後の枝に数字がついていますが、このグループに入る人たちの平均で予測をする、推定をするということになります。

このように、決定木モデルは、イエスとノーの分岐でモデルを当てはめますので、解釈についても分かりやすいのですが、このあと見るように、残念ながら予測精度は、あまり良くないということがあります。

## Tree Model - Bagging

- **Bagging (Bootstrap Aggregating)**
- 予測精度を向上させる機械学習の手法



そこで機械学習では、幾つかの工夫をすることで、予測精度を上げを考えています。大きく三つ紹介しますが、一つめは、Bagging と呼ばれる手法です。これは、Bootstrap Aggregating と呼ばれる手法で、文字どおり、ブートストラップを使います。ブートストラップは、平たく言うと復元抽出になります。左の真ん中辺りをごらんいただくと、これが例えば観測データだとします。赤・青・緑の球が三つずつ、計9個ある状態から、復元抽出をします。つまり、重複を許して9個データを選びますので、ブートストラップで選ばれるサンプル、ブートストラップサンプルと呼びますが、9個の間の色の構成割合が毎回違います。例えば、青が2個だったり、青が5個だったり、青が1個だったり、ばらばらです。

Bagging では何をするかというと、このブートストラップサンプルに対して毎回木を当てはめて、毎回ブートストラップサンプルの内容は違いますので、生成される木は違うのですが、それら生成された木を最後に平均化して、予測モデルとするというのが、Bagging になります。つまり、ブートストラップでサンプルを複数のサンプルのコピーを作って、統合するというモデルです。

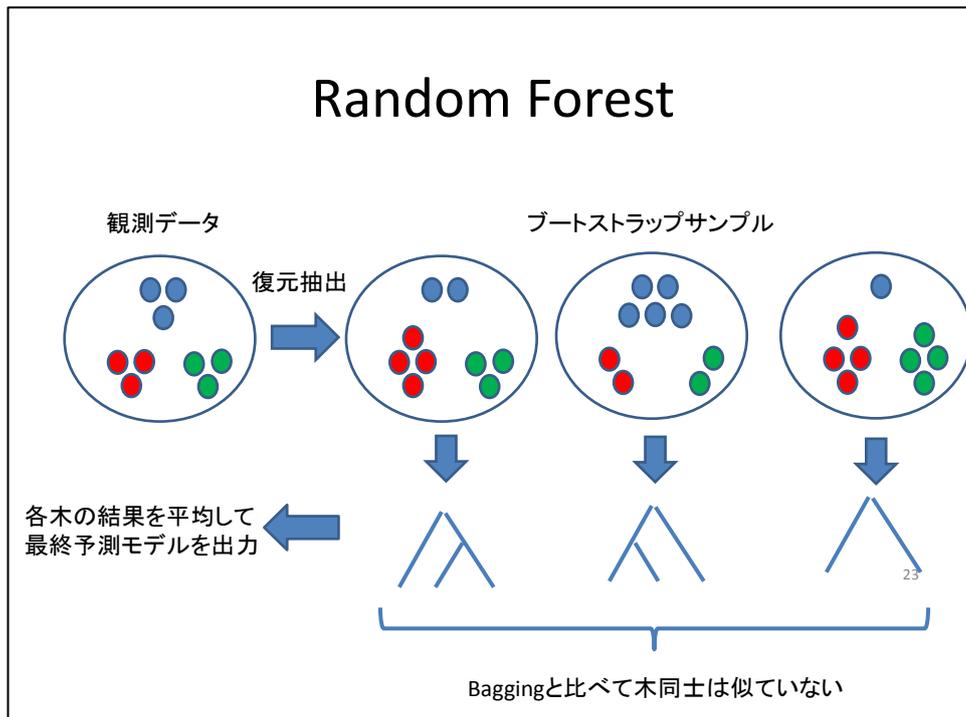
## Tree Model – Random Forest

- 説明変数が多いと木が大きくなりやすい(分割が煩雑になる)
- ランダムに説明変数を少数選び、木を学習することで相関の小さい(似ていない)木を多数生成
- 各ランダムに生成された相関の小さな木の結果を平均して最終予測モデルとする

生成される木同士の相関を弱くし、集団でよりよい予測値の構成

22

これを少し改良したものが、Random Forest という手法です。先ほどの Bagging と似ているのですが、Bagging は毎回ブートストラップサンプルに対して当てはめる木が大きくなると、計算時間もかかりますし何よりも木同士が似てしまうという欠点があります。



そこで Random Forest は、毎回木の学習に使用する変数の数を減らします。このようにすることで、似た木が出ないようにして学習する手法が、Random Forest です。絵で見せると、このような形になります。Bagging には似ているのですが、ブートストラップサンプルごとに学習される木同士は、Bagging と比較して似なくなります。毎回出てくる木の相関を小さくして、学習を効率的にします。つまり、木どうしが似

ている場合は、平均しても同じものしか出てきませんので、あえて違う木が出てくるようにしているものが Random Forest になります。

## Tree Model - Boosting

- 逐次的に「**少しずつ**モデルを構成する」

1. 初期モデルを  $\hat{f}_0 = 0$ ,  $r_0 = y$  とする。
2. 残差  $r_k$  に対して決定木モデル  $\hat{g}_k$  を当てはめる
3.  $\hat{f}_k$  を以下のとおり「**少しだけ**」更新

$$\widehat{f}_{k+1} \leftarrow \widehat{f}_k + \lambda \widehat{g}_k$$

4. 残差も更新

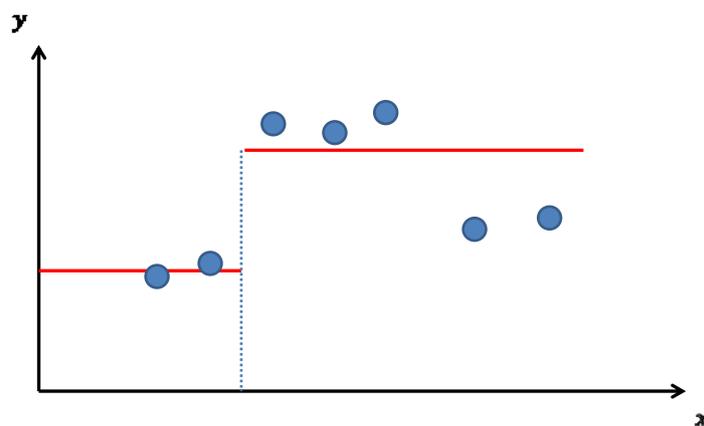
$$r_{k+1} \leftarrow r_k - \lambda \widehat{g}_k$$

5. 最終予測モデルは  $\sum_k \lambda \widehat{g}_k$

24

三つめの工夫は、Boosting と呼ばれる手法です。これは、標語的に言うと、少しずつモデルを構成することになります。Bagging は、ブートストラップサンプルを使って毎回データを変えていたのに対して、こちらは、一つのデータに対して少しずつ木を当てはめるということになります。文字列だけだとなかなか難しいと思いますので、絵で紹介します。

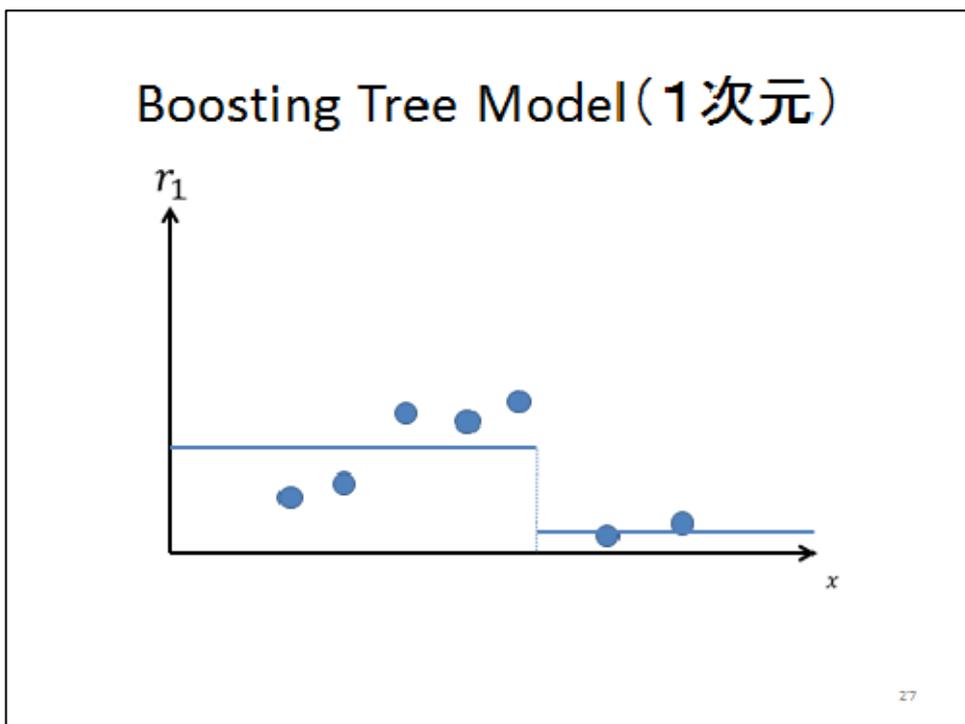
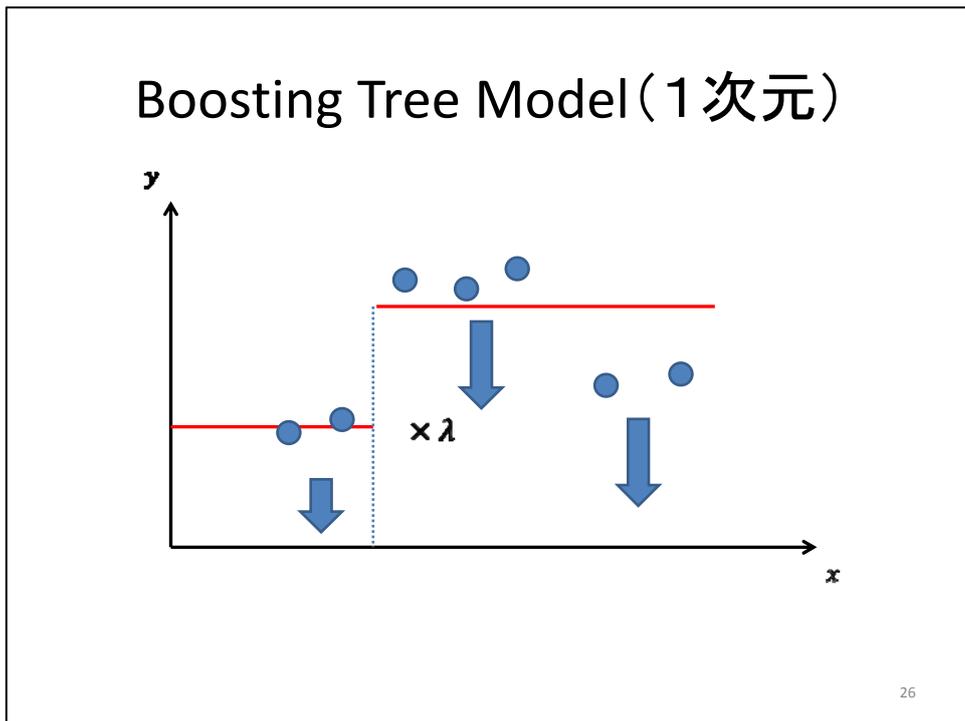
## Boosting Tree Model (1次元)



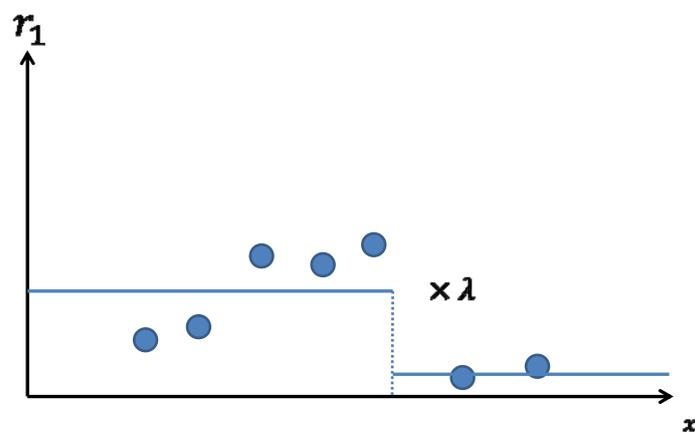
25

最初は、データに対して普通に木を当てはめます。それが、赤線のようにになっています。基本的な決定

木のモデルではこれで終わりなのですが、Boosting の場合には、このうち  $\lambda$  分だけを使います。



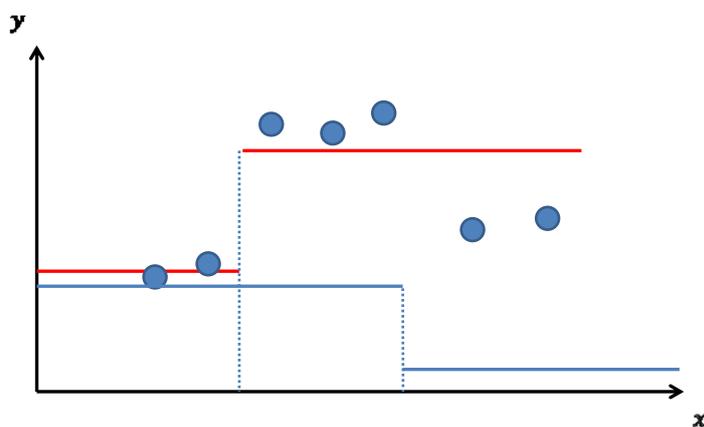
## Boosting Tree Model(1次元)



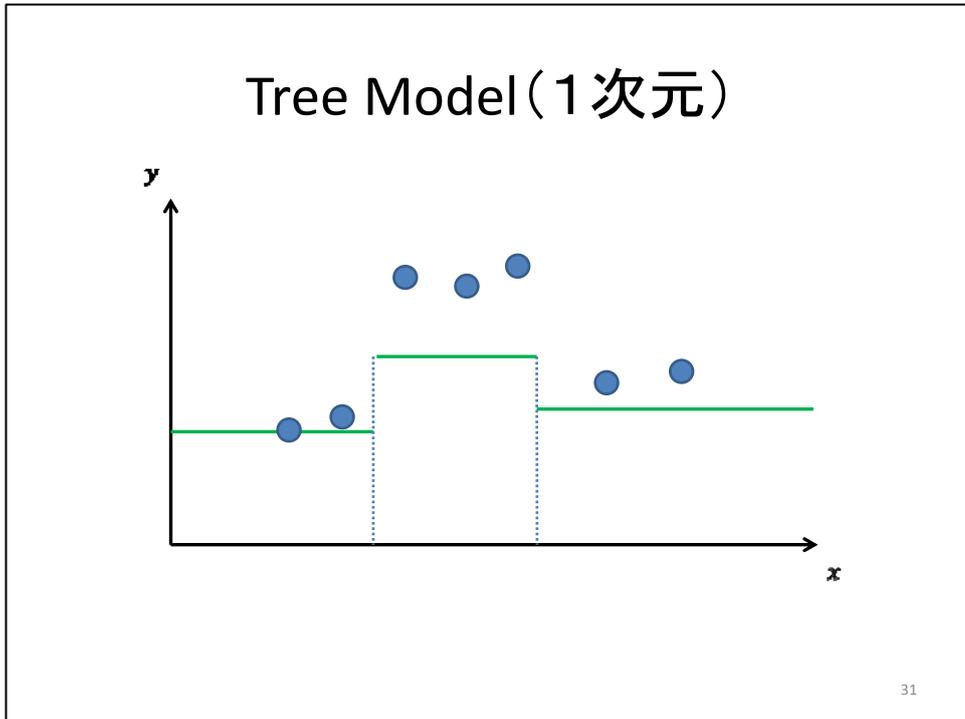
28

つまり、赤線の $\lambda$ 分だけを各データから差っ引いて、残った残差を計算します。そして、残った残差に対して、また別の木、青線を当てはめて、その $\lambda$ 分を使って残差を計算します。

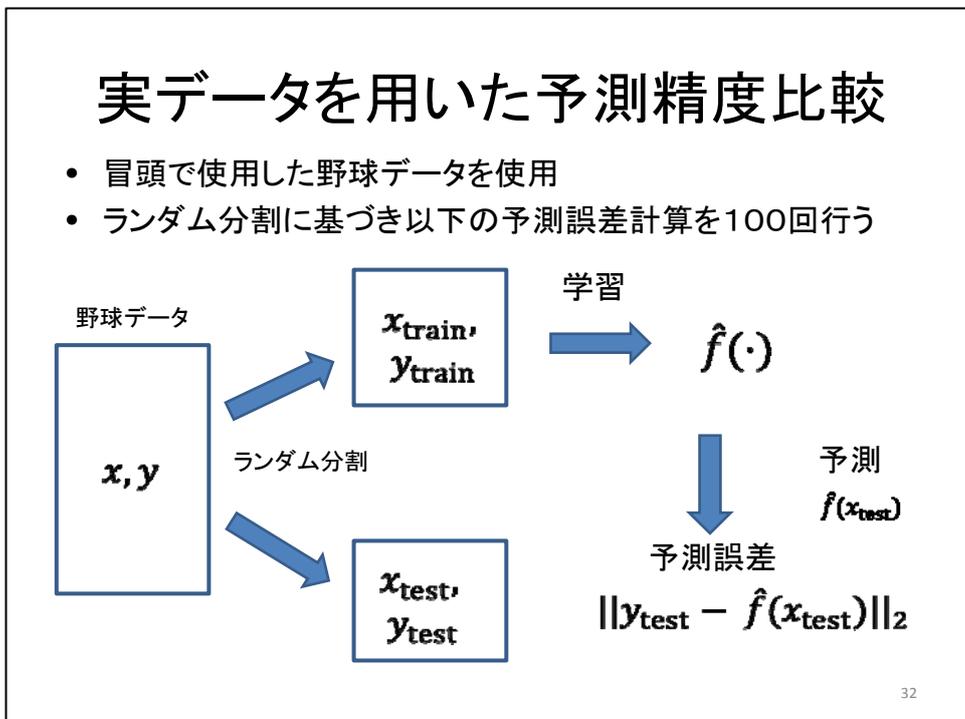
## Tree Model(1次元)



30



このように、毎回少しずつモデルを当てはめて、残った残差に対してまた別のモデル、木を当てはめてということを繰り返して、逐次的にモデルを学習していくというのが、Boosting になります。最後の出力としては、これまで当てはめた赤と青、二つのモデルがありますが、これらを平均化して、緑のようなものを最終的な出力にします。これが Boosting になります。



これまで幾つか手法を紹介してきましたが、そのうちの一部について、予測精度の比較を実データを用いて行いたいと思います。使用するデータは、また野球のデータです。どのように予測の精度を比較するかというと、野球のデータを毎回ランダムに学習データ、train と、テストデータ、test の二つに分けま

す。訓練データ、train の方についてモデルを当てはめて、当てはめたモデル  $\hat{f}$  に  $x_{test}$  を代入して、予測量  $\hat{f}(x_{test})$  を作ります。そして、実際の  $y_{test}$  と予測量  $\hat{f}(x_{test})$  の間の二乗誤差を計算します。これを一連の動作として 100 回行って、予測誤差の分布で各手法の良さを比較します。

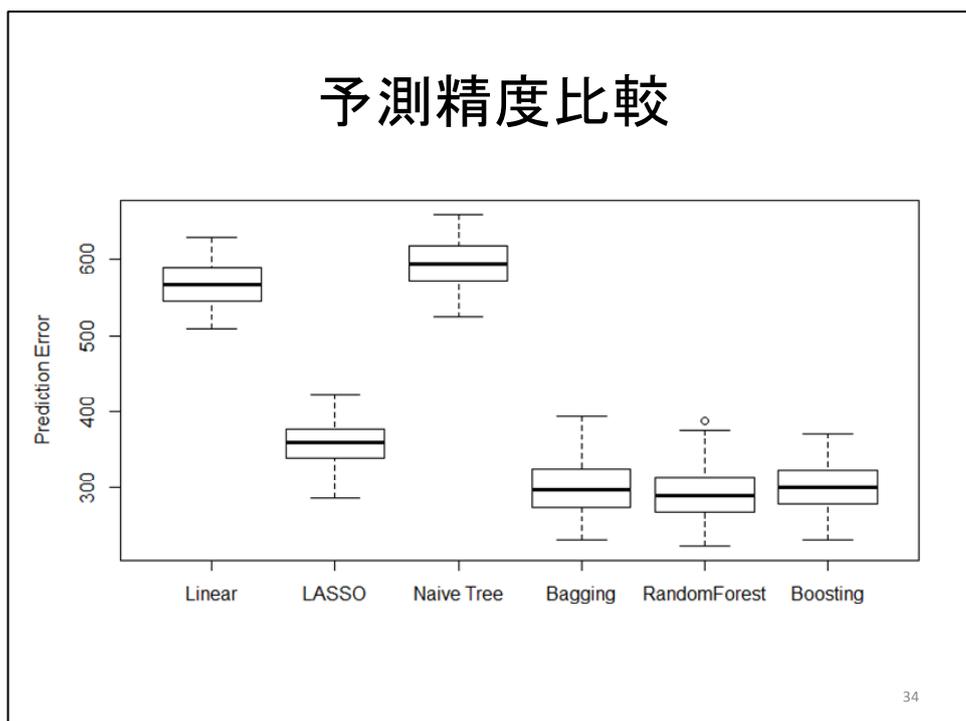
### 予測精度比較 (Hitters Data)

	Linear	LASSO	Naïve Tree	Bagging	Random Forest	Boosting
予測誤差	568	357	595	<b>299</b>	<b>293</b>	<b>301</b>
解釈可能性	容易	容易	容易	複雑	複雑	複雑

33

比較結果が、こちらになります。左から、線形回帰、LASSO、素朴な決定木モデル、Bagging、Random Forest、Boosting。最後の 3 つは決定木モデルの予測精度を向上させる手法でした。ご覧いただくと、右の Bagging 以下三つが、予測誤差が小さいことが見て取れます。一方、これまでご紹介したとおり、解釈可能性、モデルの複雑さで言うと、右の三つはモデルをたくさん混ぜて統合しているため、解釈可能性は難しい、複雑であるといえます。残念ながら線形回帰や決定木モデルは、分かりやすいのですが、予測誤差が非常に悪いといえます。そのような観点で見ますと、LASSO は、予測誤差の点で、一番ではないにしても、比較的良いですし、解釈可能性も線形回帰よりも、一部をスパース 0 にしていますので、容易であるということで、いいモデルだということが、このデータからは言えます。

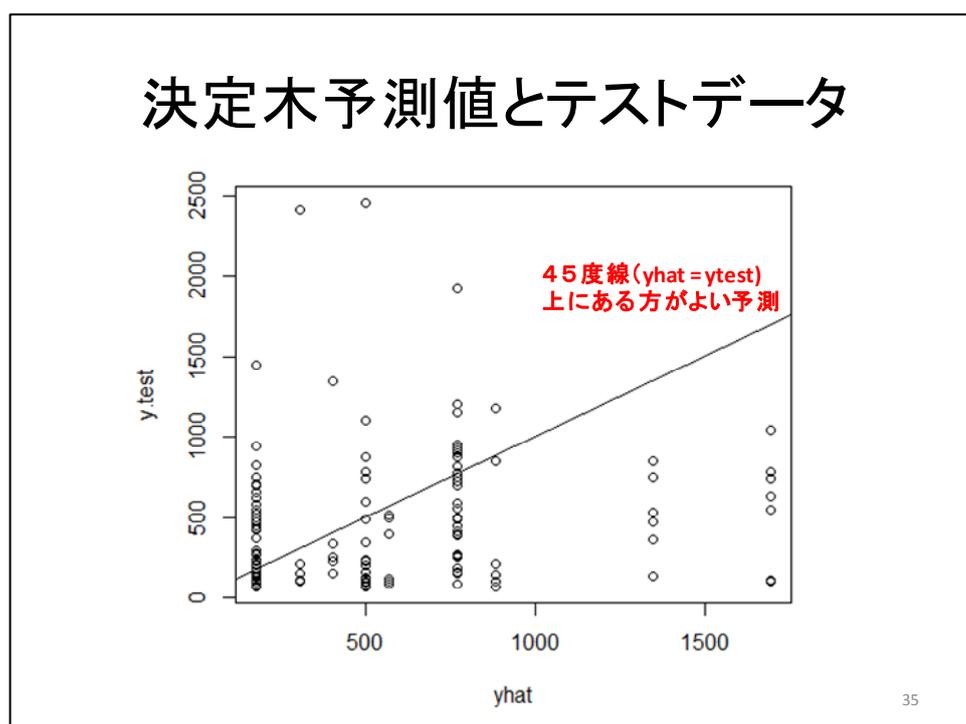
## 予測精度比較



34

実際に 100 回やったときの予測誤差を Box Plot したものが、こちらになります。今、説明したとおり、右三つは良いのですが、線形回帰や素朴な決定木モデルはあまり良くないということが、一目瞭然で分かります。LASSO は、その間に行くというモデルになると思います。

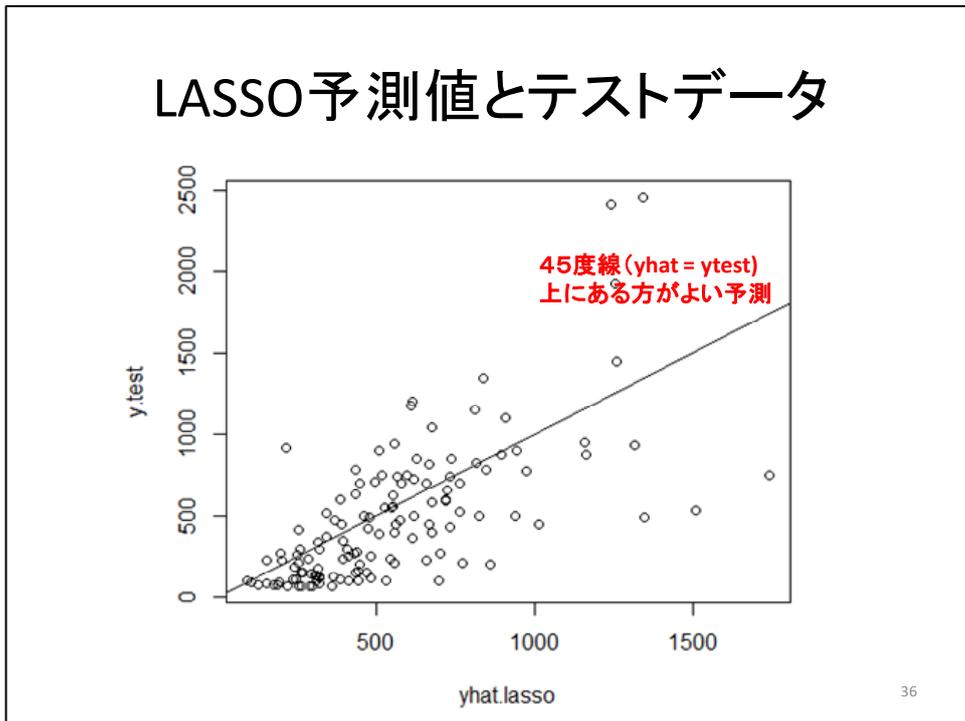
## 決定木予測値とテストデータ



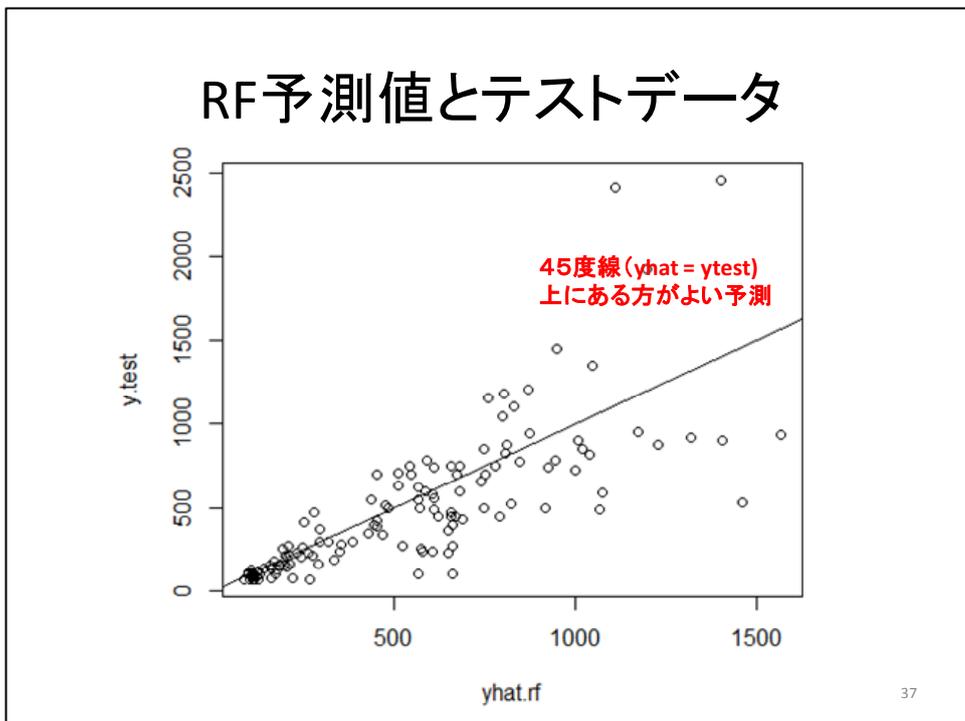
35

さらに、予測がどの程度うまくいっているか、いっていないかということを紹介したいと思います。x 軸は  $\hat{y}$ 、予測値になりまして、縦軸は  $y_{test}$ 、実際のテストデータの値になります。45 度線にあるときは、完全にうまく予測できているということになりますので、ここからのばらつきで、予測がうまくいっているか、いっていないか。つまり、45 度線にぴったりあれば、完全にうまく予測できていることになります。

決定木の場合は、ごらんいただくとおり、45度線から大きく乖離しています。というのも、これは縦線が9つあるのですが、つまり区分が9つしかないため、予測の当てはめが雑になっているといえます。



一方、LASSO は、より 45 度線に集中するようになっていまして、ただし、少し  $y$  が大きいときはうまくいっていないのですが、おおむね 45 度線にいる。



さらに Random Forest の場合は、さらに 45 度線に近くなって、より予測がうまくいっていることが分かります。

## 第一部まとめ

- 機械学習のモデルについて、データ適用例を含めて紹介
- **LASSOが簡便で予測精度も比較的よい**

	モデル	予測精度
線形回帰・決定木	簡素	×
<b>LASSO</b>	<b>簡素</b>	<b>○</b>
Bagging, RF, Boosting	複雑	◎

38

第1部については以上でして、まとめとしましては、幾つかの機械学習の手法を紹介いたしました。特に Bagging や Random Forest といった手法は、モデルが複雑になるのですが、予測精度は良いといえます。一方、線形回帰や決定木は逆になりますので、トレードオフが見て取れると思います。一言でメッセージを言うと、LASSO がうまくいっているということが、データを通して言えるのではないかと思います。以上です。

司会 小島さん、ありがとうございました。ご質問のある方、挙手をお願いいたします。

A 貴重なお話を、ありがとうございました。13 ページで、メジャーリーガーの予測の年俸の LASSO と線形回帰を比べた所があったのですが、直感的には、ホームランが 0 になるということが、ヒット数などに集約されるからデータとしてなくてもいいという解釈なのか、本当にホームラン数は関係ないという解釈なのか、どちらなのだろうと。直感的には、ヒット数に比例は多少するだろうから、こちらのデータがあればいいということで、ホームラン数が係数が 0 になって、予測上はヒット数が分かればいいし、モデルの複雑さとしても、変数が減ることでありがたいという結果になっているという解釈でいいのか。

特に、これだと分かりやすいのですが、実務などで、このデータが直感的には便利そうなのに 0 になったというときに、それが別の変数によるものだったのか、本当に関係ないことが分かったのか、どのように判断するのだろうかということが、この結果を見た際の直感的な疑問だったので、何かお考えがあれば、教えていただきたいと思います。

小島 はい。今、ご指摘いただいたとおりでして、LASSO は、二つ似ている変数があると一つ消してしまう、つまり 0 にしてしまうという性質があって、ある意味では、多重共線性と呼ばれるような変数の相関が強い場合には、回帰係数の信頼感を上げるということがあるのですけれども、今いただいたとおり、本当に残った変数が説得的かということは、また少し話が変わってきて、ここは各分野の知識も求められると思

います。今のホームランの場合は分かりやすいのですが、確かに保険の場合は、注意して使う必要があるのではないかと思います。お答えになっていますでしょうか。

A すみません。別の質問なのですが、決定木の変数選択にあまりなじみがなくて、単純に知らなかったのですが、9個の群団に最後は分かれて、45度の線を引いていただいている所があるかと思います。9個なのでばらついているという話があったのですが、例えば線形回帰では、要る変数、要らない変数のようなものを、p値で「これは棄却されるね」とやったりすると思うのです。木などは、やろうと思えば、もっとたくさん分割ができるような気がするのですが、そのあたりは、どのような変数の選択基準があるのでしょうか。

小島 つまり、木をどのように学習するかということですか。

A そうですね。木を、どこまで細かく切っていいのかというようなものが。

小島 なるほど。幾つかありますが、大きく分かれて、一つは、各枝、各分類に入るデータ数があまりにも少ないと信頼度が落ちてしまうので、各グルーピングのデータが、例えばデータ数が10個以下になったら分割をやめるなど、データの個数に応じて分割を停止するという停止則もあります。もしくは、毎回木を学習するときに、どれくらいデータ値と予測値の間の二乗誤差が小さくなるのかということを見て、それがあまり大きく変わらないようであれば、木の学習を止めるというものもあります。大きくそのような二つがあると認識しています。

A 分かりました。ありがとうございます。

司会 他に何かございますか。特になければ、次の部に移らせていただきたいと思います。続きまして、共栄火災の佐野さん、よろしく願いいたします。

小島 ありがとうございます。

司会 ありがとうございます。

佐野 第2部はあまりタイトルと中身は関係ないかもしれませんが、いわゆるつなぎの話をしようと思っています。つなぎというのは、先ほどの話と後の話との間という意味もありますし、聞いている人と話している人の間をつなぐような話という意味でもあります。

## アクチュアリー視点でのモデル選択

---

2018年11月9日  
ASTIN関連研究会  
佐野 誠一郎(共栄火災)

なぜそのような話を間に入れるかという、年次大会やセミナーで「データサイエンスとアクチュアリー」のような話をよく聞くことがあるのですが、大体は「ふうん」と思って終わってしまうのです。そこに距離感のようなものを感じるのです。先ほどの話も、もしここで終わって「はい、さようなら」となると、家に帰って寝たら忘れてる。ここではその距離感のようなものを共有して、後のいい話につながるよう、その入口に片足を突っ込んでもらおうというのが趣旨です。

## 目次

1. モデル選択とは
2. (アクチュアリー視点で)モデル選択において求めるもの
3. 新たな手法の導入における課題
4. 次への方向性

目次はこのようなものです。まずは双方向がありますので、単純なアンケートですが皆さん教えてください。どれに該当しますか。

## はじめに

### • 質問:どれに該当しますか？

- ① 機械学習は、アクチュアリー業務として既に導入している

**7.9%**

- ② ビッグデータが手元に存在するなど機械学習を導入する必要性に迫られているが、まだ勉強中

**27%**

- ③ 特に機械学習を導入する動機はないが、時流に流されて勉強中

**33.3%**

- ④ 機械学習を勉強する動機もないし実際していないが、ざわざわするテーマなので見に来た

**31.7%**

一番下に結構いるというのは思ってもみなかったですが、分かりました。では、そのようなつもりで進めます。

はじめに「モデル選択とは」という話です。「よりよいモデルを選ぶこと」というのは、機械学習の本を読むとモデル選択の話があって、読んでいくと、bias と variance のトレードオフ、過適合、汎化性能と続いて、次に AIC を見ましょう、クロスバリデーションを使いましょうということが一つのゴールのような形で出てくるのですが、それがより良いモデルなのかというと、読む人が、入口で、そもそも何をしたいかによるわけです。

例えば、AIC が何かと大ざっぱに言うと予測誤差の相対的な評価基準であり、真のモデルを探す推定よりも新しいデータがどの値を取るかという予測を目的としたときに先ほどのような話になって、予測誤差の相対的な評価基準として AIC が小さいものを選びましょうということになるのであって、入口で、そもそもそれがやりたく読んでいるのか、それが自分の目的なのかということとはまた別の話なのです。

# 1. モデル選択とは

## • よりよいモデルを選ぶこと

### • いいモデルとは？

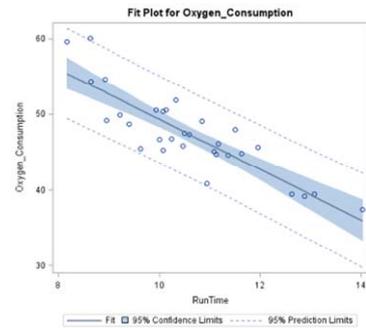
…モデル選択というと、bias-variance、過適合、汎化性能の話があって、AICやCVなどの基準・手法の話になるが

## • それは目的による

### • 例えばAICは予測誤差の相対的な評価基準

…では、予測誤差の最小化が  
よりよいモデルの基準なのか？

### • やろうとしているのは推定？予測？



右下のプロットは簡単な線形回帰のアウトプットですが、真ん中にある青い部分は、真のモデルの形を探すという意味での  $\beta$  の推定による信頼区間です。外にある点線は、新しいデータがどのような値を取るかという意味での  $y$  の予測区間です。これらのどこを見るかというのは、自分がどのような目的でこれを出したかによります。

また、正則化の話であれば、線形回帰は皆さん知っていますね。ですから知っているという前提ですが、ここでペナルティを付けますと言われても、そのペナルティがいいのかどうかは、線形回帰といっても、同じ手法で出てくるアウトプットの何を見るかは目的によるのですから、それがいいのかも、入口で自分が何を目的としているかによるのです。それが合わないと、一方的によいモデルの話をして、全然合わない。ミスマッチが発生して、それが「ふうん」というセミナーが発生する要因になる。

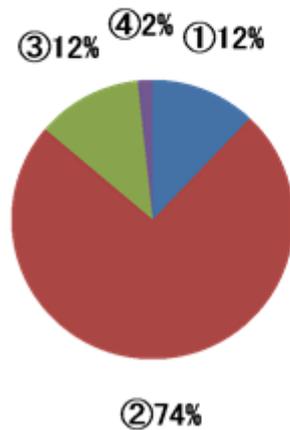
だからこそ、今回双方向ツールがあるということは、皆さんが何を求めて来たかを主張できるということだと思っているので、そのような意味で教えてください。何をしに来ましたか。

ちなみに遅れましたが、今回の話は全て個人の意見ですので、謝っておきます。

## 2. モデル選択に求めるもの

- 質問:どれを重視することが多いですか？

- ① とにかく精度
- ② 説明や解釈のしやすさ
- ③ 安定性・再現性
- ④ スピード感



では集計を見ます。ああ、②なのですね。なるほど、言われてみればそうですね。分かりました。

いきなり「何しに来た」という質問ですが、それぞれの項目に対して皆さんが思っていることもバラバラだと思うので、少し強引ですが中身について簡単に整理していきます。

## 2. モデル選択に求めるもの

- 精度

- 何をもちて精度がいいと言うのか・・・例えばAICが最小となること？

Fit Statistics	
-2 Log Likelihood	29011
AIC (smaller is better)	29017
AICC (smaller is better)	29017
BIC (smaller is better)	29035

とにかくいっぱい  
出てくる統計量

Parameter Estimates								
Parameter	Estimate	Standard Error	DF	t Value	Pr >  t	95% Confidence Limits		Gradient
b0	3.6850	0.02343	3155	157.28	<.0001	3.6391	3.7309	0.032837
b1	-0.2013	0.03558	3155	-5.66	<.0001	-0.2711	-0.1316	0.012100
scale	1.0194	0.02264	3155	45.03	<.0001	0.9750	1.0638	-0.00029

- どれを見ればやりたいことの評価になる？ AIC？ BIC？ Estimate？ t値？
- 伝える人と聞く人は共通認識があるのか？

まず精度ですが、先ほどと似たような話です。何をもちて精度と言うのかは、できれば共有してほしいということです。そうでないと手法の話などを聞くときに、話す方はそれがいいのだと確信していますから「これはいいモデルだ」と言って、初めて聞かされたこちらは「そうなんだ」と思いながらも、恐る恐る隣の知らない人の顔を見たらうなずいたりして「ああ、やっぱりいいモデルなのだ」と。そして家

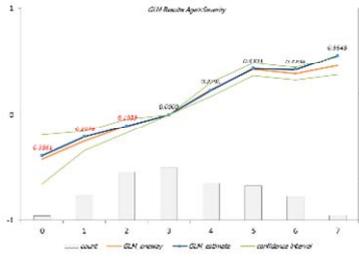
に帰って寝たらやはり忘れるのですね。

ですから、そこは気をつけてほしいというか、共有したい。何をもって精度なのかを。

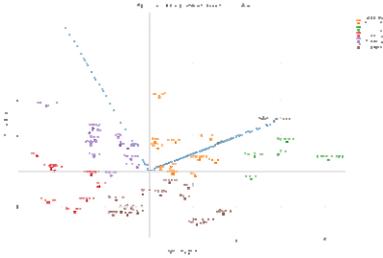
7

## 2. モデル選択に求めるもの

- 説明や解釈のしやすさ
  - おそらく、アクチュアリーが最も必要とするもの
  - 結果の解釈も重要だが、手法そのものの理解が特に必要
    - 例えば「エクセルで再現可能」であることが説明可能性だという謎の要求



手法は複雑だが定量的なので  
説明しやすいGLM



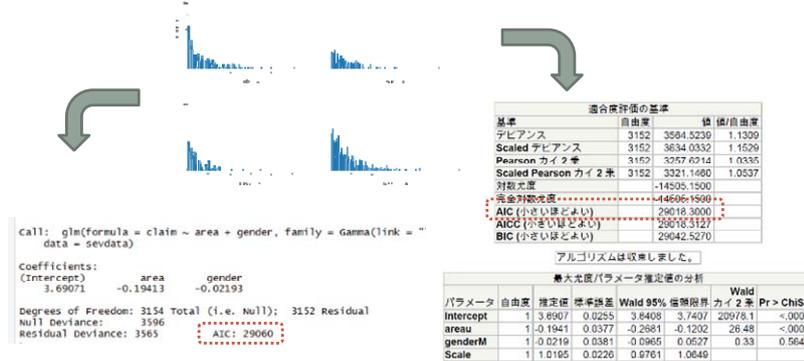
手法は単純だが定性的なので  
説明しづらいクラスタリング

次が、説明や解釈のしやすさです。これは、先ほどの回答もたくさん出ていましたけれども、恐らくアクチュアリーが最も必要とするものだろうと思います。個人的な意見ですが、「データを送ってくれ」と言われて送って見たら「エクセルで再現できないから、ダメだね」と。それをもって説明可能性と言われると、少しつらいというのがあります。

あとは、本を読むとクラスタリングなどは解釈のしやすさで出てきますが、アクチュアリーの場合、定量的な評価をもって説明可能性だというのは、一つの特性なのかなと思います。

## 2. モデル選択に求めるもの

- 安定性・再現性
  - 恣意的なパラメータは少ないほうが運用しやすい
  - ツールが異なっても作業者が異なっても、同じ値を算出してほしい



次は、安定性・再現性です。若い人にはそれほど響かないかもしれませんが、少し年を取ると非常に大事になります。「再現できません」と電話がかかってきて理由を聞いたら、「担当者がいなくなったらしいです」と。そんな理由でできなくなる手法ではつらい。

ですから、これを志向するのであれば、よりシンプルな手法が好ましいということかと思えます。

## 2. モデル選択に求めるもの

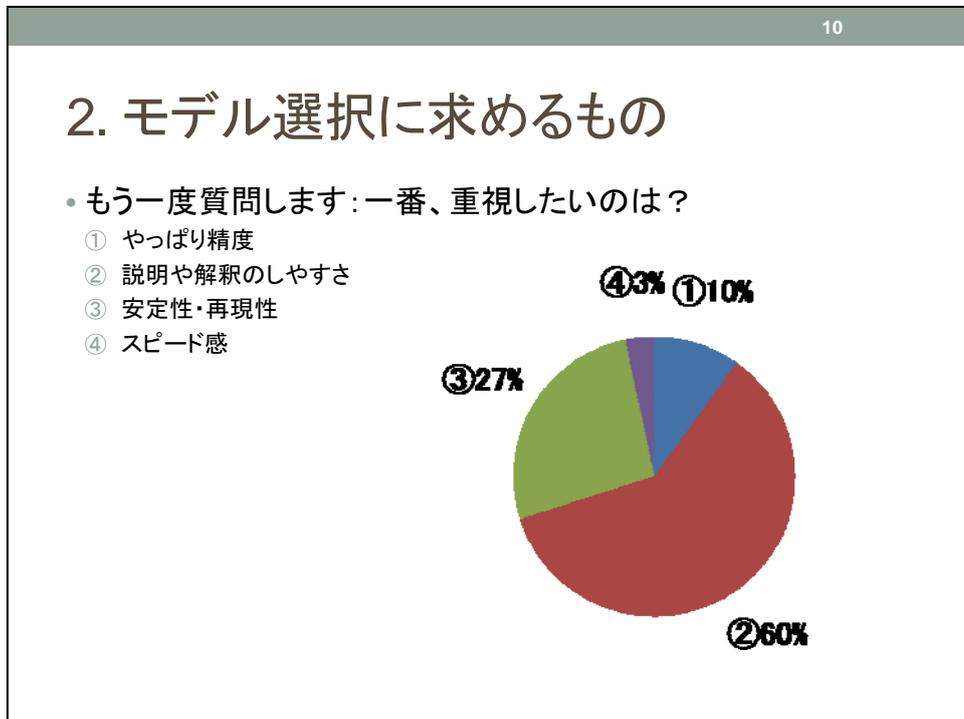
- 計算速度
  - ここでは手法としての速度というより、判断に要する速度



最後は、計算速度です。先ほど「スピード感」として聞いていましたが、僕が思っているのはアルゴリズムなどの速度よりも、判断に要する速度です。右側のプロットのように、どうにかまとめたいたいという時がありますよね。こういう作業を夜の7時頃にやりだすと止まらなくなってきて「このような地道な作

業がアクチュアリー姿なのだ」とか格好のいいフレーズを考えているうちに、9時になるとPCが切れる。それで我に返って、考えずにできる手法があればいいなと思うという話です。

非常に強引ですけど、これで各項目がどのようなことかを簡単にまとめましたので、それを踏まえて、このあとの話につなげるために、もう一度聞きます。何を求めますか。

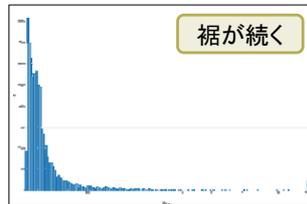


先ほども言いましたが、情報を共有するということですので、結果は後の話に引き継がれます。皆さんが安心して聞きたいものを聞けるチャンスです。今、言っておいたので、このあとに期待してください。

これで、何を聞きに来て何を話すのかというお互いに一方的なものは共有しましたので、次は、そうはいつでも、新しい手法を勉強するときに何かある壁や課題についての話をして、それから後の話の入口に入っていきたいと思います。

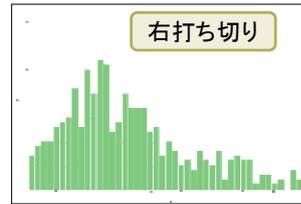
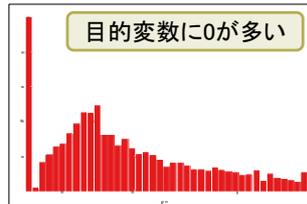
### 3. 新たな手法を導入するときの課題

#### • 保険データの特性



ロスデータの精度

gender	度数		累積	
	度数	パーセント	度数	パーセント
Female	22	4.51	22	4.51
Male	50	10.25	72	14.75
Other	416	85.25	488	100.00



- 手元にある保険データはクセが凄くて、本を読んでも参考にしづらい
- 解釈のしやすさを考えると、あまりデータを加工したくない

まず一番大きな課題としてあるのは、保険データの特性です。保険データに癖があって、そのせいで、機械学習の本を読んで勉強すると「データを使ってやってみましょう」となるのですが、ただただ関係ないデータに詳しくなっていくばかりで、保険データに対応すると別の話という問題です。

このような特性は昔からありますから、個々の課題に対して言えば、昔から対応方法はあります。けれども、それらを組み合わせると途端に情報がなくなってしまう。

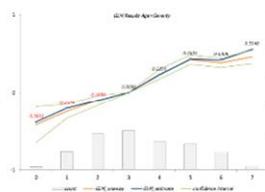
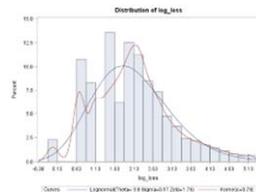
### 3. 新たな手法を導入するときの課題

#### • 保険データの特性

- 課題に対応しても、組み合わせると情報がなくなる

裾の対応  
○ リスクの表現  
× 予測に向かない

GLM  
○ 柔軟な変数による予測  
× 平均しか見ない



裾に対応した  
GLM

よく見ない

例えば GLM にはあまり関係ないですけど、裾に対する対応は伝統的なロスモデルがあり、GLM も一般的です。しかし、それらを組み合わせると裾に対応した GLM を作ろうとすると、一般的には EDM ではない分布を

使うような考え方になります。そうすると、GLMを使うときに守られていたEDMのルールを再確認しなければいなくなって、途端に安定性がなくなってしまう。そして、手法自体はシンプルなもの組み合わせなのに、急に情報がなくなってしまうのです。

ただ情報がなくなってしまうということは、逆に言えば、新しいアイデアを生み出したいという考えからすると、一つのチャンスでもあるのです。皆がよく知っている手法の組み合わせで、でき上りは誰も見たことがないものが作れる。それほど難しいことをしているような感じではないのに、誰もやったことがないものが作れるチャンスでもあるのです。

次が、規制や制限の存在でビックデータがそもそも手元にないという話です。

実務的には、訳の分からない変数がたくさんあるデータを使うことよりも、非常によく知っている2、3の変数とずっと付き合うようなケースの方が多いのではないかと思います。自分たちの実務では、そのようなデータだから機械学習の手法は使えないということではなくて、知識がたくさんあるのならそれを横に広げて、やりたいことをよりよく表現するために、新しい手法が使えないかと考えればいだけだと思います。

13

### 3. 新たな手法を導入するときの課題

- 規制や制限
  - モデル選択するほど変数の選択肢がない
    - ...ならば、限られた選択肢の中でよりよい表現を求めたい

• 既存手法の理解を壊さず平滑化したい  
• こういう辺りを時間をかけずまとめたい

例えばこの図の場合では、よく知っている既存の手法の理解を壊さない範囲で、外から変な手法を入れるのではなく平滑化したい。また先ほどの話にもありましたが、点線枠のあたりを、時間をかけずに、いちいち手作業ではなくまとめる方法はないか。よく見るこういう課題があるから新しい手法が使えないのではなくて、その課題をクリアするために新しい手法が使えないかと考えればいいのだと思うのです。そう思うのですが、セミナーなどでは手法の話が先に来る。手法の話が先にされると、どうしても入ってこないのです。ピンとこなくなってしまう。

そうではなくて、自分たちは、何かがしくてここに来ている。先ほども双方向で主張してくれたということは、何かやりたいことがあって、その課題をクリアするために来ている。ですから、それをクリアするために新しい手法が使えるという話がしたいということだと思うのです。

## 4. 次への方向性

- 課題への対応
  - 手法が先に来ると理解しづらい・・・そもそもは機械学習を知りたいが先ではなく、(保険)データをどうにかしたいのではないか
- 実践
  - 自分の目的に合った分析手法が聞けたら、それが一番いい勉強になる  
⇒保険データが持つ課題などに対応した、機械学習の事例とか
  - 始めはみんなデータを見ながら手法を学ぶ・・・それが保険データなら悩みは減る
  - ただ提案するだけでは意味がない・・・保険データと新たな手法を使った事例で、複雑でなく、かつ固有の課題に対応したのが見たい

アクチュアリーが強みを無理やり考えるとすれば、先ほどの保険データもそうですが、普通の本を読んだのでは出てこない特殊な課題をシェアして取り組めるという集団だということではないかと思います。そのように考えると、課題をクリアするために機械学習の手法が使えるというような話ができれば、アクチュアリー業務と相容れるのかというようなことを、わざわざ考えなくてもいいのではないかと。

さらに、後から来た人で機械学習もアクチュアリーも知らないような人が初めて勉強するときを考えると、どのような人も初めに勉強するときは、本を読んでデータを見て入れてみてという作業をするのですから、そのときに保険データを使った事例が存在していれば、アクチュアリーとデータサイエンスのような複雑なことを考えずに、変な概念もなく、すっと入って、すっと成長して、自分たちを飛び抜けていってしまうという環境ができるのではないかと思います。

このように、自分たちが持っている特殊な課題をクリアするための新しい手法、しかも、保険データを使った事例があればよいという話をして、ここまでだとやはり「ふうん」で終わってしまうのですが、あのなら見てみたいですね。

ということで、僕の話は終わりますが、話はこの後に続きます。ありがとうございました。

司会 佐野さん、ありがとうございました。質問は、ないかもしれませんが、ございましたらどうぞ。特になければ、引き続き、Guy Carpenter の藤田さん、よろしくお願いいたします。

藤田 Guy Carpenter の藤田と申します。第3部の「AGLM-Accurate Generalized Linear Model」について、発表したいと思います。佐野さんにハードルを上げられてしまったのですが、ASTIN 関連研究会として、アクチュアリーとしてのスタンスは本来どのようなものだったかということを考え直して、研究したモデルについて、ご紹介したいと思います。

## アクチュアリーとモデル選択 (Accurate GLM) 第3部 AGLM - Accurate Generalized Linear Model

日本アクチュアリー会  
2018年度年次大会

ASTIN関連研究会  
藤田 卓 (Guy Carpenter Japan, Inc.)

本発表のアジェンダです。まず、簡単に背景をご説明して、次に、早速 AGLM とは何かということをご説明したいと思います。次に、ASTIN 関連研究会としてどのような活動を行ってきたかをご紹介して、最後にまとめたいと思います。

### アジェンダ

- はじめに
- AGLM - Accurate GLMとは
  - 概要
  - 問題設定
  - GLM
  - 3つのコンセプト
- ASTIN関連研究会での取り組み
  - 『AGLMの研究』
  - 基本プログラムの概要
  - 数値実験と考察
- おわりに
- Appendix

1

まず、「はじめに」です。このあたりは、第1部で小島さんに機械学習の手法についてご紹介いただきましたけれども、アクチュアリーの中でも機械学習はホット・トピックです。この年次大会の中でも、機械学習に関するセッションが幾つかあったと思います。これらをアクチュアリーの実務にどのように取り入れるかを考えることは、重要な課題の一つになっているかと思います。実際に私も機械学習の手法は

非常に興味があって、日頃勉強して、どうにか実務に生かせないかと考えているのですが、この会場にいらっしゃる皆さんも、同じように考えている人はいると思います。一方で、アクチュアリーが抱える悩みが多いことも事実だと思います。特に説明可能性という部分が、先ほど2番で割合が多かったですけれども、キーポイントなのかなと。

#### はじめに

##### - 今までの内容のおさらい

- アクチュアリーは機械学習に興味があるし、実務に取り入れたいと考えている。
  - アクチュアリー “も” ビッグデータを扱う場面が増えてきている。
  - 現代の潮流：データサイエンスの台頭
- 一方、取り入れる上でアクチュアリーが抱える悩みは多い。
  - 保険データの特性にあった使い方
  - 規制や制約など

3

そこで、われわれが着目したいポイントですが、機械学習の手法は、ある意味たくさんあります。Random Forest や XGBoost など、最近皆さんも耳にしていると思います。しかし、それぞれには課題があって、必ずしも実務に使えるわけではないという状況ですよね。実際非常に格好がいいし、すごそうだけれども、説明できるかどうかでつまずいてしまう。この点を踏まえ、私たちは、アクチュアリーとしてどのような手法が求められているのかという点に立ち返るべきだと考えています。われわれアクチュアリーが抱えている課題と、それに対してどのような手法が求められているかを認識したうえで、機械学習の手法を使っていくべきではないか、というわけです。具体的に抱えている課題があって、それに対してうまく腑に落ちる手法があれば、それは非常にうれしいわけです。そして、それが機械学習に関連する手法であれば、なおさら皆さんうれしいのではないのでしょうか。その解決策の一つが、AGLM、Accurate GLM だと考えています。

はじめに  
- 着目すべきポイント

- 「いかに既存手法を実務に落とし込むか」に捕らわれすぎ？
  - ランダムフォレスト
  - XGBoost
  - ニューラルネットワーク etc.

カッコいいし、すごそう...  
でも説明できるか...?



- 「アクチュアリーとしてどのような手法が求められているか」という点に立ち返るべき。

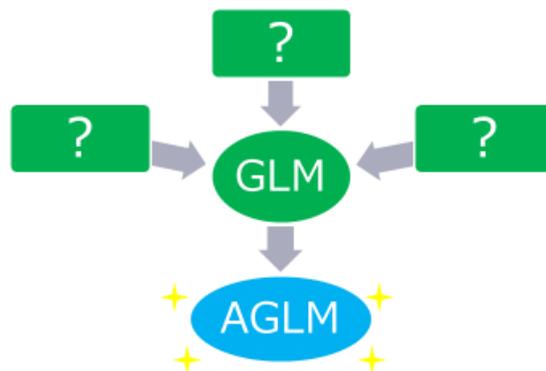
その解決策の一つが  
“AGLM - Accurate GLM”

4

早速、AGLM とは何かということですが、皆さんよくご存じの一般化線形モデル、GLM に、三つのシンプルなコンセプトを組み込んだものです。AGLM のAは、Accurate のAで、私たちは、予測性能や予測精度が高いというメッセージを込めています。この三つのシンプルなコンセプトですが、一つ一つに真新しいものはないのですが、これらを組み合わせると、非常に面白くてうれしいことが起きます。AGLM は、アクチュアリーと非常に親和性が高い機械学習の手法としての可能性があると考えています。まず GLM について簡単に振り返ったあとに、一つ一つのコンセプトを説明して、AGLM の全貌についてご紹介します。

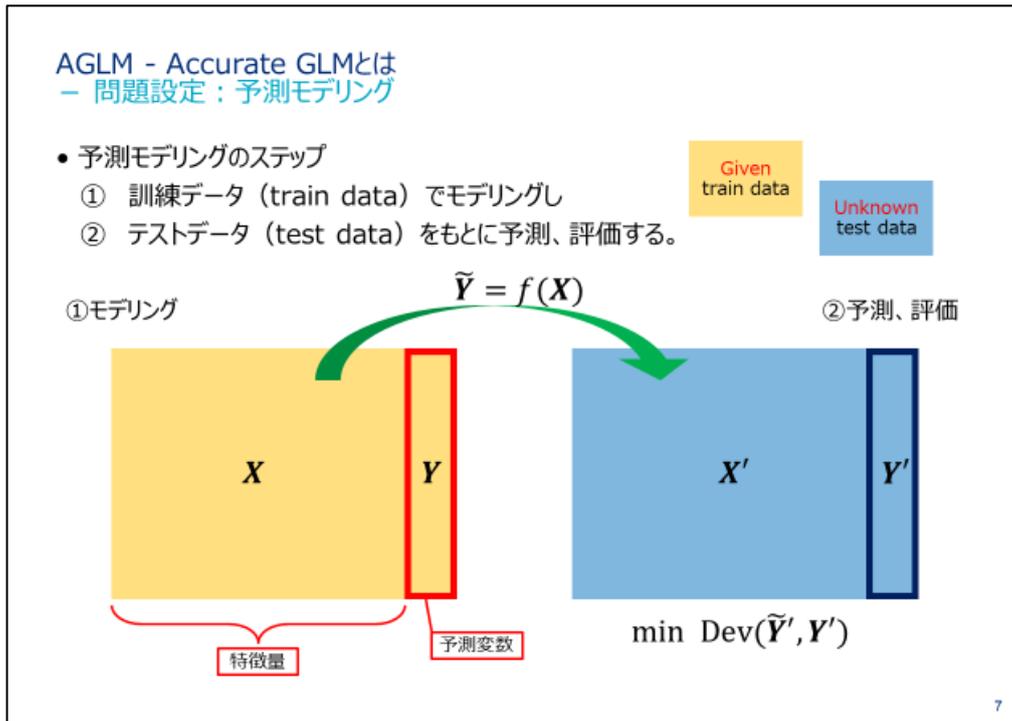
AGLM - Accurate GLMとは  
- 概要

- 一般化線形モデル (GLM) に“3つ”のシンプルなコンセプトを組み込んだもの
  - 一つ一つに新規性はないものの
  - これらを組み合わせたときの効果は大きい！ (先行研究見つからず)
- AGLMはアクチュアリーにとって強力な機械学習手法となる可能性がある。



6

その前に、AGLMの問題設定です。問題設定としては、予測モデリングを想定しています。予測モデリングは、簡単に申し上げますと、訓練データや学習データと呼ばれる、ある与えられたデータがあって、それを基にモデリングを行います。そして、モデリングで使わなかったデータ、テストデータや評価データと呼ばれるものですが、それを基に予測評価するといったステップを取ります。図で簡単に表すと、このモデリングで使う黄色のデータのうち、Yという予測したい変数、予測変数と、その他のX、特徴量というものがあって、Xを組み合わせ、何らかの関数のようなもので予測モデルを作ります。そして、次の予測評価のステップでテストデータに対して予測変数に対する予測を行って、真の値と、モデルで予測した結果とのディバイエンス、誤差や逸脱度を最小化するように評価します。



われわれはアクチュアリーなので、問題設定としては、損害保険の総損害額の予測モデリングを考えたいと思います。総損害額の予測モデリングのステップとしては、まず発生頻度をモデリングして、その次に損害規模をモデリングして、最後に総損害額をモデリングします。これは、よく皆さんがアクチュアリー試験で勉強したような損保数理のFD法と同様です。式で表すと、Nが発生頻度、Xが損害規模、Sが総損害額を表す確率変数で、NとXの期待値の掛け算で総損害額の期待値が表せます。

## AGLM - Accurate GLMとは - 問題設定：予測モデリング（続き）

- 損害保険の総損害額の予測モデリング
  - ① 頻度（Frequency）をモデリング
  - ② 損害規模（Severity）をモデリング
  - ③ 総損害額（Aggregate Loss）をモデリング
- いわゆる損保数理のFD法（純保険料法）と同様

$$E[S] = E[N]E[X] \quad (S = X_1 + X_2 + \dots + X_N)$$

- ✓ 純保険料 = 期待総損害額 = 発生頻度 × 損害規模
- ✓  $N, X, S$  はそれぞれ頻度、損害規模、総損害額を表す確率変数
- ✓  $X$  は  $N$  が 0 より大きい場合に値をとる（条件付）

8

それでは、GLM についてです。まずは線形回帰モデルから振り返ってみたいと思いますが、線形回帰モデルは、 $y$  という予測変数があって、各特徴量  $x$  に係数  $\beta$  を掛けた線形和で表したのに対して、更に誤差項を付け加えたモデルになっています。ここで、 $n$  はデータのレコード数で、 $p$  は特徴量の数だと思ってください。回帰モデルでは、 $x$  は説明変数という言葉の方が皆さんなじみがあると思うのですが、ここではこれと同義の「特徴量」という言葉を使用いたします。

これに対して GLM は、このような式で書き表せます。特徴量の線形和と予測変数の期待値を結びつける、リンク関数というものが導入されています。この  $g^{-1}$  がそうですね。また、予測変数の分布が指数分布族と呼ばれるものに拡張されています。ここで GLM を取り上げたのは、アクチュアリーとしてどのような手法が求められているかと考えたときに、常日頃から使い慣れていて、分かりやすい手法である GLM をベースにすることは、非常に意義があるのではないかと思うからです。

アクチュアリーがなぜ GLM に親しみがあるかという点、ここに書いてあるような指数分布族と呼ばれるもので、かつ保険データによく使われる Poisson 分布やガンマ分布、Tweedie 分布も使えますし、例えば、リスク間で乗法的に働くと想定される場合に有効となる対数リンク関数が使えたり、オフセットというものにより、エクスポージャーの大きさを踏まえたモデリングをすることが可能ということが挙げられると思います。アクチュアリーは GLM に精通していて、GLM が大好きだということが、ポイントの一つだと思います。

AGLM - Accurate GLMとは  
- GLM

- 線形回帰モデル

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \epsilon_i \quad (i = 1, 2, \dots, n)$$

✓  $y, x, \epsilon$ はそれぞれ予測変数、特徴量（説明変数）、誤差項（正規分布に従う）、 $\beta$ は係数  
✓  $n$ はデータのレコード数、 $p$ は特徴量の数 ※“説明変数”とほぼ同義の“特徴量”という用語を使用します。

- GLM

$$E[y_i] = g^{-1}(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \quad (i = 1, 2, \dots, n)$$

✓  $g(\cdot)$ はリンク関数：特徴量の線形結合と予測変数の“期待値”を結びつける。  
✓ 予測変数の分布を指数分布族と呼ばれるものに拡張

**ポイント** アクチュアリーはGLMに精通している。  
アクチュアリーはGLMが大好き！

10

それでは、三つのコンセプトですね。一つは、離散化というものです。英語に直すと、DiscretizationやBinningと呼ばれます。二つめが、<sup>オー</sup>ダミー変数と呼ばれるものです。英語ではOrdinal Dummy Variables。三つめが、正則化。先ほどから出ていますが、英語ではRegularizationといいます。簡単にするために厳密な理論や定式化は割愛しますが、これらの三つを組み合わせると、面白いことが起きます。一つ一つご説明します。

AGLM - Accurate GLMとは  
- 3つのコンセプト

- ① 離散化 (Discretization/Binning)
- ② Oダミー変数 (Ordinal Dummy Variables)
- ③ 正則化 (Regularization)

※簡単のため、厳密な理論や定式化は割愛します。

11

まず、離散化からです。離散化は、簡単に申し上げますと、数値型の特徴量を幾つかのビンにグルーピングすることを言います。数値型特徴量というのは、われわれが普段使うデータで言うと、年齢や給料、

車の走行距離などです。例えば給料の場合、年収が 200 万円から 500 万円の人をグループ 1、500 万円から 1,000 万円の人をグループ 2 というようにグルーピングするものを、離散化と言います。いろいろな手法が実は研究されていますが、AGLM では、シンプルにできるだけ細かく、等間隔に離散化すればいいことになっています。これについては、後述したいと思います。

**AGLM - Accurate GLMとは**  
- 3つのコンセプト ①離散化

- 数値型特徴量（連続値データ）をいくつかのビンにグループ化すること
  - 例) 年齢、給料、車の走行距離など

給料の場合：  
[200...500万]→グループ1 [500...1,000万]→グループ2 ...

- どうやって離散化する？さまざまなやり方があるのでは？



- 実はAGLMでは、シンプルにできるだけ細かく、等間隔に離散化すればよい！
  - 何故かは後述（3つのコンセプトのシナジー効果）

13

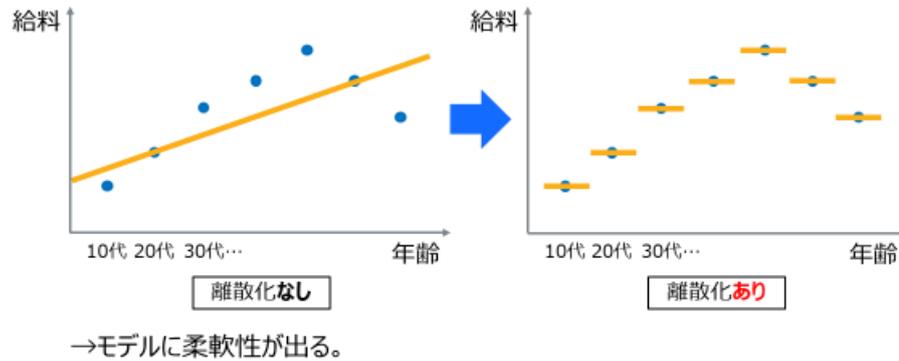
離散化すると何がいいかというと、アクチュアリーとしての視点に立った場合に、年齢や給料などの数値型特徴量をそのまま説明変数として使うと、なかなかうまくいかないことが多いのではないのでしょうか。離散化をすると、よりモデルを柔軟にすることが可能です。ですから、離散化の効能として、「未学習（アンダーフィッティング）を回避する」と書いています。

イメージとしては、例えば横軸が年齢で 10 代、20 代、30 代となっていて、縦軸に給料をプロットした場合に、普通の線形回帰モデルでは、そのまま年齢を説明変数としてこのように予測されますが、果たしてこれはいいモデルかという、疑問が湧くわけですね。離散化は、それぞれグルーピングしたものに対して予測するというイメージです。これによってモデルに柔軟性が出るのが、離散化の効能の一つです。

AGLM - Accurate GLMとは  
- 3つのコンセプト ①離散化 (続き)

- 離散化の効能 **ポイント**
  - 未学習 (アンダーフィッティング) を回避する。

- イメージ図



14

続いて、二つめのOダミー変数についてです。Oダミー変数は、順序関係を導入した特殊なダミー変数のことを言います。順序型の Ordinal から、Oダミー変数と呼んでいます。対比のために、通常のダミー変数を Usual Dummy として、Uダミー変数と呼びたいと思います。Uダミー変数は、皆さんご存じの、カテゴリカル変数に対して一般的に用いるものです。カテゴリカル変数は、例えば春夏秋冬などの離散的な変数に対して、春であれば1を取る、それ以外の夏・秋・冬の場合は0を取るといった感じですね。ここに具体的に数式を書いています、i番目というのはここでは特段関係ないのですけれども、特徴量がmレベルまで取るものに対する値を取るとして、各レベルに対応するダミー変数を  $d_1$  から  $d_m$  とします。つまり、ダミー変数  $d_j$  は、その特徴量が j に所属すれば1になるし、それ以外は0になるということです。

AGLM - Accurate GLMとは  
 - 3つのコンセプト ②Oダミー変数

- 順序関係を導入した特殊ダミー変数
  - 順序型ダミー：Ordinal Dummyから**Oダミー変数**と呼ぶ。
  - なお、通常のダミー変数を**Uダミー変数**（Usual Dummy）とする。
- Uダミー変数はお馴染みの、カテゴリカル変数に対して一般的に用いるもの
- 具体的に
  - $i$  番目の特徴量  $x_i$  が、レベル  $\{1, 2, \dots, m\}$  に対する値をとるとする。
  - 各レベルに対応するダミー変数を  $d_{i,1}, d_{i,2}, \dots, d_{i,m}$  とする。

$$d_{i,j}^U = \begin{cases} 1 & \text{if } x_i \in j \\ 0 & \text{otherwise} \end{cases}$$

16

一方でOダミー変数は、このように定式化されます。先ほどの  $x_j$  は、 $j$  だけに所属している場合だったのですが、1から  $j-1$  番目のレベルに属するときは1を取り、それ以外の場合は0を取ります。数式だと分かりにくいかもしれないので、表でまとめると、このような感じになります。上三角のように1が並んでいることが、見て取れると思います。属するレベルより小さいレベルを、全て1とするようなダミー変数を作ります。

AGLM - Accurate GLMとは  
 - 3つのコンセプト ②Oダミー変数（続き）

- Oダミー変数は次のように定式化される。

$$d_{i,j}^O = \begin{cases} 1 & \text{if } x_i \in \{1, 2, \dots, j-1\} \\ 0 & \text{otherwise} \end{cases}$$

- つまり、属するレベルより小さいレベルはすべて1とする。

レベル	$d_{i,1}^O$	$d_{i,2}^O$	$d_{i,3}^O$	...	$d_{i,m-1}^O$	$d_{i,m}^O$
1	0	1	1	...	1	1
2	0	0	1	...	1	1
3	0	0	0	...	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$m-1$	0	0	0	...	0	1
$m$	0	0	0	...	0	0

17

これの何がいかをご説明します。Oダミー変数の効能としては、Uダミー変数で失われてしまう順序関係を保持することです。具体例で示したいと思います。特徴量は、年齢の場合を考えて、レベルを10歳

刻みで設定したいと思います。10代はレベル1、20代はレベル2、80代はレベル8といった具合で、例えばこのような図をごらんください。35歳と45歳を例に挙げているのですが、35歳は30代なので、レベル3に該当するし、45歳は40代なので、レベル4に該当します。Oダミー変数の先ほどの定義式にのっとると、このようにダミー変数を設定することができるのですが、例えば $d_4$ という所に着目してみると、 $d_4$ の特徴量として選択する・しないかは、30代と40代で年齢の説明力に差をつけるか、つけないかということに相当します。なぜかという、 $d_4$ を選択しないとすると、35歳と45歳のダミー変数の並びが一緒になりますね。ですから、30代と40代は説明力に差がないということになるわけです。

**AGLM - Accurate GLMとは**  
 - 3つのコンセプト ②Oダミー変数 (続き)

• Oダミー変数の効能 **ポイントQ**  
 - Uダミー変数では失われてしまう順序関係を保持する。

【例】特徴量を年齢とし、レベルを10歳刻みで設定  
 (10代 : レベル1, 20代 : レベル2, ..., 80代 : レベル8)

年齢	レベル	$d_1^O$	$d_2^O$	$d_3^O$	$d_4^O$	$d_5^O$	$d_6^O$	$d_7^O$	$d_8^O$
35	3	0	0	0	1	1	1	1	1
45	4	0	0	0	0	1	1	1	1

$d_4^O$ を特徴量として選択する or しない  
 ⇕  
 30代と40代で年齢の説明力に“定量的な”差をつける or につけない

18

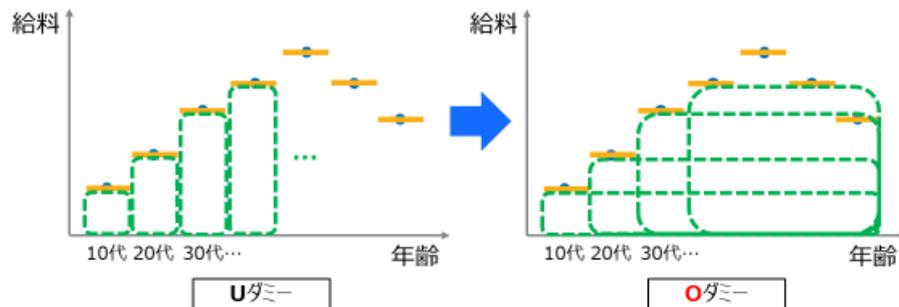
Uダミー変数では個々の特徴量を独立に表すので、解釈はしやすいかもしれませんが、このような意味で順序関係はなくなってしまいます。それをイメージ図で表したものが、こちらになります。先ほど離散化したグラフがこれなのですが、Uダミー変数の場合は、それぞれ独立にやっていくわけですね。一方、Oダミー変数の場合はどうかというと、隣のものも合わせて説明変数を予測していきます。つまり、隣接するレベル間に対する意味づけを持たせることで、表現力が向上するということです。

これは、アクチュアリー視点に立つとすると、リスク区分型の商品が当たり前の時代になっていますが、例えば20代と10代で、20代は10代の何倍くらいリスク量なのかと考えるような場合、必ず隣接する年齢群団のところを考えますね。しかし、Uダミー変数の場合は、それを独立にやってしまうため、その関係が不明確になるといったデメリットがあるのです。

AGLM - Accurate GLMとは  
- 3つのコンセプト ②Oダミー変数 (続き)

- Oダミー変数の効能 **ポイントQ**
  - Uダミー変数では失われてしまう順序関係を保持する。

- イメージ図



→表現力が向上する (隣接するレベル間に対する意味付けが可能となる)。

19

最後に、正則化についてです。正則化とは、いわゆる罰則付回帰のことで、線形回帰モデルの場合ですと、目的関数の二乗誤差に係数に関する正則化項を付加したのになります。GLM の場合は、対数尤度関数に正則化項を付加したものです。正則化項を加えることで、目的関数の最小化という条件を一定程度緩和して、その代わりに係数の  $\beta$  を縮小する仕組みになっています。

正則化項の代表例としては、第1部にも出てきた LASSO が挙げられます。これは、L1 ノルムに基づく正則化項です。そして、Ridge、Elastic Net というものが代表的です。Ridge はL2 ノルムに基づくもので、Elastic Net は、LASSO と Ridge の折衷になります。正則化項に  $\lambda$  が入っているのですが、これはハイパーパラメータと呼ばれていて、分析者が設定するもので、 $\lambda$  の値を複数変えたモデルを構築して、クロスバリデーションにより決定するものです。

LASSO も Ridge も、変数選択または係数縮小の効果を持ちます。これらは Shrinkage Method というのですが、LASSO はパラメータ推定と変数選択を同時に行い、先ほどの小島さんの発表にもあったのですが、係数を0にする特徴があります。Ridge は、係数は必ずしも0になるとは限らないのですが、相関のある係数の組を同時に縮小する特徴があります。いずれも特徴量間に強い相関がある場合、つまり多重共線性でも結果が安定するといわれているものです。

正則化のいいところは、多数の特徴量から真に効果のあるものを機械的に変数選択することです。先ほどの佐野さんの第2部の説明で、延々と続く変数選択がどうにかならないかということをおっしゃっていましたが、普通の GLM を組むときも恣意的に変数を選んだりする場面があると思うのですが、この正則化をやることで、真に効果のあるものを自動的に選択してくれるという効果があるのです。特に LASSO や Elastic Net の場合ですけれども。

## AGLM - Accurate GLMとは - 3つのコンセプト ③正則化

### • 罰則付回帰

- ✓ 線形回帰モデルの場合、目的関数の二乗誤差に正則化項（罰則項） $p_\lambda(|\beta|)$ を付加する。

$$\sum_{i=1}^n \{y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip})\}^2 + p_\lambda(|\beta|)$$

- ✓ GLMの場合は、対数尤度関数に正則化項を付加する。
- ✓ 正則化項の導入により、**係数縮小**の効果が得られる。

### • 正則化項の代表例（ $\lambda$ はハイパーパラメータ）

Lasso (L1)	$p_\lambda( \beta ) = \lambda \sum_{i=1}^p  \beta_i $	一部の係数を0にする (変数選択)
Ridge (L2)	$p_\lambda( \beta ) = \lambda \sum_{i=1}^p  \beta_i ^2$	相関のある係数を 同時に縮小する
Elastic Net (L1-L2)	$p_\lambda( \beta ) = \lambda_1 \sum_{i=1}^p  \beta_i  + \lambda_2 \sum_{i=1}^p  \beta_i ^2$	LassoとRidgeの 折衷法

21

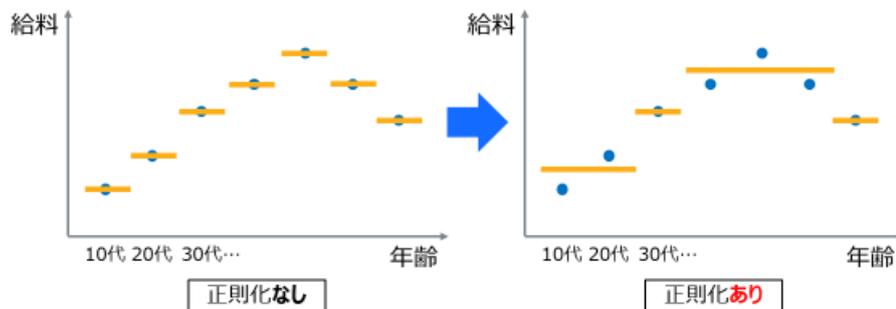
イメージ的には、今まで離散化と0ダミー変数でモデルを細かく柔軟にし、ある意味モデルを凝ったものに行っているのですが、過剰な細かさは、オーバーフィッティングしてしまう可能性もあるため、なるべく避けたいです。そのときに正則化を行うことで、イメージとしては、この図のような感じです。本当に真のあるものだけ、例えば10代と20代は一緒のものにしてしまうという効果があって、平滑化する特徴があるということです。つまり、訓練データのオーバーフィッティングを回避して平滑化するということが、正則化の効能です。特にこの要素が、自動的に行うものですので、機械学習的と言えるかもしれません。この機械学習手法が、今までの離散化と0ダミー変数をカバーする役割があると考えています。

## AGLM - Accurate GLMとは - 3つのコンセプト ③正則化（続き）

### • 正則化の効能 **ポイント**

- 多数の特徴量から真に効果のあるものを機械的に変数選択する。

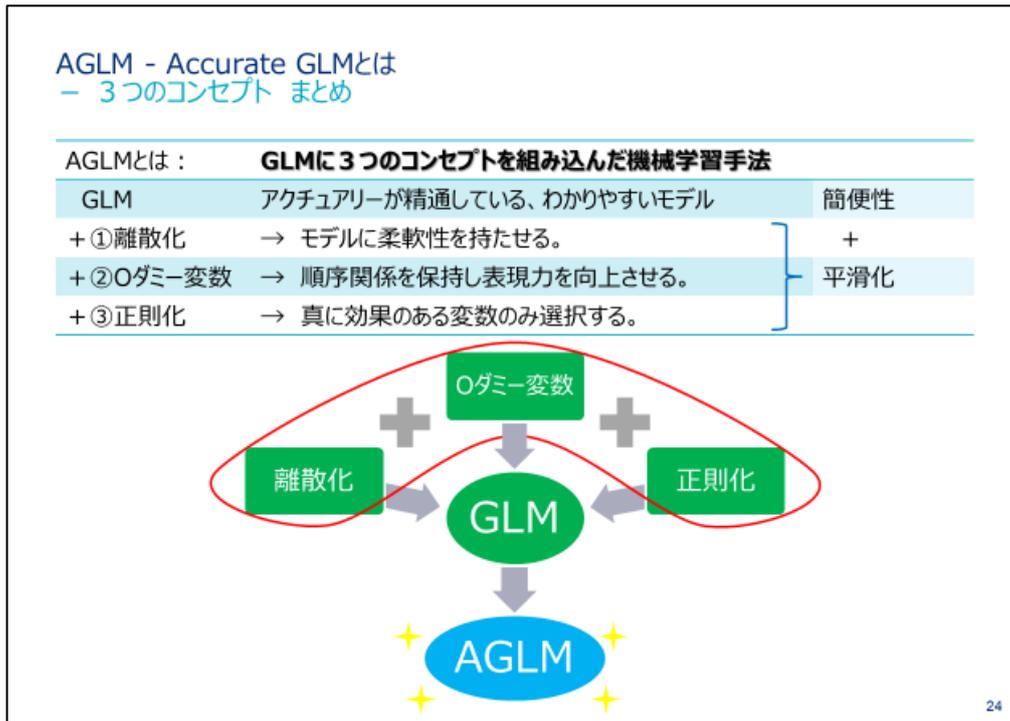
### • イメージ図



→訓練データの過学習（オーバーフィッティング）を回避し平滑化する。

22

以上をまとめますと、AGLMは、GLMに三つのコンセプトを組み込んだ機械学習手法です。GLMは、アクチュアリーが精通していて分かりやすいモデルです。離散化はモデルに柔軟性を持たせ、Oダミー変数は、順序関係を保持することで隣接関係を考慮した表現力を向上させるものです。正則化は、真に効果のある変数を自動的に選択します。この意味で、ベースとなるモデルは簡便で説明しやすく、それに特徴量処理の工夫と機械学習手法を追加することで、平滑化を実現するというモデルになっています。これは、先ほどの図で、はてなマークを取ったものですが、これら一つ一つがGLMに効果を働かせているわけではなくて、三つ同時に組み合わせることで、AGLMとしての真の効果を発揮することを示しています。AGLMは、アクチュアリー目線に立った、やりたいことができていない手法なのではないかと考えています。



具体例として、また特徴量が年齢の場合を考えてみましょう。離散化により、年齢を10歳刻みでグルーピングされたとします。10代がレベル1、20代がレベル2といった具合です。Oダミー変数により、順序関係を設定します。そして正則化によって、例えば  $d_3$ 、 $d_6$ 、 $d_8$  が選択されたとします。これはどのようなことかということ、10代・20代というグループと、30・40・50代というグループ、60代・70代というグループ、80代というグループにグルーピングされたことになります。ある意味でこれは、リスク区分が自動的に行われていることになります。離散化では、あらかじめできるだけ細かくした方がいいと申し上げたのですが、これはなぜかということ、最初に細かく刻む所はなるべく細かくしておいた方が、後々の正則化のステップで自動的にくっつけてくれるということが理由でした。以上が、AGLMのご紹介です。

## AGLM - Accurate GLMとは - 3つのコンセプト まとめ (続き)

- ① 離散化 + ② Oダミー変数 + ③ 正則化のシナジー効果 **ポイントQ**

【例】特徴量が年齢（10歳～89歳）の場合を考える。

① 離散化により、10歳刻みでグルーピングされたとする。

→ [10...19歳] → レベル1, [20...29歳] → レベル2, ..., [80...89歳] → レベル8

② Oダミー変数により、順序関係を設定する。

レベル	$d_1^O$	$d_2^O$	$d_3^O$	$d_4^O$	$d_5^O$	$d_6^O$	$d_7^O$	$d_8^O$
1	0	1	1	1	1	1	1	1
2	0	0	1	1	1	1	1	1
3	0	0	0	1	1	1	1	1

③ 正則化により、 $d_3^O, d_6^O, d_8^O$ が選択されたとする（それ以外は選択されなかったとする）。

→ [10,20代], [30,40,50代], [60,70代], [80代]でグルーピングされたことになる。

ある意味、「リスク区分（グループ内の平準化）」が自動的に行われている。

自動的にグループをくっつけてくれるので、予めできるだけ細かく離散化した方が良い（前述）。

25

では、ASTIN 関連研究会でどのような活動をしてきたか、ご紹介したいと思います。『AGLMの研究』と題して、十数名から成るチームで取り組みを実施しました。AGLM そのものの基本プログラムを実装したり、少しアドバンスな内容をやったり、比較のための既存手法の調査として、GLM や決定木、CART と呼ばれるものや GAM などを調べたり。CART や GAM は第1部で出たのですが、覚えていらっしゃるでしょうか。それから、検証するためのデータの整備や、学会発表等に向けて文書化などを行いました。

## ASTIN関連研究会での取り組み - 『AGLMの研究』

- ① 10数名からなるチームで、以下の取り組みを実施

① 基本プログラムの実装

→ オリジナルの“aglm”関数、様々なモデルの比較プログラムを実装

② L0およびL0-L1正則化プログラムの実装

→ 既存パッケージにはない正則化プログラムを独自で実装

③ 既存手法の調査（GLM, CART, GAMなど）

④ 離散化手法の調査（別の調査研究として発展する可能性！）

⑤ 検証用データの整備（オープンデータ収集、人工データ作成）

⑥ 文書化、モデル選択の調査

→ 論文執筆、学会発表に向けて

27

このうち、ここでは基本プログラムの概要について、ご紹介したいと思います。プログラミング言語は、Rを使用しました。用いたデータは、保険関係の八つのデータをインプットして、頻度、損害規模、総損害

額を予測するモデルを作りました。このデータは、保険数理関連書籍に付属されているデータや、最近流行っている kaggle のコンペティションのデータ等を拝借しました。

そして、AGLM とさまざまな既存モデルとで、予測精度を比較します。さまざまな既存モデルは、正則化項で、Ridge、LASSO、Elastic Net と、従来の GLM に正則化項を変えたものと、CART と GAM という、九つのモデルを比較します。AGLM もこの3つの正則化項を適用します。「with full」と書いてあるものは、全変数を使う GLM で、従来の GLM ということです。また、予測精度の評価は、機械学習やデータサイエンスの文脈では、用いた分布に基づく誤差、逸脱度がより小さい方がいいモデルとしているので、それに倣って、今回もそのようにしています。頻度は、Poisson 分布を仮定しているので、Poisson 逸脱度で評価して、総損害額については Tweedie 逸脱度で評価しました。

### ASTIN関連研究会での取り組み - 基本プログラムの概要

- プログラミング言語は“R”を使用
- 保険関連のデータ計 8 つをインプットし、頻度/損害規模/総損害額を予測する。
  - 保険数理関連書籍の付属データや過去の kaggle コンペティションのデータなど
- AGLM と様々な既存モデルで **予測精度を比較** する。
  - AGLM (with Ridge, Lasso, ENET)  
vs. GLM (with full, Ridge, Lasso, ENET), CART, GAM の 9 つ
- 予測精度の評価は、用いた分布に基づく逸脱度とする（小さい方が良い）。
  - 頻度：Poisson 逸脱度、総損害額：Tweedie 逸脱度など

28

これらの純粋なモデルそのものの能力を見ることにしました。すなわち、データサイエンスのコンペなどでは、よく複数モデルをブレンドしたりする場合があるのですが、モデルの純粋な能力を見るための実験になります。用いたデータはこのような八つのもので、自動車保険のクレームデータ、傷害保険のクレームデータ、対人賠償のクレームデータなどを使っています。これらのデータはオープンデータだったりしますので、欠損値などはあまり含まない、比較的きれいなデータでした。また、これから新たな特徴量は作ったりはせずに、データ中に含まれる特徴量をそのまま用いました。

## ASTIN関連研究会での取り組み - 基本プログラムの概要 (続き)

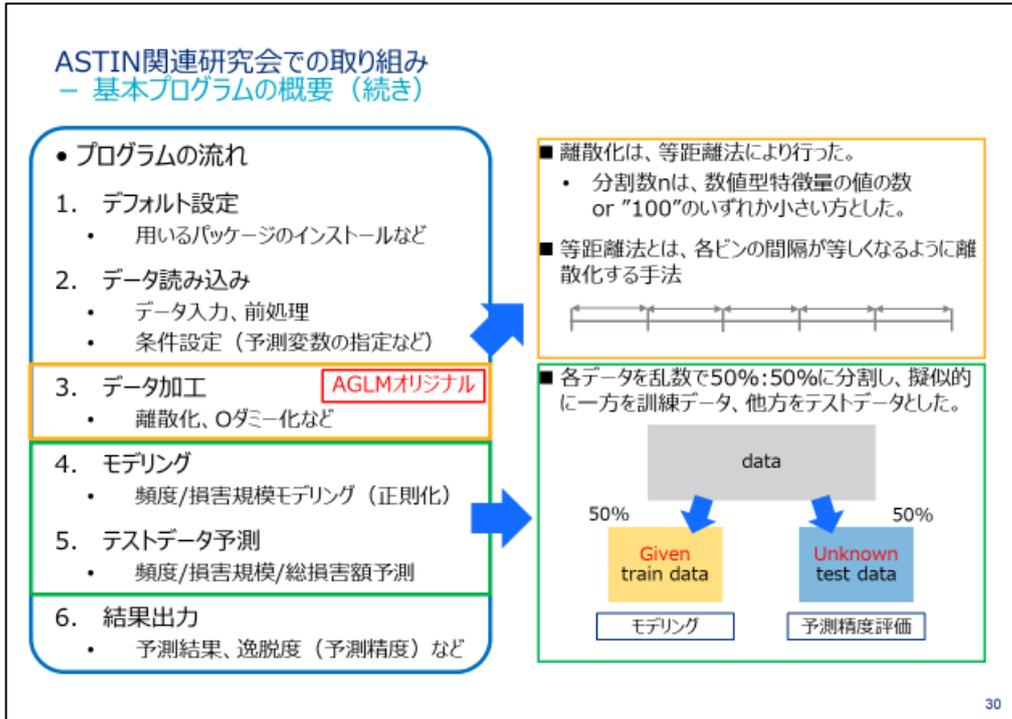
### • 入力データ

No.	データの内容	レコード数	予測変数	特徴量例	出典
1	自動車保険クレームデータ	40,760	頻度/損害規模	年齢、性別、 車体長さ、燃料	Predictive Modeling vol.2
2	オートバイ保険クレームデータ	64,548	頻度/損害規模	年齢、性別、 地域、車両タイプ	Non-Life Ins Pricing with GLM
3	自動車保険クレームデータ	67,856	頻度/損害規模	年齢、性別、 車両価格、車齢	MACQUARIE Univ.
4	傷害保険クレームデータ	22,036	損害規模	傷害程度、 事故～報告日数	MACQUARIE Univ.
5	自動車保険クレームデータ	164	損害規模	等級、車種、 エンジンタイプ	kaggle
6	医療費データ	1,338	損害規模 (費用)	年齢、性別、BMI、 子供の数、喫煙有無	kaggle
7	対人賠償保険クレームデータ	413,169	頻度/損害規模	年齢、車齢、 車種、燃料タイプ	R package/"CASdatasets"
8	自動車保険クレームデータ	30,595	頻度/損害規模	年齢、車体種類、 用途、エネルギー	R package/"CASdatasets"

- 欠損値などは含まない比較的綺麗なデータ
- 新たな特徴量生成などは行わず、データに含まれる特徴量をそのまま用いた。

29

プログラムの流れとしては、次のとおりです。まず、用いるパッケージをインストールしてデータを入力して、予測変数をどれにするか指定を行います。次にデータの加工ですね。どれを離散化するか、どれを0ダミー変数化するか。そして、モデリングして、テストデータの予測を行って、最後に結果を出力・評価するといった流れです。特にデータ加工の3番目がAGLMのオリジナルのところでした。離散化には等距離法を使っています。これは各ビンの間隔が等しくなるように離散化するもので、分割数nは、数値型特徴量の数か100の、いずれか小さい方としました。このモデリングの部分では、各データを乱数で50%と50%に分割して、擬似的に一方を訓練データ、他方をテストデータとして、訓練データをモデリングに用いて、テストデータを予測精度の評価に用いました。



まず、AGLM を構成する三つのコンセプトの変動要因分析についてご紹介します。一つめの自動車保険のデータを使った頻度モデルの例です。ここで Poisson 逸脱度の推移はどのようになったかと申し上げますと、一番左が従来の GLM の逸脱度で、離散化、Oダミー化、正則化と右に行くにつれて書いていますが、離散化とOダミー化をすると、モデルに柔軟性は出るものの、モデルが凝ったものになってしまって、Oダミー化の部分で少し値が大きくなって、オーバーフィッティングしてしまうのですが、そこで正則化がカバーすることで、最後に通常の GLM より小さい逸脱度の値になっています。ここでは、正則化は LASSO を使って、罰則の  $\lambda$  のハイパーパラメータは、クロスバリデーションによって逸脱度が最小になるものとししました。3要素を組み込んだ AGLM の予測精度が、一番良いことを確認しています。イメージ図としては、このような感じですね。

ASTIN関連研究会での取り組み  
 - 数値実験と考察

• AGLMを構成する3要素の変動要因分析

- データNo.1の頻度モデルの例

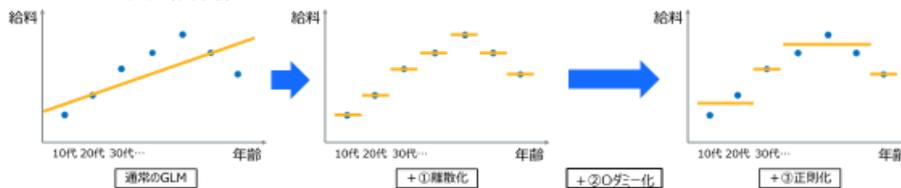
- Poisson逸脱度の推移

3要素を組み込んだAGLMの予測精度が良いことを確認!

	通常のGLM	+①離散化	+②Oダミー化	+③正則化 =AGLM
Poisson逸脱度	0.39128	0.39684	overfitting	0.39037

※正則化はLassoとした。  
 ハイパーパラメータは交差検証によって逸脱度が最小になるものとした。

- イメージ図 (再掲)



31

続いて、AGLMと既存手法の計九つのモデルを比較した例です。ここでは、時間の関係上、頻度モデルの例のみをお見せしたいと思います。GLMではlogリンク関数とPoisson分布を使用しました。テストデータに対する予測精度の単純なランキングをお見せしたいと思います。Poisson逸脱度が最小のものを1位として、最大を9位として、単純なランキングを並べたものです。頻度モデルの例なので、一部のデータは頻度データがないため、割愛させていただいているのですが、五つのデータを用いた結果がこれです。万能なモデルは必ずしも存在しないことは、皆さんご存じだと思うのですが、その中でもAGLMは、それぞれのデータのランキングに対して一番右側でアベレージ・ランクを書いているのですけれども、平均的に上位のものになっていることが分かると思います。1位がAGLMのElastic Netで、2位がAGLMのLASSOで、3位がAGLMのRidgeと並んでいて、AGLMの有効性を確認できました。

平均的にモデルの性能がいいということは、どのようなことかということ、汎用性が高いことを示すと考えています。汎用性が高いということは、ある決まったデータに対してだけ非常にいいというわけではなくて、他のデータに対しても応用が利くという意味です。よく例えられることが、宿題だけできるのではなくて、本番のテストに強い人が評価されると思うのですが、まさにそのようなことが、AGLMに言えるのではないのでしょうか。

ASTIN関連研究会での取り組み  
 - 数値実験と考察 (続き)

- AGLM vs. 既存モデル：頻度モデルの例
  - logリンク関数とPoisson分布を使用
  - テストデータに対する予測精度ランキング (Poisson逸脱度が最小：1位、最大：9位)

Frequency Model:	Data1	Data2	Data3	Data7	Data8	Ave. Rank
① GLM w/ full	7	8	8	9	9	8.2
② GLM w/ Ridge	5	7	4	8	1	5.0
③ GLM w/ ENET(alpha=0.5)	3	6	5	7	4	5.0
④ GLM w/ Lasso	4	5	6	6	3	4.8
⑤ AGLM w/ Ridge	8	3	1	4	2	3.6
⑥ AGLM w/ ENET(alpha=0.5)	1	2	3	2	6	2.8
⑦ AGLM w/ Lasso	2	4	2	3	5	3.2
⑧ GAM	6	1	7	1	8	4.6
⑨ CART	9	9	9	5	7	7.8

※Data4-6は頻度データがないため、対象外とした。



1位：AGLM w/ ENET 2位：AGLM w/ Lasso 3位：ALGM w/ Ridge  
 となり、AGLMの有効性を確認

32

まとめですが、AGLMは、特徴量が多いデータに対しても、予測性能の高いモデルを自動的に作り出す、汎用的で、簡便で、分かりやすい手法なのではないかと思います。上の部分は、数値実験で効果を確認して、自動的に作り出す汎用的な三つのコンセプトで平滑化を実現していることから言えるかと思えますし、簡便で分かりやすいということは、何よりも皆さんが常日頃から使っているGLMがベースとなっているところです。更に取り組みたい分析等も考えているのですが、ここでは割愛させていただきます。

ASTIN関連研究会での取り組み  
 - 数値実験と考察 (続き)

- 数値実験から、AGLMの予測精度の高さや汎用性の高さを確認することができた！

- AGLMは、
    - 特徴量が多いデータに対して
    - 予測性能の高いモデルを
    - 自動的に作り出す汎用的で
    - 簡便でわかりやすい手法
- 数値実験で効果を確認  
 3つのコンセプトで平滑化を実現  
 GLMがベースとなっている

- さらに取り組んでいる (or 取り組みたい) 分析
  - 交互作用項などの新たな特徴量を導入
  - LO正則化などの導入
  - 保険関連データ以外のデータへの適用 etc.

33

最後に、第1部の小島さんのお話から第2部の佐野さんのお話、第3部を通じて、アクチュアリー視点での機械学習手法やモデル選択について、ずっと考えてきました。AGLMは、アクチュアリーとしてどのよ

うな手法が求められているかという観点を踏まえて、何らかの可能性を示唆するのではないかと思います。アクチュアリーにとって、やりたいことの一部ができていない手法ではないでしょうか。何よりも、AGLM のこのような考え方が、機械学習の手法をアクチュアリーの実務に導入するヒントになればと思っています。アクチュアリーにおける機械学習の手法の可能性を、引き続き皆さんで模索していければと思っています。ご清聴ありがとうございました。

## おわりに

- 本セッションを通じて、
  - アクチュアリー視点での機械学習手法、およびモデル選択について考えました。
  - 「アクチュアリーとしてどのような手法が求められているか」という観点を踏まえ、
  - 我々が提案するAGLMは、何らかの可能性を示唆するものではないでしょうか。
  - アクチュアリーにとってやりたいことの一部ができていない手法ではないでしょうか。
- アクチュアリーが機械学習手法を導入するヒントになれば幸いです。

これからもアクチュアリーにおける機械学習手法の可能性を  
皆さんで模索していきましょう！



35

司会 藤田さん、ありがとうございました。ご質問等がございましたら、挙手をお願いいたします。

ヒラマツ アクサ生命のヒラマツと申します。大変貴重なお話を、ありがとうございました。1点質問がございまして、AGLMの1番と2番の効果がくっついたけれども、オーバーフィッティングしてしまって、3番の罰則を付けたら良くなったというスライドが、中盤くらいにありました。これは、通常のGLMに1番と2番を飛ばして3番を付けた場合の数値はどれくらいだったのか、既に実験はされていますか。

藤田 ありがとうございます。ここに確かに載せるべきで、ごもっともなご質問だと思いますが、実はやっております。GLMに単に3番を付けた結果は、具体的な数字を述べますと、0.39063となっていて、その差をどのように見るかということは、また別の議論になると思いますが。

ヒラマツ しかし、基本的には、それよりは良くなっているということですね。ありがとうございます。

司会 ありがとうございます。他にご質問はございますか。

藤田 第2部に関するご質問でも大丈夫です。

B 貴重なお話を、ありがとうございました。離散化するときに10代、20代、30代で切っていたかと思うのですが、一番予測の精度を上げようと思うと、各歳幅になるのかなと思うのです。その場合、例えばこのデータで切った時は25歳くらいに壁があったと出て、それがある程度説明できるものであればいいと思うのですが、数年後同じようなことをした時に、次は27歳に壁ができたとなったときに、27を果たして使えるのかどうかということは少し悩ましいと思ったのですが、そのあたりは、何かお考えはありますでしょうか。

藤田 モデリングしたときは25歳で壁のようなものがあったことに対して、新しく適用するデータに対しては、未知のパターンがある場合、ということでしょうか。

B それもそうですし、私は年金のアクチュアリーなので、同じ会社の給与を推計するという、昇給のモデルを作ることがよくあるのです。その場合、会社の構造が変わると、ぽこっと上がる所が変わったりすることがあって、経験的に、規程類を読んで「ここで変わるのが自然だよ」という所で昇給を折ったりするわけです。それが自動的にできるのはいいのですが、5年前に切った時はちょうど25歳だったのに、この手法を導入して切った時には27歳だったとして、前と同じ規程で、同じ会社で27歳で果たして切れるかという、少し悩ましいと思ったので、そのようなことに関して何かしらのアイデアがあれば、お伺いしたいと思ったのです。

藤田 ありがとうございます。今のご質問は、つまりデータの中の属性が新しく変わった場合ということですね。

B そうですね。

藤田 よくある対策としては、未知なものが来ても、別の特徴量をあらかじめ恣意的に用意しておいて、新しく来たものに関しては、それをベースに対処しておくことを考える場合があるようです。また、実際に手元にあるデータがどのようなものなのか、事前によく分析しておくことが何より大事でして、いわゆる探索的データ解析と言ったりしますが、今回はそれを深掘りしていないので、そこに対する議論が必要だと思いますが、そのような前処理を、もう少し緻密にする必要があるかと思います。

B ありがとうございます。

司会 ありがとうございました。時間ももう少しなのですが、ご質問はございますか。

コンドウ 東京海上日動あんしん生命のコンドウと申します。今日は、大変素晴らしいプレゼンテーションをありがとうございました。一つ質問させていただきたいことは、データのリストがあるスライドがあったと思います。この手法では、最初に離散化ということでもかなり変数を増やしたり、ダミー変数を入れていくということもあるので、変数の数がかかなり増えていくので、正則化を入れて選択するときにも計算時間がかかってくると思うのです。例えば、一番大きいところで40万件くらいのデータを扱われていると思うのですが、そのあたりの計算時間の感じや、あるいは、更にデータが増えたときにどのようなことが

考えられるのか、教えていただけるとありがたいと思います。

藤田 ありがとうございます。実際に離散化を行うことによって、現時点での数値実験では、お示したように、数値型特徴量の値の数か100の、いずれか小さい方で分割しました。ですから、100以上の値を取るものに関しては、100で打ち切られてしまうという事実があるのですが、実際にそれをベースにしてやると、計算時間的には高々数十分程度です。現時点で、データのうち最多の40万件使っていた場合でそのくらいです。AGLMは、数値型特徴量を単に使うだけではなくて、それを離散化して、説明変数の数を大きくしているので、計算時間は普通のGLMよりも長くなってしまふことは避けられないのですが、現実的なものになっていると考えています。

あとは、離散化のビン数をどのようにするかというところに次の議論が来るのかなと思っていて、それが100でいいのか、もしくは1,000まで耐えられるのかというあたりは、今後の課題だと思います。

コンドウ ありがとうございます。

司会 ありがとうございます。他は、ご質問はございますか。特になければ、こちらでこのセッションを終わらせていただきたいと思います。小島さん、佐野さん、藤田さん、ありがとうございました。最後に拍手をお願いいたします。