プレディクティブモデリングの保険データへの応用
（ASTIN COLLOQUIUM 2017 参加報告）

日本生命保険相互会社　遠藤　史博

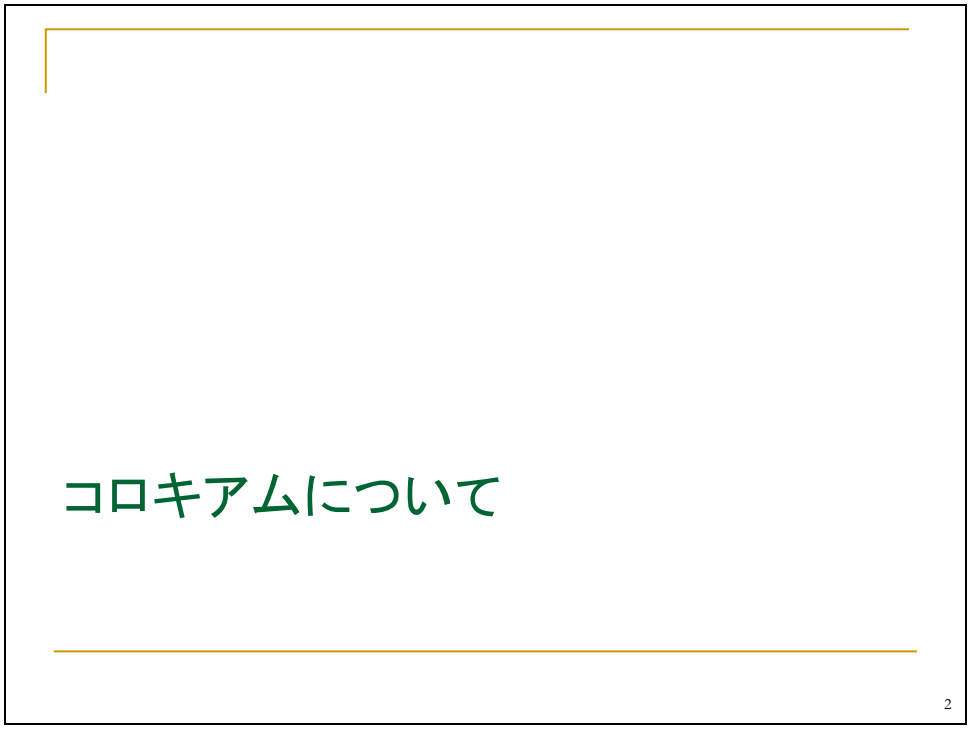
スマートニュース株式会社　小田　秀匡

【司会】　時間となりましたので、セッションＦ、「プレディクティブモデリングの保険データへの応用（ASTIN COLLOQUIUM 2017 参加報告）」を開始します。発表者は、ASTIN 関連研究会のメンバーである、日本生命の遠藤さんとスマートニュースの小田さんです。

それでは遠藤さん、よろしくお願いします。

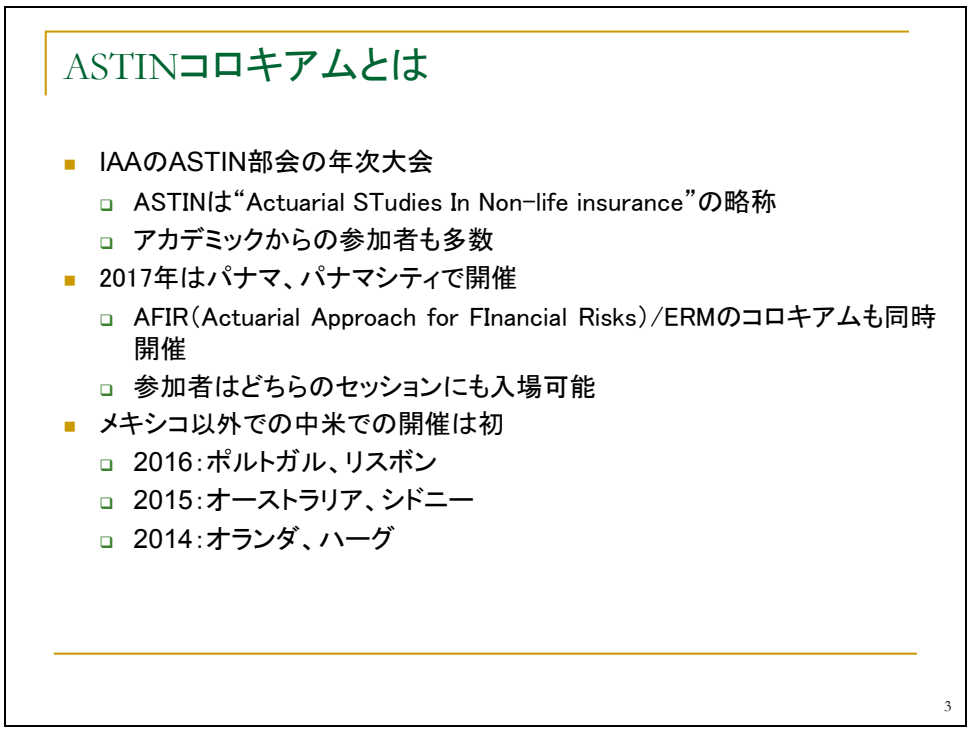
(1-1)

【遠藤】　ご紹介いただきました、日本生命の遠藤と申します。よろしくお願いします。



(1-2)

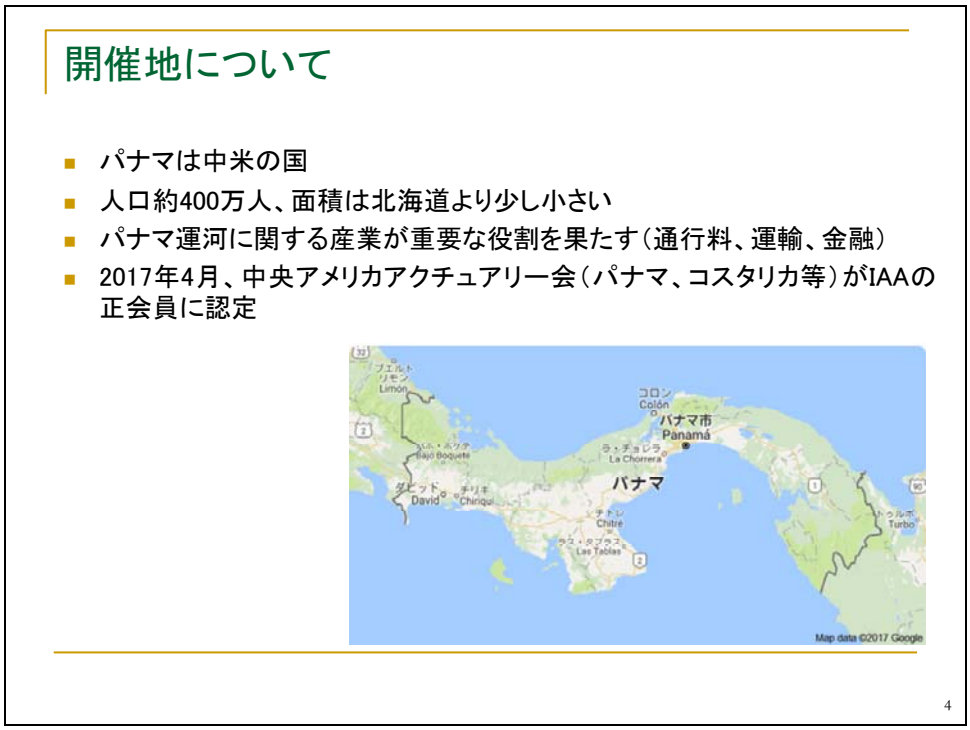
今年、私とスマートニュース株式会社の小田さんの 2 名は、ASTIN 関連研究会より、ASTIN コロキアム 2017 に派遣をされました。今回発表する「プレディクティブモデリングの保険データへの応用」は、パナマで行われましたこのコロキアムで報告したものとなります。まず、この ASTIN コロキアム 2017 の概要についてお話しいたしまして、それから私と小田さん、それぞれが発表した内容をご説明するという流れで進めてまいります。



(1-3)

まず、ASTIN コロキアムはどのようなものかといいますと、IAA の中の ASTIN 部会の年次大会でございます。ASTIN は、スライドにあるように、「Non-life insurance」という言葉が入っていますが、近年は生保

や投資分野などからもメンバーが入ってきておりまして、今回は、実務家だけではなく大学など、アカデミックからの参加者も多数おりました。今年度はパナマで開催されましたが、AFIR/ERM のコロキアムも同時に開催されておりまして、参加者は、どちらのセッションにも入場可能でした。なお、メキシコ以外での中米の開催は初でありまして、過去 3 年間はスライド記載のところで開催されておりました。

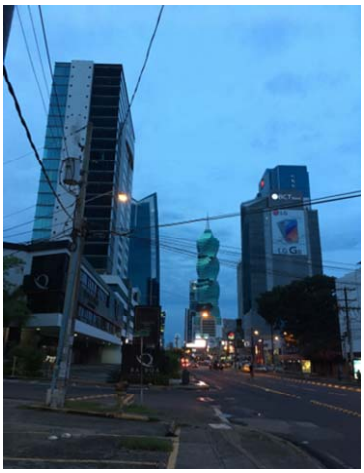


(1-4)

パナマはなじみのない国だと思いますので、簡単に開催地についてお話しさせていただきます。中米の国、この地図のくびれているところがパナマなのですけれども、人口は約 400 万人で、面積は北海道より少し小さい程度です。パナマ運河は有名だと思うのですが、これに関する産業が重要な役割を果たしておりまして、この通行料だけでも GDP の 7、8%で、パナマ運河に関連する貿易や金融を合わせると、8 割程度がこのパナマ運河によった経済になっていると、一説では言われております。実は、今年の 4 月に、パナマやコスタリカなどのアクチュアリーで構成される、中央アメリカアクチュアリー会が、IAA の正会員に認定されまして、今回のコロキアムのパナマでの開催というのも、これが一因でもあったのかもしれません。

開催地について

- 市内の様子



5

(1-5)

このスライドは、市内の様子です。中米のシンガポールと呼ばれることもあるぐらい、経済発展が目覚ましい国で、こういった高層ビルなども、市内では見ることができました。ただこういった高層ビルは、大半が銀行のビルであり、保険業については、まだ発展途上という印象をセッションの中等でも受けました。

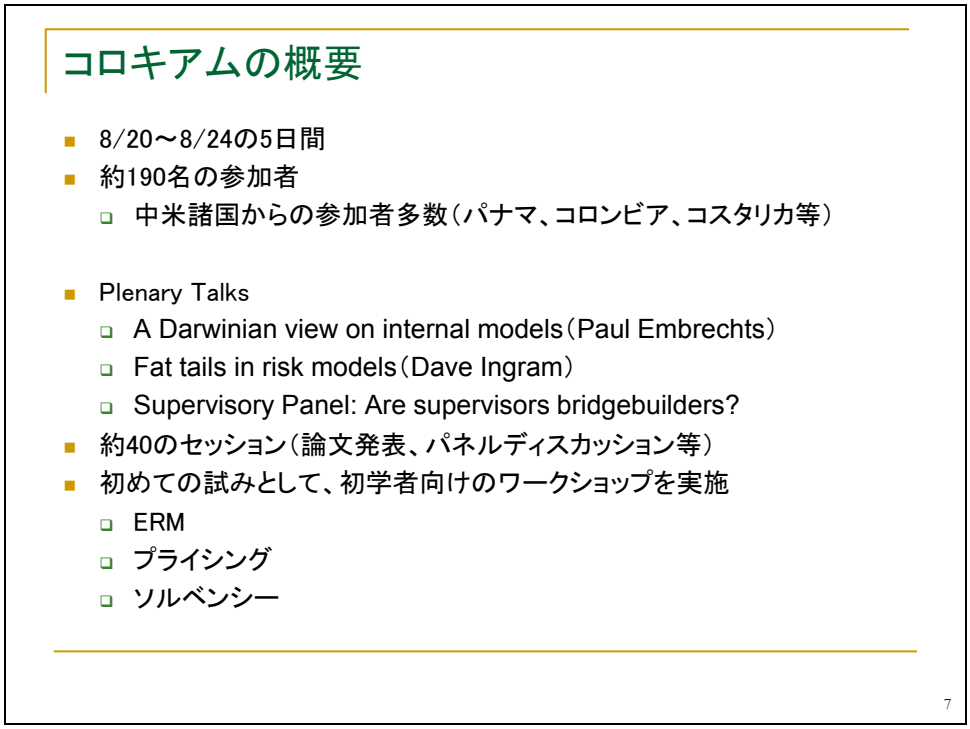
開催地について

- パナマ運河

6

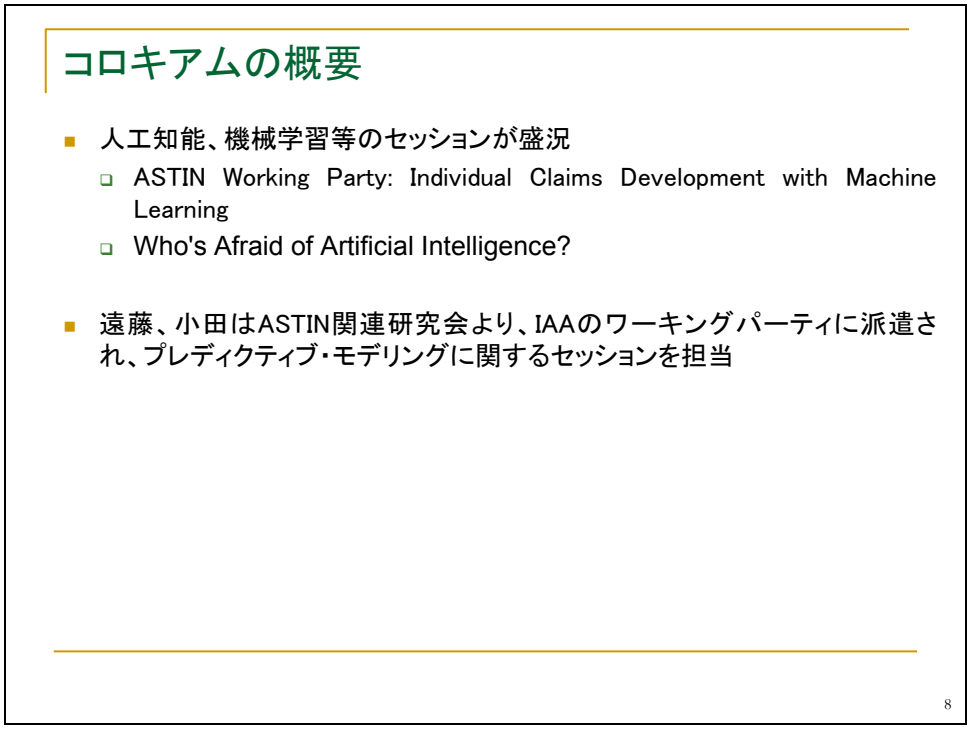
(1-6)

このスライドはパナマ運河で、コロキアムに参加しますと、今回は 2 日目の午後だったのですが、全員で巡るシティーツアーというものがありまして、パナマ運河など、そういった観光名所を回らせていただいて、参加者の親睦を深めたり、開催地への造詣を深めたりと、よい経験となりました。



(1-7)

コロキアムの概要です。8 月 20 日から 24 日の 5 日間、約 190 名の参加者でした。場所柄、パナマ、コロンビア、コスタリカなど、中米諸国からの参加者も多数いらっしゃいました。コロキアムは、大きく 3 つのパートに分かれております。「Plenary Talks」はキーノートスピーチのようなものです。全員参加で、一つの話を聞きます。中でも「Supervisory Panel」というものは、パナマやメキシコやカナダなどの保険監督官が集まったパネルディスカッションで、特に発展途上国において、どのように保険監督者と保険会社が関係を築いていくかという内容で、大変興味深く聞くことができました。それから論文発表がありまして、今回初めての試みとして、初学者に向けたワークショップが実施されています。ERM やプライシングなどテーマを決めて、3 コマから 4 コマ、5 時間ぐらいの時間を当てて、概要から少し深入りしたところまでをさらうといった内容で行われております。

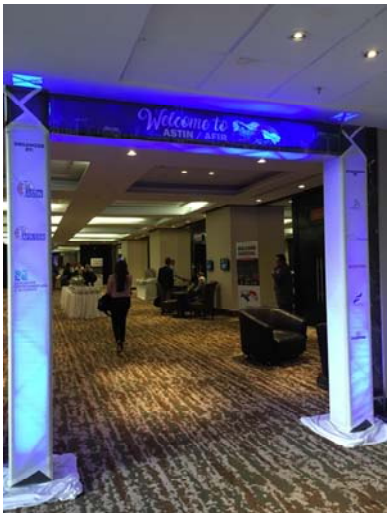


(1-8)

セッションに関しては、人工知能や機械学習のセッションがとても盛況でした。上にあるセッションは、私と小田さんが参加したワーキングパーティとは別のワーキングパーティですが、クレーム・デベロップメントを、機械学習を用いて分析するという内容を研究したワーキングパーティで、参加者も多かったですし、議論も非常に活発に行われておりました。やはり人工知能、機械学習は、避けて通れない分野だということが ASTIN の中でも認識はございまして、ただ一方で、まだ明確にこうやって使っていこうなど、解があるわけではなく、手探りであっても研究を進めていこうという気概が伝わってきた、そのようなコロキアムでございました。

私と小田さんは、ASTIN 関連研究会から IAA の別のワーキングパーティに派遣されまして、プレディクティブモデリングに関するセッションを担当いたしました。

コロキアムの概要



9

(1-9)

このスライドは、コロキアムの会場であるシェラトングランドホテルと総合案内です。パナマ・アクチュアリー会の方が非常にいろいろな事務をこなしてくれまして、とても気持ちよく過ごすことができました。

ASTIN Big Data/Data Analytics Working Party

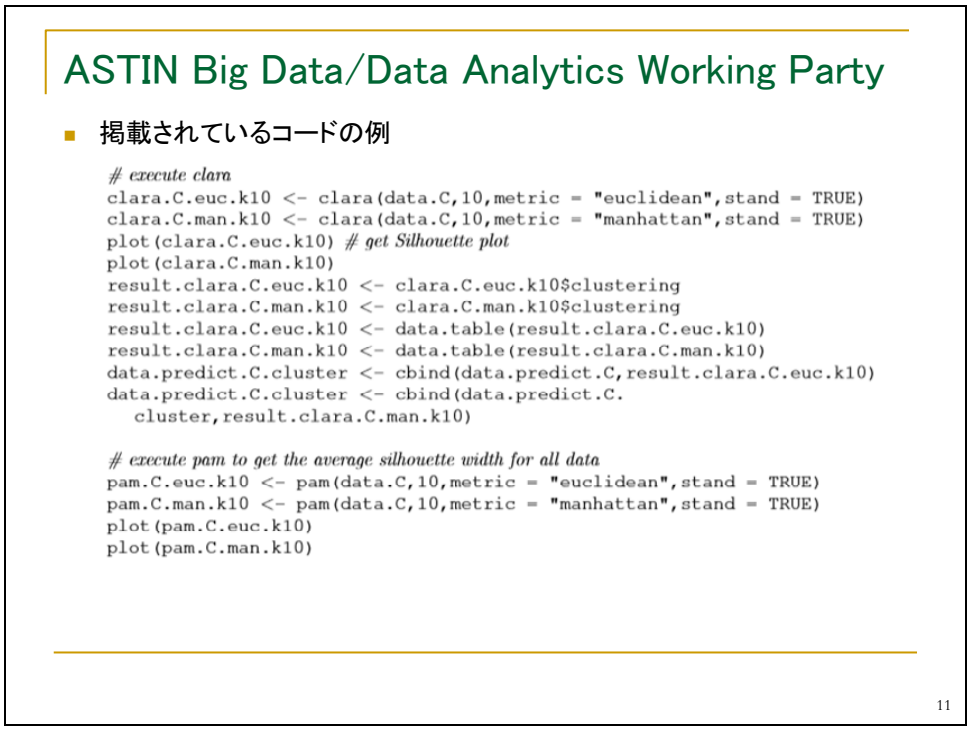
- ASTINでは、データサイエンスへの関心の高まりから、 ASTIN Big Data/Data Analytics Working Party を設置
- 2015年4月、ASTIN Big Data/Data Analytics Working Party はPhase 1 Paperを発表
  - ビッグデータの活用を巡るトレンドの概要の説明や、リファレンスの提供が主な目的
  - http://www.actuaries.org/ASTIN/Documents/ASTIN_Data_Analytics_Final_20150518.pdf
- 2017年8月に行ったコロキアムでの発表は、Phase 2にあたる
  - Phase 2 は「実際のデータへ各分析手法の適用」をテーマとする
  - 公開データを使用し、コードも提供することで、アクチュアリーが実際に様々な手法を体験できるようにすることが目的
  - 計算はR とExcel で実行し、実際に使用したプログラムも掲載

10

(1-10)

ワーキングパーティの概要です。私と小田さんの参加したのは「ASTIN Big Data/Data Analytics Working Party」というワーキングパーティです。データサイエンスへの関心の高まりから、元々「Big Data/Data Analytics Working Party」というワーキングパーティが設置されておりまして、2015 年 4 月に Phase 1 Paper が発表されております。スライドの URL から見られるのですが、Phase 1 Paper は、主にビッグデータの活

用をめぐるトレンドの概要や、リファレンス提供が主な目的になっております。今回のコロキアムで発表した Phase 2 Paper はどちらかというと、実際のデータに各分析手法を適用することをテーマとしております。2 つ大きなポイントがありまして、まず公開データを使うこと、これは、他のアクチュアリーの方も再現ができるように、一般に公開されているデータで分析をしようという意図でございます。それから、コードを提供すること。後でご覧になっていただきますが、プログラムコードを公開しています。このデータとプログラムコードを使えば、ご自身のコンピューターで、今回やった内容というものは誰でも再現できるところが、今回意図したところでございます。



(1-11)

このスライドは、私の担当した「clustering」の中のコードの例なのですけれども、このように 1 行 1 行書かれておりまして、小田さんの内容とあわせれば、データの取り込みからグラフの出力、最終結果のアウトプットまで、全部再現することができます。こういったものを公開できたということが、一つの成果だと認識しております。

ASTIN Big Data/Data Analytics Working Party

- Phase 2 Paper のトピック
  - 教師あり学習（supervised learning）
    - MARS (Multivariate Adaptive Regression Splines)
    - Trees
  - 教師なし学習（unsupervised learning）
    - clustering
    - PCA
- 「教師なし学習」について、遠藤と小田が担当

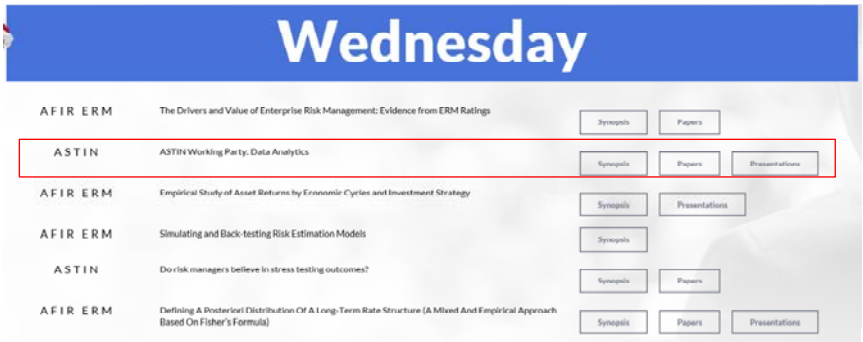
12

(1-12)

Phase 2 Paper のトピックですけれども、大きく教師あり学習、教師なし学習がございます。この分類は後で小田さんに説明いただくのですが、教師なし学習について２人で担当いたしました。私が「clustering」、小田さんが「PCA」を担当しています。

ASTIN Big Data/Data Analytics Working Party

- 以下のサイトにプレゼン資料とペーパーが掲載されている
  - http://www.actuaries.org/panama2017/papers-and-presentation.html
    (Wednesday, ASTIN Working Party: Data Analytics)



13

(1-13)

このスライドに記載のサイトに、プレゼン資料とペーパーが掲載されており、今申しあげたコードなどをご覧になることができます。

Phase 2 paper のサマリー

- PCAの手法を用いたロジスティック回帰が最も精度の高い予測となった（PCAのパートで手法等について詳述）
- AUC（Area Under the ROC Curve）の比較

Model	AUC
Logistic regression	0.719
MARS Combined model	0.716
Trees/ Random Forest	0.716
Deepnet	0.717
Logistic regression with PCA	0.745

14

(1-14)

このスライドが Phase 2 Paper のサマリーですが、実は、後ほど小田さんに発表いただく方法、「PCA」を用いたロジスティック回帰というものが、一番いい結果になりました。その辺りも、後半に聞いていただければと思います。

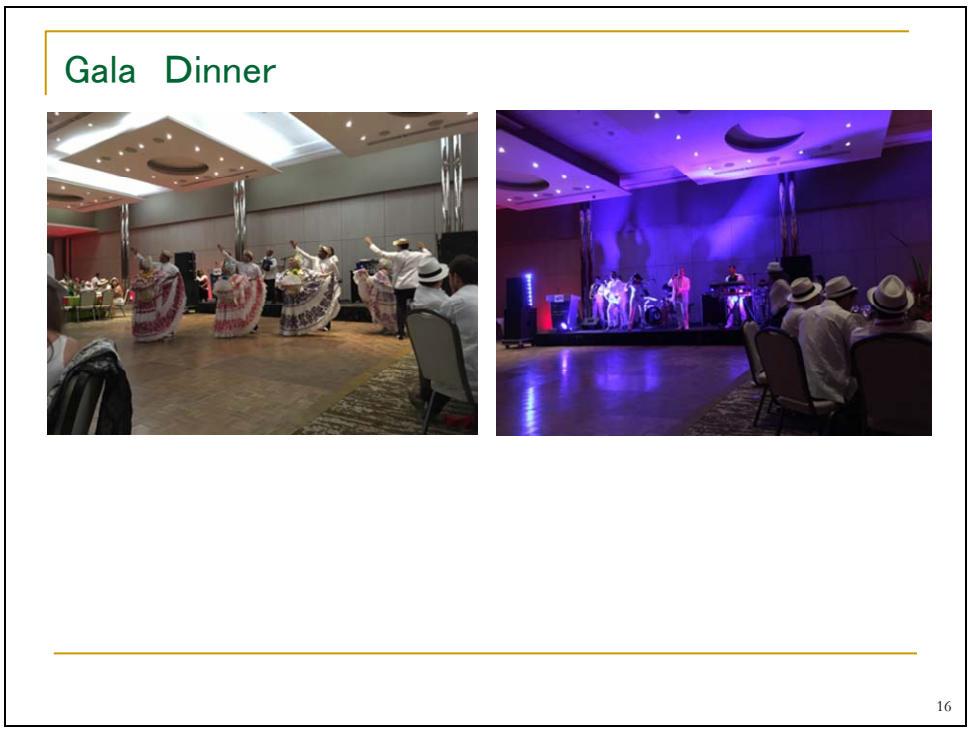
発表の様子とワーキングパーティのメンバー

15

(1-15)

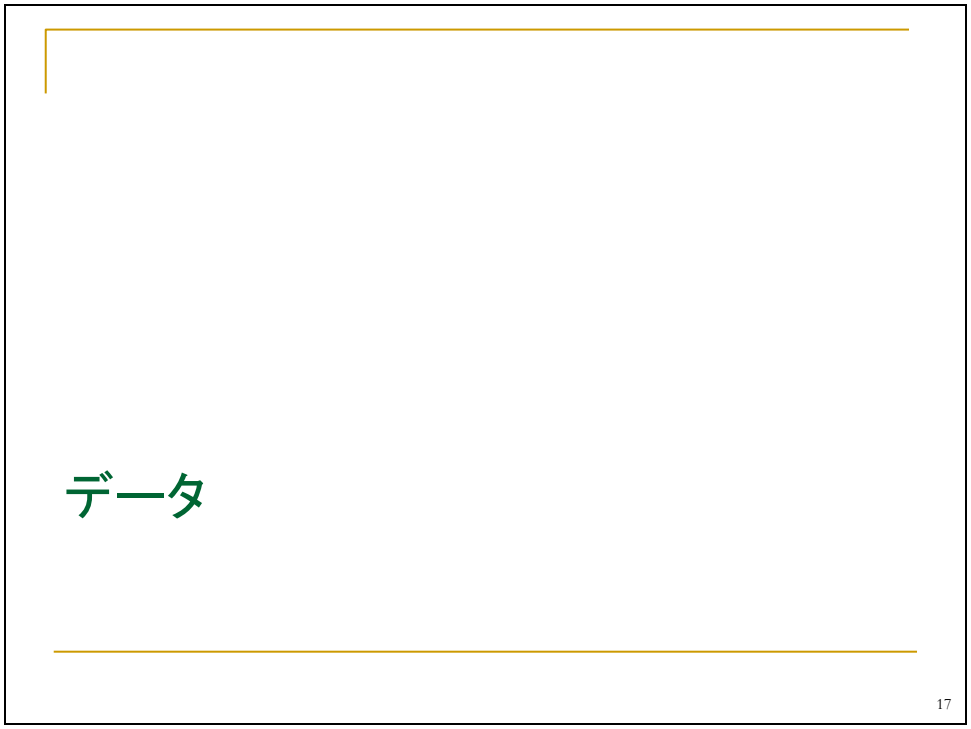
実際の発表の様子とワーキングパーティのメンバーです。ワーキングパーティでの発表ということもありまして、一番大きな部屋を使わせていただいて、参加者もたくさん集まっていただきました。プレディクティブモデリング自体が旬なトピックということもあったので、非常に注目していただけたことは、よかったと思います。右の写真は、メンバーです。女性の方がリーダーのルイーズ・フランシスという方で、

カナダでアクチュアリーの会社を自分でやっている方です。それから、ドイツのアクセルという方と、あとは IAA 会長のトム・テリー氏にも最後交ざっていただいて、記念撮影をした様子です。



(1-16)

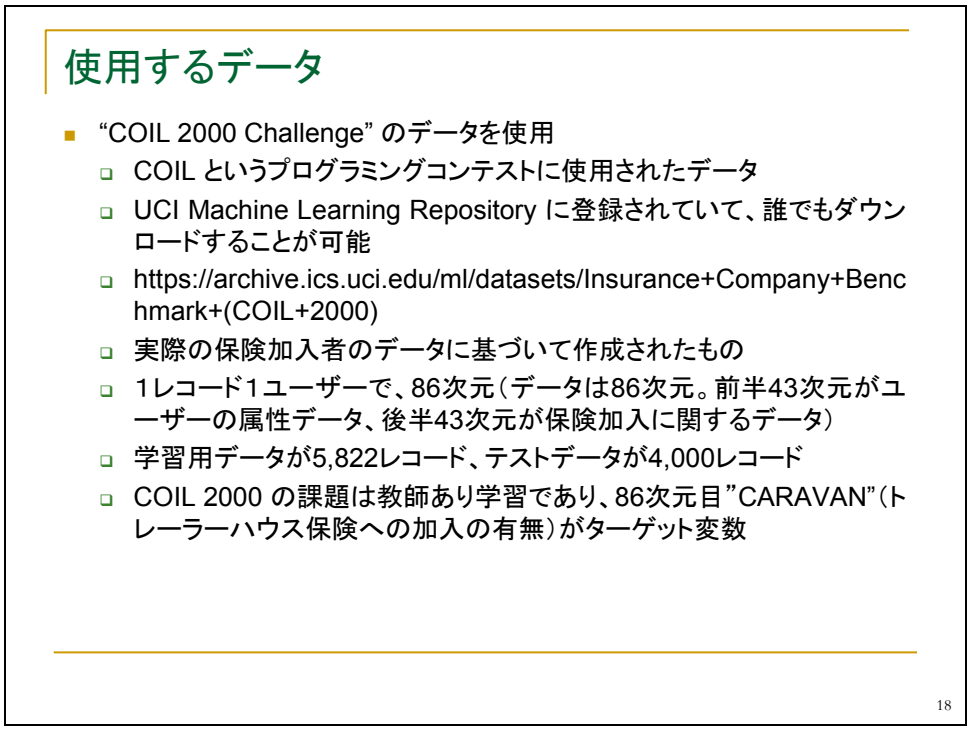
パナマ・アクチュアリー会は非常に歓迎してくれまして、このスライドは「Gala Dinner」の様子ですけれども、民族衣装のダンスやライブパフォーマンスをしていただいて、そのあとはみんなでダンスパーティをするなど、学術的な面以外でも、文化的な刺激にもなりましたし、非常に有意義な経験だったと思っております。以上、コロキアムの概要です。



(1-17)

このあと、2 人で使った共通のデータについてまずお話ししまして、そのあと分析の内容をお話ししよう

と思います。



(1-18)

データです。「COIL 2000 Challenge」というプログラミングコンテストに使ったデータを今回使用しています。「UCI Machine Learning Repository」というサイトに登録されておりまして、どなたでもダウンロード可能なデータです。オランダの保険会社の実際のデータに基づいて作られたもので、大体、全部で 10,000 レコードあります。学習用データが 6,000、テストデータが 4,000 ぐらいです。1 レコード、1 ユーザーを表していまして、86 次元のデータです。最初の 43 次元がユーザーの属性データで、後半 43 次元が保険加入に関するデータとなっております。元々は、この課題は教師あり学習で、86 次元目にトレーラーハウス保険への加入の有無を表す「CARAVAN」という変数がありまして、これがターゲット変数でした。学習用データで、トレーラーハウス保険にどのような人が入っているのかということをモデルに学習させて、テストデータで当てに行くことを競うというプログラミングコンテストでした。今回も、元のプログラミングコンテストに倣いまして、このトレーラーハウス保険への加入の有無「CARAVAN」を予測しに行くということを、各メンバーで、様々な手法でやってみたというところでございます。

使用するデータ

- 顧客の属性を表すデータ（前半43次元）
  - すべて”zip-code variable”（郵便番号の区分する地域ごとに家族構成、収入等の分布や平均を登録したデータ）
  - 主な情報は、家族構成、学歴、職業、収入など
  - 家族構成等について、zip code内の割合（分布）を表す変数が多い（TYPE=“PERCENTAGE”）
    - それぞれ0-10の値をとり、0: 0%, 1: 1-10%, 2: 11-23%, … , 8: 89- 99%, 9: 100% が割り当てられている。“GENRE”が同じ変数の合計は、おおむね“10”(100%)となる（次ページ参照）

19

(1-19)

まず、前半の 43 次元、顧客の属性を表すデータですが、すべて「zip-code variable」になっています。少し耳慣れないのですが、一人ひとりの実際の値ではなくて、その人の住んでいる郵便番号の区分する地域ごとに、家族構成、収入の分布や平均を登録したデータを使っております。主な情報としては、家族構成、学歴、職業、収入などがあります。

使用するデータ

- 顧客の属性を表すデータ（抜粋）

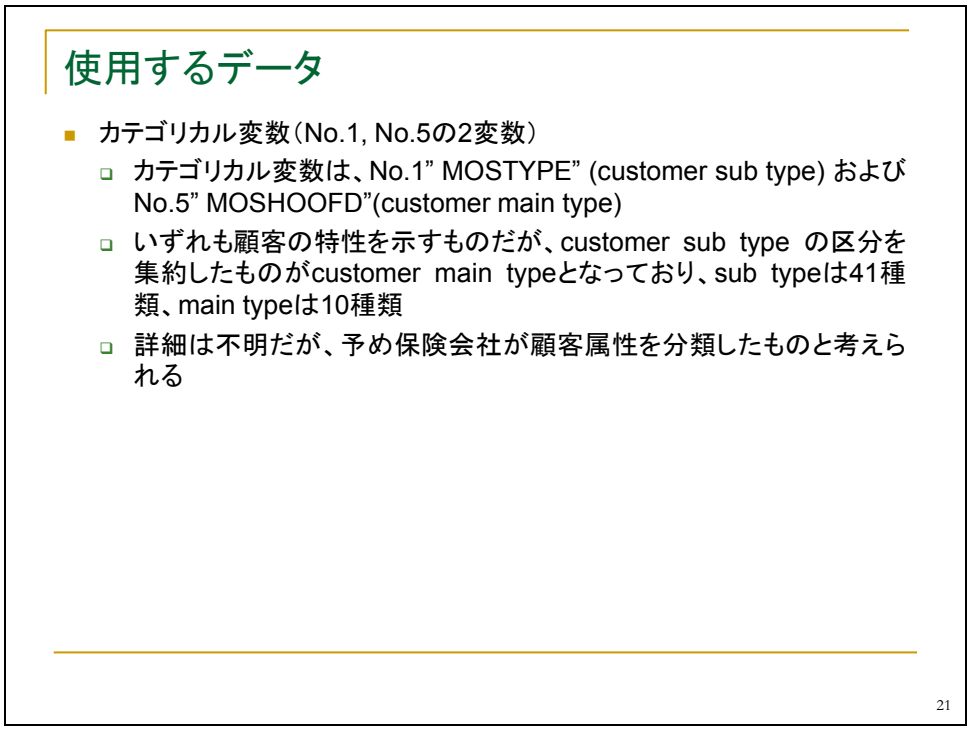
ID	NAME	TYPE	GENRE	DESCRIPTION
1	MOSTYPE	CATEGORICAL	-	Customer sub type
5	MOSHOOFD		-	Customer main type
10	MRELGE	PERCENTAGE	FAMILY_01	Married
11	MRELSA			Living together
12	MRELOV			Other relation
13	MFALLEEN		FAMILY_02	Singles
14	MFGEKIND			Household without children
15	MFWEKIND			Household with children
30	MHHUUR		HOUSE_03	Rented house
31	MHKOOP			Home owners
42	MINKGEM	ORDINAL	-	Average income

20

(1-20)

このスライドは、顧客の属性を表すデータの抜粋です。例えば、「HOUSE_03」というジャンルの変数なのですけれども、持ち家か、借りているかを表す 2 つの変数になっていまして、今言った「zip-code variable」になっており、両方足して 100%になるように、それぞれの郵便番号ごとに区分しています。ある郵便番号

なら「Home owners」が 9 割いる、別の郵便番号なら 5 割いるなど、そのようにデータは管理されています。その他にも、家族構成、結婚の有無、子供の有無、学歴などの分布を表す変数があります。それからこの 42 番目の変数は、平均収入を表すのですけれども、100 万円以下なら 1、100 万から 200 万なら 2 のように、順番が振ってある変数でございます。



(1-21)

スライドの 2 つのカテゴリカル変数は、少し特殊なものになります。

実はあらかじめ、詳細不明なのですけれども、保険会社が顧客属性を分類した変数が入っていました。それがこのカテゴリカル変数の 1 番、5 番なのですけれども、特にこの 5 番「customer main type」というものに、今後の分析では焦点を当てていきます。

使用するデータ

- カテゴリカル変数のまとめ

Customer Main Type	Customer Sub Type	レコード数(学習データ)
1: Successful Hedonists	1 ~ 5	552
2: Driven Growers	6 ~ 8	502
3: Average Family	9 ~ 13	886
4: Career Loners	15 ~ 19	52
5: Living Well	20 ~ 24	569
6: Cruising Seniors	25 ~ 28	205
7: Retired and Religious	29 ~ 32	550
8: Family with Grownups	33 ~ 37	1,563
9: Conservative Families	38 ~ 39	667
10: Farmers	40 ~ 41	276

22

(1-22)

どのようなものかというと、保険会社が予め分類した顧客のタイプになります。スライドのように、例えば 3 番は平均的な家族、7 番は退職後の信仰心に厚い人など、10 個のカテゴリーに分けていました。このカテゴリカル変数に対して、クラスタリングでは、これよりもよい分類を目指して分析を行いました。今の顧客分類よりも、より保険加入と結びつくような分類はないかという視点です。それから、この 10 個の分類は、何らかの視点に基づいて分類されたものですので、これを自分たちの分析結果のベンチマークにするといったこともしております。

使用するデータ

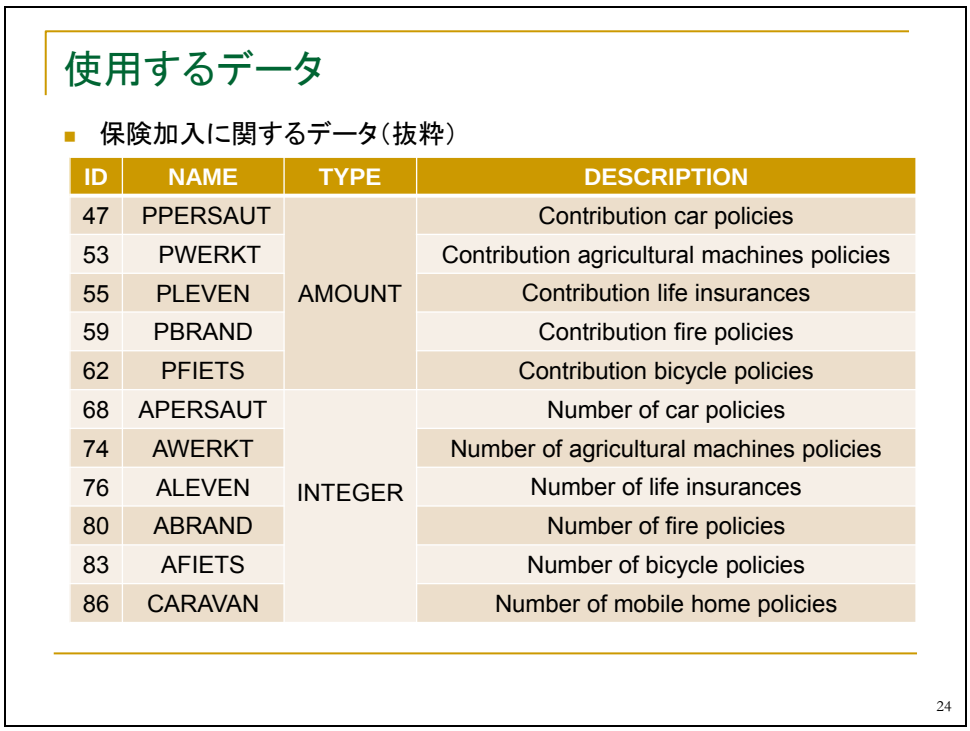
- 保険加入に関するデータ(後半43次元)
  - 火災保険、自動車保険、生命保険等への加入件数および支払保険料のデータ
  - トレーラー保険への加入有無を表す変数("CARAVAN")の予測が元のプログラミング・コンテストの内容

23

(1-23)

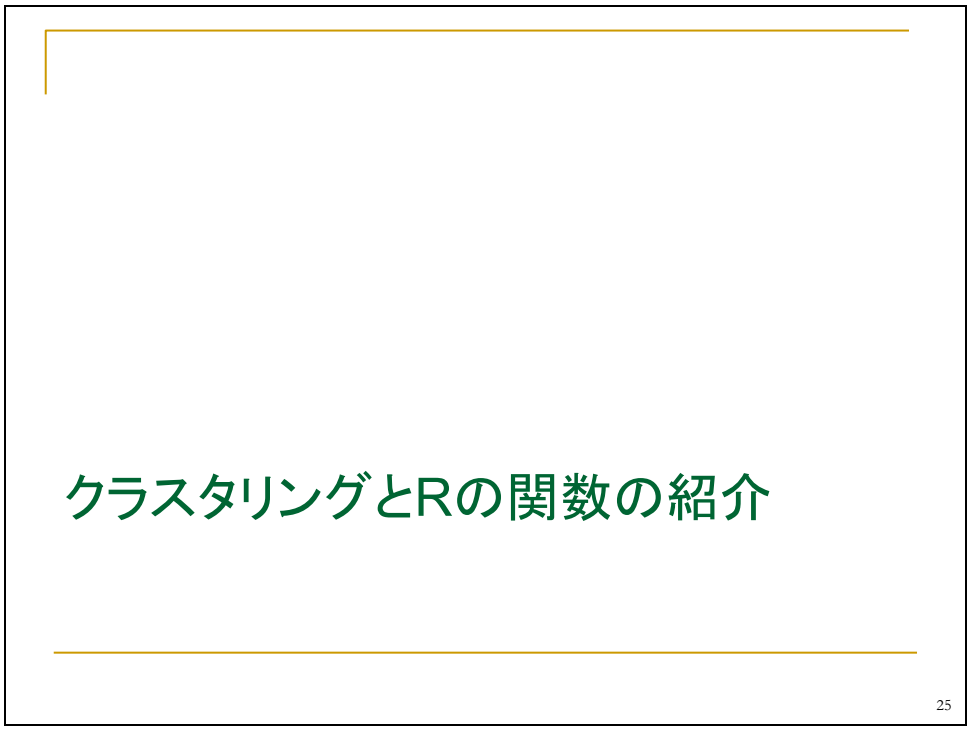
後半 43 次元が保険加入に関するデータなのですが、火災保険や生命保険への加入件数や支払保険料のデ

ータになっています。この「CARAVAN」という変数が、トレーラー保険への加入有無を表す変数で、元のプログラミングコンテストのターゲット変数でした。



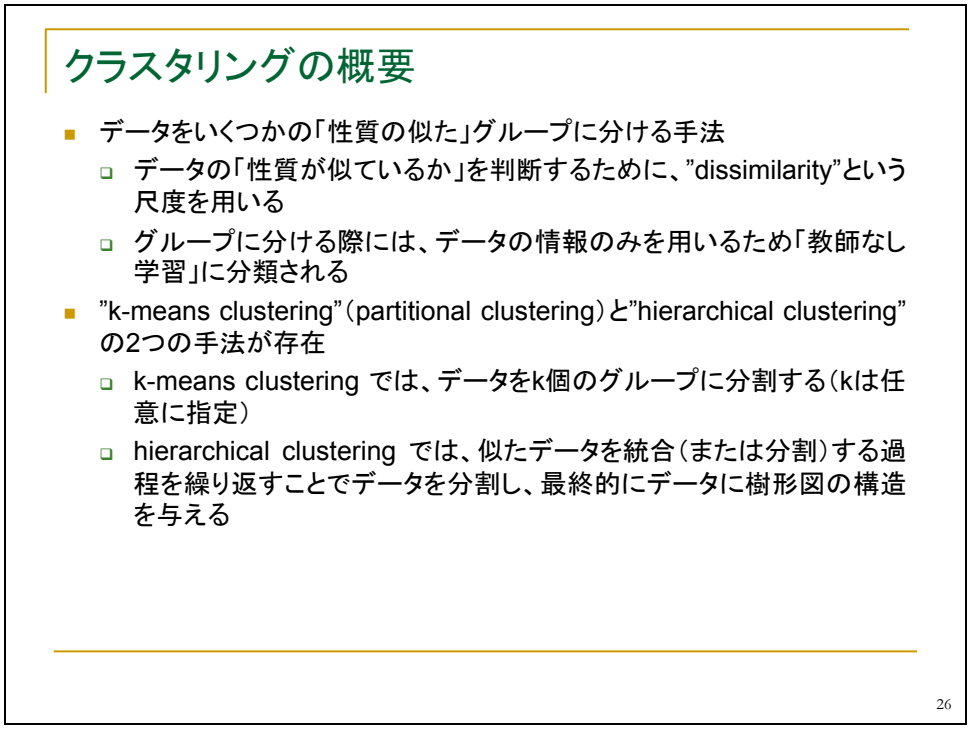
(1-24)

このスライドが抜粋です。自動車保険や生命保険などについて、保険料と加入件数を管理しております。
以上、駆け足でしたが、データのご紹介でした。



(1-25)

ここからは、私の行ったクラスタリングの説明に入りたいと思います。



(1-26)

クラスタリングは、おおまかに言うと、データを幾つかの性質の似たグループに分けるという手法です。この性質が似ているかを判断するために、2 つのデータ間について、「dissimilarity」という尺度を用います。グループに分ける際には、データの情報のみを用いるために、教師なし学習に分類されております。

クラスタリングは、大きく分けて 2 つの方法が存在していまして、「k-means clustering」と呼ばれる方法と、「hierarchical clustering」と呼ばれ、階層的クラスタリングと訳される方法です。k-means は、データを k 個のグループに 1 回で分けるという方法で、k は任意に指定する必要があります。一方、階層的クラスタリングは、似たデータを統合する過程や分割する過程を繰り返すことによって、最終的にデータに樹形図の構造を与えるもので、まさに階層的にクラスタリングを行っていくというものでございます。

dissimilarity

- クラスタリングにおいて、2つのデータがどの程度似ているかを示す尺度
- 以下の3つの性質を満たす
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
- 距離の性質に似ているが、三角不等式（$d(i,j) \leq d(i,k) + d(j,k)$ ）が含まれない点が異なる
- 逆に言うと、距離の性質を満たすものはdissimilarityとして用いることができる
  - Euclidean distance: $d(i,j) = \sqrt{(x_j - x_i)^2 - (y_j - y_i)^2}$
  - Manhattan distance: $d(i,j) = |x_j - x_i| + |y_j - y_i|$
- 各データ間のdissimilarityを行列にまとめたdissimilarity matrixがクラスタリングを行う際のインプットとなる（実際は、Rの関数内でデータからdissimilarity matrixを作成できるため、データのままインプット可能）

27

(1-27)

まず、尺度になる dissimilarity は、どのようなものかということなのですけれども、スライドにある 3 つの性質を満たしていまして、距離の性質に似ていますが、三角不等式が含まれないという点だけが異なります。逆に申し上げると、距離の性質を満たすものは、どれも dissimilarity として使うことはできますので、今回の分析では、普通のユークリッド距離と、あとマンハッタン距離という、スライドに定義された 2 つを使っております。

この各データ間の dissimilarity を行列にまとめたマトリックスが、クラスタリングを行う際のインプットになるのですけれども、実は、R の関数では、データを入れてしまうと、内部でこのマトリックスを作ってくれますので、データをインプットするだけでプログラムは実行することができます。

R cluster ライブラリー

- k-means clustering
  - pam
  - clara
- hierarchical clustering
  - agnes
  - diana

- それぞれの関数を実行すると、付随してクラスタリング結果の評価に関する情報も得ることができる (Silhouette value, AC/DCなど)
- 結果を分析するための有用な関数も存在（cutreeなど）

28

(1-28)

今回、R の cluster ライブラリーを使いました。これを使うと、本当に簡単にクラスタリングが実行できます。k-means では 2 つの関数、pam と clara を用いまして、階層的クラスタリングでは、agnes と diana を使いました。それぞれの関数を実行しますと、クラスタリングの結果だけでなくて、その結果を評価する尺度など、そのようなものも返してくれます。そういった意味でも、非常に使いやすい関数です。

k-means clustering : pam

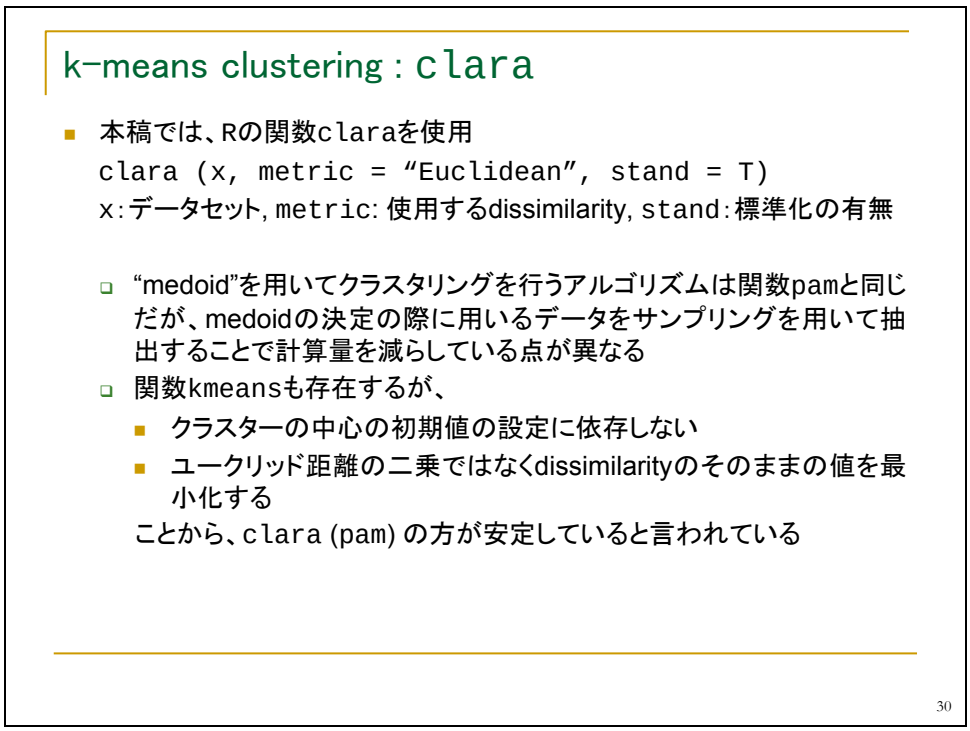
- pamの目的は、n個のデータからk個の代表点（medoid）を選び、データから最も近いmedoidとのdissimilarityの総和を最小とすること
- 言い換えると、次の目的関数を最小化するようなk個のmedoidを選ぶ
  - $\sum_{i=1}^{n} \min_{t=1,\cdots k} d(i,m_t), m_t : medoid \cdots (1)$
- 手法は以下のとおり
  - Step 1
    - medoid の初期値の設定
      - $m_1 : \sum_{i=1}^{n} d(i,m_1)$ を最小とするデータ
      - m_2 : (1) を最も減少させるようなデータ
      - m_k までこのプロセスを繰り返す
  - Step 2
    - (1) が減少するようなmedoid の交換を収束するまで繰り返す
    - 具体的には、$i \in \{m_1,\cdots,m_k\}, j \notin \{m_1,\cdots,m_k\}$ であるすべてのペア(i,j) に関して、(1) が最も減少する交換を行うことを繰り返す

29

(1-29)

まず、k-means クラスタリングのご紹介をするのですが、pam と clara、この 2 つはよく似たものになっています。pam の方で説明しますけれども、pam はデータを k 個のクラスターに分けるもので、データの中から k 個の代表点、medoid と呼ぶのですが、これを選び、各データから最も近い medoid との dissimilarity

の総和を最小とするということがアルゴリズムになっています。式にするとスライドにあるような感じです。アルゴリズムが書いてあるのですけれども、説明はスキップさせていただきます。



(1-30)

実際にペーパーで使ったものは clara という関数で、実は、この medoid という代表点を見つけてくるという点では、アルゴリズムは pam と同じなのですけれども、よりデータ量が大きい場合に適しているものとなっています。この代表点を決める際に、データのサンプリングを行うために計算量が非常に少なくなっています。逆に言うと、マシンパワーがあれば、pam を愚直に実行してもいいのですけれども、今回はデータが多かったので、clara を使っております。スライドに記載されているのが、clara を実行するコードです。非常に簡単で、データセットを指定して、あとは使用する dissimilarity として、ユークリッド距離やマンハッタン距離などを選び、標準化の有無を指定すると結果が出力されます。標準化は基本的にはすべきと言われております。

k-means clustering : Silhouette value

- クラスタリングの結果の評価は困難だが、一つの方法として、以下で定義されるSilhouette valueを用いるものがある

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} \in [-1,1]$$

$a(i)$：i番目のデータと、i番目のデータが属するクラスターの他のデータとのdissimilarityの平均値

$b(i)$：i番目のデータと、i番目のデータの2番目に近いクラスターの他のデータとのdissimilarityの平均値

  - $s(i)$が1に近いほどそのデータは現在のクラスターによく当てはまり、逆に-1に近いほど現在のクラスターによく当てはまっていないと解釈できる
  - clara を実行すると、各データに対するSilhouette value だけでなく、medoidの算出に用いたデータのSilhouette value の平均値も出力される

31

(1-31)

クラスタリングは、分けることは非常に簡単なのですけれども、分けた後に、よい分け方だったかを評価することは、非常に難しいです。一つの方法として、R の clara や pam を実行すると、「Silhouette value」というものが返ってきます。定義はスライドのようになっているのですけれども、簡単に言うと、1 に近いほどそのデータの現行のクラスターへの当てはまりがよくて、マイナス 1 に近いほど、現行のクラスターへの当てはまりが悪いというような尺度になっています。clara も pam も実行すると、全データに対するこの Silhouette value の平均値を返してくれますので、これが 1 に近ければ、クラスタリングはうまくいっている、逆に 0 などに近ければ、あまりうまくいっていないといった判断に使えるものになっております。

hierarchical clustering : 2つの手法

- hierarchical clustering には、大きく分けて以下の2つの手法がある
  - agglomerative (bottom-up) method （凝集型）
  - divisive (top-down) method （分割型）
- agglomerative method では、最初のクラスターは一つ一つのデータからなり、そこから最も近いクラスター同士を2つずつ結合していき、データ全体が一つのクラスターとなるまで結合を繰り返す
- divisive method はその逆で、最初のクラスターはデータ全体からなり、そこから分割を行い、一つ一つのデータがクラスターとなるまで分割を繰り返す
- agglomerative method については、関数agnes が、divisive method については関数diana が対応している

32

(1-32)

次に、階層的クラスタリングに入っていきます。大きく 2 つ方法がありまして、凝集型「agglomerative (bottom-up) method」と分割型「divisive (top-down) method」とあります。2 つとも、bottom-up、top-down という言葉が示すように、逆の発想でやっていまして、この agglomerative method では、最初は 1 個 1 個のデータをクラスターとみなして、そこから一番近いクラスター同士を 2 個ずつくっつけていきます。最後には、データ全体が 1 個のクラスターになるといった方法です。一方、この divisive method は逆で、まずデータ全体を 1 個のクラスターとみなして、それを 2 個ずつに分割するという作業を繰り返していきます。R の cluster ライブラリーは、両方に対応していまして、この agglomerative method（凝集型）では agnes という関数を使って、divisive method（分割型）では diana という関数を使っております。

hierarchical clustering : agnes

- agnesでは最も近い2つのクラスター同士を結合していくが、この近さの比較にdissimilarity を用いる
- クラスターにデータが複数存在する場合、クラスター間のdissimilarity をどのように測定するかを定義する必要がある
- agnes ではクラスター間のdissimilarity を測定する方法をmethod で指定する
  - single linkage: `single`
    - それぞれのクラスターに属するデータから、最も近くなるようにデータをそれぞれ1つずつ選んだときのdissimilarity （nearest neighbor clustering）
  - complete linkage: `complete`
    - それぞれのクラスターに属するデータから、最も遠くなるようにデータをそれぞれ1つずつ選んだときのdissimilarity （furthest neighbor clustering）
  - average linkage: `average`
    - それぞれのクラスターに属するデータ間のdissimilarity の平均値とする
  - Ward's method: `ward`
    - Ward's method では、それぞれのクラスター結合のステップにおいて、クラスター内のdissimilarity の総和の増加が最小となるようにクラスター結合を行う

33

(1-33)

この agnes なのですけれども、最も近い 2 つのクラスター同士を結合していくのですが、この近さの比較には、先ほど申し上げた dissimilarity を使います。ただ、元々これは 2 つのデータ間の尺度ですので、クラスター同士の dissimilarity というものをどう測定するか、定義する必要があります。R の cluster ライブラリーの agnes では、主に 4 つの方法が使えるようになっていますが、今回は、average linkage という手法と Ward's method という手法の 2 つを使いました。この手法に何を選ぶかで、結果がかなり変わってきてしまうということが、今回わかった一つのポイントかと思います。

hierarchical clustering : diana

- 分割していく方法には多くのアプローチが考えられるが、diana では"dissimilarity analysis"と呼ばれる以下の方法でクラスターを分割する
  - Step 1
    - 最大の半径(diameter)を持つクラスター(C)を選ぶ($diam(C) = \max_{i,j \in C} d(i,j)$)
  - Step 2
    - A = C, B = ∅ と置く
    - 以下の手順に沿って、Aからデータを1つBへと動かす
      - Aに属するデータiについて、Aに属するi以外のデータに対するdissimilarity の平均値$a(i)$を計算する
      - 最も$a(i)$の大きいデータ(mとする)をAからBへと動かす(A = A \ {m}, B = m)
  - Step 3
    - 以下の手順に沿って、他のデータをAからBへ動かす
      - Aに属するデータiに対して、 $a(i)$および、Bに属するすべてのデータに対するdissimilarity の平均値$d(i,B)$を計算する
      - 以下の条件を満たすデータhを選ぶ
        - $a(h) - d(h,B) = \max_{i \in A}(a(i) - d(i,B)$
      - $a(h) - d(h,B) > 0$ ⇒ hをAからBへ動かし、このStepを繰り返す

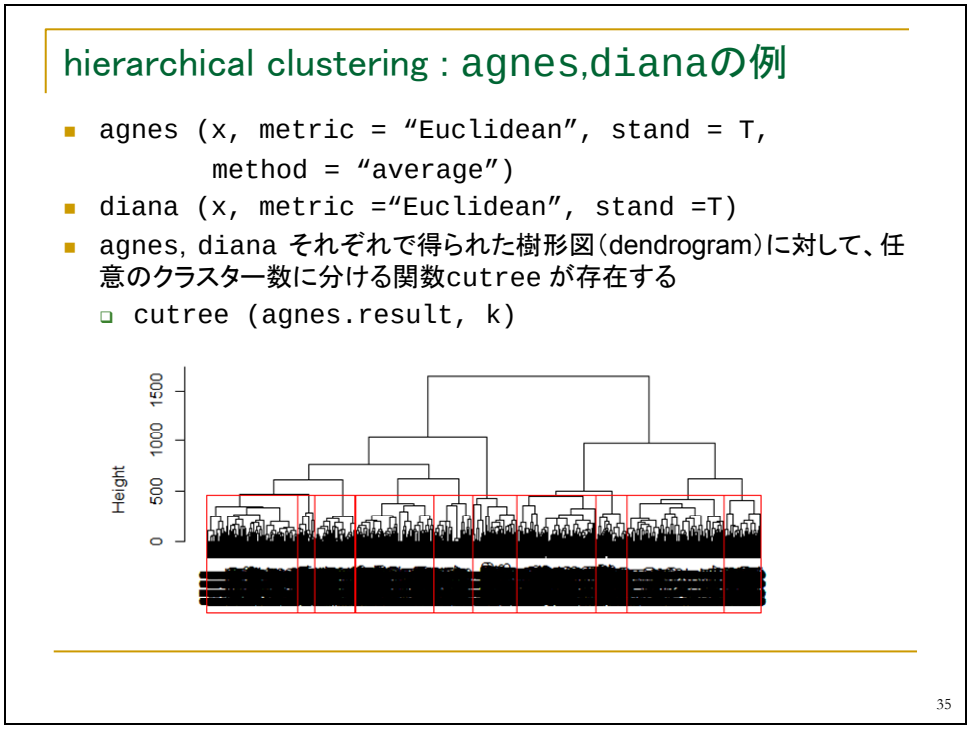
        $a(h) - d(h,B) \leq 0$ ⇒ このStepを止める

34

(1-34)

diana の方ですけれども、diana の手法は、cluster ライブラリーでは 1 つしか定義されていません。本

当は、いろいろなアプローチがあるのですけれども、R の cluster ライブラリーでは、こういった「dissimilarity analysis」と呼ばれているアルゴリズムで分割を行っています。データ全体を見て、どう 2 個に分けていくかを決めていくということです。



(1-35)

階層的クラスタリングの例なのですけれども、diana のコードは、clara と似ています。データセットと「metric」と標準化の有無を決めてあげれば、結果が出力されるというものです。一方、agnes であれば、先ほど申し上げたクラスター間の距離を測る方法としてに、「average」なのか、「ward」なのかといった、「method」も指定してあげる必要があります。

結果が樹形図で出力されると先ほど申し上げたのですが、このスライドのような形式で出力されます。まさに階層的にクラスターを分けていっているのですけれども、これだけだとどう解釈してよいのか非常に難しいので、データを任意のクラスター数に分ける関数 cutree というものが存在しています。例えばこのスライドでは k＝10 で行っているので、10 個のデータセットに分けてくれているわけです。ですから、階層的クラスタリングでも、k-means クラスタリングのような分析が可能になるということで、スライドのようになります。

hierarchical clustering : agglomerative coefficient

- k-means clustering におけるSilhouette value と同様、結果の評価を行う尺度が存在
- agglomerative coefficient（agnesを実行すると計算される）
  - $AC = \frac{1}{n}\sum_{i=1}^{n}(1 - \sigma(d(i))$
    - $\sigma(d(i))$:データi の属する1ステップ目に結合されたクラスターに対するdissimilarity を最後に 結合されたクラスター（データ全体）に対するdissimilarity で割ったもの
- divisive coefficient（dianaを実行すると計算される）
  - $DC = \frac{1}{n}\sum_{i=1}^{n}(1 - \sigma(d(i))$
    - $\sigma(d(i))$:データiの属する最後に分割されたクラスター（データ1つになる直前のクラスター）の半径（diameter）をデータ全体のデータセットの半径(diameter) で割ったもの
- それぞれ、1に近い方がクラスタリングに適した構造をデータが有しているとされるが、Silhouette value と異なり、データ数が大きくなるにつれて1に近づいていくことから、大きなデータでは尺度となりづらい

36

(1-36)

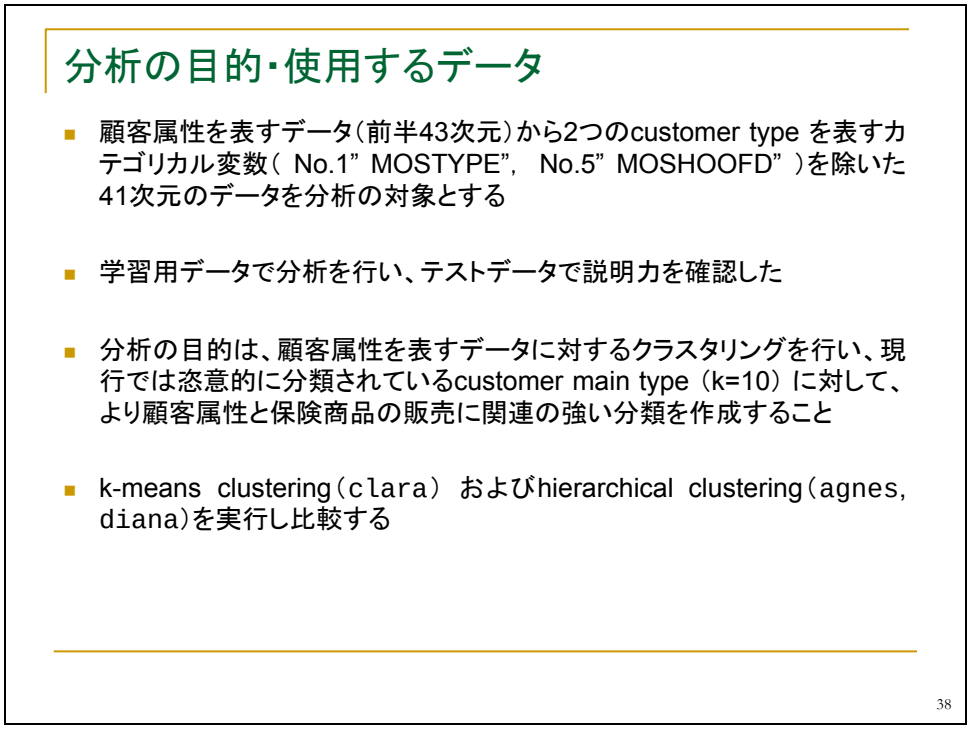
実は、この階層的クラスタリングにも、先ほど申し上げた Silhouette value のような結果を評価する尺度というものがあり、それぞれ 1 に近い方がいいと言われているのですが、データ数が大きくなると、定義上どうしても 1 に近づいてしまうという欠点を持っていまして、今回用いたデータセットなど、大きなデータでは尺度になりづらいという欠点がございます。

クラスタリングの結果

37

(1-37)

駆け足ですが、R のクラスターの理論的な部分をお話ししました。ここから実際の結果に入っていきたいと思います。



(1-38)

まず、分析の目的です。何をしたいかというと、今回は顧客属性を表すデータ 43 次元から、保険会社が予めに作成していたカテゴリカル変数（顧客の属性タイプ）2 次元を除いた 41 次元のデータを分析の対象にしています。学習用データで分析をして、テストデータで説明の妥当性を確認しました。実務に置き換えれば、既存のデータでモデルを作ってみて、新規のデータで当てはめてみて、妥当かどうか判断したということになります。

目的は、顧客属性を表すデータに対してクラスタリングを行って、恣意的に分類されている 10 個に分けられた「customer main type」という元々のカテゴリカル変数よりも、より顧客属性と保険商品の販売に関連の強い分類を作ることになります。保険商品としては、先ほど申し上げたトレーラーハウス保険への加入の有無「CARAVAN」をターゲットにしました。k-means では clara を使いまして、階層的クラスタリングでは、凝集型、分割型、どちらも実行しました。

主成分分析（PCA）の活用

- k-means clustering でも hierarchical clustering でも客観的な結果の評価は難しく、評価は主観的になってしまう
- それぞれのクラスタリングの結果の妥当性を検討するため、本稿ではPCAの結果を利用することとした
- 具体的には、クラスタリングの結果をPCでプロットすることにより、クラスターの分かれ方の妥当性を検討する

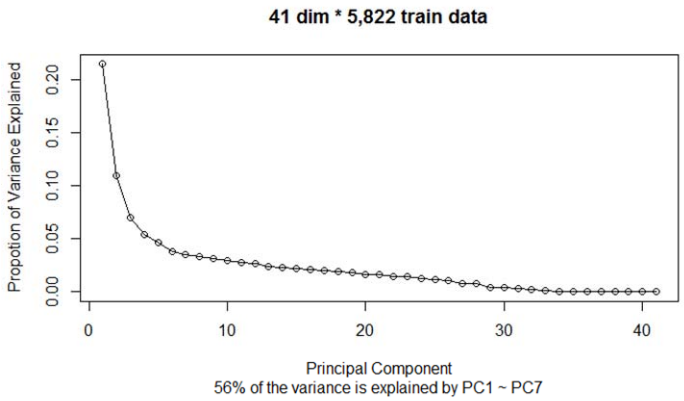
39

(1-39)

先ほど申し上げたのですけれども、クラスターを分けたはいいけれども、結果をどう評価するかは主観的になってしまいます。ここではその妥当性を検討するために、主成分分析（PCA）の結果を利用します。PCA について、詳しくは後半、小田さんに説明してもらうのですけれども、具体的には、クラスタリングの結果を主成分（PC）でプロットすることによって、クラスターの分かれ方の妥当性を検討しております。

PCAの結果

- 41次元のデータでPCAを実行した際のscree plot
- 第2主成分までに着目して以下では分析を行う

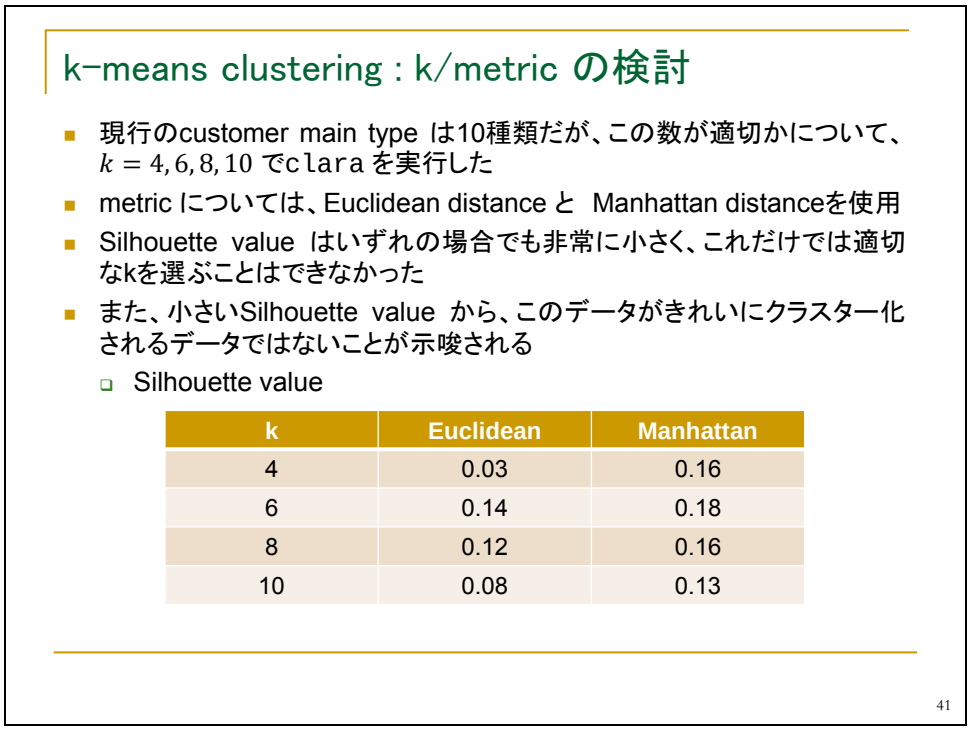


40

(1-40)

まず、PCA の結果です。「scree plot」といって、どの主成分がどのぐらい説明力があるかを示したものなのですけれども、第 1 主成分で、20%強、第 2 主成分が、10%強程度を説明している状況です。いろいろ試したのですが、結局、第 2 主成分までが一番よかったので、第 2 主成分までに着目して、分析を行いま

す。



(1-41)

まず、k-means クラスタリングを行いました。現行の customer main type は 10 に分かれているのですけれども、この数は適切かということを調べるために、k = 4、6、8、10 で R の関数を実行しております。metric は、どちらがいいか分からないので、両方試しました。ユークリッド距離とマンハッタン距離です。まず、Silhouette value で評価しようとしたのですけれども、どれも非常に小さく、これだけで適切な k はどれかということを選ぶことはできませんでした。Silhouette value が 0.2 以下ですと、最初に Silhouette value を提案した論文では、ほとんどクラスタリングに向いていないデータだと書かれています。ただ逆に、実務上、きれいにクラスターに分かれてくれるデータはほとんどないことが想定されまして、この小さい Silhouette value を見て、クラスタリングに向いていないのだと判断するのではなくて、このデータは、元々きれいにクラスター化できるようなデータではないことが示唆されていると考え、その先の分析を行いました。それでも、クラスタリングによって何か知見が得られないかということを探っていこうという姿勢です。

k-means clustering : PCによるプロット

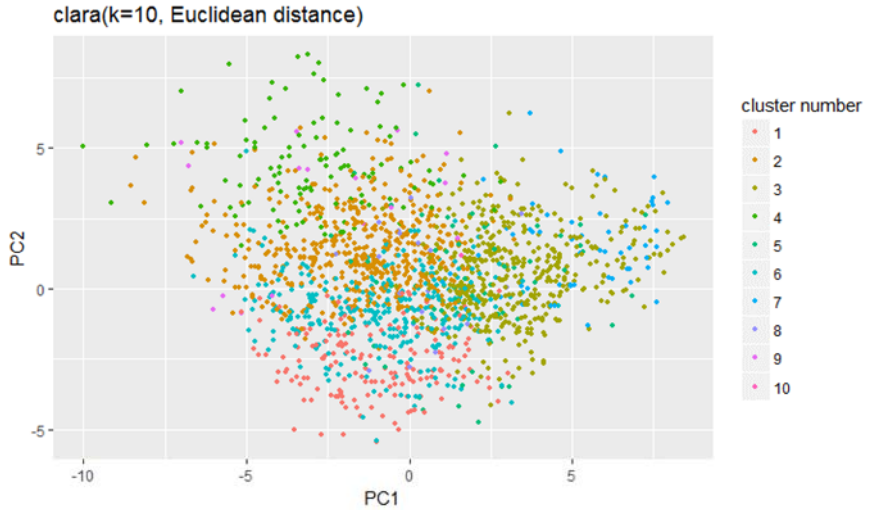
- k=10 の場合、Euclidean distance を用いるか Manhattan distance を用いるかでSilhouette value に0.05の差が生じている（0.08と0.13）
- この差の意義について評価することは難しいが、PCによるプロットによって、視覚的な検討が可能となる
- プロットを見ると、どちらもPCに沿った分割はできていないが、例えば、PC1に対するクラスターの分かれ方に着目すると、Manhattan distance を用いた方がPCに沿った分割ができているように見える（Manhattan distance を用いた場合のクラスター2とクラスター4など）

42

(1-42)

次に PC によるプロットを行います。k＝10 のときに、ユークリッド距離だと、Silhouette value は 0.08 で、マンハッタン距離だと、Silhouette value は 0.13 になっていまして、ではマンハッタン距離の方がいいのかというと、この 0.05 の差が有意かということは、これだけではよく分かりません。ここの差が有意かということを評価するために、PC によるプロットを使って、視覚的な検討を行いました。

k-means clustering : PCによるプロット

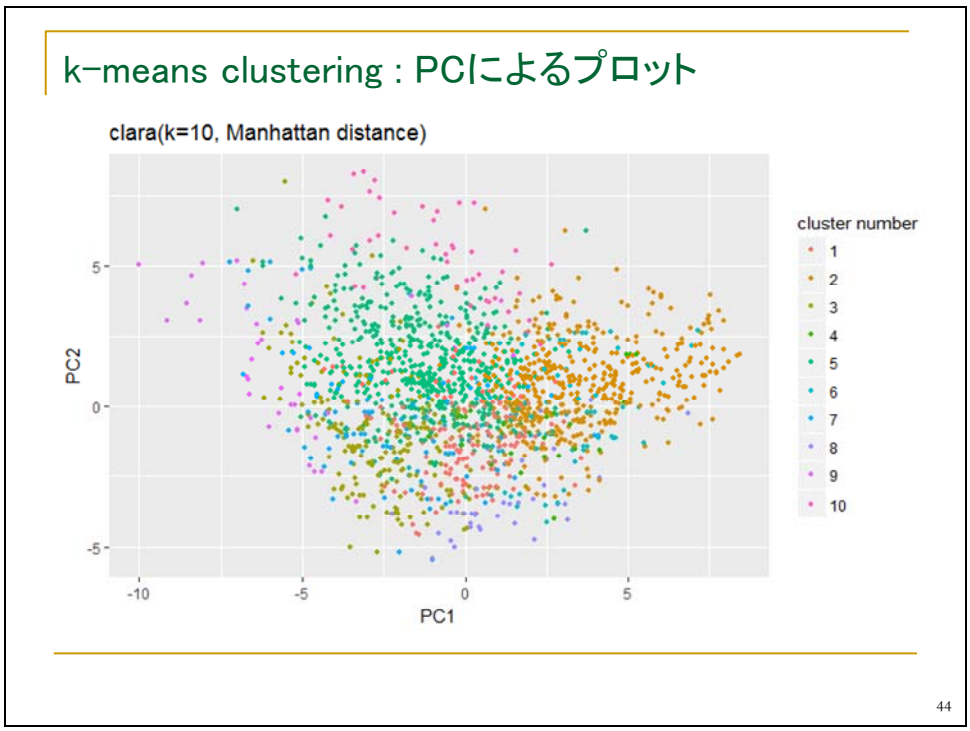


43

(1-43)

このスライドが、clara を k＝10 で、ユークリッド距離で行った際のクラスタリングの結果になります。1 から 10 までのクラスターに分かれていまして、それぞれに色が対応しています。横軸が PC1（主成分 1）で、縦軸が PC2 です。あまりきれいに分かれていないということが、見た感じで分かると思います。これ

は、Silhouette value が 0.08 ですので、このぐらいぐちゃぐちゃでも、そういったものかなというように、見た目でも、Silhouette value の評価ができるということです。



(1-44)

ではマンハッタン距離はどうか。マンハッタン距離では、Silhouette value が 0.08 から 0.13 に上がっています。見ると、若干ですけれども、この緑色の点と黄土色の点が、きれいに PC1 に沿って分かれているのではないかと思うことができます。主成分というものは、後で触れますけれども、データの特徴量を抜いてきたものですので、これに沿ってきれいに分かれていれば、クラスタリングの結果も妥当だろうという推論も立つのです。そういった意味では、このマンハッタン距離の方が、PC1 について一定程度きれいに分かれているので、ユークリッド距離と比べれば、マンハッタン距離の方がいいだろうといったことが、視覚的に検討できるということでございます。

hierarchical clustering : 実行した手法

- 以下の6つの手法を実行した
- いずれもAC(DC)はほとんど1に近く、それだけでは手法の比較は困難

function	metric	method	AC/DC
agnes	Euclidean	average	0.978
		Ward	0.997
	Manhattan	average	0.970
		Ward	0.998
diana	Euclidean		0.983
	Manhattan		0.974

45

(1-45)

k-means はあまり芳しくなかったので、階層的クラスタリングも実行しました。以下の 6 個の手法を実行しております。ただどちらも、評価できる指標の AC/DC がほぼ 1 に近いので、これだけでは、どれが一番いいかは分からないという状態です。

hierarchical clustering : 樹形図(dendrogram)

- cutree(*.result, 10) を実行し、それぞれのデータを10個のクラスターに分割した
- 選んだ関数(agnes, diana)、metric(Euclidean, Manhattan)、agnes内のmethod(average, Ward)によって結果は大きく異なった
- 例えば、agnes(method ="average")については、データがほとんど1つのクラスターに集まってしまった

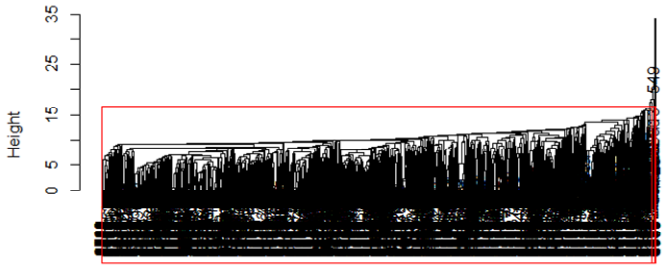


Figure.6: agnes(average linkage, Euclidean distance, k=10)

46

(1-46)

それぞれのクラスタリングについて、先ほど申し上げた cutree を使いまして、同じように 10 個にクラスターを分けてみました。ポイントは、凝集型か分割型か、それからユークリッド距離かマンハッタン距離か、あとは凝集型の中でも method をどうするかで、全く結果が変わってしまうという点です。クラスタリングはあまり頑健ではない手法だということが分かります。例えば凝集型で、method を average にして

クラスタリングを行いますと、10 個に分けたのですが、ほとんどのデータが 1 個に集まってしまって、使いものにならない結果になってしまいました。

hierarchical clustering : 樹形図（dendrogram）

- その他では、一定の大きさのクラスターにデータが分割されている
- その分かれ方は手法によって大きく異なり、樹形図のみで分割の適切性を評価することは難しい

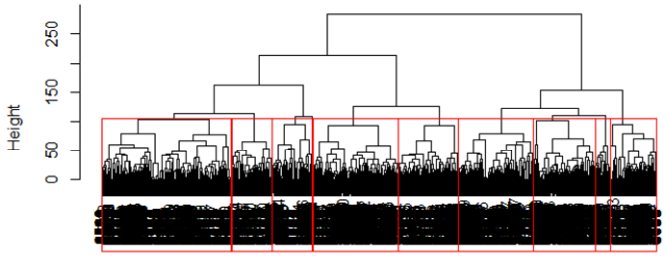


Figure.8: agnes(Ward’s method, Euclidean distance, k=10)

47

(1-47)

その他の手法では、一定の大きさにクラスターが分かれています。ただ、この分かれ方も手法によって大きく異なっていまして、この樹形図を見るだけで、何が適切かということは、なかなか言い難いものがあります。

hierarchical clustering : 樹形図（dendrogram）

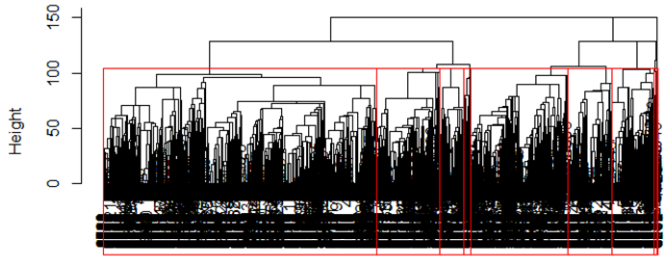


Figure.11: diana(Manhattan distance, k=10)

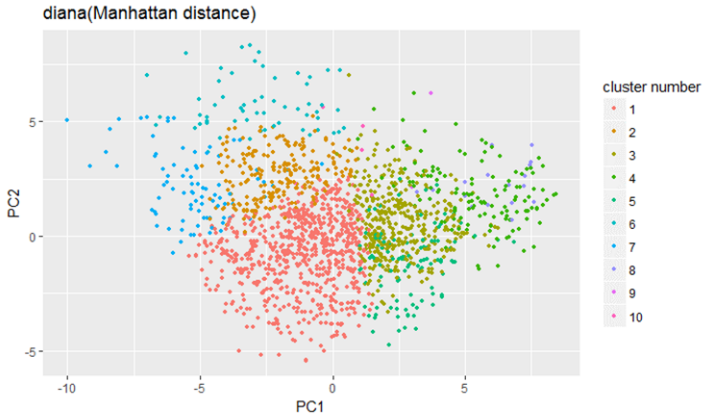
48

(1-48)

例えば diana を使うと、このような感じです。

hierarchical clustering : PCによるプロット

- PCでクラスタリング結果をプロットして検討を行った
- その結果、Manhattan distance を用いたdiana の結果が最もPCAとの整合性の高いため、このクラスタリングを採用して分析を進めていく



49

(1-49)

ここでも、k-means と同様に PC を使って、クラスタリング結果をプロットしました。その結果、スライドのマンハッタン距離を用いた diana（分割型クラスタリング）が視覚的に見て、一番 PCA と整合性が高かったので、このクラスタリングを採用して、分析を進めることにいたしました。先ほど見ていただいた図に比べると、かなり主成分 1 や主成分 2 に沿って、クラスターが分かれているということが見ていただけると思います。ですから、主成分という観点で見たときに、データの特徴をうまく捉えたクラスタリングをしているというように判断をして、これを基にして、この先の分析を行っていきます。

採用したクラスタリングの結果

- 現行の”customer main type”の数は10→6と集約される
- CARAVANはクラスター4およびクラスター5で高い値を示している

cluster	number of records	CARAVAN (average)	PC1 (average)	PC2 (average)
1	2,863	0.05	-1.22	-1.10
2	668	0.03	-2.22	2.57
3	1,022	0.08	2.58	0.27
4	446	0.12	5.41	1.69
5	464	0.11	2.79	-1.25
6	71	0.00	-2.84	5.64
7	249	0.03	-5.43	2.01
8	27	0.00	6.24	2.12
9	2	0.00	3.34	4.17
10	10	0.00	0.91	2.91
Total	5,822	0.06	0.00	0.00

50

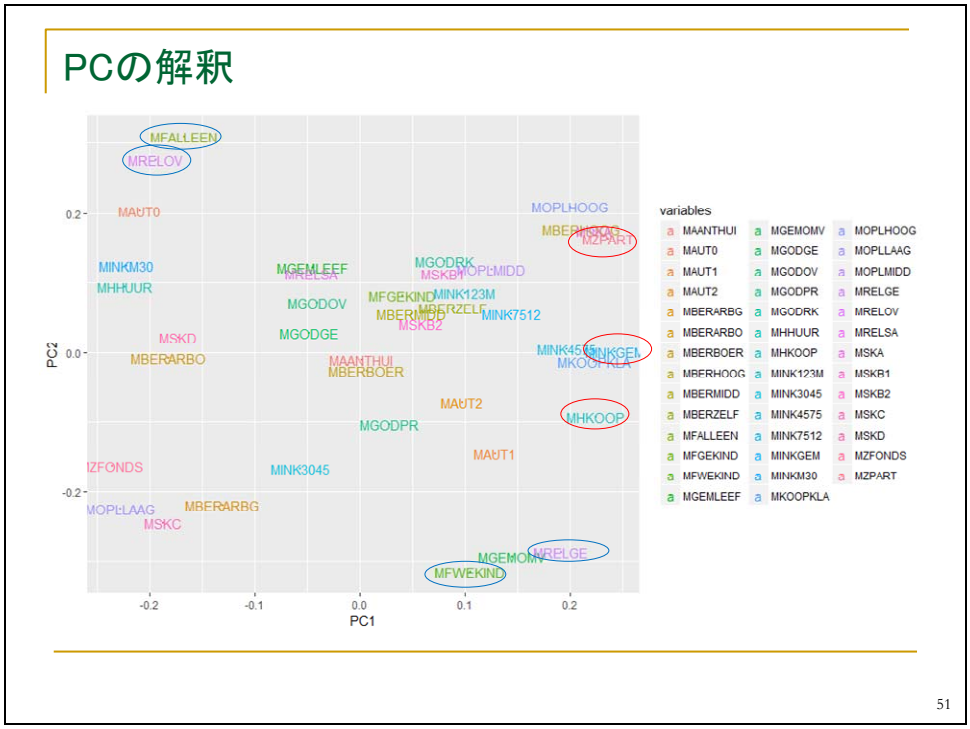
(1-50)

今のマンハッタン距離を用いた分割型クラスタリングの結果ですけれども、このようなサマリーとなっ

ております。まず、クラスターに属するレコードの数から、現行の 10 というものは多すぎると判断をしました。丸で囲った 6 個、一定のレコード数のあるクラスターに絞ります。

それから、今回注目しているトレーラーハウス保険への加入率を見ますと、平均 0.06%なのですが、このクラスター4 とクラスター5 では、倍に近い加入率になっています。テストデータを持ってきたときに、つまり新規のお客さんのデータが入ってきたときに、クラスタリングをして、4 や 5 に入ったら強く勧めればいいという結果がこのクラスタリングだけから言えるのですけれども、実務を考えれば、こういった機械的な説明ではなかなか難しくて、やはり、ではこの人たちはどのようなお客さんなのか、なぜ加入率が高いのかなどといった説明が求められるということが、実際のアクチュアリー業務では発生すると思います。ですから、ここでは主成分に注目をして、そういった解釈、説明にトライいたしました。

PC に注目しますと、クラスター4 は、PC1 はとても高くなっています。5.41 あります。平均は 0 になるはずの指標ですので、非常に高い PC1 を持っている。一方、クラスター5 は、そこそこ高い PC1 に加えて、マイナスの PC2 を持っている。この PC から、解釈、説明を作っていこうということが、このあとの分析です。



(1-51)

PC の解釈なのですけれども、それぞれの PC の寄与度を変数ごとにプロットしたものになります。これも R を使うと簡単に作れまして、そのコードも公開していますので、こういったものが簡単に作れます。コントリビューション・プロットと呼ばれるのですが、例えば PC1 で見ると、読みづらいのですけれども、「MINKGEM」という、前半に出てきた平均収入を表す変数というものが 0.2 強のプラスで、一番プラスに働いているということが読めます。平均収入が高いほど、PC1 にプラスの影響を与えるということが読めるのです。同じように、「MHKOOP」や「MZPART」も、PC1 にプラスの影響を与えるということが分かります。このように、PC1、PC2 で解釈してみました。

PCの解釈、クラスターの解釈

- PC1：MINKGEM(average income)，MHKOOP(home owners)，
  MZPART(private insurance)…(+)
  ⇒一般的な高収入の層で高くなるPC
- PC2：MRELOV(other relation), MFALLEEN (singles)…(+)
  MRELGE(married), MFWEKIND (household with children)…(-)
  ⇒単身や結婚していない層で高くなるPC

- クラスター4（PC1: 5.41, PC2: 1.69）
  ⇒高収入層で、結婚していない顧客
- クラスター5（PC1: 2.79, PC2: -1.25）
  ⇒一定の収入を持ち、結婚しているまたは子どものいる顧客

- トレーラー保険は、収入が高いほど、また、結婚していたり子どもがいる者の方が購入する可能性が高いという定性的な仮定が生まれる

52

(1-52)

その結果が、このスライドです。PC1 ですと、平均収入や持ち家比率が高ければ上がります。あとは、「private insurance」というものは、公的な保険に加えて、私的な保険に入っているかを表す指標で、これも高いほどプラスに寄与します。どれも、一般的に高収入の層で高くなる主成分だと解釈ができます。

一方 PC2 は、家族構成に非常に影響を受けていました。結婚していなかったり、単身家庭であったりすると、プラスになりまして、逆に結婚していたり、子どもがいたりすると、マイナスになるというような PC です。ですから、単身や結婚していない層で高くなる PC というように解釈できます。そうすると、先ほどのクラスター4、クラスター5 というものは、PC の数字を当てはめてみると、例えばクラスター4 は、高収入層で結婚していない人たち、クラスター5 は、一定の収入があって、結婚している、子どもがいる人たちというように解釈ができます。そうすると、トレーラー保険というものは、収入が高いほど、また、結婚していたり子供がいたりすると、購入する可能性が高いのではないかという定性的な仮定にたどり着くことができます。

これは、直感的にトレーラーハウスを持っているような人の特徴に合致するものであり、先ほどのクラスタリングの結果も妥当なのではないかと推論できるということです。

テストデータによる確認

- 採用したクラスタリングの結果の妥当性を、テストデータを用いて確認することとした
- クラスターの数
  - 一定以上のデータ数を持つクラスター(1, 2, 3, 4, 5, 7)を採用し、6個とした
- テストデータの分割
  - 以下のように簡潔に行った
    - 学習用データにおける、6個の各クラスターの平均値からなる点をmedoid の初期値とする
    - テストデータについて、6 個の medoid に対する Manhattan distance を計算し、最も近いmedoid のクラスターに属するとした
- 学習用データにおけるクラスターごとのCARAVAN の平均値を確認した

53

(1-53)

この分析の結果の妥当性を、テストデータで確認をしました。先ほどの 6 個のクラスターを採用します。以下のように、簡単にテストデータをクラスターに分割しました。まず、学習用データのそれぞれのクラスターの平均値を、各クラスターの中心点としまして、テストデータについて、その 6 個の中心点に対するマンハッタン距離を計算して、各データは一番近いクラスターに属するものとしました。

テストデータによる確認

- CARAVAN の値が学習用データで高かったクラスター4，5については、同じ傾向をテストデータでも示していた

cluster	number of records	CARAVAN (average)
1	1,519	0.05
2	587	0.04
3	713	0.06
4	384	0.10
5	470	0.11
7	327	0.04
Total	4,000	0.06

54

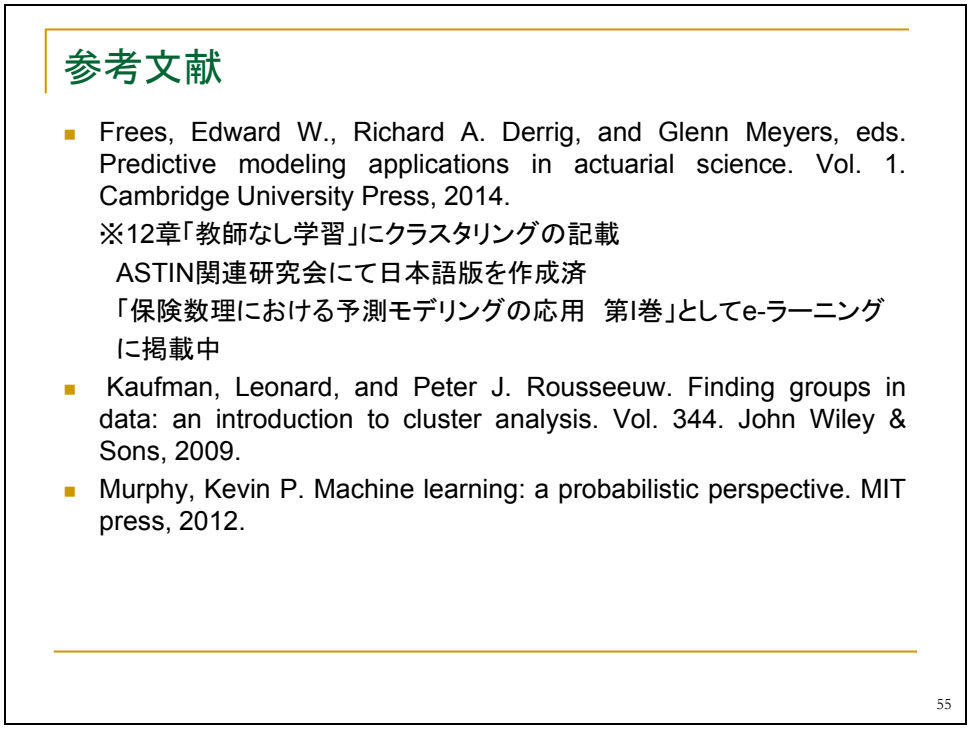
(1-54)

結果は、このスライドです。同様に、クラスター4、クラスター5 で CARAVAN の平均値が高くなりましたので、今回の分析の結果は一定妥当であるということが、テストデータでも確認ができました。

クラスター分析は、プレディクティブモデリングの一分野であり、機械学習の一つとして数えられるこ

ともありますが、今回はこれを用いて、「どういった人にどのような保険を勧めるべきか」という分析を行いました。同じように、例えば「どういった人が疾病にかかりやすいか」、「どのような車が事故を起こしやすいか」など、いろいろな分野に応用できると思います。

ここからは私見ですけれども、実際にやってみると計算自体は簡単なのです。R のコードを 1 行書きますとクラスタリングできます。データをインプットする、R をインストールするなど一手間はありますけれども、私は今回、これを機に R に触ってみたのですが、それほど大変ではありませんでした。ただ、やはりポイントは解釈や説明のところで、PC から何を読み解くのか、どういったクラスターに何が売れていると説明するのかなど、そのようなところの方が苦労しました。ここは、やはり通常のアクチュアリー業務と変わらないのかと思います。ですから、何百万次元のようなデータを用いるのではなくて、低次元のデータを用いるのであれば、これまでのアクチュアリーがしてきたことと今回行った分析は、それほど変わらないのではないかということが感想です。アクチュアリーがこれまでやってきたことが、プレディクティブモデリングや機械学習で大きく変わるのではなくて、使用するツールの幅が少し広がるというように認識して、プレディクティブモデリングや機械学習に取り組んでいけばよいのではないかと感じました。



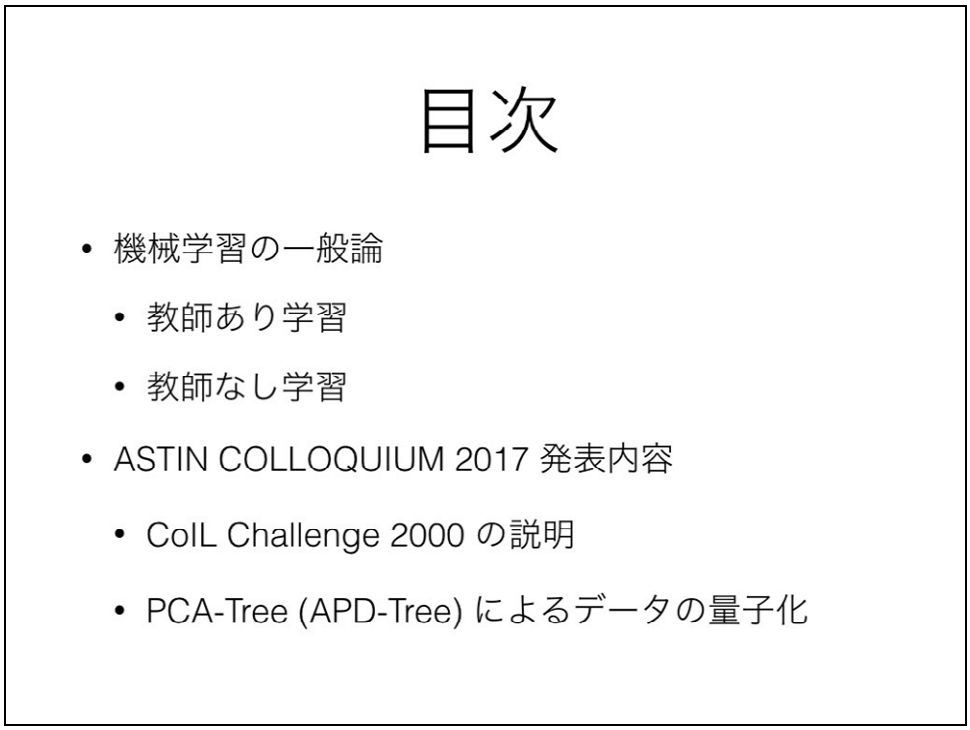
(1-55)

最後に参考文献なのですけれども、一番上の本「Predictive modeling applications in actuarial science. Vol. 1」の 12 章に教師なし学習が入っていまして、PCA やクラスタリングの記載があるのですが、ASTIN 関連研究会で翻訳を作成済みで、「保険数理における予測モデリングの応用　第Ⅰ巻」として、アクチュアリー会の e-ラーニングシステムに掲載されています。これを読んでいただければ、クラスタリングや PCA の内容、概要がつかめます。また今回分析に用いたデータも先ほど照会したスライドの箇所からダウンロードできますし、コードも全て公開しています。今回の発表をきっかけにして、実務のデータに対してクラスタリングにトライしてみようかと、少しでも思っていただければ非常に幸いです。ありがとうございました。

【司会】　遠藤さん、ありがとうございました。質問などは、最後にまとめて受け付けたいと思います。

引き続き、スマートニュースの小田さん、よろしくお願いします。

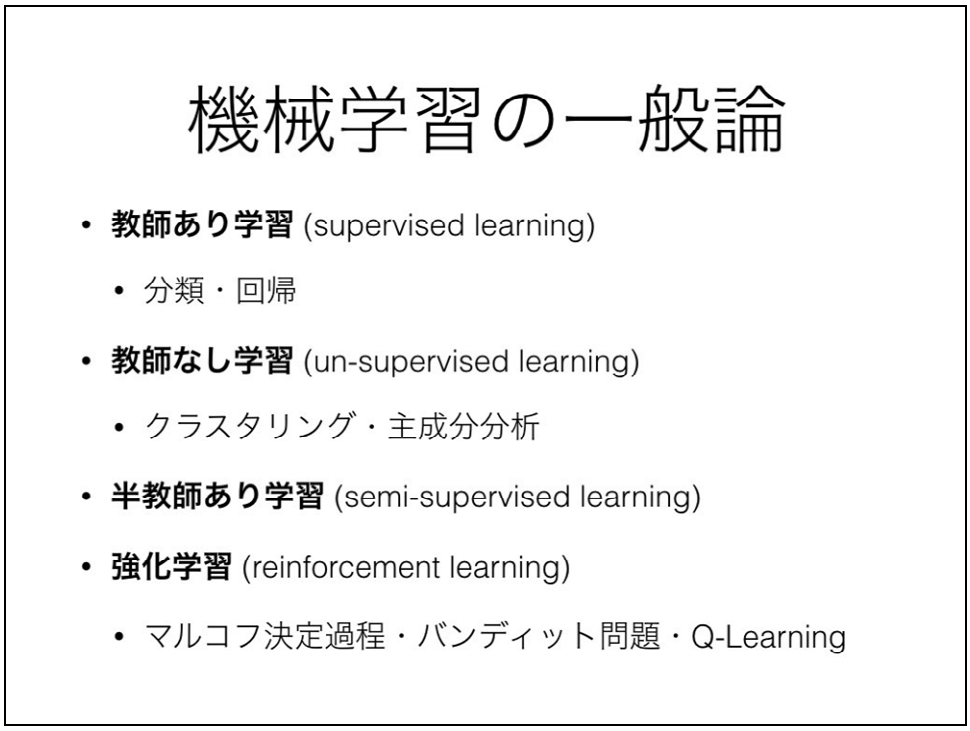
(2-1)



(2-2)

【小田】　はい。スマートニュースの小田です。遠藤さんからは、コロキアムの概要と、遠藤さん自身の発表の内容を話させていただきました。私からは、後半に私自身の発表内容についても発表しますが、まず前半に「機械学習って、何でしたっけ」という話を少ししたいと考えています。機械学習とアクチュアリー数学とは学問としてオーバーラップももちろんあるとは思うのですけれども、結構違うところもあります。仮に数式や手法に共通するものがあったとしても、物の見方というのですか、解釈のしかたはいろ

いろ違うところがありますので、そこを 20 分ぐらいかけて、少し説明をしようかと思っています。



(2-3)

まず、機械学習はいろいろな種類があるのですけれども、これも、明確にここからここまでがこのような学習で、ここからここまでがそこの何とかというわけではないのですけれども、大枠でいうと、教師ありと、教師なしと、それ以外というように分かれると思うのです。

まず、教師あり学習と呼ばれているものが、これはデータに何か正解が埋め込まれているようなケースです。表形式などでデータが与えられていて、学習用のデータにはその答えが載っているのだけれども、評価用のデータにはそのデータが欠落していて、その答えがないという状態で、評価用のデータに関して、その正解を当ててくださいというものが教師あり学習です。分類や回帰などが有名な例でしょう。一方の教師なし学習というものは、そのように学習用のデータと評価用のデータに分かれていなくて、例えば似たようなデータでグルーピングしてくださいといったタスクになります。

教師あり学習の例を少し紹介しましょう。皆さん、友達の家などに行って、ご飯を出してもらうなどすると、それが和食や洋食などと分かると思うのですけれども、それはなぜかというと、以前に和食や洋食を食べたことがあって、このような料理だったら和食、このような料理だったら洋食などということを何回も経験していて、友達の家でそれが出されたりすると、そこに含まれている材料や見た目から判断して、これは和食だ、洋食だ、中華料理だなどということが分かるわけです。

一方の教師なし学習というものはそうではなくて、例えば友達の家に行ったら、料理が 3 種類ぐらい出て、それは今まで見たことがない料理かもしれないわけです。例えば何料理かを当てることはできないかもしれないけれども、「この料理とこの料理は味がよく似ている」、「この料理とこの料理はあまり味が似ていない」などということで、過去の経験がなくても、それぐらいのことは言えたりするわけです。先ほど遠藤さんが主成分分析に関して発表しましたが、主成分分析というのは、例えばよい線型部分空間を見付ける、よい特徴量を選んでくる、などということに相当します。このような手法も教師なし学習の 1 つの例と考えて良いと思います。

教師あり学習と教師なし学習との最も大きな違いはどこにあるのでしょうか。結局のところ、学習とい

う以上は、誰かが教師の役割を果たさないといけないのです。例えば、教師あり学習の場合は、明示的にデータに正解が埋め込まれています。だから、手法を変えても、基本的にやりたいことは変わりません。もちろん、手法により精度がよくなったりコストが変化したりすることはありますが、基本的には、手法によって、やりたいことが変わるということはないのです。しかし、教師なし学習の場合は、基本的にその正しさは、どのような手法を使うかというその手法そのものに正しさが入り込んでいます。例えば A という手法を使った場合と、B という手法を使った場合、答えが全然違うということがもちろん起こり得るし、どちらがいいのかということも、それは教師あり学習の場合と違って、どちらの方がいいともなかなか明示的には言えないところがあります。

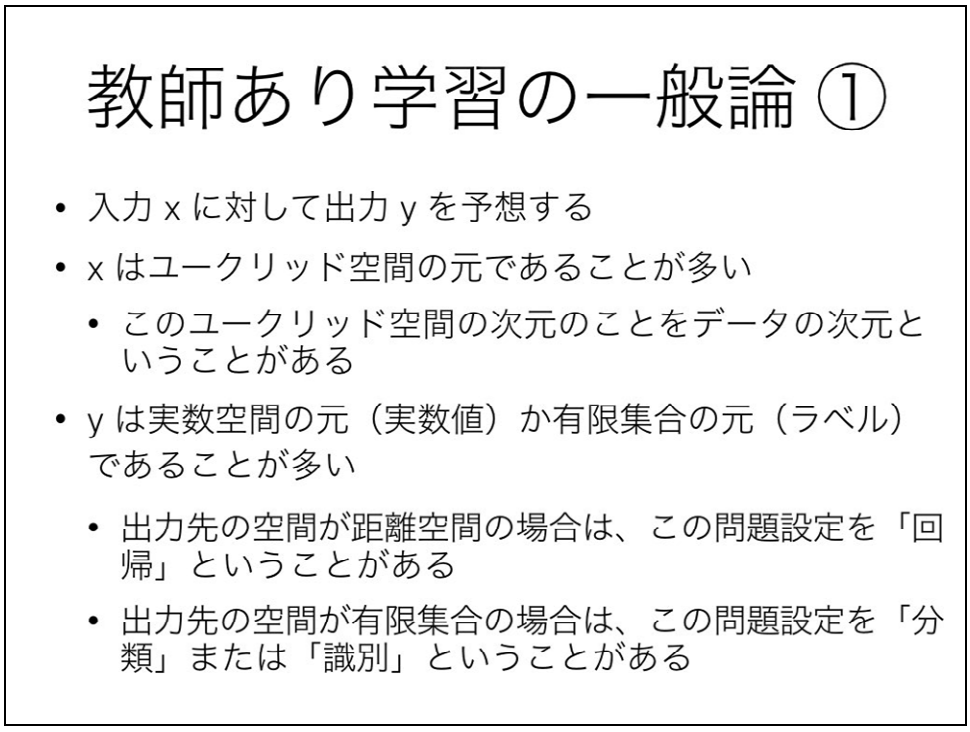
半教師あり学習というものは、あまりなじみがないかもしれないのですけれども、教師あり学習の一種だと思ってもらってもいいと思うのです。学習用のデータにも正解ラベルが欠落している場合があるというものです。それでは、そのラベルがついていないものは全部捨て去って学習すればいいのではないかという意見もあるでしょう。それはそうで、それはただの教師あり学習になるのですけれども、ラベルがついていないようなデータも交ぜて学習した方が、実は予測精度が上がるケースもあると言われていて、このような手法を積極的に実践されている方もいます。例えばアンケートなどで、「あなたは男性ですか、女性ですか」のように聞いて、でも、答えてくれない人などがいるわけです。そうすると、その学習のデータには、「男性」、「女性」と、「答えてくれなかった人」がいるわけだけれども、答えてくれなかった人のデータを全部外して学習するのがいいのか、それとも、答えてくれなかった人のデータを含めて学習した方がいいのかということは、そのときの状況によって変わってくるでしょう。

最近とてもはやっている手法は、強化学習と言われているもので、これはマルコフ決定過程と呼ばれる、その状態とその状態の遷移に関する情報をもらって、そのうえで、アクチュアリー的な言葉で言うと「将来収入現価」を最大化しようという問題です。特に状態が 1 個しかないような場合は、バンディット問題と呼ばれていて、広告収益最大化などによく応用されています。それから、最近 Q-学習というものがとてもはやっています。マルコフ決定過程を与えられた際に、状態から状態へと遷移する確率や、その状態に遷移した際に発生する報酬の値を知りたくなるのが古典的な統計学になじみのある方の発想だと思います。機械学習の人たちは必ずしもそうではなくて、正直、遷移確率や報酬の期待値が分からなくても良いから、「どのような意思決定をすれば、一番もうかるのか」、「どのような意思決定をすれば、将来収入現価を最大化できるのか」という問題に興味を持っています。問題の背景にある「環境」を理解することを放棄して、そのかわり、最もよい方策を学んでいこうとしています。

Q-学習とは、「今この状態にいて、この行動を取ったら、そのとき将来収入現価はどれくらいか」という Q-関数の形を学習する学問領域です。その Q-関数の学習のしかたはいろいろありますので、Q-学習というのは手法ではなくて、Q-関数の形をどうにかして知ろうという行為に相当します。Q-関数の学習の一例としては、fitted-Q-iteration と呼ばれる手法があり、これはアルファ碁という囲碁のソフトが実際に利用した手法だと言われているのですけれども、それで韓国のプロ棋士を打ち負かしたということで、非常に話題になりました。アルファ碁は、Q-関数をニューラルネットワークを用いて関数近似したとされています。

(2-4)

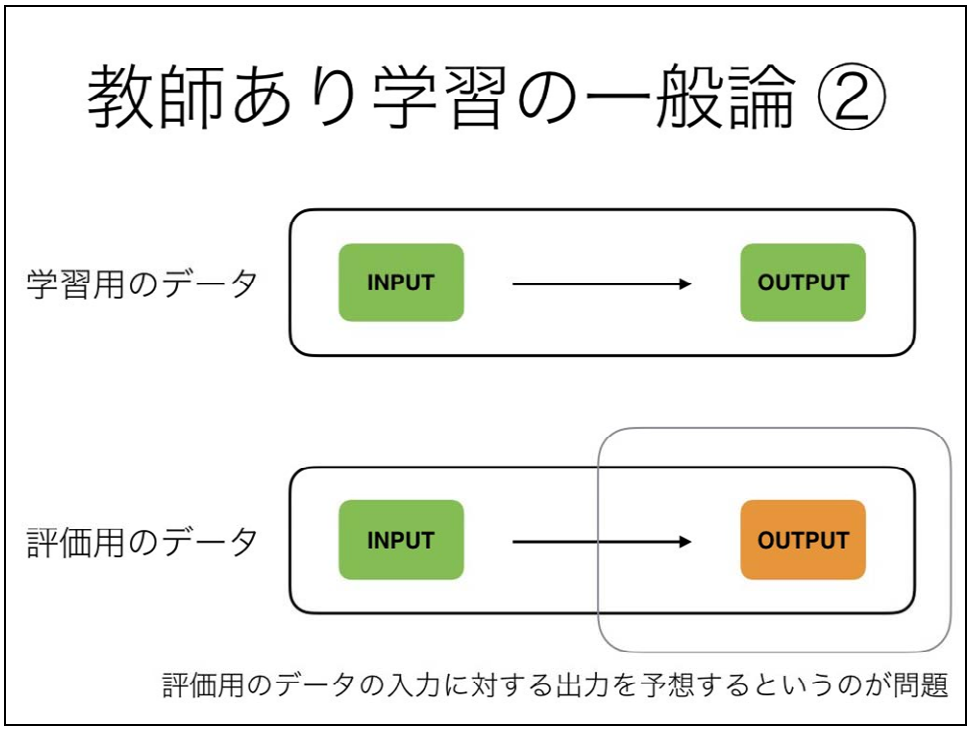
教師あり学習と教師なし学習について説明しようと思っているのですけれども、まず今日のところは、教師あり学習と教師なし学習の違いだけ分かっていただければ、それで結構です。



(2-5)

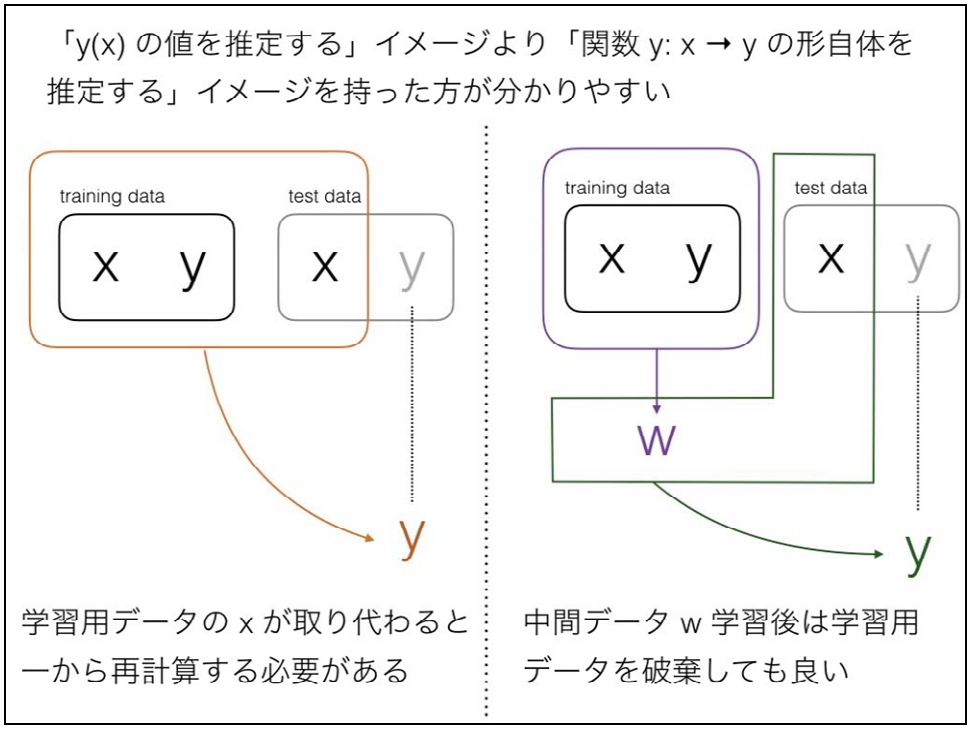
教師あり学習というものは、何か入力に対して、出力を当てるというものなのです。教師なし学習と、まずここが違います。何か入力が来たら、出力を予想してくださいというものが教師あり学習になります。入力は、普通はユークリッド空間の元であることが多いです。もちろん、これは本質的な仮定ではなくて、別に多様体でもいいし、離散集合でもいいし、何でもいいのですけれども、普通はユークリッド空間を仮定します。このユークリッド空間の次元のことを、よくデータの次元と言います。

出力の方は、普通は実数値、もしくは有限集合の元である場合はラベルです。もちろん、これも本質的な制約ではなくて、出力先が多様体でもいいし、複素空間でもいいし、何でもいいのですけれども、普通は実数です。出力先が例えば実数のように、距離空間のときは、よくこの問題を「回帰」と言います。それから、出力先が有限集合の場合は、「分類」と呼ぶことが多いです。出力先が有限集合だと、出力値はラベルとみなせるので、そのラベルに分類するということで「分類」、もしくは、そのラベルの中で特別重要な何かラベルがあるのだとすれば、そのラベルだけを識別するという意味で、「識別」というような呼び方をすることもあります。



(2-6)

教師あり学習は、先ほども申し上げましたけれども、まず学習用のデータと評価用のデータが与えられていて、学習用のデータに関しては、その入力値（インプット）と出力値（アウトプット）のペアが与えられているわけです。けれども、評価用のデータは、入力値の方は教えてもらっているけれども、出力値の方は教えてもらえません。このような状況で、学習用のデータの入力値と出力値の関係を用いて、評価用のデータの入力値に対応する出力値を当ててみてくださいというものが教師あり学習です。そして、どのぐらいの正答率で合わせられたかということが、その学習の成果なわけです。だから、たくさん手法があったら、その中で最も出力値を正しく当てられた手法が、最も評価されるというわけです。



(2-7)

ここは、機械学習と古典的な統計学と少し考え方が違うところかなと思っているのですけれども、与えられているものは「学習用のデータの入力値と出力値」と「評価用データの入力値」であり、当てに行かないといけないものは「評価用データの出力値」なので、引き数が 3 つあって返り値が 1 つあると思われがちなのですけれども、機械学習の人たちのイメージはそのような感じではなくて、入力値と出力値の関係を与える関数の形そのものを推定しに行くというイメージなのです。だから、引き数は「学習用のデータの入力値と出力値」であり、返り値は「入力値に対して出力値を対応させるという関数の形」そのものなのです。これは結構、機械学習的なものの考え方で、心に留めておかなくてはいけないと思います。それは、物の考え方というだけではなくて、実務的な事務処理手順も変わってきます。もし「学習用のデータの入力値と出力値」と「評価用のデータの入力値」の 3 つから「評価用のデータの出力値」1 つを当てるとなると、評価用のデータの入力値が変わるたびに、何回も何回も計算をし直さないといけなくなってしまいます。けれども、関数の形そのものを推定するというように考えると、「学習用のデータの入力値と出力値」だけから関数形に関する学習結果に相当する中間データをまず作ってしまえば良いことが分かります。いったんこの中間データが作れると、「学習用のデータの入力値と出力値」は、もう捨て去っていいわけです。この中間データさえどこかのメモリーやハードディスクに置いておけば、どのような評価用データの入力値がやって来ても、すぐに対応する出力値を算出できます。多くの場合、この「学習用データの入力値と出力値」から中間データを求めるところに圧倒的にコストがかかって、中間データと「評価用のデータの入力値」とから「対応する評価用データの出力値」を求めるところにはコストはかかっていません。だから、機械学習の人たちにとっては、中間データを作るところまでにはコストをかけてもいいけれども、そこから先の計算にはコストがかからないような学習方法を考えることが重要になります。

教師あり学習の一般論 ③

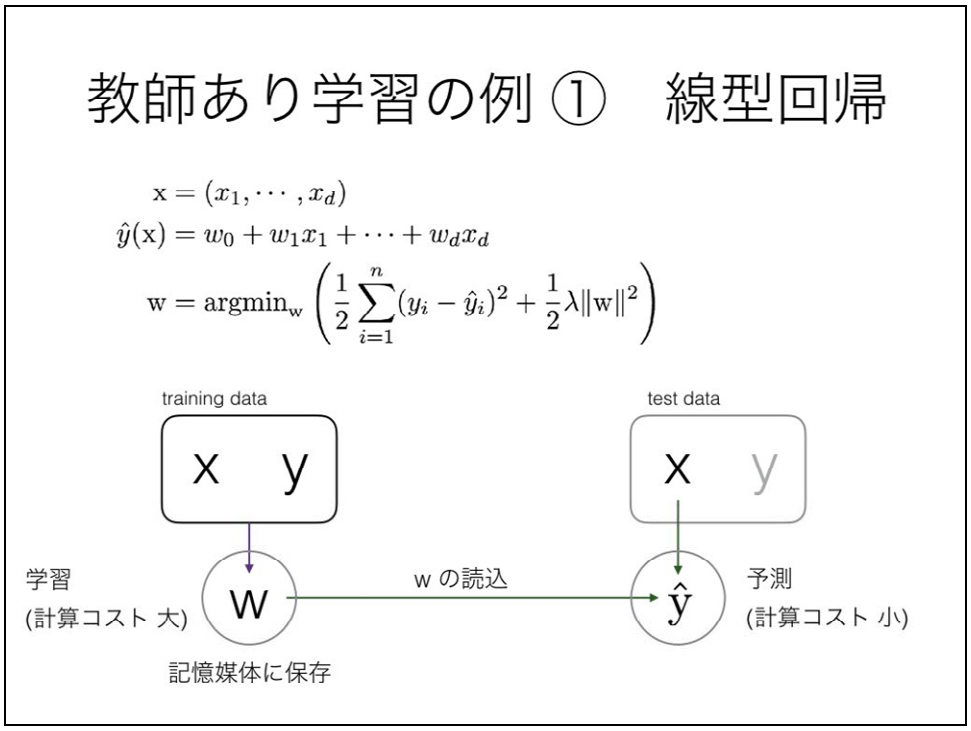
- 学習と予測
  - 学習：「学習用データ (x, y) → 中間データ w 」
  - 予測：「中間データ w と 評価用データ x → y の推定値」
- 古典的な統計学の解釈
  - 予測部分は「モデリング」
  - 学習部分は「モデルのパラメータの推定」
  - w はモデルのパラメータの推定値なのでデータの要約にもなっている
- 機械学習の解釈
  - w は y の**予測精度の向上**と予測コストの低減のためのトリック
  - 予測部分が「背後に存在する（かもしれない）真のモデル」と一致している必要はない
  - 中間データ w がデータを説明 (要約) する必要はない
  - w は **関数・確率変数・乱数**の場合もある

(2-8)

古典的な統計学と機械学習とでは、同じ数式でもものの見方が結構違っています。例えば、先ほど説明したプロセスを、「学習」と呼ばれる「学習用のデータの入力値と出力値」から「中間データ」を作る箇所と、「予測」と呼ばれる「中間データ」と「評価用データの入力値」から「評価用データの出力値」を推定する箇所に分解したときに、古典的な統計学の解釈だと、この「予測」部分を「モデリング」と呼んでいて、中間データを求めるところは「パラメーターの推定」というように呼ぶわけです。古典的な統計学においては、この中間データ（パラーメーター）そのものを精度よく求めたいというモチベーションがあります。この中間データ（パラメーター）というものは、モデルの要約やデータの要約になっていると信じられているので、この中間データ（パラメーター）を精度よく求めることがとても大事だと考えられています。

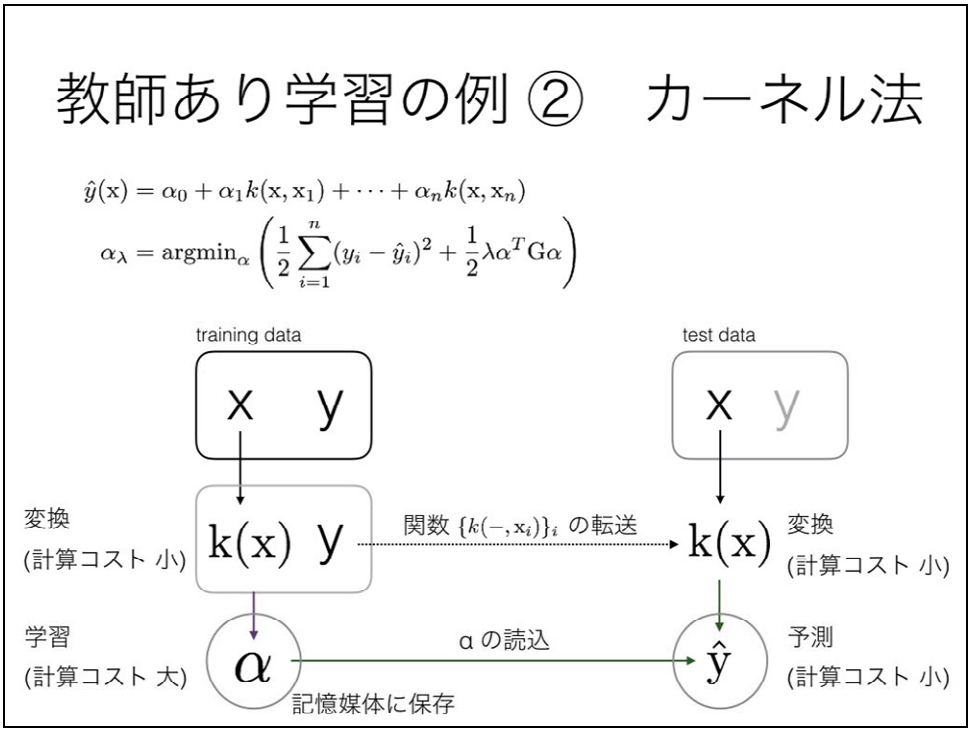
一方の機械学習というものは、中間データ（パラメーター）そのものには、ほとんど興味がありません。中間データ（パラメーター）は、あくまでも評価用データの出力値の予測精度を向上させて、かつその予測コストを減らしたいという、そのトリックなのです。あくまでも学習していることは、入力値から出力値を対応させるという関数の形そのものであって、中間データ（パラメーター）を精度よく求めたいというわけではないところに注意をします。その結果として、モデル（のパラメーター）が真のモデルと一致しているかというと、別に一致している必要はないし、実際全然一致していないわけです。ですから中間データが、そのデータを説明し要約している必要などもないです。さらに言うと、機械学習では、中間データによく乱数を交ぜてしまうのです。ここが非常にポイントで、「学習用データの入力値と出力値」とから「中間データ」ができて、「中間データ」と「評価用データの入力値」から「評価用データの出力値」ができるので、中間データにノイズが乗っかると、最終成果物である評価用データの出力値にノイズが乗っかってしまうのではないか、むしろ中間データからノイズを減らすことにより最終成果物からノイズが取り除かれるのではないかと、普通の人はそう考えるのだけれども、ここが統計学の面白いところで、そうではないのです。モデル、モデルのパラメーター、このモデルによる出力値、この 3 者に乗っかっているノイズというものはトレードオフの関係があって、どれかを精度よく求めようと思うと、そのしわ寄せが

他のところに行ってしまうのです。だから、中間データ（パラメーター）を精度よく求めようと思うのだったら、最終成果物である出力値の方にノイズが乗っかることを許容しないといけないし、最終成果物である出力値を精度よく求めたいと思うのであれば、中間データ（パラメーター）の方にどうしてもノイズを乗せないといけないのです。機械学習というものは本当に面白くて、最終成果物のバリアンスを減らすために、あえてモデルや中間データに人為的に乱数を加えて、モデルや中間データのバリアンスを増やすわけです。このような行為は、古典的な統計学の人からするととてもショッキングで、モデル（のパラメーター）というものを知りたいはずなのに、そのモデルやパラメーターにゴミをたくさん交ぜ込んでしまうと、そのようなことをしたら、そのモデルによる予測値がどんどん悪化してしまうのではないのかと思ってしまうのだけれども、必ずしもそうではないというところが統計学の面白さだと思います。



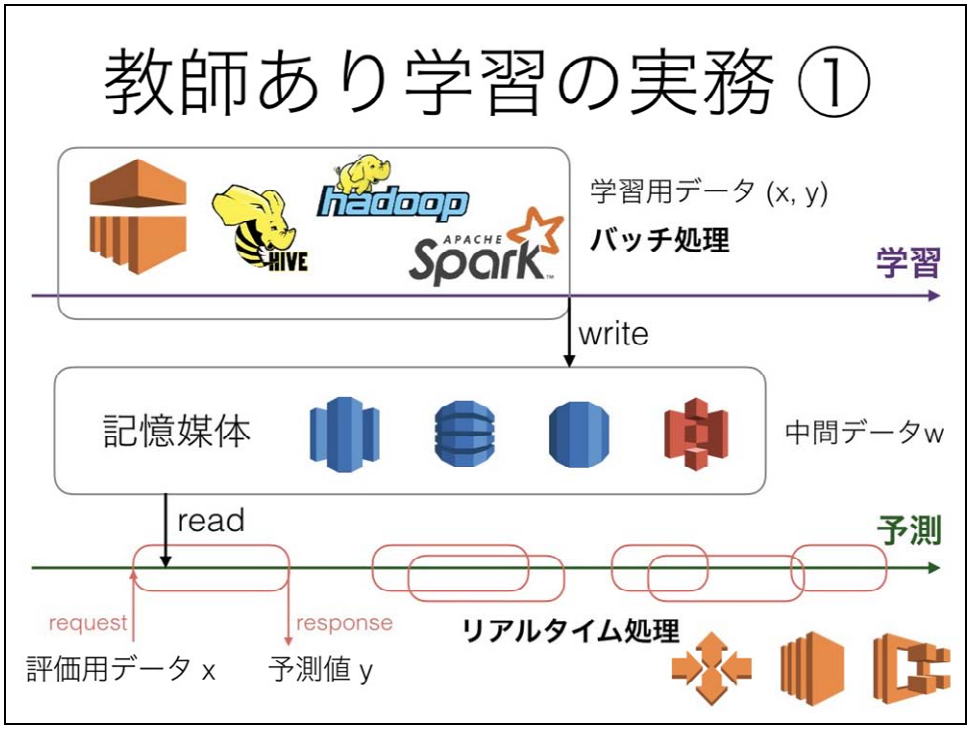
(2-9)

例えば、教師あり学習の典型的な例だと、線型回帰と言われている手法があって、これは、出力値 y は入力値 x とパラメーターw の内積で与えるというものです。そのパラメーターw は、学習用のデータの入力値 x と出力値 y からだけで計算できるという事実が重要です。つまり、評価用のデータの入力値は必要ないというところがポイントです。だから、学習用データと評価用データとを同じタイミングでもらう必要は全くなくて、学習用のデータだけ先にもらってもいいのです。仮に「評価用のデータはあげません」と言われても、仕事が開始できるのです。つまり、学習用のデータだけからまず中間データだけを計算してしまいます。この作業は普通とてもコストがかかります。例えば線型回帰の場合だと、仮に逆行列を求めるのだとしたら、そのサイズの 3 乗に比例する計算量がかかるから、大変な時間がかかります。でも、いったん中間データが計算できたら、出力値 y は入力値 x とパラメーターw の内積だから、ほぼ線型の計算量で計算できる。したがって、仮に中間データの算出に計算コストがかかったとしても、いったん中間データが算出できれば、これにより、入力値から出力値に対応させる関数の形が完全に決まるというわけです。



(2-10)

カーネル法を使った場合、カーネル法について詳しく説明するつもりはないのですけれども、そこは分かってもらう必要はなくて、仮にカーネル法を使ったとしても、やることは同じだということです。α というパラメーターがあって、α というパラメーターさえ分かったら、あとは一瞬で計算ができるのです。α を計算するところが、とてもコストがかかるのです。これは、そのレコード数の 3 乗に比例する計算量ですが、非常にここもコストがかかるのだけれども、いったん計算できてしまったら、もうそれで関数の形が学習できているというわけです。

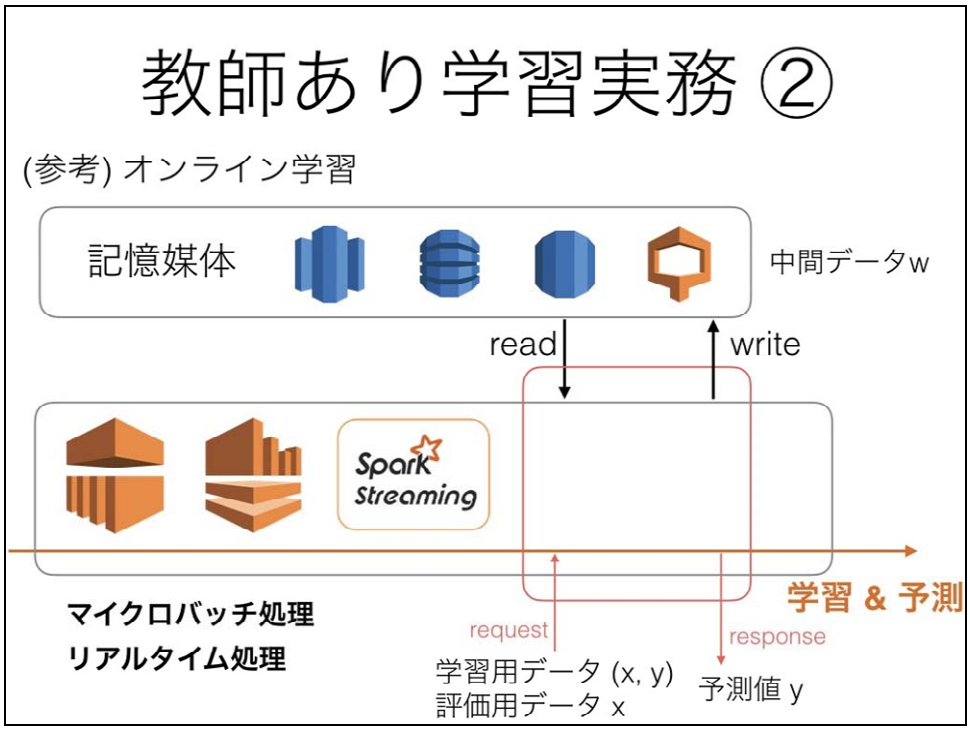


(2-11)

少しイメージがつきにくいかもしれないので、実際に業務をするとしたらどのような感じになるのかと

いうことを踏まえて、いかに関数の形を学ぶことが重要かを説明したいと思います。普通、機械学習では、学習と予測とをばらばらに実施します。まず学習に関しては、バッチ学習と言って、例えば 1 日に 1 回や 1 時間に 1 回という頻度で実施します。それは、例えば 1 時間に 1 回やるのだったら、過去 60 分間のデータを全部集めてきて、それを使って学習して、その学習した結果つまり中間データを記憶媒体に保存します。記憶媒体は、メモリーかもしれないし、ハードディスクかもしれないし、他の機構によるものかもしれないです。この中間データというものは、先ほども述べましたけれども、別にこのデータ自体には意味はないのです。それは、ただのデータでしかないです。ただ、これがあると予測ができるのです。だから、中間データというものはまさに関数の形そのものを表したデータであり、リクエストが来る度に記憶媒体から中間データを読み込んできて予測を行うというわけです。

このやり方のいいところは、予測するときに記憶媒体に書き込みを実施する必要がないということです。だから、予測系が並列で動いていてもいいし、例えば実際にスライドでも書いているのですけれども、リクエストが来て、まだリスポンスを返していないときに、その処理中にまたリクエストが来たとしても、それはそれでかまわないわけです。なぜかというと、記憶媒体に書き込みはしていないですから、コンフリクトは起きないというわけです。これは一般的な学習及び予測のしかたで、サービスに実際に応用されています。これがまさに評価用のデータが来たらすぐにリスポンスを返せる理由なのです。Amazon などで買い物をするとしたら、0.1 秒でリスポンスが返ってこないと困ってしまいます。だから、この予測という処理は、本当に一瞬で実行できる必要があります。



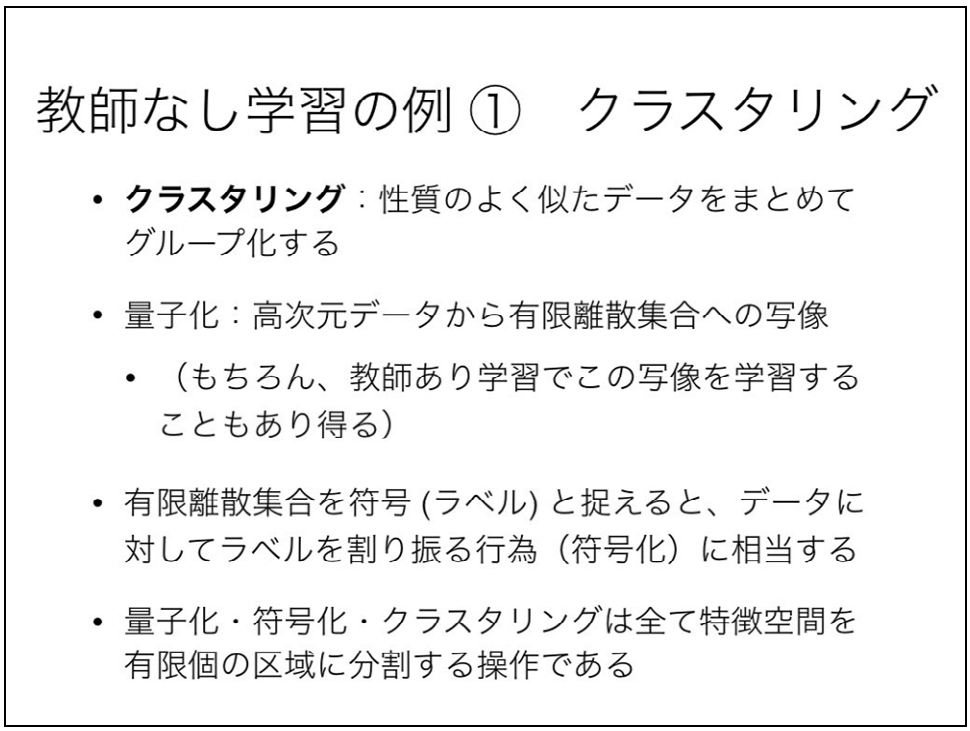
(2-12)

一方、オンライン学習と呼ばれているやり方もあって、こちらはバッチ学習とは少し考え方が違います。こちらは予測系と学習系を分けずに、データが来たらそのデータをすぐ学習に使ってしまうというやり方です。この方法は実装が難しい。なぜかというと、リクエストが来るたびに記憶媒体に書き込みをしないといけないからです。この処理でコンフリクトが起きることもあります。また、先ほどのバッチ学習の場合だと、例えば学習系で障害が起きたとしても、中間データさえ記憶媒体に残っていれば、予測は常にできるわけです。企業（サービス）にとって重要なのは「学習」ではなくて「予測」の方です。学習系は正

直 2 日間止まったとしても、「予測」さえできていればそれでサービスは稼働し続けるわけですけれども、予測系がつぶれてしまうと、もうこれはサービスが終わってしまうわけです。ですから、オンライン学習というものは、結構リスキーな学習の運用方法で、このようなやり方を実施してしまうと、何か障害が起きるなどすると、もうサービス全体が止まってしまうというわけです。

(2-13)

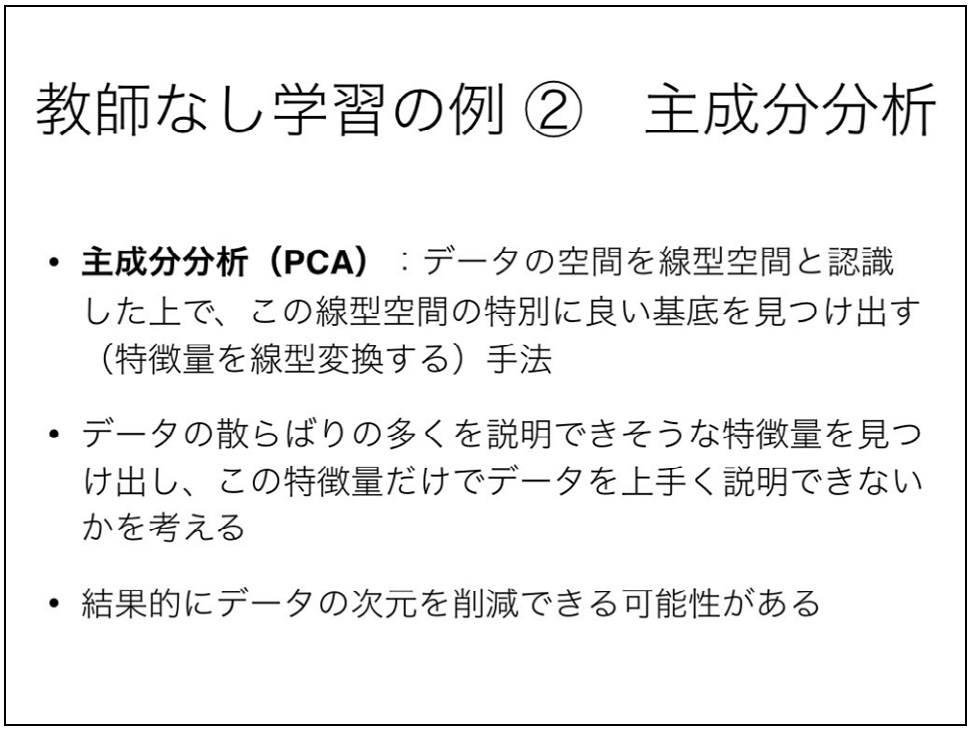
次に、少し教師なし学習を見ていこうと思うのです。



(2-14)

教師なし学習は教師あり学習と全然違っています。教師なし学習の例として、クラスタリングのようなものがあります。クラスタリングは、性質のよく似たデータをまとめるということです。クラスタリング

を実施すると、そのクラスターにラベルを与えることができるので、高次元データを有限離散集合に写像するということでもあります。このように、高次元データを有限個の点に写像するということを量子化というのですけれども、クラスタリングすると結果的にデータの量子化もできるということでもあります。量子化や符号化やクラスタリングというものは、全部、特徴空間を有限個の区域に分割するという意味では同じです。そうやって特徴空間を有限個の区域に分割して何がうれしいのかというと、クラスタリングと言った場合は、クラスタリングと言った人の心の中では、もうそれを分けたということ、それ自体に意味があるでしょうというスタンスです。量子化と言った場合は、多分それを言った人は、別に分けたこと自体には、まだ今のところ意味はないと思われるが、それを新しい特徴量として使えば、きっといいことが起きるのではないかという、そのような気持ちが後ろに含まれていると思われます。



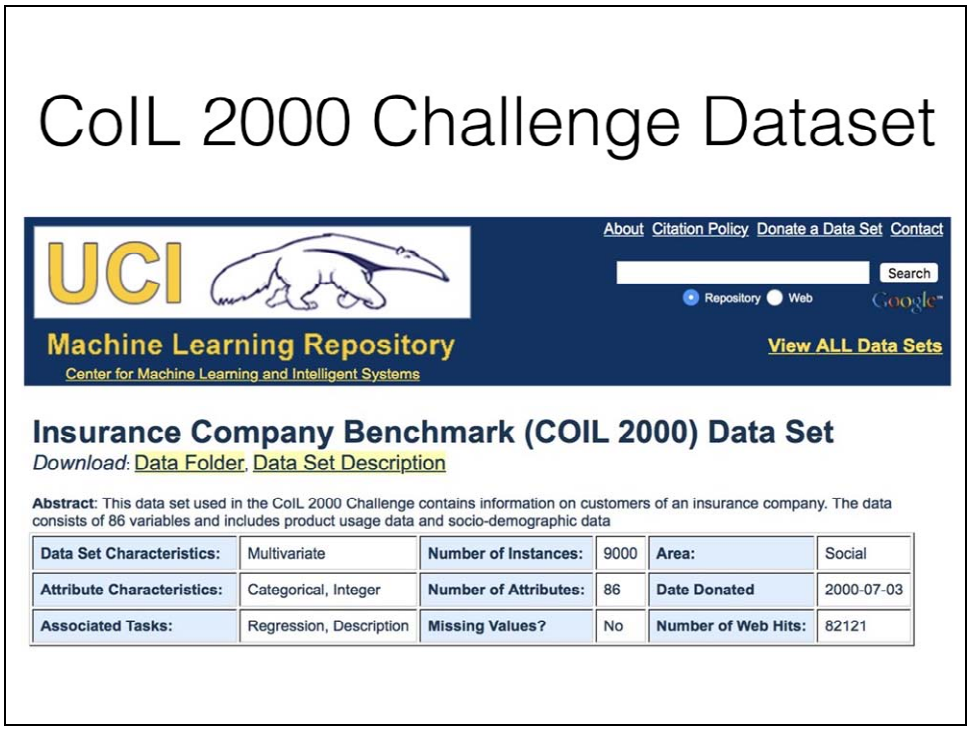
(2-15)

もう一つ、教師なし学習の例としてよく言及される手法が、主成分分析と言われているものです。主成分分析には、定式化が幾つもあります。機械学習の人たちの定式化は、確率的定式化と呼ばれる定式化をすることが多いのですけれども、多分、アクチュアリーの方になじみがあるものは、非確率的定式化と呼ばれている定式化です。例えば、元のデータ空間を低次元の部分空間に射影するのだけれども、射影後にバリアンスが最も高くなるように線型部分空間を見付けなさいとか、もう一つは、射影するのだけれども、射影する距離が最小になるように線型部分空間を見付けなさいとか、そのような定式化です。この定式化は、確率という概念が出てこないので、非確率的定式化と言われています。

一方の機械学習の人は、確率的定式化というものが好きで、これはベイジアン的な定式化です。メリットとデメリットがあるのですけれども、メリットの一つは因子分析との関係性などが見えやすいということです。機械学習の人は、確率的な定式化を好む人が多いです。でもいずれにせよ、主成分分析とは、元の高次元空間を何らかの低次元空間で上手く説明する手法だということです。そうすることによって、データの散らばりをなるべく少数の説明変数で解釈でき、データの次元が削減できるのではないかということが、主成分分析でよく言われていることです。

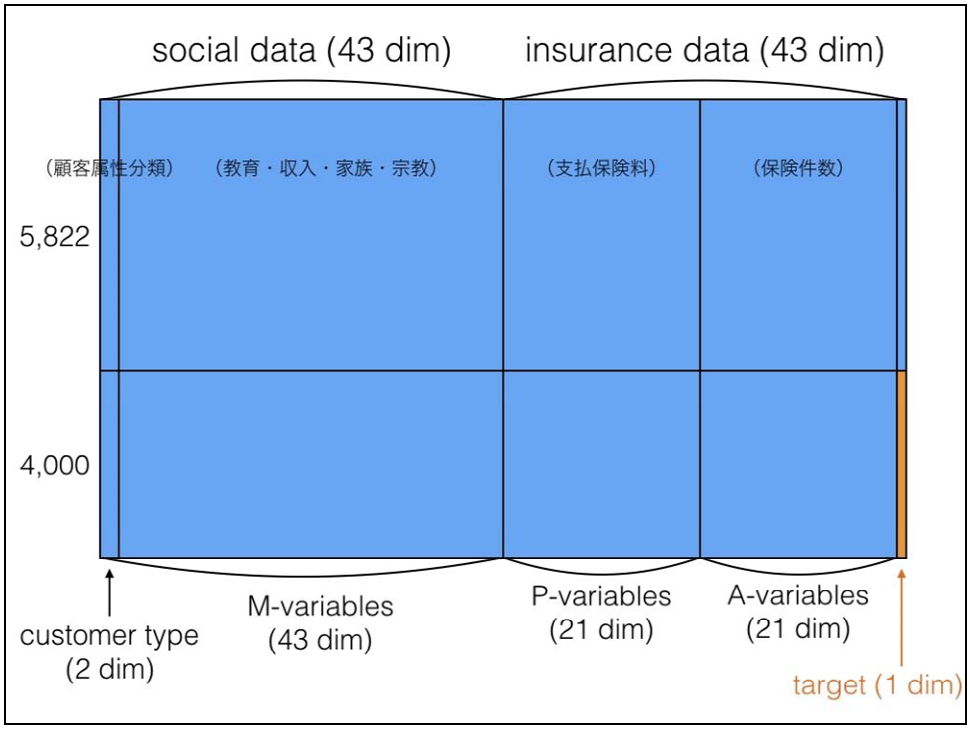
(2-16)

ここから、私の ASTIN コロキアムでの発表内容に入っていきたいと思います。



(2-17)

今回使ったデータは、「CoIL 2000 Challenge」と呼ばれているデータで、「UCI Machine Learning Repository」というところからダウンロードできます。



(2-18)

これは典型的な教師あり学習用のデータで、正解データは「0」か「1」、つまり二値変数（バイナリー変数）なので、このような課題をよく二値分類（バイナリー・クラシフィケーション）と言います。このデータに欠陥値はないです。86 次元のデータで、学習用のデータが 5,822 レコード、評価用のデータが 4,000 レコードです。もちろん、先ほども説明したとおり、評価用のデータには目的変数の値は与えられておらず、このオレンジ色の部分を青色の部分を使って求めてください、ということが問題です。86 次元あるのですけれども、前半はソーシャルデータで、例えば教育や収入や家族や宗教に関する情報が入っていて、後半の方は保険に関する情報が入っていて、支払保険料や保険件数に関する情報が入っています。ソーシャルデータの方は M から始まる変数で、支払保険料は P から始まる変数で、保険件数は A から始まる変数です。今回は、P から始まる変数に非常に着目した手法を提案しようとして、紹介しました。

- 目的変数は 0-1 の２値変数
- “1” が入る (caravan insurance を保有する) 割合は 6% 程度
- 800 レコードを提出して、目的変数が “1” であるレコードの数を競う
- 理論上の最高得点は 238

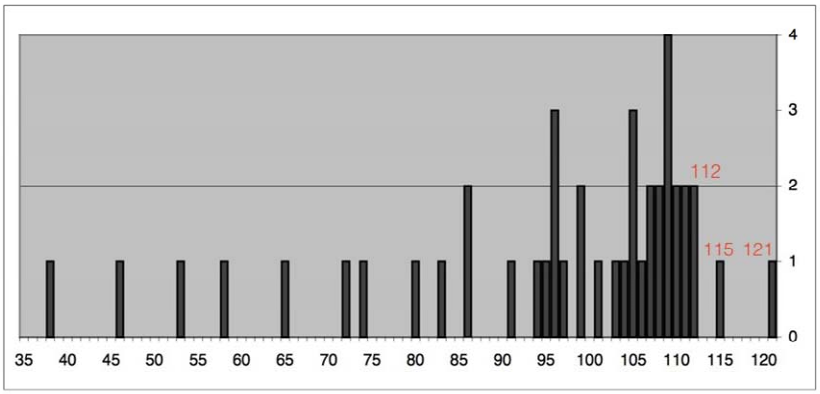


Figure 1: Frequency distribution of prediction scores.
The x-axis displays number of real policy owners in the selection that was sent in.

[出典: CoIL Challenge 2000 Tasks and Results: Predicting and Explaining Caravan Policy Ownership]

(2-19)

このチャレンジ自体は、2000 年に行われたものなのです。ですから、今から 17 年前に行われたものです。そのとき、どれくらいの得点が出せていたのでしょうか。評価用のデータは 4,000 レコードあるのですけれども、そこから、「1」が入っていると思うレコードを 800 個選んで、その 800 個の中で、本当に「1」だったものの数が得点です。評価用のレコードには 238 個しか「1」が入っていないので、理論上の最高点は 238 なのですけれども、当時は 121 という得点が最高得点でした。つまり、レコードを 800 提出したのだけれども、その中に 121 個のレコードに「1」が入っていたという人が、優勝者だったわけです。

これが、どれくらいすごい得点なのかということを述べるのは難しいです。17 年前のことなので、その当時と今では大分前提が違います。しかし、普通のロジスティック回帰で問題を解いても、110 代後半の得点は出せるので、なぜ、これほどにまで 110 未満の得点の人がいるのかということは、結構理解に苦しむところです。一方で、私がいろいろな手法を試したところでは、120 を超えることはなかなか難しいというようなデータでした。

提案手法

- 密データ P-variable 21 次元を、APD-Tree (PCA-Tree の亜種）を利用して量子化し、疎データ 100 ~ 200 次元に展開
- 元あった特徴量 134 次元（カテゴリカル変数は one-hot vector に展開) に、この形式変数 100 ~ 200 次元を加えて、L1 / L2 回帰を実施する
- 理論的背景は Ensemble trees (forest) と kernel method との関係 [Scornet 2016]

(2-20)

今回の私の提案手法、つまり ASTIN コロキアムで発表した内容は、P から始まる支払保険料に関するデータ（密データ）21 次元分を APD-Tree と呼ばれる方法を用いて量子化して、これを疎データに展開して、それを特徴量に加えるという手法になります。特徴量変換の手法には、疎データを密データに変換する手法と、密データを疎データに変換するする手法との 2 種類あるのですけれども、今回は密データを疎データに変換する手法を紹介します。そして、この特徴量変換の後に、元あった特徴量にこの形式変数を加えて、L1 または L2 回帰を実施するというものです。

理論的な背景は、いろいろあるのですけれども、木があると対応するカーネル関数を定義できるのですけれども、木を用いて直接回帰を実施するのと、その木から定義されたカーネル関数を用いて回帰を実施するのとで、大体同じぐらいの精度になるという話や、APD-Tree 自体は ICML の 2012 年の論文に出たもので、そのようなものを参考にして考えました。

データの直径を 1/2 にするために必要な分割の回数はどのくらい？

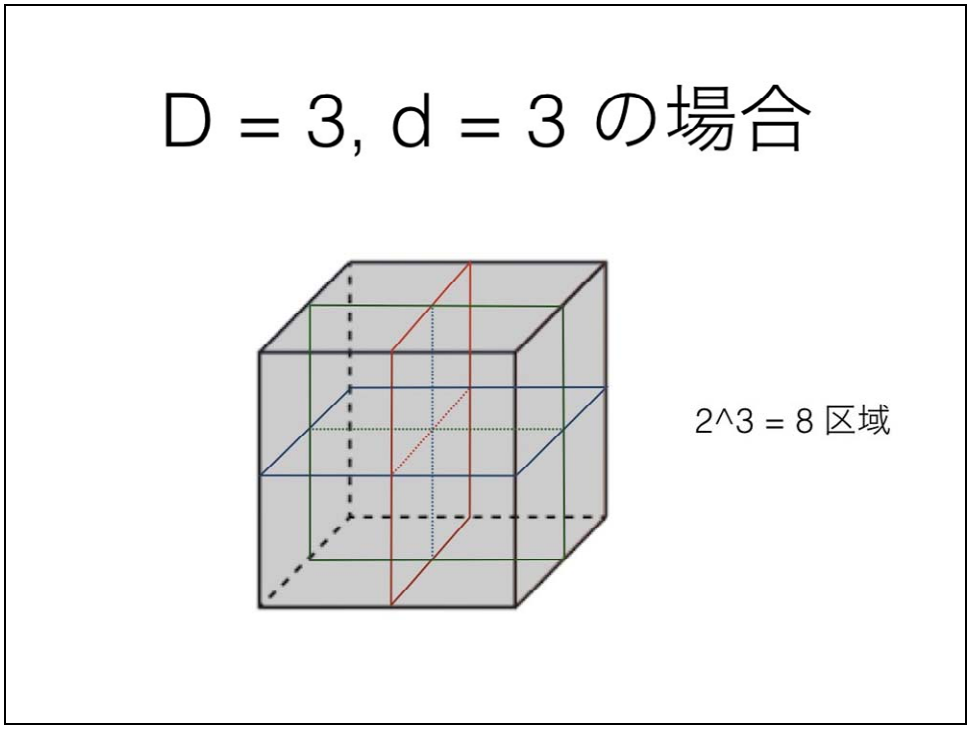
区域 S に含まれるデータの平均的直径

$$\Delta_a^2(S) := \frac{1}{|S|^2}\sum_{\mathbf{x},\mathbf{x}'\in S}\|\mathbf{x}-\mathbf{x}'\|^2 = \frac{2}{|S|}\sum_{\mathbf{x}\in S}\|\mathbf{x}-\mathrm{mean}(S)\|^2$$

- 特徴空間の次元が D の場合は、(通常) 2^D 個の区域に分割する必要がある
  - 次元の呪い (curse of dimensionality)
- 特徴空間の本質次元が d << D の場合
  - 2^d 個の区域への分割で平均的直径を半分にできる場合がある

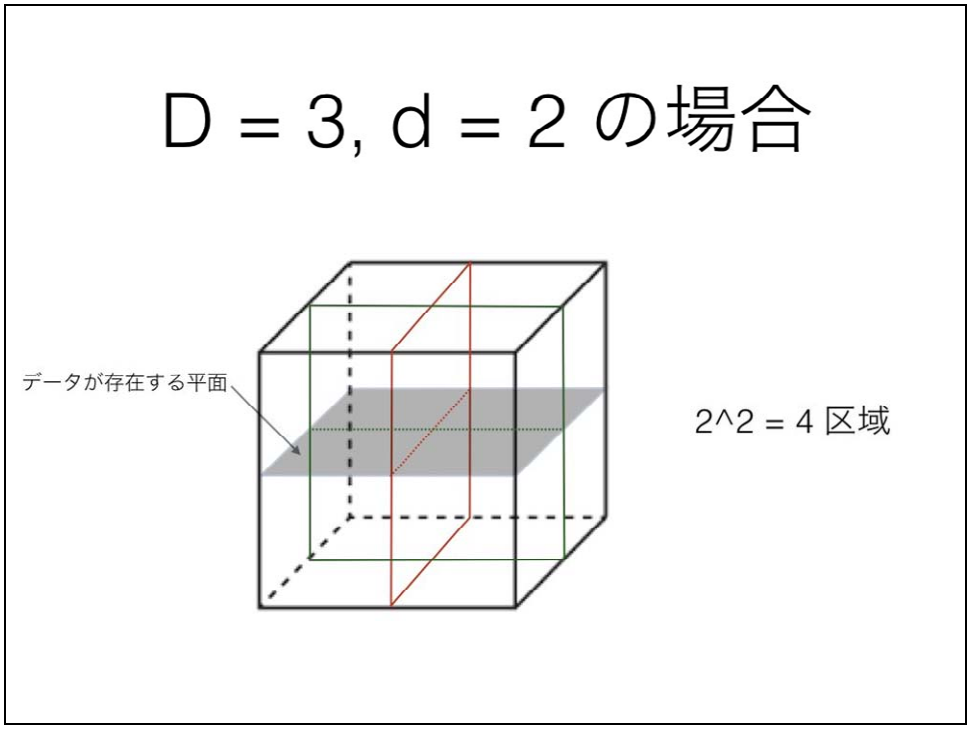
(2-21)

データの量子化の話を少ししたいのですけれども、例えば机の上にコーヒー豆などをバアッと散らばせてしまったとしましょう。コーヒー豆などを散らばせると、その直径や半径のようなものが定義できると思うのですけれども、それを例えば半分にしたいと思ったら、机を何分割しないといけないのか、少し考えてください。2 分割ではだめですね。その直径を 2 分の 1 にしたければ、4 分割せざるをえないわけです。そうすることによって、初めてそのサイズが 2 分の 1 になるわけです。それはなぜかというと、机が 2 次元だからです。机が 2 次元だから、机を分割して直径を半分にしようと思ったら、机を 2^2 個の領域に分割せざるをえないわけです。



(2-22)

例えばその説明が分かりにくいのであれば、もしそのコーヒー豆が 3 次元空間全体に散らばったとしたらどうなるのかを考えてみるのが良いでしょう。その場合は空間を 3 回切らないといけないから、空間を 2^3 の 8 個に分けないと、そのデータのサイズが半分にならないわけです。だから、特徴空間の次元が D だったら、普通は 2^D に分割しないといけないのです。ここで、肩に D が乗っかるということは、機械学習ではよくないとされていて、「次元の呪い」とも関係があります。肩に D が乗っかるような計算量は、非常に忌み嫌われるべきで、このようなものは、非常に計算コストを上げてしまうのです。

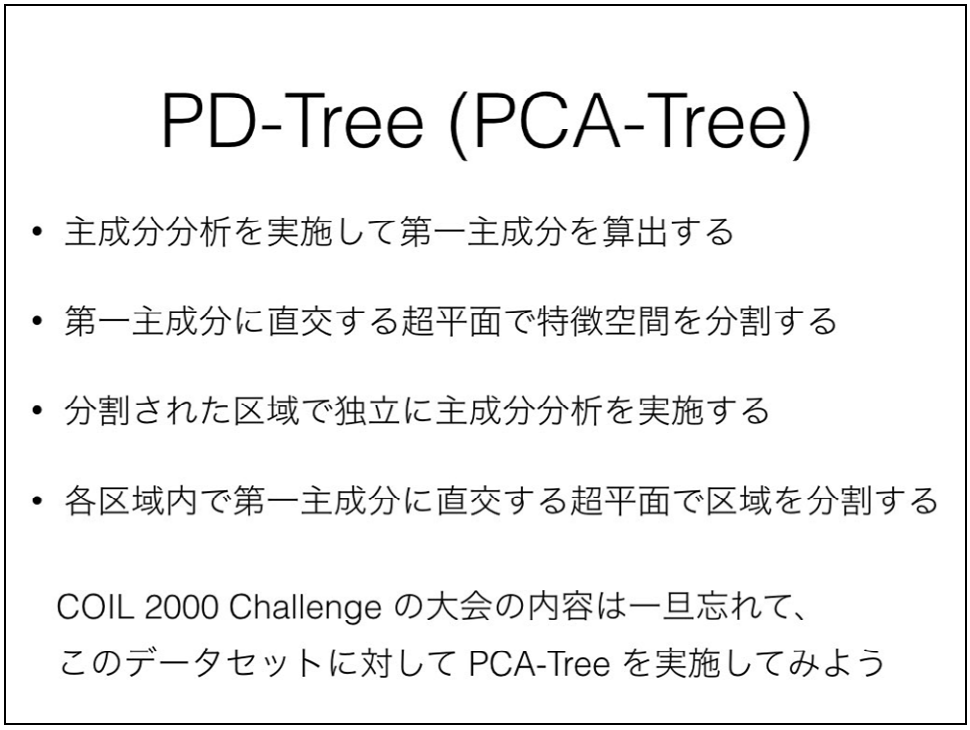


(2-23)

では、どうすればいいのかということなのですけれども、基本的に「次元の呪い」そのものを避けるこ

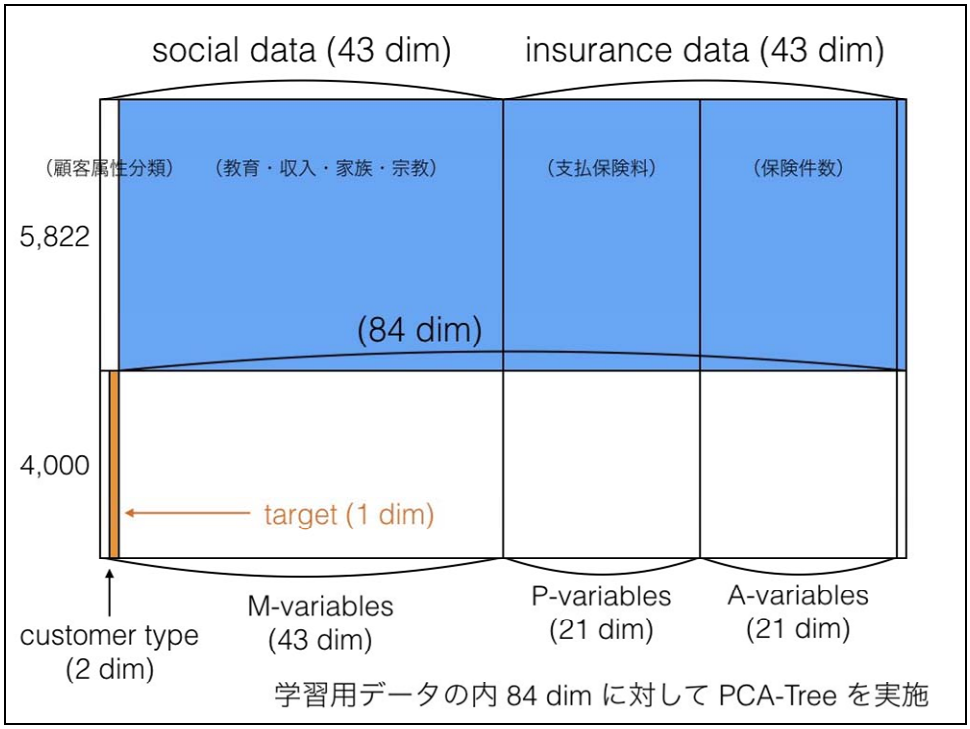
とはできないのです。しかし、もし仮に、特徴空間の次元は D なのだけれども、実際のデータは、もっと小さい d 次元の空間に押し込められているというようなことがあると、実は、2^d 個の区域への分割で、平均的直径を半分にできる場合があります。それは、先ほどの机の上にコーヒー豆を散らばせた場合がまさにそうで、空間は 3 次元なのだけれども、豆は 2 次元に広がっているから、8 個に切らなくても 4 個に切るだけで、直径を半分にできるというわけです。

例えばDが 3 で、d も 3 だと、もう 3 次元空間全部にデータが入ってしまっているから、このデータの直径を半分にしようと思ったら、どれほど頑張っても 3 回切って 8 個に分けないと、そのサイズが半分にならない訳です。しかし、例えば、もっと低い次元に押し込められている（d = 2）のだったら、2 回切るだけで、そのデータの直径を半分にできるわけです。ただ、ここで気をつけないといけないことは、間違って、このデータが埋め込まれている部分空間に水平な方向で空間を切ってしまってはいけないということです。



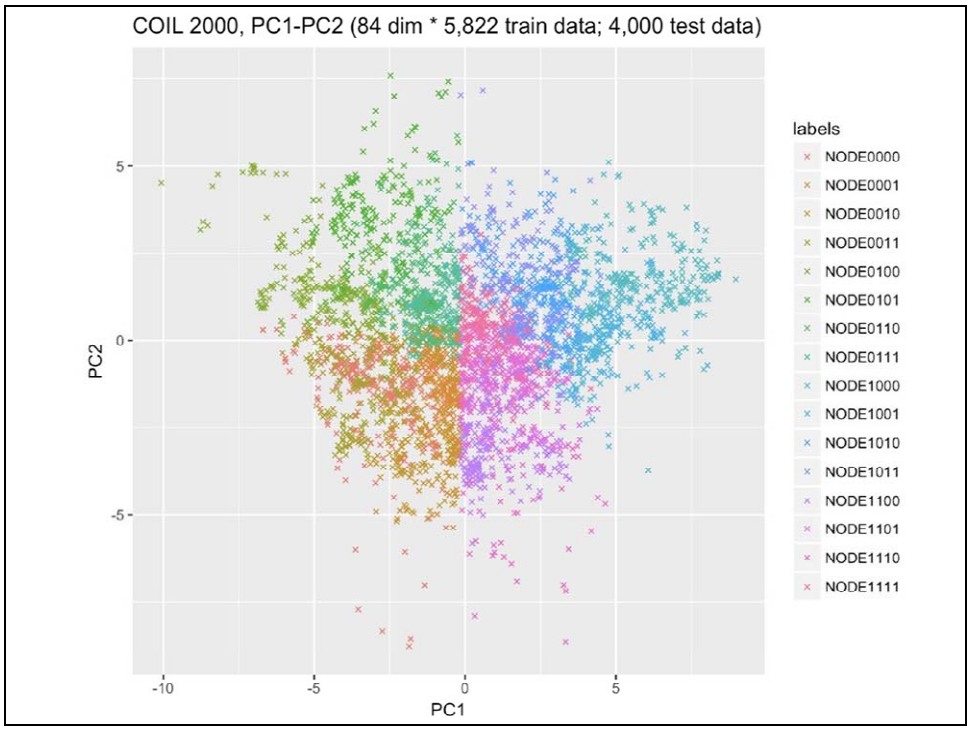
(2-24)

では、どうやったらそのようなことを避けられるかということが、次に考えないといけないことで、それが、まさに主成分分析を使いましょうということです。つまり、主成分分析を使って第一主成分を計算すれば、それに直交する超平面で切れば、必ずそのデータと直交する方向で切れるというわけです。例えばこのような場合だと、第 1 方向というものは、絶対この平面に乗っかっているから、それに垂直な方向で切れば、このデータがあるところに平行に切ることは絶対にないわけです。だから、毎回毎回主成分分析を行って、そこの第 1 方向と直交する方向で切れば、データを平行に切ることはないというわけです。データを分けたら、分けた先で、また主成分分析を行ってそこでまたデータを切ります。2 回やると 4 個に分かれて、3 回やると 8 個に分かれるというような感じで、そうやってそのデータを分けていきましょう。



(2-25)

「COIL 2000 Challenge」の大会のことを忘れて、先ほどあったデータに対して、例えば青い部分を使って、オレンジ色の部分を当てに行きましょうというようなことをやってみましょうか。

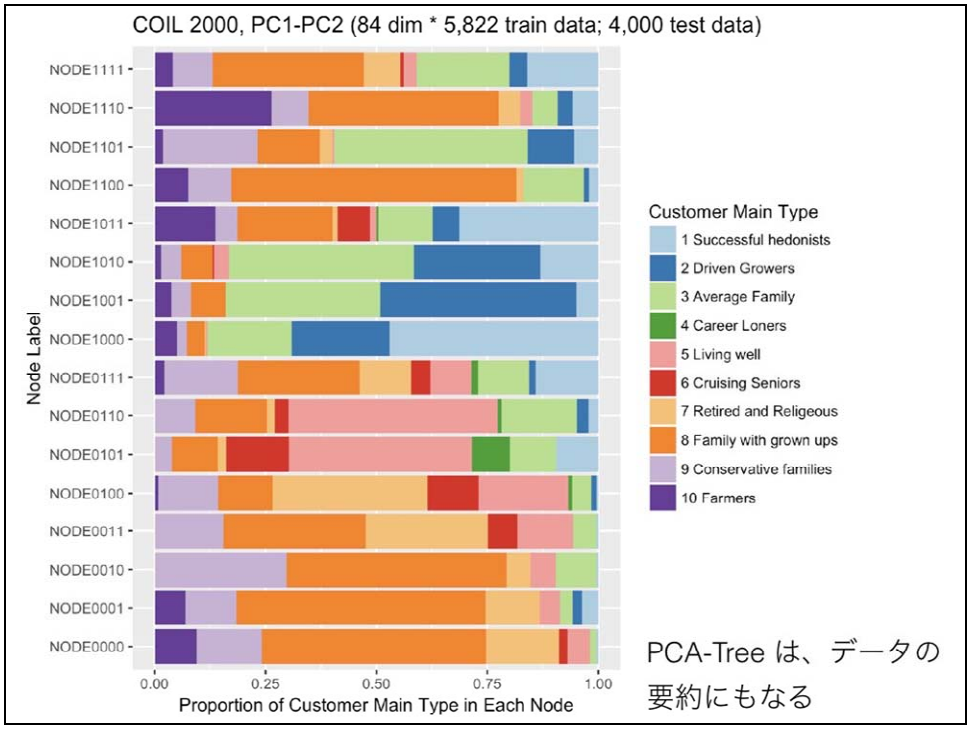


(2-26)

このスライドは、大会（CoIL Challenge）の内容とあまり関係はないですが、PCA-Tree がどのような手法かということを少し視覚的に見てもらうために用意しました。まずデータを第一主成分に直交する方向で切るので、第一主成分に直行する方向（スライドだと PC1 = 0 の超平面）でデータが区切られます。区切られた左半分の方は、またこの領域で主成分分析を行ってデータが区切られます。また、右半分の方は、こちらはこちらでまた主成分分析を行ってデータが区切られます。それでまた切られたところも、これで 4

分割されているわけだけれども、例えば左上のところは、さらにここでまた第 1 方向を求めて、それに直交する方向で切ってということをやっていくわけです。これは今、2^4 の 16 個に区切られて、それぞれの領域を別の色で表したというものです。

でも、これはただデータを区切ったというだけで、これが本当に合理的な切り方なのかはまだ分からない訳です。これが、ある程度、合理的な区切り方であると視覚的に理解していただくためにスライドを用意しました。今、スライド(2-25)の青い部分を使って PCA-Tree を実行したのですけれども、その結果が、例えばスライド(2-25)のオレンジ色の分を反映しているとすると、何かこの分割というものは、合理的だと考えられます。なぜかというと、スライド(2-25)の青い部分というものは、スライド(2-25)のオレンジ色に相当する部分（この箇所は customer type と呼ばれる属性）を使っていないから、customer type を使わずに学習したにもかかわらず、学習の結果が customer type を反映していたら、何かこの分割のしかたというものは合理的なのではないかと思えるわけです。

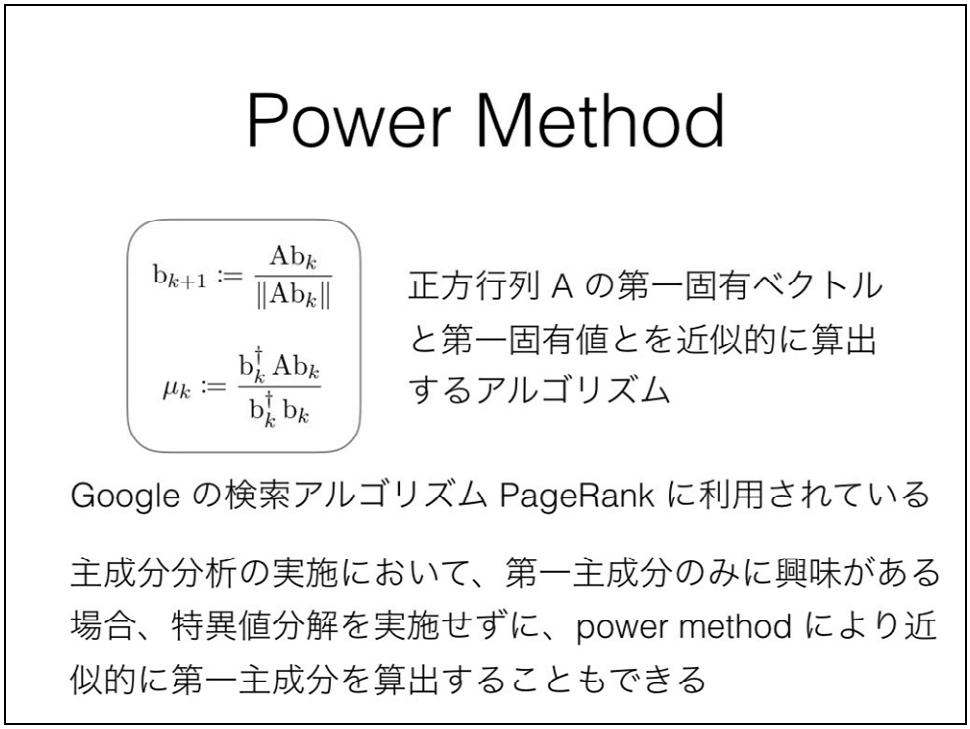


(2-27)

実際やってみると、各セグメント（PCA-Tree によって区切られた各領域）で customer type の分布は結構違うわけです。例えばここの切れ目が、ちょうどここの切れ目なのですけれども、こちら半分とこちら半分を見てみましょうか。例えば上半分の方は、「Driven Growers」という customer type が含まれているけれども、下半分には全然含まれていないです。他にも、「Living well」という customer type は、下半分には含まれているけれども、上半分にはあまり含まれていないです。よくよく見ていると、何かここで結構まず分かれているなどが分かります。例えばこの辺の領域には、濃い紫色はないなど、そのように PCA-Tree の結果が、結構 customer type を反映しているということが分かります。つまり、PCA-Tree がただデータを分けただけではなくて、結構合理的にデータを分けたのではないかと思うことができます。もちろん、これだけをもってこの手法が特別優れているということは言えないけれども、例えば、全然見当外れなことをやっているわけではなさそうだと分かります。

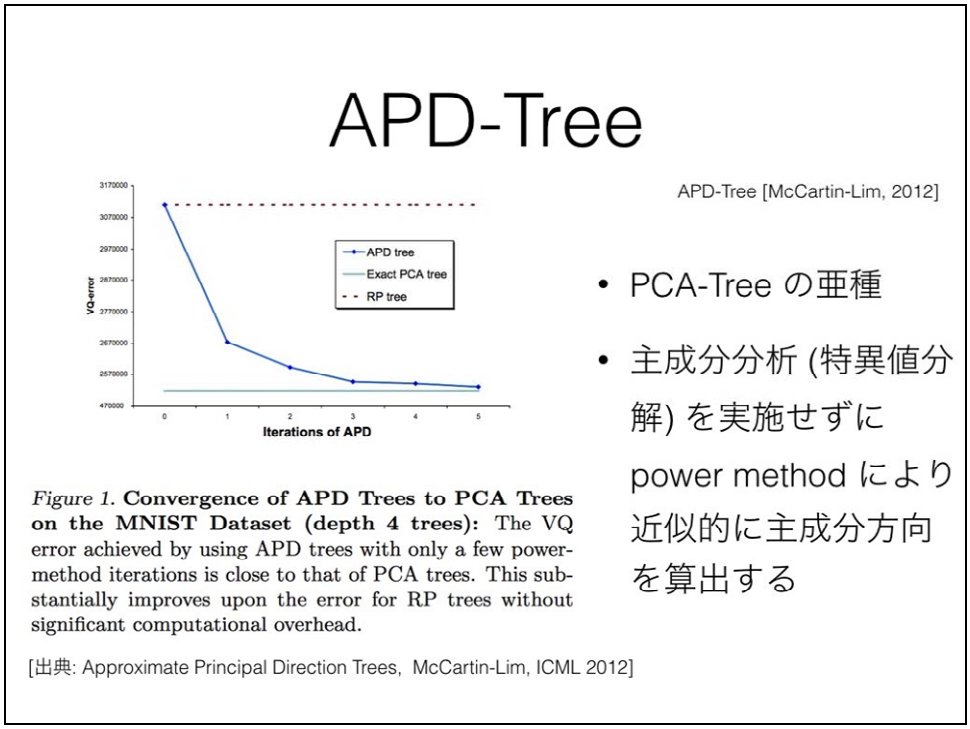
問題は、PCA-Tree を行うにしても、PCA は実際とてもコストが高いです。皆さん、線型代数などで行列の固有値や固有ベクトルなどを求めたことがあると思うのですけれども、計算過程で逆行列を求めること

になるから、基本的に、その行列のサイズの 3 乗の計算量がかかってしまうわけです。しかも、それを再帰的に行うということは、めちゃくちゃ計算コストがかかるわけです。でも、実は、それはうまく切り抜ける方法があるという話をしましょう。



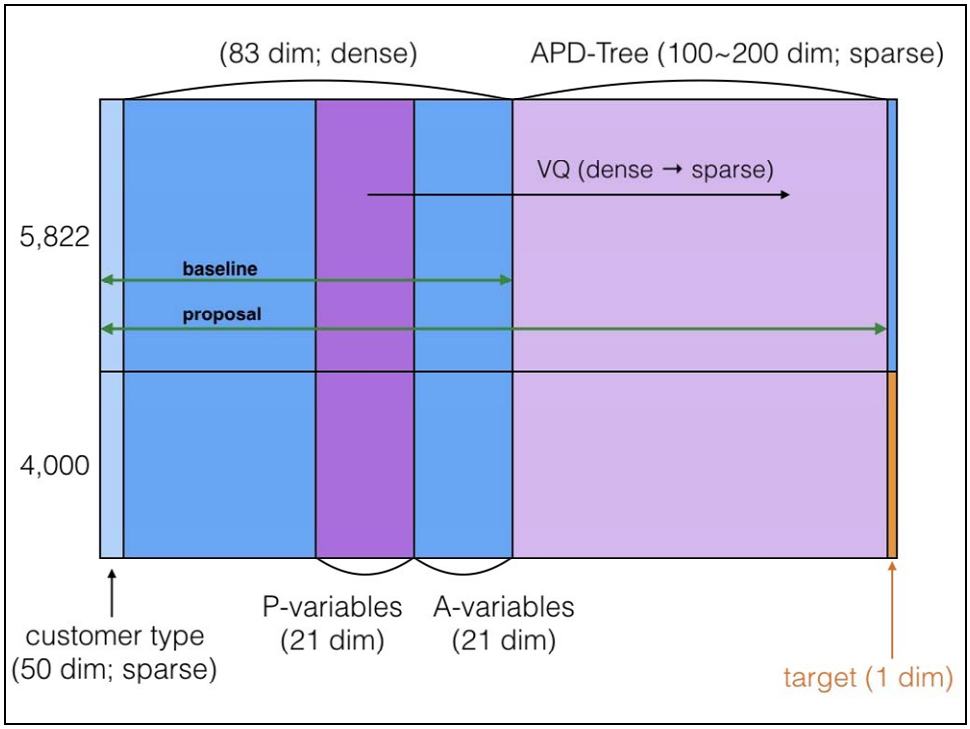
(2-28)

主成分分析は、逆行列を求めるか、そうでなければ特異値分解するしかないのですけれども、第一固有成分だけが欲しいのであれば、実は、もっと簡単に求める方法があるのです。それは、Google の検索アルゴリズムに PageRank というものがあるのですけれども、それが使っている方法と同じ方法で、第一固有成分を求めるだけだったら、実は、このイテレーションを繰り返せばよい。だから、D^3 の計算量は必要ありません。この方法により、非常に簡単に第一固有方向が求まります。この手法の収束は遅いのですけれども、実は収束の速さは重要ではないということです。なぜかというと、データが埋め込まれている部分空間に対して間違って平行にデータ空間を区切ってしまうようなことさえなければ、それで良いからです。Power Method の収束が遅いのはなぜかというと、第一固有成分と第二固有成分の固有値が同じぐらいのときに、第一固有成分と第二固有成分の間で収束値が非常にぶれてしまうからです。これは、この手法の 1 つの欠点なのですけれども、今やりたいことは、真の第一固有方向を求めることではなくて、データが存在している領域を区切ることです。第一方向であろうと第二方向であろうと、これらデータが存在する方向を表しているベクトルに垂直にデータを区切ることができれば、目的は果たせているのです。



(2-29)

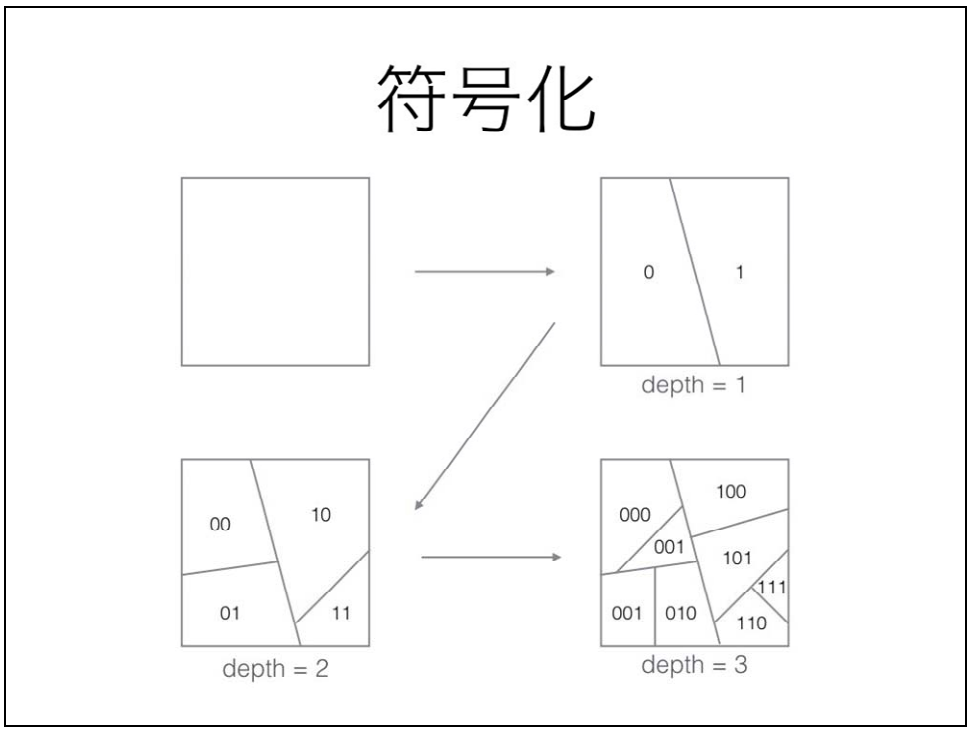
それをもっと逆手に取った方法があって、それは APD-Tree と呼ばれる方法です。これは PCA-Tree の亜種なのですけれども、Power Method を収束する程にまでイテレーションを実行する必要はないという手法です。もし 1 回もイテレーションしなければ、それはランダムプロジェクションであり、無限回やったらそれは PCA なのですけれども、実は、何か 2、3 回イテレーションするだけで、かなり PCA に近いような成績が出せるという論文が、ICML の 2012 年の論文です。実は、全然収束までイテレーションをやらなくても、PCA を実施した場合と同じぐらいの成績が出せるという報告があります。



(2-30)

今回は、どのようなことをやったかというと、元々与えられたデータは83次元あったわけですけれども、

このデータに対して線型回帰（ロジスティック回帰）を行うということをベースラインの方法と定義しました。提案手法が、この成績をどれくらい超えられるのかという話です。提案手法は、密データである P から始まる 21 次元を、APD-Tree を使って、データの量子化(vector quantization; VQ)を実施して、密データだった情報を疎データに変換しました。この操作により特徴量を 100 から 200 次元ぐらい増やして、その下で線型回帰（ロジスティック回帰）を実施するというものです。



(2-31)

なぜ APD-Tree を実施すると、それによって疎データが作れるのかというと、先ほども説明しましたけれども、クラスタリングと量子化と符号化という操作はほとんど一緒で、量子化すると、そこに符号を割り振ることができます。まず、最初の分割で空間が領域「0」と領域「1」とに分割され、2 回目の分割で領域「0」が領域「00」と領域「01」とに分割され、領域「1」が領域「10」と領域「11」とに分割され、このような分割を繰り返し、データ空間を n 回区切ると、各セグメントに対して n ビットの 2 進数ラベルを割り振ることができるので、これを形式変数だと思いましょうというわけです。

符号を特徴量に添加する

導入される形式変数

	符号	0	1	00	01	10	11	000	001	010	011	100	101	110	111
x1	**010**	1	0	0	1	0	0	0	0	1	0	0	0	0	0
x2	**101**	0	1	0	0	1	0	0	0	0	0	0	1	0	0
x3	**110**	0	1	0	0	0	1	0	0	0	0	0	0	1	0

depth = 1 (2 dim)　depth = 2 (4 dim)　depth = 3 (8 dim)

(2-32)

そうなのですけれども少し工夫があって、例えば単純に n ビットの形式変数を加えるだけだと、木の構造は実は反映できないわけです。n ビットの特徴量しか使わないと、セグメント 1 つ 1 つが特徴量になってしまい、n−1 ビットや n−2 ビットの情報が持っている木構造の幹に関する情報が失われてしまいます。上位ビットの情報を明示的に形式変数として特徴量に加えることにより、木構造を意識した線形回帰（ロジスティック回帰）を実施できます。つまり、今回は例えば 3 ビットあるわけですけれども、2 ビットや 1 ビットの情報も形式変数に入れておくと、そこに正則化が働くので、木の上の方の構造が近ければ、対応するパラメーターの値はより近い値になるように学習が進みます。

符号からカーネル関数を作る

(inspired by) [Scornet 2016], [Davies]

$$K(\mathrm{a},\mathrm{b}) := \sum_{k=1}^{\kappa} w_k K_k(\mathrm{a},\mathrm{b})$$

$$\begin{aligned} K_k(\mathrm{a},\mathrm{b}) &:= \frac{1}{2^k}\frac{1}{2^{2(\kappa-k)}}\mathbb{1}_{a_1\cdots a_k=b_1\cdots b_k} \\ &= \frac{1}{2^{2\kappa-k}}\mathbb{1}_{a_1\cdots a_k=b_1\cdots b_k} \end{aligned}$$

$$\hat{y}(\mathrm{x}) = \sum_{a_1=0,1}\cdots\sum_{a_\kappa=0,1}\alpha_{a_1\cdots a_\kappa}K_{\mathrm{a}}(\mathrm{x})$$

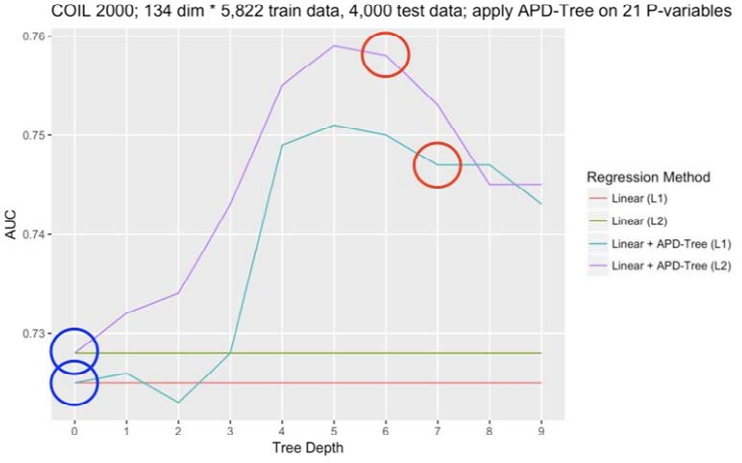
$$K_{\mathrm{a}}(\mathrm{x}) := 2^{\kappa}K(\mathrm{a},\mathrm{x})$$

(2-33)

以前発表したときは、何か統計学の世界では、そのようなことはずっと前から分かっていたのだという指摘を受けたのですけれども、一応、木があるとカーネル関数が作られるのですけれども、その木を直接使って回帰するのと、木から作られるカーネル関数を使って回帰を行うのとでは、パフォーマンスはほぼ同じだという話があって、いつ頃分かったことか分からないですが、そのようなことが分かっています。今回は木があるので、その木から作られるカーネル関数をまず頭の中で作って、それに対応する形式変数を入れて、線型回帰を実施しました。

Regression method & Regularization			#Variables	Tree depth	AUC	Score
(Baseline)	Linear	L1	134	-	0.725	115
(Baseline)	Linear	L2	134	-	0.728	112
(Proposal)	Linear + APD-Tree	L1	346	7	0.747	121
(Proposal)	Linear + APD-Tree	L2	258	6	0.758	125

Table 2.1: Performance of the baseline and the proposed regression methods



(2-34)

どれくらいパフォーマンスがよくなりましたかというと、まずグラフの一番下にあるものが L1 と L2、つ

まり Lasso と Ridge の線型回帰（ロジスティック回帰）です。この場合は L2 の方が成績がよかったです。このベースラインの手法で 112 か 115 ぐらいの得点が出せるので、実は、この最もナイーブな手法でも 17 年前に CoIL Challenge に参加していれば 3 位入賞ぐらいまでは行くという、少し衝撃的な結果です。今回は、さらにそこから工夫をして、APD-Tree を使った特徴量を加えて線型回帰を行うと、どれくらい成績がよくなるのかという話になります。もちろん木の深さを深くするほど、どんどんその説明力は上がっていくわけだけれども、あまり深くしすぎると、今度は過学習してしまう。もちろん正則化項は入れているけれども、それでもやはり過学習してしまって、少し成績が落ちてしまいます。

木の深さはどうやって決めますかというと、クロスバリデーションで決めましょう。だから、クロスバリデーションで決めると、赤丸の辺りになります。本当はもう少し手前の方がよかったのだけれども、クロスバリデーションを使っても、若干オーバーフィットしました。もしかすると工夫次第では、もう少し相応しいところの木の深さも学べたかもしれないけれども、とにかく今回は、木の深さもクロスバリデーションで決めまして、これでもかなり成績がよくなりました。

AUC も 2 ポイント上がっているので、これは機械学習をやっている人の感覚からすると、相当上がったなという印象です。もちろんスコアも 120 台に上げているので、これはもちろん 17 年前の大会に参加していれば断トツ 1 位だったという訳です。今、全く同じコンペティションが開催された場合はどうでしょうか。この方法を使ってコンペに参加しても、勝てる保証は全くないですけれども、一応、最低限の得点は出せているのではないかと思っています。

問題は、なぜこの方法を用いて成績がよくなったのかということです。それをこれから考えていきましょう。今までは機械学習の話をしていましたけれども、ここからはデータ分析の話をしていきましょう。ここまでの説明は、自分として色々とやってみた結果、P から始まる変数（P-変数）を量子化すると最も成績がよくなったという話だったのですけれども、次は、なぜ P-変数を量子化すると、成績が上がるのかというところを考えていきましょう。

- CARAVAN (移動住宅車両の保険件数) を識別する上で重要な特徴量
  - 参考：回帰木 (XGBoost) を用いて算出
  - **PPERSAUT**：自動車保険の支払保険料
  - **PBRAND**：火災保険の支払保険料

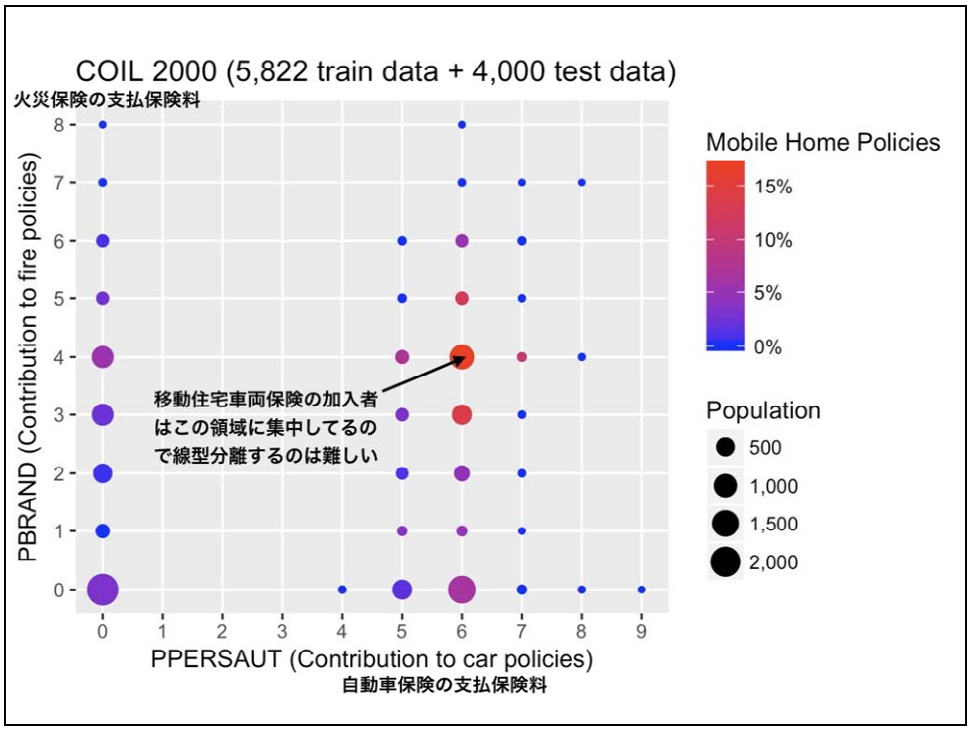
  ⟩ **P-variables**
  - MINK30：年収（低年収ではないかどうか）
  - ABROM：原動機付自転車の保険件数
  - MBERMIDD：中間管理職かどうか
  - MINKGEM：年収（平均的な年収）
  - MOPLLAAG：教育水準（低教育水準ではないかどうか）
- 移動住宅車両の保持の有無により、自動車保険及び火災保険に代表される損害保険の支払保険料の水準は異なるのだが、必ずしも関係性が線型ではなかったのではないか
  - 密データ P-variables を量子化して識別精度が向上したのではないか

(2-35)

実は、別の方法を使って少し分析してみたのですけれども、回帰木を使う方法で XGBoost という手法が

あります。この回帰木を使うと、実は自動車保険の支払保険料と火災保険の支払保険料とが、非常に特徴量として大事だということが分かります。これは、両方とも P-変数で、とても大事なのです。ところがロジスティック回帰をやると、結構この辺の係数が、それほど大きな値にならないわけです。それはなぜなのだろうかというところを考えていく。もしかすると、この変数に関して目的変数を線型分離ではないのではないか、だから P-変数について量子化すると、成績がよくなったのではないかということを疑うわけです。

その話は後ですることにして、他にどのような変数が利いているかというと、例えば年収や教育水準、他の保険の件数などというわけです。移動住宅車両の保険を買うか買わないかということが求めたいことなのですけれども、この変数の水準により、自動車保険や火災保険に代表されるような損害保険の保険料の水準は、かなり違うわけです。かなり違うということが、回帰木を使うと分かるわけだけれども、もしかして、その関係性が線型ではないのではないのか。だから、ロジスティック回帰では、このような事実を上手く拾えなかったのではないかということを考えます。密データを量子化して、このような非線型の効果を量子化することによって取り出せたから、その識別制度は上がったのではないかと疑う訳です。そうすると、次に何をするかというと、この二つの変数についてプロットすることが多分次にやることで、それがこのスライドになります。



(2-36)

色が赤ければ赤いほど CARAVAN 保険（Mobile Home Policy）に加入しています。つまりこれが赤ければ赤いほど目的変数が「1」であり、青ければ青いほど「0」だということです。その上で、赤い箇所を頑張って探せというのがコンペの出題なわけです。

横軸は自動車保険の支払保険料で、縦軸は火災保険の支払保険料なのですけれども、自動車保険というものは単価が高いので、単価が低いところにあまりデータがなくて、自動車を持っていない、自動車を持っていても、自動車保険に入っていないなどの人もいます。それで自動車保険のレンジというものがあるのです。火災保険の方は、詳しいことは分からないが、どうやら低く払うとか高く払うなどができるようであり、いろいろなレンジにあるわけです。

赤いところと青いところが、結構、はっきり違うということが分かります。実は、PRERSAUT = 6・PBRAND = 4 にとても集中している。この赤さは大体「1」の割合が 15%ぐらいなのですけれども、この 15%という値は非常に目安の値なのです。なぜかというと、800 個提出して 120 個が「1」だと優勝なのですが、120/800 がまさに 15%ですので、15%以上の領域を見付けられるととてもうれしくて、そこを提出すればいいわけです。まさに PRERSAUT = 6・PBRAND = 4 が 15%ぐらいの領域で、この領域に含まれているレコードを全部合わせても 800 レコードには至らないので、ここだけを提出するだけでは勝てないのですけれども、とにかく、この 2 次元のプロットで 15%以上の領域が数百件単位で求められるということは、とても意義深いことなのです。

よく見ると、例えば自動車保険料の支払保険料が増えれば増えるほど、例えば CARAVAN 保険に入るかというと、別にそのようなわけではないです。逆に、低ければ低いほど入るかというと、そのようなわけでもないです。なぜか、PRERSAUT = 6・PBRAND = 4 の領域の人だけは、非常に入っているわけです。例えば、この 2 次元だけでロジスティック回帰などをやってしまうと、全然この領域を取り出せないわけです。もちろん、このプロットだけで、ロジスティック回帰ではだめだということを帰結することはできないです。なぜなら、第 3 の特徴量があって、実は、この 2 次元のプロットでは、線型分離できないかもしれないけれども、他の特徴量、もしくは他の特徴量の組み合わせによって、何かうまくここの部分だけを切り出せる可能性はあるので、このプロットだけをもって、ロジスティック回帰では無理だということは言えないです。ただ、このようなプロットを見ていると、これ程の CARAVAN 保険（Mobile Home Policy）に入るか入らないかに関して情報を持っている 2 次元の非線型の関係があるのだったら、この情報を上手く切り出して新たな特徴量を作りたくなる訳です。このような非線形の関係性に一旦気がついてしまうと、例えば PRERSAUT = 6・PBRAND = 4 を中心とする、何かガウシアンカーネルを置いて、それが生成する特徴量を加えるなど、例えば思いつくかもしれません。しかし、実際には、なかなかこういった関係性を人力で探すことは難しくて、よほど暇だったらやってもいい訳ですが、実際には、なかなかそのようなことに時間をかけられないのだとすると、今回やったような、何か自動的に量子化してしまって、何が何だか分からないけれども、それで線型回帰をしてみるということも、一つのやり方なのではないかと思っています。

(2-37)

私の発表は、これで以上です。

【司会】　まだ若干時間がありますので、会場からの質問を受け付けたいと思います。質問のある方は、挙手をお願いします。

【宮崎】　どうも、プレゼン大変参考になりました。ありがとうございます。マニュライフ生命の宮崎と申します。

小田さんにお伺いしたいのですが、最初の方に、機械学習では、ソフトクラスタリングの方がはやっているようなことをおっしゃられたのですが、今回の木構造を作って分割していくことは、ハードクラスタリングですね、基本的には。クラスタリングと言っていいのかどうか分からないのですけれども、これをもしソフトでやった場合は、点数が上がるなど、そのようなことはありますか。

【小田】　まず、ソフトとハードの質問者の定義を、少し教えてもらっていいですか。

【宮崎】　いや、何となくニュアンス的に、あまりこの辺は詳しくないのですけれども。

【小田】　少しどのような質問の意図か、分からない部分もあるのですけれども、例えば遠藤さんが発表されたような階層的なクラスタリング手法を用いて、同じ分析を実行すると計算に時間がかかり過ぎる可能性があります。

データの件数が 1 万件以上あり、それを n^2 の計算量のアルゴリズムで実行するということは、データが増えるに連れてどんどん時間がかかっていくので、それは非現実的かと思っています。

しかも、そのようなやり方というものは、結構、一つ一つのデータにとても依存してしまうので、オーバーフィッティングしやすいですね。

一方、今回のような第一固有成分というものは、例えばデータが 1 個抜けたり、データが新しく 1 個入ってきたりしても、大まかな方向なので、そのようなものはデータが 1 個、2 個変わったとしてもほぼ変わらないので、よりロバストな方向だと思いますし、今回のように第一固有成分を求めるだけだったら、非常に計算コストが低く求められるので、そのような観点でも、現実的な解法かと思っています。

【宮崎】　ありがとうございました。

【元村】　住友生命の元村と申します。本日は貴重なお話をありがとうございました。

遠藤さんに 1 点伺いたいのですけれども、今回、遠藤さんにやっていただいたクラスタリングの手法というものは、教師なし学習の方に分類されるのかなというように聞いていて思っていて、この辺詳しくはないのですけれども、その中で、スライド(1-50)のところで、「採用したクラスタリングの結果」ということで、6 個のクラスターに分けていただいて、この 6 個のクラスターのうち、4 番と 5 番というものが CARAVAN に入っている可能性が高いですというようなお話だったかと思うのです。少しそこは何か説明が逆といいますか、教師なし学習なので、この訓練データの中で、CARAVAN の加入率が、ここは高いなどということを見てしまうのは、本当はよくないのではないのかと思いました。

4 番や 5 番の人に、トレーラーの保険を売ったらいいのではないかということが導かれますというお話だったかと思うのですけれども、それはどちらかというと、クラスタリングをしたうえで、次の主成分分析をしたうえで、その主成分を見て、この人の方が高そうだというようなことが、定性的に見た結果、この人たちに売ったらいいのではないかということが、教師なし学習というものの本質なのではないのかと思ったのですけれども、そこはいかがでしょうか。

【遠藤】　クラスタリングのポイントは、データを一度分けてしまう点だと思います。今回の次元は、86 次元ととても低いです。しかし、小田さんなどが業務でやられているデータは、数百万次元や何億次元など、そのようなデータを相手にしているので、まず分けてしまうということすらできないわけです。ただ、クラスタリングをしてしまえば、いったん分けることができます。8 個なら 8 個、10 個なら 10 個と、分けたうえでその分かれたクラスターに対して PC などの特徴量を見始めることができるのです。今回の分析であれば、PC だけ見ればいいのではないかという指摘はもっともだと思います。ただ、実務で考えると、クラスタリングなどの教師なし学習を用いることで、データに関する知見を多く持たなくとも、まずスタートラインとしての結果が得られるという点が、こういった手法の意義なのではないかと考えます。

【元村】　ありがとうございました。

【司会】　まだまだ質問を受け付けたいところではありますが、終了時間となっておりますので、以上をもちまして、「プレディクティブモデリングの保険データへの応用（ASTIN COLLOQUIUM 2017 参加報告)」を終了します。発表していただいたお 2 人に、いま一度大きな拍手をお願いします。