2019年度「専修学校による地域産業中核的人材養成事業」

データマイニング教材

Society5.0実現のためのIT技術者養成モデルカリキュラム開発と実証事業

2019年度「専修学校による地域産業中核的人材養成事業」

データマイニング教材

Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

目 次

第1回：データマイニングの概要

アジェンダ

- データマイニングとは
- データマイニングの手法の分類
  - クラス分類
  - 回帰
  - クラスタリング
  - パターン抽出
  - その他の手法
- データマイニングの歴史と発展

全15回の講義について

- データマイニングの各種アルゴリズムの理解を目標とする。
  - プログラミング言語としてはPython
  - Pythonの各種ライブラリを利用してデータ分析に必要なスキルの習得を目指す
  - 第2回以降の講義で詳細を取り扱う

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

事例：マーケット・バスケット分析

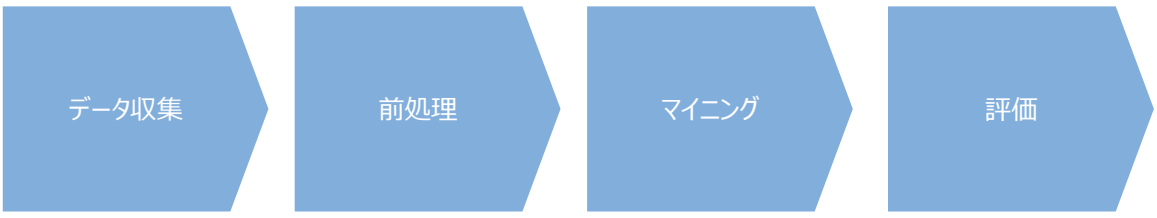
- データマイニングで有名な事例として、マーケット・バスケット分析があります。
- マーケット・バスケット分析とは、データ同士の関係性を分析するもので、どの商品とどの商品をどのような顧客が同時に購入したかを分析する手法です。

- 夕刻、紙おむつとビールが同時に購入される、という有名な事例がアメリカにあります。夕食の準備に忙しい母親に言われて商店に紙おむつを買いに来た父親が、自分へのご褒美にビールを買うため、と解釈されています。

データマイニングのプロセス

- データマイニングを行うために、まずはデータを収集することが必要です。一般的には、元となるデータが多ければ多いほど、有益な情報を採掘（マイニング）できる可能性が高まります。

- 収集されたデータは、データマイニングの各種アルゴリズムに適した形式に変換する「前処理」が施されます。



データマイニングのステップ

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

データマイニングの歴史

年号	できごと
1960年代	メインフレームが金融企業の基幹業務システムとして稼働開始した。同時に、デジタルデータの収集,蓄積,利用の試みが開始された。
1970年代	1971年から1973年にかけて、チリでサイバーシン計画が実行される。コントロールセンターが、テレックスを介して実時間でチリ各地に点在する工場からデータを収集して、収集したデータを元に、オペレーションズ・リサーチを用いて最適化した生産計画を作成し、工場に対して生産計画をフィードバックするシステムであった。
1980年代	現在の"Data mining"の定義と類似する"Knowledge Discovery in Databases"という語が出現する。関係データベースシステムとその操作用言語であるSQLが出現する。データウェアハウスの運用が開始される。
1990年代	1990年頃から始まった計算機の急激な性能向上により"Knowledge Discovery in Databases"の研究が大幅に加速される。 1999年 - 2010年代に大量の実世界データを収集・供給する基盤となるInternet of Things(IoT)の用語がKevin Ashtonにより初めて使用された。(この当時のIoTは、様々な物体にRFIDタグを貼り付け、RFIDに対応したセンサーを用いて物体からの情報収集を行い、収集した情報を活用することを指していた)
2000年代	インターネットへの常時接続が一般家庭にも普及する。インターネット上に蓄積されたデータが加速度的に増加する。後にデータの主要な供給源の1つとなる友人紹介型のソーシャル・ネットワーキング・サービスが2002年より相次いで提供され始める。コンピュータとインターネットの普及に着目し、ビジネスにおいて膨大に蓄積され活用しきれなくなったデータの分析を専門に行う企業も徐々に出現し始める。
2010年代	英国"The Economist"誌において"big data"の語が提唱された。コモディティ化によりコンピュータの計算能力が安価になり、高速データ処理用のコンピュータ・クラスタの構築が容易にできるようになった。データ分析のコストが下がり、ビッグデータ解析の応用が進むようになった。データサイエンティストという名称の職業が台頭し始めた。また、ビッグデータを用いたデータマイニングを応用したサービスが一般向けにも提供され始めた。コグニティブ・コンピューティング・システムが商用で実用化された。テレビ番組の紹介コーナーでも、インターネット上に存在するビッグデータの統計分析結果を元に流行のトレンドを紹介するようになった。 ディープラーニングの実用化が急速に進み、非常に多数の人工知能サービスが現れた。

出展： https://ja.wikipedia.org/wiki/データマイニング

データマイニングに用いられるツール・ライブラリ
R言語

Wikipediaより（https://ja.wikipedia.org/wiki/R言語）

- R言語（あーるげんご）はオープンソース・フリーソフトウェアの統計解析向けのプログラミング言語及びその開発実行環境である。ファイル名拡張子は.r, .R, .RData, .rds, .rda。
- R言語はニュージーランドのオークランド大学のRoss IhakaとRobert Clifford Gentlemanにより作られた。現在ではR Development Core Team[注 1] によりメンテナンスと拡張がなされている。

データマイニングに用いられるツール・ライブラリ
Python + Jupyter notebook + scikit-learnなど

Wikipediaより（https://ja.wikipedia.org/wiki/Scikit-learn）

- scikit-learn (旧称：scikits.learn) はPythonのオープンソース機械学習ライブラリ[2]である。サポートベクターマシン、ランダムフォレスト、Gradient Boosting、k近傍法、DBSCANなどを含む様々な分類、回帰、クラスタリングアルゴリズムを備えており、Pythonの数値計算ライブラリのNumPyとSciPyとやり取りするよう設計されている。

データマイニングに用いられるツール・ライブラリ
Weka

Wikipediaより（https://ja.wikipedia.org/wiki/Weka）

- Weka (Waikato Environment for Knowledge Analysis) は、ニュージーランドのワイカト大学で開発した機械学習ソフトウェアで、Javaで書かれている。GNU General Public License でライセンスされているフリーソフトウェアである。

第2回：単純ベイズ分類器

アジェンダ

- 前回の振り返り
- 単純ベイズ分類器による発話の分類

前回の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

単純ベイズ分類器による発話の分類

単純ベイズ分類器とは

- ベイズの定理を利用した単純な確率的分類器のことです。
- あるデータがどのカテゴリーに属するのかを判定させる機械学習の教師あり学習の手法の一つであり、スパムメールフィルターやWEBニュース記事のカテゴライズによく使われています。

ベイズの定理

- 下の式をベイズの定理といいます。
- P(Y) は事前分布と呼ばれ、P(X|Y) は尤度（条件付き確率）と呼ばれます。

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)}$$

- 例えば、
  X=受信メール, Y={迷惑メールである, 迷惑メールでない}
  という事例のように、Xには何らかの起こった事象や入力データを、Yにはそこから推論したい事柄などを当てはめます。
- ここで分母はYに依存しておらず分母が実質的に一定であるようにXが与えられるため、分子だけを考慮して下記のように取り扱います。

$$P(Y|X) \propto P(Y)P(X|Y)$$

単純ベイズ

- 下の式の尤度P(X|Y)の部分に「条件付き独立性」の仮定を置き、各特徴変数Yiが条件付きで他の特徴変数Yjと独立であるとします。

$$P(Y|X) \propto P(Y)P(X|Y)$$

- すると尤度P(X|Y)の部分は下記のように簡略されるため、単純ベイズと呼ばれます。

$$P(X|Y) = \prod_{i=1}^{N} P(x_i|Y)$$

演習：Naive Bayes（単純ベイズ）による対話分類モデルの構築

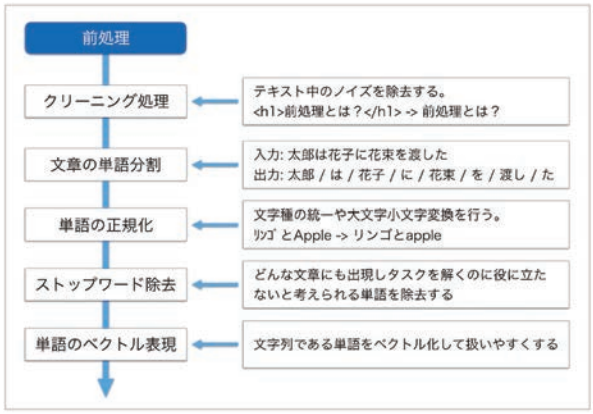
- 人間の発話をカテゴリごとに分類するモデルを単純ベイズで作成します。

演習1：独自データ収集

- 下記の問いかけの発話事例を記載してください。
  class_0: 名前を聞かれる
  class_1: 好きな色を聞かれる
  class_2: 好きな食べ物を聞かれる
  class_3: 年齢を聞かれる
  class_4: 挨拶される

自然言語処理における前処理

- コンピュータは自然言語そのものを理解できないため、コンピュータが理解できるデータに自然言語を変換する必要があります。
- 自然言語処理においてモデルを作成するときは、単語や文章をベクトル（多次元の要素を持つ量）に変換し、コンピュータで処理します。



出展： https://qiita.com/Hironsan/items/2466fe0f344115aff177

文章のベクトル化：Bag of Words

- ベクトル表現の一種で、文章に単語が含まれるかどうかのみを考え、単語の並び方などは考慮しない形式のことです。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ て いる ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

分類に寄与する文言

- 発話の種類ごとに、分類に寄与しそうな文言を入れるとモデルの精度が向上します。
- 例えば「class_0: 名前を聞かれる」の場合だと、"名前"という文言のことです。
- 「あなた」や「教えて」などの文言はどのクラスにも登場する可能性があり、発話の分類には寄与しないと考えられます。

```
sentences = [
    "名前は何",
    "名前をなんていうの",
    "名前教えて",
    "あなたのお名前は",
    "お名前教えてよ",

    "どんな色が好きなの",
    "何色が好き",
    "好きな色は何",
    "黄色は好き",
    "好きな色を教えてちょうだい",

    "どんな食べものが好きなの",
    "ピーマンは食べれる",
    "好きな食べものは",
    "食べものは何が好きなの",
    "何が美味しい",

    "歳はいくつですか",
    "歳はいくつになった",
    "何歳なの",
    "何歳か教えて",
    "何歳ですか",

    "おはよう",
    "おはようございます",
    "こんにちは",
    "こんばんは",
    "おやすみなさい",
]
```

演習2：BoWの実装

- 文章をベクトル化するプログラムを実装してください。

演習3：単純ベイズによる分類モデル

- ベクトル化したデータで単純ベイズのモデルを作成し、モデルの精度を確認してください。

単純ベイズによる分類結果

- 下記の表は単純ベイズによるモデルの分類結果の一例です。
- 「Class_2：好きな食べ物を聞かれる」の誤認識が多いことがわかります。
- 次回の講義では、モデル精度を向上させるための工夫について学んでいきます。

テストデータに対する正解率: 0.75

predict	0	2	3	4
class				
0	1	0	0	0
1	0	1	0	0
2	0	1	0	0
3	0	1	2	0
4	0	0	0	2

コーパスの充実

- 学習データに表記揺れが含まれた方が、頑健なモデルが作成できる場合があります（※逆の発想で、ベクトル化の際にこのような表記揺れを落としてしまう手法もあります）。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ て いる ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

演習4：独自データの拡充

- データ数を増やして、モデルの精度の変化を確認してください。

```
sentences = [
    "名前は何",
    "名前は何ですか",
    "名前を何ていうの",
    "何ていうの名前は",
    "何ですか名前は",
    "名前を何ていうの",
    "あなたのお名前は",
    "お名前はあなたの",
    "お名前教えて",
    "お名前教えてよ",
```

表現の揺れを足してみる

単純ベイズによる分類結果

- 単純ベイズでモデルを作成し、分類結果を考察しました。
- 全体の精度は向上し「Class_2：好きな食べ物を聞かれる」の誤認識も改善しました。

テストデータに対する正解率: 0.75

predict	0	2	3	4
class				
0	1	0	0	0
1	0	1	0	0
2	0	1	0	0
3	0	1	2	0
4	0	0	0	2

テストデータに対する正解率: 0.93

predict	0	1	2	3	4
class					
0	2	0	0	0	0
1	0	3	0	0	0
2	0	0	4	0	0
3	0	0	1	4	0
4	0	0	0	0	1

第3回：決定木

アジェンダ

- 第1回講義の振り返り
- 決定木による発話の分類

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

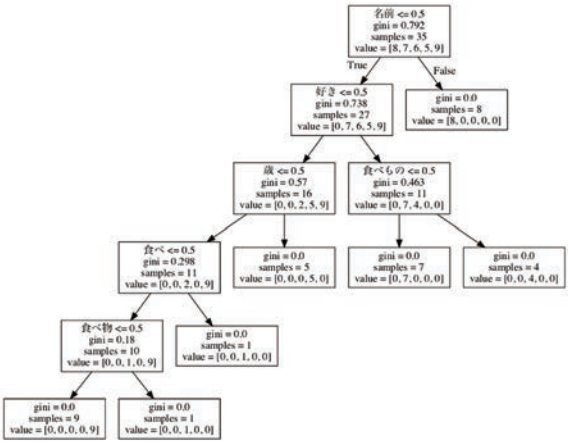
データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

決定木による発話の分類

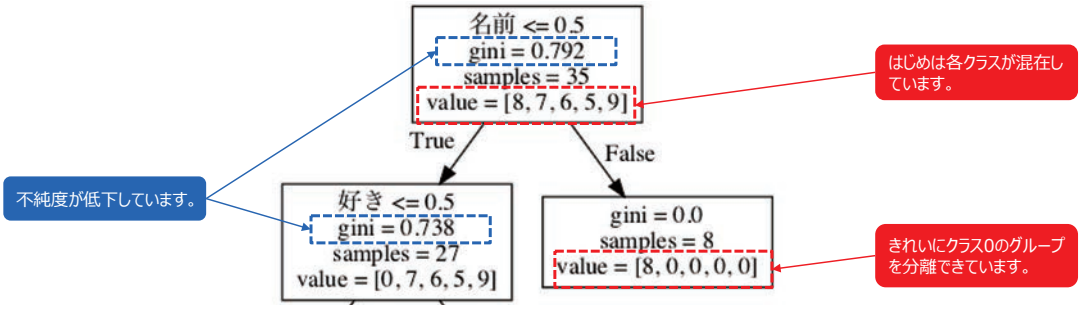
決定木とは

- 決定木分析は回帰や分類を実行する手法です。
- 顧客情報やアンケート結果などの目的変数に寄与する説明変数を見つけ、樹木状のモデルを作成します。



決定木の分岐の指標

- 決定木においてデータを分割する基準は「情報利得」と「不純度」です。
- 情報利得とは、いかにうまく分割できたか（分割前の不純度 - 分割後の不純度）を表します。
- 不純度とは、分割後のグループにおける、複数のクラスの混在の度合い（どれだけごちゃごちゃしているか）を表します。



演習：Decision Tree（決定木）による対話分類モデルの構築

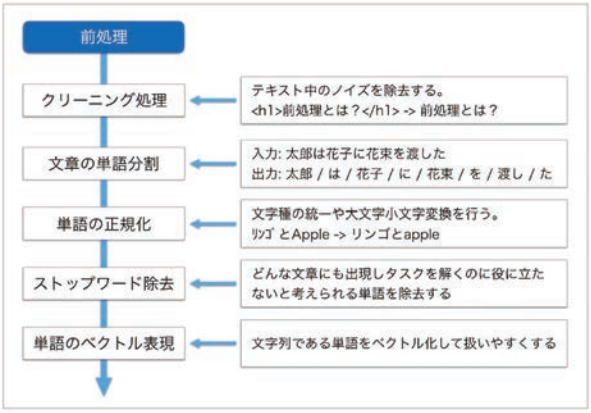
- 人間の発話をカテゴリごとに分類するモデルを決定木で作成します。

演習1：独自データ収集

- 下記の問いかけの発話事例を記載してください。
  class_0: 名前を聞かれる
  class_1: 好きな色を聞かれる
  class_2: 好きな食べ物を聞かれる
  class_3: 年齢を聞かれる
  class_4: 挨拶される

自然言語処理における前処理

- コンピュータは自然言語そのものを理解できないため、コンピュータが理解できるデータに自然言語を変換する必要があります。
- 自然言語処理においてモデルを作成するときは、単語や文章をベクトル（多次元の要素を持つ量）に変換し、コンピュータで処理します。



出展： https://qiita.com/Hironsan/items/2466fe0f344115aff177

文章のベクトル化：Bag of Words

- ベクトル表現の一種で、文章に単語が含まれるかどうかのみを考え、単語の並び方などは考慮しない形式のことです。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ て いる ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

分類に寄与する文言

- 発話の種類ごとに、分類に寄与しそうな文言を入れるとモデルの精度が向上します。
- 例えば「class_0: 名前を聞かれる」の場合だと、"名前"という文言のことです。
- 「あなた」や「教えて」などの文言はどのクラスにも登場する可能性があり、発話の分類には寄与しないと考えられます。

```
sentences = [
    "名前は何",
    "名前をなんていうの",
    "名前教えて",
    "あなたのお名前は",
    "お名前教えてよ",

    "どんな色が好きなの",
    "何色が好き",
    "好きな色は何",
    "黄色は好き",
    "好きな色を教えてちょうだい",

    "どんな食べものが好きなの",
    "ピーマンは食べれる",
    "好きな食べものは",
    "食べものは何が好きなの",
    "何が美味しい",

    "歳はいくつですか",
    "歳はいくつになった",
    "何歳なの",
    "何歳か教えて",
    "何歳ですか",

    "おはよう",
    "おはようございます",
    "こんにちは",
    "こんばんは",
    "おやすみなさい",
]
```

演習2：BoWの実装

- 文章をベクトル化するプログラムを実装してください。

演習3：決定木による分類モデル

- ベクトル化したデータで決定木モデルを作成し、モデルの精度を確認してください。

決定木による分類結果

- 下記の表は決定木によるモデルの分類結果の一例です。
- 「Class_2：好きな食べ物を聞かれる」の誤認識が多いことがわかります。
- 次回の講義では、モデル精度を向上させるための工夫について学んでいきます。

テストデータに対する正解率: 0.88

predict	0	1	3	4
class				
0	1	0	0	0
1	0	1	0	0
2	0	0	0	1
3	0	0	3	0
4	0	0	0	2

コーパスの充実

- 学習データに表記揺れが含まれた方が、頑健なモデルが作成できる場合があります（※逆の発想で、ベクトル化の際にこのような表記揺れを落としてしまう手法もあります）。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ ている ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

演習4：独自データの拡充

- データ数を増やして、モデルの精度の変化を確認してください。

```
sentences = [
    "名前は何",
    "名前は何ですか",
    "名前を何ていうの",
    "何ていうの名前は",
    "何ですか名前は",
    "名前を何ていうの",
    "あなたのお名前は",
    "お名前はあなたの",
    "お名前教えて",
    "お名前教えてよ",
```

表現の揺れを足してみる

決定木による分類結果

- 決定木でモデルを作成し、分類結果を考察しました。
- 全体の精度は向上し「Class_2：好きな食べ物を聞かれる」の誤認識も改善しました。

テストデータに対する正解率: 0.88

predict	0	1	3	4
class				
0	1	0	0	0
1	0	1	0	0
2	0	0	0	1
3	0	0	3	0
4	0	0	0	2

テストデータに対する正解率: 0.93

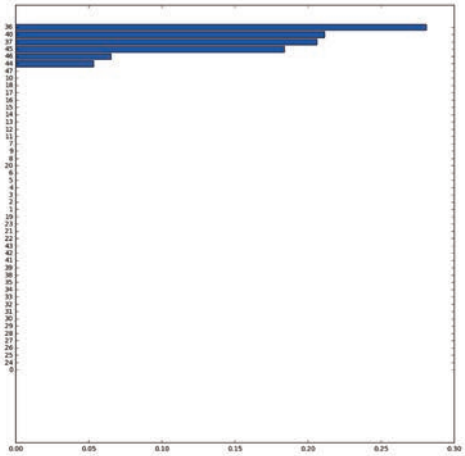
predict	0	1	2	3	4
class					
0	2	0	0	0	0
1	0	3	0	0	0
2	0	0	3	0	1
3	0	0	0	5	0
4	0	0	0	0	1

演習5：項目の寄与度

- 決定木モデルの判定に寄与した項目を確認してください。

判定に寄与した項目の確認

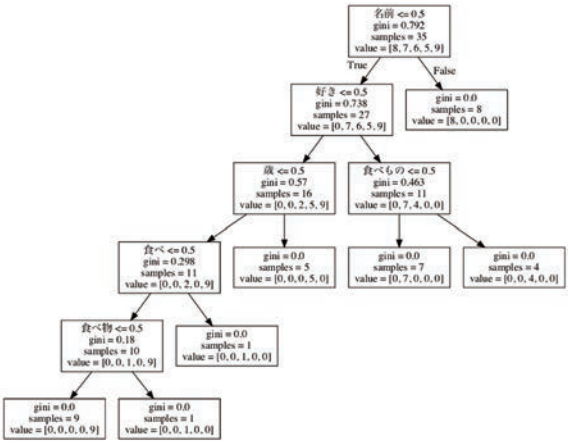
- 決定木における判定に寄与した項目を数値化した結果です。赤字の項目のみが判定に寄与し、その他の項目は判定に寄与していないことが確認できます。



{'あなた': 0, 'いくつ': 1, 'お': 2, 'おはよう': 3, 'おやすみ': 4, 'おやすみなさい': 5, 'か': 6, 'が': 7, 'こんち': 8, 'こんにちは': 9, 'こんばんは': 10, 'ござい': 11, 'さようなら': 12, 'す': 13, 'た': 14, 'だい': 15, 'ちょうだい': 16, 'て': 17, 'ていう': 18, 'です': 19, 'と': 20, 'どんな': 21, 'な': 22, 'なっ': 23, 'なに': 24, 'に': 25, 'の': 26, 'は': 27, 'ます': 28, 'よ': 29, 'れる': 30, 'を': 31, 'セロリ': 32, 'ピーマン': 33, 'ー': 34, '何': 35, '名前': 36, '好き': 37, '思う': 38, '教え': 39, '歳': 40, '美味しい': 41, '色': 42, '赤色': 43, '食べ': 44, '食べもの': 45, '食べ物': 46, '黄色': 47}

判定に寄与した項目の確認

- 寄与度の大きな項目で分岐していることが確認できます。



第4回：サポートベクターマシン

アジェンダ

- 第1回講義の振り返り
- サポートベクターマシンによる発話の分類

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

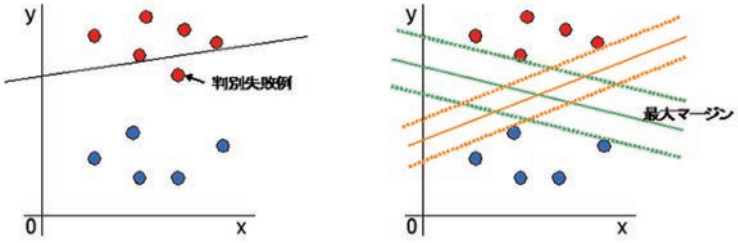
データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

サポートベクターマシンによる発話の分類

サポートベクターマシンとは

- サポートベクターマシン（SVM）はパターン識別用の教師あり機械学習方法であり、局所解収束の問題を軽減できるという長所があります。
- 「マージン最大化」というアイデア等で汎化能力も高め、非常に優秀なパターン識別能力を持つとされています。
- SVMでは、右図のように、2つのグループ間の最も距離の離れた箇所（最大マージン）を見つけ出し、その真ん中に識別の線を引くことによって、多くの未学習データの判別が可能になる事を「汎化能力」の向上を狙います。

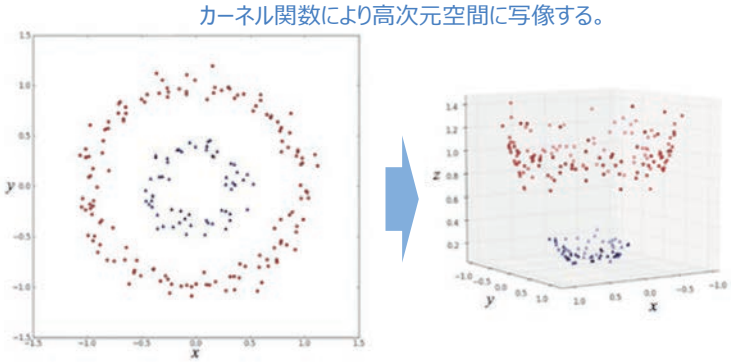


出展：http://www.sist.ac.jp/~kanakubo/research/neuro/supportvectormachine.html

線形分離不可能な問題への適用

- カーネル・トリックという方法で、線形分離不可能な問題に関してもSVMが適用可能になりました。
- カーネルトリックとは元々のデータ空間から高次元空間にデータを写像することです。そうすることで、高次元空間上で線形データ解析を行うことが可能になります。

カーネル関数により高次元空間に写像する。



出展：http://enakai00.hatenablog.com/entry/2017/10/13/145337

演習：Support Vector Machine（SVM）による対話分類モデルの構築

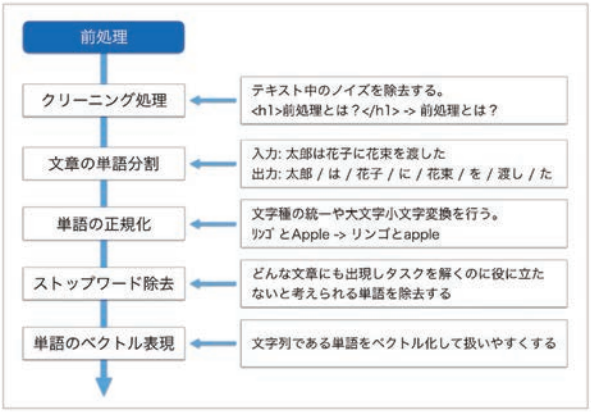
- 人間の発話をカテゴリごとに分類するモデルをSVMで作成します。

演習1：独自データ収集

- 下記の問いかけの発話事例を記載してください。
  class_0: 名前を聞かれる
  class_1: 好きな色を聞かれる
  class_2: 好きな食べ物を聞かれる
  class_3: 年齢を聞かれる
  class_4: 挨拶される

自然言語処理における前処理

- コンピュータは自然言語そのものを理解できないため、コンピュータが理解できるデータに自然言語を変換する必要があります。
- 自然言語処理においてモデルを作成するときは、単語や文章をベクトル（多次元の要素を持つ量）に変換し、コンピュータで処理します。



出展： https://qiita.com/Hironsan/items/2466fe0f344115aff177

文章のベクトル化：Bag of Words

- ベクトル表現の一種で、文章に単語が含まれるかどうかのみを考え、単語の並び方などは考慮しない形式のことです。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ て いる ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

分類に寄与する文言

- 発話の種類ごとに、分類に寄与しそうな文言を入れるとモデルの精度が向上します。
- 例えば「class_0: 名前を聞かれる」の場合だと、"名前"という文言のことです。
- 「あなた」や「教えて」などの文言はどのクラスにも登場する可能性があり、発話の分類には寄与しないと考えられます。

```
sentences = [
    "名前は何",
    "名前をなんていうの",
    "名前教えて",
    "あなたのお名前は",
    "お名前教えてよ",

    "どんな色が好きなの",
    "何色が好き",
    "好きな色は何",
    "黄色は好き",
    "好きな色を教えてちょうだい",

    "どんな食べものが好きなの",
    "ピーマンは食べれる",
    "好きな食べものは",
    "食べものは何が好きなの",
    "何が美味しい",

    "歳はいくつですか",
    "歳はいくつになった",
    "何歳なの",
    "何歳か教えて",
    "何歳ですか",

    "おはよう",
    "おはようございます",
    "こんにちは",
    "こんばんは",
    "おやすみなさい",
]
```

演習2：BoWの実装

- 文章をベクトル化するプログラムを実装してください。

演習3：SVMによる分類モデル

- ベクトル化したデータでSVMモデルを作成し、モデルの精度を確認してください。

SVMによる分類結果

- 下記の表はSVMによるモデルの分類結果の一例です。
- 「Class_2：好きな食べ物を聞かれる」の誤認識が多いことがわかります。
- 次回の講義では、モデル精度を向上させるための工夫について学んでいきます。

テストデータに対する正解率: 0.75

predict	0	1	2	3	4
class					
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	0	0	1
3	0	0	1	2	0
4	0	0	0	0	2

コーパスの充実

- 学習データに表記揺れが含まれた方が、頑健なモデルが作成できる場合があります（※逆の発想で、ベクトル化の際にこのような表記揺れを落としてしまう手法もあります）。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ ている ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

演習4：独自データの拡充

- データ数を増やして、モデルの精度の変化を確認してください。

```
sentences = [
    "名前は何",
    "名前は何ですか",
    "名前を何ていうの",
    "何ていうの名前は",
    "何ですか名前は",
    "名前を何ていうの",
    "あなたのお名前は",
    "お名前はあなたの",
    "お名前教えて",
    "お名前教えてよ",
```

表現の揺れを足してみる

SVMによる分類結果

- SVMでモデルを作成し、分類結果を考察しました。
- 全体の精度は向上し「Class_2：好きな食べ物を聞かれる」の誤認識も改善しました。

テストデータに対する正解率: 0.75

predict	0	1	2	3	4
class					
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	0	0	1
3	0	0	1	2	0
4	0	0	0	0	2

テストデータに対する正解率: 0.87

predict	0	1	2	3	4
class					
0	2	0	0	0	0
1	0	3	0	0	0
2	0	0	2	0	2
3	0	0	0	5	0
4	0	0	0	0	1

文章のベクトル化：TF-IDF

- 文章をベクトル化する際に、単語に重みを付けて評価する手法です。単語の出現頻度であるTF（Term Frequency）と、IDF（Inverse Document Frequency）という2つの指標を使用します。
- TF（Term Frequency）は、「各文書においてその単語がどのくらい出現したのか」を意味します。よく出現する単語は、その文章の特徴を捉えるのに有用だろうという考え方です。
- IDF（Inverse Document Frequency）は、単語が稀にしか出現しないなら高い値を、「色々な文書によく出現する単語」なら低い値を示すものです。稀少な単語は、その文書の特徴を捉えるのに有用だろうという考え方です。

$$tf = \frac{\text{文書}A\text{における単語}X\text{の出現頻度}}{\text{文書}A\text{における全単語の出現頻度の和}}$$

$$idf = log(\frac{\text{全文書数}}{\text{単語}X\text{を含む文書数}})$$

$$tfidf = tf * idf$$

演習5：TF-IDFの実装

- 作成したベクトルにTF-IDFを適用し、新たなベクトルを作成してください。
- TF-IDFで作成したベクトルの、スコアの高い形態素の上位N件を表示するプログラムを作成してください。

TF-IDFによる重要単語の判定

- TF-IDFで作成したベクトルの、スコアの高い形態素の上位N件を取得した例です。

```
sentences = [
  "名前は何",
  "名前は何ですか",
  "名前を何ていうの",
  "何ていうの名前は",
  "何ですか名前は",
  "名前を何ていうの",
  "あなたのお名前は",
  "お名前はあなたの",
  "お名前教えて",
  "お名前教えてよ",

  "どんな色が好きなの",
  "好きなのはどんな色",
  "何色が好き",
  "好きなのは何色",
  "好きなのは何色ですか",
  "好きな色は何",
  "黄色は好き",
  "赤色は好き",
  "好きな色を教えてちょうだい",
  "教えて好きな色を",

  "どんな食べものが好きなの",
  "好きなのはどんな食べもの",
  "ピーマンは食べれる",
  "食べれるのかなセロリは",
  "好きな食べものは",
  "食べものは何が好き",
  "食べものは何が好きなの",
  "何が好きなの食べものは",
  "何が美味しい",
  "美味しいと思う食べ物はなに",
  "歳はいくつですか",
  "いくつですか歳は",
  "歳はいくつになった",
  "いくつになったの歳は",
  "何歳なの",
  "何歳ですか",
  "何歳か教えて",
  "教えてよ何歳か",
  "何歳ですか",
  "何歳だい",

  "おはよう",
  "おはようございます",
  "おはようす",
  "おはー",
  "こんにちは",
  "こんちは",
  "こんばんは",
  "さようなら",
  "おやすみなさい",
  "おやすみ",
]
```

```
[1.0, 'こんばんは']
[1.0, 'こんにちは']
[1.0, 'おやすみなさい']
[1.0, 'おはよう']
[0.7820413819688841, '黄色']
[0.7427159045117733, '美味しい']
[0.6555597340352133, '名前']
[0.6257109448664917, '食べもの']
[0.5991717283560805, 'ます']
[0.5991717283560805, 'ござい']
[0.5912077691887732, 'です']
[0.5905831399346878, '教え']
[0.5905831399346878, 'て']
[0.5787358088645832, 'あなた']
[0.569998276611747, '色']
[0.5504073054685807, '食べ']
[0.5504073054685807, 'ピーマン']
[0.5504073054685807, 'れる']
[0.5477859014171177, '何']
[0.5457519094247155, '歳']
```

演習6：TF-IDFでベクトル化したデータによるモデル作成

- TF-IDFで作成したベクトルをSVMに投入してモデルを作成し、モデルの精度を確認してください。

SVMによる分類結果

- TF-IDFにより分類に寄与しない形態素を除去した結果、モデルの分類精度が向上したことが確認できました。

テストデータに対する正解率: 0.75

class \ predict	0	1	2	3	4
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	0	0	1
3	0	0	1	2	0
4	0	0	0	0	2

テストデータに対する正解率: 0.87

class \ predict	0	1	2	3	4
0	2	0	0	0	0
1	0	3	0	0	0
2	0	0	2	0	2
3	0	0	0	5	0
4	0	0	0	0	1

テストデータに対する正解率: 1.00

class \ predict	0	1	2	3	4
0	2	0	0	0	0
1	0	3	0	0	0
2	0	0	4	0	0
3	0	0	0	5	0
4	0	0	0	0	1

第5回：線形回帰

アジェンダ

- 第1回講義の振り返り
- 線形回帰モデルの構築

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

線形回帰モデルの構築

線形回帰とは

- 線形回帰は連続値をとる目的変数yと説明変数x（特徴量）の関係を下記の数式でモデル化します（ $X_0=1$とし、w_0は切片を表します。）。
- 説明変数が一つの場合を単回帰、複数の場合を重回帰といいます。

$$y = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \sum_{i=0}^{m} w_ix_i$$

演習：線形回帰による住宅価格の予測モデルの構築

- ボストン市郊外の地域別住宅価格を予測する線形回帰モデルを作成します。

演習1：データ項目の確認

- ボストン市郊外の地域別住宅価格データの項目を確認してください。

変数	説明
CRIM	町ごとの一人当たりの犯罪率
ZN	25,000平方フィートを超える敷地に区画された宅地の割合
INDUS	非小売業種の土地面積の割合
CHAS	Charles Riverダミー変数（敷地が川の境界にある場合は1、それ以外の場合は0）
NOX	窒素酸化物の濃度（1000万分の1）
RM	1住戸あたりの平均部屋数
AGE	1940年以前に建設された所有者居住ユニットの割合
DIS	ボストンの5つの雇用センターまでの重み付き距離
RAD	ラジアルハイウェイ（放射状に各方面へ伸びる高速道路）へのアクセスのしやすさの指標
TAX	10,10,000ドルあたりの全額固定資産税率
PTRATIO	町による生徒 - 教師比率
B	1000（Bk - 0.63）^2ここでBkは町による黒人の割合
LSTAT	低所得者の割合
MEDV	住宅価格の中央値（1,000単位）

演習2：項目間の相関

- MEDV（住宅価格の中央値）と他の項目の相関を確認してください。

項目間の相関

- MEDV（住宅価格の中央値）とRM（1住戸あたりの平均部屋数）は比較的強い正の相関があることがわかります。
- MEDV（住宅価格の中央値）とLSTAT（低所得者の割合）は比較的強い負の相関があることがわかります。

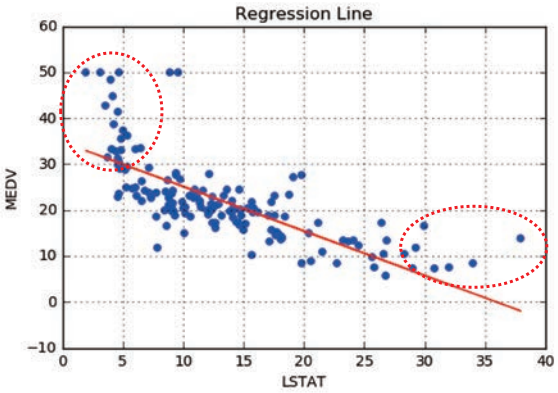
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

演習3：単回帰モデル

- 住宅価格の中央値を低所得者の割合から予測する線形回帰モデルを構築し、精度を確認してください。

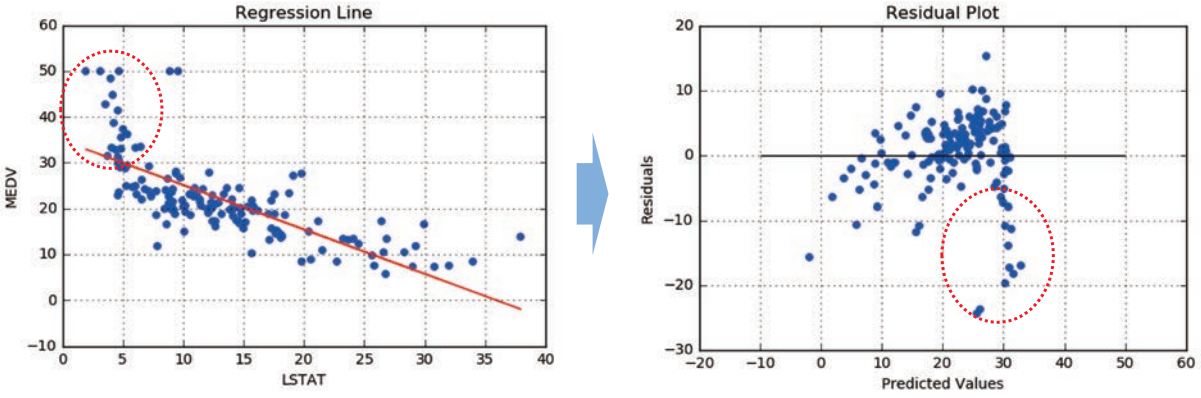
回帰線の確認

- 目的変数である「MEDV: 住宅価格」と比較的強い負の相関がある説明変数である「LSTAT: 低所得者の割合」の散布図と、予測線を図示します。
- LSTATが7～28の範囲では、予測線は住宅価格をよく表現できていますが、その範囲外では乖離があることが確認できます。



残差の確認

- 目的変数である「MEDV: 住宅価格」と予測線の差分（残差）を図示してみます。
- 残差が0を中心にばらついていれば良いモデルが作れたと言えます。今回はLSTATが7～28の範囲外にあるデータへの残差が大きなモデルとなっていることが確認できます。



モデル性能の評価

- モデルの性能を評価するために、何かしらの指標を設定したほうが便利です。線形回帰モデルの性能評価として、下記の指標を用いることが一般的です。
  - 平均二乗誤差：残差平方和をデータ数で正規化した値
  - 決定係数：相関係数の二乗

```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  lr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

```
r^2 train data:  0.5524780757890007
r^2 test data:  0.5218049526125568
```

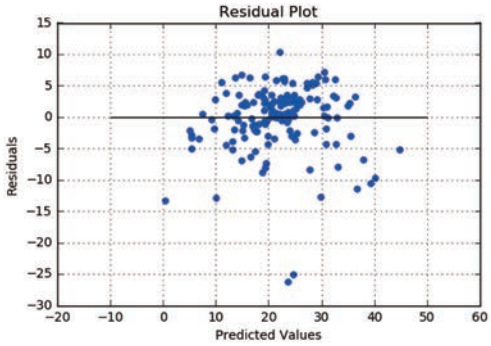
決定係数の例。モデル作成に使用した学習データに対する決定係数が、テストデータに対する決定係数より若干高いことが確認できます。

演習4：重回帰モデル

- 住宅価格の中央値を全項目から予測する線形回帰モデルを構築し、精度を確認してください。

重回帰モデルの精度

- 説明変数を追加したことにより、決定係数が大幅に改善されていることが確認できます。
- モデルの精度向上のみが目的であれば説明変数を増やして精度向上を図るのは1つの方法ですが、「過学習」の問題が発生することがあります。
- また、モデルの出力結果への説明変数の寄与度を正しく評価できなくなる「多重共線性」の問題が発生することがあります。




```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  lr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

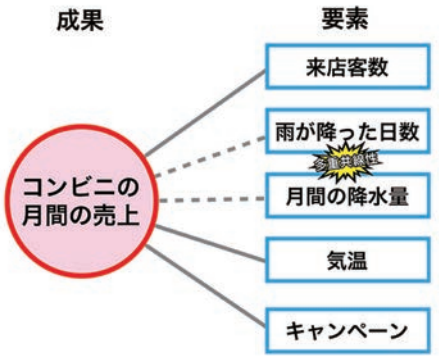
```
r^2 train data:  0.7644563391821222
r^2 test data:  0.6735280865347231
```

多重共線性

- 説明変数を増やしていくと一般的にモデルの表現力が向上し、精度が向上します。
- モデルの精度を高めることのみが目的であれば支障がないこともありますが、モデルの説明性（モデルはなぜそのような予測をしたのか、の説明）が問われる場合、説明変数を闇雲に増やすことには注意が必要です。

多重共線性

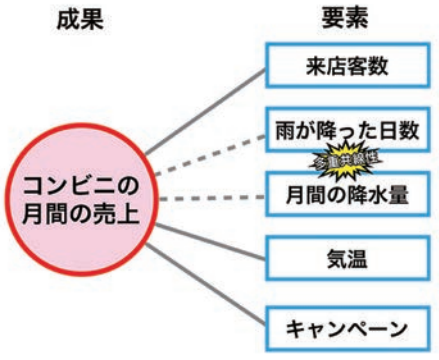
- 説明変数間で相関係数が高い時に多重共線性（multicollinearity）という問題が発生します。
- 多重共線性とは、モデル式の係数が不安定（符号と大きさが安定しない）になり、モデルの予測結果に対する係数の寄与度を正しく評価することができなくなってしまいます。



出展：https://xica.net/vno4ul5p/

多重共線性

- 多重共線性の回避策としては、相関が高い係数のどちらか一方をモデルから外す、ことが一般的です。



出展：https://xica.net/vno4ul5p/

多重共線性の事例

- 説明変数に全項目を使用した重回帰モデルの係数を表1に示します。INDUSとNOXの符号が逆になっているのが確認できます。

表1：重回帰モデルの係数

	0	1	2	3	4	5	6	7	8	9	10	11	12
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.119859	0.0444233	0.0118612	2.51295	-16.271	3.8491	-0.00985472	-1.50003	0.241508	-0.0110672	-1.01898	0.00695273	-0.488111

符号が逆になっている。

表2：全項目の相関係数

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

INDUSとNOXは正の相関がある。

MEDV（住宅価格）に対し、INDUSとNOXは負の相関がある。

過学習、多重共線性の回避

- 過学習や多重共線性を回避するために正則化という手法が存在します。
  - L1正則化：いくつかの説明変数の係数を0にする手法（特徴選択を行っていることになる）です。スパース（疎な）な行列で表現するため、高速に計算できるようになる。
  - L2正則化：各説明変数の係数が大きくなりすぎないようにする（個々の特徴量が出力に与える影響をなるべく小さくした）手法です。

演習5：L1正則化

- モデルにL1正則化を適用し、精度を確認してください。

演習6：L2正則化

- モデルにL2正則化を適用し、精度を確認してください。

正則化の効果

- L1正則化を実施することでINDUS、CHAS、NOXの係数が0となっていることが確認できます。
- L2正則化を実施することで係数の大きさが均されているのが確認できます。またINDUSとNOXの符号がちぐはぐになっていた問題が解消されていることが確認できます。
- 今回の例では、正則化を実施した影響で精度が若干低下していることも確認できます。精度向上を目的とするのか、モデルの説明性を高めることを目的とするのかで正則化を実施するか否かが異なってきます。

正則化なし

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.12131	0.0444664	0.0113417	2.51125	-16.2313	3.85907	-0.00998517	-1.50027	0.242143	-0.0110716	-1.01775	0.00681447	-0.486738

r^2 train data: 0.7645451026942549
r^2 test data: 0.6733825506400193

L1正則化後

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.0658619	0.0483293	-0	0	-0	0.868985	0.01218	-0.751094	0.200074	-0.0139506	-0.846024	0.00668818	-0.732666

r^2 train data: 0.7084095500978868
r^2 test data: 0.6115433359595557

L2正則化後

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	-0.118309	0.046126	-0.0208626	2.45869	-8.25958	3.89749	-0.017914	-1.39737	0.218432	-0.0116338	-0.931711	0.00726996	-0.494047

r^2 train data: 0.7623440182689594
r^2 test data: 0.6665819091486689

第6回：ロジスティック回帰

アジェンダ

- 第1回講義の振り返り
- ロジスティック回帰モデルの構築

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

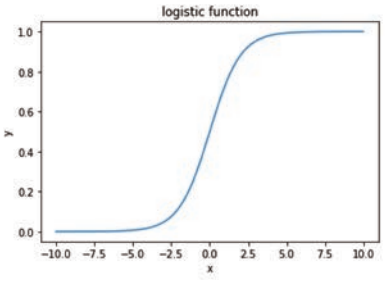
- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

ロジスティック回帰モデルの構築

ロジスティック回帰とは

- ロジスティック関数とは、シグモイド関数（ロジスティック関数）を用いて二値の分類を行うアルゴリズムです。
- ある現象の発生確率を、複数の因子の組み合わせとそれらの重みからモデル化します。

$$S_a(x) = \frac{1}{1+e^{-ax}} = \frac{\tanh(ax/2)+1}{2}$$



演習：ロジスティック回帰による乳がん診断モデルの構築

- sklearnのデータセットの中に含まれるbreast-cancerというデータセットを用いて、乳がんを診断するモデルを作成します。

演習1：データ項目の確認

- 乳がんデータの項目を確認してください。

演習2：項目間の相関

- benign（良性腫瘍）と他の項目の相関を確認してください。

項目間の相関

- benign（良性腫瘍）とmean perimeter（細胞の周囲長の平均値）、mean concave points（平均凹点）は比較的強い負の相関があることがわかります。

	mean radius	mean texture	mean perimeter	mean area	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension	benign
mean radius	1.000000	0.323782	0.997855	0.987357	0.413463	0.526911	0.744214	0.163953	0.007066	-0.730029
mean texture	0.323782	1.000000	0.329533	0.321086	0.277830	0.301025	0.295316	0.105008	0.119205	-0.415185
mean perimeter	0.997855	0.329533	1.000000	0.986507	0.455774	0.563879	0.771241	0.189115	0.051019	-0.742636
mean area	0.987357	0.321086	0.986507	1.000000	0.390410	0.512606	0.722017	0.143570	0.003738	-0.708984
mean smoothness	0.170581	-0.023389	0.207278	0.177028	0.472468	0.434926	0.503053	0.394309	0.499316	-0.358560
mean compactness	0.506124	0.236702	0.556936	0.498502	0.865809	0.816275	0.815573	0.510223	0.687382	-0.596534
mean concavity	0.676764	0.302418	0.716136	0.685983	0.754968	0.884103	0.861323	0.409464	0.514930	-0.696360
mean concave points	0.822529	0.293464	0.850977	0.823269	0.667454	0.752399	0.910155	0.375744	0.368661	-0.776614
mean symmetry	0.147741	0.071401	0.183027	0.151293	0.473200	0.433721	0.430297	0.699826	0.438413	-0.330499

演習3：ロジスティック回帰モデル

- 良性腫瘍の割合を予測するロジスティック回帰モデルを構築し、精度を確認してください。

第7回：サポートベクター回帰

アジェンダ

- 第1回講義の振り返り
- サポートベクター回帰モデルの構築

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

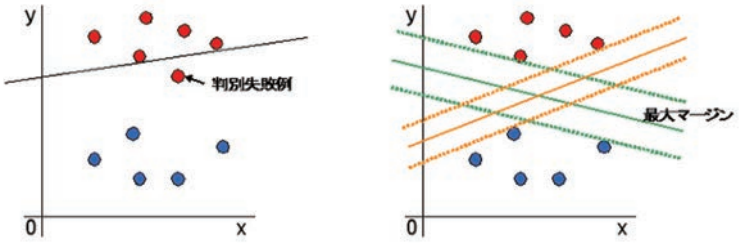
サポートベクター回帰モデルの構築

サポートベクター回帰とは

- サポートベクター回帰（SVR）とは、サポートベクターマシン（SVM）を利用した回帰のことです。
- カーネルトリックを使用することにより、非線形回帰も可能です。

参考：サポートベクターマシンとは

- サポートベクターマシン（SVM）はパターン識別用の教師あり機械学習方法であり、局所解収束の問題を軽減できるという長所があります。
- 「マージン最大化」というアイデア等で汎化能力も高め、非常に優秀なパターン識別能力を持つとされています。
- SVMでは、右図のように、2つのグループ間の最も距離の離れた箇所（最大マージン）を見つけ出し、その真ん中に識別の線を引くことによって、多くの未学習データの判別が可能になる事を「汎化能力」の向上を狙います。

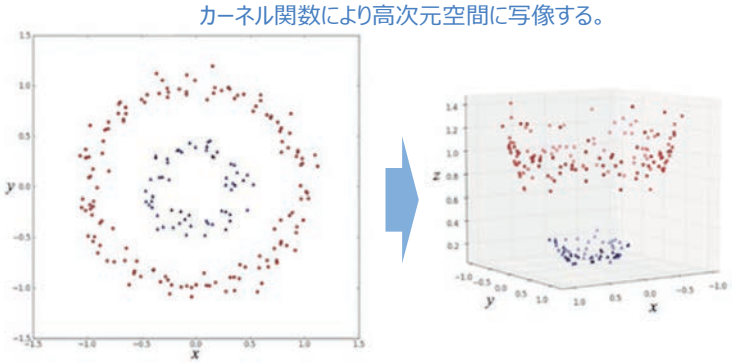


出展：http://www.sist.ac.jp/~kanakubo/research/neuro/supportvectormachine.html

参考：線形分離不可能な問題への適用

- カーネル・トリックという方法で、線形分離不可能な問題に関してもSVMが適用可能になりました。
- カーネルトリックとは元々のデータ空間から高次元空間にデータを写像することです。そうすることで、高次元空間上で線形データ解析を行うことが可能になります。

カーネル関数により高次元空間に写像する。



出展：http://enakai00.hatenablog.com/entry/2017/10/13/145337

演習：サポートベクター回帰による住宅価格の予測モデルの構築

- ボストン市郊外の地域別住宅価格を予測するサポートベクター回帰モデルを作成します。

演習1：データ項目の確認

- ボストン市郊外の地域別住宅価格データの項目を確認してください。

変数	説明
CRIM	町ごとの一人当たりの犯罪率
ZN	25,000平方フィートを超える敷地に区画された宅地の割合
INDUS	非小売業種の土地面積の割合
CHAS	Charles Riverダミー変数（敷地が川の境界にある場合は1、それ以外の場合は0）
NOX	窒素酸化物の濃度（1000万分の1）
RM	1住戸あたりの平均部屋数
AGE	1940年以前に建設された所有者居住ユニットの割合
DIS	ボストンの5つの雇用センターまでの重み付き距離
RAD	ラジアルハイウェイ（放射状に各方面へ伸びる高速道路）へのアクセスのしやすさの指標
TAX	10,10,000ドルあたりの全額固定資産税率
PTRATIO	町による生徒 - 教師比率
B	1000（Bk - 0.63）^2ここでBkは町による黒人の割合
LSTAT	低所得者の割合
MEDV	住宅価格の中央値（1,000単位）

演習2：項目間の相関

- MEDV（住宅価格の中央値）と他の項目の相関を確認してください。

項目間の相関

- MEDV（住宅価格の中央値）とRM（1住戸あたりの平均部屋数）は比較的強い正の相関があることがわかります。
- MEDV（住宅価格の中央値）とLSTAT（低所得者の割合）は比較的強い負の相関があることがわかります。

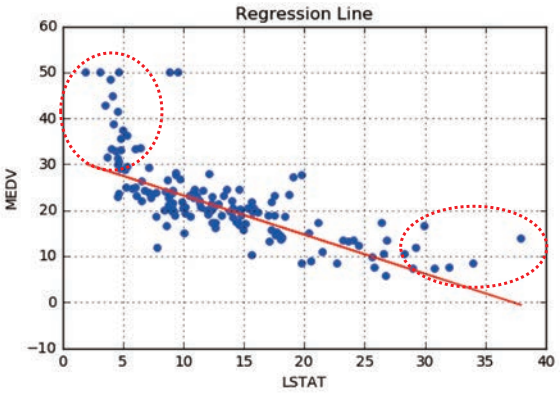
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
MEDV	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

演習3：単変量回帰モデル

- 住宅価格の中央値を低所得者の割合から予測するサポートベクター回帰モデルを構築し、精度を確認してください。

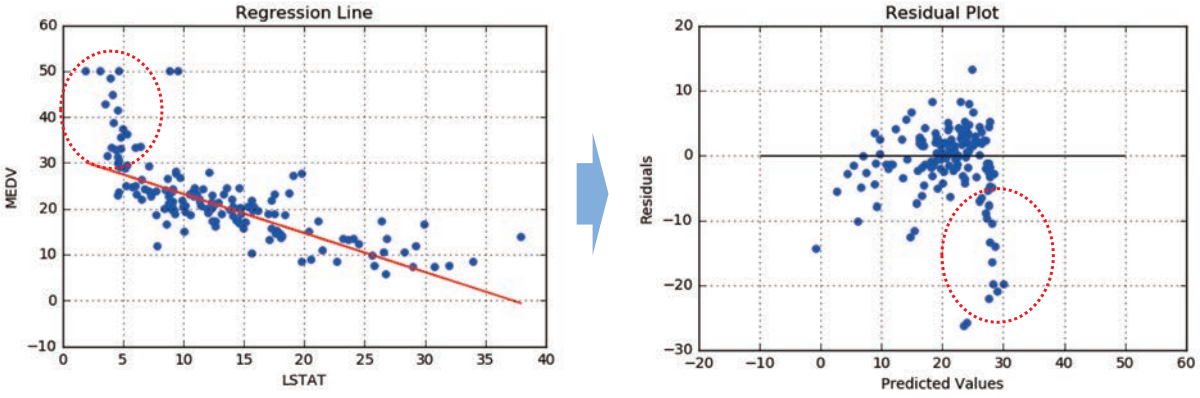
回帰線の確認

- 目的変数である「MEDV: 住宅価格」と比較的強い負の相関がある説明変数である「LSTAT: 低所得者の割合」の散布図と、予測線を図示します。
- LSTATが7～28の範囲では、予測線は住宅価格をよく表現できていますが、その範囲外では乖離があることが確認できます。



残差の確認

- 目的変数である「MEDV: 住宅価格」と予測線の差分（残差）を図示してみます。
- 残差が0を中心にばらついていれば良いモデルが作れたと言えます。今回はLSTATが7～28の範囲外にあるデータへの残差が大きなモデルとなっていることが確認できます。



モデル性能の評価

- モデルの性能を評価するために、何かしらの指標を設定したほうが便利です。回帰モデルの性能評価として、下記の指標を用いることが一般的です。
  - 平均二乗誤差：残差平方和をデータ数で正規化した値
  - 決定係数：相関係数の二乗

```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  svr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

```
r^2 train data:  0.5133879705847997
r^2 test data:  0.4941954185394163
```

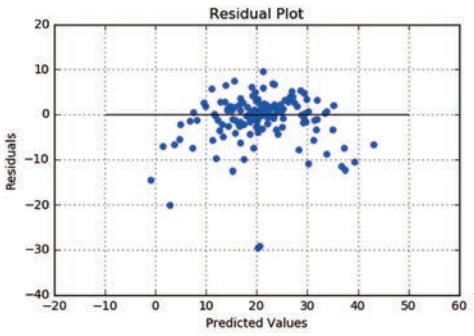
決定係数の例。モデル作成に使用した学習データに対する決定係数が、テストデータに対する決定係数より若干高いことが確認できます。

演習4：多変量回帰モデル

- 住宅価格の中央値を全項目から予測するサポートベクター回帰モデルを構築し、精度を確認してください。

多変量回帰モデルの精度

- 説明変数を追加したことにより、決定係数が大幅に改善されていることが確認できます。




```
# R2スコアを表示します。
from sklearn.metrics import r2_score

print('r^2 train data: ', r2_score(Y_train,  svr.predict(X_train)))
print('r^2 test data: ', r2_score(Y_test, Y_pred))
```

```
r^2 train data:  0.7362973043042667
r^2 test data:  0.6168014926864142
```

演習5：説明変数の標準化

- 説明変数に標準化を適用し、精度を確認してください。

演習6：カーネルトリックの使用

- ガウシアンカーネルを使用したモデルを適用し、精度を確認してください。

標準化、カーネルトリックの効果

説明変数の標準化とカーネルトリックの使用によって、回帰モデルの精度が向上していることが確認できます。

標準化なし カーネルトリックなし	r^2 train data: 0.7362973043042667 r^2 test data: 0.6168014926864142
標準化あり カーネルトリックなし	r^2 train data: 0.738592163183546 r^2 test data: 0.6227047688726721
標準化あり カーネルトリックあり	r^2 train data: 0.9699948811627955 r^2 test data: 0.8224137241371853

第8回：階層的クラスタリング

アジェンダ

- 第1回講義の振り返り
- 階層的クラスタリングによる発話の分類

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

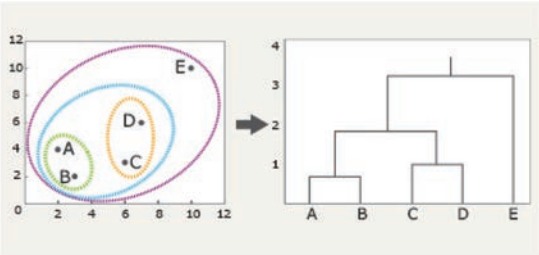
データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

階層的クラスタリングによる発話の分類

階層的クラスタリングとは

- 最も似ているデータ同士から順番にグループ（クラスター）にしていく方法で、途中過程が階層のように表せます。
- グループにしていく順番は、右図のような樹形図（デンドログラム）で表現できます。



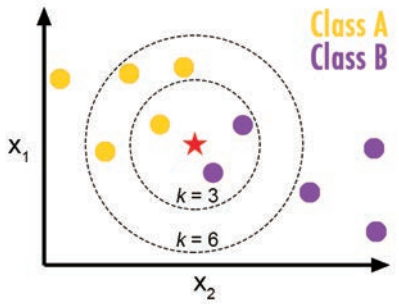
出展：https://www.albert2005.co.jp/knowledge/data_mining/cluster/hierarchical_clustering

演習：階層的クラスタリングによる対話分類モデルの構築

- 人間の発話をカテゴリごとに分類するモデルをK近傍法、Ward法（ウォード法）で作成します。

K近傍法

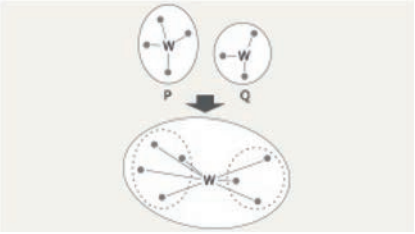
- K近傍法は階層的クラスタリング手法の1つで、教師あり学習です。
- まず学習データをベクトル空間上にプロットします。テストデータから距離が近い順に任意のK個の学習データを取得し、多数決でテストデータが属するクラスを推定します。
- K=3であればテストデータ（赤い星）はClass Bだと判定されますが、K=6にするとテストデータはClass Aと判定されます。



出展：https://qiita.com/yshi12/items/26771139672d40a0be32

Ward（ウォード）法

- クラスター同士が結合するとした時、結合後の全てのクラスターにおいて、クラスターの重心とクラスター内の各点の距離の二乗和の合計が最小となるようにクラスターを結合させていく方法のことです。



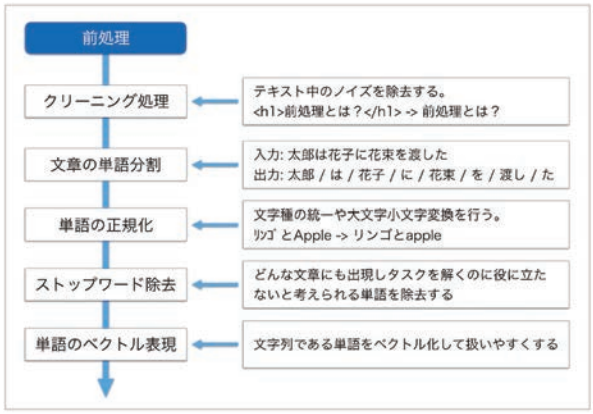
出展： https://www.albert2005.co.jp/knowledge/data_mining/cluster/hierarchical_clustering

演習1：独自データ収集

- 文章をベクトル化するプログラムを実装してください。
- 下記の問いかけの発話事例をベクトル化します。
  class_0: 名前を聞かれる
  class_1: 好きな色を聞かれる
  class_2: 好きな食べ物を聞かれる
  class_3: 年齢を聞かれる
  class_4: 挨拶される

自然言語処理における前処理

- コンピュータは自然言語そのものを理解できないため、コンピュータが理解できるデータに自然言語を変換する必要があります。
- 自然言語処理においてモデルを作成するときは、単語や文章をベクトル（多次元の要素を持つ量）に変換し、コンピュータで処理します。



出展： https://qiita.com/Hironsan/items/2466fe0f344115aff177

文章のベクトル化：Bag of Words

- ベクトル表現の一種で、文章に単語が含まれるかどうかのみを考え、単語の並び方などは考慮しない形式のことです。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ て いる ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

分類に寄与する文言

- 発話の種類ごとに、分類に寄与しそうな文言を入れるとモデルの精度が向上します。
- 例えば「class_0: 名前を聞かれる」の場合だと、"名前"という文言のことです。
- 「あなた」や「教えて」などの文言はどのクラスにも登場する可能性があり、発話の分類には寄与しないと考えられます。

```
sentences = [
    "名前は何",
    "名前をなんていうの",
    "名前教えて",
    "あなたのお名前は",
    "お名前教えてよ",

    "どんな色が好きなの",
    "何色が好き",
    "好きな色は何",
    "黄色は好き",
    "好きな色を教えてちょうだい",

    "どんな食べものが好きなの",
    "ピーマンは食べれる",
    "好きな食べものは",
    "食べものは何が好きなの",
    "何が美味しい",

    "歳はいくつですか",
    "歳はいくつになった",
    "何歳なの",
    "何歳か教えて",
    "何歳ですか",

    "おはよう",
    "おはようございます",
    "こんにちは",
    "こんばんは",
    "おやすみなさい",
]
```

演習2：K近傍法の実装

- 作成したベクトルをK近傍法でクラスタリングしてください。

演習3：Ward法による分類モデル

- 作成したベクトルをWard法でクラスタリングしてください。

文章のベクトル化：TF-IDF

- 文章をベクトル化する際に、単語に重みを付けて評価する手法です。単語の出現頻度であるTF（Term Frequency）と、IDF（Inverse Document Frequency）という2つの指標を使用します。
- TF（Term Frequency）は、「各文書においてその単語がどのくらい出現したのか」を意味します。よく出現する単語は、その文章の特徴を捉えるのに有用だろうという考え方です。
- IDF（Inverse Document Frequency）は、単語が稀にしか出現しないなら高い値を、「色々な文書によく出現する単語」なら低い値を示すものです。稀少な単語は、その文書の特徴を捉えるのに有用だろうという考え方です。

$$tf = \frac{\text{文書}A\text{における単語}X\text{の出現頻度}}{\text{文書}A\text{における全単語の出現頻度の和}} \qquad idf = log(\frac{\text{全文書数}}{\text{単語}X\text{を含む文書数}}) \qquad tfidf = tf * idf$$

演習4：TF-IDFの実装

- 作成したベクトルにTF-IDFを適用し、新たなベクトルを作成してください。
- TF-IDFで作成したベクトルの、スコアの高い形態素の上位N件を表示するプログラムを作成してください。

演習5：TF-IDFでベクトル化したデータによるクラスタリング

- TF-IDFで作成した新しいベクトルを使用して、Ward法で再度クラスタリングしてください。

第9回：非階層的クラスタリング

アジェンダ

- 第1回講義の振り返り
- 非階層的クラスタリングによる発話の分類

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

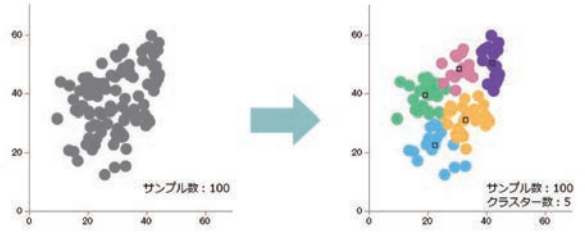
データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

非階層的クラスタリングによる発話の分類

非階層的クラスタリングとは

- 予めいくつのクラスターに分けるかを分析者が決定し、決めた数の塊（排他的部分集合）にサンプルを分割する方法です。



出展：https://www.albert2005.co.jp/knowledge/data_mining/cluster/non-hierarchical_clustering

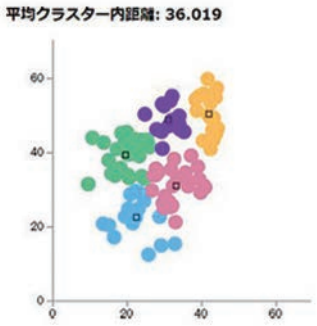
演習：非階層的クラスタリングによる対話分類モデルの構築

- 人間の発話をK-means法で分類します。

K-means法

K-Means法は以下のステップに沿ってデータをクラスタリングします。

1. 各点にランダムにクラスタを割り当てる。
2. クラスタの重心を計算する。
3. 点のクラスタを、一番近い重心のクラスタに変更する。
4. 変化がなければ終了。変化がある限りは 2. に戻る。



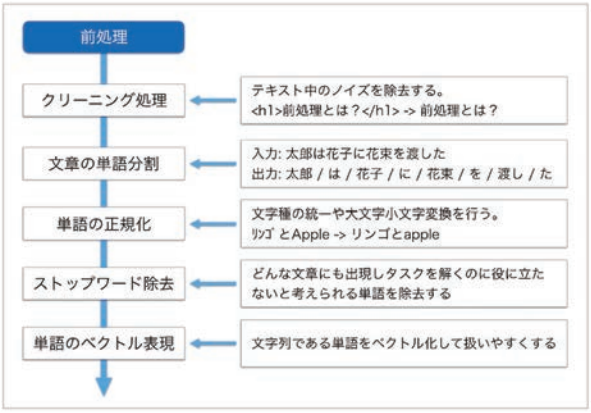
出展：
https://www.albert2005.co.jp/knowledge/data_mining/cluster/non-hierarchical_clustering

演習1：独自データ収集

- 文章をベクトル化するプログラムを実装してください。
- 下記の問いかけの発話事例をベクトル化します。
  class_0: 名前を聞かれる
  class_1: 好きな色を聞かれる
  class_2: 好きな食べ物を聞かれる
  class_3: 年齢を聞かれる
  class_4: 挨拶される

自然言語処理における前処理

- コンピュータは自然言語そのものを理解できないため、コンピュータが理解できるデータに自然言語を変換する必要があります。
- 自然言語処理においてモデルを作成するときは、単語や文章をベクトル（多次元の要素を持つ量）に変換し、コンピュータで処理します。



出展： https://qiita.com/Hironsan/items/2466fe0f344115aff177

文章のベクトル化：Bag of Words

- ベクトル表現の一種で、文章に単語が含まれるかどうかのみを考え、単語の並び方などは考慮しない形式のことです。

Step1：解析対象の文章群を準備します。

```
['天気 を 教え て ください 。',
 '明日 の 天気 は どう です か ?',
 '今日 の 天気 を 教え て よ 。',
 '新宿 の 天気 は どう なっ て いる ?',
 '横浜 の 明日 の 天気 は どう か の かな ?',
 '気温 を 教え て よー 。',
 '明日 の 気温 は どう な の ?',
 '今日 の 気温 は 低い ね !',
 '横浜 の 気温 は ?',
 '新宿 の 昨日 の 気温 を 教え て ください 。']
```

Step2：重複しない形態素のリストを作成し、次元を決定します。

```
{'いる': 0,
 'かな': 1,
 'ください': 2,
 'です': 3,
 'どう': 4,
 'なっ': 5,
 'よー': 6,
 '今日': 7,
 '低い': 8,
 '天気': 9,
 '教え': 10,
 '新宿': 11,
 '明日': 12,
 '昨日': 13,
 '横浜': 14,
 '気温': 15}
```

Step3：形態素リストを元に、解析対象の文書群をベクトルに変換します。

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
       [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
       [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1],
       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1],
       [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]], dtype=int64)
```

分類に寄与する文言

- 発話の種類ごとに、分類に寄与しそうな文言を入れるとモデルの精度が向上します。
- 例えば「class_0: 名前を聞かれる」の場合だと、"名前"という文言のことです。
- 「あなた」や「教えて」などの文言はどのクラスにも登場する可能性があり、発話の分類には寄与しないと考えられます。

```
sentences = [
    "名前は何",
    "名前をなんていうの",
    "名前教えて",
    "あなたのお名前は",
    "お名前教えてよ",

    "どんな色が好きなの",
    "何色が好き",
    "好きな色は何",
    "黄色は好き",
    "好きな色を教えてちょうだい",

    "どんな食べものが好きなの",
    "ピーマンは食べれる",
    "好きな食べものは",
    "食べものは何が好きなの",
    "何が美味しい",

    "歳はいくつですか",
    "歳はいくつになった",
    "何歳なの",
    "何歳か教えて",
    "何歳ですか",

    "おはよう",
    "おはようございます",
    "こんにちは",
    "こんばんは",
    "おやすみなさい",
]
```

演習2：K-means法の実装

- 作成したベクトルをK-means法でクラスタリングしてください。

文章のベクトル化：TF-IDF

- 文章をベクトル化する際に、単語に重みを付けて評価する手法です。単語の出現頻度であるTF（Term Frequency）と、IDF（Inverse Document Frequency）という2つの指標を使用します。
- TF（Term Frequency）は、「各文書においてその単語がどのくらい出現したのか」を意味します。よく出現する単語は、その文章の特徴を捉えるのに有用だろうという考え方です。
- IDF（Inverse Document Frequency）は、単語が稀にしか出現しないなら高い値を、「色々な文書によく出現する単語」なら低い値を示すものです。稀少な単語は、その文書の特徴を捉えるのに有用だろうという考え方です。

$$tf = \frac{\text{文書}A\text{における単語}X\text{の出現頻度}}{\text{文書}A\text{における全単語の出現頻度の和}} \qquad idf = log(\frac{\text{全文書数}}{\text{単語}X\text{を含む文書数}}) \qquad tfidf = tf * idf$$

演習3：TF-IDFの実装

- 作成したベクトルにTF-IDFを適用し、新たなベクトルを作成してください。

演習4：TF-IDFでベクトル化したデータによるクラスタリング

- TF-IDFで作成した新しいベクトルを使用して、K-means法で再度クラスタリングしてください。

第10回：異常検知

アジェンダ

- 第1回講義の振り返り
- 異常検知アルゴリズムの実装

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

異常検知アルゴリズムの実装

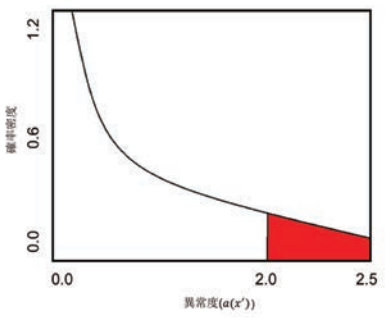
異常検知とは

- 大多数のデータと異なる挙動を示すデータを検出する技術のことを異常検知といいます。クレジットカードの不正使用検知、システムの故障予知など様々な分野で応用されている技術です。
- 異常検知は「統計モデルに基づく手法」、「データ間の距離に基づく手法」があります。
- 異常検知のプロセスは下記のステップで構成されます。
  1. 正常データのパターン抽出する。
  2. 正常・異常の境目となる閾値を設定する。
  3. 被テストデータが閾値を超えた場合、異常と判断する。

ホテリング理論

- ホテリング理論では異常度を右のような式で定義します。σは標準偏差、μは平均を表しています。
- ホテリング理論ではデータが正規分布に従っている事を仮定しています。異常度はデータ数が十分に大きければ自由度1のカイ二乗分布に従うとしています。
- 右図に示されたカイ二乗分布では曲線で囲まれた面積が確率を表しています。面積が小さければごく稀に発生する異常時データである可能性が高いと言えます。

$$a(x') = \left(\frac{x' - \hat{\mu}}{\hat{\sigma}} \right)^2$$



出展：
https://qiita.com/Zepprix/items/f6a5de2e3f6689bd2c1f

ホテリング理論の問題点

- データが単一の正規分布から発生していると仮定しています。正規分布から著しく外れているデータの場合や分布が複数の山を持つ場合などは、異常値を正しく判断できなくなります。
- 正規分布のパラメータは変化しないと仮定しているため、分布のパラメータが変化する時系列データのようなデータには適用することができません。

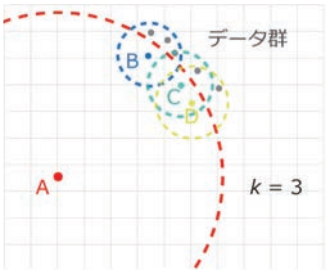
演習1：ホテリング理論による異常検知

- ホテリング理論によって異常検知アルゴリズムを実装してください。

局所外れ値因子法（LOF法）

- 右式の局所密度を利用して異常検知をするのがLOF（Local Outlier Factor ）法です。
- 外れ値である点Aを基準とすると、A自身の局所密度は低く、近傍点である点B, C, D の局所密度は高くなっています。自身の局所密度と近傍点の局所密度が等しいときほど正常データであり、その差が大きいほど外れ値である可能性が高いと解釈できます。

$$\text{局所密度} = \frac{1}{\text{近傍}k\text{個の点との距離の平均}}$$



出展：
https://qiita.com/Zepprix/items/f6a5de2e3f6689bd2c1f

演習2：LOF法による異常検知

- 局所外れ値因子法（LOF法）によって異常検知アルゴリズムを実装してください。

第11回：アソシエーション分析

アジェンダ

- 第1回講義の振り返り
- アソシエーション分析アルゴリズムの実装

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

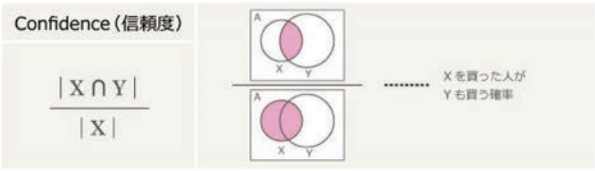
アソシエーション分析アルゴリズムの実装

アソシエーション分析とは

- アソシエーション分析（Association Analytics）とは、マーケティングで利用される代表的なデータ分析手法です。商品の購入履歴から購入パターンや売買履歴を分析することで、ある商品Aと商品Bの売れ行きについて関連性を抽出します。
- アソシエーション分析では、独立した複数の指標の組み合わせを基にして「商品Aが売れるときは商品Bが一緒に売れることが多い」などのルールを見つけ出すことが可能となります。
- アソシエーション分析で使用する指標については次頁以降で説明します。

Confidence（信頼度）

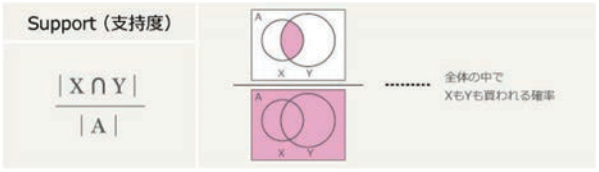
- レコメンデーションで最も基本となる指標がConfidence（信頼度）です。
- 例えば「トマトを買った人のうち、どれくらいの人がきゅうりも買ったか」という確率となります。トマトを買った人は3人で、このうち2人がきゅうりも買った場合は信頼度は66.7%になります。



出展：
https://www.albert2005.co.jp/knowledge/marketing/customer_product_analysis/abc_association

Support（支持度）

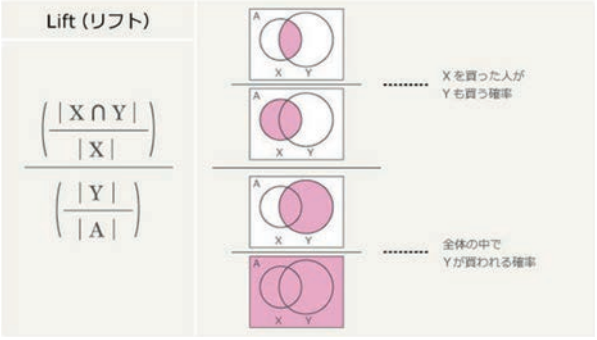
- Support（支持度）は「そもそもトマトときゅうりがどのくらい一緒に売れているのか」という指標です。
- たまたま1人がトマトを購入したときにきゅうりも購入すると Confidence（信頼度）は100%となり、Condifenceだけで判断すると、次にトマトを買う人に最も薦めるべき商品はきゅうりになってしまいます。
- もしトマトときゅうりをセットで買う人があまりいなければ、トマトを買う人にきゅうりを薦めるのは、あまりよいレコメンドとは言えません。



出展：
https://www.albert2005.co.jp/knowledge/marketing/customer_product_analysis/abc_association

Lift（リフト）

- いくらConfidence（信頼度）が高くても、「推薦される商品を誰もが買っている場合はあまりよいレコメンデーションではない」という考えに基づいた指標です。



出展：
https://www.albert2005.co.jp/knowledge/marketing/customer_product_analysis/abc_association

演習1：POSデータの可視化

- POSデータに含まれる製品の分布を可視化してください。

演習2：指標の計算

- 条件部(lhs)、結論部(rhs)、support（支持度）、confidence（確信度）を表示してください。

演習3：指標の計算

- 条件部(lhs)、結論部(rhs)、support（支持度）、confidence（確信度）に加えてリフト(lift)を表示してください。

第12回：主成分分析

アジェンダ

- 第1回講義の振り返り
- 主成分分析アルゴリズムの実装

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

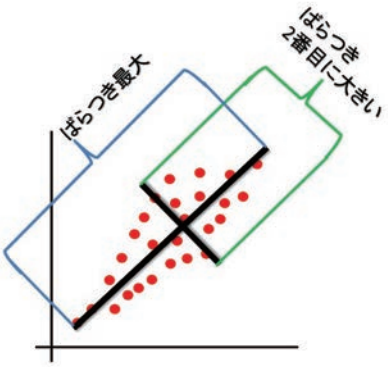
データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

主成分分析アルゴリズムの実装

主成分分析とは

- 分散が最大になるように主成分軸を引くことを主成分分析といいます。
- 最も長い軸を第1主成分軸、次に長い軸を第2主成分軸といいます。
- 主成分分析を実施すると、主成分軸を通して多次元のデータを要約することができます。



出展： https://logics-of-blue.com/principal-components-analysis/

演習1：主成分分析の実施

- 2次元のダミーデータを作成し、主成分分析を実施してください。

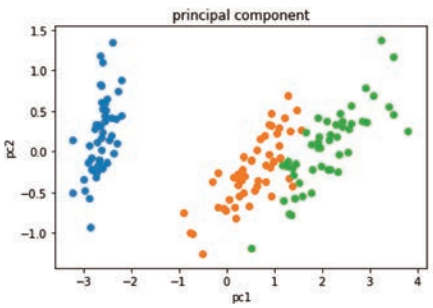
演習2：次元圧縮

- アイリスデータで主成分分析を実施してください。

次元の圧縮

- アイリスデータの特徴部分の次元は4次元です。主成分分析を実施して2次元に圧縮したのが右図です。
- 様々なデータについて分類器を作成する際は次元を圧縮しないほうが精度が高くなることが多いですが、次元を圧縮して単純化することによって、データの特性に対する気付きが得られることがあります。

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2]])
```



第13回：グラフ分析

アジェンダ

- 第1回講義の振り返り
- グラフ分析例

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

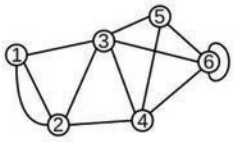
- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

グラフ分析例

グラフ理論とは

- ノード（節点・頂点）の集合とエッジ（枝・辺）の集合で構成されるグラフに関する数学の理論をグラフ理論といいます。
- 路線図の検討、電気回路の設計、SNS上でのインフルエンサーの分析などに応用されています。

●隣接行列



	1	2	3	4	5	6
1	0	2	1	0	0	0
2	2	0	1	1	0	0
3	1	1	0	1	1	1
4	0	1	1	0	1	1
5	0	0	1	1	0	1
6	0	0	1	1	1	1

出展： https://www.slideshare.net/KMC_JP/graph-and-tree

ノードの評価

ネットワーク上の各ノードを表す指標として、下記のような指標が存在します。

- 次数中心性：他のノードとつながっているエッジが多いほど、中心性が高いとする指標です。
- 媒介中心性：そのノードを通る経路が多いほど、中心性が高いとする指標です。
- ページランク：ウェブページのリンク関係からページの重要度を測るアルゴリズムです。PageRankは、1998年にGoogle創業者のラリー・ペイジとセルゲイ・ブリンにより発明されました。

演習1：ネットワークの描画

- NetworkX（グラフ/ネットワーク理論系の計算ライブラリ）から空手クラブの友人関係ネットワークを読み込み、図示してください。

演習2：次数中心性の計算

- 次数中心性（グラフの頂点に接合する辺の数）の大きな人物を表示してください。

演習3：媒介中心性の計算

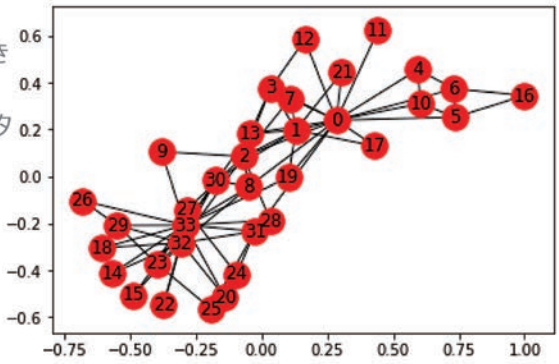
- 媒介中心性（当該ノードを通る経路数）の大きな人物を表示してください。

演習4：ページランクの計算

- ページランク（Googleがページの重要度を図るために使用していることで知られている指標）の大きな人物を表示してください。

グラフによる表現が向いている事象

- ノード0にとって、ノード1/4/10などは1次のつながり（Tier1）です。ノード4の先にいるノード6は、ノード0に取っては2次のつながり（Tier2）です。
- Tier1のノード群の中心性が小さくても、その背後にいるTier2のノード群の中心性が大きい場合、ノード0はTier2のノードから何かしらの影響を受け得る（もしくは影響を与え得る）、と考えることができます。
- グラフにしてみると、リレーショナルデータベースのような2次元のデータ表示では気づかなかった特徴に気づく可能性があります。



第14回：テキストマイニング

アジェンダ

- 第1回講義の振り返り
- WordCloudの作成

第1回講義の振り返り

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

WordCloud作成

テキストマイニングとは

- テキストマイニングは文字列を対象としたデータマイニングのことです。
- 自然言語からなるデータを単語や文節で区切り、それらの出現の頻度や共出現の相関、出現傾向、時系列などを解析することで有用な情報を取り出すことを目的とします。

テキストマイニングの事例

テキストマニングは下記のような分析で応用されています。

- TwitterなどSNSの呟きや書き込みを分析
- アンケートの分析
- 新聞から株式市場の予測

WordCloudとは

- WordCloudとは、文章中で出現頻度が高い単語を選んで、その頻度に応じた大きさで図示する手法のことです。
- 頻度の多い単語が視覚的に大きく表示されるため、どのような単語が注目されているのか、直感的に捉えることができます。
- 頻度が多く注目されている単語を捉え、他の単語との共起を分析するなど、その他の詳細な分析へと進めていきます。

出展： https://www.pc-koubou.jp/magazine/2646

演習1：分析対象データの収集

- 解析したい口コミデータを収集してください。

演習2：WordCloudの準備

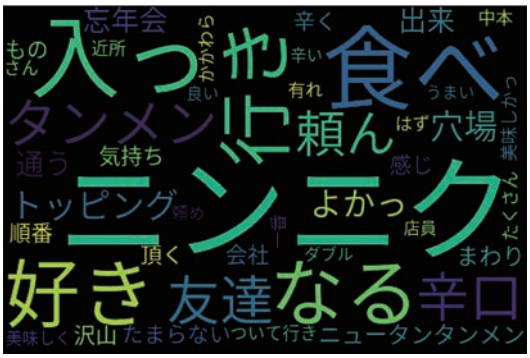
- 口コミを形態素解析して特定の品詞のみ抽出し、半角スペースで連結した文字列を作成する処理を実装してください。

演習3：WordCloudの作成

- WordCloudを作成してください。

注目ワードの抽出事例

- ニンニクが特徴的な担々麺のお店の口コミデータを分析した事例です。
- 「トッピングでニンニク」という点に、多くのユーザーが注目していることがわかります。
- 抽出する単語の品詞を名詞/動詞/形容詞と絞ることで、得られる情報が変わってきますので、いろいろ試行してみてください。



第15回：データマイニング総復習

アジェンダ

- データマイニングとは
- データマイニングの手法の分類
  - クラス分類
  - 回帰
  - クラスタリング
  - パターン抽出
  - その他の手法
- データマイニングの歴史と発展
- 各講義の振り返り

全15回の講義について

- データマイニングの各種アルゴリズムの理解を目標とする。
  - プログラミング言語としてはPython
  - Pythonの各種ライブラリを利用してデータ分析に必要なスキルの習得を目指す
  - 第2回以降の講義で詳細を取り扱う

データマイニングとは

- データマイニング（Data mining）とは、統計学、パターン認識、人工知能等のデータ解析の技法を大量のデータに網羅的に適用することで知識を取り出す技術・プロセスの総称です。
- 英語では“Data mining”の語の直接の起源となった研究分野であるknowledge-discovery in databases（データベースからの知識発見）の頭文字をとってKDDとも呼ばれます。

事例：マーケット・バスケット分析

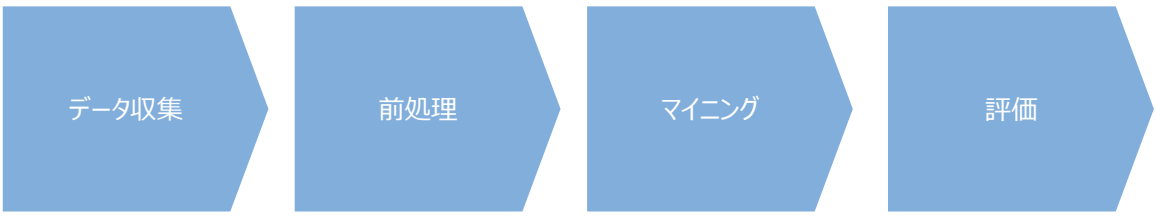
- データマイニングで有名な事例として、マーケット・バスケット分析があります。
- マーケット・バスケット分析とは、データ同士の関係性を分析するもので、どの商品とどの商品をどのような顧客が同時に購入したかを分析する手法です。

- 夕刻、紙おむつとビールが同時に購入される、という有名な事例がアメリカにあります。夕食の準備に忙しい母親に言われて商店に紙おむつを買いに来た父親が、自分へのご褒美にビールを買うため、と解釈されています。

データマイニングのプロセス

- データマイニングを行うために、まずはデータを収集することが必要です。一般的には、元となるデータが多ければ多いほど、有益な情報を採掘（マイニング）できる可能性が高まります。

- 収集されたデータは、データマイニングの各種アルゴリズムに適した形式に変換する「前処理」が施されます。



データマイニングのステップ

データマイニングの代表的なアルゴリズム

- クラス分類
  与えられたデータに対応するカテゴリを予測する問題に対する手法です。
  - 単純ベイズ分類器
  - 決定木
  - サポートベクターマシン
- 回帰
  与えられたデータに対応する実数値を予測する問題に対する手法です。
  - 線形回帰
  - ロジスティック回帰
- クラスタリング
  データの集合をグループに分ける問題に対する手法です。
  - K-means法
- 次元削減
  変数を削減し、結果に寄与する変数を特定したり、変数間の関係を分析する手法です。
  - 主成分分析

データマイニングの歴史

年号	できごと
1960年代	メインフレームが金融企業の基幹業務システムとして稼働開始した。同時に、デジタルデータの収集,蓄積,利用の試みが開始された。
1970年代	1971年から1973年にかけて、チリでサイバーシン計画が実行される。コントロールセンターが、テレックスを介して実時間でチリ各地に点在する工場からデータを収集して、収集したデータを元に、オペレーションズ・リサーチを用いて最適化した生産計画を作成し、工場に対して生産計画をフィードバックするシステムであった。
1980年代	現在の"Data mining"の定義と類似する"Knowledge Discovery in Databases"という語が出現する。関係データベースシステムとその操作用言語であるSQLが出現する。データウェアハウスの運用が開始される。
1990年代	1990年頃から始まった計算機の急激な性能向上により"Knowledge Discovery in Databases"の研究が大幅に加速される。 1999年 - 2010年代に大量の実世界データを収集・供給する基盤となるInternet of Things(IoT)の用語がKevin Ashtonにより初めて使用された。(この当時のIoTは、様々な物体にRFIDタグを貼り付け、RFIDに対応したセンサーを用いて物体からの情報収集を行い、収集した情報を活用することを指していた)
2000年代	インターネットへの常時接続が一般家庭にも普及する。インターネット上に蓄積されたデータが加速度的に増加する。後にデータの主要な供給源の1つとなる友人紹介型のソーシャル・ネットワーキング・サービスが2002年より相次いで提供され始める。コンピュータとインターネットの普及に着目し、ビジネスにおいて膨大に蓄積され活用しきれなくなったデータの分析を専門に行う企業も徐々に出現し始める。
2010年代	英国"The Economist"誌において"big data"の語が提唱された。コモディティ化によりコンピュータの計算能力が安価になり、高速データ処理用のコンピュータ・クラスタの構築が容易にできるようになった。データ分析のコストが下がり、ビッグデータ解析の応用が進むようになった。データサイエンティストという名称の職業が台頭し始めた。また、ビッグデータを用いたデータマイニングを応用したサービスが一般向けにも提供され始めた。コグニティブ・コンピューティング・システムが商用で実用化された。テレビ番組の紹介コーナーでも、インターネット上に存在するビッグデータの統計分析結果を元に流行のトレンドを紹介するようになった。 ディープラーニングの実用化が急速に進み、非常に多数の人工知能サービスが現れた。

出展： https://ja.wikipedia.org/wiki/データマイニング

データマイニングに用いられるツール・ライブラリ
R言語

Wikipediaより（https://ja.wikipedia.org/wiki/R言語）

- R言語（あーるげんご）はオープンソース・フリーソフトウェアの統計解析向けのプログラミング言語及びその開発実行環境である。ファイル名拡張子は.r, .R, .RData, .rds, .rda。
- R言語はニュージーランドのオークランド大学のRoss IhakaとRobert Clifford Gentlemanにより作られた。現在ではR Development Core Team[注 1] によりメンテナンスと拡張がなされている。

データマイニングに用いられるツール・ライブラリ
Python + Jupyter notebook + scikit-learnなど

Wikipediaより（https://ja.wikipedia.org/wiki/Scikit-learn）

- scikit-learn (旧称：scikits.learn) はPythonのオープンソース機械学習ライブラリ[2]である。サポートベクターマシン、ランダムフォレスト、Gradient Boosting、k近傍法、DBSCANなどを含む様々な分類、回帰、クラスタリングアルゴリズムを備えており、Pythonの数値計算ライブラリのNumPyとSciPyとやり取りするよう設計されている。

データマイニングに用いられるツール・ライブラリ
Weka

Wikipediaより（https://ja.wikipedia.org/wiki/Weka）

- Weka (Waikato Environment for Knowledge Analysis) は、ニュージーランドのワイカト大学で開発した機械学習ソフトウェアで、Javaで書かれている。GNU General Public License でライセンスされているフリーソフトウェアである。

第2回：「単純ベイズ分類器」の振り返り

- ベクトル化したデータで単純ベイズのモデルを作成し、モデルの精度を確認してください。

第3回：「決定木」の振り返り

- ベクトル化したデータで決定木モデルを作成し、モデルの精度を確認してください。
- 決定木モデルの判定に寄与した項目を確認してください。

第4回：「サポートベクターマシン」の振り返り

- ベクトル化したデータでSVMモデルを作成し、モデルの精度を確認してください。
- TF-IDFで作成したベクトルをSVMに投入してモデルを作成し、モデルの精度を確認してください。

第5回：「線形回帰」の振り返り

- 住宅価格の中央値を低所得者の割合から予測する線形回帰モデルを構築し、精度を確認してください。
- 住宅価格の中央値を全項目から予測する線形回帰モデルを構築し、精度を確認してください。
- モデルにL1正則化を適用し、精度を確認してください。
- モデルにL2正則化を適用し、精度を確認してください。

第6回：「ロジスティック回帰」の振り返り

- 良性腫瘍の割合を予測するロジスティック回帰モデルを構築し、精度を確認してください。

第7回：「サポートベクター回帰」の振り返り

- 住宅価格の中央値を全項目から予測するサポートベクター回帰モデルを構築し、精度を確認してください。
- ガウシアンカーネルを使用したモデルを適用し、精度を確認してください。

第8回：「階層的クラスタリング」の振り返り

- 文章をベクトル化するプログラムを実装してください。
- 作成したベクトルをK近傍法でクラスタリングしてください。
- 作成したベクトルをWard法でクラスタリングしてください。

第9回：「非階層的クラスタリング」の振り返り

- 文章をベクトル化するプログラムを実装してください。
- 作成したベクトルをK-means法でクラスタリングしてください。

第10回：「異常検知」の振り返り

- ホテリング理論によって異常検知アルゴリズムを実装してください。
- 局所外れ値因子法（LOF法）によって異常検知アルゴリズムを実装してください。

第11回：「アソシエーション分析」の振り返り

- POSデータに含まれる製品の分布を可視化してください。
- 条件部(lhs)、結論部(rhs)、support（支持度）、confidence（確信度）を表示してください。
- 条件部(lhs)、結論部(rhs)、support（支持度）、confidence（確信度）に加えてリフト(lift)を表示してください。

第12回：「主成分分析」の振り返り

- 2次元のダミーデータを作成し、主成分分析を実施してください。
- アイリスデータで主成分分析を実施してください。

第13回：「グラフ分析」の振り返り

- NetworkX（グラフ/ネットワーク理論系の計算ライブラリ）から空手クラブの友人関係ネットワークを読み込み、図示してください。
- 次数中心性（グラフの頂点に接合する辺の数）の大きな人物を表示してください。
- 媒介中心性（当該ノードを通る経路数）の大きな人物を表示してください。
- ページランク（Googleがページの重要度を図るために使用していることで知られている指標）の大きな人物を表示してください。

第14回：「テキストマイニング」の振り返り

- 解析したい口コミデータを収集してください。
- 口コミを形態素解析して特定の品詞のみ抽出し、半角スペースで連結した文字列を作成する処理を実装してください。
- WordCloudを作成してください。

2019 年度「専修学校による地域産業中核的人材養成事業」
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

2019 年度「専修学校による地域産業中核的人材養成事業」
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

データマイニング教材

令和２年２月

学校法人電子学園（日本電子専門学校）
〒169-8522　東京都新宿区百人町 1-25-4
TEL 03-3369-9333　FAX 03-3363-7685