COM実験（2017/9/28, 10/5）

櫻井彰人

実験の目的

- データの（客観的、自動的）分析を試みる、その基礎を知る

レポートについて1

- レポートに報告して戴きたいことは次の点です。
- スライド中、実験1及び「実験2または実験3」の「実験手順」となっている部分に書かれている手順に従い、実験を行い、その結果（自分が行ったこと、得られた結果）とそれに関する考察。
  - 実験1「文字認識」
    - 実験は、かなりうまくいくように作られています。「作られている」のはどこかについても考察して下さい（文字を手作りして貰います）
  - 実験2「歌詞の分類」
  - 実験3「ドル円レートの予測」
- 「更なる実験と考察」があれば、大歓迎
- 感想

レポートについて2

- 言うまでもないことですが、他人のレポート・著作物等を写してはいけません。自分の独力で作成してください。
- 他人の著作物からの引用には、出典を明記してください。
- 締め切りは、2週間後の木曜日一杯です。
  - 紙媒体と電子的な方法による提出を行って下さい。
    - 電子的には、keio.jp から提出して下さい（第六回）
  - 但し、事情がある場合には、柔軟に対応します。遅れる事情・理由を説明して下さい。

資料等

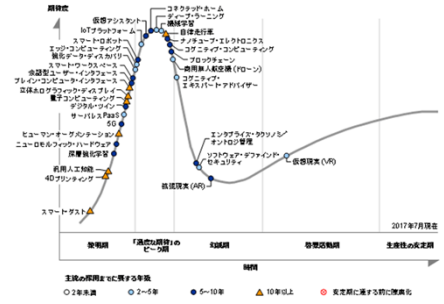
COM実験資料

- 課題の説明
- Rプログラム例
- 実験用データ
- 文字自作用プログラム
- Weka入門
- 「Weka入門」用データ
- Wekaダウンロードサイト（文字化け等については上記Weka入門を参照のこと）
- R入門

では、本論に

といいつつも背景を少し

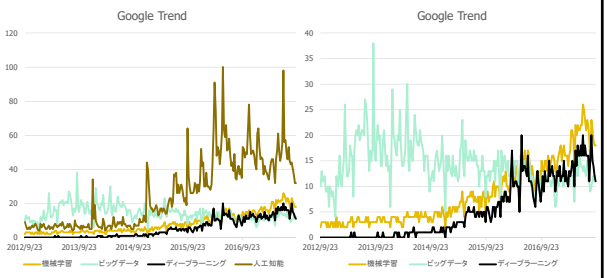
図1. 先進テクノロジのハイプ・サイクル：2017年



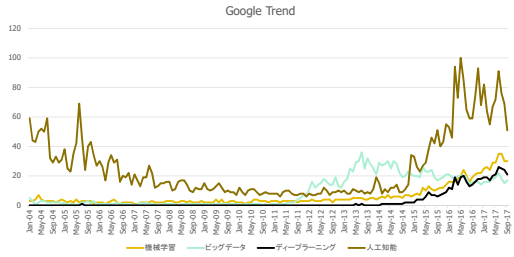
出典：ガートナー（2017年8月）

https://www.gartner.co.jp/press/html/pr20170823-01.html

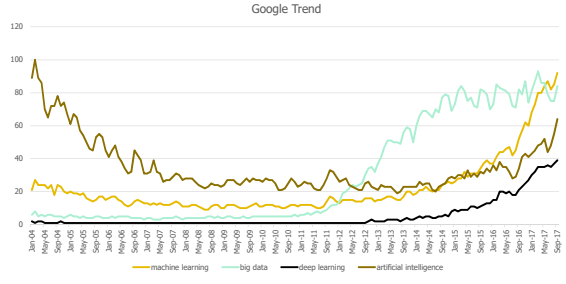
機械学習、ビッグデータ、他



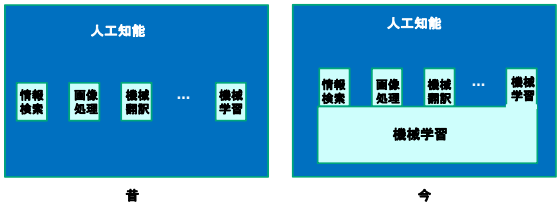
もう少し長い目でみると



英語でみると



機械学習の位置づけ



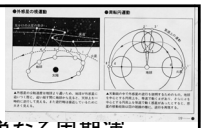
12

では、本論に

予測をしたい

- 人間は、太古の昔でも、予測をしていました。
- 有名なのは、季節の予想です
  - カレンダーさえないのですから。
  - 農耕をするためには、種まきを初めとして、「いつすべきか」が大切
    - 各時点の気温なんて当てにならない
- 天体観測は、極めて大切でした。
  - それができた人は、それだけで、王になれた

予測は難しい

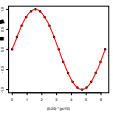


- 天体の運行は、モデルが悪くても（単なる周期運動、よくても天動説）、結構よくあたる。
  - （当時の人が必要とする精度であれば）単純な物理法則で高精度で近似可能な動きをする物体であるから。
- 多くの現象は、裏にある現実世界が複雑すぎる。
  - 複雑すぎて、一般には予測不可能。
- 多くの現象は、ほとんど確率的。
  - 何回も繰り返せば、当たる確率を高くできるかというとそうでもない。

それでも予測する

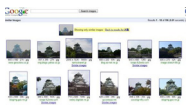
- それでも、人間は予測をする。
- 現実が複雑でも現象は（比較的）単純なこともある。
  - 「単純だと信じている」という方が正しいかも。
  - 経済予測なんて、その典型。
  - 専門家がデータを駆使し、予測するだけでなく、素人や政治家が予測する。

コンピュータに出来ること



見かけ上の類似を探す

01745329	→	1736482
03490659	→	3420201
05235988	→	5000000
06981317	→	6427876
08726646	→	7660444
10471976	→	8660254
12217305	→	9396926
13962634	→	9848078
06981428	→	???



規則を探す

x		y
01745329	→	1736482
03490659	→	3420201
05235988	→	5000000
06981317	→	6427876
08726646	→	7660444
10471976	→	8660254
12217305	→	9396926
13962634	→	9848078

x/10000000 → sin() → *10000000

06981428 → ???

データの構造

ここで、データとは

- 数字の列
- 文字の列（言葉というべき）。言語
- 写真の集まり
- 絵画の集まり
- 行動の表現（って何だろう？）の集まり
- 音の表現（って何だろう？）の集まり

その昔
商売に結び付けた人がいた

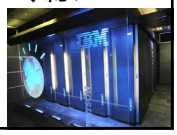
- 米国の小売業の人たち
  - 膨大なデータがあった
  - しかし、あっただけ。
  - 何かできないか？
- コンピュータサイエンスの研究者
  - 機械学習という手法があるよ
- 一見すると（よく見ても）全く違う。
  - 双方に努力は必要であった

そして

- 非常にたくさんのアルゴリズムが開発されてきた
- たくさんありすぎて、説明できない（私も全部知っているわけでは、当然、ない）。
  - というわけで、説明しません。
- 実験では、道具を使って、体感することで我慢してください。
  - 申し訳なし
- もっと知りたい場合には、是非、私の講義を聴いてください。
  - シラバス以外に、「予測」を強調する予定です（まだ準備できていませんが）

典型的学習の実例1 IBM's Watson

- 米国のクイズ番組Jeopardy!（ジョパディ!）に挑戦し、2ゲームを通じて、最高金額を獲得した(2011年2月16日（米国時間））。
- 知識は学習手法を用いて蓄積
  - 100万冊の本を読むのに相当する自然言語で書かれた情報
- ラック10本分、総メモリー容量15TB、総プロセッサー・コア数は2,880個



最近非常に活用されているGPUである 1080ti であれば、CUDAコア数で3584個（誤解なきよう）、メモリは11GB である

学習の実例2 コンピュータ将棋

- よく知られるようになったのは、渡辺竜王 vs. Bonannza戦（大和証券杯特別対局,2007年）。



- 第2回将棋電王戦の対戦成績はコンピュータ側が3勝1敗1持将棋（2013年4月20日）



- 電王戦タッグマッチ2014, 2015年春「将棋電王戦FINAL」, 2016年「第一期電王戦」 http://denou.jp/2016/

http://blogs.yahoo.co.jp/tannowa/51252262.html
http://news.mynavi.jp/articles/2013/04/21/denousen/
http://www.yomiuri.co.jp/zoom/MM20070321222909803M0.htm

コンピュータ将棋（続）

- Ponanza が、2017年第2期電王戦で佐藤天彦名人と対局し、第1局（4月1日）は71手、第2局（5月20日）は94手で勝利



http://number.ismcdn.jp/mwimgs/1/a/-/img_1ac64a7802fe03f9fed1cee4c9f40431161320.jpg
http://ascii.jp/elem/000/001/463/1463357/ph00_1000x667.jpg
http://www.sankei.com/images/news/170520/wst1705200079-p3.jpg

学習の実例3 コンピュータ囲碁

- AlphaGo の衝撃
  - Google傘下（2014年買収）のDeepMindが開発

Mastering the game of Go with deep neural networks and tree search
Nature 529, 484–489 (28 January 2016)

- 2016年3月9日、トップ棋士・李世乭(イ・セドル)九段と対戦し、4勝1敗
- 2017年5月23～27日、AlphaGoと中国最強棋士柯潔(か・けつ)九段が対戦し、AlphaGoが3戦3勝（The Future of Go Summit）

データの例

Make	Model	Year	Head inj. c.	Chest decel	L. Leg	R. Leg	D/P	Protection
Acura	Integra	87	599	35	791	262	Driver	manual belts
Acura	Integra RS	90	585		1545	1301	Driver	Motorized belts
Acura	Legend LS	88	435	50	926	708	Driver	d airbag
Audi	80	89	600	49	168	1871	Driver	manual belts
Audi	100	89	185	35	998	894	Driver	d airbag
BMW	325i	90	1036	56	865		Driver	d airbag
Buick	Century	91	815	47	1340	315	Driver	passive belts
Buick	Elect. Park	88	1467	54				
Buick	Le Sabre	90		33				
Buick	Regal	88	880	54				
Cadillac								

萼片長	萼片幅	花弁長	花弁幅	種別
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa

通算	曜日	室温	朝夕のアルコール量	血圧(mmHg) (LOW)	(HIGH)
0	火	18	なし	107	152
1	水	20	少々	78	135
2	木	20	少々	92	133
3	金	20	少々	87	136
5	日	20	少々	86	134
6	月	20	適度	90	134
7	火	18	少々	87	134
8	水	18	少々	104	149
9	木	20	少々	83	136
10	金	20	適度	94	131

Date	Open	High	Low	Close	Volume	Adj. Cl	YearWeek
27/11/2000	53.6875	54.5156	51.0312	51.25	40198100	51.250	200049
28/11/2000	51.9375	53.1875	50.625	51	52037000	51.000	200049
29/11/2000	51.3125	53	50.3125	51.6875	55316000	51.688	200049
30/11/2000	50.1875	50.9375	45.1875	47.875	10840500	47.875	200049
01/12/2000	49.1875	51.625	47.25	48.5	70468000	48.500	200049
04/12/2000	49.0625	49.5625	45	45.8125	9501200	45.813	200050
05/12/2000	47.75	52.125	47.3125	52.125	90848900	52.125	200050
06/12/2000	52	53.5625	51.2656	51.4375	71419200	51.438	200050
07/12/2000	50.3125	51	49	49.9375	46448400	49.938	200050
08/12/2000	51.9375	53.25	51	52.375	55400200	52.375	200050
11/12/2000	52.875	55.75	52.625	54.8125	78621500	54.813	200051
12/12/2000	54.75	55.125	53.3125	54.375	39485300	54.375	200051
13/12/2000	55.1875	55.25	50.8125	51.125	54330600	51.125	200051
14/12/2000	51.0625	52.5625	50.875	50.9375	46244400	50.938	200051
15/12/2000	50.0625	50.1875	47.125	48.1719	100237900	48.172	200051
18/12/2000	49	50.125	42.3125	42.9375	126032400	42.938	200052
19/12/2000	43	46	41.5	41.75	99018800	41.750	200052

機械学習での規則の表現

- 条件文
  - IF むにゃむにゃ THEN あれこれ
  - 信頼性は ?? %
- 決定木
  - 次のスライド
- ニューラルネットワーク
- ほかにもいろいろ

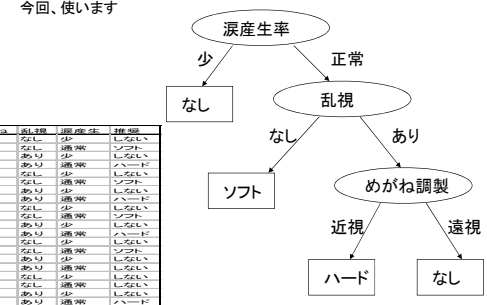
If-then 規則

If 涙産生 = 少 then 推薦 = しない
If 年齢 = 若い and 乱視 = なし and 涙産生 = 通常 then 推薦 = ソフト
If 年齢 = 老眼以前 and 乱視 = なし and 涙産生 = 通常 then 推薦 = ソフト
If 年齢 = 老眼 and めがね = 近視 and 乱視 = なし then 推薦 = なし
If めがね = 遠視 and 乱視 = なし and 涙産生 = 通常 then 推薦 = ソフト
If めがね = 近視 and 乱視 = あり and 涙産生 = 通常 then 推薦 = ハード
If 年齢 = 若い and 乱視 = あり and 涙産生 = 通常 then 推薦 = ハード
If 年齢 = 老眼以前 and めがね = 遠視 and 乱視 = あり then 推薦 = しない
If 年齢 = 老眼 and めがね = 遠視 and 乱視 = あり then 推薦 = しない

年齢	めがね	乱視	涙産生	推薦
若い	近視	なし	少	しない
若い	近視	なし	通常	ソフト
若い	近視	あり	少	しない
若い	近視	あり	通常	ハード
若い	遠視	なし	少	しない
若い	遠視	なし	通常	ソフト
若い	遠視	あり	少	しない
若い	遠視	あり	通常	ハード
老眼以前	近視	なし	少	しない
老眼以前	近視	なし	通常	ソフト
老眼以前	近視	あり	少	しない
老眼以前	近視	あり	通常	ハード
老眼以前	遠視	なし	少	しない
老眼以前	遠視	なし	通常	ソフト
老眼以前	遠視	あり	少	しない
老眼以前	遠視	あり	通常	しない
老眼	近視	なし	少	しない
老眼	近視	なし	通常	しない
老眼	近視	あり	少	しない
老眼	近視	あり	通常	ハード
老眼	遠視	なし	少	しない
老眼	遠視	なし	通常	ソフト
老眼	遠視	あり	少	しない
老眼	遠視	あり	通常	しない

決定木

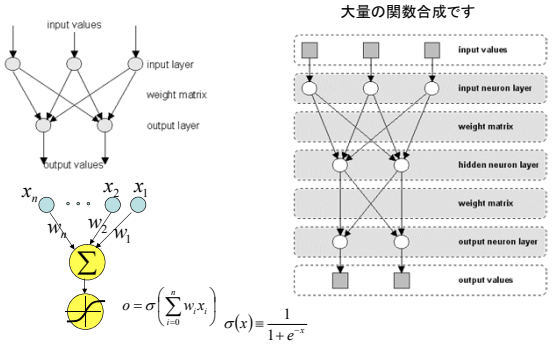
今回、使います



ニューラルネットワーク

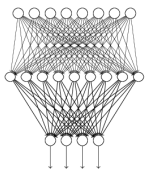
今回、使います

大量の関数合成です

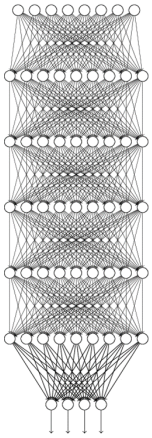


今回、使いません

Deep Neural Network



学習アルゴリズムの工夫
データの大量化
ハードウェアの進歩・安価化



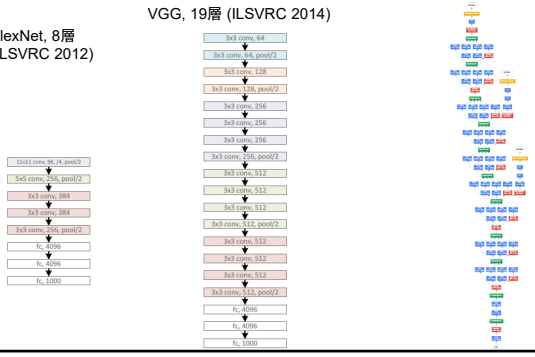
ちょっと脱線

層数の増加

AlexNet, 8層
(ILSVRC 2012)

VGG, 19層 (ILSVRC 2014)

GoogleNet, 22層 (ILSVRC 2014)



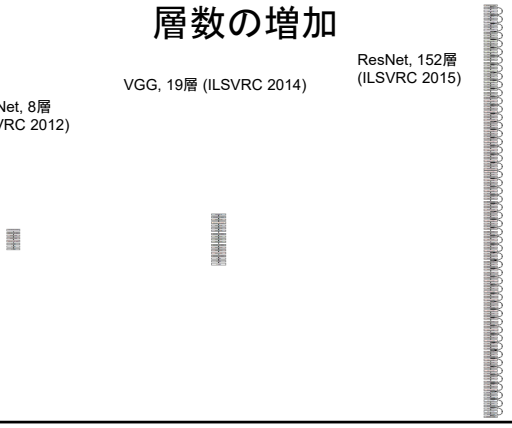
ちょっと脱線

層数の増加

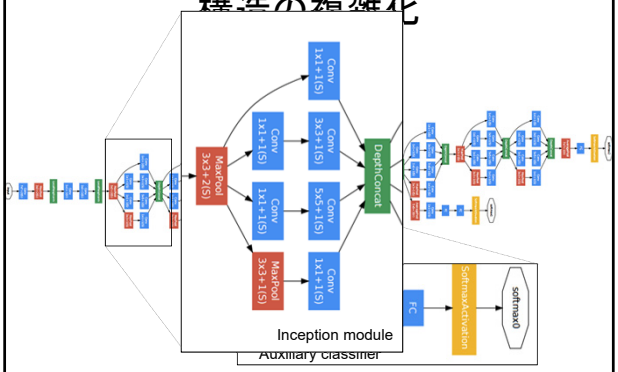
AlexNet, 8層
(ILSVRC 2012)

VGG, 19層 (ILSVRC 2014)

ResNet, 152層
(ILSVRC 2015)



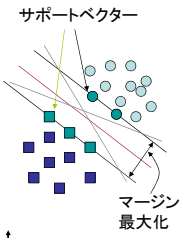
構造の複雑化

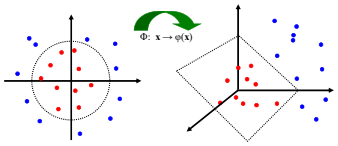


今回、使います

Support Vector Machine (SVM)

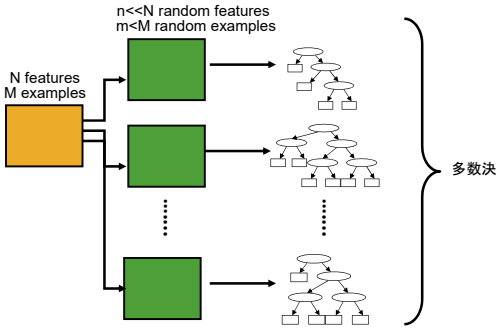
- SVM は、分離超平面周囲のマージンを最大化する.
  - ラージマージン分類器ともいう
  - 決定関数はサポートベクターと呼ばれる訓練データによって完全に定まる.
- カーネル化: もともとの特徴空間は、いつでも、ある高次元特徴空間に写像すれば、線形分離可能となる:





今回、使います

Random Forest



機械学習での結果表現

- データマイニングならなお更
- 結果を表現する必要が(たいていは)ある。
  - 予測だけを要求されれば、その限りではない。
  - 一般に結果だけを要求されれば、その限りではない。
  - しかし、「説明責任」があるといわれるのは、世の常
  - そこで、「分かりやすく」表現する必要があることがある。
- しかし、(予測等の)「精度」と「分かりやすさ」はtrade-offの関係にある
- 機械学習では両方用意している。例えば、
  - 決定木:分かりやすさ
  - SVM, NN, Random Forest: 精度や柔軟性

まとめ

- 対象
  - 数値、記号で表現できるもの
    - e.g. 言葉、画像、音、数値、
- 目的と考え方
  - 予測、推測する。売り物は「予測値」
    - 人間が明確には記述できない対象（予測・推測規則）
  - ノイズと対象に分離し、対象を「記述する」。「記述」が売り物
    - ノイズ: ランダム、無相関
    - 対象: 構造がある
- 道具・手段
  - 統計、人工知能、機械学習！！

「表」にできるもの

実際面での注意

- ツールは山ほどありますが、実用的にするには、「特徴量」をうまく選ぶことが大切。
- 「特徴量」とは、データから計算されるある量で、機械学習に用いる
  - ダメな特徴量:
    - 風邪か否かを診断するのに、身長を使う
    - ある日の株価予測: 雲の量を用いる（尤も晴雨が影響するという説あり）

「ダメ」かどうかは誰が決めるのだろう？
コンピュータが決めてくれたら、楽だろうなぁ

- 技法よりは特徴量
  - どんな特徴量をとるかで勝負はかなり決まる
  - 特徴量: 対象から計算される値
    - 一般には、複数個組み合わせると、対象が属するグループが決まるとよい。
- 特徴量の学習（発見・作成）は興味深いテーマ

Deep learning で結構できるということが知られている

実験

- 機械学習ツールとして Weka を用いる
  - 別のスライドへ
  - 少しRも使います
- 実験1: 文字識別
  - ためしに、数字。
- 実験2: 歌詞の50音分布から、歌のジャンルの推定
  - 自分でデータが作れる。
- 実験3: ドル円レートの予測

実験1：文字認識

- 実世界で（昔から）活躍中
  - 高速道路の料金所でナンバープレートを読む
    - Nシステムもそうらしい

Wikipediaより

  - 郵便番号自動読み取り（「区分け」と「配達順序」が重要）。手書きの住所も読む。高速。
- そして
  - Scannerに付属するのは当たり前
  - Google ドライブにも光学式文字認識。
    - 歴史的理由で Optical Character Recognition という

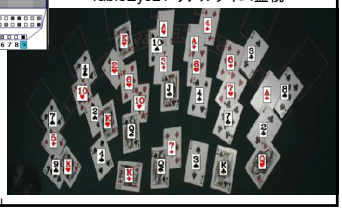
ちょっと昔の文字認識応用

人口約150万人

WSJは、公記録法に基づく請求を通じてカリフォルニア州リバーサイド郡の保安官事務所から2年分のデータを入手した。2010年9月10日から2012年8月27日までの期間に保安官事務所のカメラがナンバープレートをスキャンした回数は約600万回に上る。

http://jp.wsj.com/IT/node_522948

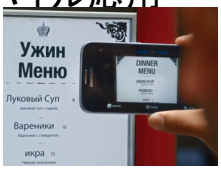
TableEye21-リアルタイム監視



http://www.gizmodo.jp/2011/09/post_9428.html

ちょっと昔のモバイル応用

- 手軽に



https://www.youtube.com/watch?v=0zKU7jDA2nc

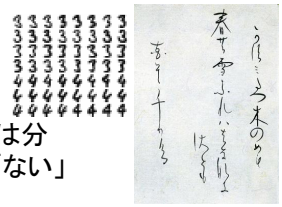
- Universal access

http://www.thepotteries.org/walks/fenton1/7.htm

http://www.afb.org/afbpress/pub.asp?DocID=aw070605

文字認識は結構難しい

- 人間なら、崩し字でなければ、簡単だと思う
- けれども、平仮名の読み方を、日本語を勉強したことのない人に教えることを想像してみてください。

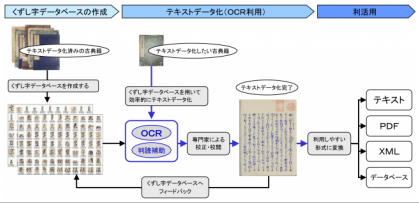


- 何が難しいか？
- 規則が書けない！（実は分からないのかも、実は「ない」のかも）

崩し字: 一つの対応

- 規則が書けなければ、「パターン」を見つけよ
  - 規則の主要部分を「パターン」にするのだが

凸版印刷、江戸期以前のくずし字を
高精度でテキストデータ化する新方式OCR技術を開発
～江戸期以前のくずし字が80%以上の精度でOCR処理可能に～



実験に入る前に

- 現在の Windows では、zip ファイルをフォルダと同じように扱うことができます。
- しかし、アプリケーションの中には、そのような扱いができないものがあります。
- Weka は「できない」アプリケーションです。
- ですから、ダウンロードしたzipファイルは、必ず解凍してください。

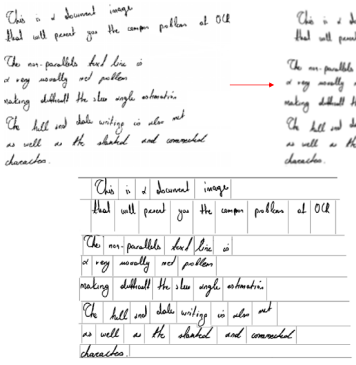
実験に入る前に

- 今回の実験程度の問題は、ニューラルネットワークのツールを使うと非常に簡単に、かつ精度よく解けます。
  - ディープラーニングのツール（Chainer, Caffe, Tensorflow 他）のインストールのチェック用に、もう少し大きな MNIST というデータを対象としたプログラムが提供されています。
- 今回はブラックボックスでない「決定木」を用いつつ、他の方法と比較してみようという目的です。

実験データの説明

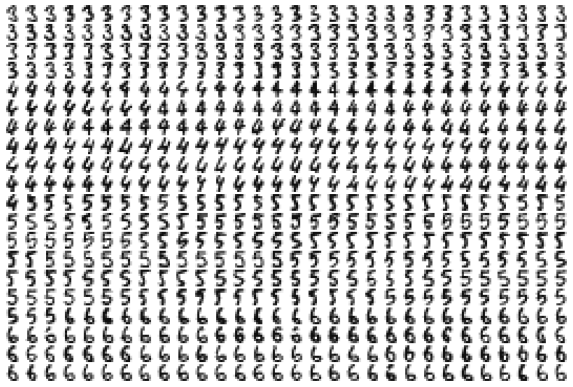
- 簡単な文字認識
  - 数字のみ。
- データの前処理（これが大変）済み
  - 分離（他の文字から分離）済み
  - 整形（大きさ、傾き、重心等）済み
- それでも、結構、難しそう。
  - 「数字」を知らない（！）人に区別の仕方を説明してみよう
- 本質「分類規則が表現できない」
  - データから得るしかない
- データのもと:
  UCI Machine Learning Repository
  Optical Recognition of Handwritten Digits Data Set

データの前処理



Giorgos Vamvakas

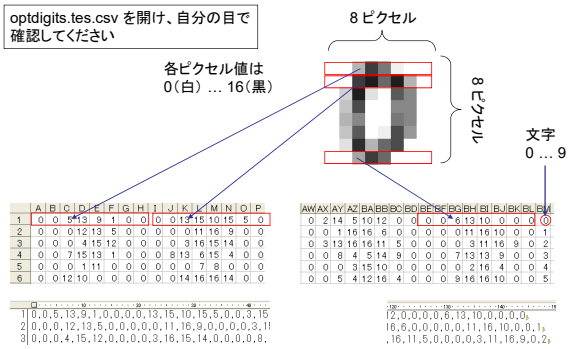
データを図にして並べてみました



UCI Machine Learning Repository

データ形式は単純

optdigits.tes.csv を開け、自分の目で確認してください



データ形式は単純

4. Relevant Information:

We used preprocessing programs made available by NIST to extract normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13 to the test set. 32x32 bitmaps are divided into nonoverlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates an input **matrix of 8x8** where each element is an integer in the range **0..16**. This reduces dimensionality and gives invariance to small distortions.

5. Number of Instances

optdigits.tra Training 3823
optdigits.tes Testing 1797

6. Number of Attributes

64 input+1 class attribute

optdigits.names.txt を開けて自分の目で確認してください

作業手順

optdigits.tra.csv

- データを入手（講義のサイトにあり）
  - ピクセル値を特徴量として用いる（最悪だけどね）
- arffファイルを作る（csvになっているので、ヘッダーをつけるだけ）。メモ帳で開こう。間違えないように。

```
@relation OptDigitsTraining
@attribute 00 real
@attribute 01 real
      ……
@attribute 06 real
@attribute 07 real
@attribute 10 real
@attribute 11 real
@attribute 12 real
      ……
      ……
@attribute 76 real
@attribute 77 real
@attribute class {0,1,2,3,4,5,6,7,8,9}
@data
以下はデータ
```

自由に

自由に。ただし、8 x 8 =64 個

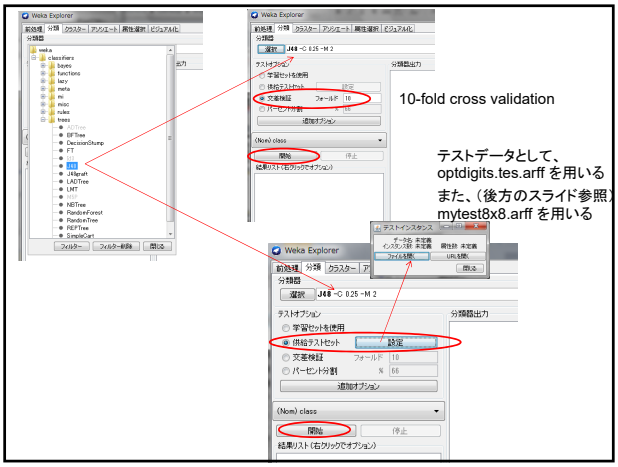
実験手順：学習と検証

- 決定木を作る
  - Weka: Trees にある J48
- 10-fold cross validation での正解率を求める
- テストデータとして、optdigits.tes.arff を用いたときの正解率を求める
- 決定木を見てみる。何か意味が見つかるか？多分見つからないと思う。何故だろう？
- 他の手法を試してみる。
  - SMO（support vector machineの一つ）
  - naïve Bayes
  - neural network
  - random forest
- それぞれの正解率、実行時間などを比べてみよう

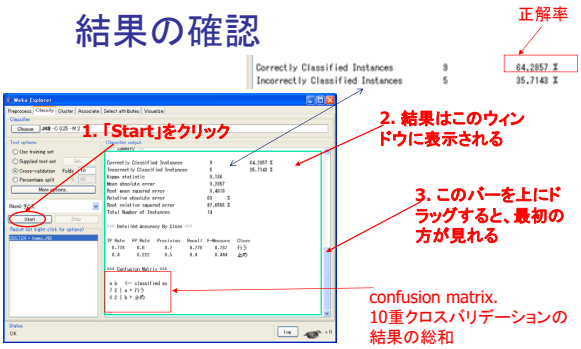
データを図にしたもの:
optdigits-test_image,
optdigits-train_image

実験手順：学習と検証（続）

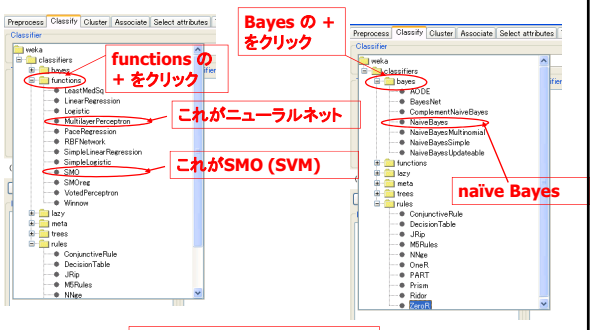
- 前の実験で得たモデル（学習結果）の一つ以上を、つまり複数個の手法を、各自が作成した文字（10文字以上）のデータに適用する。
  - Excel ファイルで作成（例: mytest8x8.xls）
  - csvファイルで保存（例: mytest8x8.csv）
  - 変換プログラム（conv8x8toArff.rb）で前述の書式のarffファイルに変換
    - ruby conv8x8toArff.rb mytest8x8.csv
  - できたarffファイル（例: mytest8x8.arff）をテストファイルとして、学習・テストする
  - さて、90%という精度はでるでしょうか？　でないとしたらどうしてでしょうか？　どうしたら出るようになりますか？それは妥当でしょうか？



結果の確認



「他の手法」の選び方



RandomForest は trees の中にある

各自作成する文字のファイル

optdigits-selfmade.zip に含まれています

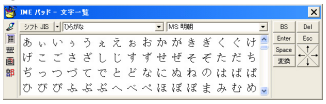
実験2: 歌詞の分類

- 楽曲のジャンルごとに、歌詞に使う言葉や言い回しは違う。
- 同様なことは、短歌でもいえる。
  - 水谷静夫先生(計量国語学の創始者. 計量国語学会は1957年(!)創立)が、その昔、白樺派とアララギ派の短歌が、そこに使われている単語を用いると、綺麗に分離されることを示した。
- 単語で分類するのは、ちょっと手間なので(準備がネ)、モーラ(音節)で分類できるか試してみよう
  - 実はモーラも少し面倒なので、「50音」を用いる。

注: 実は、MeCab 他のツールを使えば、それほど困難なく、単語を用いて分類ができる。

データの説明

- 童謡とJ-POPそれぞれ10曲の歌詞を「50音」にわけ、その頻度を数える。
- 文字コードの並びが50音ではないので並び替えたいのであるが、少々面倒
  - 「あいうえお」は纏めたい etc.
  - Excelファイルを用意しました

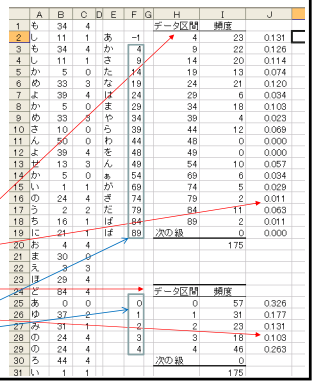


- 頻度は正規化する(頻度の総和が1となるように)。歌によって長さが異なるからである。

実験手順: データの準備1

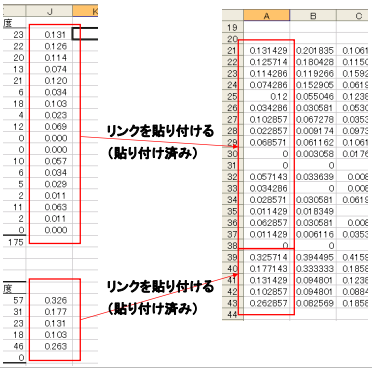
列Aに平仮名で一文字ごと歌詞を書く。
そうすると、列Bには列Aの文字を五十音のコード（自作）に変換するマクロが書かれているので、列Bには、値として、そのコードが得られる。
列Cには「あいうえお」の区別が得られる。

ヒストグラムを作るために、頻度を数える。
列E,Fはヒストグラム作成用のデータである。
Excel メニューで、「データ」⇒「データ分析」⇒「ヒストグラム」とし、ヒストグラムを作成する。
正規化した頻度が得られる
「あいうえお」を数えて、そのヒストグラムも作成する。
正規化した頻度が得られる。

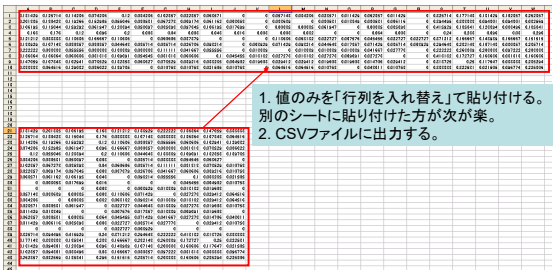


「excel ヒストグラム」で検索してみましょう

実験手順: データの準備2



実験手順: データの準備3



1. 値のみを「行列を入れ替え」て貼り付ける。別のシートに貼り付けた方が次が楽。
2. CSVファイルに出力する。

実験手順: データの準備4

- 童謡10曲、J-POP10曲について、データを作成する。
- arff ファイルとする。

ここまでは作成済み
童謡.xls
J-POP.xls

  - 右のような、ヘッダを作る
  - 各データ（一行に一曲、全20行）の最後に、童謡なら「,0」、J-POPなら「,1」をつける。

```
@relation 'songs'
@attribute あ real
@attribute か real
@attribute さ real
@attribute た real
@attribute な real
@attribute は real
@attribute ま real
@attribute や real
@attribute ら real
@attribute わ real
@attribute ざ real
@attribute あ real
@attribute が real
@attribute ぎ real
@attribute ば real
@attribute A real
@attribute I real
@attribute U real
@attribute E real
@attribute O real
@attribute class {0,1}
@data
```

実験手順: 実験と検証

- Weka を用いてみる
  - J48, SMO, naïve Bayes, nn, random forest など
- 正解率が高いかな？
- 決定木がまともにできているかどうか、わかりますか？
- では、皆さんご自分で、
  - 童謡・J-POPに10曲ずつ追加するか、
  - 新たなジャンルで、最低10曲ずつデータを作成して

実験を行ってください。

協力して行っても結構です。
1グループは最大三人としましょう。
協力者の名前を明記して下さい。

補足

J-POP.xls と 童謡.xls を一つにしたファイルを作りました。
童謡とJ-POP.xls です。下に説明するように 童謡とJ-POP.csv と 童謡とJ-POP.arff とを作成し、data2016.zip に含めてあります。

(1)「まとめ(童謡)」シートは、童謡.xls の「まとめ」シートと同じですが、
class の値である 0 を右端の列として付加してあります。
(2)「まとめ(J-POP)」シートは、J-POP.xls の「まとめ」シートと同じですが、
class の値である 1 を右端の列として付加してあります。
(3) 童謡とJ-POP.xls の「全体まとめ」シートは、「まとめ(童謡)」シート
と「まとめ(J-POP)」シートへのリンクだけで構成されています。従って、変更は不要です。
(4)「全体まとめ」シートを csv ファイルとして、保存します。
(5) その後、ツールで（または手で）、arff ファイルに変換します。

実験3 ドル円レートの予測

- FX: 外国為替証拠金取引
  - 証拠金（保証金）を業者に預託し、主に差金決済による通貨の売買を行なう取引
- FXで利益を上げることができるのだろうか？
  - 仲介業者の取り分（スプレッド）は小さい。宝くじとは違う
  - しかし、典型的なギャンブル。ゼロサムゲーム（厳密には、投資家には利子の出入りがあるのでゼロサムではない）。勝者がいれば、敗者がいる。そして、敗者が圧倒的に多い（80-20の法則、冪法則）と思う。
- 値動きは、基本的には、ランダムウォークのはず
  - すなわち、予測不能のはず。
- しかし、少し、試してみよう

実験データの説明

- 米ドル（USD）を日本円（JPY）で売買する
- 価格(?)の単位は 0.01 円
  - 最近は 0.005円や0.001円 刻み
- 分単位の値動きを見る
  - 分足という
    - 実際はティックデータ
- 次のどちらか一方
  - ARモデルを用いる
  - 機械学習的手法

東京金融取引所

ARモデル

- USD/JPYの分足を用いる
  - まず、（いつでもよいのだが）2017年8月31日にしよう
  - Forex Tester というサイトのものを用いる。時刻は GMT (夏時間なし). データの正確性の保証はない。
  - 24時間中の、一分毎、Open（始値）、High（高値）、Low（安値）、Close（終値）が時系列に記されている。
  - ある「分」の「終値」を予測する
  - 単純に、それまでの「終値」を用いて予測しよう
  - （まずは）5分前からの各「分」の収益を用いよう

USDJPY-20170831.txt

```
1 <TICKER>,<DTYYYYMMDD>,<TIME>,<OPEN>,<HIGH>,<LOW>,<CLOSE>
2 USDJPY,20160831,000000,102.96,102.96,102.95,102.96
3 USDJPY,20160831,000100,102.97,102.97,102.97,102.97
4 USDJPY,20160831,000200,102.97,102.97,102.97,102.97
5 USDJPY,20160831,000300,102.97,102.97,102.96,102.96
```

ここから

ARモデルの簡単な説明

- 時系列 $\cdots, X_{-1}, X_0, X_1, X_2, \ldots, X_T, \ldots$ のモデル（数式表現）の一つ
- AR(p) は

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \cdots + \alpha_p X_{t-p} + \varepsilon_t$$

ただし、 $\varepsilon_t \sim N(0, \sigma^2)$

正規分布

  - なお、このようにモデル化してよい条件等考慮すべき点はあるが、今回は目を瞑る。

FXデータの性質

- FX rateや株価の動きをモデル化するに当たって、値そのものではなく、その比（前日比や1分前との比）を考えるのが妥当である。
- しかし、今回は、値そのものと比の対数（1分前との比の対数）を考える。

$$X_t,\ X_t - X_{t-1},\ \log X_t / X_{t-1} = \log X_t - \log X_{t-1}$$

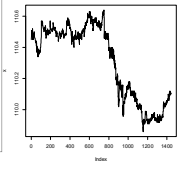
作業手順

USDJPY-20160831.txt から closeを全部

- ツールとしてはRを用いる（Wekaには機能がないので）
- 2016年8月31日のデータを得る。
  - データを読み込み、USDJPYの終値のみを取り出す。
- 図示する
- arima を用いる。次数（p,i,q）のうち、i=q=0 とすれば、AR(p)となる。

皆さんのフォルダを指定

```
setwd("D:/R/")
# Read a file,
x.tmp <- read.csv("USDJPY-20170831.txt", header=T)
# pick up UDSJPY rows and then select X.CLOSE. columns,
x     <- subset( x.tmp, X.TICKER. == "USDJPY" )$X.CLOSE.
# plot it,
plot( x, type="l")
# and fit AR(2) model to the data
(fit2 <- arima(x, c(2, 0, 0)))
```

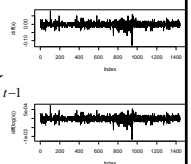


作業手順（続）

- なお、次の値は簡単な式で表せる

$$X_t - X_{t-1},\ \log X_t / X_{t-1} = \log X_t - \log X_{t-1}$$

```
par( mfrow=c( 2, 1 ) )
plot( diff(x), type="l" ); plot( diff(log(x)), type="l" )
 par( mfrow=c( 1, 1 ) )
```



- arima の結果出力中、係数は右のように
なっている。その意味は次の通り

```
Coefficients:
          ar1     ar2  intercept
      0.8716  0.1264   110.3100
s.e.  0.0262  0.0262     0.1471
```

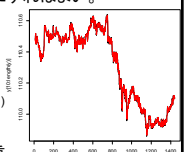
$$X_t - c = ar1(X_{t-1} - c) + ar2(X_{t-2} - c) + \varepsilon_t$$

ただし、c はinterceptである110．3100を表す
s.e. はstandard error である。

作業手順（続々）

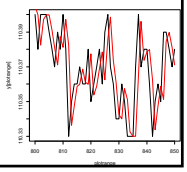
- テストデータに対する予測値を求めるには次のようにすればよい。

```
# Read a test data file
y.tmp <- read.csv("USDJPY-20170831.txt", header=T)
y     <- subset( y.tmp, X.TICKER. == "USDJPY" )$X.CLOSE.
# Prediction based on fit2 <- arima(x,c(2,0,0)) will be in y.ar
y.ar <- array( 0, dim = c( length(y) ) )
int <- fit2$coef["intercept"]
for ( i in (2+1):length(y) ) y.ar[i] <- int + coef(fit2)[1:2] %*% (y[(i-1):(i-2)] - int )
plot(10:length(y), y[10:length(y)], type="l")
 lines(10:length(y), y.ar[10:length(y)], col=2)
```



- 真値である黒線の上に、予測値である赤線がほぼ重なっているようにみえる。
- しかし、そうではないことは、次のようにして分かる

```
plotrange= 800:850
plot(plotrange,y[plotrange],type="l"); lines(plotrange,y.ar[plotrange],col=2)
```



この予測は適切か？
テストデータとして他の日のデータを用いたらどうなるか？
比の対数を用いるとどうなるか？

機械学習手法

USDJPY-20170831.txt

- USD/JPYの分足を用いる
  - まず、(いつでもよいのだが)2017年8月31日にしよう
  - Forex Tester というサイトのものを用いる。時刻は GMT (夏時間なし).データの正確性の保証はない。
  - 24時間中の、一分毎、Open（始値）、High（高値）、Low（安値）、Close（終値）が時系列に記されている。
  - ある「分の終値 」–「前の分の終値」（収益）を予測する
  - 難しい: 何を、予測の根拠に用いるか？ i.e. どんな特徴量を用いるか
  - （まずは）5分前からの各「分」の収益を用いよう

```
1 <TICKER>,<DTYYYYMMDD>,<TIME>,<OPEN>,<HIGH>,<LOW>,<CLOSE>,<VOL>
2 USDJPY,20170831,000000,110.45,110.45,110.45,110.45,4
3 USDJPY,20170831,000100,110.48,110.48,110.48,110.48,4
4 USDJPY,20170831,000200,110.51,110.51,110.51,110.51,4
5 USDJPY,20170831,000300,110.49,110.49,110.49,110.49,4
```

ここから

作業手順

済: USDJPY-20170831.txt
または USDJPY.zip 中の
20170831 分を全部

- 2017年8月31日のデータを得る。
- Excelファイルとし、各分について5分前からの収益を計算する。当「分」の収益も求める。
- 予測問題を簡単にするために、up or down を示す値をつける (上昇していたら +1, 下降していたら –1)
  - このファイルは用意してある。

USDJPY170831.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	<DTYYYYMMDD>	<TIME>	<OPEN>	<HIGH>	<LOW>	<CLOSE>	5 before	4 before	3 before	2 before	1 before	predict	updown	
2	20170831	0	110.45	110.45	110.45	110.45								
3	20170831	100	110.48	110.48	110.48	110.48								
4	20170831	200	110.51	110.51	110.51	110.51								
5	20170831	300	110.49	110.49	110.49	110.49								
6	20170831	400	110.5	110.5	110.5	110.5								
7	20170831	500	110.51	110.51	110.51	110.51								
8	20170831	600	110.5	110.5	110.5	110.5	0.03	0.03	-0.02	0.01	0.01	-0.01	-1	
9	20170831	700	110.49	110.49	110.49	110.49	0.03	-0.02	0.01	0.01	-0.01	-0.01	-1	
10	20170831	800	110.47	110.47	110.47	110.47	-0.02	0.01	0.01	-0.01	-0.01	-0.02	-1	
11	20170831	900	110.48	110.48	110.48	110.48	0.01	0.01	-0.01	-0.01	-0.02	0.01	1	
12	20170831	1000	110.49	110.49	110.49	110.49	0.01	-0.01	-0.01	-0.02	0.01	0.01	1	
13	20170831	1100	110.5	110.5	110.5	110.5	-0.01	-0.01	-0.02	0.01	0.01	0.01	1	
14	20170831	1200	110.5	110.5	110.5	110.5	-0.01	-0.02	0.01	0.01	0.01	0	0	
15	20170831	1300	110.52	110.52	110.52	110.52	-0.02	0.01	0.01	0.01	0	0.02	1	

実験手順: ファイルの準備

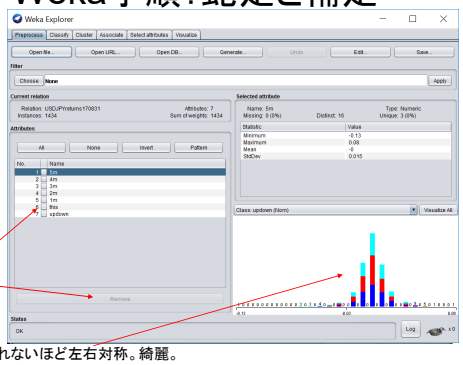
- 収益の部分のみを取り出し、csv ファイルを作る。
- arff のヘッダーをつけ、Weka用の arff ファイルとする。メモ帳なりエディタなりを使う方が間違いが少ない

```
1  @relation USDJPYreturns170831
2  @attribute 5m numeric
3  @attribute 4m numeric
4  @attribute 3m numeric
5  @attribute 2m numeric
6  @attribute 1m numeric
7  @attribute this numeric
8  @attribute updown {-1,0,1}
9  @data
10 0.03,0.03,-0.02,0.01,0.01,-0.01,-1
11 0.03,-0.02,0.01,0.01,-0.01,-0.01,-1
12 -0.02,0.01,0.01,-0.01,-0.01,-0.02,-1
13 0.01,0.01,-0.01,-0.01,-0.02,0.01,1
14 0.01,-0.01,-0.01,-0.02,0.01,0.01,1
15 -0.01,-0.01,-0.02,0.01,0.01,0.01,1
```

実験手順: 味見

- Weka を使う
  - 使うときに、当「分」の収益という属性は "Preprocess" で remove して下さい（次スライド）
  - trees にある J48: 決定木
  - functions にある Multilayer Perceptron: neural network
  - functions にあるSMO: support vector machine の一つ
  - Bayes にある naïve Bayes
- どれも正解率は 36~42%であろう。
  - 「上がる、下がる、同じ」の3値を当てるのに、40前後の正解率はまぁまぁか！と思わないで下さい。Confusion matrix をよく見て下さい。何が分りますか？

Weka手順：蛇足と補足



実験手順: 他の日との比較

- 別の日のデータではどうか？
- 8月22日、8月23日、8月24日で試してみよう
  - xls, csv, arffファイルは自分で作りましょう
- どうですか？

USDJPY170822.txt
USDJPY170823.txt
USDJPY170824.txt

- 予測するための情報が不足でしょうか？
- きっとそうでしょう。では、一分前の「分」の高値安値を含めてみましょう。
  - 8月31日のファイルが作ってあります。他の日のファイルも作り、試しましょう。
  - でもうまくいきません。

USDJPY170831A.xls

実験手順

- では、5分足（5分間一区切り）を試してみよう。
  - 1分足では、他の人の動きをみて動く（相関が発生する）ということが少ないので、動きはランダムになり、従って、予測できない。
  - しかし、5分程度みれば、データ間に相関が発生し、従って、予測可能となる可能性がある。
  - 本当か？

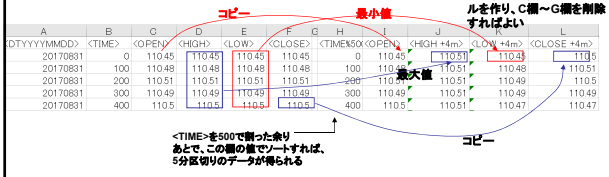
実際には逆で、収益の時間相関は20分ぐらいまでは存在すると報告されている。例えば
P. Gopikrishnan, et al. Scaling of the distribution of fluctuations of financial market indices, Physical Review E vol. 60, 5305 - 5316 (1999)

実験手順

- まず、5分ごとのデータにする前に、「5分間」の始値、高値、安値、終値を求める
- 次に、5分区切りのデータを抽出する。

USDJPY170831-5m.xls にペーストすればよい

H欄でソートし、H欄の値0以外の行は削除し、csvファイルを作り、C欄～G欄を削除すればよい



A	B	C	D	E	F	G	H	I	J	K	L
<DTYYYYMMDD>	<TIME>	<OPEN>	<HIGH>	<LOW>	<CLOSE>		<TIME+500>	<OPEN>	<HIGH +4m>	<LOW +4m>	<CLOSE +4m>
20170831	0	110.45	110.45	110.45	110.45		0	110.45	110.51	110.45	110.5
20170831	100	110.48	110.48	110.48	110.48		100	110.48	110.51	110.48	110.51
20170831	200	110.51	110.51	110.51	110.51		200	110.5	110.51	110.49	110.5
20170831	300	110.49	110.49	110.49	110.49		300	110.49	110.51	110.49	110.49
20170831	400	110.5	110.5	110.5	110.5		400	110.5	110.51	110.47	110.47

<TIME>を500で割った余り
あとで、この欄の値でソートすれば、
5分区切りのデータが得られる

実験手順

- Weka を用いてみる
  - J48, SMO, naïve Bayes, Random Forest など
- 正解率は34～44%かな。
  - 「上がる、下がる、同じ」がほぼ同比率ゆえ
  - やはり予測できないのか。
  - しかし、データ数が少ないからかもしれない。
- 8月22日~8月24日で試してみよう
  - 少しよいか？
  - しかし、別の日、例えば、8月14日~8月17日でテストをしてみたらどうだろうか？

USDJPY-20170814.txt
USDJPY-20170815.txt
USDJPY-20170816.txt
USDJPY-20170817.txt

実験手順

- 別のデータでテストする方法
- 学習データと同じ属性数（並び）のテストデータを用意する。
- 今回は、当「分」の収益という属性を持ったファイルしかないので、そのままではテスト用のデータファイルにはならない。そこで、当「分」の収益という属性を削除したデータファイルを作る。Weka でできる。

さらに試みたい人のために

- 2016年の8月分の データが用意してあります。

  USDJPY-20160801-20160831.txt

  一年違うと、事情が違うかもしれない

- さらに、10分足の長期データを用意しました。
  - 実はこちらですと、少し予測ができることが分かります。ただし、実用には程遠い。

  USDJPY-20160102-20160831-10m.csv
  USDJPY-20160102-20160831.txt

Weka補足: テストデータの指定