Computational Social Science: exploration

Tiphaine Viard

Features

The Data Mining Process

- Data collection
- Data Preprocesing
  - Feature extraction
  - Data cleaning
  - Feature selection and transformation
- Analytical processing and algorithms
- Data Postprocesing

Multidimensional Data

- Example:

Competitor Name	Swim	Cycle	Run	Total
John T	13:04	24:15	18:34	55:53
Norman P	8:00	22:45	23:02	53:47
Alex K	14:00	28:00	n/a	n/a
Sarah H	9:22	21:10	24:03	54:35

Triathlon results

- Example or Instance
  - data point, transaction, entity, tuple, object, or feature-vector
- Attribute or Feature
  - field, dimension

Instance Types

- Dense
  - red, white, Barcelona, 3, up
  - red, red, Barcelona, 4, down
  - black, white, Paris, 2, up
  - red, green, Paris, 3, down
- Sparse
  - 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  - 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  - 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  - 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  - 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  - 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Attribute Type

- Numerical
  - 0, 1, 3.43, 2.34, 4.23
- Categorical or Discrete
  - +, -
  - red, green, black
  - yes, no
  - up, down
  - Barcelona, Paris, London, New York
- Text Data: vector-space representation
  - The cat is black
- Binary: Categorical or Numerical

Exercise EDF scenario

You are working at EDF and you want to predict the energy consumption for tomorrow. What features do you use?

Exercise EDF scenario

You are working at EDF and you want to predict the energy consumption for tomorrow. What features do you use?

Note

The same data can be presented in different manners!

Example: a timestamp vs (year,day in the year, hour, minute)

Feature Extraction

- Sensor data (time series): Wavelets or Fourier Transforms

Feature Extraction

- Sensor data (time series): Wavelets or Fourier Transforms
- Image Data: histograms or visual words

Feature Extraction

- Sensor data (time series): Wavelets or Fourier Transforms
- Image Data: histograms or visual words
- Web logs: multidimensional data

Feature Extraction

- Sensor data (time series): Wavelets or Fourier Transforms
- Image Data: histograms or visual words
- Web logs: multidimensional data
- Network traffic: specific features as network protocol, bytes transferred

Feature Extraction

- Sensor data (time series): Wavelets or Fourier Transforms
- Image Data: histograms or visual words
- Web logs: multidimensional data
- Network traffic: specific features as network protocol, bytes transferred
- Text Data: remove stop words, stem data, multidimensional data

Data Cleaning

Data Cleaning

- Handling missing entries
  - Eliminate entries with a missing value
  - Estimate missing values
  - Algorithms can handle missing values
- Handling incorrect entries
  - Duplicate detection and inconsistency detection
  - Domain knowledge
  - Data-centric methods
- Scaling and normalization
  - Standardization: for instance i, attribute j:

$$z_i^j = \frac{x_i^j - \mu_j}{\sigma_j}$$

  - Normalization:

$$y_i^j = \frac{x_i^j - \min_j}{\max_j - \min j}$$

Feature Conversion

- Numeric to Discrete
  - Equi-width ranges
  - Equi-log ranges
  - Equi-depth ranges

Feature Conversion

- Numeric to Discrete
  - Equi-width ranges
  - Equi-log ranges
  - Equi-depth ranges
- Discrete to Numeric
  - Binarization: one numeric attribute for each value

Feature Conversion

- Numeric to Discrete
  - Equi-width ranges
  - Equi-log ranges
  - Equi-depth ranges
- Discrete to Numeric
  - Binarization: one numeric attribute for each value
- Text to Numeric
  - remove stop words, stem data, tf-idf, multidimensional data

Feature Conversion

- Numeric to Discrete
  - Equi-width ranges
  - Equi-log ranges
  - Equi-depth ranges
- Discrete to Numeric
  - Binarization: one numeric attribute for each value
- Text to Numeric
  - remove stop words, stem data, tf-idf, multidimensional data
- Time Series to Discrete Sequence Data
  - SAX: equi-depth discretization after window-based averaging

Feature Conversion

- Numeric to Discrete
  - Equi-width ranges
  - Equi-log ranges
  - Equi-depth ranges
- Discrete to Numeric
  - Binarization: one numeric attribute for each value
- Text to Numeric
  - remove stop words, stem data, tf-idf, multidimensional data
- Time Series to Discrete Sequence Data
  - SAX: equi-depth discretization after window-based averaging
- Time Series to Numeric Data
  - Discrete Wavelet Transform
  - Discrete Fourier Transform

Term Frequency-Inverse Document Frequency

- Term frequency
  - Boolean "frequencies"
    - $tf(t, d) = 1$ if t occurs in d and 0 otherwise;
  - Logarithmically scaled frequency
    - $tf(t, d) = 1 + log(f_{t,d})$, or zero if $f_{t,d}$ is zero;
  - Augmented frequency,

$$\mathrm{tf}(t, d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

- Inverse document frequency

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

- Term frequency-inverse document frequency

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

Feature conversion exercise

"What is the cure for Tuberculosis?" (in context of search engine)

Feature conversion exercise

“What is the cure for Tuberculosis?” (in context of search engine)

Predicting the next position of a car given the last positions?

Feature conversion exercise

"What is the cure for Tuberculosis?" (in context of search engine)

Predicting the next position of a car given the last positions?

Predicting disease from noisy input (body temperature, etc.)

Feature conversion exercise

"What is the cure for Tuberculosis?" (in context of search engine)

Predicting the next position of a car given the last positions?

Predicting disease from noisy input (body temperature, etc.)

Recognizing authors through their unique style?

Feature selection and transformation

- Sampling for Static Data
  - Sampling with Replacement
  - Sampling without Replacement: no duplicates
  - Biased Sampling
  - Stratified Sampling
- Reservoir Sampling for Data Streams
  - Given a data stream, choose k items with the same probability, storing only k elements in memory.

Reservoir Sampling

Reservoir Sampling

```
for every item i in the first k items of the stream
        do store item i in the reservoir
n = k
for every item i in the stream after the first k items of the stream
        do select a random number r between 1 and n
            if r ≤ k
               then replace item r in the reservoir with item i
            n = n + 1
```

Algorithm Reservoir Sampling

- For all $i > k$, i^{th} element chosen with probability $\frac{k}{i}$
- For all $j \geq k$ is chosen to be replaced with probability $\frac{1}{k} \cdot \frac{k}{i}$
- At the end, each element of the population has probability $\frac{k}{n}$ to be in the reservoir [out of scope, by induction]

Reservoir Sampling has complexity $\mathcal{O}(n)$

- *Algorithm L*, and runs in $\mathcal{O}(k(1 + \log \frac{n}{k}))$
- This is optimal

Feature selection and transformation

- Feature Subset Selection
  - Supervised feature selection
  - Unsupervised feature selection
  - Biased Sampling
  - Stratified Sampling
- Dimensionality reduction with axis rotation
  - Principal Component Analysis
  - Singular Value Decomposition
  - Latent Semantic Analysis

- Goal: **Principal component analysis** computes the most meaningful basis to re-express a noisy, garbled data set. The hope is that this new basis will filter out the noise and reveal hidden dynamics

More on that in a later course!

Feature selection is an important step

- Simplification of models

Feature selection is an important step

- Simplification of models
- Shorten training phase

Feature selection is an important step

- Simplification of models
- Shorten training phase
- Avoids the curse of dimensionality

Feature selection is an important step

- Simplification of models
- Shorten training phase
- Avoids the curse of dimensionality
- Reduce overfitting

Analytical processing and algorithms

- Attribute/Column Relationships
  - **Classification** : predict value of a discrete attribute
  - **Regression**: predict value of a numeric attribute
- Instance/Row Relationships
  - **Clustering**: determine subsets of rows, in which the values in the corresponding columns are similar
  - **Outlier Detection**: determine the rows that are very different from the other rows

The Data Mining Process

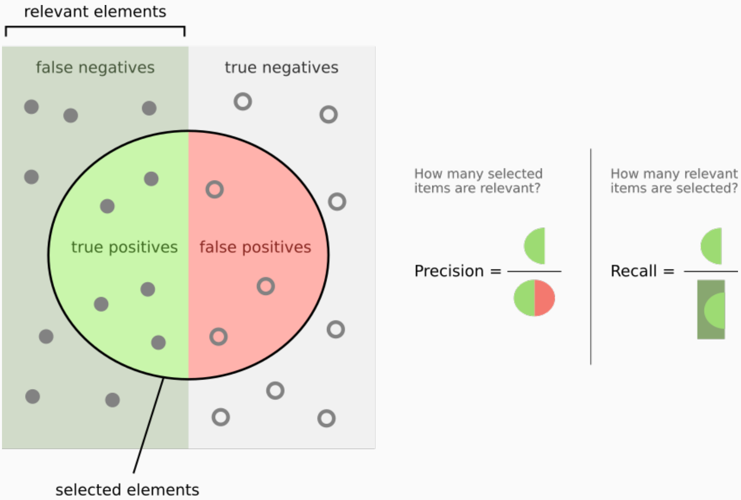
- Data collection
- Data Preprocesing
    - Feature extraction
    - Data cleaning
    - Feature selection and transformation
- Analytical processing and algorithms
- Data Postprocesing

Big Data Scalability

- Distributed Systems:
  - Hardware: Hadoop cluster
  - Software: MapReduce, Spark, Flink, Storm
- Streaming Algorithms
  - Single pass over the data
  - Concept Drift

Classifier evaluation

Precision and recall



wikipedia

What are the most important in these situations

What should be prioritized (true negative, false positive, true positive?) in these situations:

What should be prioritized (true negative, false positive, true positive?) in these situations:

- “You might like this movie” recommendation

What are the most important in these situations

What should be prioritized (true negative, false positive, true positive?) in these situations:

- "You might like this movie" recommendation
- Medical test (pregnancy / cancer / 50-50-cure)

What are the most important in these situations

What should be prioritized (true negative, false positive, true positive?) in these situations:

- "You might like this movie" recommendation
- Medical test (pregnancy / cancer / 50-50-cure)
- Danger prediction (pedestrian detection / hurricane alert)

Parameterized models

Most classification models come with a parameter that can be tuned between **precision** and **recall**.

Example: set of predictions with some "score".

Item	A	B	C	D	E	F	G	H	I	J
Prediction Score	97	91	85	75	63	51	42	32	18	7
Reality	1	1	0	1	1	0	0	1	0	0

Most classification models come with a parameter that can be tuned between **precision** and **recall**.

Example: set of predictions with some "score".

Item	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
Prediction Score	97	91	85	75	63	51	42	32	18	7
Reality	1	1	0	1	1	0	0	1	0	0

What are the precision and recall if we put the threshold at 50%?

Parameterized models

Most classification models come with a parameter that can be tuned between **precision** and **recall**.

Example: set of predictions with some "score".

Item	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
Prediction Score	97	91	85	75	63	51	42	32	18	7
Reality	1	1	0	1	1	0	0	1	0	0

What are the precision and recall if we put the threshold at 50%? 30% ?

Parameterized models

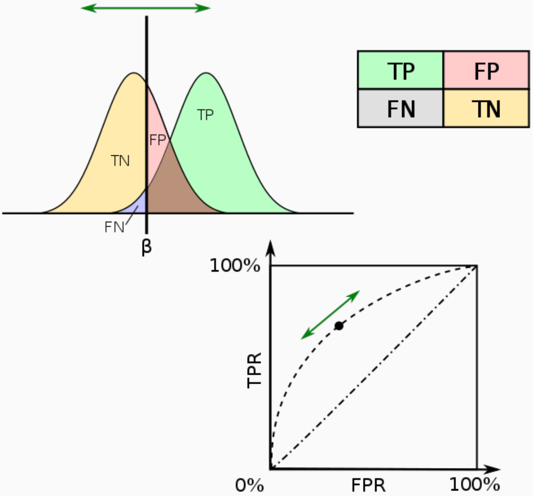
Most classification models come with a parameter that can be tuned between **precision** and **recall**.

Example: set of predictions with some "score".

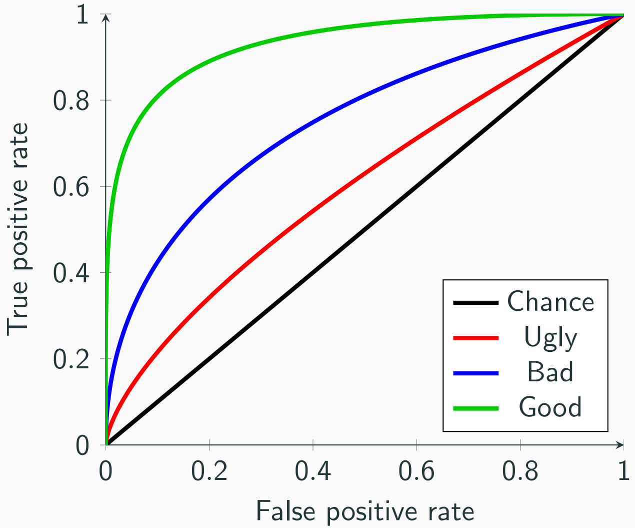
Item	A	B	C	D	E	F	G	H	I	J
Prediction Score	97	91	85	75	63	51	42	32	18	7
Reality	1	1	0	1	1	0	0	1	0	0

What are the precision and recall if we put the threshold at 50%? 30% ?80% ?

Comparing two models: ROC curves



Comparing two models: ROC curves



Evaluation

1. Error estimation: *Hold-out or Cross-Validation*
2. Evaluation performance measures: *Accuracy or κ-statistic*
3. Statistical significance validation: *MacNemar or Nemenyi test*

Error Estimation

Data available for testing

- Holdout an independent test set
- Apply the current decision model to the test set
- The loss estimated in the holdout is an unbiased estimator

Not enough data available for testing

- Divide dataset in 10 folds
- Repeat 10 times: use one fold for testing and the rest for training

	Predicted Class+	Predicted Class-	Total
Correct Class+	75	8	83
Correct Class-	7	10	17
Total	82	18	100

Simple confusion matrix example

2. Evaluation performance measures

	Predicted Class+	Predicted Class-	Total
Correct Class+	tp	fn	tp+fn
Correct Class-	fp	tn	fp+tn
Total	tp+fp	fn+tn	N

Simple confusion matrix example

- Precision $= \frac{tp}{tp+fp}$
- Recall $= \frac{tp}{tp+fn}$
- Accuracy $= \frac{tp+tn}{total}$
- $F_1 = 2 \cdot \frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}$

2. Evaluation performance measures

	Predicted Class+	Predicted Class-	Total
Correct Class+	75	8	83
Correct Class-	7	10	17
Total	82	18	100

Simple confusion matrix example

- Accuracy $= \frac{75}{100} + \frac{10}{100} = \frac{75}{83}\frac{83}{100} + \frac{10}{17}\frac{17}{100} = 85\%$
  Others:
- Arithmetic mean $= (\frac{75}{83} + \frac{10}{17})/2 = 74.59\%$
- Geometric mean $= \sqrt{\frac{75}{83}\frac{10}{17}} = 72.90\%$

	Predicted Class+	Predicted Class-	Total
Correct Class+	75	8	83
Correct Class-	7	10	17
Total	82	18	100

Simple confusion matrix example

	Predicted Class+	Predicted Class-	Total
Correct Class+	0.689	0.141	83
Correct Class-	0.141	0.028	17
Total	82	18	100

Confusion matrix for chance predictor

Kappa Statistic

- p_0: classifier's prequential accuracy
- p_c: probability that a chance classifier makes a correct prediction.
- κ statistic

$$\kappa = \frac{p_0 - p_c}{1 - p_c}$$

- $\kappa = 1$ if the classifier is always correct
- $\kappa = 0$ if the predictions coincide with the correct ones as often as those of the chance classifier

Matthews correlation coefficient (MCC)

$$\frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}$$

	Predicted Class+	Predicted Class-	Total
Correct Class+	tp	fn	tp+fn
Correct Class-	fp	tn	fp+tn
Total	tp+fp	fn+tn	N

Simple confusion matrix example

AUC Area under the curve

A ROC space is defined by FPR and TPR (recall)

- FPR $= \frac{fp}{fp+tp}$
- TPR $= \frac{tp}{tp+fn}$

Comparing two models: Accuracy

	Classifier A correct	Classifier B wrong
Classifier A correct	Both	A only
Classifier B wrong	B only	Both

Statistical significance validation (2 Classifiers)

	Classifier A Class+	Classifier A Class-	Total
Classifier B Class+	c	a	c+a
Classifier B Class-	b	d	b+d
Total	c+b	a+d	a+b+c+d

$$M = |a - b - 1|^2 / (a + b)$$

The test follows the χ^2 distribution. At 0.99 confidence it rejects the null hypothesis (the performances are equal) if $M > 6.635$.

Statistical significance validation (> 2 Classifiers)

2 classifiers are performing differently if the corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

- k is the number of learners, N is the number of datasets,
- critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$.

Statistical significance validation (> 2 Classifiers)

Two classifiers are performing differently if the corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

- k is the number of learners, N is the number of datasets,
- critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$.

# classifiers	2	3	4	5	6	7
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693

Critical values for the Nemenyi test