Introduction to Data Mining

Small and Big Data

Daniel Rodriguez David F. Barrero

University of Alcala, Spain

AI Session

Outline

What is Data Mining?

Data Mining

- It is process of discovering *structural patterns* in data [**?**].
  - The patterns discovered must be meaningful in that they lead to some advantage
  - *Structural* Patterns mined are represented in terms of a structure that can be examined, reasoned about, and used to inform future decisions (help to explain something about the data)
- The process must be (semi)automatic
- It can be used to classify/predict unknown examples
- We will need more and more data scientist!

Data Mining

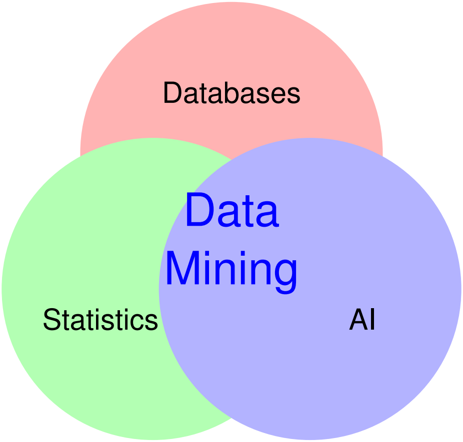
Data...

- Data is increasing without an end. The amount of data stored doubles every 20 months.
    - The Web overwhelms us with information
    - Electronic devices (smartphones), supermarkets, financial habits, health...

... Mining

- Looking for patterns in data.
    - It like extracting large volume of earth & raw material (data) from a mine, process it, obtain a small amount of very precious material (model with valuable use)
    - Analyzing data intelligently can lead to new insights and, in commercial settings, to competitive advantages

What is Data Mining?



Confusing terms

- Data Mining = Statistics + Databases + Artificial Intelligence
- Machine Learning = Field of AI
- Statistics = Field of Mathematics
- Big data = A lot of data
- ML engineer = Professional role
- Data scientist = Professional role
- KDD = A process

Outline

1. Introduction
   - What is Knowledge Discovery in Databases?
   - Knowledge Discovery in Databases
   - Applications
2. Data Mining/KDD Process
   - KDD Process: Integration
   - KDD Process: Selection, cleaning and transformation
   - Machine learning
3. Evaluation
   - Evaluation of Supervised Models
   - Unsupervised Evaluation
4. References

What is Knowledge Discovery in Databases?

The non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data - Fayyad, Piatetsky-Shapiro, Smyth (1996)

non-trivial process	multiple processes
valid	justified patterns/models
novel	previously unknown
useful	can be used
understandable	by human and/or machine

Outline

Applications

It is just a (fantastic) tool that can be applied everywhere!

- If there are data, you can use DM
- If there are no data, you can gather it and use DM

Business information

- Marketing and sales data analysis
- Investment analysis
- Loan approval
- Fraud detection
- etc

Recommender Systems

Recommender systems

What we were interested in:

- High quality *recommendations*

Proxy question:

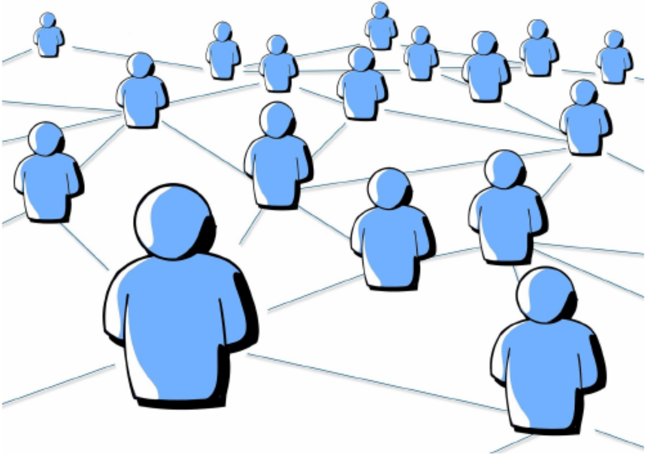
- Accuracy in predicted rating
- Improve by 10% = $1million!

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

NETFLIX

Xavier Amatriain – July 2014 – Recommender Systems

Social Networks

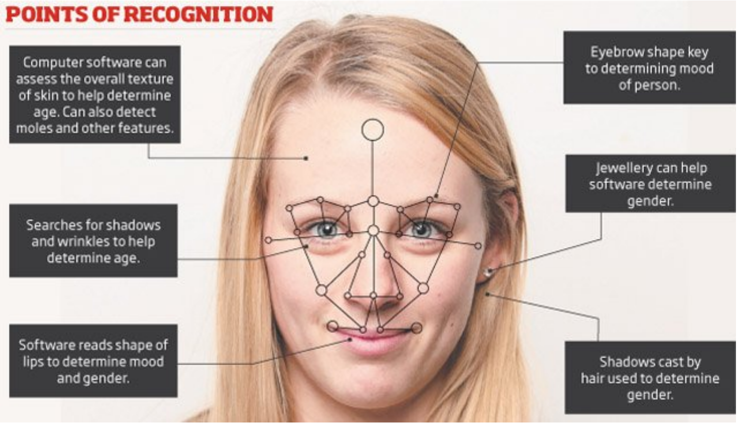


Games

Robotics, Autonomous cars, etc

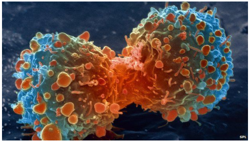
Computer Vision

Computer vision



Scientific information

- Sky survey cataloging
- Biosequence Databases
- Geosciences: Quakefinder
- etc.



Application examples

Robotics

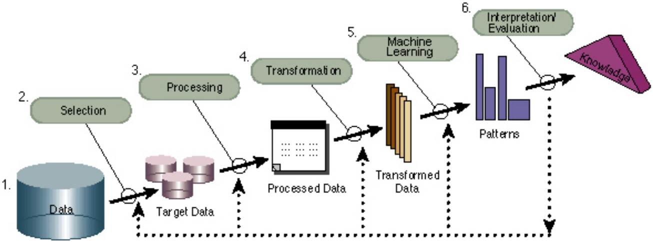
- MarI/O - Machine Learning for Video Games (Video)
- Artificial vision (Video)
- Machine Learning (Video)
- Reinforcement Learning (Video)
- Evolved Electrophysiological Soft Robots (Video)

Deep learning

- DeepBach (Video)
- Deep Neural Network learns Van Gogh's (Video)
- Deep Learning on Drones (Video)
- Emotion recognition (Video)

Data Mining/KDD Process

- Data integration, Selection, cleaning and transformation of data
- Machine Learning (patterns)., classifiers, rules, etc
- Evaluation and interpretation
- Decision making (knowledge)

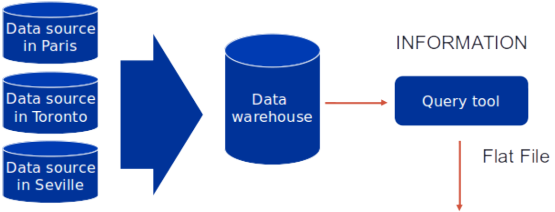


An Overview of the Steps That Compose the KDD Process

Outline

1. Introduction
   - What is Knowledge Discovery in Databases?
   - Knowledge Discovery in Databases
   - Applications
2. Data Mining/KDD Process
   - KDD Process: Integration
   - KDD Process: Selection, cleaning and transformation
   - Machine learning
3. Evaluation
   - Evaluation of Supervised Models
   - Unsupervised Evaluation
4. References

KDD Process: Integration



customer

Cust-ID	name	address	age	income	credit-info	
C1	Smith, Sandy	5463 E Hasting, Burnaby	21	$27000	1	...
C2	Doe, Joe	BC V5A 459, Canada	...	...	...	...

Instances characterized by the values of features (attributes) that measure different aspects of the instance.

Outline

KDD Process: Selection, cleaning and transformation

- Removing outliers
- Data sampling (if we have too much data we can select instances)
- Missing values
- Noisy data: wrongly recorded values
- Feature selection: removing redundant and irrelevant attributes
- Derive new attributes from existing ones, e.g., population density from from inhabitant and area
- Data transformation: discretization, normalization

Outline

KDD Model Classification

- DM algorithms are traditionally divided into:
- Supervised learning which aims to discover knowledge for classification or prediction (*predictive*)
- Unsupervised learning which refers to the induction to extract interesting knowledge from data (*descriptive*).

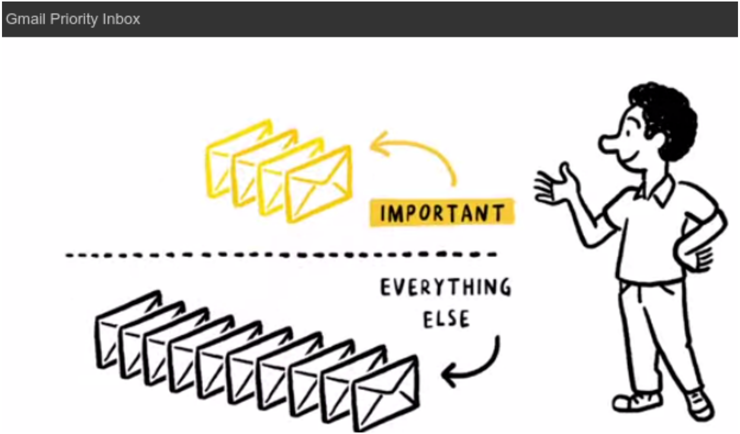
There are also semi-supervised approaches: goal is classification but the input contains both unlabeled and labeled data.
Subgroup Discovery approaches generate descriptive rules are also half way between descriptive and predictive techniques.

Supervised Learning

- A classifier resembles a function in the sense that it attaches a value (or a range or a description) to a set of attribute values. It induces a classification model.
- Given m instances (samples) characterized by n predicted attributes, $A_1, \ldots, A_n$, and the class variable, C

	X_1	$\ldots$	X_n	C_M
$(\mathbf{x}^{(1)}, C^{(1)})$	$x_1^{(1)}$	$\ldots$	$x_n^{(1)}$	$C_M^{(1)}$
$(\mathbf{x}^{(2)}, C^{(2)})$	$x_1^{(2)}$	$\ldots$	$x_n^{(2)}$	$C_M^{(2)}$
$\ldots\ldots$		$\ldots$		$\ldots$
$(\mathbf{x}^{(N)}, C^{(N)})$	$x_1^{(N)}$	$\ldots$	$x_n^{(N)}$	$C_M^{(N)}$
$(\mathbf{x}^{(N+1)}, C^{(N+1)})$	$x_1^{(N+1)}$	$\ldots$	$x_n^{(N+1)}$	??

Supervised Learning



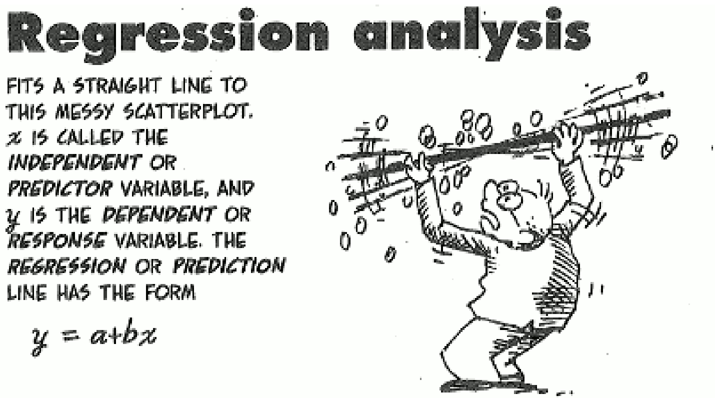
Supervised Learning: Models

- Decision trees Trees where each leaf indicates a class and internal nodes specifies some test to be carried out (e.g. C4.5).
- Rule induction
  - `If condition then class`
  - `If ... then ... else if ...` (hierarchical rules)
- Lazy techniques store previous instances and search similar ones when performing classification with new instances
  - *k-Nearest Neighbour* (*k*-NN) is a method for classifying objects based on closest training example(s) in the feature space.

Supervised Learning: Models (Cont.)

- Regression techniques (numerical prediction)
- Neural Networks are composed by a set of nodes (units, neurons, processing elements) where each node has input and output and performs a simple computation by its node function.
- Statistical techniques. For example:
  - Bayesian networks classifiers assign a set of attributes $A_1, A_2, ..., A_n$ to a class C_j such that $P(C_j|A_1, A_2, \ldots, A_n)$ is maximal
- Meta-techniques combine *multiple classifier models* (there are several ways to do so)

Supervised Learning: Numeric Prediction



Unsupervised Learning

There is no class attribute

- Clustering
  - Tree clustering: join together objects (e.g., animals) into successively larger clusters, using some measure of similarity or distance.
  - Algorithms: K-Means, EM (Expectation Maximization)
- Association rules, e.g., rules among supermarket items
  - Algorithms: APRIORI, etc.

	X_1	$\dots$	X_n
$(\mathbf{x}^{(1)}, C^{(1)})$	$x_1^{(1)}$	$\dots$	$x_n^{(1)}$
$(\mathbf{x}^{(2)}, C^{(2)})$	$x_1^{(2)}$	$\dots$	$x_n^{(2)}$
$\dots\dots$		$\dots$	
$(\mathbf{x}^{(N)}, C^{(N)})$	$x_1^{(N)}$	$\dots$	$x_n^{(N)}$

Unupervised Learning: Clustering

Unupervised Learning: Association Rules

Outline

Evaluation of Supervised Models

- Once we obtain the supervised model with the training data, we need to evaluate it with some new data (testing data)
  - We cannot use the the same data for training and testing. E.g. evaluating a student with exercises previouly solved. Student s marks will be optimistic and we dont know about student capability to generalise learned concepts.

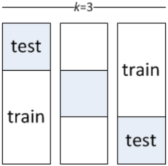
Holdout approach

- Holdout approach consists of dividing the dataset into training (approx. 2/3 of the data) and testing (approx 1/3 of the data).
- Problems: Skewed data, missing classes, etc. if randomly divided
- Stratification ensures that each class is represented with approximately equal proportions, e.g., if data contains aprox 45% of positive cases, the training and testing datasets should mantain similar proportion of positive cases.
- Holdout estimate can be made more reliable by repeating the process with different subsamples (repeated holdout method)
- The error rates on the different iterations are averagerate)

Cross Validation

Cross-validation (CV) avoids overlapping test sets.
The k-fold CV consists on:

- First step: split dataset ($\mathcal{D}$) into k subsets of equal size $C_1, \ldots, C_k$. Subsets are generally stratified before the CV is performed
- Second step: we construct a dataset $D_i = D - C_i$ used for *training* and test the accuracy of the classifier $f(D_i)$ on C_i subset for *testing*



The error estimates are averaged to yield an overall error estimate, i.e., having done this for all k, usually $k = 10$, we estimate the accuracy of the method by averaging the accuracy over the k cross-validation.

Confusion matrix

Confusion matrix

		Actual		
		Pos	*Neg*	
Pred	*Pos*	True Pos (TP)	False Pos (FP) (False alarm)	$PPV =$ $Conf =$ $Prec = \frac{TP}{TP+FP}$
	Neg	False Neg (FN)	True Neg (TN)	$NPV = \frac{TN}{FN+TN}$
		$Recall =$ $Sens =$ $TP_r = \frac{TP}{TP+FN}$	$Spec =$ $TN_r = \frac{TN}{FP+TN}$	

Evaluation Metrics

- Number of correct classifications:

$$\sum_{i=1}^{N} \delta(c^{(i)}, c_M^{(i)})$$

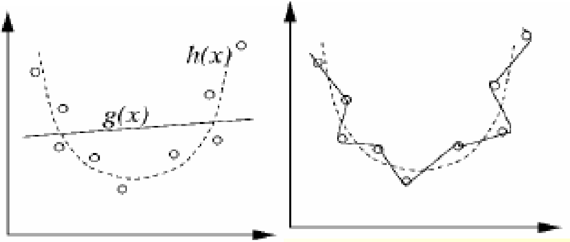
  where $\delta(c^{(i)}, c_M^{(i)}) = \{1 if c^{(i)}, c_M^{(i)}, 0 otherwise$
- For probabilistic classifiers Brier score (1950) $bs(\mathcal{D}) = \frac{1}{N} \sum \sum \ldots$
- Many times we need to combine the TP and FP to estimate the goodness of a classifier. For example, with imbalanced data, the accuracy of a classifer needs to improve the percentage of the mayority class. In a binay problem and 50/50 distribution, we need improve accuracy over 50%. However if the distribution is 90/10, accuracy needs to be over 90%

Graphical Evaluation

- AUC (Area under the ROC)
- Precision Recall curve

Evaluation: Underfitting vs. Overfitting

Evaluation: Underfitting vs. Overfitting

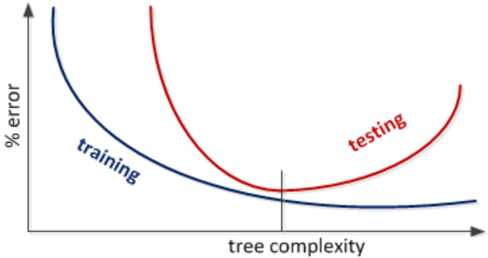


Too simple vs. Too complex

Evaluation: Underfitting vs. Overfitting (cont.)

Increasing the tree size, decreases the training and testing errors. However, at some point after (tree complexity), training error keeps decreasing but testing error increases

Many algorithms have parameters to determine the model complexity (e.g., in decision trees is the prunning parameter)



Outline

Unsupervised Evaluation

- Support: the proportion of times that the rule applies
- Confidence: the proportion of times that the rule is correct

```
if  colour = light and  #nuclei = 1
Then #tails = 1
                (suport = 25%;
                 confidence = 50%)
}
```

References