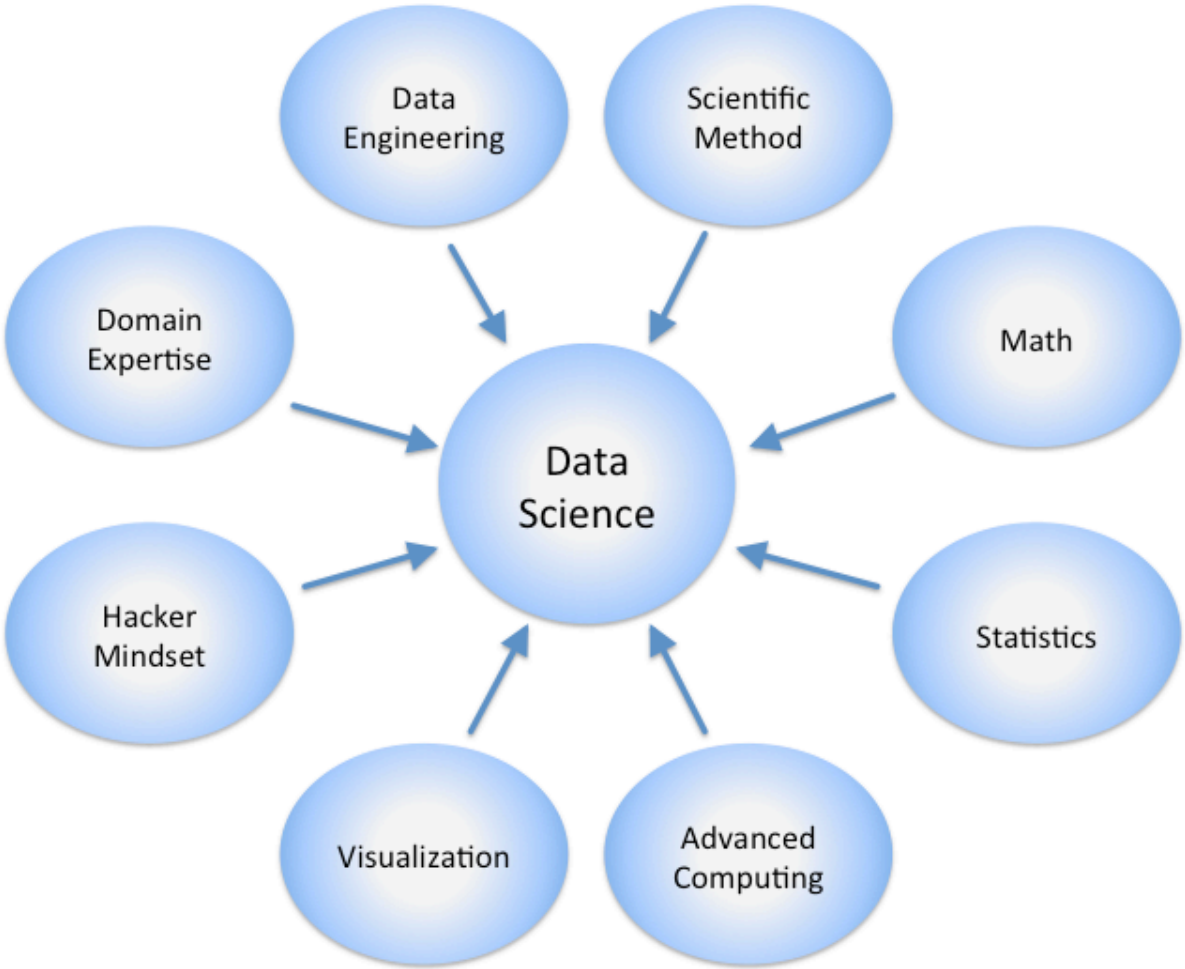
Introduction to Machine Learning

Judit Ács judit@aut.bme.hu

March 14, 2018

BME AUT
Department of
Automation and
Applied Informatics



source: wikipedia.org

Data mining

The nontrivial extraction of implicit, previously unknown, and potentially useful information from data.

- non-trivial
- relationship between data points
- (large) dataset
- make predictions on unknown examples

Examples and counterexamples

Convenience store statistics

- Number of customers
  - trivial information
- Last month's income
  - trivial information
- Items most frequently bought together
  - *finding frequent itemsets*
- How many cashiers need to be open at Friday 16 pm?
  - customer queuing model
  - *time series modeling*

Multidisciplinary field

mathematics linear algebra, matrix algebra, optimization, statistics, analysis

software engineering data collection, data cleaning, employing machine learning algorightms

other bioinformatics, computational linguistics, computational social sciences

Data representation 1.

- Vector space models
  - one vector corresponds to one observation or sample
  - the number of features of the observation space is the dimension of the vectors
  - e.g. spam detection - one vector represents one email
    - length of the mail
    - sender
    - Does it contain the word *Rolex*?
    - Does it contain the expression *Trust fund*?

Data representation 2.

- Time series
  - there exists a natural ordering of the samples
  - doesn't have to be 'time'
  - e.g. daily precipitation in Budapest
  - words in a document
- Graph
  - data points have explicit relationship with each other such as casual relations
  - e.g. modeling medical problems, diseases (What causes cancer?)
  - geographic relationship: traffic in Budapest

Choosing the right representation is crucial.

Vector representation

sample one data point - vector

feature a property or attribute of a sample - one element of a vector

dataset collections of all samples - matrix

label correct *answers* for all samples in a dataset - vector

training set part of the dataset used for training - matrix

validation or development dataset part of the dataset used for cross-validation, early stopping and hyperparameter tuning - matrix

test set part of the dataset used for testing trained models

Dataset categorization

1. labeled vs. unlabeled
   - labeled
     - the answer is known
     - e.g. movie ratings
     - typically expensive to create
     - we could always use more of it
     - *supervised learning*
   - unlabeled
     - the answer is unknown
     - cheaper, more plentiful
     - *unsupervised learning*
2. continuous vs. discrete
3. categorical vs. quantitative / numerical

The data mining process

1. Data collection
2. Data cleaning
   - noise and outlier filtering
   - handling missing data
3. Data transformation / preprocessing
   - dimensionality reduction (less used nowadays)
   - normalization, standardization
4. Training the model
5. Evaluating the model

Data mining problems 1.

- Classification
  - assign a label for each sample
  - labels are predefined and usually not very numerous
  - e.g. is an email a spam or a ham?
- Regression
  - predict a continuous variable
  - e.g. predict real estate prices, stock market based on history, location, amenities

Data mining problems 1.

- Classification – supervised, discrete classes
  - assign a label for each sample
  - labels are predefined and usually not very numerous
  - e.g. is an email a spam or a ham?
- Regression
  - predict a continuous variable
  - e.g. predict real estate prices, stock market based on history, location, amenities

Data mining problems 1.

- Classification – supervised, discrete classes
  - assign a label for each sample
  - labels are predefined and usually not very numerous
  - e.g. is an email a spam or a ham?
- Regression – supervised, continuous target
  - predict a continuous variable
  - e.g. predict real estate prices, stock market based on history, location, amenities

Data mining problems 2.

- Clustering
  - group samples into clusters according to a similarity measure
  - goal: high intra-group similarity (samples in the same cluster should be similar to each other), low inter-group similarity (samples in different clusters shouldn't be similar)
  - e.g. market segmentation

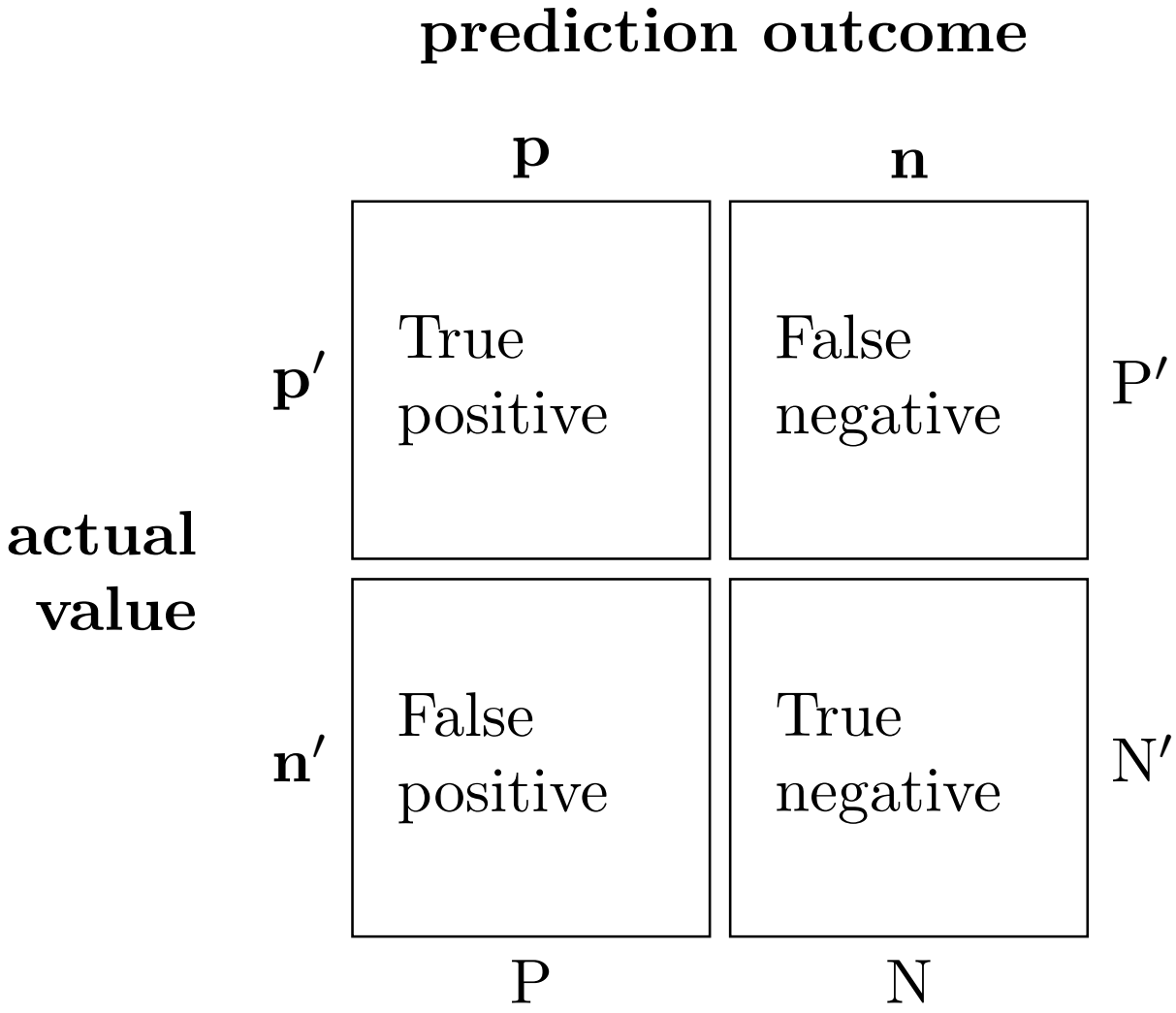
Data mining problems 2.

- Clustering – unsupervised, discrete clusters
  - group samples into clusters according to a similarity measure
  - goal: high intra-group similarity (samples in the same cluster should be similar to each other), low inter-group similarity (samples in different clusters shouldn't be similar)
  - e.g. market segmentation

Data mining problems 3.

- Time series analysis and prediction
  - pattern discovery and prediction in time series
- Frequent itemset mining
  - e.g. What products do people buy at the same time?
- Recommendation systems
  - e.g. movies similar to the ones the user already likes

Evaluation - Binary classification



Precision, recall and F-score

Precision: fraction of positive samples among those labeled positive

$$\text{Precision} = \frac{tp}{tp + fp}$$

Recall: fraction of recovered positive samples of all positive samples

$$\text{Recall} = \frac{tp}{tp + fn}$$

Precision, recall and F-score

Precision: fraction of positive samples among those labeled positive

$$\text{Precision} = \frac{tp}{tp + fp}$$

Recall: fraction of recovered positive samples of all positive samples

$$\text{Recall} = \frac{tp}{tp + fn}$$

F-score: harmonic mean of precision and recall

$$\text{F-score} = 2 * \frac{\text{precrec}}{\text{prec} + \text{rec}}$$

Evaluation - multiclass classification

- one-versus-rest precision, recall and F-score
  - samples from class *i* are the positive, everything else are the negative examples
  - *k* scores for *k* classes
- average or weighted average of all *k* scores

Evaluation - regression

Root-mean-square error

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}},$$

where $\hat{y}_t$ are the predicted values, y_t is the true value and n is the number of samples.

Evaluation - clustering

- high intra-cluster similarity, low inter-cluster similarity
- direct evaluation on the application of interest
- against gold standard labeled set

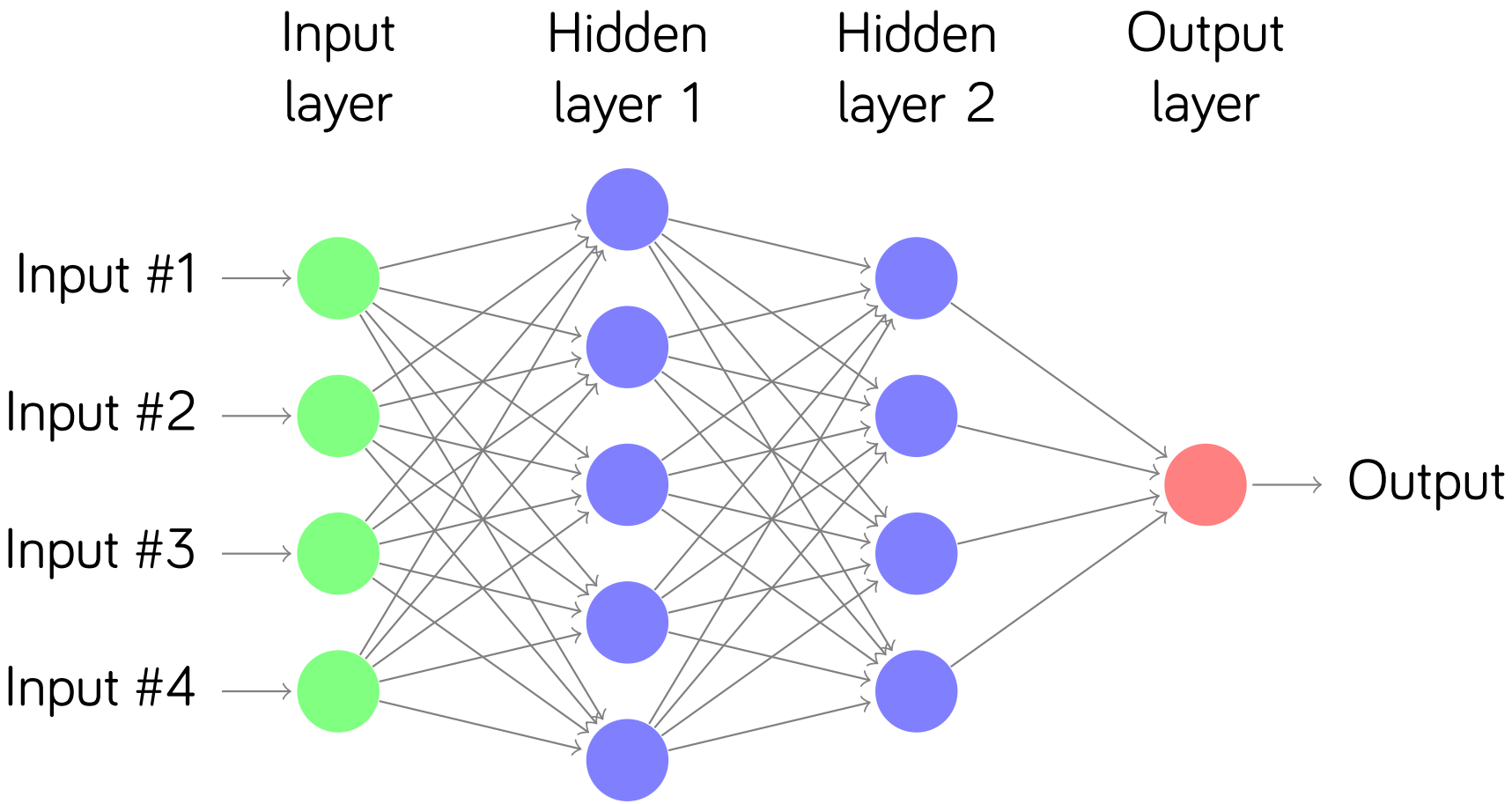
Technology

- Python, R, Java, Lua
- Linux, Windows less so
- 'traditional' machine learning: scikit-learn (Python), Weka (Java)
- deep learning: TensorFlow, PyTorch, Keras etc.
- plain text, CSV, TSV, XML (less popular)

Deep learning

- representation learning instead of task-specific manual features
- biological inspiration (neurons, activation)

Feed forward neural network



Feed forward neural network

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{x})$$

$$\mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1)$$

$$\mathbf{y} = \sigma(\mathbf{W}_3\mathbf{h}_2)$$

σ: activation function, typically non-linear such as the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Feed forward neural network

- fixed input ($\mathbf{x}$) and output (y)
- weights are learned through backpropagation
- capacity or memory of the network

Drawbacks:

- data flows in one direction
- temporal and spacial relationships are not explicitly modeled

Recurrent neural network

- the network has directed cycles
- temporal relationships are easier to learn
- Long-short term memory (LSTM)
  - LSTM cells have *memory*, can retain, update or forget previous information
  - a network typically uses hundreds of these cells
- Gated recurrent unit (GRU)
  - memory cell similar to LSTM

Convolutional neural network

- spacial structures are explicitly modeled
- very successful in image processing
- can be applied to one dimensional data (1D convolution) such as text or audio

Other architectures

- Generative adversarial network (GAN)
  - two networks compete against each other: a generator and a discriminator
  - generator: tries to create fake samples similar to real samples
  - discriminator: tried to distinguish real samples from fake samples
  - hard to train
  - extremely popular, hundreds of variants

Other architectures

- (Variational) autoencoder (VAE)
  - the input and the output of the network are the same
  - the network learns to compress the input, then recover the original image from the compressed representation
  - not very good at compression, but learns useful representation
  - variational: generate real-like samples from noise

Thank you for your attention

Demo

https://github.com/bi-labor/pandas_jupyter/tree/master/notebooks/bi_ea_demo