Machine Learning and Common Sense

Dr. Lothar Richter

Lothar Richter
Structure of the Talk

- general ideas of machine learning
- some vocabulary and basics
- machine learning and bioinformatics
- hidden assumptions

Lothar Richter
Ideas

- a machine learning device can generalize from real world observations into a “formal” model
- each model reflects only a few aspect of reality
- no model can completely represent the reality, i.e. a photograph of a dog remains a photograph and not a real dog
- the model should reflect a concept or commonalities and not individual characteristics

Lothar Richter
Inductive Bias

- every learning scheme discards some aspects of reality to construct a model
- this may differ between different learning schemes
- this might also already happens on the level of feature extraction, i.e. choosing the types and values to represent an observation
- this is not to be mistaken with the predictive bias

Lothar Richter
Some vocabulary

- learning scheme: a specific learning algorithm producing a model like decision trees, rule based systems, SVMs, Bayesian networks, etc.
- attribute/feature: a variable describing a specific aspect of real world observations, like body weight, color, certain property found yes/not
- instance: a single observation describing an observed event by assigning values for each feature used to represent this observation
Some vocabulary II

- training: phase of analyzing real world observations in a formalized representation to derive parameters and/or internal structure
- test phase: phase of model application to determine the reliability of statements (predictions) on instances not used for training
- label: an attribute selected to be predicted
Types of Learning

- depending on the presence of a label we distinguish between supervised and unsupervised learning
- unsupervised learning: concept learning, frequent item sets, clustering
- supervised learning: everything with labeled data which allows to make a prediction
Synonyms

- Data Mining and Machine Learning are widely used as synonyms
- if there is a difference at all, both emphasize different aspects
- also commonly used: statistical learning, KDD (knowledge discovery in databases)
- sometimes you also distinguish descriptive and inductive data mining or descriptive and predictive modelling

Lothar Richter
Categorial Types

- **nominal**: set of distinct values an attribute can be assigned like red, blue, man, woman, a.s.f.
- **ordinal**: like a nominal type plus an order relationship for the values, like newborn, infant, pupil, adult, old man
- the boolean type is a two values nominal type with true and false
- arithmetic operations are not defined

Lothar Richter
Continuous Types

- integer: very similar to the ordinal type but arithmetic makes sense, like number of children
- interval-scaled: contains numerical values measured in equal intervals from an arbitrary set origin, e.g. temperature in °C; it is ordered, arithmetic makes sense
- ratio-scaled: similar to interval-scaled with the extension that a value of 0 indicates the absence of this property

Lothar Richter
Data Preprocessing

Before you can start to build a model in most cases the data are subjected to various preprocessing steps:

- Feature Extraction
- Discretization
- Feature Selection
Feature Extraction/Construction

- Conversion of observation records into a formalized, computer-readable representation
- Definition of an attribute type
- Assignment of appropriate attribute values
- This implies a strong involvement of the analyst
- Important: common sense, background knowledge from expert domains

Lothar Richter
Discretization

- conversion of an numeric attribute into an categorial one
- the values range is split into a set of discrete intervals and the interval label is used as value
- this can help avoid overfitting
- unsupervised: domain dependent, equal width, equal frequency
- supervised: construct bins in respect to class label

Lothar Richter
Feature Selection

- remove values from instances, i.e. discard some features of a data set because these are:
  - irrelevant
  - redundant
  - noisy/faulty
- possible benefits:
  - improve efficiency and accuracy
  - prevent overfitting
  - save space

Lothar Richter
Feature Selection Strategies

- unsupervised: based on domain knowledge, random sampling

- supervised:
  - measures consider the class (filtering): Gini-index, information gain, relief, ...
  - use a learning scheme’s performance (wrapping):
    - select the set of attribute which leads to best performance
    - forward selection: increase the set of attribute 1 by 1
    - backward elemination: decrease the set of attributes 1 by 1
Machine Learning and Bioinformatics

- today biology has to span between two extremes:
  - statements on the nucleotide level (one level below genes)
  - statements the individual/population level on the other hand

- the gain in speed to generate sequence data (nucleotide sequences) has clearly outpaced the speed of analysis and knowledge discovery

- current lab technology even cannot fill the gap between sequence and structure

Lothar Richter
Primary Data Growth Nucleotides

National Library of Medicine
Twenty Three Years of Growth:
NCBI Data and User Services

GenBank Base Pairs
Users (Average)

GenBank at NCBI
Entrez
BLAST
Genomes
OMIM
PubMed
Genome Reference Consortium
Genome-Wide Association Studies
NIH Public Access
PubChem
PubMed Central
Human Genome
Genetic Testing Registry
ClinVar
1000 Genomes

Years:
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012

Base Pairs (Billions):
0
20
40
60
80
100
120
140

Users/Weekday (Millions):
0.0
0.5
1.0
1.5
2.0
2.5

taken from http://www.nlm.nih.gov/about/image/2014CJ_fig_5.png

Lothar Richter
Primary Data Growth Nucleotides

取自http://www.ebi.ac.uk/ena/about/statistics

Lothar Richter
Automatically Derived Proteins

taken from http://www.ebi.ac.uk/uniprot/TrEMBLstats

Lothar Richter
Proteins with Some Evidence

Number of entries in UniProtKB/Swiss-Prot


Lothar Richter
All Known Structures

Yearly Growth of Total Structures in PDB

Lothar Richter
Role of DM / ML

Data Mining helps to:

- structure the data and compress the data
- filter out mistakes and outliers because of experimental errors and other noise
- reduce redundancy
- replace wet lab analyses with predictions
- detect interesting relationship and models and directs man power towards points where it is needed

Lothar Richter
Overview of the Steps in KDD

Here, data are a set of facts (for example, cases in a database), and pattern is an expression in some language describing a subset of the data or a model applicable to the subset. Hence, in our usage here, extracting a pattern also designates fitting a model to data; finding structure from data; or, in general, making any high-level description of a set of data. The term process implies that KDD comprises many steps, which involve data preparation, search for patterns, knowledge evaluation, and refinement, all repeated in multiple iterations. By nontrivial, we mean that some search or inference is involved; that is, it is not a straightforward computation of predefined quantities like computing the average value of a set of numbers. The discovered patterns should be valid on new data with some degree of certainty. We also want patterns to be novel (at least to the system and preferably to the user) and potentially useful, that is, lead to some benefit to the user or task. Finally, the patterns should be understandable, if not immediately then after some postprocessing. The previous discussion implies that we can define quantitative measures for evaluating extracted patterns. In many cases, it is possible to define measures of certainty (for example, estimated prediction accuracy on new data) or utility (for example, gain, perhaps in dollars saved because of better predictions or speedup in response time of a system). Notions such as novelty and understandability are much more subjective. In certain contexts, understandability can be estimated by simplicity (for example, the number of bits to describe a pattern). An important notion, called interestingness (for example, see Silberschatz and Tuzhilin [1995] and Piatetsky-Shapiro and Matheus [1994]), is usually taken as an overall measure of pattern value, combining validity, novelty, usefulness, and simplicity. Interestingness functions can be defined explicitly or can be manifested implicitly through an ordering placed by the KDD system on the discovered patterns or models.

Given these notions, we can consider a pattern to be knowledge if it exceeds some interestingness threshold, which is by no means an attempt to define knowledge in the philosophical or even the popular view. As a matter of fact, knowledge in this definition is purely user oriented and domain specific and is determined by whatever functions and thresholds the user chooses.

Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data. Note that the space of articles taken from U. Fayyad, G. Piatetsky-Shapiro, P. Smyth “From Data Mining to Knowledge Discovery in Databases” (1996) AI Magazine, 17, 37-54
ML Tools employed in Bioinformatics

taken from “The rise and fall of supervised machine learning techniques”

Lothar Richter
Possible Explanations the Prevalence of ANNs and SVMs

- they are capable to handle a huge number of attributes
- they are quite robust against uninformative features
- they implicitly adjust feature weights during the training phase
- they work sufficiently well

Lothar Richter
How is Machine Learning Influenced by Underlying Assumptions?

- there are a number of assumptions in the various processing steps
- the performance depends on that these assumptions hold
- very often we cannot really check or proof if this is true

Lothar Richter
Background distribution

- we assume that the background distribution is uniform
- i.e. the underlying source emits instances with constant probabilities
- possible solutions:
  - use many features to represent complex scenarios
  - use stream mining algorithms which update parameters
Unfair Sampling

- due to “experimental” reason the sample represents only a special subset of the entities
- especially difficult for lazy learning methods like k-nearest-neighbors
- possible solutions:
  - remove redundancy
  - use stratification
  - check variance and identify difficult instances
Suitable Representation

- we assume that the selected feature set can represent the concept to learn
- since we often do not know causal relationships we might mistake employed features with real causing ones
  - e.g. number of storks and births in Germany
  - e.g. opening umbrellas and rain
- remedy: use background knowledge, careful interpretation
Representation of the Concept

<table>
<thead>
<tr>
<th>Weather</th>
<th>Temperature</th>
<th>Wind</th>
<th>playTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>rainy</td>
<td>high</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>dry</td>
<td>medium</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>medium</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>dry</td>
<td>medium</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

- with this attributes it is really hard to learn favorable conditions for playing tennis
- we unconsciously assume that these attributes are sufficient to describe the scenario

Lothar Richter
Representation of the Concept

<table>
<thead>
<tr>
<th>Weather</th>
<th>Temperature</th>
<th>Wind</th>
<th>Buddy</th>
<th>playTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>rainy</td>
<td>high</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>dry</td>
<td>medium</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>medium</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>dry</td>
<td>medium</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

- actually there no proof to show whether important attributes are missing or not
- for this aspect background knowledge and experience are most important

Lothar Richter
Performance Estimation

- training a model means discarding and keeping some information from the observed instances
- depending on the learning scheme instance specific information is stored too
- instance specific information leads to overfitting
- overfitting: a prediction model is biased towards the training examples, i.e.:
  - better performance on training examples
  - worse performance on new instances

Lothar Richter
More Realistic Estimations

- most optimistic estimation: Resubstitution error (determine performance on the set completely used for training)
- if you have a lot of data: determine the error on an independent test set
More Realistic Estimations II

- LOOCV: Leave one out cross validation
  - always one example is hold out for testing, the remaining for training
  - n iterations with n instances, final result is the average
  - still quite biased
  - use to check the influence of individual instances
  - if you have a small number of instances

Lothar Richter
More Realistic Estimations II

- n-fold cross validation, typically n=10
  - partition the data in n partition
  - use n-1 partitions for training
  - use 1 partition for performance assessment
  - repeat with a different hold-out partition
  - average performance

- every step where class information is considered has to be included in the loop! (done after partitioning)

Lothar Richter
Text Books

- Data Mining 3rd edition, Ian Witten, Eibe Frank, Mark Hall, 2011, ISBN 978-0-12-374856-0, Morgan Kaufmann

Lothar Richter