Text Classification

Ulf Leser, Jurica Seva

Text Classification

- Text: Any sequence of tokens
  - Typical: Books, scientific articles, news, emails, letters, ...
  - Assumption: Grammatically correct sentences; use of standard words; monolingual; no special characters
  - Atypical: Tweets, reports with images and tables, spoken lang., ...
- Classification: Assign each text to one of a set of classes
  - Set of classes is given
  - Single-label: Every text gets exactly one class
    - Variations: Multi-label classification, “unknown” class, ...
  - Classes have no internal structure
    - Variation: Hierarchical classification
  - Binary classification: There are only two classes (relevant or not)
    - Variation: Multi-class; typically 1-10

Applications of Text Classification

- Language identification
- Topic identification
- Spam detection
- Content-based message routing
- Author identification (which plays were really written by Shakespeare?)
- Named entity recognition (is this token part of a NE?)
- ...

Text Classification

- Given a set D of docs and a set of classes C. A classifier is a function f: D$\rightarrow$C
- How does this work in general (supervised learning)?
  - Function v mapping a doc into vector of features (feature space)
    - For instance, its bag-of-words, possibly weighted by TF*IDF
  - Obtain a set S of docs with their classes (training data)
  - Find the characteristics of the docs in each class (model)
    - Which feature values / ranges are characteristic?
    - What combinations or properties are characteristic?
  - Encode the model in a classifier function f operating on the feature vector: v: D$\rightarrow$V, and f: V$\rightarrow$C
  - Classification: Compute f(v(d))

Good Classifiers

- Problem: Finding a good classifier
  - Assigning as many docs as possible to their correct class
  - Involves finding a proper feature space
- How do we know?
  - Use a (separate) gold standard data set
  - Use training data twice (beware of overfitting)
    - Learning the model
    - Evaluating the model
  - f is the better, the more docs it assigns to their correct classes

Overfitting

- Let S be a set of instances with their classes (training data)
- We can easily build a perfect classifier for S
  - f(d) = {f(d'), if $\exists$d'$\in$S with d'=d; random otherwise)
  - f is perfect for any doc from S
- But: Produces random results for any new document
- Improvement
  - f(d) = {f(d'), if $\exists$d'$\in$S with d'~d; random otherwise)
  - Improvement depends on |S| and definition of "~"
  - See kNN classifiers
- Overfitting
  - If the model strongly depends on S, f overfits – it will only work well if all future docs are very similar to the docs in S
  - You cannot find overfitting when evaluation is performed on S only

Against Overfitting

- f must generalize: Capture features that are typical for all docs in D, not only for the docs in S
- Still, often we only have S for evaluation ...
  - We need to extrapolate the quality of f to unknown docs
- Usual method: Cross-validation (leave-one-out, jack-knife)
  - Divide S into k disjoint partitions (typical: k=10)
    - Leave-one-out: k=|S|
  - Learn model on k-1 partitions and evaluate on the k'th
  - Perform k times, each time evaluating on another partition
  - Estimated quality on new docs = average performance over k runs

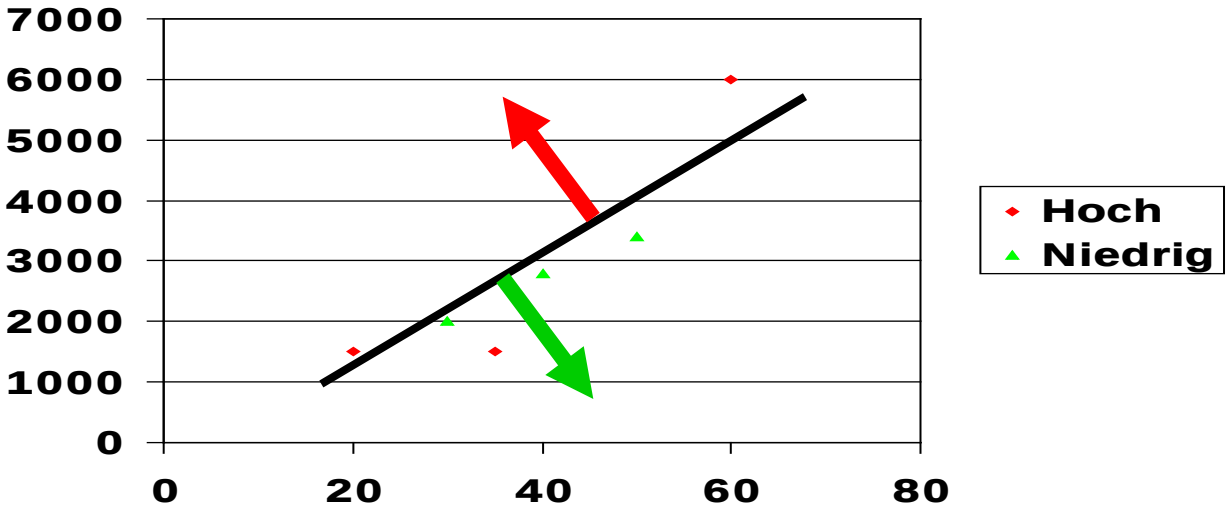
A Simple Example

- Aggregated history of credit loss in a bank

ID	Age	Income	Risk
1	20	1500	High
2	30	2000	Low
3	35	1500	High
4	40	2800	Low
5	50	3000	Low
6	60	6000	High

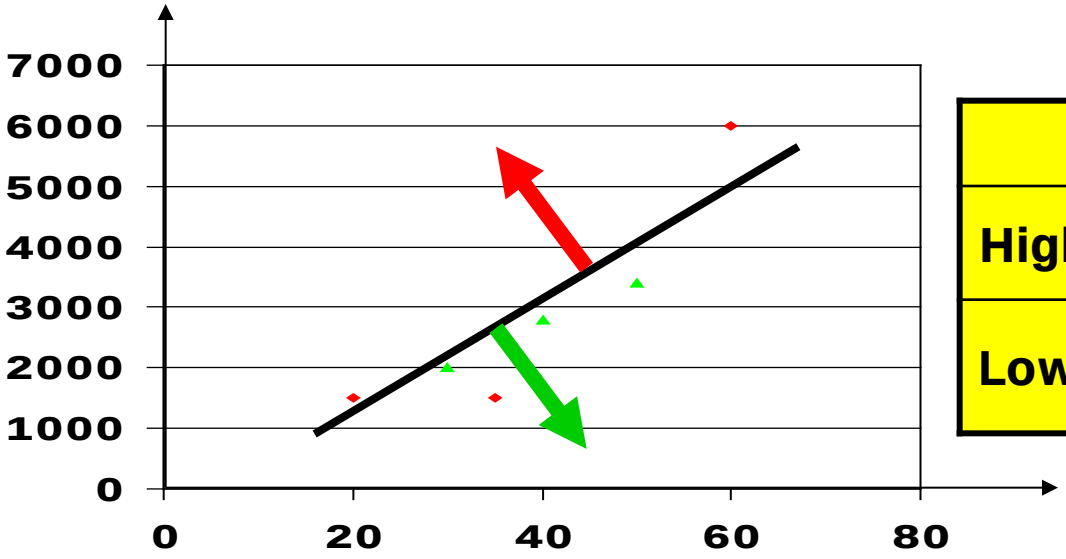
- Now we see a new person, 45 years old, 4000 Euro income
- What is his/her risk?

Regression



- Simple approach: Linear separation by line achieving the minimum squared error (regression)
- Use location relative to regression line as classifier
  - Bad method – regression does not take classes into account
  - But there are classifier based on regression

Performance on the Training Data



	High	Low
High	2	0
Low	1	3

- Quality of predicting “high risk”
  - Precision = 2/2, Recall = 2/3, Accuracy = 5/6
- Assumptions: Linear problems, numerical attributes

Categorical Attributes

ID	Age	Type of car	Risk of Accident
1	23	Family	High
2	17	Sports	High
3	43	Sports	High
4	68	Family	Low
5	25	Truck	Low

- Assume this is analyzed by an insurance agent
- What will he/she infer?
  - Probably a set of rules, such as

```
if      age > 50      then risk = low
elseif  age < 25      then risk = high
elseif  car = sports  then risk = high
else    risk = low
```

Decision Rules

ID	Age	Type of car	Risk of Accident
1	23	Family	High
2	17	Sports	High
3	43	Sports	High
4	68	Family	Low
5	25	Truck	Low

- Can we find less rules which, for this data set, result in the same classification quality?

```
if      age > 50      then risk = low
elseif  car = truck   then risk = low
else    risk = high
```

A Third Approach

ID	Age	Type of car	Risk of Accident
1	23	Family	High
2	17	Sports	High
3	43	Sports	High
4	68	Family	Low
5	25	Truck	Low

- Why not:

```
If      age=23 and car = family then risk = high
elseif  age=17 and car = sports then risk = high
elseif  age=43 and car = sports then risk = high
elseif  age=68 and car = family then risk = low
elseif  age=25 and car = truck  then risk = low
else    flip a coin
```

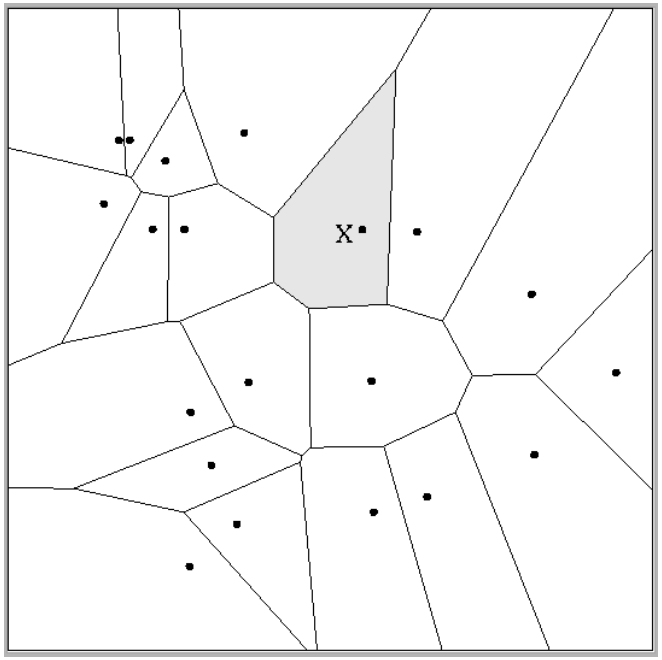
Feature Selection

- Good idea: Use only subset of all features
  - Faster, reduction of noise
- Simple method: Use those t where p(t|c) show the biggest differences between the different classes
  - Needs to assess differences; e.g., entropy, information gain, ...
- Numerous methods for feature selection
  - Finding the best features is not the same as finding the best subset of features
  - Overfitting is an issue: "Best features for S" ≠ "best features for D"
- Some methods benefit more than others
  - MaxEnt and SVM usually not much, Bayes usually a lot (think of redundant features)

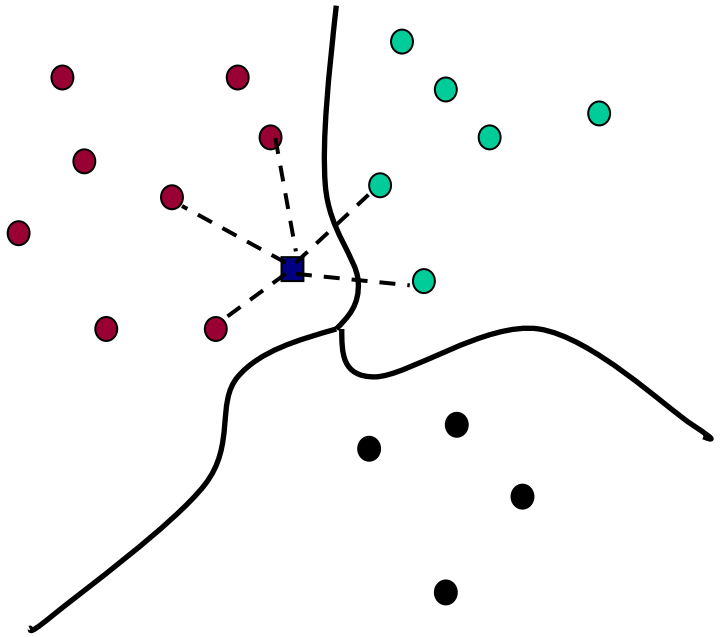
Classification Methods

- There are many different classification methods
  - k-nearest neighbor
  - Naïve Bayes, Bayesian Networks, Graphical models
  - Decision Trees and Random Forests
  - Maximum Entropy
  - Support Vector Machines
  - Perceptrons, Neural Networks
  - ...
- Effectiveness of classification depends on problem, algorithm, feature selection method, sample, evaluation, ...
- Differences when using different methods on the same data/representation are often astonishing small

Example: Nearest Neighbor Classifiers



Voronoi diagram in 2D-space
(for 1NN)



5NN

Bayes‘ Classification

- Uses frequencies of feature values in the different classes
- Given
  - Set S of docs and set of classes C={c_1, c_2, ... c_m}
  - Docs are described as a set F of discrete features
    - Usually the presence/absence of terms in d
- We seek p(c_i|d), the probability of a doc d∈S being a member of class c_i
- d eventually is assigned to c_i with argmax p(c_i|d)
- Replace d with feature representation

$$p(c \mid d) = p(c \mid v(d)) = p(c \mid f_1[d],..., f_n[d]) = p(c \mid t_1,...,t_n)$$

Probabilities

- What we learn from the training data (MLE)
  - The a-priori probability p(t) of every term t
    - How many docs from S have t?
  - The a-priori probability p(c) of every class $c \in C$
    - How many docs in S are of class c?
  - The conditional probabilities p(t|c) for term t being true in class c
    - Proportion of docs in c with term t among all docs in c
- Rephrase and use Bayes‘ theorem

$$p(c \mid t_1,...,t_n) = \frac{p(t_1,...,t_n \mid c) * p(c)}{p(t_1,...,t_n)} \approx p(t_1,...,t_n \mid c) * p(c)$$

Naïve Bayes

- We have $p(c \mid d) \approx p(t_1,...,t_n \mid c) * p(c)$
- The first term cannot be learned accurately with any reasonably large training set
  - There are 2^n combinations of (binary) feature values
- „Naïve“ solution: Assume statistical independence of terms
- Then

$$p(t_1,...,t_n \mid c) = p(t_1 \mid c) * ... * p(t_n \mid c)$$

- Finally

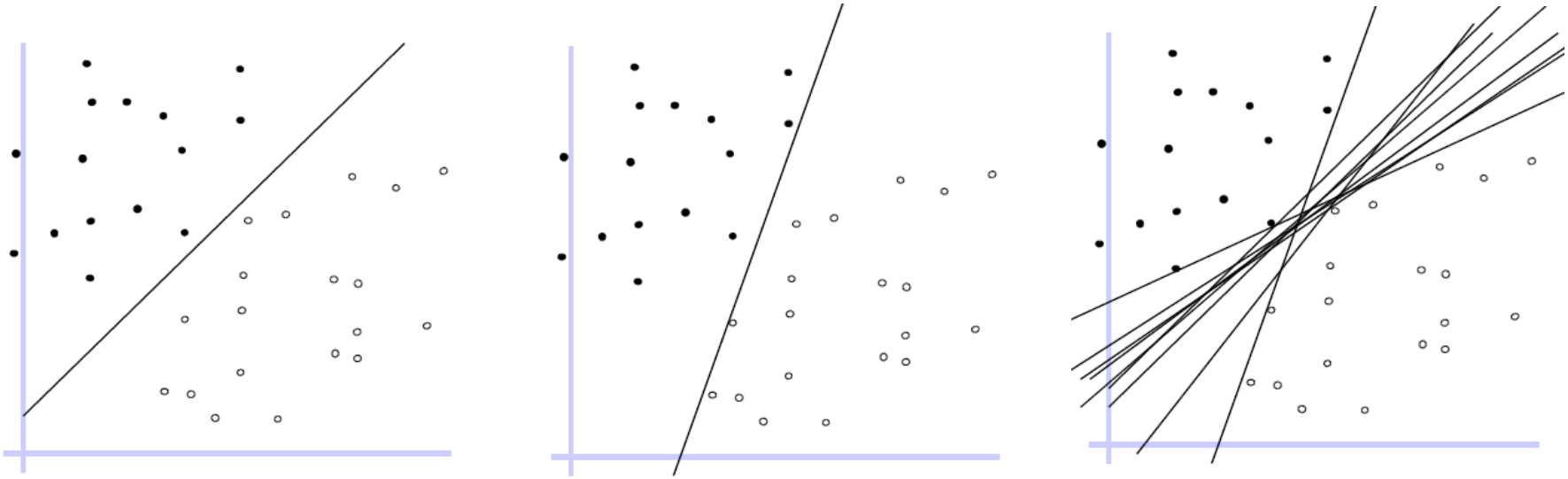
$$p(c \mid d) \approx p(c) * \prod_{i=1}^{n} p(t_i \mid c)$$

Text = High Dimensional Data

- High dimensionality: 100k+ features
- Sparsity: Feature values are almost all zero
- Most document pairs are very far apart (i.e., not strictly orthogonal, but only share very common words)
- Consequence: Most document sets are well separable
  - This is part of why linear classifiers are quite successful in this domain
- The trick is more of finding the "right" separating hyperplane instead of just finding (any) one

Linear Classifiers (2D)

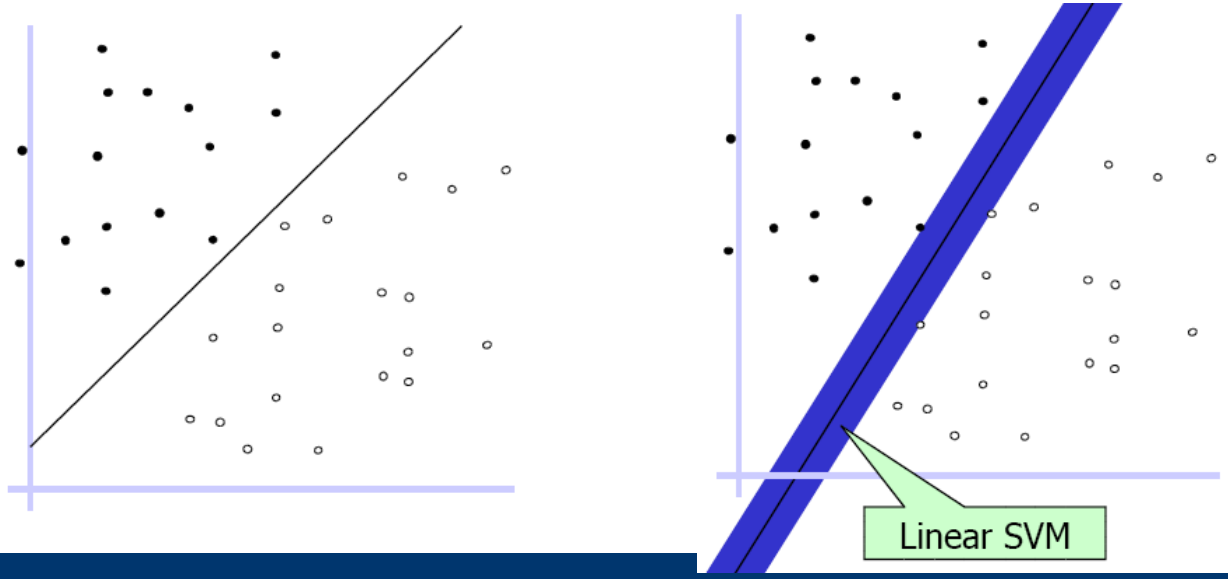
- Hyperplane separating classes in high dimensional space
- But which?



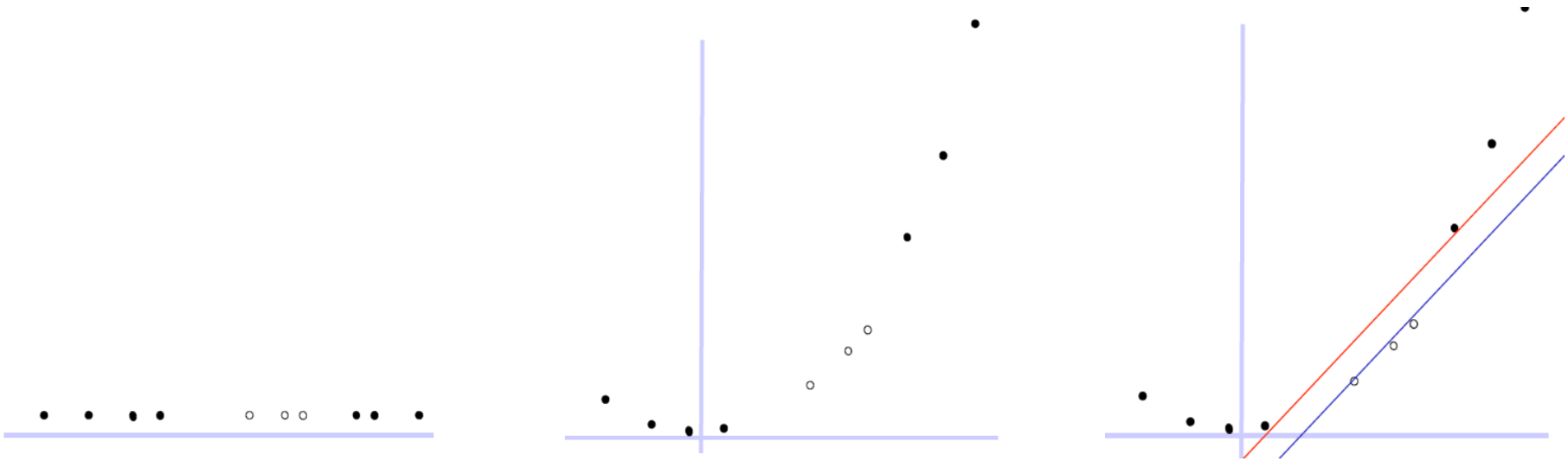
Quelle: Xiaojin Zhu, SVM-cs540

Support Vector Machines (sketch)

- SVMs: Hyperplane which maximizes the margin
  - I.e., is as far away from any data point as possible
  - Cast in a linear optimization problem and solved efficiently
  - Classification only depends on support vectors – efficient
    - Points most closest to hyperplane
  - Minimizes a particular type of error



Kernel Trick: Problems not Linearly Separable



- Map data into an even higher dimensional space
- Not-linearly separable sets may become linearly separable
- Doing this efficiently requires a good deal of work
  - The "kernel trick"

- Questions?

Who should be here

- Master Informatik
  - Also: Wirtschaftsinformatik, Ms.Edu, Diplominformatik
- Ability to read English papers
- Knowledge on text processing
  - Parsing, tokenization, vector space model, stop words, ...
- Knowledge in statistics, probability theory, math
  - Linear equation systems, Bayes theorem, normal distributions, ...

How it will work – traditional part

- Today: Introduction, group formation and choice of topics
- 28.4.17: Lecture on fundamental issues in text processing
  - Also on scientific writing? Presentations?
- Meet advisor before 12.05.17 to discuss topic
- 19.5.2017: Present ideas and method in 5min presentation
- Meet advisor whenever necessary
- Meet your advisor before 30.6.17 to discuss slides
- Present your topic (30-40min) at the Blockseminar
- Write seminar thesis (~20 pages) by 30.8.2017

How it will work – competition part

- Every group has to implement a text classification method
  - Text preprocessing, feature selection, classification
  - Every group uses a different base classifier = topic
  - All other processing steps are free-choice
  - Usage of libraries (Weka, sci-kitlearn, NLTK, OpenNLP...) is allowed
- We will release (probably two) training sets at mid-May
  - Sets of documents with classes assigned
  - Build and optimize your model per set
  - Include description in seminar talk and presentation
- We will evaluate on held-back gold standard data
  - We will use an automated evaluation web platform (Kaggle)
  - Submissions will be possible 1st July until July 7th
  - Small price for best average accuracy among all groups

ToC

- Introduction
- Topics
- Assignment
- Hints on presenting your topic and writing your thesis

General Literature

- To be read by everyone
  - F Sebastiani: Machine learning in automated text categorization, ACM computing surveys, 2002
- Background text processing
  - Manning, C. D., Raghavan, P. and Schütze, H. (2008). "Introduction to Information Retrieval", Cambridge University Press.
  - Manning, C. and Schütze, H. (1999). "Foundations of Statistical Natural Language Processing". MIT Press.
- On classification / machine learning
  - Alpaydin: Introduction to Machine Learning, MIT press 2014
  - Ertel: Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung, Springer 2016

Topics and Groups

Topic	Assigned to (groups of 2-3)	Supervision
Support Vector Machines	Gudd, Wegge, Nguyen	
k-Nearest Neighbors	Tang, Stolte, Abegg	
Decision Trees and Random Forests	Heemann, Gastegger, Velinova	
Artificial Neural Networks	Zambelli, Köhn, Menzel	
(Naive) Bayesian Methods	Wagner, Bauer, Löffler	
(Logistic) Regression	Lakshman, Meyer-Eschenbach, Herholz	
Maximum Entropy Classifier		

ToC

- Introduction
- Topics
- Assignment
- Hints on presenting your topic and writing your thesis

Allgemeine Hinweise

- Dozenten sind ansprechbar!
  - Vorbesprechung des Themas
  - Folien durchgehen
  - Abgrenzung der Ausarbeitung
- Diskussion erwünscht
  - Keine Angst vor Fragen: Fragen sind keine Kritik
  - Eine Frage nicht beantworten können ist in Ordnung
- Tiefe, nicht Breite
  - Lieber das Thema einengen und dafür Details erklären
- Bezug nehmen
  - Vergleich zu anderen Arbeiten (im Seminar)

Allgemeine Hinweise

- Werten und bewerten
  - Keine Angst vor nicht ganz zutreffenden Aussagen – solange gute Gründe vorhanden sind
  - Begründen und argumentieren
  - Kritikloses Abschreiben ist fehl am Platz
- Literaturrecherche ist notwendig
  - Die ausgegebenen Arbeiten sind Anker
  - Weiterführende Arbeiten müssen herangezogen werden
  - Auch Grundlagen nachlesen

- Wir schicken eine Liste zum Abhaken rum

Wie halte ich einen Seminarvortrag

1. Wenn man nun so einen Seminarvortrag halten muss, dann empfiehlt es sich, möglichst lange Sätze auf die Folien zu schreiben, damit die Zuhörer nach dem Vortrag aus den Folienkopien noch wissen, was man eigentlich gesagt hat.

2. Während so einem Vortrag schaut sowieso jeder zum Projektor, also kann man das selbst ruhig auch tun - damit kontrolliert man gleichzeitig auch, ob der Beamer wirklich alles projeziert, was auf dem Laptop zu sehen ist. Ausserdem kann man so den Strom für das Laptop-Display sparen.

3. Übersichtsfolien am Anfang sind langweilig, enthalten keinen Inhalt und nehmen den Zuhörern die ganze Spannung. Schliesslich gibt‘s im Kino am Anfang auch keine Inhaltsangabe.

4. Powerpoint kann viele lustige Effekte, hat tolle Designs und Animationen. Die sollte man zur Auflockerung des Vortrags unbedingt alle benutzen, um zu zeigen, wie gut man das Tool im Griff hat.

5. Nicht zu wenig auf die Folien schreiben. Man weiß ja nie, ob man sie nicht doch ausdrucken muss, und man kann so wertvolle Zeit sparen, wenn man nicht weiterschalten muss.

6. Man sollte versuchen, möglichst lange zu reden. Die Zeitvorgaben sind nur für die Leute, die nicht genug wissen - eigentlich will der Prüfer sehen, dass man sich auch darüber hinaus mit dem Thema beschäftigt hat.

Bloß keine Hervorhebungen im Text – sonst müssen die Zuhörer ja gar nicht mehr aufpassen!

Hinweise zum Vortrag

- 30-40 Minuten plus Diskussion
- Klare Gliederung
- Ab und an Hinweise geben, wo man sich befindet
- Themenauswahl: Lieber verständlich als komplett
- Bilder und Grafiken; Beispiele
- Font: mind. 16pt
- Eher Stichwörter als lange Sätze
- Vorträge können auch unterhaltend sein
  - Gimmicks, Rhythmuswechsel, Einbeziehen der Zuhörer, etc.
- Adressat sind alle Teilnehmer, nicht nur die Betreuer
- Technik: Laptop? Powerpoint? Apple?

Hinweise zur Ausarbeitung

- Eine gedruckte Version abgeben
  - Selbstständigkeitserklärung unterschreiben
- Eine elektronische Version schicken
- Referenzen: Alle verwendeten und nur die
  - Im Text referenzieren, Liste am Schluss
- Korrekt zitieren
  - Vorsicht vor Übernahme von kompletten Textpassagen; wenn, dann deutlich kennzeichnen
  - Aussagen mit Evidenz oder Verweis auf Literatur versehen
- Verwendung von gefundenen Arbeiten im Web
  - Möglich, aber VORSICHT
  - Eventuell Themenschwerpunkt verschieben – Betreuer fragen

Hinweise zur Ausarbeitung –2-

- Gezielt und sachlich schreiben
- Füllwörter vermeiden (dabei, hierbei, dann, ...)
- Knappe Darlegung, präzise Sprache
- Eine gute Gliederung ist die halbe Miete
- Kommen Sie zu Aussagen
  - Vorteile, Nachteile, verwandte Arbeiten, mögliche Erweiterungen, Anwendbarkeit, eigene Erfahrungen, ...

Format

- Benutzung unserer Latex-Vorlage
- Nur eine Schriftart, wenig und konsistente Wechsel in Schriftgröße und –stärke
- Inhaltsverzeichnis
- Bilder: Nummerieren und darauf verweisen
- Referenzen:
  - [1] Yan, X., Yu, P. S. and Han, J. (2004). "Graph Indexing: A Frequent Structure-Based Approach". SIGMOD, Paris, France.
  - [YYH04] Yan, X., Yu, P. S. and Han, J. (2004). "Graph Indexing: A Frequent Structure-Based Approach". SIGMOD, Paris, France.
- Darf man Wikipedia zitieren?
  - Ja, aber nicht dauernd