Processing of large document collections

Part 2 (Text categorization)

Helena Ahonen-Myka

Spring 2006

Text categorization, continues

- problem setting
- machine learning approach
- example of a learning method: Rocchio

2

Text categorization: problem setting

- let
  - D: a collection of documents
  - C = $\{c_1, \ldots, c_{|C|}\}$: a set of predefined categories
  - T = true, F = false
- the task is to approximate the unknown target function Φ': D x C -> {T,F} by means of a function Φ : D x C -> {T,F}, such that the functions "coincide as much as possible"
- function Φ' : how documents should be classified
- function Φ : classifier (hypothesis, model…)

3

Some assumptions

- categories are just symbolic labels
  - no additional knowledge of their meaning is available
- no knowledge outside of the documents is available
  - all decisions have to be made on the basis of the knowledge extracted from the documents
  - metadata, e.g., publication date, document type, source etc. is not used

4

Some assumptions

- methods do not depend on any application-dependent knowledge
  - but: in operational ("real life") applications all kind of knowledge can be used (e.g. in spam filtering)
- note: content-based decisions are necessarily subjective
  - it is often difficult to measure the effectiveness of the classifiers
  - even human classifiers do not always agree

5

Variations of problem setting: single-label, multi-label text categorization

- single-label text categorization
  - exactly 1 category must be assigned to each $d_j \in D$
- multi-label text categorization
  - any number of categories may be assigned to the same $d_j \in D$

6

Variations of problem setting: single-label, multi-label text categorization

- special case of single-label: binary
  - each d_j must be assigned either to category c_i or to its complement $\neg c_i$
- the binary case (and, hence, the single-label case) is more general than the multi-label
  - an algorithm for binary classification can also be used for multi-label classification
  - the converse is not true

7

Variations of problem setting: single-label, multi-label text categorization

- in the following, we will use the binary case only:
  - classification under a set of categories C = set of |C| independent problems of classifying the documents in D under a given category c_i, for i = 1, ..., |C|

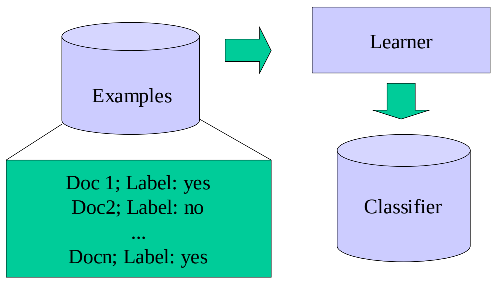
8

Machine learning approach to text categorization

- a general program (learner) automatically builds a classifier for a category c_i by observing the characteristics of a set of documents manually classified under c_i or $\neg c_i$ by a domain expert
- from these characteristics the learner extracts the characteristics that a new unseen document should have in order to be classified under c_i
- use of classifier: the classifier observes the characteristics of a new document and decides whether it should be classified under c_i or $\neg c_i$

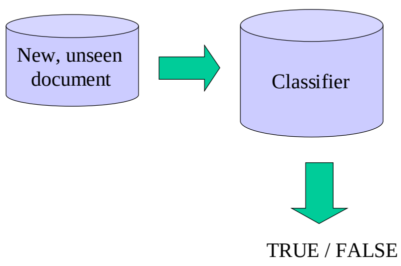
9

Classification process: classifier construction



10

Classification process: use of the classifier



11

Supervised learning from examples

- initial corpus of manually classified documents
  - let d_j belong to the initial corpus
  - for each pair $<d_j, c_i>$ it is known if d_j is a member of c_i
- positive and negative examples of each category
  - in practice: for each document, all its categories are listed
    - if a document d_j has category c_i in its list, document d_j is a positive example of c_i
    - negative examples for c_i : the documents that do not have c_i in their list

12

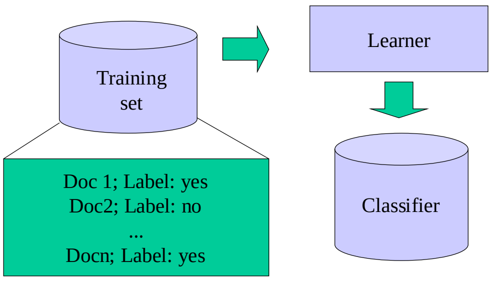
Training set and test set

- the initial corpus is divided into two sets
  - a training set
  - a test set
- the training set is used for building the classifier
- the test set is used for testing the effectiveness of the classifier
  - each document is fed to the classifier and the decision is compared to the manual category
- the documents in the test set are not used in the construction of the classifier

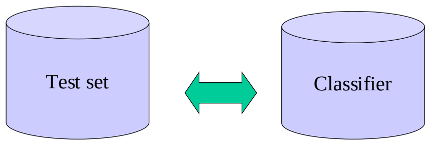
Training set and test set

- the classification process may have several implementation choices: the best combination is chosen by testing the classifier
- alternative: k-fold cross-validation
  - k different classifiers are built by partitioning the initial corpus into k disjoint sets and then iteratively applying the train-and-test approach on pairs, where k-1 sets construct a training set and 1 set is used as a test set
  - individual results are then averaged

Classification process: classifier construction



Classification process: testing the classifier



Strengths of machine learning approach

- learners are domain independent
  - usually available 'off-the-shelf'
- the learning process is easily repeated, if the set of categories changes
  - only the training set has to be replaced
- manually classified documents often already available
  - manual process may exist
  - if not, it is still easier to manually classify a set of documents than to build and tune a set of rules

Examples of learners

- Rocchio method
- probabilistic classifiers (Naïve Bayes)
- decision tree classifiers
- decision rule classifiers
- regression methods
- on-line methods
- neural networks
- example-based classifiers (k-NN)
- boosting methods
- support vector machines

Rocchio method

- learning method adapted from the relevance feedback method of Rocchio
- for each category, an explicit profile (or prototypical document) is constructed from the documents in the training set
  - the same representation as for the documents
  - benefit: profile is understandable even for humans
- profile = classifier for the category

19

Rocchio method

- a profile of a category is a vector of the same dimension as the documents
  - in our example: 118 terms
    - categories medicine, energy, and environment are represented by vectors of 118 elements
  - the weight of each element represents the importance of the respective term for the category

20

Rocchio method

- weight of the k^{th} term of category i:

$$w_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{w_{kj}}{|NEG_i|}$$

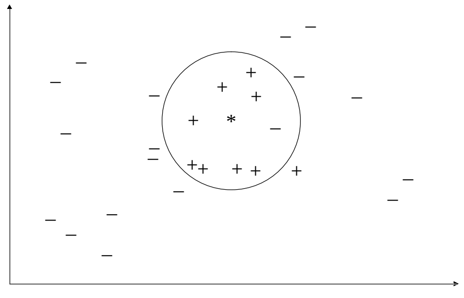
- W_{kj}: weight of the k^{th} term of document j
- POS_i: set of positive examples
  - documents that are of category i
- NEG_i: set of negative examples

21

Rocchio method

- in the formula, β and γ are control parameters that are used to set the relative importance of positive and negative examples
- for instance, if β=2 and γ=1, we do not want the negative examples to have as strong influence as the positive examples
- if β=1 and γ=0, the category vector is the centroid (average) vector of the positive sample documents

22



23

Rocchio method

- in our sample dataset: what is the weight of term 'nuclear' in the category 'medicine'?
  - $POS_{medicine}$ contains the documents Doc1-Doc4
  - $NEG_{medicine}$ contains the documents Doc5-Doc10
  - $|POS_{medicine}|$ = 4 and $|NEG_{medicine}|$ = 6

24

Rocchio method

- the weights of term ´nuclear´ in documents in $POS_{medicine}$
  - w_nuclear_doc1 = 0.5
  - w_nuclear_doc2 = 0
  - w_nuclear_doc3 = 0
  - w_nuclear_doc4 = 0.5
- and in documents in $NEG_{medicine}$
  - w_nuclear_doc6 = 0.5

25

Rocchio method

- let $\beta=2$ and $\gamma=1$
- weight of 'nuclear' in the category 'medicine':
  - w_nuclear_medicine = 2* (0.5 + 0.5)/4 – 1 * 0.5/6 = 0.5 - 0.08 = 0.42

26

Rocchio method

- using the classifier: cosine similarity of the category vector c_i and the document vector d_j is computed
  - |T| is the number of terms

$$S(c_i, d_j) = \frac{\sum_{k=1}^{|T|} w_{ki} \cdot w_{kj}}{\sqrt{\sum_{k=1}^{|T|} w_{ki}^2} \cdot \sqrt{\sum_{k=1}^{|T|} w_{kj}^2}}$$

27

Rocchio method

- the cosine similarity function returns a value between 0 and 1
- a threshold is given
  - if the value is higher than the threshold -> true (the document belongs to the category)
  - otherwise -> false (the document does not belong to the category)

28

Rocchio method

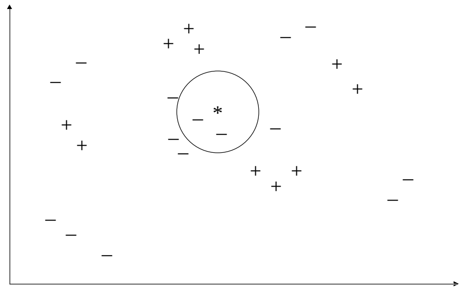
- a classifier built by means of the Rocchio method rewards
  - closeness of a (new) document to the centroid of the positive training examples
  - distance of a (new) document from the centroid of the negative training examples

29

Strengths and weaknesses of Rocchio method

- strengths
  - simple to implement
  - fast to train
- weakness
  - if the documents in a category occur in disjoint clusters, a classifier may miss most of them
    - e.g. two types of Sports news: boxing and rock-climbing
    - the centroid of these clusters may fall outside all of these clusters

30



31

Enhancement to the Rocchio Method

- instead of considering the set of negative examples in its entirety, a smaller sample can be used
  - for instance, the set of near-positive examples
- near-positives ($NPOS_c$): the most positive amongst the negative training examples

32

Enhancement to the Rocchio Method

- the new formula:

$$w_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NPOS_i\}} \frac{w_{kj}}{|NPOS_i|}$$

33

Enhancement to the Rocchio Method

- the use of near-positives is motivated, as they are the most difficult documents to distinguish from the positive documents
- near-positives can be found, e.g., by querying the set of negative examples with the centroid of the positive examples
  - the top documents retrieved are most similar to this centroid, and therefore near-positives
- with this and other enhancements, the performance of Rocchio is comparable to the best methods

34

Other learners

- we discuss later:
  - Boosting (AdaBoost)
  - Naive Bayes (in text summarization)

35