Feature Subset Selection for Text Categorization

Jana Novovičová and Petr Somol

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Prague, Czech Republic

http://ro.utia.cas.cz

2nd International Workshop on Data-Algorithm-Decision Making, 2006, Třešť, Czech Republic

Outline

Objective

Aim of Text Classification:

partition an unstructured collection of documents expressed in natural language into meaningful groups (categories, classes, labels).

Two main variants of text classification:

- **Text clustering** - finding a latent yet undetected group structure
- **Text categorization** (TC) (a.k.a. classification or topic spotting) - labelling text documents from a domain with thematic classes from a set of predefined classes

Definition of TC

- **Given:**
  - a fixed set of predefined classes: $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$
  - a document $d_i \in \mathcal{D}$, where $\mathcal{D}$ is the domain of documents
- **We want:**
  - to assign a Boolean value to each pair $(d_i, c_j) \in \mathcal{D} \times \mathcal{C}$
  - a value of T indicates a decision to file d_i under c_j, while a value of F indicates a decision not to file d_i under c_j
- **We essentially want:**
  - to approximate the unknown **target (classification) function**

$$\Psi : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$$

by means of a function

$$\hat{\Psi} : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$$

called the **classifier** *(**rule, hypothesis**)*, such that

Ψ and $\hat{\Psi}$ "coincide as much as possible".

Main Approaches to TC

- **The knowledge engineering approach**
  - **manually** building a set of rules
- **The machine learning approach**
  - a classifier for set $\mathcal{C}$ can be built **automatically** by supervised machine learning techniques from a training set of documents pre-classified under $\mathcal{C}$

Main Phases in Classification

- Document indexing - i.e. creation of representations for documents
- Classifier learning - i.e. creation of a classifier by learning from the representation of the documents from training set
- Evaluation the effectiveness of the classifier
  tested by applying it to test set

Document Representation

Document Indexing Procedure:

maps a text document into a compact representation of its content

Text document - represented as a vector of terms

Terms (a.k.a features) - associated with words that occur in the documents of the training set:

- single words
- word combinations
- phrases

Document Representation

Bag of Words approach

Each document - represented by vector $d_i = (t_{i1}, ..., t_{i|\mathcal{V}|})$

$\mathcal{V} = \{w_1, \ldots, w_{|\mathcal{V}|}\}$ – the vocabulary set of size $|\mathcal{V}|$ containing distinct words occurred in the training documents

Each term variable t_{iv} indicates:

- the presence or absence of the word w_v
- some measure of the frequency of the word w_v

- **High dimensionality** (tens of thousands) of the term space
  - a common characteristic of text data
- Many learning algorithms do not cope with a large term space
- **Term (Feature) Selection**
  - dominant approach to dimensionality reduction in TC

  A "good" subset of terms
  - may result in higher classification accuracy
  - reduces the computational complexity
- **Term evaluation criteria and term selection methods**
  - two dominating factors in designing a term selection algorithm

Traditional TEF

Term Evaluation Functions (TEF):

- **Document frequency** of a certain word $w \in \mathcal{V}$
- **Information-theoretic term selection functions**
  - Information gain (IG)
  - Chi-square statistic (χ^2)
  - Mutual information (MI)

Generally, IG and χ^2 better than MI

TEF - specified "locally" to a specific class in $\mathcal{C}$

Globalization techniques:

- the *sum*
- the *weighted sum*
- the *maximum*

of their class-specific values are usually computed.

Traditional TEM

Term Evaluation Methods (TEM):

Feature subset selection in text learning – simplified with the assumption of feature independence.

Best individual features (BIF) method consists:

- in scoring each term by means of class-based term evaluation function
- in selecting a subset of terms that maximize term evaluation function

BIF methods completely ignore the existence of other words and the manner how the words work together.

Types of Classifiers in TC

Supervised learning methods often used in TC:

- Naive Bayes (McCallum et al., 1998)
- Neural networks (Weiner, 1995)
- Nearest neighbors (Yang, 1999)
- Decision trees (Lewis and Ringuette, 1996)
- Support vector machines (Joachims, 1998)
- Regression methods (Yang, 1999)
- Boosting methods (Schapire and Singer, 2000)

What Classifiers are Best?

What Classifiers are Best in TC?

- Support Vector Machines and Boosting methods generally performs well.
- Naive Bayes has displayed a low performance among learning classifiers.
  - Advantages:
    - explicit theoretical foundation
    - simple, easy to implement
    - fast in learning and classification
  - Disadvantages:
    - conditional independence assumption is violated by real-world data
- The performance of classifiers may depend on a number of experimental factors
  - e.g. characteristics of the document sets, the number of training examples per class, etc.

Probabilistic Document Model

Proposed Document Representation:

- **Bag of words approach**

  The document d_i is considered as $|\mathcal{V}|$-dimensional vector

  $d_i = (N_{i1}, \ldots, N_{i|\mathcal{V}|})$

  N_{iv} – the number of times certain word $w_v \in \mathcal{V}$ occurs in d_i

Probabilistic Document Model

- **Multinomial Model**
  - class-conditional probability

$$p(d_i|c_j) = \frac{|d_i|!}{\prod_{v=1}^{|\mathcal{V}|} N_{iv}!} \prod_{v=1}^{|\mathcal{V}|} P(w_v|c_j)^{N_{iv}}$$

$P(w_v|c_j)$ – the probability that a word chosen randomly in a document from c_j equals w_v

$|d_i| = \sum_{v=1}^{|\mathcal{V}|} N_{iv}$ – the length of d_i

  - unconditional probability of d_i

$$p(d_i) = \sum_{j=1}^{|\mathcal{C}|} P(c_j) p(d_i|c_j), \quad 0 \leq P(c_j) \leq 1, \quad \sum_{j=1}^{|\mathcal{C}|} P(c_j) = 1$$

$P(c_j)$ – the prior probability that d_i belongs to c_j

Global Term Subset Selection

Proposed Approach to Dimensionality Reduction:

Global Term Subset Selection

- **Given:**
  the initial set $\mathcal{V}$ of words
- **Determine:**
  the subset $\mathcal{S}_r \subset \mathcal{V}$ of r words that maximizes the global term evaluation function J:

$$\mathcal{S}_r = \arg\max_{\mathcal{S} \subseteq \mathcal{V}} \{J(\mathcal{S})\}$$

Bhattacharyya distance

The Bhattacharyya distance between two class-conditional density functions $p(\mathbf{x}|c_j)$ and $p(\mathbf{x}|c_k)$, $\mathbf{x} \in \mathcal{X}$ - pairwise Bhattacharyya - distance is defined as follows:

$$B_{jk} = -\log \int_{\mathcal{X}} \sqrt{p(\mathbf{x}|c_j)p(\mathbf{x}|c_k)}d\mathbf{x}$$

Distance measure can be extended to the multiclass case by evaluating all pairwise distances between classes

$$B = \sum_{j=1}^{|\mathcal{C}|-1} \sum_{k=j+1}^{|\mathcal{C}|} P(c_j)P(c_k)B_{jk}$$

Bhattacharyya distance for multinomial model

Proposed Term Evaluation Function:

- **Multiclass Bhattacharyya distance** of d_i for multinomial distribution:

$$B(d_i) = \sum_{j=1}^{|\mathcal{C}|-1} \sum_{k=j+1}^{|\mathcal{C}|} P(c_j)P(c_k)B_{jk}(d_i)$$

$B_{jk}(d_i)$ – pairwise Bhattacharyya distance of d_i between c_j and c_k:

$$B_{jk}(d_i) = -|d_i| \log \sum_{v=1}^{|\mathcal{V}|} \sqrt{P(w_v|c_j)P(w_v|c_k)}$$

Individual Bhattacharyya distance for multinomial model

- **Individual Bhattacharyya distance** for one term in the document d_i corresponding to w_v

$$B(w_v) = \sum_{j=1}^{|\mathcal{C}|-1} \sum_{k=j+1}^{|\mathcal{C}|} P(c_j)P(c_k)B_{jk}(w_v)$$

$$B_{jk}(w_v) =$$

$$-|d_i| \log \left(\sqrt{P(w_v|c_j)P(w_v|c_k)} + \sqrt{(1 - P(w_v|c_j))(1 - P(w_v|c_k))} \right)$$

Oscillating Search

Proposed Term Selection Search Method:

Oscillating Search (OS) (Somol and Pudil, 2000)

A new suboptimal subset search method for FS

As opposed to other sequential subset selection methods OS:

- is not dependent on pre-specified direction of search (forward or backward)
- overcomes effectively the "nesting" problem
- may be restricted by a time-limit, what makes it usable in real-time systems

Oscillating Search

OS is based on repeated modification of the current subset $\mathcal{V}_r$

- Down-swing: removes o "worst" features from the current set $\mathcal{V}_r$ to obtain a new set $\mathcal{V}_{r-o}$ at first, then adds o best features from $\mathcal{V} \setminus \mathcal{V}_{r-o}$ to $\mathcal{V}_{r-o}$ to obtain a new current set $\mathcal{V}_r$.
- Up-swing: adds o "best" features from $\mathcal{V} \setminus \mathcal{V}_r$ to the current set $\mathcal{V}_r$ to obtain a new set $\mathcal{V}_{r+o}$ at first, then removes o "worst" ones from $\mathcal{V}_{r+o}$ to obtain a new current set $\mathcal{V}_r$ again.
- The up- and down-swings are repeated as long as the set $\mathcal{V}_r$ gets improved.

  $o = 1$ initially and may be later increased to allow more thorough search at a cost of more computational time.
- The algorithm then terminates when o exceeds a user-specified limit Δ.

Initialization of OS

Initialization of OS

The simplest ways

- Random selection
- Sequential Forward Selection procedure

The perfect way for TC

- **Best Individual Features**

Oscillating Search Algorithm

The simplest form of the algorithm ($o = 1$):

1. **Step 1: Initialization**
   Find the initial set $\mathcal{V}_r$ by means of BIF. Let $c = 0$.

2. **Step 2: Down-swing**
   - Remove such feature from $\mathcal{V}_r$, so that the new set $\mathcal{V}_{r-1}$ retains the highest criterion value.
   - Add such feature from $\mathcal{V} \setminus \mathcal{V}_{r-1}$ to $\mathcal{V}_{r-1}$, so that the new subset $\mathcal{V}_r^{new}$ yields the highest criterion value.
   - If $\mathcal{V}_r^{new}$ is better than $\mathcal{V}_r$, let $\mathcal{V}_r = \mathcal{V}_r^{new}$, $c = 0$ and go to Step 4.

3. **Step 3: Last swing did not find better solution**
   Set $c = c + 1$. If $c = 2$, then none of previous two swings has found better solution; stop the algorithm.

Oscillating Search

4. **Step 4: Up-swing**

   - Add such feature from $\mathcal{V} \setminus \mathcal{V}_r$ to $\mathcal{V}_r$, so that the new set $\mathcal{V}_{r+1}$ has the highest criterion value.
   - Remove such feature from $\mathcal{V}_{r+1}$, so that the new set $\mathcal{V}_r^{new}$ yields the highest criterion value.
   - If $\mathcal{V}_r^{new}$ is better than $\mathcal{V}_r$, let $\mathcal{V}_r = \mathcal{V}_r^{new}$, $c = 0$ and go to Step 2.

5. **Step 5: Last swing did not find better solution**
   Let $c = c + 1$. If $c = 2$, then none of previous two swings has found better solution; stop the algorithm. Otherwise go to Step 2.

Data set:

Reuters-21578 (news articles)

http://www.daviddlewis.com/resources/testcollections/reuters21578

Reuters-21578 after pre-processing

(Stop-words elimination, Stripping, Stemming)

- number of training documents: 9603
- number of classes: 33
- vocabulary size: 10 105 words
- the largest class contained 3924 non-zero documents
- the smallest class contained 19 non-zero documents.

Examined FS Methods

Feature selection methods used in our experiments:

1. Best individual features (BIF)
   - Individual Bhattacharyya distance (BIF BD)
   - Information gain (BIF IG)
2. Oscillating search
   - Bhattacharyya distance on groups of features (initialized by feature subsets found by means of BIF IB).

Bayes Classifier for Text

Bayes classifier with multinomial model

- **Bayes Theorem:**

$$P(c_j|d_i) = \frac{P(c_j)p(d_i|c_j)}{p(d_i)}$$

- **Bayes Classifier:**
  - predict class for document d_i with largest posterior probability

$$c^* = \arg\max_{c_j \in \mathcal{C}} P(c_j|d_i) = \arg\max_{c_j \in \mathcal{C}} P(c_j) \frac{|d_i|!}{\prod_{v=1}^{|\mathcal{V}|} N_{iv}!} \prod_{v=1}^{|\mathcal{V}|} P(w_v|c_j)^{N_{iv}}$$

Linear Support Vector Machine

- **Linear Support Vector Machines** (Joachims, 1998)

  SVMs attempt to build a classifier that maximizes the margin i.e., the minimum distance between the hyperplane that represents the classifier and the vectors that represent the documents.

  For our experiments we used:

  - LibSVM implementation – `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
  - Standard C-SVC form of the classifier with default value of $C = 1$.
  - No data scaling has been done.

k-fold cross-validation

Text classifier construction relies:

- on the existence of an initial set $\Omega = \{d_1, \cdots, d_{|\Omega|}\} \subset \mathcal{D}$ of documents pre-classified under $\mathcal{C}$

k different classifiers are built by

- partitioning the initial pre-classified set into k disjoint sets $\mathcal{D}e_1, \cdots, \mathcal{D}e_k$
- then iteratively applying train and test approach on pairs $(\mathcal{D}V_i = \Omega \setminus \mathcal{D}test_i, \mathcal{D}test_i)$
- The final effectiveness is obtained by individually computing the effectiveness of the k classifiers, and then averaging the individual results in some ways.

Accuracy

Measuring Classification Effectiveness:

- **Accuracy:**

  estimated as

$$\hat{A} = \frac{\sum_{k=1}^{|\mathcal{C}|} T_k}{\sum_{k=1}^{|\mathcal{C}|}(T_k + F_k)}$$

T_k (F_k) - the number of documents correctly (incorrectly) assigned to c_k;

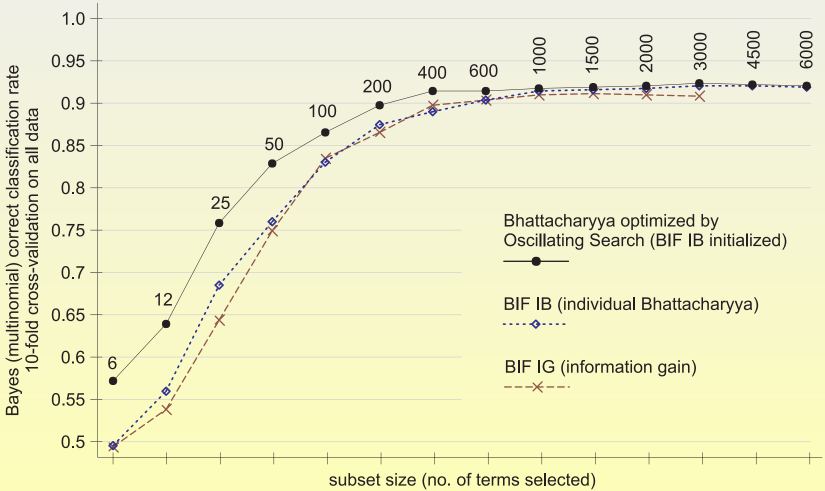
All tests have been done by means of 10-fold cross-validation over the whole data set.

Experimental Results

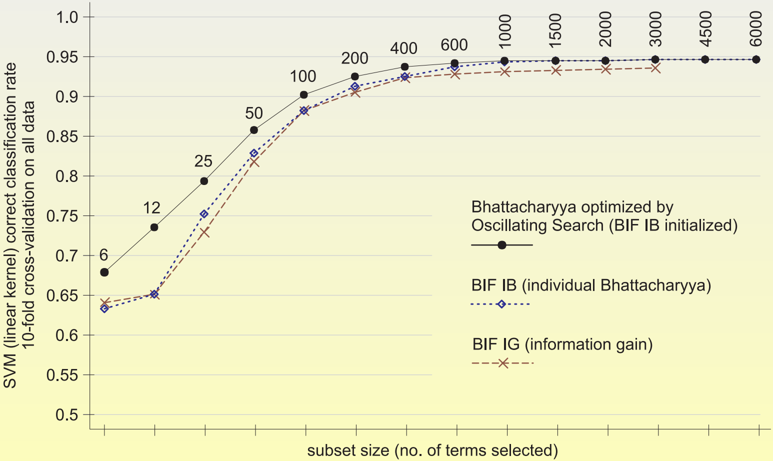
The presented experimental results illustrate that:

- Oscillating Search is constantly superior to BIF approach for subset sizes roughly ≤ 1000.
- Improvement of accuracy is equally notable for both of the tested classifiers.
- For larger subsets the improvement is hardly observable or not present at all. The search time then becomes inadequate.
- The time requirements of the OS procedure stay in reasonable limits.
- Slight superiority of individual Bhattacharyya over information gain in BIF search.

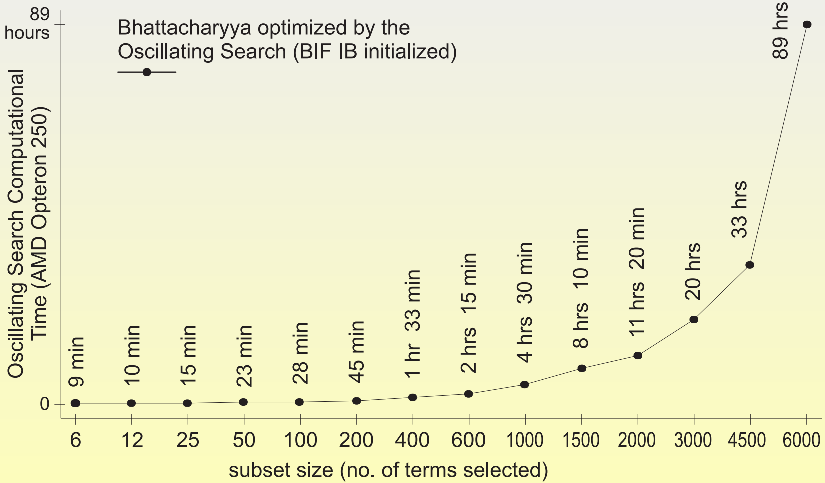
Multinomial Bayes classifier: 10-fold cross-validated classification rate.



SVM classifier: 10-fold cross-validated classification



Oscillating Search computational time



Conclusions

We have proposed for Text Classification problem:

- To use the multiclass Bhattacharyya distance for multinomial model as the global term selection criterion.
- Oscillating Search method as a term selection search procedure.

Conclusions

Experimental results illustrate that proposed OS algorithm

- brings substantial improvement in classification accuracy over traditional individual term evaluation based methods.
- is computationally feasible.
- Multinomial Bhattacharyya distance is a good measure for both group-wise and individual term selection

Future work

Ongoing work could include:

- Investigation in more detail the applicability of alternative Oscillating Search versions (Somol, 2000).
- SVM parameter optimization in the FS process.
- Simultaneous feature selection and classification of text documents using mixture model for class-conditional probabilities.
  (Pudil, Novovičová and Kittler, PR, 1995;
  Novovičová, Pudil and Kittler, IEEE PAMI 1996).
- Semi-supervised learning- the problem of learning text classifiers mainly from unlabelled data unfortunately is still open.

References

- G. Forman. *An Experimental Study of Feature Selection Metrics for Text Categorization.* Journal of Machine Learning Research, 3, 1289–1305, 2003.
- T. Joachims. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features.* Proc. of the ECML 98, 137–142, 1998.
- K. Nigam and A. K. McCallum and S. Thrun and T. Mitchell. *Text classification from labeled and unlabeled documents using EM.* Machine Learning, 39: 103–134, 2000.
- J. Novovičová, P. Somol, and P. Pudil. *Oscillating Feature Subset Search Algorithm for Text Categorization.* Lecture Notes in Computer Science, 4225: 572-587, 2006.

- F. Sebastiani. *Machine Learning in Automated Text Categorization.* ACM Computing Surveys, 34: 1–47, 2004.
- P. Somol and P. Pudil. *Oscillating search algorithms for feature selection.* Proc. of the 15th IAPR Internat. Conf. Pattern Recognition, 406–409, 2000.
- Y. Yang, J. Zhang and B. Kisiel. *A scalability analysis of classifier in text categorization.* Proc. of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, 2003.