The Theory of Text Categorization

Ko, Youngjoong

Sungkyunkwan University

A definition of the TC task

- Text Categorization (Sebastiani, 2002)
  - Assign documents to one or more of a predefined set of categories
  - The task of automatically determining an assignment of a value from {0,1} to each entry of the decision matrix.

	d_1	…	…	d_j	…	…	d_n
c_1	a_{11}	…	…	a_{1j}	…	…	a_{1n}
…	…	…	…	…	…	…	…
c_i	a_{i1}	…	…	a_{ij}	…	…	a_{in}
…	…	…	…	…	…	…	…
c_m	a_{m1}	…	…	a_{mj}	…	…	a_{mn}

  - where
    - $C = \{c_1,\ldots,c_m\}$ is a set of pre-defined categories
    - $D = \{d_1,\ldots,d_n\}$ is a set of documents to be categorized

A definition of the TC task

- The formal notation
  - To approximate the unknown function $f : D*C \rightarrow \{0,1\}$ by means of a function $\hat{f} : D*C \rightarrow \{0,1\}$ (the classifier, or model, or hypothesis) such that f and $\hat{f}$ coincide *as much as possible*.

A definition of the TC task

- Different constraints depending on the application
  - Single-label case : exactly one category must be assigned to each document
  - Multi-label case : general case

  - We will focus on the more general multi-label case and assume that categories are stochastically independent of each other
    - $f(d_j, c')$ does not depend on $f(d_j, c'')$.
  - The classification problem for the $D*C$ decision matrix
    - The m independent problems of categorizing the documents in D under a category c_i, for $i = 1,\ldots,m$.
    - A classifier for c_i is a function $f_i': D \rightarrow \{0,1\}$ that approximates an unknown function $f_i': D \rightarrow \{0,1\}$

A definition of the TC task

- Two category and document-pivoted categorization methods to make a decision matrix
  - CPC (category-pivoted categorization) : one row at a time
  - DPC (document-pivoted categorization) : one column at a time
- The sets of categories and documents are not always available from the start
  - DPC :
    - If a user submits one document at a time for categorization, the categories may be ranked in decreasing order of estimated appropriateness for the document
  - CPC :
    - a new category may be added to a set of categories after a number of documents have already been categorized under the set of categories

Information Retrieval Techniques

- Content-based document management tasks
  - TC and IR
- Three phases of the TC *system life cycle*
  - IR-style *indexing* is always performed on all documents
  - IR-style *techniques* (such as document-request matching, …) are typically used in the inductive construction of the classifiers
  - IR-style *evaluation* of the effectiveness of the classifiers is performed

Indexing Technique

- The two choices of text representation
  - ***Lexical semantics (Unigram)***
  - Compositional semantics (Bigram, trigram …)
  - Lewis have found that more sophisticated representations (linguistic phrases, statistical phrases, etc) yield worse effectiveness

- The *bag of words* approach
  - Consider a document as a bag; it contains many words
  - The vector of a document: n weighted terms (or *features*) t_k that occur in d_j.
  - Weight (w_{kj})
    - [0,1]: the most frequent case
    - {0,1}: presence or absence of t_k in d_j

The Preprocessing of Indexing

- Removal of stop words
  - Topic-neutral words
  - Function words (articles, prepositions, conjunctions, etc)

- Stemming
  - Its utility is controversial.
  - http://www.tartarus.org/~martin/PorterStemmer/index.html

The machine learning approaches for TC

- In the 80's, the typical approach is a hand-crafting *expert system* which uses a set of rules of type
  - **If** <*conjunction of terms*> **then** <*category*>
    - bushels & expert → wheat
  - The drawback of this "manual" approach
    - *Knowledge acquisition bottleneck*

- In the 90's, the machine learning approach appears
  - A general inductive process automatically builds a classifier for a category
  - Advantages of this approach
    - construction not of a classifier, but of an automatic builder of classifiers (learner)
    - The effectiveness of these classifiers matches that of hand-crafted classifiers

Methods for Constructing Classifiers

- Two Phases of the inductive construction
  - The *Categorization Status Value (CSV)* function
    - The classifier is a function to generate a prediction score (CSV)
    - $CSV_i : \mathrm{D} \rightarrow [0,1]$ (given d_j, for category c_i)
  - The definition of a *threshold* τ_i
    - $CSV_i(d_j) \geq \tau_i$: *a decision to categorize* d_j *under* c_i
    - $CSV_i(d_j) < \tau_i$: *a decision not to categorize* d_j *under* c_i
  - A particular case
    - The classifier already provides a binary judgment
      - $CSV_i : \mathrm{D} \rightarrow \{0,1\}$
      - Decision Tree

Probabilistic Classifiers

- Naïve Bayes classifiers (McCallum & Nigam, 1998)
  - View $CSV_i(d_j)$ in terms of Bayes' theorem

$$P(c_i \mid d_j) = \frac{P(c_i)P(d_j \mid c_i)}{P(d_j)}$$

  - Use of the independence assumption for $P(d_j|c_i)$

$$P(d_j \mid c_i) = \prod_{k=1}^{r} P(w_{kj} \mid c_i)$$

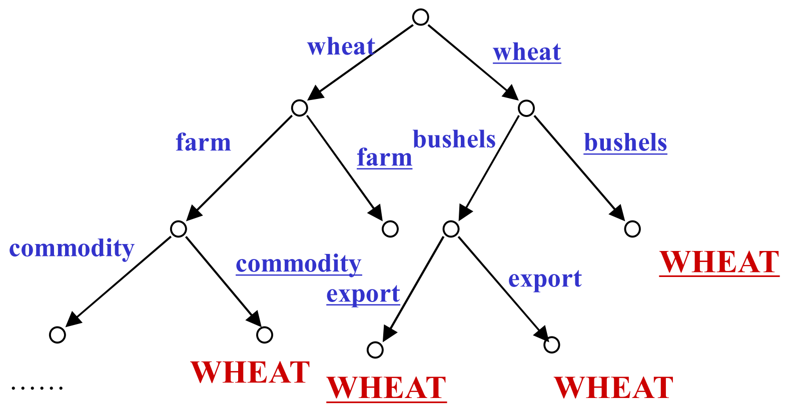
- Important research directions
  - To introduce non-binary document weights
  - To introduce document length normalization
  - To relax the independence assumption

Neural Networks

- (Wiener, Pedersen, and Weigend, 1995)
- A *neural network (NN)* TC system is a network of units
  - *Input units* : terms appearing in the document
  - *Output units* : categories to be assigned
- NNs are trained by backpropagation
- Linear *NNs* vs. non-linear *NNs*
  - Non-linear component provides absolutely no advantage

Decision Tree Classifiers

- Build a binary Tree (Lewis and Ringuette, 1994)
  - Internal nodes : labeled by index terms
  - Branches : the values that the index term has in the representation of the test document
  - Leaf nodes : labeled by categories



Example-based Classifiers

- The distance weighted *k*-NN (Yang, 94)
  - The ***k*-nearest neighbors algorithm** is amongst the simplest of all machine learning algorithms
  - Classified by a majority vote of its neighbors:

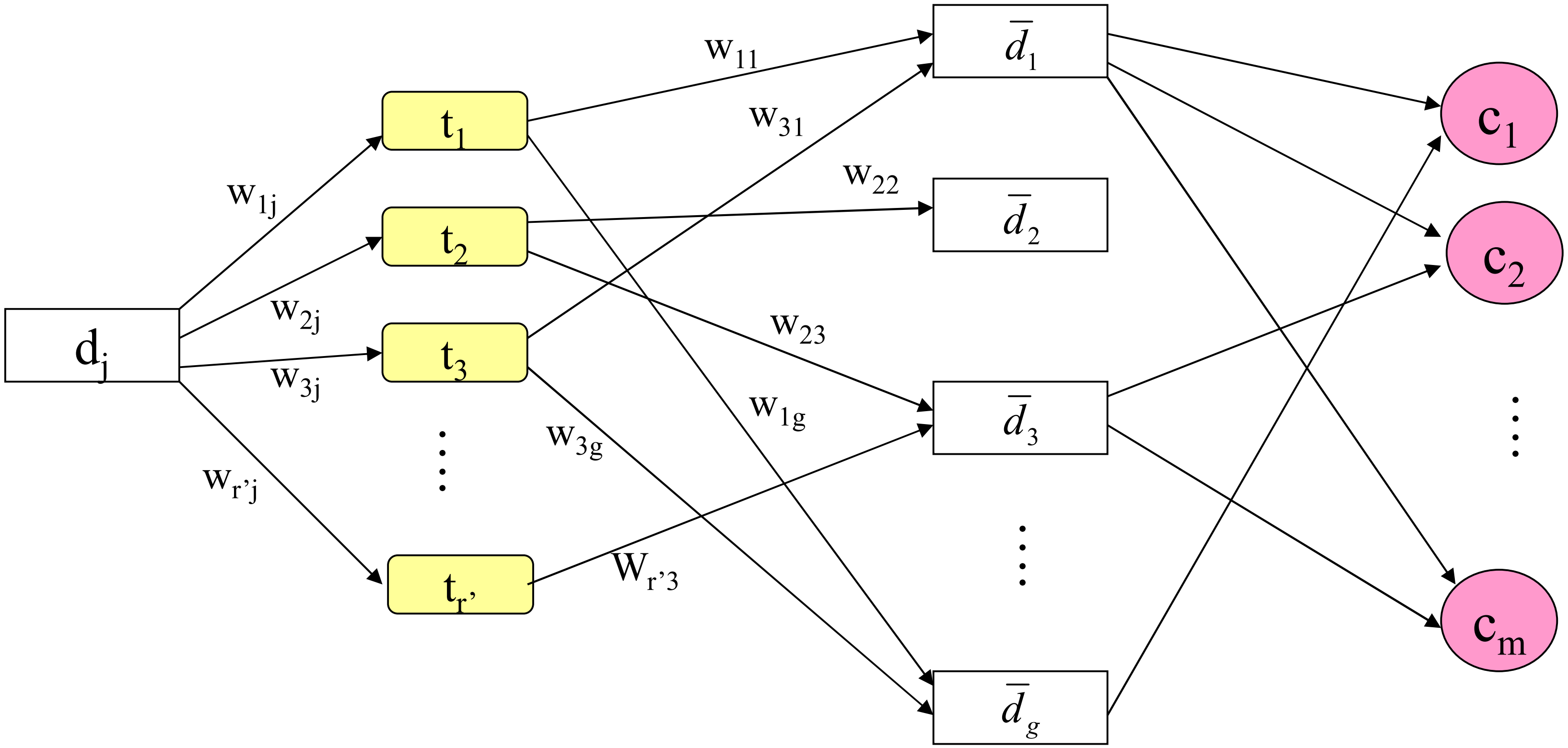
$$\boxed{CSV_i(d_j) = \sum_{\bar{d}_z \in Tr_k(d_j)} RSV(d_j, \bar{d}_z) \cdot b_{iz}}$$

  - $RSV(d_j, \bar{d}_z)$: a measure or semantic relatedness between
    - Ex) vector-based measures : inner-product, cosine similarity d_j *and* $\bar{d}_z$
  - The b_{iz} values are from the correct decision matrix of {0,1}
  - $Tr_k(d_j)$ is the set of the *k* documents $\bar{d}_z$ for which $RSV(d_j, \bar{d}_z)$ is maximum : the *k* value should be determined on a validation set

Example-based Classifiers

- Advantages
  - *High performance*, Not suffer from the “linear separation problem”
- Drawbacks
  - *Too late running time*, lazy learners.

The *k*-NN Classifier



REQUEST → TERMS → TRAINING DOCUMENT → CATEGORIES

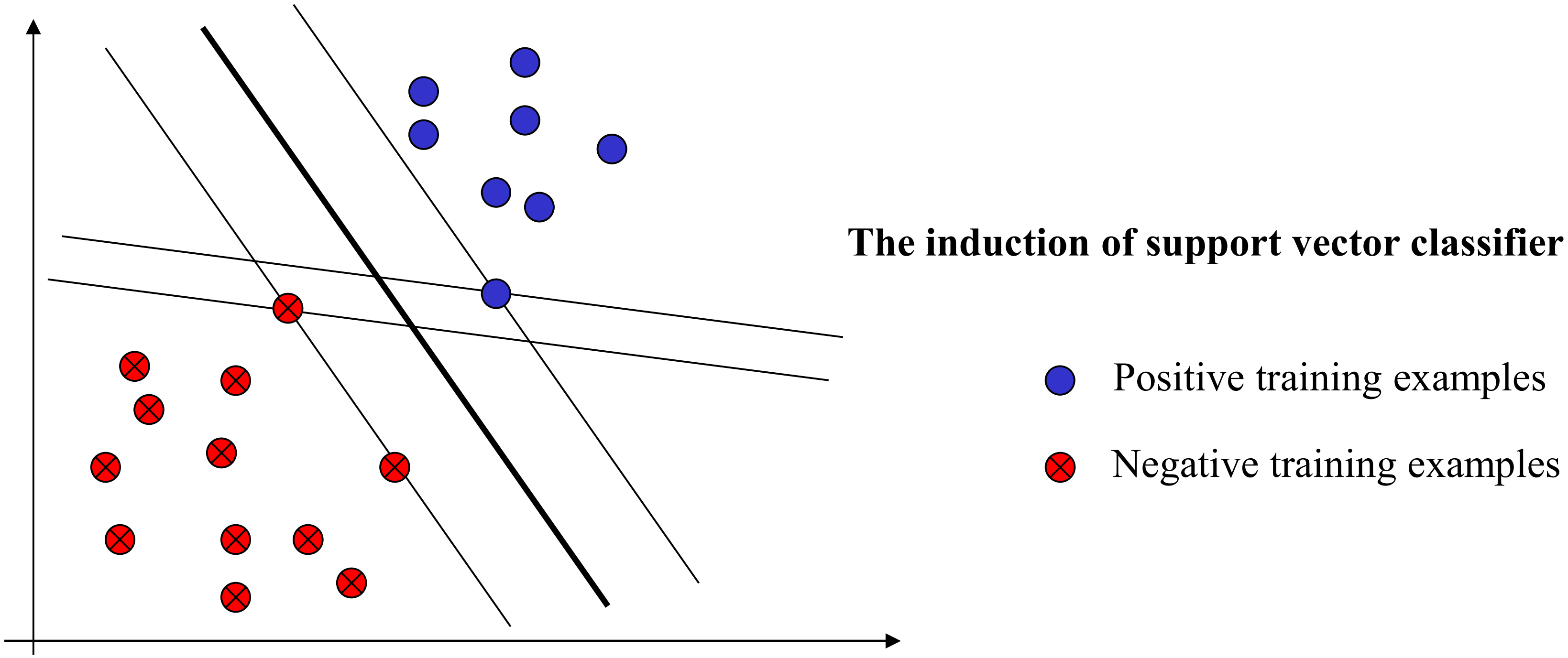
Request Weights

Training Document Weights

Category Assignments

SVM

- The support vector machine (Joachims, 1998)
  - To find *the surface* σ that separate the positive from the negative training examples in the *best* possible way
  - Structural risk minimization principle



The induction of support vector classifier

SVM

- Advantages
  - Top performing classifier
  - Applicable for not linearly separable cases
  - No feature selection is needed. SVM does not suffer from overfitting
  - *Default choice of parameter settings* : Not need human and machine effort in parameter tuning.
  - The best decision surface is determined by only a small set of training examples, *support vectors*

Evaluation Issues for TC

- The contingency table for c_i

Category c_i		expert judgments YES	expert judgments NO
classifier judgments	**YES**	TP_i	FP_i
	NO	FN_i	TN_i

  – Precision of c_i (Pr_i) : the *degree of soundness* of the classifier

$$\widehat{\Pr}_i = \frac{TP_i}{TP_i + FP_i}$$

  – Recall of c_i (Re_i) : the *degree of completeness* of the classifier

$$\widehat{\mathrm{Re}}_i = \frac{TP_i}{TP_i + FN_i}$$

Other Measures of TC Effectiveness

- Other measures alternative to *Pr* and *Re*
  - *Accuracy*

$$\boxed{\hat{A}c = \frac{TP + TN}{TP + TN + FP + FN}}$$

  - *Error*

$$\boxed{\hat{E}c = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \hat{A}c}$$

  - Two reasons for not widely being used in TC
    - The typically large value of the denominator makes them much more insensitive to a variation in the number of correct decisions (*TP*+*TN*) than *Pr* and *Re*.
    - *Trivial rejector* tends to outperform all non-trivial classifiers

Combined Effectiveness Measures

- The inverse proportion relation between *Pr* and *Re*
  - In order to obtain 100% Re, one only needs to set every threshold τ_i to 0
  - Tune thresholds τ_i
    - *More liberal :* high *Re* to the detriment of *Pr*
    - *More conservative :* high *Pr* to the detriment of *Re*

- Various combined Measures
  - (interpolated) 11-point average precision
    - Each τ_i is set to the values for which *Re* takes up values of 0.0, 0.1, …, 0.9, 1.0
    - *Prs* are computed for the 11 resulting values and averaged

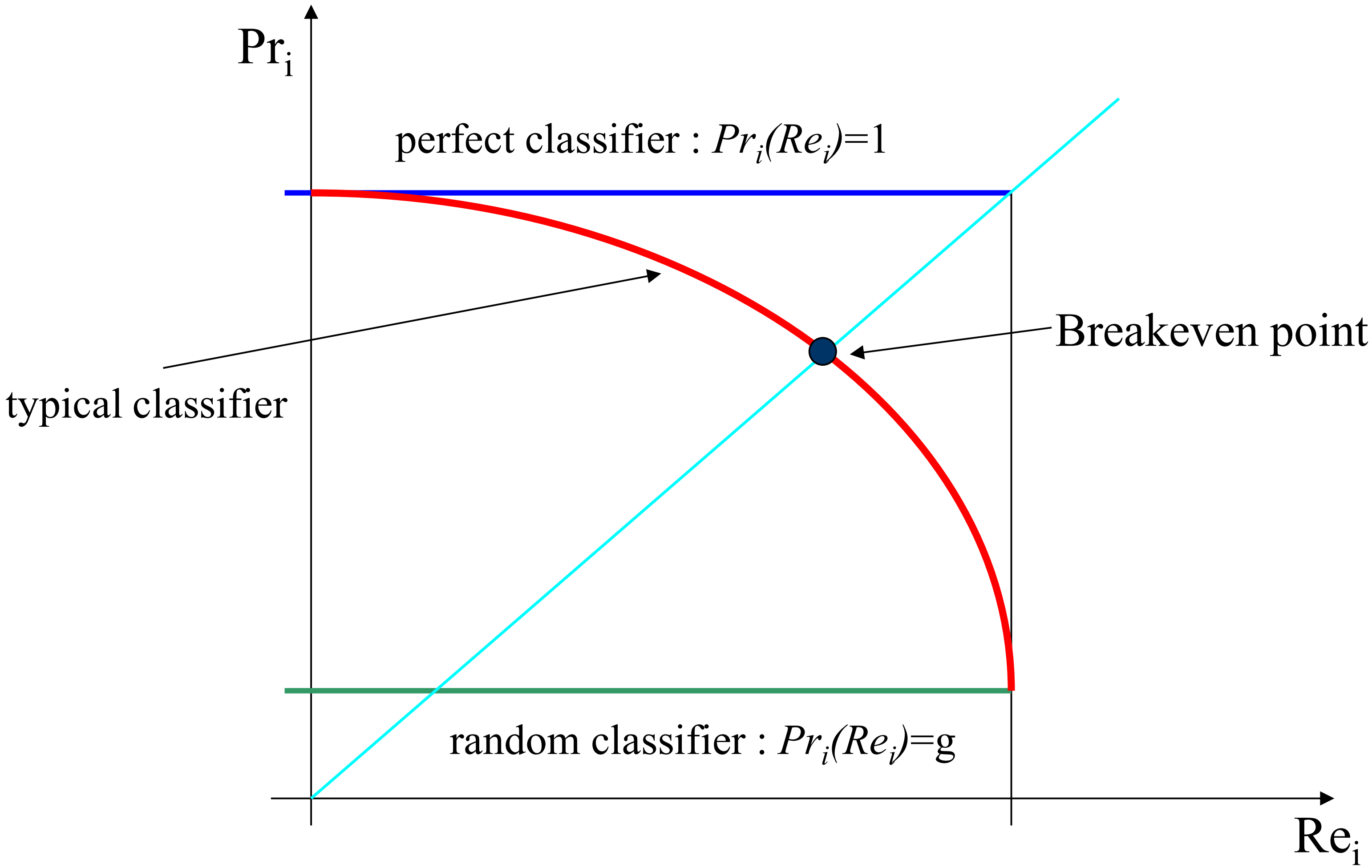
Combined Effectiveness Measures

- F_1 function

$$\boxed{F_1 = \frac{2 \cdot \Pr \cdot \mathrm{Re}}{\Pr + \mathrm{Re}}}$$

- Breakeven point
  - The value at which at which *Pr* equals *Re*.
  - Breakeven is always less or equal than F_1 (Yang, 1999)

Combined Effectiveness Measures



Test Collections

- Standard Initial Corpora
  - The REUTERS-21578 corpus, consisting of a revised version of an older corpus known as REUTERS-22173.
    - Newspaper articles
  - The OHSUMED corpus
    - Medical journal
  - The Newsgroup corpus
    - Newsgroup postings
  - The WebKB corpus
    - Web pages from 4 Universities
  - A Linguistic Link Database Web Site
    - http://www.phil.uni-passau.de/linguistik/linguistik_urls/urls.php?CAT=computing:Language+Resources:Machine+Learning+Data+Sets

What is the best learner?

- Experiments should be performed under the following conditions
  - The same collection
    - Same documents and same categories
  - The same choice of *Te* and *Tr*
  - The same effectiveness measure and the same parameter choice
  - All internal parameters have not been tuned on *Te*

What is the best learner?

- Some tentative indications can be obtained:

  - Top performers
    - *Classifiers* : SVM, *k*-NN

  - Average performers
    - *Classifiers* : Neural Networks, Decision Trees, linear classifiers, DNF decision rules

  - Under performers
    - *Classifiers* : Rocchio, Naïve Bayes