Text Categorization Using Unlabeled Data and its Theory

Ko Youngjoong (yjko@dau.ac.kr)
Dept. of Computer Engineering
Dong-A University
http://web.dong.ac.kr/yjko/

The Theory of Text Categorization

Ko Youngjoong (yjko@dau.ac.kr)
Dept. of Computer Engineering
Dong-A University

Warming Up!!

- Pattern classification (Duda & Hart)

```
Fig1. The process of the pattern classification system
```

```
Fig2. The design cycle of the pattern classification system
```

Contents

- A definition of the text categorization (TC) task
- The machine learning approach to text categorization
- Indexing and dimensionality reduction
- Methods for constructing classifiers
- Evaluation issues for text categorization
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.

\[
\begin{array}{cccc}
C_1 & \ldots & C_m & \ldots & D_1 & \ldots & D_n \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_1 & \ldots & C_m & \ldots & D_1 & \ldots & D_n \\
\end{array}
\]

- 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 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

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

• Category and document-pivoted categorization
  - CPC (category-pivoted categorization): one row at a time
    - 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
  - DPC (document-pivoted categorization): one column at a time
    - A user submits one document at a time for categorization
    - The categories may be ranked in decreasing order of estimated appropriateness for the document

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

Training Set and Test Set

• A correct decision matrix

\[
\begin{array}{cccc}
\text{Training Set} & \ldots & \text{Test Set} \\
\hline
C_1 & b_{11} & \ldots & b_{1g} & \ldots & b_{21} & \ldots & b_{2g} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
C_m & b_{m1} & \ldots & b_{mg} & \ldots & b_{(m+1)1} & \ldots & b_{(m+1)g} \\
\end{array}
\]

- A positive example of \( c_i \) if \( b_{ij} = 1 \)
- A negative example of \( c_i \) if \( b_{ij} = 0 \)

• A validation set
  - Use for optimizing its internal parameters
  - A training set may be split into a true training set and a validation set
Indexing and Dimensionality Reduction

- The choice of a text representation
  - Lexical semantics
  - Compositional semantics

- The bag of words approach
  - The vector of a document: \( n \) weighted terms (or features) \( t_j \) that occur in \( d_j \).
  - Weight \( w_{kj} \)
    - [0,1]: the most frequent case
    - \{0,1\}: presence or absence of \( t_j \) in \( d_j \)

- Lewis have found that more sophisticated representations (linguistic phrases, statistical phrases, etc) yield worse effectiveness.

TFIDF Term Weighting Scheme

- TFIDF term weight
  \[
  tfidf(t_i, d_j) = \frac{\#(t_i, d_j)}{\#(t_i)} \log \frac{|D|}{|T(d_j)|}
  \]

- Cosine Normalization
  - The weights resulting from \( tfidf \) so as to account for document length
  \[
  w_{ij} = \frac{tfidf(t_i, d_j)}{\sqrt{\sum_{k=1}^{r}(tfidf(t_i, d_k))^2}}
  \]

The Indexing Process

From: xxx@sciences.sdsu.edu
Newsgroups: comp.graphics
Subjects: Need specs Apple QT

I need to get the specs, or at least a very verbose interpretation of the specs. For QuickTime, Technical articles from magazines and references to books would be nice, too. I also need the specs in a format usable on a Unix or MS-DOS system. I can't do much with the QuickTime stuff they have on ...

Dimensionality Reduction (DR)

- Why?
  - Sophisticated learning algorithms for TC do not scale well to high values of \( r \)
  - DR reduces overfitting

- Two ways of viewing DR
  - Local DR: for one category
  - Global DR: for all categories

- The second distinction
  - DR by feature selection: the chosen \( r' \) terms are a subset of the original terms \( r \)
  - DR by feature extraction: the \( r' \) terms are not a subset of the original \( r \) terms. They are usually obtained by combinations or transformations of the original ones.
### Feature Selection Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Denoted by</th>
<th>Mathematical Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document frequency</td>
<td>#(g_1, c_i)</td>
<td>(P(d_i</td>
</tr>
<tr>
<td>Mutual information</td>
<td>MI(g_1, c_i)</td>
<td>(\log \frac{P(d_i</td>
</tr>
<tr>
<td>Information Gain</td>
<td>IG(g_1, c_i)</td>
<td>(P(d_i</td>
</tr>
<tr>
<td>Chi-square</td>
<td>(\chi^2)(g_1, c_i)</td>
<td>(\sum \left[ \frac{P(d_i, c_i) P(d_i</td>
</tr>
<tr>
<td>Correlation coefficient</td>
<td>CC(g_1, c_i)</td>
<td>(\sum \frac{P(d_i, c_i) - P(d_i) P(c_i</td>
</tr>
<tr>
<td>Odds Ratio</td>
<td>OR(g_1, c_i)</td>
<td>(\frac{P(d_i</td>
</tr>
</tbody>
</table>

### Global DR & Feature Extraction

- The forms for global DR
  - Sum: \(f_{\text{sum}}(t_i) = \sum P(c_i | d_i)\)
  - Weighted average: \(f_{\text{wavg}}(t_i) = \sum P(c_i | d_i) P(d_i)\)
  - Maximum: \(f_{\text{max}}(t_i) = \max P(c_i | d_i)\)

- The best among such measures
  - \{OR, CC\} \(\geq\) \{\(\chi^2\), IG\} \(\geq\) \{\#, MI\}

- Two approaches of feature extraction
  - Term clustering
  - Latent semantic indexing: singular value decomposition

### Probabilistic Classifiers

- The *Categorization Status Value (CSV)* function
  - \(CSV_i: D \rightarrow [0,1]\) (given \(d_j\), for category \(c_i\))
  - The definition of a threshold \(\tau_j\)
    - \(CSV(d_j) \geq \tau_j\): a decision to categorize \(d_j\) under \(c_i\)

- Naive Bayes classifiers (McCallum & Nigam, 1998)
  - View \(CSV(d_j)\) in terms of Bayes’ theorem
    
    \[
    P(c_i | d_j) = \frac{P(c_i) P(d_j | c_i)}{P(d_j)}
    \]
  - Use of the independence assumption for \(P(d_j | c_i)\)
    
    \[
    P(d_j | c_i) = \prod_{t=1}^{D} P(t_j | c_i)
    \]

### 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
Decision Tree Classifiers

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

![Decision Tree Diagram]

The Rocchio Classifier

- An adaptation to TC of Rocchio’s formula for relevance feedback
  - To compute a profile for \( c_i \) by means of the formula
    \[
    w_{b_i} = \frac{\beta}{|\mathcal{S}_{r_1}(d_j)|} \sum_{d_j \in \mathcal{S}_{r_1}(d_j)} w_k - \frac{\gamma}{|\mathcal{S}_{r_0}(d_j)|} \sum_{d_j \in \mathcal{S}_{r_0}(d_j)} w_k
    \]
  - Typical choices for the control parameters \( \beta \) and \( \gamma \)
    - \( \beta = 16 \) and \( \gamma = 4 \), \( \beta = 1 \) and \( \gamma = 0 \)
  - Advantages
    - the learner is easy to implement
    - Quite efficient
    - Easily interpretable
  - Drawbacks
    - Seldom very effective, categories are not linearly separable

Example-based Classifiers

- The distance weighted \( k \)-NN (Yang, 94)
  \[
  dSV(d_j) = \sum_{k \in \mathcal{T}_{\text{R}(d_j)}} \text{RSV}(d_j, \mathcal{T}) \cdot h_k
  \]
  - \( \text{RSV}(d_j, \mathcal{T}) \): a measure or semantic relatedness between \( d_j \) and \( \mathcal{T} \)
  - Ex) vector-based measures: inner-product, cosine similarity
  - The \( h_k \) values are from the correct decision matrix of \{0,1\}
  - \( \mathcal{T}_{\text{R}(d_j)} \) is the set of the \( k \) documents \( \mathcal{T} \) for which \( \text{RSV}(d_j, \mathcal{T}) \) is maximum: the \( k \) value should be determined on a validation set
  - Advantages
    - High performance, Not suffer from the “linear separation problem”
  - Drawbacks
    - Too late running time, lazy learners.

SVM

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

![SVM Induction Diagram]
Boosting

- The Boosting Method for the Classifier Committees
  - By the same learning method (weak learner)
  - Trained sequentially, one after the other.
    - The training of classifier \( F_i \) may take into account how classifiers \( F_1, \ldots, F_{i-1} \) perform on the training examples, and concentrate on getting right those examples in which \( F_1, \ldots, F_{i-1} \) have performed worst
  - The ADABOOST algorithm (Schapire & Singer, 2000)
    - Weak learner: decision tree
    - Each pair is attributed an importance weight \( h_{ij} \)
    - \( F_t \) is then applied to the training documents, and as a result weights \( h_{ij} \) are updated to yield \( h_{i+1} \)
      - Pairs correctly classified by \( F_i \) will have their weight decreased
      - Pairs misclassified by \( F_i \) will have their weight increased

Evaluation Issues for TC

- The contingency table for \( c_i \)
  - Precision of \( c_i \) (\( Pr_i \)): the degree of soundness of the classifier
  - Recall of \( c_i \) (\( Re_i \)): the degree of completeness of the classifier

\[
\begin{array}{c|ccc}
\text{Category} & \text{expert judgments} & \text{classifier judgments} \\
\hline
& \text{YES} & \text{NO} & \text{YES} & \text{NO} \\
\hline
\text{TP} & \text{FP} & \text{FN} & \text{FP} & \text{FN} \\
\end{array}
\]

- Precision of \( c_i \) (\( Pr_i \))
- Recall of \( c_i \) (\( Re_i \))

Combined Effectiveness Measures

- The inverse proportion relation between \( Pr \) and \( Re \)
  - To obtain 100% \( Re \), one only needs to set every threshold \( z_i \) to 0
- Various combined measures
  - (interpolated) 11-point average precision
    - Each \( z_i \) is set to the values for which \( Re \) takes up values of 0.0, 0.1, ..., 0.9, 1.0
    - \( Pr \) is computed for the 11 resulting values and averaged
  - \( F_b \) function
    - For some \( 0 \leq b \leq +\infty \)
      \[
      F_b = \frac{(F^2 + b) \cdot Pr \cdot Re}{F^2 \cdot Pr + Re}
      \]
    - When \( b = 1 \), \( F_b \) has equal importance of \( Pr \) and \( Re \), called by \( F_1 \) measure
    - Breakeven point
      - The value at which \( Pr \) equals \( Re \).
      - Breakeven is always less or equal than \( F_1 \) (Yang, 1999)

\[
\begin{array}{c|c|c}
\text{Pr}_i & \text{perfect classifier: \( Pr_i(Re_i)=1 \)} & \text{random classifier: \( Pr_i(Re_i)=g \)} \\
\text{Re}_i & \text{Breakeven point} \\
\end{array}
\]
Research Issues on Text Categorization

- The state-of-the-art classification systems
- *Unsupervised manner Text Categorization*
- Hypertext classification problems
- Hierarchical classification problems
- Etc.
  - Filtering (Ex. Email)
  - TDT (Topic Detection Tracking)

## Contents

- Introduction
- Learning with Unlabeled Data Using a Title Word of Each Category
- TCFP Classifiers for Learning with Machine-labeled Data
- Conclusions and Future Works

## Introduction (1)

- Text Categorization (TC)
  - Classify documents into one (or several) of a set of pre-defined categories (topics of interest)
  - Prominent status in the information system field
    - Explosion of electronic texts from the WWW, E-mail, Digital library etc
  - Until the late ’80s
    - Manual construction of rule sets
    - High accuracy but significant cost
  - In the ’90s, the machine learning paradigm
    - *Supervised learning*
      - Find decision rule from an example set of labeled documents for each category
    - High accuracy and less expensive

Text Categorization Using Unlabeled Data

Ko Youngjoong (yjko@dau.ac.kr)
Dept. of Computer Engineering
Dong-A University
Introduction (2)

- **Difficulties of supervised learning in TC**
  - Require large, often prohibitive, number of labeled training data
  - Various application areas: article, web pages, e-mail, and newsgroup, digital library, CRM, biomedical text etc

- **Our proposal**
  - Automatically constructs labeled training data from unlabeled documents and the title word of each category
    - How can we automatically generate labeled training documents (machine-labeled data) from only title words
      - Bootstrapping Framework
    - How can we handle incorrectly labeled documents in the machine-labeled data.
      - TCFP Classifier

Overview

- **Pre-defined category?**
  - If so, I can know title words!
- **Bootstrapping!**
  - Robust classifier from noisy data!

Constructing Context-Clusters for Training(1)

- **Context**
  - A unit of meaning in our method for bootstrapping
  - Part of a text that surrounds the particular word or a passage
  - Define a context as 60 words

- **Creating Keyword list**
  - Creating keyword list of each category
    - Keyword: words to be semantically related to a title word
    - Co-occurrence information
    - Cosine similarity

\[
\cos(T, X) = \frac{\sum_{i=1}^{n} t_i x_i}{\sqrt{n} \sqrt{\sum_{i=1}^{n} t_i^2 \sum_{i=1}^{n} x_i^2}}
\]

Constructing Context-Clusters for Training(2)

- **Extracting and verifying centroid-contexts**
  - Centroid-context
    - Contain a keyword or a title word of a category
  - Importance score of centroid-context
    - TF-ICF
      - \( w_i = TF_i \times ICF = TF_i \times (\log(M) - \log(CF_i)) \)
    - Importance Score
      - \( \text{Score}(CC, cr) = \frac{\sum w_i}{n} \)
Constructing Context-Clusters for Training(3)

- Creating Contexts-Clusters
  - The goal
    - Assign remaining contexts to each category
  - The Assigning algorithm
    - Measuring similarity based on word & context similarity
      - By Karov & Edelman, 1998
    - Improve this algorithm for our method

Naive Bayes Classifier

- Naive Bayes with minor modification
  - Kullback-Leibler Divergence
  - Produce classification scores with less extreme
    \[
    P(c_1 | d, \hat{\theta}) = \frac{P(c_1 | \hat{\theta}) P(d | c_1, \hat{\theta})}{P(d | \hat{\theta})} = \prod_{c_i} P(c_i | \hat{\theta})^{n_{c_i}(d)} \times \log P(c_1 | \hat{\theta}) + \sum_{c_i} P(c_i | d, \hat{\theta}) \log \frac{P(c_i | d, \hat{\theta})}{P(c_i | d, \hat{\theta})} \]
  - Laplace parameter estimation
    \[
    \hat{\theta}_{c_i} = \frac{1 + N(c_i, G_i)}{|F| + \sum_{c_i} N(c_i, G_i)} \quad \hat{\theta}_c = \frac{1 + N(c, G_c)}{|C| + \sum_{c_i} N(c_i, G_i)}
    \]

Constructing Context-Clusters for Training(4)

- Affinity Formula
  \[
  \text{aff}(W, C) = \max_{W'} \text{sim}(W, W')
  \]
  \[
  \text{aff}(C, W) = \max_{C'} \text{sim}(C, C')
  \]
- Similarity Formulae
  \[
  \text{sim}_{c_i}(C, C_i) = \sum_{W} \text{weight}(W, C_i) \cdot \text{aff}(W, C_i)
  \]
  \[
  \text{if} \quad W' = W_i
  \]
  \[
  \text{sim}_{c_i}(W, W_i) = 1
  \]
  \[
  \text{else}
  \]
  \[
  \text{sim}_{c_i}(W, W_i) = \sum_{W} \text{weight}(W, W_i) \cdot \text{aff}(W, W_i)
  \]

Empirical Results (1)

- Data sets
  - 3 different types : UseNet newsgroups, web pages, newswire articles
    - Newsgroups data set
    - WebKB data set
    - Reuters-21578 Test Collection
- Experimental setting
  - Five-Fold validation
  - Feature Selection : $\chi^2$ statistics
  - Performance Measure
    - Micro-average F1 measure : Newsgroups, WebKB
    - Precision-recall Brekevens Point : Reuters
Empirical Results (2)

• Comparing with supervised NB classifier

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Our method</th>
<th>Supervised NB</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsgroups</td>
<td>79.36</td>
<td>91.72</td>
<td>-12.36</td>
</tr>
<tr>
<td>WebKB</td>
<td>73.63</td>
<td>85.29</td>
<td>-11.66</td>
</tr>
<tr>
<td>Reuters</td>
<td>88.62</td>
<td>91.64</td>
<td>-3.02</td>
</tr>
</tbody>
</table>

Learning with Machine-labeled Data

• Learning with Machine-labeled Data
  – Obtain finally labeled data of a document unit
  – We can learn supervised classifiers using them
  – A problem
    • Machine-labeled data has a lot of incorrectly labeled documents
    • Need a new robust classifier from noisy data

• TCFP classifier
  – A new type of text classifier using the feature projection technique
    • With robustness from noisy data
    • Fast execution speed
    • High performance
    • Simple algorithm: easily implement and quickly learn

A New Approach on Feature Projections

• An example of feature projections in Text Categorization

```
begin
  for each category \( c_j \)
    vote[\( c_j \)] = 0
  for each feature \( t_i \)
    \( w(t_i) \) is calculated
  for each category \( c_j \)
    for each feature \( t_i \)
      vote[\( c_j \)] += \( w(t_i) \times vs(c_j, t_i) \)
  prediction = \( \text{argmax} \) vote[\( c_j \)]
return prediction
end
```

A New Approach on Feature Projections

• A New Text Categorization Algorithm: TCFP
Empirical Evaluation

(1)

- Comparison of TCFP with conventional text classifiers

<table>
<thead>
<tr>
<th>Data Set</th>
<th>TCFP</th>
<th>k-NN</th>
<th>SVM</th>
<th>NB</th>
<th>Rocchio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsgroups</td>
<td>86.57</td>
<td>85.92</td>
<td>87.97</td>
<td>82.79</td>
<td>82.37</td>
</tr>
<tr>
<td>WebKB</td>
<td>88.07</td>
<td>84.82</td>
<td>91.75</td>
<td>85.29</td>
<td>86.05</td>
</tr>
<tr>
<td>Reuters</td>
<td>90.03</td>
<td>88.95</td>
<td>91.42</td>
<td>88.62</td>
<td>86.47</td>
</tr>
</tbody>
</table>

- Running Time Observation

  - TCFP is about one hundred times faster classifier than k-NN

<table>
<thead>
<tr>
<th>Data Set</th>
<th>TCFP without context</th>
<th>k-NNFP</th>
<th>Rocchio</th>
<th>TCFP</th>
<th>NB</th>
<th>SVM</th>
<th>k-NN with pruning</th>
<th>k-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsgroups</td>
<td>0.68</td>
<td>0.85</td>
<td>0.8</td>
<td>1.25</td>
<td>1.22</td>
<td>14.71</td>
<td>37.97</td>
<td>142.54</td>
</tr>
<tr>
<td>WebKB</td>
<td>0.11</td>
<td>0.14</td>
<td>0.55</td>
<td>0.17</td>
<td>2.72</td>
<td>4.91</td>
<td>15.21</td>
<td></td>
</tr>
<tr>
<td>Reuters</td>
<td>2.45</td>
<td>2.7</td>
<td>3.34</td>
<td>2.89</td>
<td>7.01</td>
<td>39.94</td>
<td>15.88</td>
<td>65.86</td>
</tr>
</tbody>
</table>

Empirical Evaluation

(2)

- Robustness from Noisy Data

  - 4 data sets with from 10% to 40% noisy data in Newsgroups

Empirical Evaluation

(3)

- Results using machine-labeled documents data

<table>
<thead>
<tr>
<th>Data Set</th>
<th>OurMethod (basis)</th>
<th>OurMethod (NB)</th>
<th>OurMethod (Rocchio)</th>
<th>OurMethod (k-NN)</th>
<th>OurMethod (SVM)</th>
<th>Supervised NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsgroups</td>
<td>78.36</td>
<td>83.46</td>
<td>83</td>
<td>79.95</td>
<td>82.49</td>
<td>86.19</td>
</tr>
<tr>
<td>WebKB</td>
<td>73.63</td>
<td>73.22</td>
<td>75.28</td>
<td>68.04</td>
<td>73.74</td>
<td>75.47</td>
</tr>
<tr>
<td>Reuters</td>
<td>88.62</td>
<td>88.23</td>
<td>86.26</td>
<td>85.65</td>
<td>87.41</td>
<td>89.99</td>
</tr>
</tbody>
</table>

Empirical Evaluation

(4)

- Final Results

<table>
<thead>
<tr>
<th>Data Set</th>
<th>OurMethod (basis)</th>
<th>OurMethod (TCFP)</th>
<th>basis vs. TCFP</th>
<th>Supervised NB</th>
<th>TCFP vs. Supervised NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsgroups</td>
<td>79.36</td>
<td>86.19</td>
<td>+6.83</td>
<td>91.72</td>
<td>-5.53</td>
</tr>
<tr>
<td>WebKB</td>
<td>73.63</td>
<td>75.49</td>
<td>+1.84</td>
<td>85.29</td>
<td>-9.82</td>
</tr>
<tr>
<td>Reuters</td>
<td>88.62</td>
<td>89.09</td>
<td>+0.47</td>
<td>91.64</td>
<td>-2.55</td>
</tr>
</tbody>
</table>
Conclusions

- We propose a new method for learning with only title words and unlabeled documents
- Contributions
  - A text classifier can be built from unlabeled data
  - TCFP classifier with robustness from noisy data, fast execution speed, and high performance
  - Our method is superior to clustering methods
- Application Area
  - Required low-cost text categorization without labeling task
  - Creating training data
- Future Works
  - Improve the bootstrapping method from title words
  - Need more studies for voting ratio of the TCFP classifier

Publications


Lewis, D.D., 1992, Representation and learning in information retrieval, Ph.D. thesis. Dept. of Computer Science, University of Massachusetts, Amherst, US.