CS630 Representing and Accessing Digital Information

Text Classification: KNN and Rocchio
Thorsten Joachims
Cornell University

Test Collections

- Reuters-21578
  - Reuters newswire articles classified by topic
  - 90 categories (multi-label)
  - 9603 training documents / 3299 test documents (ModApte)
  - ~27,000 features

- WebKB Collection
  - WWW pages classified by function (e.g., personal HP, project HP)
  - 4 categories (multi-class)
  - 4183 training documents / 226 test documents
  - ~38,000 features

- Ohsumed MeSH
  - Medical abstracts classified by subject heading
  - 20 categories from “disease” subtree (multi-label)
  - 10,000 training documents / 10,000 test documents
  - ~38,000 features

Example: Reuters Article (Multi-Label)

Categories: COFFEE, CRUDE
KENYAN ECONOMY FACES PROBLEMS, PRESIDENT SAYS
The Kenyan economy is heading for difficult times after a boom last year, and the country must tighten its belt to prevent the balance of payments swinging too far into deficit, President Daniel Arap Moi said.

In a speech at the state opening of parliament, Moi said high coffee prices and cheap oil in 1986 led to economic growth of five pct, compared with 4.3 pct in 1985. The same factors produced a two billion shilling balance of payments surplus and inflation fell to 5.6 pct from 10.7 pct in 1985, he added.

“But both these factors are no longer in our favour ... As a result, we cannot expect an increase in foreign exchange reserves during the year,” he said.

Example: Ohsumed Abstract

Categories: Animal, Blood_Proteins/Metabolism, DNA/Drug_Effects, Mycotoxins/Toxicity, …
How aspartame prevents the toxicity of ochratoxin A.

Multi-Class / Multi-Label

- Cannot learn multi-label rules directly
  - Most classifiers assume that each document is in exactly one class
  - Many classifiers can only learn binary classification rules

- Most common solution: Multi-Label
  - Learn one binary classifier for each label
  - Attach all labels, for which some classifier says positive

- Most common solution: Multi-Class
  - Learn one binary classifier for each label
  - Put example into the class with the highest probability (or some approximation thereof)

Performance Measures

- Precision/Recall Break-Even Point
  - Intersection of PR-curve with the identity line

- Macro-averaging
  - First compute the measure, then compute average
  - Results in average over tasks

- Micro-averaging
  - First average the elements of the contingency table, then compute the measure
  - Results in average over each individual classification decision
Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Reuters Newswire</th>
<th>WebKB Collection</th>
<th>Ohsumed MeSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 cate gories</td>
<td>4 cate gories</td>
<td>20 cate gories</td>
<td></td>
</tr>
<tr>
<td>9603 training doc.</td>
<td>4183 training doc.</td>
<td>10000 training doc.</td>
<td></td>
</tr>
<tr>
<td>3299 test doc.</td>
<td>226 test doc.</td>
<td>10000 test doc.</td>
<td></td>
</tr>
<tr>
<td>~27000 features</td>
<td>~38000 features</td>
<td>~38000 features</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>microaveraged precision/recall breakeven-point [0..100]</th>
<th>Reuters</th>
<th>WebKB</th>
<th>Ohsumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>72.3</td>
<td>82.0</td>
<td>62.4</td>
</tr>
<tr>
<td>Rocchio Algorithm</td>
<td>79.9</td>
<td>74.1</td>
<td>61.5</td>
</tr>
<tr>
<td>C4.5 Decision Tree</td>
<td>79.4</td>
<td>79.1</td>
<td>56.7</td>
</tr>
<tr>
<td>k-Nearest Neighbors</td>
<td>82.6</td>
<td>80.5</td>
<td>63.4</td>
</tr>
<tr>
<td>SVM</td>
<td>87.5</td>
<td>90.3</td>
<td>71.6</td>
</tr>
</tbody>
</table>

Rocchio Algorithm (Learning)

- Given: (x_1, y_1), ..., (x_n, y_n) ~ P(X, Y)
- Preprocessing:
  - Bring into vector space model representation (e.g. TFIDF)
  - Vectors normalized to Euclidian length 1
  - Split into set of positive / negative examples (i.e. D_+, D_-)
- Training:
  - Build prototype vector for each class
  - Compute weight vector as weighted difference between prototypes
  \[ w' = \frac{1}{|D_+|} \sum_{x \in D_+} x - \frac{1}{|D_-|} \sum_{x \in D_-} x \]
  - Often: set negative elements of w vector to zero

Rocchio Algorithm (Prediction)

- Compute cosine between weight vector w and new example x'
- Prediction rule
  \[ \hat{y}(x') = \begin{cases} 1 & \text{if } \cos(w, x') > 0 \\ -1 & \text{else} \end{cases} \]
- Threshold is a parameter, or often the cosine is just used to get a ranking

Representing Text as Attribute Vectors

K-Nearest Neighbor

- Given: (x_1, y_1), ..., (x_n, y_n) ~ P(X, Y)
- Preprocessing:
  - Bring into vector space model representation (e.g. TFIDF)
- Learning:
  - None
- Prediction rule
  \[ \hat{y}(x') = \text{sign} \left( \sum_{x \in \text{nbrs}(x')} y \cos(x, x') \right) \]

Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Reuters Newswire</th>
<th>WebKB Collection</th>
<th>Ohsumed MeSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 cate gories</td>
<td>4 cate gories</td>
<td>20 cate gories</td>
<td></td>
</tr>
<tr>
<td>9603 training doc.</td>
<td>4183 training doc.</td>
<td>10000 training doc.</td>
<td></td>
</tr>
<tr>
<td>3299 test doc.</td>
<td>226 test doc.</td>
<td>10000 test doc.</td>
<td></td>
</tr>
<tr>
<td>~27000 features</td>
<td>~38000 features</td>
<td>~38000 features</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>microaveraged precision/recall breakeven-point [0..100]</th>
<th>Reuters</th>
<th>WebKB</th>
<th>Ohsumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>72.3</td>
<td>82.0</td>
<td>62.4</td>
</tr>
<tr>
<td>Rocchio Algorithm</td>
<td>79.9</td>
<td>74.1</td>
<td>61.5</td>
</tr>
<tr>
<td>C4.5 Decision Tree</td>
<td>79.4</td>
<td>79.1</td>
<td>56.7</td>
</tr>
<tr>
<td>k-Nearest Neighbors</td>
<td>82.6</td>
<td>80.5</td>
<td>63.4</td>
</tr>
<tr>
<td>SVM</td>
<td>87.5</td>
<td>90.3</td>
<td>71.6</td>
</tr>
</tbody>
</table>
Feature (Subset) Selection

- Some classifiers perform worse when using all features
  - E.g. K-NN, Rocchio, C4.5, sometimes Naïve Bayes
- Some classifiers are too inefficient to use all features
  - E.g. C4.5
- Methods
  - Document Frequency Thresholding: Use only those words that occur at least \( m \) times in the training documents
  - Empirical Mutual Information: Pick words \( w \) with largest

\[
I(x, p) = \sum_{x \in (1, L-1)} \sum_{w \in (L, \infty)} P(x,w) \log \left( \frac{P(x,w)}{P(p)P(w)} \right)
\]

- Also odds-ratio, chi-square score, stopword-removal, stemming

Comparison of Methods

<table>
<thead>
<tr>
<th>Naive Bayes</th>
<th>Rocchio</th>
<th>C4.5</th>
<th>K-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplicity (conceptual)</td>
<td>+</td>
<td>++</td>
<td>-</td>
<td>++</td>
</tr>
<tr>
<td>Efficiency at training</td>
<td>+</td>
<td>+</td>
<td>--</td>
<td>++</td>
</tr>
<tr>
<td>Efficiency at prediction</td>
<td>++</td>
<td>++</td>
<td>-</td>
<td>--</td>
</tr>
<tr>
<td>Handling many classes</td>
<td>+</td>
<td>+</td>
<td>--</td>
<td>++</td>
</tr>
<tr>
<td>Theoretical understanding</td>
<td>o</td>
<td>--</td>
<td>-</td>
<td>o</td>
</tr>
<tr>
<td>Prediction accuracy</td>
<td>-</td>
<td>o</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Stability and robustness</td>
<td>-</td>
<td>-</td>
<td>--</td>
<td>o</td>
</tr>
</tbody>
</table>