Computational Methods for Data Analysis

Vector Space Categorization;
(On Text)

Alessandro Moschitti

Department of Computer Science and Information Engineering
University of Trento
Email: moschitti@disi.unitn.it
Outline

Text Categorization and Optimization
- TC Introduction
- TC designing steps
- Rocchio text classifier
- Support Vector Machines
- The Parameterized Rocchio Classifier (PRC)
- Evaluation of PRC against Rocchio and SVM
Introduction to Text Categorization

Berlusconi acquires Inzaghi before elections

Politic $C_1$

Economic $C_2$

Sport $C_n$
Text Classification Problem

- Given:
  - a set of target categories: \( C = \{ C^1, \ldots, C^n \} \)
  - the set \( T \) of documents,

define

\[
f : T \rightarrow 2^C
\]

- VSM (Salton89’)
  - Features are dimensions of a Vector Space.
  - Documents and Categories are vectors of feature weights.
  - \( d \) is assigned to \( C^i \) if

\[
\vec{d} \cdot \vec{C}^i > th
\]
The Vector Space Model

$\mathbf{d}_1$: Politic

Bush declares war.
Berlusconi gives support

$\mathbf{d}_2$: Sport

Wonderful Totti in the yesterday match against Berlusconi’s Milan

$\mathbf{d}_3$: Economic

Berlusconi acquires Inzaghi before elections

$\mathbf{C}_1$: Politics Category

$\mathbf{C}_2$: Sport Category
Automated Text Categorization

- A corpus of pre-categorized documents
- Split document in two parts:
  - Training-set
  - Test-set
- Apply a supervised machine learning model to the training-set
  - Positive examples
  - Negative examples
- Measure the performances on the test-set
  - e.g., Precision and Recall
Feature Vectors

Each example is associated with a vector of \( n \) feature types (e.g. unique words in TC)

\[
\vec{x} = (0, \ldots, 1, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 1)
\]

acquisition buy market sell stocks

The dot product \( \vec{X} \cdot \vec{Z} \) counts the number of features in common

This provides a sort of similarity
Text Categorization phases

- Corpus pre-processing (e.g. tokenization, stemming)
- Feature Selection (optionally)
  - Document Frequency, Information Gain, $\chi^2$, mutual information,...
- Feature weighting
  - for documents and profiles
- Similarity measure
  - between document and profile (e.g. scalar product)
- Statistical Inference
  - threshold application
- Performance Evaluation
  - Accuracy, Precision/Recall, BEP, f-measure,...
Feature Selection

- Some words, i.e. features, may be irrelevant
- For example, “function words” as: “the”, “on”, “those”…
- Two benefits:
  - efficiency
  - Sometime the accuracy
- Sort features by relevance and select the $m$-best
Statistical Quantity to sort feature

- Based on corpus counts of the pair \( \langle \text{feature}, \text{category} \rangle \)
  
  - \( A \) is the number of documents in which both \( f \) and \( c \) occur, i.e. \( (f, c) \);
  
  - \( B \) is the number of documents in which only \( f \) occurs, i.e. \( (f, \bar{c}) \);
  
  - \( C \) is the number of documents in which only \( c \) occurs, i.e. \( (\bar{f}, c) \);
  
  - \( D \) is the number of documents in which neither \( f \) nor \( c \) occur, i.e. \( (\bar{f}, \bar{c}) \);
  
  - \( N \) is the total number of documents, i.e. \( A + B + C + D \).
Statistical Selectors

- Chi-square, Pointwise MI and MI

\[ \chi^2(f, c) = \frac{N \times (AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \]

\[ PMI(f, c) = \log \frac{P(f, c)}{P(f) \times P(c)} \]

\[ MI(f, C) = - \sum_{c \in C} P(c) \log(P(c)) + P(f) \sum_{c \in C} P(c|f) \log(P(c|f)) + P(\bar{f}) \sum_{c \in C} P(c|\bar{f}) \log(P(c|\bar{f})) \]
Chi-Square Test

\[ X^{2} = \sum_{i=1}^{n} \frac{(O_{i} - E_{i})^{2}}{E_{i}} \],

- \( O_{i} \) = an observed frequency;
- \( E_{i} \) = an expected (theoretical) frequency, asserted by the null hypothesis;
- \( n \) = the number of cells in the table.
Just an intuitions from Information Theory of MI

- $\text{MI}(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$
- If $X$ very similar to $Y$, $H(Y|X) = H(X|Y) = 0$
  $\Rightarrow \text{MI}(X,Y)$ is maximal
Probability Estimation

- $P(f, c)$ is the probability that $f$ and $c$ co-occurs and can be estimated by $A/N$;

- $P(f)$ is the probability of $f$, estimated by $(A + B)/N$;

- $P(c)$ is the probability of $c$, estimated by $(A + C)/N$;

- $P(c|f)$ is the probability of $c$ by considering only the documents that contain $f$. It can be estimated by $\frac{P(f,c)}{P(f)}$.

- $P(\bar{f})$ is the probability that $f$ does not occur, estimated by $(C + D)/N$;
- $P(c|\bar{f})$ is the probability of $c$ by considering only the documents that do not contain $f$. It can be estimated by $\frac{P(\bar{f},c)}{P(f)}$. In turn, $P(\bar{f},c)$ is estimated by $C/N$.

- $C$ is the collection of categories, i.e. $\{c_1, c_2, \ldots, c_n\}$. Note that $PMI$ and $\chi^2$ are defined on only two categories, i.e. $c$ and $not\ c$ whereas $MI$ can be evaluated on $n > 2$ categories\(^7\).

For example, we can apply the above formulas to evaluate the $PMI$ as follows:

$$PMI = \log \frac{N}{A + B} \times \frac{N}{A + C} \times \frac{A}{N} = \log \frac{A \times N}{(A + C)(A + B)}$$
Global Selectors

\[
PMI_{\text{max}}(f) = \max_{c \in \mathcal{C}} \text{ PMI}(f, c)
\]

\[
PMI_{\text{avg}}(f) = \sum_{c \in \mathcal{C}} P(c) \times \text{PMI}(f, c)
\]

\[
\chi^2_{\text{max}}(f) = \max_{c \in \mathcal{C}} \chi^2(f, c)
\]

\[
\chi^2_{\text{avg}}(f) = \sum_{c \in \mathcal{C}} P(c) \times \chi^2(f, c)
\]
Document weighting: an example

- N, the overall number of documents,
- \( N_f \), the number of documents that contain the feature \( f \)
- \( o_f^d \), the occurrences of the features \( f \) in the document \( d \)

The weight \( f \) in a document is:

\[
\omega_f^d = \left( \log \frac{N}{N_f} \right) \times o_f^d = IDF(f) \times o_f^d
\]

The weight can be normalized:

\[
\omega_f^{d'} = \frac{\omega_f^d}{\sqrt{\sum_{t \in d} (\omega_t^d)^2}}
\]
Profile Weighting: the Rocchio’s formula

- $\omega_f^d$, the weight of $f$ in $d$
- Several weighting schemes (e.g. TF * IDF, Salton 91’)

- $\vec{C}_f^i$, the profile weights of $f$ in $C_i$:

$$\vec{C}_f^i = \max\left\{ 0, \frac{\beta}{|T_i|} \sum_{d \in T_i} \omega_f^d - \frac{\gamma}{|\overline{T}_i|} \sum_{d \in \overline{T}_i} \omega_f^d \right\}$$

- $T_i$, the training documents in $C_i$
Similarity estimation

Given the document and the category representation

\[ \vec{d} = \left< \omega^d_{f_1}, ..., \omega^d_{f_n} \right>, \quad \vec{C}_i = \left< \Omega^i_{f_1}, ..., \Omega^i_{f_n} \right> \]

It can be defined the following similarity function (cosine measure)

\[ s_{d,i} = \cos(\vec{d}, \vec{C}_i) = \frac{\vec{d} \cdot \vec{C}_i}{\|\vec{d}\| \times \|\vec{C}_i\|} = \frac{\sum_{f} \omega^d_f \times \Omega^i_f}{\|\vec{d}\| \times \|\vec{C}_i\|} \]

\[ d \text{ is assigned to } C^i \text{ if } \vec{d} \cdot \vec{C}^i > \sigma \]
Bidimensional view of Rocchio categorization
Rocchio problems

- Prototype models have problems with polymorphic (disjunctive) categories.
The Parameterized Rocchio Classifier (PRC)

- Which pair values for $\beta$ and $\gamma$ should we consider?
- Literature work uses a bunch of values with $\beta > \gamma$ (e.g. 16, 4)
- Interpretation of positive ($\beta$) vs. negative ($\gamma$) information

Our interpretation [Moschitti, ECIR 2003]:
- One parameter can be bound to the threshold

By rewriting $\vec{C}^i \cdot \vec{d} > \sigma$ as
Binding the $\beta$ parameter

\[
\left( \frac{\beta}{|C|} \sum_{\tilde{d} \in C} \tilde{d} - \frac{\gamma}{|\tilde{C}|} \sum_{\tilde{d} \in \tilde{C}} \tilde{d} \right) \cdot \tilde{d} \geq \sigma
\]

and dividing by $\beta$,

\[
\left( \frac{1}{|C|} \sum_{\tilde{d} \in C} \tilde{d} - \frac{\gamma}{\beta|\tilde{C}|} \sum_{\tilde{d} \in \tilde{C}} \tilde{d} \right) \cdot \tilde{d} \geq \frac{\sigma}{\beta} \Rightarrow \left( \frac{1}{|C|} \sum_{\tilde{d} \in C} \tilde{d} - \frac{\rho}{|\tilde{C}|} \sum_{\tilde{d} \in \tilde{C}} \tilde{d} \right) \cdot \tilde{d} \geq \tilde{\sigma}.
\]
Rocchio parameter interpretation

\[ \tilde{C}_f^i = \max \left\{ 0, \frac{1}{|T_i|} \sum_{d \in T_i} \vec{d}_f - \frac{\rho}{|T_i|} \sum_{d \in \overline{T_i}} \vec{d}_f \right\} \]

- 0 weighted features do not affect similarity estimation
- A \( \rho \) increase causes many feature weights to be 0
  \[ \Rightarrow \rho \text{ is a feature selector and we can find a maximal value} \]
  \[ \rho_{\text{max}} \text{ (all features are removed)} \]
- This interpretation enabled \( \gamma \gg \beta \)
Feature Selection interpretation of Rocchio parameters

- Literature work uses a bunch of values for $\beta$ and $\gamma$
- Interpretation of positive ($\beta$) vs. negative ($\gamma$) information
  \[ \Rightarrow \text{value of } \beta > \gamma \text{ (e.g. 16, 4)} \]

- **Our interpretation** [Moschitti, ECIR 2003]:
  Remove one parameters
  \[
  \tilde{C}_f^i = \max \left\{ 0, \frac{1}{|T_i|} \sum_{d \in T_i} \tilde{d}_f - \frac{\rho}{|\tilde{T}_i|} \sum_{d \in \tilde{T}_i} \tilde{d}_f \right\}
  \]
- 0 weighted features do not affect similarity estimation
- increasing $\rho$ causes many feature to be set to 0 $\Rightarrow$ they are removed
Feature Selection interpretation of Rocchio parameters (cont’d)

- By increasing $\rho$:
  - Features that have a high negative weights get firstly a zero value
  - High negative weight means very frequent in the other categories
  - $\Rightarrow$ zero weight for irrelevant features

- If $\rho$ is a feature selector, set it according to standard feature selection strategies [Yang, 97]

- Moreover, we can find a maximal value $\rho_{\text{max}}$ (associated with all feature removed)

- This interpretation enabled $\gamma \gg \beta$
Nearest-Neighbor Learning Algorithm

- Learning is just storing the representations of the training examples in $D$.
- Testing instance $x$:
  - Compute similarity between $x$ and all examples in $D$.
  - Assign $x$ the category of the most similar example in $D$.
- Does not explicitly compute a generalization or category prototypes.
- Also called:
  - Case-based
  - Memory-based
  - Lazy learning
K Nearest-Neighbor

- Using only the closest example to determine categorization is subject to errors due to:
  - A single atypical example.
  - Noise (i.e. error) in the category label of a single training example.

- More robust alternative is to find the $k$ most-similar examples and return the majority category of these $k$ examples.

- Value of $k$ is typically odd, 3 and 5 are most common.
3 Nearest Neighbor Illustration (Euclidian Distance)
K Nearest Neighbor for Text

Training:
For each training example \( <x, c(x)> \in D \)
   Compute the corresponding TF-IDF vector, \( d_x \), for document \( x \)

Test instance \( y \):
Compute TF-IDF vector \( d \) for document \( y \)
For each \( <x, c(x)> \in D \)
   Let \( s_x = \cosSim(d, d_x) \)
Sort examples, \( x \), in \( D \) by decreasing value of \( s_x \)
Let \( N \) be the first \( k \) examples in \( D \). (get most similar neighbors)
Return the majority class of examples in \( N \)
Illustration of 3 Nearest Neighbor for Text
A state-of-the-art classifier: Support Vector Machines

- The Vector $\tilde{C}^i$ satisfies:

$$\min |\tilde{C}^i|$$

$$\tilde{C}^i \times \tilde{d} - th \geq +1, \text{ if } d \in T_i$$

$$\tilde{C}^i \times \tilde{d} - th \leq -1, \text{ if } d \notin T_i$$

- $d$ is assigned to $C^i$ if $\tilde{d} \times \tilde{C}^i > th$
SVM

Decision Hyperplane

Support Vectors
Other Text Classifiers

- **RIPPER** [Cohen and Singer, 1999] uses an extended notion of a profile. It learns the contexts that are positively correlated with the target classes, i.e. words co-occurrence.

- **EXPERT** uses as context nearby words (sequence of words).

- **CLASSI** is a system that uses a neural network-based approach to text categorization [Ng *et al.*, 1997]. The basic units of the network are only perceptrons.

- **Dtree** [Quinlan, 1986] is a system based on a well-known machine learning model.

- **CHARADE** [I. Moulinier and Ganascia, 1996] and **SWAP1** [Apt´e *et al.*, 1994] use machine learning algorithms to inductively extract Disjunctive Normal Form rules from training documents.
Experiments

- Reuters Collection 21578 Apté split (Apté94)
  - 90 classes (12,902 docs)
  - A fixed splitting between training and test set
  - 9603 vs 3299 documents

- Tokens
  - about 30,000 different

- Other different versions have been used but …
  most of TC results relate to the 21578 Apté
  - [Joachims 1998], [Lam and Ho 1998], [Dumais et al. 1998],
  - [Li Yamanishi 1999], [Weiss et al. 1999],
  - [Cohen and Singer 1999]…
CRA SOLD FORREST GOLD FOR 76 MLN DLRS - WHIM CREEK

SYDNEY, April 8 - <Whim Creek Consolidated NL> said the consortium it is leading will pay 76.55 mln dlrs for the acquisition of CRA Ltd's <CRAA.S> <Forrest Gold Pty Ltd> unit, reported yesterday.

CRA and Whim Creek did not disclose the price yesterday. Whim Creek will hold 44 pct of the consortium, while <Austwhim Resources NL> will hold 27 pct and <Croesus Mining NL> 29 pct, it said in a statement.

As reported, Forrest Gold owns two mines in Western Australia producing a combined 37,000 ounces of gold a year. It also owns an undeveloped gold project.
FTC URGES VETO OF GEORGIA GASOLINE STATION BILL

WASHINGTON, March 20 - The Federal Trade Commission said its staff has urged the governor of Georgia to veto a bill that would prohibit petroleum refiners from owning and operating retail gasoline stations.

The proposed legislation is aimed at preventing large oil refiners and marketers from using predatory or monopolistic practices against franchised dealers.

But the FTC said fears of refiner-owned stations as part of a scheme of predatory or monopolistic practices are unfounded. It called the bill anticompetitive and warned that it would force higher gasoline prices for Georgia motorists.
Performance Measurements

- Given a set of document $T$
- Precision = $\frac{\# \text{Correct Retrieved Document}}{\# \text{Retrieved Documents}}$
- Recall = $\frac{\# \text{Correct Retrieved Document}}{\# \text{Correct Documents}}$
Precision and Recall of $C_i$

- a, corrects
- b, mistakes
- c, not retrieved

The *Precision* and *Recall* are defined by the above counts:

\[
\text{Precision}_i = \frac{a_i}{a_i + b_i}
\]

\[
\text{Recall}_i = \frac{a_i}{a_i + c_i}
\]
Performance Measurements (cont’d)

- Breakeven Point
  - Find thresholds for which \( \text{Recall} = \text{Precision} \)
  - Interpolation

- f-measure
  - Harmonic mean between precision and recall

- Global performance on more than two categories
  - Micro-average
    - The counts refer to classifiers
  - Macro-average (average measures over all categories)
F-measure e MicroAverages

\[ F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]

\[ \mu\text{Precision} = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + b_i} \]

\[ \mu\text{Recall} = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + c_i} \]

\[ \mu\text{BEP} = \frac{\mu\text{Precision} + \mu\text{Recall}}{2} \]

\[ \mu f_1 = \frac{2 \times \mu\text{Precision} \times \mu\text{Recall}}{\mu\text{Precision} + \mu\text{Recall}} \]
The Impact of $\rho$ parameter on Acquisition category

BEP $\rho$

<table>
<thead>
<tr>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
</tr>
<tr>
<td>0.85</td>
</tr>
<tr>
<td>0.86</td>
</tr>
<tr>
<td>0.87</td>
</tr>
<tr>
<td>0.88</td>
</tr>
<tr>
<td>0.89</td>
</tr>
<tr>
<td>0.9</td>
</tr>
</tbody>
</table>

$\rho$
The impact of $\rho$ parameter on Trade category
Mostly populated categories
Medium sized categories

![Graph showing BEP trends with different categories]
Low size categories

![Graph showing BEP (Base Emission Point) over different categories with categories labeled 0.5, 0.6, 0.7, 0.8, 0.9 across the X-axis and BEP values from 0.55 to 0.9 along the Y-axis. Legend includes Reserves, Rubber, and Dlr.]
Parameter Estimation Procedure

- Validation-set of about 30% of the training corpus
- for all $\rho \in [0,30]$
  - TRAIN the system on the remaining material
  - Measure the BEP on the validation-set
- Select the $\rho$ associated with the highest $BEP$
- re-TRAIN the system on the entire training-set
- TEST the system based on the obtained parameterized model
- For more reliable results:
  - 20 validation-sets and made the $\rho$ average
- The Parameterized Rocchio Classifier will refer to as PRC
Comparative Analysis

- Rocchio literature parameterization
  - $\rho = 1 \ (\gamma = \beta=1) \ and \ \rho = \frac{1}{4} \ (\gamma = 4, \ \beta=16)$
- Reuters fixed test-set
  - Other literature results
- SVM
  - To better collocate our results
- Cross Validation (20 samples)
  - More reliable results
- Cross corpora/language validation
  - Reuters, Ohsumed (English) and ANSA (Italian)
## Results on Reuters fixed split

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>PRC</th>
<th>Std Rocchio ($\gamma = \frac{1}{4} \beta$ or $\gamma = \beta$)</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(~30.000)</td>
<td>82.83%</td>
<td>72.71% - 78.79%</td>
<td>85.34%</td>
</tr>
<tr>
<td>Tokens</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Literature (stems)</td>
<td>-</td>
<td>75 % - 79.9%</td>
<td>84.2 %</td>
</tr>
</tbody>
</table>

- Rocchio literature results (Yang 99’, Choen 98’, Joachims98’)
- SVM literature results (Joachims 98’)

**Tokens:**

- 82.83% for PRC
- 72.71% - 78.79% for Std Rocchio
- 85.34% for SVM

**Literature (stems):**

- 75% - 79.9% for Std Rocchio
- 84.2% for SVM
Breakeven points of widely known classifiers on Reuters

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>PRC</th>
<th>KNN</th>
<th>RIPPER</th>
<th>CLASSI*</th>
<th>Dtree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>85.34%</td>
<td>82.83%</td>
<td>82.3%</td>
<td>82%</td>
<td>80.2%</td>
<td>79.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SWAP1*</th>
<th>CHARADE*</th>
<th>EXPERT</th>
<th>Rocchio</th>
<th>Naive Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>80.5%</td>
<td>78.3%</td>
<td>82.7%</td>
<td>72%-79.5%</td>
<td>75 % - 79.9%</td>
</tr>
</tbody>
</table>

* Evaluation on different Reuters versions
Cross-Validation

1. Generate \( n \) random splits of the corpus. For each split \( j \), 70\% of data can be used for training \((LS^j)\) and 30\% for testing \((TS^j)\).

2. For each split \( j \)
   
   (a) Generate \( m \) validation sets, \( ES^j_k \) of about 10/30\% of \( LS^j \).

   (b) Learn the classifiers on \( LS^j - ES^j_k \) and for each \( ES^j_k \) evaluate:
       (i) the threshold associated to the BEP and (ii) the optimal parameter \( \rho \).

   (c) Learn the classifiers Rocchio, SVMs and \( PRC \) on \( LS^j \): in case of \( PRC \) use the estimated \( \tilde{\rho} \).

   (d) Evaluate \( f_1 \) on \( TS_j \) (use the estimated thresholds for Rocchio and \( PRC \)) for each category and account data for the final processing of the global \( \mu f_1 \).

3. For each classifier evaluate the mean and the Standard Deviation for \( f_1 \) and \( \mu f_1 \) over the \( TS_j \) sets.
N-fold cross validation

- Divide training set in $n$ parts
  - One is used for testing
  - $n-1$ for training
- This can be repeated $n$ times for $n$ distinct test sets
- Average and Std. Dev. are the final performance index
### Cross-Validation on Reuters (20 samples)

<table>
<thead>
<tr>
<th></th>
<th>Rocchio</th>
<th></th>
<th>PRC</th>
<th></th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RTS</td>
<td>TS&lt;sup&gt;o&lt;/sup&gt;</td>
<td>RTS</td>
<td>TS&lt;sup&gt;o&lt;/sup&gt;</td>
<td>RTS</td>
</tr>
<tr>
<td></td>
<td>ρ=.25</td>
<td>ρ=1</td>
<td>ρ=.25</td>
<td>ρ=1</td>
<td>ρ=.25</td>
</tr>
<tr>
<td>earn</td>
<td>95.69</td>
<td>95.61</td>
<td>92.57±0.51</td>
<td>93.71±0.42</td>
<td>95.31</td>
</tr>
<tr>
<td>acq</td>
<td>59.85</td>
<td>82.71</td>
<td>60.02±1.22</td>
<td>77.69±1.15</td>
<td>85.95</td>
</tr>
<tr>
<td>money-fx</td>
<td>53.74</td>
<td>57.76</td>
<td>67.38±2.84</td>
<td>71.60±2.78</td>
<td>62.31</td>
</tr>
<tr>
<td>grain</td>
<td>73.64</td>
<td>80.69</td>
<td>70.76±2.05</td>
<td>77.54±1.61</td>
<td>89.12</td>
</tr>
<tr>
<td>crude</td>
<td>73.58</td>
<td>80.45</td>
<td>75.91±2.54</td>
<td>81.56±1.97</td>
<td>81.54</td>
</tr>
<tr>
<td>trade</td>
<td>53.00</td>
<td>69.26</td>
<td>61.41±3.21</td>
<td>71.76±2.73</td>
<td>80.33</td>
</tr>
<tr>
<td>interest</td>
<td>51.02</td>
<td>58.25</td>
<td>59.12±3.44</td>
<td>64.05±3.81</td>
<td>70.22</td>
</tr>
<tr>
<td>ship</td>
<td>69.86</td>
<td>84.04</td>
<td>65.93±4.69</td>
<td>75.33±4.41</td>
<td>86.77</td>
</tr>
<tr>
<td>wheat</td>
<td>70.23</td>
<td>74.48</td>
<td>76.13±3.53</td>
<td>78.93±3.00</td>
<td>84.29</td>
</tr>
<tr>
<td>corn</td>
<td>64.81</td>
<td>66.12</td>
<td>66.04±4.80</td>
<td>68.21±4.82</td>
<td>89.91</td>
</tr>
<tr>
<td>MicroAvg. 90 cat.</td>
<td><strong>72.61</strong></td>
<td><strong>78.79</strong></td>
<td><strong>73.87±0.51</strong></td>
<td><strong>78.92±0.47</strong></td>
<td><strong>82.83</strong></td>
</tr>
</tbody>
</table>
Ohsumed and ANSA corpora

- **Ohsumed:**
  - Including 50,216 medical abstracts.
  - The first 20,000 documents year 91,
  - 23 MeSH diseases categories [Joachims, 1998]

- **ANSA:**
  - 16,000 news items in Italian from the ANSA news agency.
  - 8 target categories,
  - 2,000 documents each,
  - e.g. Politics, Sport or Economics.

- Testing 30 %
Replacement of an aortic valve cusp after neonatal endocarditis. Septic arthritis developed in a neonate after an infection of her hand. Despite medical and surgical treatment endocarditis of her aortic valve developed and the resultant regurgitation required emergency surgery. At operation a new valve cusp was fashioned from preserved calf pericardium. Nine years later she was well and had full exercise tolerance with minimal aortic regurgitation.
### Cross validation on Ohsumed/ANSA (20 samples)

<table>
<thead>
<tr>
<th></th>
<th>Rocchio</th>
<th>PRC</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ohsumed</strong></td>
<td>BEP</td>
<td>f1</td>
<td>f1</td>
</tr>
<tr>
<td>MicroAvg.</td>
<td>ρ=.25</td>
<td>ρ=1</td>
<td></td>
</tr>
<tr>
<td>(23 cat.)</td>
<td>54.4 ± .5</td>
<td>61.8 ± .5</td>
<td>65.8±.4</td>
</tr>
<tr>
<td><strong>ANSA</strong></td>
<td>BEP</td>
<td>f1</td>
<td></td>
</tr>
<tr>
<td>MicroAvg.</td>
<td>ρ=.25</td>
<td>ρ=1</td>
<td></td>
</tr>
<tr>
<td>(8 cat.)</td>
<td>61.76 ± .5</td>
<td>67.23 ± .5</td>
<td>71.00 ±.4</td>
</tr>
</tbody>
</table>
Computational Complexity

- **PRC**
  - Easy to implement
  - Low training complexity: $O(n \times m \log n \times m)$
    - $(n =$ number of doc and $m =$ max num of features in a document)$
  - Low classification complexity:
    - $\min\{O(M), O(m \times \log(M))\}$ ($M$ is the max num of features in a profile)
    - *Good accuracy: the second top accurate classifier on Reuters*

- **SVM**
  - More complex implementation
  - Higher Learning time > $O(n^2)$ (to solve the quadratic optimization problem)
  - Actually is linear for linear SVMs
  - Low complexity of classification phase (for linear SVM) = $\min\{O(M), O(m \times \log(M))\}$
From Binary to Multiclass classifiers

- Three different approaches:

- **ONE-vs-ALL (OVA)**
  - Given the example sets, \{E_1, E_2, E_3, \ldots\} for the categories: \{C_1, C_2, C_3, \ldots\} the binary classifiers: \{b_1, b_2, b_3, \ldots\} are built.
  - For b_1, E_1 is the set of positives and E_2 \cup E_3 \cup \ldots is the set of negatives, and so on
  - For testing: given a classification instance x, the category is the one associated with the maximum margin among all binary classifiers
ALL-vs-ALL (AVA)

- Given the examples: \{E_1, E_2, E_3, \ldots\} for the categories \{C_1, C_2, C_3, \ldots\}
  - build the binary classifiers:
    \{b_{1_2}, b_{1_3}, \ldots, b_{1_n}, b_{2_3}, b_{2_4}, \ldots, b_{2_n}, \ldots, b_{n-1_n}\}
  - by learning on \(E_1\) (positives) and \(E_2\) (negatives), on \(E_1\) (positives) and \(E_3\) (negatives) and so on…

- For testing: given an example \(x\),
  - all the votes of all classifiers are collected
  - where \(b_{E_1E_2} = 1\) means a vote for \(C_1\) and \(b_{E_1E_2} = -1\) is a vote for \(C_2\)
  - Select the category that gets more votes
From Binary to Multiclass classifiers

- **Error Correcting Output Codes (ECOC)**
  - The training set is partitioned according to binary sequences (codes) associated with category sets.
  - For example, 10101 indicates that the set of examples of C1, C3 and C5 are used to train the C_{10101} classifier.
  - The data of the other categories, i.e. C2 and C4 will be negative examples.
  - In testing: the code-classifiers are used to decode one the original class, e.g.
    \[ C_{10101} = 1 \text{ and } C_{11010} = 1 \]
    indicates that the instance belongs to C1. That is, the only one consistent with the codes.
References

- Machine Learning for TC
  - Lecture slides: http://disi.unitn.it/moschitti/teaching.html
  - In Defense of One-Vs-All Classification, R Rifkin, JMLR jmlr.csail.mit.edu/papers/volume5/rifkin04a/rifkin04a.pdf