Combining Labeled and Unlabeled Data for Multiclass Text Categorization

Rayid Ghani
Accenture Technology Labs
Supervised Learning with Labeled Data

Labeled data is required in large quantities and can be very expensive to collect.
Why use Unlabeled data?

- Very Cheap in the case of text
  - Web Pages
  - Newsgroups
  - Email Messages

- May not be equally useful as labeled data but is available in enormous quantities
Recent work with text and unlabeled data

- Expectation-Maximization (Nigam et al. 1999)
- Co-Training (Blum & Mitchell, 1999)
- Transductive SVMs (Joachims 1999)
- Co-EM (Nigam & Ghani, 2000)
BUT…

- Most of the empirical studies have only focused on binary classification tasks
- ALL Co-Training papers have used two-class datasets
- The largest dataset with EM was 20 Newsgroups (Nigam et al. 1999)
Do current semi-supervised approaches work for “real” and “multiclass” problems?
Empirical Evaluation

- Apply
  - EM
  - Co-Training

- To Multiclass, “real-world” data sets
  - Job Descriptions
  - Web pages
The EM Algorithm

Learn from labeled data

Estimate labels

Add Probabilistically to labeled data
The Co-training Algorithm

[Blum & Mitchell, 1998]

Naïve Bayes on A

Estimate labels

Select most confident

Learn from labeled data

Naïve Bayes on B

Estimate labels

Select most confident

Add to labeled data
Data Sets

- Jobs-65 (from WhizBang!)
  - Job Postings (Two feature sets – Title, Description)
  - 65 categories
  - Baseline 11%

- Hoovers-255 (used in Ghani et al. 2000 and Yang et al. 2002)
  - No natural feature split - random partition of vocab.
  - Collection of Web pages from 4285 corporate websites
  - Each company is classified into one of 255 categories
  - Baseline 2%
Mixed Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Naïve Bayes</th>
<th>EM</th>
<th>Co-Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobs-65</td>
<td>50.1</td>
<td>58.2</td>
<td>54.1</td>
</tr>
<tr>
<td>Hoovers-255</td>
<td>15.2</td>
<td>9.1</td>
<td>10.2</td>
</tr>
</tbody>
</table>

Unlabeled Data Helps!

Unlabeled Data Hurts!
How to extend the unlabeled data framework to multiclass problems?

- N-Class problem $\rightarrow$ N binary problems
- N-Class problem $\rightarrow \binom{N}{2}$ binary problems
- Error-Correcting Output Codes (ECOC)
  - N-Class problem $\rightarrow$ fewer than N binary problems
• **ECOC** (Error Correcting Output Coding)
  
  very accurate and efficient for text categorization with a large number of classes (Berger ’99, Ghani ’00)

  
  +

• **Co-Training**
  
  useful for combining labeled and unlabeled data with a small number of classes
What is ECOC?

- Solve multiclass problems by decomposing them into multiple binary problems (Dietterich & Bakiri 1995)
- Learn the binary problems
Testing ECOC

<table>
<thead>
<tr>
<th></th>
<th>f₁</th>
<th>f₂</th>
<th>f₃</th>
<th>f₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The matrix represents the mapping of the input classes to the ECOC codes.
ECOC - Picture

<table>
<thead>
<tr>
<th></th>
<th>( f_1 )</th>
<th>( f_2 )</th>
<th>( f_3 )</th>
<th>( f_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
ECOC - Picture

<table>
<thead>
<tr>
<th></th>
<th>f₁</th>
<th>f₂</th>
<th>f₃</th>
<th>f₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

X 1 1 1 1 1
ECOC + Co-Training

- ECOC decomposes multiclass problems into binary problems
- Co-Training works great with binary problems

- **ECOC + Co-Train** = Learn each binary problem in ECOC with Co-Training
Important Caveat

- “Normal” binary problems
- “ECOC-induced” binary problems
## Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Naïve Bayes</th>
<th>ECOC</th>
<th>EM</th>
<th>Co-Training</th>
<th>ECOC + Co-Training</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10% Labeled</td>
<td>100% Labeled</td>
<td>10% Labeled</td>
<td>100% Labeled</td>
<td>10% Labeled</td>
</tr>
<tr>
<td>Jobs-65</td>
<td>50.1</td>
<td>68.2</td>
<td>59.3</td>
<td>71.2</td>
<td>58.2</td>
</tr>
<tr>
<td>Hoover s-255</td>
<td>15.2</td>
<td>32.0</td>
<td>24.8</td>
<td>36.5</td>
<td>9.1</td>
</tr>
</tbody>
</table>
Results - Summary

- More labeled data helps
- ECOC outperforms Naïve Bayes (more pronounced with less labeled data)
- Co-Training and EM do not always use unlabeled data effectively
- Combination of ECOC and Co-Training shows great results
Important Side-Effects

- Extremely Efficient (sublinear in the number of categories)
- High Precision Results
A Closer Look…

- Incorrectly Classified examples
- Correctly Classified Examples

% of the total documents

HD to the nearest class

0 1 2 3 4 5 6 7 8 9 10
Results

Precision vs Recall for different algorithms:

- ECOC + CoTrain
- Naive Bayes
- EM

The graph shows the precision and recall values for each algorithm at various recall levels.
Summary

- Combination of ECOC and Co-Training for learning with labeled and unlabeled data
- Effective both in terms of accuracy and efficiency
- High-Precision classification
Teaser...

- Does this only work with Co-Training?