Content-Based Trust and Bias Classification via Biclustering

Dávid Siklósi  Bálint Daróczy  András A. Benczúr

Institute for Computer Science and Control, Hungarian Academy of Sciences

{sdavid,daroczyb,benczur}@ilab.sztaki.hu

WebQuality Workshop 2012
What is the goal?

- Mining opinion from the Web and assessing its quality and trustworthiness
- Help archivest institutions to select trustworthy subcorpora of the Web
- Host level classification is scalable to the size of the Web
- Works well for Spam
- Not for ECML/PKDD Discovery Challenge 2010 new tasks
  - neutrality, bias, trust
- Participants achieved an AUC of $\sim 0.5$ (random) over quality categories
Discovery Challenge 2010 Data Set

- 190K Hosts in the .eu domain crawled by the Internet Memory Foundation
- Labeled into 9 categories
- Train ~ 2500, Test ~ 1300
- Spam (excludes other categories)
- Hosts were labeled by genre into five categories:
  - News/Editorial, Commercial, Educational, Discussion, Personal/Leisure
- Three quality categories:
  - Trustworthiness, Neutrality, Bias
- Publicly available features:
  - tf, idf, content features, link features
Detailed Description of Quality Categories

- **Trustworthiness:**
  - I do not trust this
  - I trust this marginally
  - I trust this fully

- **Neutrality:**
  - Facts
  - Facts & Opinion
  - Opinion

- **Bias:**
  - We adapted the definition from Wikipedia \(^1\)
  - Flame, assaults, dishonest opinion without reference to facts.

\(^1\)http://en.wikipedia.org/wiki/NPOV
Evaluation metrics

1. Area under ROC curve (AUC)
2. Normalized Discounted Cumulative Gain (NDCG) with a slight modification:
   - discount function is changed from the common definition to be linear:
     \[ 1 - \frac{i}{N} \]
   - \[ \text{NDCG} = \frac{\text{DCG}}{\text{Ideal DCG}} \], where
     \[
     \text{DCG} = \sum_{\text{rank}=1}^{N} \text{utility}(\text{rank}) \times \left(1 - \frac{\text{rank}}{N}\right)
     \]
   - Ideal DCG is obtained with utility decreasing with rank
   - NDCG and AUC produces numerically very close values
   - Both measures show certain symmetry over the values 0.5
   - NDCG over an order and its reverse not add up to 1
Baseline

- **DC2010 results:**
  - bags of words representation
  - Decision trees, random forest, SVM, boosting, bagging
  - Feature selection (Fisher, Wilcoxon, Information Gain)

- **Our Previous results:**
  - Ensemble selection, only content features (best)
  - Very strong ingredient: Random forest on BM25

<table>
<thead>
<tr>
<th></th>
<th>spam</th>
<th>news</th>
<th>commercial</th>
<th>research</th>
<th>education</th>
<th>discussion</th>
<th>personal</th>
<th>(non)neutral</th>
<th>biased</th>
<th>(dis)trusted</th>
<th>quality</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NDCG * 1000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC2010 best</td>
<td>833</td>
<td>740</td>
<td>883</td>
<td>885</td>
<td>784</td>
<td>828</td>
<td>620</td>
<td>553</td>
<td>510</td>
<td>561</td>
<td>737</td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>893</td>
<td>811</td>
<td>852</td>
<td>875</td>
<td>865</td>
<td>838</td>
<td>624</td>
<td>656</td>
<td>586</td>
<td>617</td>
<td>771</td>
<td></td>
</tr>
<tr>
<td>BM25</td>
<td>879</td>
<td>791</td>
<td>838</td>
<td>868</td>
<td>848</td>
<td>825</td>
<td>587</td>
<td>656</td>
<td>534</td>
<td>589</td>
<td>704</td>
<td></td>
</tr>
</tbody>
</table>
Overview of the framework

1. Biclustering on the term frequencies
   - Using distances to clusters as feature vectors (bags of concepts)
   - Dimension reduced to the number of clusters
   - Computationally costly classifiers
Overview of the framework

1. **Biclustering on the term frequencies**
   - Using distances to clusters as feature vectors (bags of concepts)
   - Dimension reduced to the number of clusters
   - Computationally costly classifiers

2. **Run SVM on cluster distances**
   - SVM performs well on bags of words representation
   - Hard to find the best kernel
   - Different kernels, different parameters $\rightarrow$ aggregation
Overview of the framework

1. Biclustering on the term frequencies
   - Using distances to clusters as feature vectors (bags of concepts)
   - Dimension reduced to the number of clusters
   - Computationally costly classifiers

2. Run SVM on cluster distances
   - SVM performs well on bags of words representation
   - Hard to find the best kernel
   - Different kernels, different parameters → aggregation

3. Combine results with Random Forest over BM25
Motivation comes from image classification

<table>
<thead>
<tr>
<th>Quality classification</th>
<th>Image classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>hosts = {pages}</td>
<td>images = {points of interests}</td>
</tr>
<tr>
<td>high-dimensional bags of words</td>
<td>high-dimensional image description features</td>
</tr>
<tr>
<td>biclustering</td>
<td>soft clustering</td>
</tr>
<tr>
<td>new representation by distances to clusters</td>
<td>new representation by cluster histograms</td>
</tr>
<tr>
<td>SVM kernel combination methods</td>
<td></td>
</tr>
</tbody>
</table>

What is biclustering?

- Biclustering is a bidirectional clustering method
- It clusters Web hosts and terms at the same time
- Better quality of clusters by using the clustering along the other axis
- Tries to explore a deeper connection between instances and attributes
Our method is based on Dhillon’s information theoretic co-clustering algorithm.

We have substituted Kullback-Leibler divergence to its symmetric version: Jensen-Shannon.

Previous results of ours showed that this modification improves a lot on the quality of clustering.

We used the most frequent 25000 words (good compromise between quality and scalability).

TF outperforms both TF.IDF and BM25.

The final setup:

- 500 host clusters and 1000 term clusters
- 20 iterations (less than 1% of the elements change their class)
Evaluation Method

1. Based on the training set for each cluster and for each category we evaluate the probability that the given cluster belongs to the given category (500 by 9 matrix).

2. Based on the similarity of hosts to clusters we evaluate the probability that a given host belongs to a given cluster ($\#\{\text{number of hosts}\}$ by 500 matrix).

3. By multiplying the above two matrices we get the probability that a given host belongs to a given category.
Re-weighting important words

category-specific words: \[
\frac{\text{overall } tf}{\text{tf in positive instances of category}} > 10
\]

- For every category we find its category-specific words
- We used the category tf as a new weight for these words
- If a term turned out to be specific on more than one categories we used the lower weight
## Biclustering results

<table>
<thead>
<tr>
<th></th>
<th>spam</th>
<th>news</th>
<th>commercial</th>
<th>research</th>
<th>education</th>
<th>discussion</th>
<th>personal</th>
<th>leisure</th>
<th>(non)neutral</th>
<th>biased</th>
<th>(dis)trusted</th>
<th>quality</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NDCG * 1000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC2010 best</td>
<td>833</td>
<td>740</td>
<td>883</td>
<td>885</td>
<td>784</td>
<td>828</td>
<td>620</td>
<td>553</td>
<td>510</td>
<td>561</td>
<td>737</td>
<td></td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>893</td>
<td>811</td>
<td>852</td>
<td>875</td>
<td>865</td>
<td>838</td>
<td>624</td>
<td>656</td>
<td>586</td>
<td>617</td>
<td>771</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BM25</td>
<td>879</td>
<td>791</td>
<td>838</td>
<td>868</td>
<td>848</td>
<td>825</td>
<td>587</td>
<td>656</td>
<td>534</td>
<td>589</td>
<td>704</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bicluster</td>
<td>817</td>
<td>711</td>
<td>770</td>
<td>803</td>
<td>653</td>
<td>719</td>
<td>516</td>
<td>481</td>
<td>450</td>
<td>482</td>
<td>657</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wght. Bicluster</td>
<td>817</td>
<td>719</td>
<td>757</td>
<td>814</td>
<td>771</td>
<td>699</td>
<td>512</td>
<td>592</td>
<td>572</td>
<td>558</td>
<td>694</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Word cluster examples

- yorkie adorable puppy teacup capuchin affectionate akc parrots
  - maltese puppies lovely cute
- serbia croatia bosnia albania montenegro macedonia herzegovina
  - belarus moldova kosovo azerbaijan slovak balkans estonian
- welcome tel fax submit home please mail click contact reserved
- plated earrings necklace pendants necklaces bracelets studs
  - jewelry jewellery
- laptops cheap discount buy
- yeah awesome folks wondering okay yes nice maybe pretty hello
  - yesterday guys wow guess
- tabs erectile erection pfizer impotence generic

- Quite a few one word clusters:
  - ebay, image, friend, lifestyle
SVM

- libSVM

- For every host we assigned the distance of the host to the host clusters as a feature vector

- We used different SVM kernels with different parameters:
  - linear: $K(x, y) = x' \ast y$
  - polynomial: $K(x, y) = (\frac{1}{D} x' \ast y)^d$
  - radial basis function: $K(x, y) = e^{-\gamma(x-y)^2}$

- Where $D = \text{number of features}$, $d=1, 2, 4$ and $\gamma = \frac{1}{|T|}$, where $T$ is the training set
Kernel aggregation strategies

1. Select best: for each category we select the kernel that performs the best on held out.

2. Early aggregation: we combine the kernels according to the ideal weight over the held out:

\[
pred_{early}(x) = \sum_{i=1}^{N} \alpha_i \sum_{k=1}^{K} \beta_k K_k(x, y_i) + b
\]

where \(K_k(x, y_i)\) is the \(k\)th kernel and \(b\) is the bias.

3. Late fusion: we combine the svm outputs according to the ideal weight over the held out:

\[
pred_{late}(x) = \sum_{k=1}^{K} \beta_k \left( \sum_{i=1}^{N} \alpha_i K_k(x, y_i) + b_k \right)
\]

where \(K_k(x, y_i)\) is the \(k\)th kernel and \(b_k\) is the bias for the \(k\)th SVM classifier.
Results of SVM Kernel Aggregations

<table>
<thead>
<tr>
<th>NDCG * 1000</th>
<th>spam</th>
<th>news</th>
<th>commercial</th>
<th>research</th>
<th>education</th>
<th>discussion</th>
<th>personal</th>
<th>(non)neutral</th>
<th>biased</th>
<th>(dis)trusted</th>
<th>quality average</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DC2010 best</strong></td>
<td>833</td>
<td>740</td>
<td>883</td>
<td>885</td>
<td>784</td>
<td>828</td>
<td>620</td>
<td>553</td>
<td>510</td>
<td>561</td>
<td>737</td>
<td></td>
</tr>
<tr>
<td><strong>best</strong></td>
<td>893</td>
<td>811</td>
<td>852</td>
<td>875</td>
<td>865</td>
<td>838</td>
<td>624</td>
<td>656</td>
<td>586</td>
<td>617</td>
<td>771</td>
<td></td>
</tr>
<tr>
<td><strong>BM25</strong></td>
<td>879</td>
<td>791</td>
<td>838</td>
<td>868</td>
<td>848</td>
<td>825</td>
<td>587</td>
<td>656</td>
<td>534</td>
<td>589</td>
<td>704</td>
<td></td>
</tr>
<tr>
<td><strong>Bicluster</strong></td>
<td>817</td>
<td>711</td>
<td>770</td>
<td>803</td>
<td>653</td>
<td>719</td>
<td>516</td>
<td>481</td>
<td>450</td>
<td>482</td>
<td>657</td>
<td></td>
</tr>
<tr>
<td><strong>WBicluster</strong></td>
<td>817</td>
<td>719</td>
<td>757</td>
<td>814</td>
<td>771</td>
<td>699</td>
<td>512</td>
<td>592</td>
<td>572</td>
<td>558</td>
<td>694</td>
<td></td>
</tr>
<tr>
<td><strong>Bic. Comb. SVM</strong></td>
<td>825</td>
<td>795</td>
<td>902</td>
<td>898</td>
<td>800</td>
<td>855</td>
<td>638</td>
<td>615</td>
<td>637</td>
<td>630</td>
<td>774</td>
<td></td>
</tr>
<tr>
<td><strong>Fusion Bicluster</strong></td>
<td>819</td>
<td>801</td>
<td>896</td>
<td>898</td>
<td>810</td>
<td>856</td>
<td>614</td>
<td>539</td>
<td>627</td>
<td>593</td>
<td>763</td>
<td></td>
</tr>
<tr>
<td><strong>Fusion WBicluster</strong></td>
<td>828</td>
<td>747</td>
<td>899</td>
<td>898</td>
<td>824</td>
<td>849</td>
<td>636</td>
<td>615</td>
<td>641</td>
<td>630</td>
<td>771</td>
<td></td>
</tr>
<tr>
<td><strong>Fusion all</strong></td>
<td>838</td>
<td>798</td>
<td>904</td>
<td>897</td>
<td>836</td>
<td>860</td>
<td>643</td>
<td>615</td>
<td>641</td>
<td>633</td>
<td>781</td>
<td></td>
</tr>
</tbody>
</table>
Combination with Random Forest over BM25

- We took the average of the predictions
- For RF over BM25 didn’t had predictions for held out
## Overall results

<table>
<thead>
<tr>
<th>Method</th>
<th>spam</th>
<th>news</th>
<th>commercial</th>
<th>research</th>
<th>education</th>
<th>discussion</th>
<th>personal</th>
<th>leisure</th>
<th>(non)neutral</th>
<th>biased</th>
<th>(dis)trusted</th>
<th>quality average</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC2010 best</td>
<td>833</td>
<td>740</td>
<td>883</td>
<td>885</td>
<td>784</td>
<td>828</td>
<td>620</td>
<td>553</td>
<td>510</td>
<td>586</td>
<td>737</td>
<td>561</td>
<td>771</td>
</tr>
<tr>
<td>best</td>
<td><strong>893</strong></td>
<td>811</td>
<td>852</td>
<td>875</td>
<td>865</td>
<td>838</td>
<td>624</td>
<td>656</td>
<td>586</td>
<td>534</td>
<td>704</td>
<td>589</td>
<td>694</td>
</tr>
<tr>
<td>BM25</td>
<td>879</td>
<td>791</td>
<td>838</td>
<td>868</td>
<td>848</td>
<td>825</td>
<td>587</td>
<td>656</td>
<td>534</td>
<td>589</td>
<td>704</td>
<td>589</td>
<td>704</td>
</tr>
<tr>
<td>Bicluster</td>
<td>817</td>
<td>711</td>
<td>770</td>
<td>803</td>
<td>653</td>
<td>719</td>
<td>516</td>
<td>481</td>
<td>450</td>
<td>482</td>
<td>657</td>
<td>482</td>
<td>657</td>
</tr>
<tr>
<td>WBicluster</td>
<td>817</td>
<td>719</td>
<td>757</td>
<td>814</td>
<td>771</td>
<td>699</td>
<td>512</td>
<td>592</td>
<td>572</td>
<td>558</td>
<td>694</td>
<td>558</td>
<td>694</td>
</tr>
<tr>
<td>Bic. Comb. SVM</td>
<td>825</td>
<td>795</td>
<td>902</td>
<td>898</td>
<td>800</td>
<td>855</td>
<td>638</td>
<td>615</td>
<td>637</td>
<td>630</td>
<td>774</td>
<td>630</td>
<td>774</td>
</tr>
<tr>
<td>Fusion Bicluster</td>
<td>819</td>
<td>801</td>
<td>896</td>
<td>898</td>
<td>810</td>
<td>856</td>
<td>614</td>
<td>539</td>
<td>627</td>
<td>593</td>
<td>763</td>
<td>593</td>
<td>763</td>
</tr>
<tr>
<td>Fusion WBicluster</td>
<td>828</td>
<td>747</td>
<td>899</td>
<td>898</td>
<td>824</td>
<td>849</td>
<td>636</td>
<td>615</td>
<td>641</td>
<td>630</td>
<td>771</td>
<td>630</td>
<td>771</td>
</tr>
<tr>
<td>Fusion all</td>
<td>838</td>
<td>798</td>
<td>904</td>
<td>897</td>
<td>836</td>
<td>860</td>
<td><strong>643</strong></td>
<td>615</td>
<td>641</td>
<td>633</td>
<td>781</td>
<td>633</td>
<td>781</td>
</tr>
<tr>
<td>Fusion Bicluster + BM25</td>
<td>876</td>
<td>836</td>
<td>899</td>
<td>904</td>
<td>867</td>
<td>870</td>
<td>601</td>
<td>673</td>
<td>570</td>
<td>614</td>
<td>789</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fusion WBicluster + BM25</td>
<td>884</td>
<td>804</td>
<td>899</td>
<td>902</td>
<td>866</td>
<td>860</td>
<td>628</td>
<td>685</td>
<td>581</td>
<td>634</td>
<td>790</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fusion all + BM25</td>
<td>883</td>
<td>834</td>
<td>900</td>
<td>904</td>
<td>874</td>
<td>870</td>
<td>628</td>
<td>685</td>
<td>581</td>
<td>634</td>
<td>795</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

First attempt to give practically useful classification for quality categories (as we know)
Strong improvement over baseline results
Neutrality and trust behaves very different from genre classification

Future work

Better combination with random forest over BM25
Early aggregation of SVM kernels (performs better for image classification)
More refined feature selection then re-weighting important terms
Natural language processing features
Questions?