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 ~ 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 $\sim$ 2500, Test $\sim$ 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:
     $1 - \frac{i}{N}$
   - $NDCG = \frac{DCG}{Ideal\ DCG}$, where

     $DCG = \sum_{rank=1}^{N} utility(rank) * (1 - \frac{rank}{N})$
   - 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

NDCG $*$ 1000	spam	news	commercial	research education	discussion	personal leisure	(non)neutral	biased	(dis)trusted	quality average	average
DC2010 best	833	740	883	885	784	828	620	553	510	561	737
best	893	811	852	875	865	838	624	656	586	617	771
BM25	879	791	838	868	848	825	587	656	534	589	704

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 $\rightarrow$ aggregation
3. Combine results with Random Forest over BM25

Motivation comes from image classification

Quality classification	Image classification
• hosts = {pages}	• images = {points of interests}
• high-dimensional bags of words	• high-dimensional image description features
• biclustering	• soft clustering
• new representation by distances to clusters	• new representation by cluster histograms

- SVM kernel combination methods

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

Information Theoretic Biclustering

- Our method is based on Dhillon's information theoretic co-clustering algorithm
- We have substituted Kullback-Leibler divergence to it's 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 then 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 ($\#\{$*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 a category

Re-weighting important words

category-specific words: $\frac{\textit{overall tf}}{\textit{tf in positive instances of category}} > 10$

- 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

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

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: $K(x,y) = x' * y$
  - polynomial: $K(x,y) = (\frac{1}{D}x' * y)^d$
  - radial basis function: $K(x,y) = e^{(-\gamma(x-y)^2)}$
- Where $D = \textit{number of features}$, 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 kth kernel and b is the bias
3. Late fusion: we combine the svm outputs according to the ideal weight over the held out:

   $pred_{late}(x) = \sum_{k=1}^{K} \beta_k (\sum_{i=1}^{N} \alpha_i K_k(x, y_i) + b_k)$ where

   $K_k(x, y_i)$ is the kth kernel and b_k is the bias for the kth SVM classifier

Results of SVM Kernel Aggregations

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

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

$NDCG * 1000$	spam	news	commercial	research education	discussion	personal leisure	(non)neutral	biased	(dis)trusted	quality average	average
DC2010 best	833	740	883	885	784	828	620	553	510	561	737
best	**893**	811	852	875	865	838	624	656	586	617	771
BM25	879	791	838	868	848	825	587	656	534	589	704
Bicluster	817	711	770	803	653	719	516	481	450	482	657
WBicluster	817	719	757	814	771	699	512	592	572	558	694
Bic. Comb. SVM	825	795	902	898	800	855	638	615	637	630	774
Fusion Bicluster	819	801	896	898	810	856	614	539	627	593	763
Fusion WBicluster	828	747	899	898	824	849	636	615	**641**	630	771
Fusion all	838	798	**904**	897	836	860	**643**	615	**641**	633	781
Fusion Bicluster + BM25	876	**836**	899	**904**	867	870	601	673	570	614	789
Fusion WBicluster + BM25	884	804	899	902	866	860	628	**685**	581	**634**	790
Fusion all + BM25	883	834	900	**904**	**874**	**870**	628	**685**	581	**634**	**795**

Summary

- 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?