The MultiRank Bootstrap Algorithm: Semi-Supervised Political Blog Classification and Ranking Using Semi-Supervised Link Classification

Frank Lin and William W. Cohen

CMU-LTI-08-003

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

The MultiRank Bootstrap Algorithm: Semi-Supervised Political Blog Classification and Ranking Using Semi-Supervised Link Classification

Frank Lin
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
frank@cs.cmu.edu

William W. Cohen
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
wcohen@cs.cmu.edu

Abstract

We present a new, intuitive semi-supervised learning algorithm for classifying political blogs in a blog network and ranking them within classes. In the algorithm each link is assigned a label as well as the blogs. Using only the link structure as input and by exploiting the linking properties found in political blog communities, we bootstrap the classification of links and blogs and blog rankings from a set of known seed blogs. We test our algorithm on two datasets and achieve blog classification accuracy of 81.9% in a network of 404 blogs and 84.6% in a network of 1222 blogs using only 2 seed blogs in each case. We analyze the results our algorithm and show that the misclassifications tend to be less important or less authoritative blogs.

Keywords

Social Network Analysis, Communities Identification, Centrality/ Influence of Bloggers/Blogs, Ranking/Relevance of blogs, Polarity/Opinion Identification and Extraction

1. Introduction

The link structure of hypertext has been used for many purposes, including detecting web communities [12, 7, 3], web page classification [6, 5, 10], and ranking web pages by authority [13, 11]. In most uses of hypertext, links between web sites are interpreted as *recommendations*. This is a simplification, as web pages link to one another for many reasons. In the domain of political discourse, for instance, a link might represent *agreement* with a position taken by the linked-to page; *disagreement* with a position; a citation of a page as *factual support* for a claim being made; or a news story that is the *subject* of a discussion.

With this in mind, Kale et al. proposed an algorithm using polar links to model trust and influence in the blogosphere [10]. Using the trust propagation algorithm proposed in [8], they use a positive and negative words dictionary to annotate links with polar weights through which trust scores are propagated from a set of known influential seed nodes. They evaluated the algorithm by using trust scores to classify a graph of 300 Republican and Democratic political blogs. The results, which show the benefit of using link polarity in trust propagation, were disappointing in terms of overall classification accuracy, perhaps due to the inadequacy of a dictionary-based method for link classification.

In this work, we propose a novel algorithm that both classifies political blogs and ranks the blogs within the predicated class. Our work, like Kale et al., is also based on link classification and propagation of information an initial set of known seed nodes. However, instead of classifying each link as a weighted positive link or negative link, we assign each link as one that *endorses* a particular class. For example, a political blog with a Democratic leaning may have three links, two to other Democratic blogs and one to a Republican blog; the two links then can be labeled as having a Democratic endorsement and the one link as having a Republican endorsement. In predicting the link label, instead of using a dictionary or a trained classifier based on textual features, we exploit a particular link property found in the political blogosphere: that blogs with similar political leaning tend to link to each other more often [2]. Using the link structure alone [1], we bootstrap the classification of the blogs and the links and the ranking of the blogs by propagating political leaning from an initial set of known seed nodes via MultiRank - algorithm based on PageRank. We show that our algorithm achieves high classification accuracy compared with Kale et al. when applied to a network of political blogs containing Republican and Democratic blogs and analyze the results.

2. Proposed Algorithm

In the heart of our algorithm lies the idea that, in a directed network or a graph where every node is assigned a label (this can be soft or hard), each edge can also be assign a similar label. We call this label assignment the *faction* of the node or the edge, and faction of an edge should be the same as the faction of its target node. In the context of political blogosphere, where each node is a blog in a network of blogs, we can see each link as *endorsing* a particular political faction by linking to a political blog aligned with that particular faction.

[1] We may also incorporate textual features into the algorithm via personalized PageRank algorithm [9] by training on some a separate dataset and initialize personalization vector accordingly

Language Technologies Institute Tech Report: CMU-LTI-08-003

2.1 The MultiRank Ranking Algorithm

This idea naturally gives rise to MultiRank, a simple algorithm based on PageRank [13, 9]. The original PageRank is defined as follows: given a direct graph G and a personalization vector $\mathbf{u}$, it returns a ranking vector $\mathbf{r}$ satisfying the following equation:

$$\mathbf{r} = (1-d)\mathbf{u} + dW\mathbf{r} \quad (1)$$

where W is weighted transition matrix of graph G where transition from i to j is given by $W_{ij} = 1/degree(i)$ and d is a constant damping factor. If we assume a uniform personalization vector (which we shall do throughout the rest of this paper), Equation 1 can be simply defined in terms of G ($\mathbf{r} = PageRank(G)$) and the $\mathbf{r_i}$ can be interpreted as the probability of a random walk on G arriving at node i, with teleportation probability $(1-d)$ at any given time to any node with a uniform distribution.

We then define MultiRank based on PageRank as follows: given a directed graph G and a set of edges E_f belonging to faction f, it returns a ranking vector $\mathbf{r_f}$ of faction f satisfying the following equation:

$$\mathbf{r_f} = (1-d)\mathbf{u} + dW_f\mathbf{r_f} \quad (2)$$

where W_{fij} is W_{ij} if the edge from i to j is in E_f, otherwise zero; and $\mathbf{u}$ is the uniform personalization vector where $u_i = 1/|V|$. This simple modification of the PageRank is very intuitive and $\mathbf{r_f}$ can be seen as the probability of a random walk on G if the we only follow edges belongs to faction f. In context of a political blog network, we can see this as the probability of a Democratic or Republican random blog surfer randomly clicking on links pointing to a Democratic or Republican blog; and this probability can be interpreted as a ranking score on how popular or authoritative a blog is [13].

2.2 The MultiRank Bootstrap Algorithm

The original PageRank algorithm requires only the graph as the input. However, in order to calculate $\mathbf{r_f}$, we need E_f. In terms of a political blog network, this means we need to either a) label all the links with a political leaning or b) label all the blogs with a political leaning, from which we can derive the link labels. In order to solve this problem, we propose a intuitive iterative bootstrapping algorithm to gradually expand the set of edges E_f from a set of initial seeds nodes S until the every edge in the entire graph has been labeled.

The algorithm begins with labeling all edges incident to nodes S with the same faction as the seed nodes they are incident to. Then, on each iteration, the algorithm runs a inner loop where it alternatively run MultiRank on the graph using currently labeled set of edges, label each node according to its highest ranked faction, and then change the labeling of the set edges according to node labeling. This inner loop is ran until there is no change on edge labeling. After the inner loop terminates, we expand the set of edges according to an expansion metric $M(G, f)$. This repeats until all the edges have been labeled. The formal definition of the algorithm is shown in Figure 1.

In our experiments we have only tried two simple metrics: the first metric simply expand on all currently unlabeled edges neighboring currently labeled edges. An unlabeled edge u neighbors a labeled edge v if they share a common endpoint, and the newly expanded edge is labeled with the faction of the common endpoint. The second metrics is similar to the first, except we limit it to n unlabeled edges incident to the nodes with the highest combined ranking $\sum_f \mathbf{r_f}(v)$, where n is equal to the number of nodes incident to currently labeled edges. This basically means that the subgraph on which we run the algorithm will at most double in size in terms of the number of nodes on each iteration. We refer to the first metric as *infinite* expansion and the second as *controlled* expansion.

After the algorithm converges, we can classify the edges according to the final faction labeling, rank the nodes within each faction according to $\mathbf{r_f}$, and classify the nodes according to $argmax_f \mathbf{r_f}(v)$.

Input: A graph $G = (V, E)$, set of seed nodes S, an edge expansion metric on the graph $M(G, f)$ that returns a set of previously unlabeled edges and label them f
Output: Ranking vectors $r_{f=1...n}$ where f correspond to each faction
Algorithm:

- initialize E_f using S
- while $|\bigcup_{f=1...n} E_f| \neq |E|$ do
  - $e \leftarrow infinity$
  - while $e > 0$
    - $\mathbf{r_f} \leftarrow MultiRank(G, E_f) \forall f$
    - $label(v) \leftarrow argmax_f \mathbf{r_f}(v) \forall v \in V$
    - $E'_f \leftarrow \{e(x \rightarrow v) \in E : label(v) = f\} \forall f$
    - $e \leftarrow |E'_f - E_f|$
    - $E_f \leftarrow E'_f \forall f$
  - $E_f \leftarrow E_f \bigcup M(G, f) \forall f$

Fig. 1: *The MultiRank bootstrap algorithm (Exploratory Phase)*

2.3 Settling Phase

We also present a second, optional phase to the algorithm that may further improve the output of the first phase; we will refer to the original algorithm shown in Figure 2 as the *exploratory* phase and the second extension algorithm as the *settling* phase. The settling phase again exploits the link property found in political blog network, that blogs from the same political faction are more likely to link each other. This is done by first finding all the nodes where the majority of the neighbors are of an different faction (which violates the link property), changing the labeling of its incoming edges to the majority neighbor faction, and running the MultiRank algorithm on the graph again. This is repeated until the algorithm converges. The algorithm converges when there are no more changes in edge labeling or when the algorithm revisits an old state due to cycling changes (it is possible for this algorithm to having cycling changes - consider a network of two nodes labeled with different factions and each node has an edge pointing to the other node).

3. Datasets

To assess the effectiveness of our algorithm, we tested it on two datasets. The first dataset is constructed in the same

way as described in [10], by finding a set of overlapping blogs between the ICWSM 2007 BuzzMetrics [1] dataset and Lada Adamic's labeled dataset [2] described in [2], and then a graph is generated using links found in the BuzzMetrics dataset. Then we take the largest (weakly) connected component of the graph and ended up with a dataset of 404 connected blogs. We will refer to this as the Kale dataset.

The second dataset is constructed by simply creating a graph from [2] and taking the largest (weakly) connected component. This dataset, which we will refer to as the Adamic dataset, contains 1222 connected blogs.

Though the two datasets are not entirely independent, as the blogs from the Kale dataset are mostly a subset of the blogs from the Adamic dataset; using the above method we effectively create two distinct datasets of different size and link structure. In the Kale dataset, the links are gathered around May 2006 from the links embedded in content of the blog posts; whereas in the Adamic dataset the links are from two months before the 2004 presidential election and are found both embedded in the text and on the sidebars). Some statistics of these datasets are shown in Table 1 and Table 2. To construct a graph from these datasets, every blog is a node in the graph, and a directed edge from node a to node b exist in the graph if a post in blog a contains a link to a post in blog b. Finally, it should be pointed out that the dataset labeling is not 100% accurate as noted in [2].

Total Blogs	404
Democratic Blogs	198
Republican Blogs	206
Total Links	2725
Links to Democratic Blogs	1250
Links to Republican Blogs	1475
Intra-Faction Links	2345
Inter-Faction Links	380

Table 1: *Some statistics of the Kale Dataset*

Total Blogs	1222
Democratic Blogs	586
Republican Blogs	636
Total Links	19021
Links to Democratic Blogs	9287
Links to Republican Blogs	9734
Intra-Faction Links	17338
Inter-Faction Links	1683

Table 2: *Some statistics of the Adamic Dataset*

4. Experiment Results and Discussions

We run our algorithm on the two datasets varying three parameters: the number of seed nodes, the expansion metric, and the inclusion or exclusion of the optional "settling phase." In all our experiments, we pick the starting seeds by first running the original PageRank on the entire graph, and for a seed set of size n, we take the top ranked $n/2$ Democratic blogs and the top ranked $n/2$ Republican blogs as the set of seed nodes with known labels. In all instances of the MultiRank algorithm the damping factor d is set to 0.85, a popular choice of damping factor that we borrowed without tuning on data. Aside from the three parameters we mentioned above and which result we will report, the algorithm has no parameters to tune; the PageRank-based heuristic for picking the initial seeds is the only seed picking we have experimented with.

[2] We added 50 additional labeled blogs to the by automatically crawling www.blogcatalog.com under Democratic and Republican blog categories

4.1 Classification Results

Kale Dataset with Infinite Expansion

	Exploratory		Settling	
Seeds	Vertex	Edge	Vertex	Edge
2	0.641	0.763	0.819	0.968
4	0.698	0.876	0.804	0.952
8	0.703	0.894	0.804	0.952
12	0.700	0.893	0.804	0.952
16	0.728	0.917	0.804	0.952
20	0.757	0.952	0.807	0.966

Kale Dataset with Controlled Expansion

	Exploratory		Settling	
Seeds	Vertex	Edge	Vertex	Edge
2	0.787	0.898	0.804	0.952
4	0.770	0.912	0.819	0.968
8	0.785	0.949	0.819	0.968
12	0.827	0.953	0.804	0.952
16	0.824	0.953	0.804	0.952
20	0.780	0.959	0.804	0.965

Fig. 2: *Blog (Vertex) and link (Edge) classification accuracy on the Kale dataset*

Adamic Dataset with Infinite Expansion

	Exploratory		Settling	
Seeds	Vertex	Edge	Vertex	Edge
2	0.700	0.835	0.846	0.978
4	0.744	0.888	0.849	0.978
6	0.745	0.892	0.849	0.978
10	0.736	0.880	0.849	0.978
20	0.731	0.889	0.847	0.977
30	0.734	0.901	0.848	0.978
40	0.708	0.909	0.846	0.977

Adamic Dataset with Controlled Expansion

	Exploratory		Settling	
Seeds	Vertex	Edge	Vertex	Edge
2	0.593	0.776	0.845	0.977
4	0.614	0.770	0.848	0.978
6	0.797	0.887	0.854	0.978
10	0.727	0.872	0.849	0.978
20	0.743	0.916	0.849	0.978
30	0.774	0.939	0.849	0.978
40	0.760	0.945	0.849	0.978

Fig. 3: *Blog (Vertex) and link (Edge) classification accuracy on the Adamic dataset*

The classification accuracies of the algorithm on the datasets are shown in Figure 2 and 3. We point out some interesting observations on the effect of the three variables.

First, inclusion of the optional settling phase tends to improve upon the results of the first exploratory phase, up to a certain extend; the resulting accuracies from including the settling phase is almost constant regardless of the number of seeds and the expansion metric. However, the improvement

gain of the settling phase can only be push so far; for controlled expansion with 12 and 16 seeds on the Kale dataset, we see the settling phase actually hurt the performance. This makes sense since settling phase would only work inasmuch that the blog network follows its simple assumption: that a the majority of the neighbors of a blog would be of the same faction as the blog.

Second, as expected, increasing the number of seeds tend to increase the performance of the algorithm (during the exploratory phase, before the settling phase levels out the differences). What is surprising is that the algorithm work very well even with the smallest of number seeds allowed by the algorithm - two, one seed per faction. This may indicate that picking the initial seeds according to their PageRank score is a good heuristic.

Looking at the accuracies varying the expansion metric (again, before the settling phase levels out the differences), we see that, in general, controlling the expansion helps to improve the algorithm. This is intuitive since for every expansion we make a "leap of faith" as to what the expanded edges are before running the algorithm's inner loop to convergence. The smaller the leap the may mean less mistakes are made during expansion. However, smaller leaps also mean more iterations of the algorithm - a speed versus performance tradeoff. The counter example to this is the infinite expansion on the Adamic dataset with smaller number of seeds, where it outperforms the controlled expansion. Here we can only conjuncture that we may have gotten "lucky" with really good seeds, and it is better to quickly expand on these.

For comparison with supervised, text-based classifier, we provide blog classification results on the Kale dataset using a Naïve Bayes classifier with bag-of-words features. Shown in Table 3, each reported accuracy is averaged over 10 runs of 10-fold cross-validation experiments. Though falling slightly below the MultiRank boostrap with settling phase even at 90% training data, the results show good classification performance that may potentially be incorporated into the MultiRank boostrap algorithm in future work.

% Data	Accuracy
10	0.547
30	0.599
50	0.636
70	0.697
90	0.774

Table 3: *Blog classification accuracy on the Kale dataset using a Naïve Bayes classifier with bag-of-words features. % Data indicates the percent of data used for training. All experiments are tested on 10% of the data.*

4.2 Ranking Results

We also look at the rankings generated by the MultiRank bootstrap algorithm by comparing the top 20 ranked lists with ones produced by Adamic et al. using the heuristic described in [2]. The comparison is shown in Table 4 and 5. Note that Adamic et al. manually removed sites with unusual format and ones that primarily function as a news filter such as *drudgereport.com*, found in our top 20 list of conservative (Republican) blogs, among other news filter-style blogs.

4.3 Classification Bias

Being a ranking-based classification algorithm, the classification accuracy of MultiRank boostrap is biased according to the ranking of the nodes. Specifically, it tends to favor (do better on in terms of accuracy) more popular or more authoritative political blogs. This is clearly shown in Figure 4 on a Precision-Recall curve, where each data point is the average precision (or accuracy) at rank i. The solid line indicates the curve drawn when going from high rank to low rank, and the dotted line indicates the curve drawn from low to high. In many applications, this bias is an advantage because users are more likely to visit and surf between more popular and authoritative blogs.

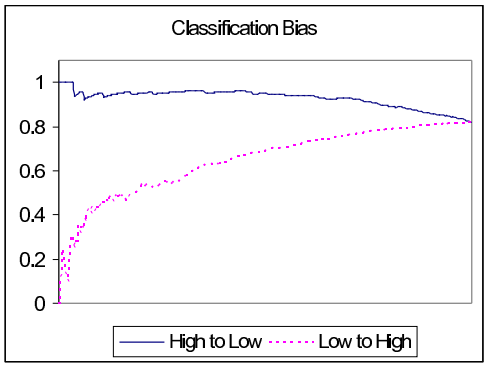

Fig. 4: *The precision-recall curve showing the classification bias. The vertical axis is the average precision (accuracy) and the horizontal axis is the rank. Each data point is the average precision (accuracy) at rank i. The solid line indicates the curve drawn when going from high rank to low rank, and the dotted line indicates the curve drawn from low to high.*

5. Conclusions and Future Work

We have introduced a new semi-supervised algorithm for simultaneously classifying and ranking political blogs based on link structure that shows high classification accuracy compared with previously reported results on a similar dataset. We showed that this algorithm requires very few initial seeds - as little as one seed per class - and with little to no parameter tuning, it achieves performance above 80% on two political blog datasets of different size and link structure. This algorithm tend favor more popular and authoritative blogs in terms of classification accuracy.

There is much waiting to be explored in terms of future work. For the algorithm itself, we can try different algorithms for picking seeds and different expansion metrics. We can even modify the expansion metric with a trained link classifier or a blog classifier, instead of always assigning the link to the faction of the common endpoint with the currently labeled link. As suggested above, we can also used a textually trained blog classifier to create a personalization vector and use personalized PageRank algorithm. To test the generality of the algorithm, we should try it on networks with more than two political factions, and also on networks that are much larger and more complicated. It may also be interesting to run the algorithm on machine learning datasets [4] where the graph is constructed based on a distance metric between data points, and compare it against a number of semi-supervised learning that has been tested on these datasets.

Top 20 Ranked Liberal (Democratic) Blogs

by Adamic et al.	by MultiRank Bootstrap
dailykos.com	**dailykos.com**
talkingpointsmemo.com	**atrios.blogspot.com**
atrios.blogspot.com	**talkingpointsmemo.com**
washingtonmonthly.com	**washingtonmonthly.com**
wonkette.com	**juancole.com**
juancole.com	**prospect.org/weblog**
yglesias.typepad.com/matthew	**digbysblog.blogspot.com**
crookedtimber.org	**j-bradford-delong.net/movable_type**
mydd.com	**talkleft.com**
oliverwillis.com	**yglesias.typepad.com/matthew**
blog.johnkerry.com	**politicalwire.com**
pandagon.net	gadflyer.com
talkleft.com	**pandagon.net**
digbysblog.blogspot.com	emergingdemocraticmajorityweblog.com
politicalwire.com	reachm.com/amstreet
j-bradford-delong.net/movable_type	**jameswolcott.com**
prospect.org/weblog	billmon.org
americablog.blogspot.com	wampum.wabanaki.net
theleftcoaster.com	maxspeak.org/mt
jameswolcott.com	**mydd.com**

Table 4: *Comparison of the top 20 ranked liberal blogs produced by Adamic et al. and the MultiRank bootstrap algorithm on the Adamic dataset using 2 seeds and infinite expansion with settling phase. The blogs in boldface appear in both lists.*

Top 20 Ranked Convervative (Republican) Blogs

by Adamic et al.	by MultiRank Bootstrap
powerlineblog.com	moorewatch.com
instapundit.com	right-thinking.com
littlegreenfootballs.com/weblog	**instapundit.com**
hughhewitt.com	**michellemalkin.com**
andrewsullivan.com	**blogsforbush.com**
captainsquartersblog.com/mt	**littlegreenfootballs.com/weblog**
wizbangblog.com	**powerlineblog.com**
indcjournal.com	**vodkapundit.com**
michellemalkin.com	**andrewsullivan.com**
blogsforbush.com	drudgereport.com
allahpundit.com	**hughhewitt.com**
belmontclub.blogspot.com	**volokh.com**
realclearpolitics.com	rightwingnews.com
volokh.com	truthlaidbear.com
timblair.spleenville.com	freerepublic.com
windsofchange.net	nationalreview.com/thecorner
vodkapundit.com	**rogerlsimon.com**
rogerlsimon.com	**captainsquartersblog.com/mt**
deanesmay.com	lashawnbarber.com
mypetjawa.mu.nu	jewishworldreview.com

Table 5: *Comparison of the top 20 ranked liberal blogs produced by Adamic et al. and the MultiRank bootstrap algorithm using 2 seeds and infinite expansion with settling phase. The blogs in boldface appear in both lists.*

References

[1] Nielsen Buzzmetrics, www.nielsenbuzzmetrics.com.

[2] L. Adamic and N. Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem*, 2005.

[3] R. Andersen and K. J. Lang. Communities from seed sets. In *Proceedings of the Fifteenth International World Wide Web Conference (WWW 2006)*, 2006.

[4] A. Azran. The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[5] W. W. Cohen. Improving a page classifier with anchor extraction and link analysis. In *Advances in Neural Information Processing Systems 15*, 2002.

[6] M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. M. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1/2):69–113, 2000.

[7] G. W. Flake, S. Lawrence, C. L. Giles, and F. Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[8] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004)*, 2004.

[9] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University, 2003.

[10] A. Kale, A. Karandikar, P. Kolari, A. Java, T. Finin, and A. Joshi. Modeling trust and influence in the blogosphere using link polarity. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[12] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1481–1493, 1999.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.