Graph-based Semi-Supervised Learning for Text Categorization through Supervised Random Walks

Tom Shen
zhixians@andrew.cmu.edu

William W. Cohen
wcohen@cs.cmu.edu

May 1, 2015

Contents

Abstract

Recently, many effective graph-based algorithms for semi-supervised learning have been developed. These algorithms can often be applied to text classification tasks, since a similarity graph can be easily constructed from a set of text documents. We introduce a graph-based semi-supervised learning algorithm based an algorithm used in ProPPR, a first-order probabilistic language for logical inference. ProPPR answers queries by converting them to graphs and running an algorithm based on supervised random walks. This algorithm was chosen for efficiency purposes and for suitability to the problem. In this paper, we will show that this algorithm is also an effective algorithm for graph-based semi-supervised learning on its own, especially for text classification. We compare it to two existing state-of-the-art algorithms, modified adsorption and Gaussian random fields, and show that it outperforms them on text classification tasks. We will also show that the algorithm has comparable efficiency, and is easily parallelizable.

1 Introduction

1.1 Semi-Supervised Learning

One of the core tasks in machine learning is *supervised learning*. Given labeled training data, a function that maps instances to labels is learned. For many problems, very little labeled training data is available, since such data must often be labeled by humans. However, a large amount of unlabeled data is often available. *Semi-supervised learning* (SSL) is a class of tasks and techniques where both a small amount of labeled training data and a large amount of unlabeled training data are available. Commonly, rather than learning a function that maps arbitrary instances to labels, we instead want to find the labels for the unlabeled data. This is often called *label propagation*. A natural representation of SSL data is a graph constructed based on similarity between instances. Common graph representations are bipartite graphs where documents are connected to their features, and k-nearest neighbor graphs, where documents are connected to other documents based on similarity.

1.2 Text Classification

A common problem in machine learning is *text classification*. Given a text document, we want to assign it to one or more categories. Often, we are given a large set of documents and class labels for a small subset of them (called *seed* documents), and we want to develop an algorithm that can assign class labels to all the documents. This is a natural application of semi-supervised learning.

1.3 Motivation

Recently, a first-order probabilistic language for logical inference called ProPPR has been developed. [11] It converts queries to graphs, and uses a graph-based semi-supervised learning method called *supervised random walks* (SRW) to answer the queries. We are interested in seeing how this method compares to state-of-the-art graph-based semi-supervised learning

methods, such as Gaussian random fields [12] and modified adsorption [9, 10].

1.4 Roadmap

We start by discussing the background of ProPPR and SRW. We also discuss the graph-based SSL methods we will be comparing ProPPR to, Gaussian random fields and modified adsorption. We then review the datasets we will be using: 20 Newsgroups, WebKB, and Freebase. We then explain the experimental setup we will be using to compare the algorithms. Finally, we present the results of our experiments and analysis of the performance of the algorithms.

Overall, ProPPR's graph-based SSL method, SRW, performs better than existing methods on text classification tasks, and comparatively on other graph-based SSL tasks. Additionally, it parallelizes fairly well. Thus, SRW is an effective graph-based SSL method that should be considered for tasks beyond ProPPR.

2 Background

2.1 Personalized PageRank

PageRank [8] is an algorithm developed to rank search engine results. It is based on random walks on a graph with nodes representing web pages and edges representing links between pages, with a chance in each iteration of restarting on a random web page.

Personalized PageRank (PPR) [4] is an extension of PageRank. There is allowed to be biases towards specific nodes in the graph. During an iteration of a random walk, jumps to nodes in the graph occur based on "preference," which is represented with a normalized distribution of weights on the nodes in the graph. Note that personalized PageRank can act as a distance metric in the graph, since nodes are more likely to jump to nodes with higher weights.

2.2 ProPPR

Programming with personalized PageRank (ProPPR) [11] is a first-order probabilistic language for logical inference. Traditionally, queries are answered through *grounding*, by mapping them to propositional representations and then performing inference. This is computationally expensive, since a large database of facts can lead to a large grounding. ProPPR instead approximates "local" grounding by grounding every query with a small, labeled graph independent of database size. Inference is then performed through PPR over a directed graph with feature-vector edges.

ProPPR uses an implementation of *supervised random walks* (SRW), [1] which were first developed as a method to predict links between nodes in a social network graph. Given a graph with feature-vector annotated edges, a PPR process is performed. This procedure is based on minimizing pairwise ranking loss. In ProPPR, this pairwise ranking loss is decomposed into a positive-negative log loss function. This function is then optimized with *stochastic gradient descent* (SGD) rather than the quasi-Newton methods used by Backstrom and Leskovec. SGD was chosen for speed and easy parallelization.

We are interested in seeing if ProPPR's backend (SRW) can be used on its own for label propagation. In particular, we want to see how it compares to state-of-the-art label propagation methods, such as the ones listed below.

2.3 Gaussian random fields

Gaussian random fields (GRF) [12] is one of the earliest label propagation methods. It "prefers" smooth labelings since the score of the label for a node is the weighted average of the scores for that label of the node's neighbors. Additionally, label assignments where two nodes connected by a highly weighted edge but are assigned different labels are penalized.

2.4 Modified adsorption

Modified adsorption (MAD) [9, 10] is a method based on *adsorption*. [2] Adsorption is an algorithm that involves a random walk over a graph where, at each vertex v, we have probability

- p_v^{inj} of stopping and returning (injecting) a predefined vector Y_v,
- p_v^{cont} of continuing the walk to a neighbor of v, randomly chosen with probabilities based on edge weights, and
- p_v^{abnd} of abandoning the walk and returning a zero vector.

MAD is an algorithm that retains most of the properties of the adsorption algorithm, but with a well-defined objective function to minimize.

3 Prior Work

Wang, Mazaitis, and Cohen [11] showed that the ProPPR system as a whole can be used to perform label propagation. They demonstrated that it had comparable performance as the voted perceptron and contrastive divergence algorithms.

Lin and Cohen [6] compared MultiRankWalk, a random-walk based SSL algorithm, to weighted-vote relational neighbor classifiers and Gaussian random fields. They showed that MultiRankWalk outperformed the other algorithms on a number of benchmark network datasets, with very little labeled data.

Talukdar and Pereira [9] compared the graph-based SSL algorithms modified adsorption, adsorption, and Gaussian random fields on a number of graph datasets, including Freebase 1 and Freebase 2. They demonstrated that modified adsorption outperforms other state-of-the-art graph algorithms.

4 Experimental Results

We are interested in seeing how ProPPR's implementation of supervised random walks compares to other graph-based semi-supervised learning algorithms.

4.1 Algorithms

We compare three different algorithms:

1. Gaussian random fields (GRF)
2. Modified adsorption (MAD)
3. ProPPR's supervised random walks (SRW)

We use implementations of GRF (called LP-ZGL) and MAD from the Junto Label Propagation Toolkit [1].

We base our implementation of SRW on the Java implementation of ProPPR [2]. Since the graphs we use are all undirected with a single weight on each edge, we first convert them to directed graphs with a feature vector on each edge.

We generate a new graph for each class by adding a special start node to the instance graph, and adding an edge from the start node to the node for each seed instance for that class. Thus, a random walk in the graph will start with the seed instances, and then proceed to their neighbors, which are the instances most similar to them.

After running ProPPR's SRW algorithm on each of these graphs, each node in each graph will have an associated weight. The weight of a node in a graph represents how highly associated that node is with the class corresponding to that graph. Thus, we can form a ranked list of class labels for each node based on its weights in the class graphs.

Psuedocode for the algorithm can be seen in Algorithm 1.

[1] `https://github.com/parthatalukdar/junto`
[2] `https://github.com/TeamCohen/ProPPR`

Input: Undirected, weighted graph $G = (V, E)$, mapping of seed nodes to labels S, list of labels L
Output: Mapping of nodes to ranked list of labels R
Let $V' = V$, $E = []$;
Create directed, feature-vector weighted graph $G' = (V', E')$;
Add start node s to V';
for $(u, v, w) \in E$ **do**
  Add (u, v) with feature vector $[w, 0]$ to E';
  Add (v, u) with feature vector $[w, 0]$ to E';
end
Let $R = \{v \mapsto [] \mid v \in V\}$;
for $l \in L$ **do**
  Let $G'' = (V'', E'')$ be a copy of G';
  for $(v \mapsto l') \in S$ **do**
    if $l = l'$ **then**
      Add (s, v) with feature vector $[0, 1]$ to E'';
    end
  end
  Run SRW on G'', and let M be the output;
  for $v \in V$ **do**
    Append $(M[v], l)$ to $R[v]$;
  end
end
for $v \in V$ **do**
  Sort in descending order $R[v]$;
end

Algorithm 1: Psuedocode for SRW. Note that we can easily parallelize across the classes.

4.2 Datasets

We use four datasets suitable for graph-based semi-supervised learning: Freebase 1, Freebase 2, 20 Newsgroups, and WebKB. Freebase 1 and Freebase 2 are derived from a *knowledge base*, which is a graph of topics their relations. They are not text classification datasets, but since they have been used to measure performance of other graph-based SSL algorithms [9], we use them to measure performance as well.

Name	**Nodes**	**Edges**	**Classes**
Freebase 1	32970	957076	46
Freebase 2	301638	2310002	384
20NG	79962	2435219	20
WebKB	11969	324254	4

Table 1: Statistics for the dataset graphs

Freebase [7,9] is a collaborative knowledge base that uses information from both open data sets and user contributions. **Freebase 1** is constructed from a small subset of relations from that knowledge base, covering topics from the domains: *astronomy, automotive, biology, book, business, chemistry, comic books, computer, film, food, geography, location, people, religion, spaceflight, tennis, travel,* and *wine*. **Freebase 2** is a larger subset of relations constructed in the same fashion and covering the same domains.

20 Newsgroups (20NG) [5] is a collection of 18,774 newsgroup documents (emails). There are 61,188 words, and 20 different classes, one for each newsgroup: *comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, misc.forsale, talk.politics.misc, talk.politics.guns, talk.politics.mideast, talk.religion.misc, alt.atheism,* and *soc.religion.christian*. A subset of the graph for 20 Newsgroups can be seen in Figure 1. Emails are connected to words that they contain, with edges weighted by word count.

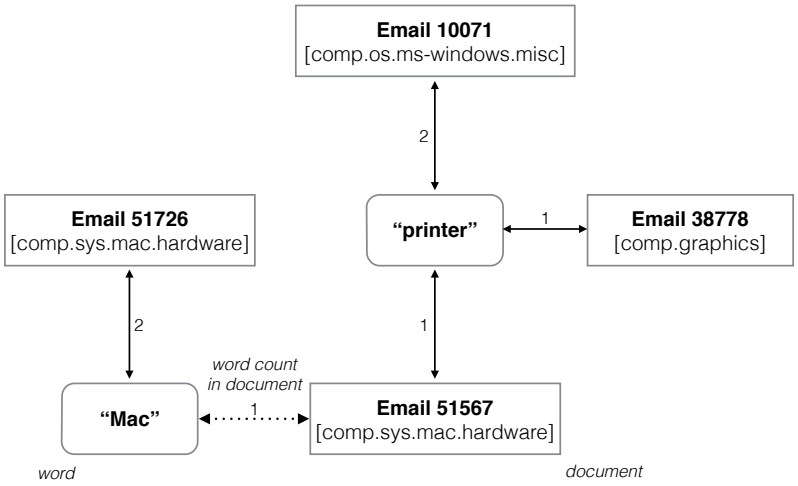


Figure 1: Subset of 20 Newsgroups graph

WebKB [3] is a set of 4199 web pages collected by the World Wide Knowledge Base project, from computer science departments of four universities (Cornell, Texas, Washington, Wisconsin) in 1997. We are using a subset of the pages with four classes: *student, faculty, course,* and *project*.

All datasets are in the form of undirected graphs with weighted edges.

4.3 Environment

Unless otherwise noted, all implementations are given access to 8 GB of memory and are run single-threaded.

4.4 Seed selection

We choose seeds in two different ways: entirely randomly, and based on highest out-degree. Randomly choosing seeds is reasonable, since, for real datasets, often arbitrary instances are labeled. Nodes with highest out-degree are those that are "influential," since propagation

through those nodes has the potential to affect the greatest number of other nodes. And often, for real datasets, the most influential instances will be those that we have the labels for.

We choose 1%, 5% and 10% of the instances as seeds. These represent three typical small proportions of labeled data compared to unlabeled data.

4.5 Evaluation

We evaluate the performance based on two metrics:

Mean reciprocal rank (MRR) is a statistical measure that represents how highly the correct classes are ranked for each node's ranked list of class labels:

$$\text{MRR} = \frac{1}{|Q|} \sum_{v \in Q} \frac{1}{\text{rank}_v}$$

where Q is the set of test nodes, and rank_v is the rank of the correct class label in the list of classes assigned to the node v.

F_1 *score* is a measure of accuracy based on the precision p and recall r:

$$F_1 = 2 \cdot \frac{p \cdot r}{p + r}$$

A node is considered positive if its highest ranked class is the correct class.

4.6 Parallelism

To test parallel performance, we run SRW on the two largest datasets, Freebase 2 and 20 Newsgroups, with varying numbers of threads. Parallelism is achieved by performing the run of SRW on each label's graph in a separate thread.

4.7 Results

SRW performs substantially better than GRF and MAD on the two text classification datasets, 20NG and WebKB, for all proportions of seeds and methods of seed selection, except on MRR for random seeds. SRW has a better F_1 score but worse MRR on the other two graph datasets, $Freebase1$ and $Freebase2$. Thus, in all cases, SRW has comparable performance, and it outperforms the other two algorithms on the text classification datasets.

4.8 Freebase 1

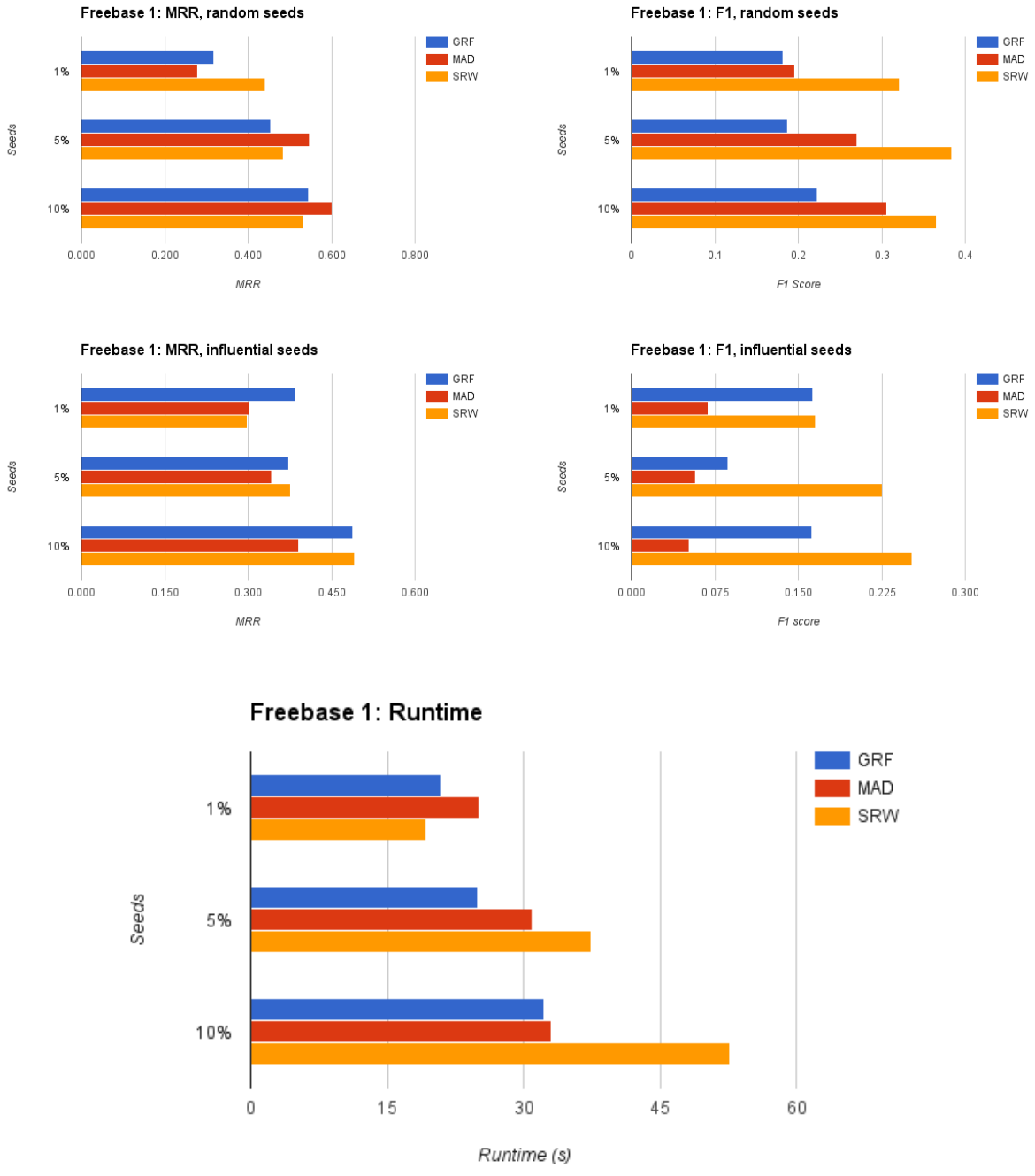


Figure 2: Freebase 1

4.9 Freebase 2

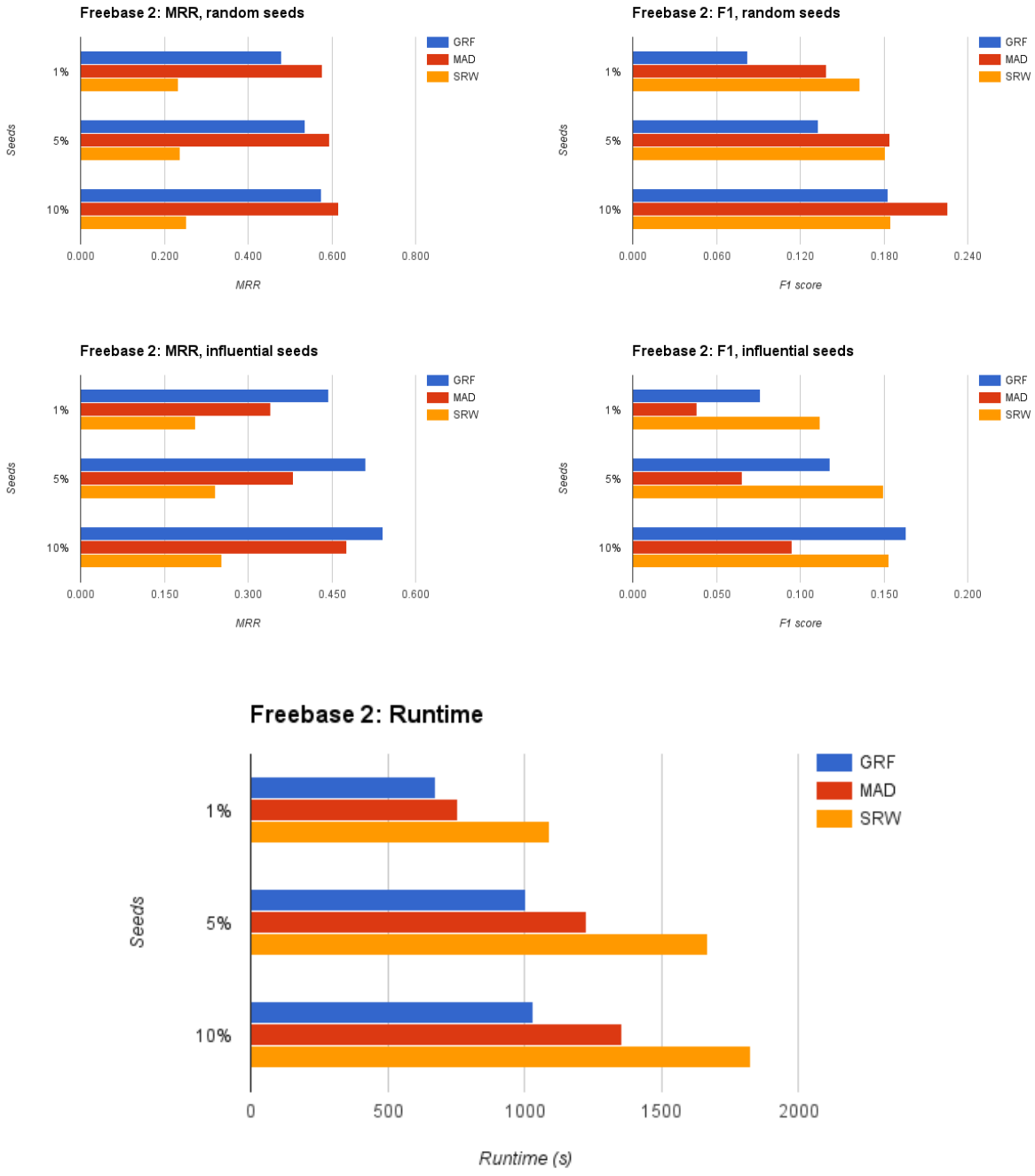


Figure 3: Freebase 2

4.10 20 Newsgroups

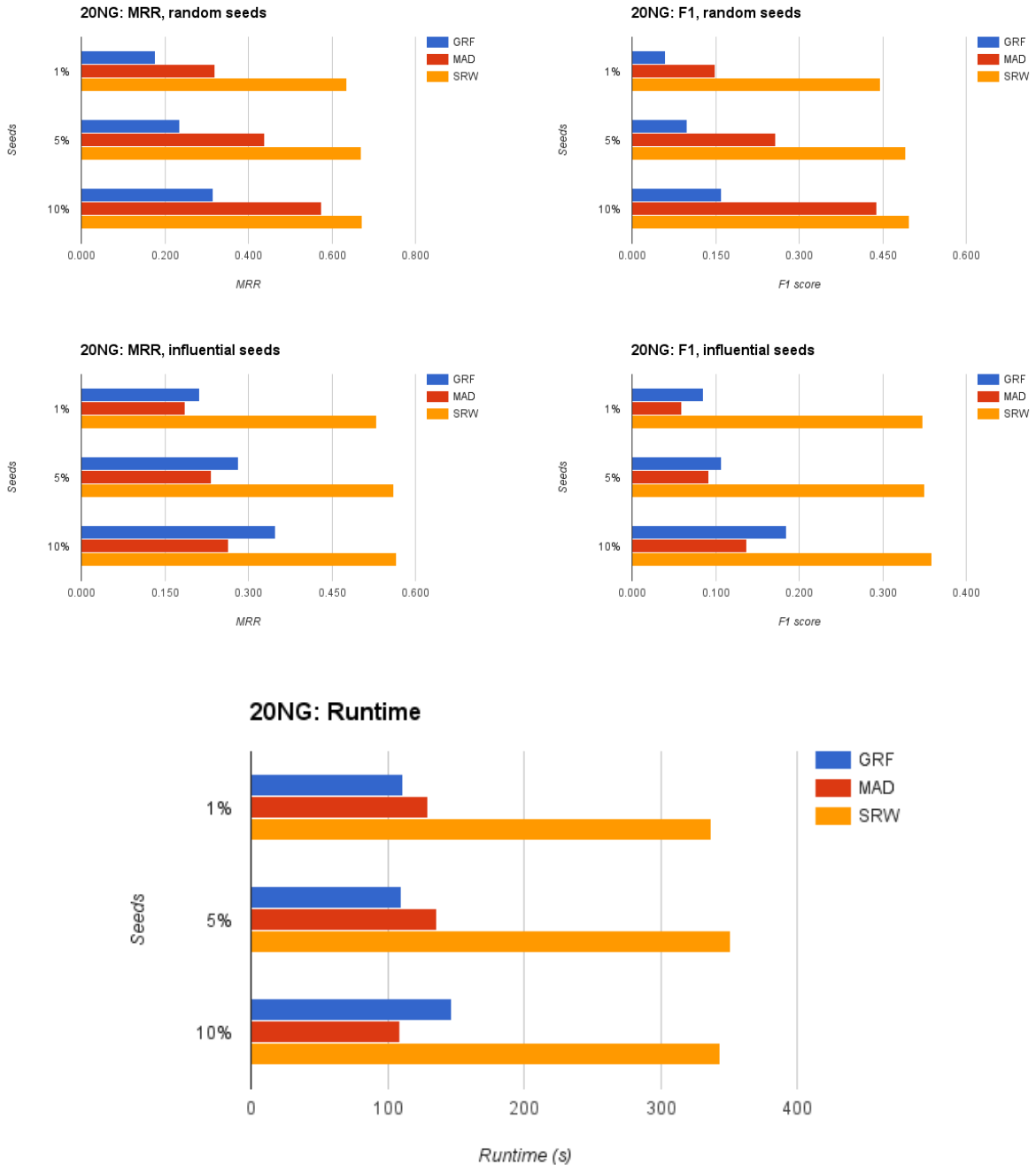
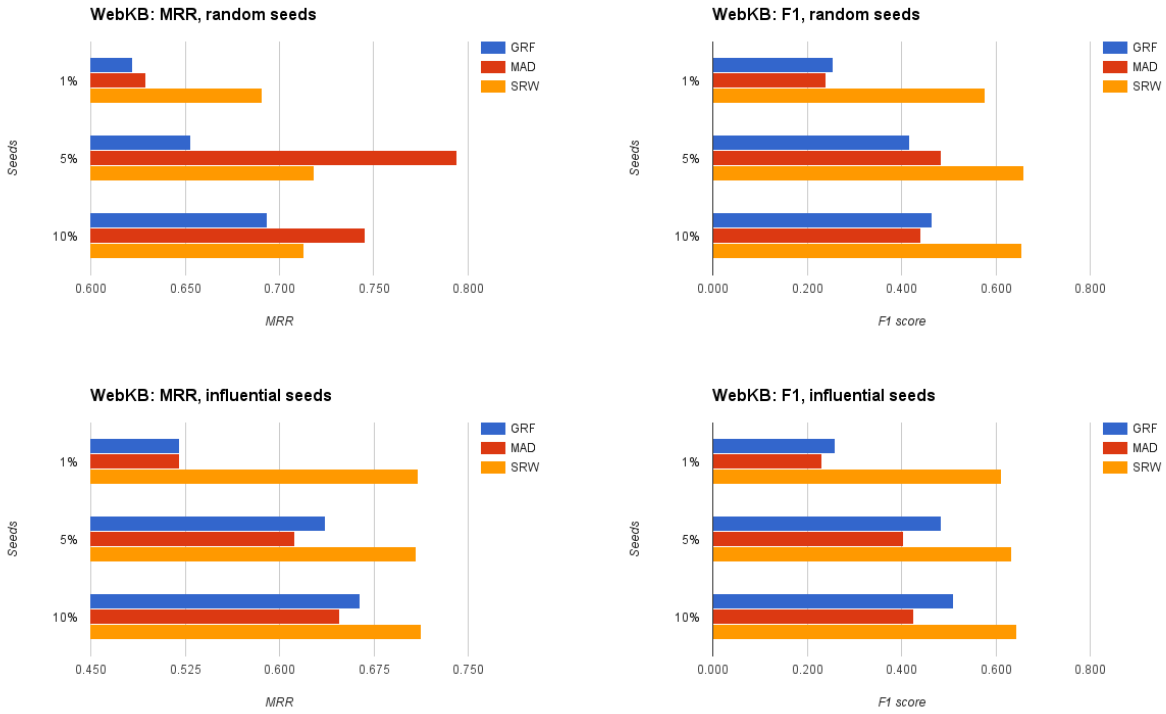


Figure 4: 20NG

4.11 WebKB



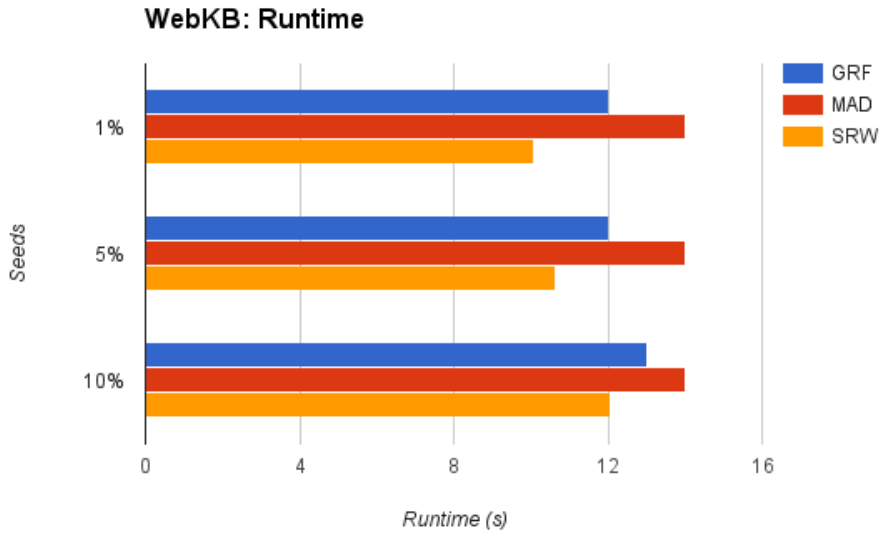


Figure 5: WebKB

5 Analysis

5.1 Runtime

Overall, the runtime of SRW is comparable to that of MAD and GRF. Though it is slower on 20NG and Freebase 2, the difference in runtime is a constant factor that does not appear to depend on number of instances.

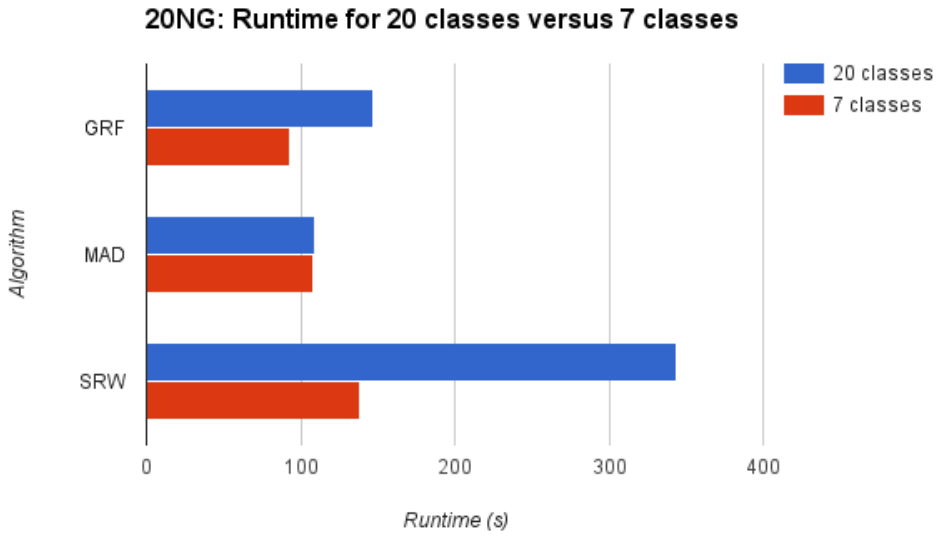


Figure 6: 20NG: 20 classes versus 7 classes

Reducing the number of classes without changing then number of instances does affect the runtime of SRW significantly. To demonstrate this, we generated a new graph by combining the 20 classes in 20NG into seven, based on the category each mailing list falls under: *comp.*, rec.*, sci.*, misc.*, talk.*, alt.*,* and *soc.**. In this way, class labels still make sense, since they simply become more general, and the number of instances does not change. As seen in Figure 6, reducing the number of classes leads to a proportional decrease in runtime for SRW, but not for GRF or MAD. This makes sense, since the algorithm must generate a graph and run a supervised random walk for each class in the dataset.

5.2 Parallelism

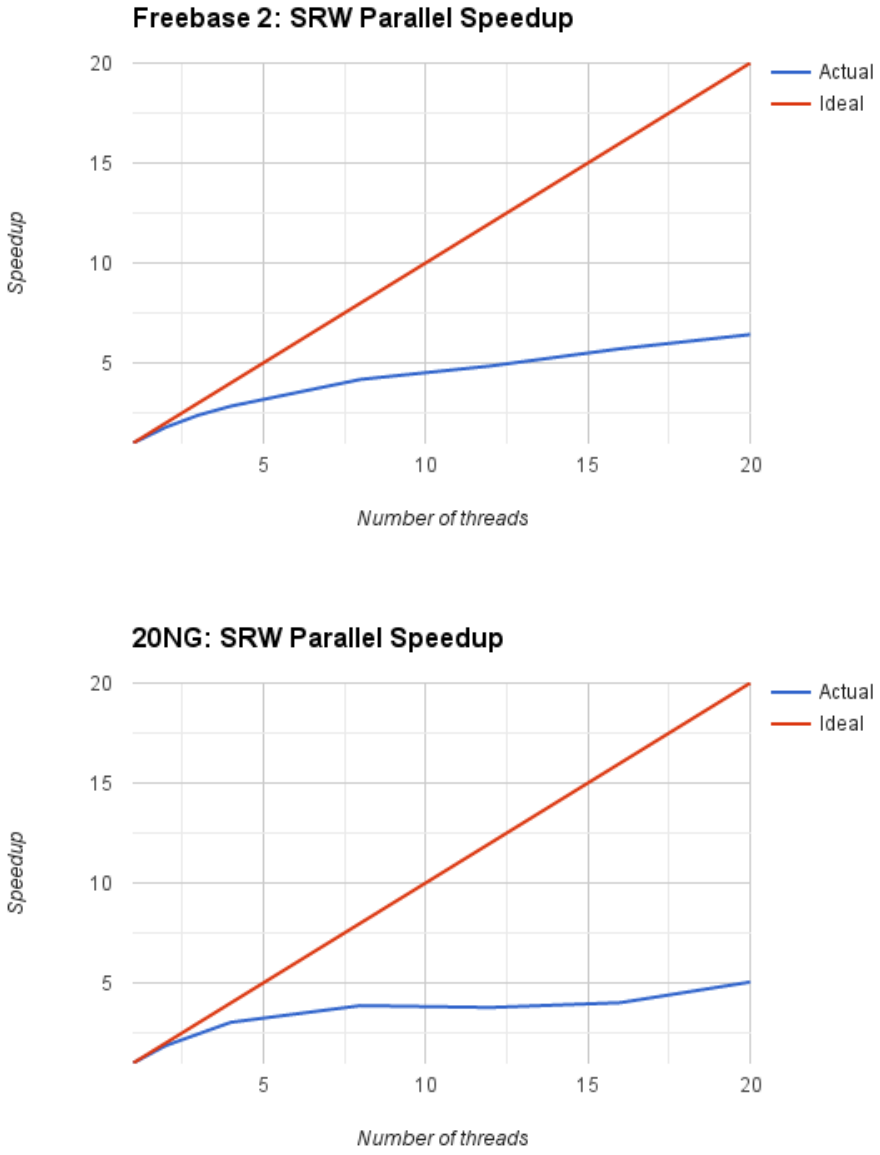


Figure 7: SRW speedup from using threads on 20NG and Freebase 2

SRW benefits from parallelization up to 4 threads, and then speedup falls off significantly. This can be attributed to substantial communication costs. A copy of the instance graph must be made for each thread, and threads must send the weights for all the nodes in the graph after completion. Additionally, generating the ranked list of class labels for each node

is inherently highly sequential. Still, reasonable speedup can be achieved through use of a small number of threads.

6 Conclusion

Overall, SRW has been shown to be not just an effective component of the ProPPR solver system, but also as a standalone graph-based SSL method for text classification. It has better performance than existing state-of-the-art graph-based SSL methods on text classification problems, and comparable performance on more general graph-based datasets. Additionally, its speed is comparable or slightly slower, and it can be easily parallelized for significant speedup.

7 Future Work

The runtime of supervised random walks is directly proportional to number of classes, since it runs the algorithm on a graph generated for each class. Since the same is not true for Gaussian random fields and modified adsorption, we could investigate adapting parts of those algorithms to improve performance of SRW on datasets with many classes. Additionally, speedup from parallelization falls off at a certain point, so we could investigate reducing communication costs and sequential work to improve parallel performance.

8 References

[1] L. Backstrom and J. Leskovec, *Supervised random walks: predicting and recommending links in social networks*, in Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 635–644.

[2] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly, *Video suggestion and discovery for youtube: taking*

random walks through the view graph, in Proceedings of the 17th international conference on World Wide Web, ACM, 2008, pp. 895–904.

[3] A. Cardoso-Cachopo, *Improving Methods for Single-label Text Categorization*. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

[4] T. Haveliwala, S. Kamvar, and G. Jeh, *An analytical comparison of approaches to personalizing pagerank*, (2003).

[5] K. Lang, *Newsweeder: Learning to filter netnews*, in Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 331–339.

[6] F. Lin and W. W. Cohen, *Semi-supervised classification of network data using very few labels*, in Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on, IEEE, 2010, pp. 192–199.

[7] Metaweb Technologies, *Freebase data dumps*.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd, *The pagerank citation ranking: Bringing order to the web.*, (1999).

[9] P. Talukdar and F. Pereira, *Experiments in graph-based semi-supervised learning methods for class-instance acquisition*, in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 1473–1481.

[10] P. P. Talukdar and K. Crammer, *New regularized algorithms for transductive learning*, in Machine Learning and Knowledge Discovery in Databases, Springer, 2009, pp. 442–457.

[11] W. Y. Wang, K. Mazaitis, and W. W. Cohen, *Programming with personalized pagerank: a locally groundable first-order probabilistic logic*, in Proceedings of the 22nd

ACM international conference on Conference on information & knowledge management, ACM, 2013, pp. 2129–2138.

[12] X. Zhu, J. Lafferty, and Z. Ghahramani, *Combining active learning and semi-supervised learning using gaussian fields and harmonic functions*, in ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining, 2003, pp. 58–65.