Solving Electrical Networks to incorporate Supervision in Random Walks

Mrinmaya Sachan
Carnegie Mellon University
mrinmays@cs.cmu.edu

Dirk Hovy
Information Sciences Institute
dirkh@isi.edu

Eduard Hovy
Carnegie Mellon University
hovy@cmu.edu

ABSTRACT

Random walks is one of the most popular ideas in computer science. A critical assumption in random walks is that the probability of the walk being at a given vertex at a time instance converges to a limit independent of the start state. While this makes it computationally efficient to solve, it limits their use to incorporate label information. In this paper, we exploit the connection between Random Walks and Electrical Networks to incorporate label information in classification, ranking, and seed expansion.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

Random Walks, Electrical Networks, Ranking, Classification

1. INTRODUCTION

Random walks are widely used in web[2], search, clustering, sentiment-analysis and classification [3], where data can be naturally represented in the form of graphs. The basic idea in random-walks is voting. When one vertex links to another, it casts a vote for the other vertex. The higher the number of votes that are cast for a vertex, the higher is the importance of the vertex. Moreover, the importance of the vertex casting a vote determines how important the vote itself is. However, the power of random walks is limited by the fact that they do not directly incorporate labeled data. If labeled data is readily available, we'd want to include it. For example, it is easy to obtain a list of words with positive/negative polarity for sentiment detection, a set of initial seeds for a seed-expansion task, or a partial ranking/preferences on a ranking task.

In this paper, we draw upon the well-studied connection between random walks and electrical networks to include labeled data in a principled manner for both classification and ranking tasks. It efficiently utilizes a combination of a) the weight of a node (based on its agreement with the labels) and b) the weight of the edge connecting to it to vote on its neighbors. Our work is related to work on label propagation and semi-supervised learning by [3]. However, our approach differs as it models graphs with labels using the same methodology as random walks. The main contributions of this paper are: we show how to directly exploit labeled data in random walks, apply it to various tasks (classification, ranking and seed expansion), and show significant improvements over standard random-walk-based techniques.

2. THE ALGORITHM

We will first describe the connection between random walks and electrical networks [1]. Recall that in the steady state of random walks, the probability of being at any node y is the sum over the probability of being at each node $x \in V\backslash\{y\}$ and taking the transition from x to y. In terms of matrices, this is the same as power iteration. Now, let us associate with every graph $G(V, E, W)$, an electrical network whose properties will closely resemble those of a random walk on the original graph. The electrical network is constructed by introducing a resistor between every pair of nodes in the graph that share an edge ($(x,y) \in E$) with resistance inversely proportional to the weight of the edge (conductance $C_{x,y} \propto W_{x,y}$). Let P be the row-normalized adjacency matrix (or Transition Matrix) of the graph where P_{xy}, the $(x, y)^{th}$ entry is the probability of a transition from node x to y. Now, consider two vertices a and b. Let the voltage $v(b) = 0$. Next, attach a battery of 1 volt across a and b so that the voltage $v(a) = 1$. Fixing the voltages at these two nodes, we compute the voltage at other nodes in the network ($v = Pv$). An important theoretical result of use here is that the voltage at an arbitrary vertex x is given by the probability of reaching a from x before reaching b [1]. Hence, to impose polarity in random walks, we will just define the boundary conditions (v_a, v_b) in the electrical network and do a power iteration ($v = Pv$) to compute v, holding the voltages at labeled nodes fixed. The newly computed voltage at any arbitrary node will give its label affiliation (probability that the node is closer to label a than label b).

Our algorithm exploits this relationship between electrical networks and random walks and extends to multiple classes. Given data D=$\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in R^d$ are d-dimensional feature representations of the data and $y_i \in \{0, 1 \ldots, C\}$ are class labels known over a subset of nodes $Z \in \{0, 1\}^V$, we construct a (complete) graph $G = (V, E, W)$ over the data points, $V = \{1 \ldots n\}$, $E = \{V \times V\}$ and $W = \{w_{ij} | w_{ij} = Sim(x_i, x_j)\}$ where Sim is some similarity metric. Then, we hold voltages at all nodes Z equal to their class labels, $v(i) = y_i \forall i \in Z$ and solve the network. We label a node $i \in \{V\backslash Z\}$ as the integer nearest to $v(i)$.

The voltage computation proceeds in a manner similar to power iteration (see Algorithm 1). In matrix notation, the power iteration when nodes Z are clamped translates to $v = (Pv) \circ Z + v \circ (1 - Z)$ where $\circ$ represents the Hadamard

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

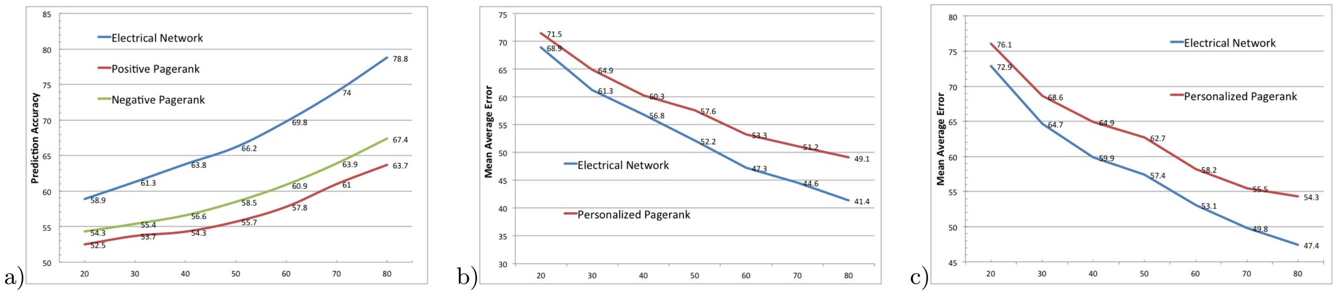

Figure 1: (a) Accuracy on IMDB review Classification Task, (b) MAE on Amazon review Ranking Task and (c) MAE on the 20 newsgroup Seed Expansion Task as percentage of labeled data is varied.

(element-wise) product of matrices and $(1 - Z)$ is a vector where all elements of Z are switched.

Algorithm 1: Electrical Network Labeler(P, Y, Z)

```
Initialize: Labels Y for all nodes, using labels for Z and
randomly for $V \backslash \{Z\}$, $Y_{new}$ and $\triangle Y \in$ random($[0, C]^V$);
while ($\| \triangle Y \|_2 > 0$) do
    $Y_{new}$ = (PY)∘Z + Y∘(1-Z);
    $\triangle Y = Y_{new} - Y$;
    $Y = Y_{new}$;
end
```

3. EVALUATION

We evaluate our approach on three real-world data sets, one each on classification, ranking, and seed expansion tasks.

First, we choose the problem of sentiment polarity detection on an IMDB movie review corpus.[1] The dataset contains 1000 positive and 1000 negatively tagged reviews. We build a complete graph on the reviews (similarity between reviews is the cosine-similarity of the tf-idf representations of the reviews after stop word removal). We split the data randomly into training and test sets (via 5-fold cross validation) and label the positive and negative instances as +1 and 0 Volt, respectively. For comparison, we employ a variant of a random-walk based technique which comes closest to exploiting labeled data, called personalized page rank (PPR) [2]. PPR, besides transitioning to its neighbors according to the transition matrix, teleports to a random node according to a pre-defined probability distribution over the nodes called the "preference vector". We construct two baselines: by instantiating the preference vector either as a uniform distribution over the positive instances or over the negative instances in the training set. In Table 1, we can see significant improvements in performance of the electrical network classifier in terms of accuracy, precision, recall and F1-score over the two baselines. Figure 1(a) plots the accuracy of the three classifiers as the proportion of labels in the data is varied. Here, we see that the electrical network has a greater slope and hence utilizes labeled data better. We claim that this is due to its ability to exploit labeled disagreement which random walk approaches like PPR cannot.

Next, we chose a dataset with fine-grained rankings[2]: product reviews taken from Amazon from four product domains. Each review also has ratings (scale 1 to 5). For each product domain, we construct a graph over the reviews in a similar manner as before. The voltages of all the training reviews are clamped to their ratings. Here, the task is to determine the rating for a random test review. We claim that this equals the voltage at the test review node. We use mean absolute error (MAE) as an evaluation metric and pagerank as the baseline. PPR is constructed by instantiating the preference vector using the rankings (ratings) known during training. Figure 1(b) shows how MAE in one of the domains (books) decreases faster for the electrical networks than for PPR as the percentage of labeled data is varied.

Table 1: Classification Results on the IMDB dataset

Classifier	Acc.	Precision	Recall	F1-Score
Electr. Network	**0.788**	**0.764**	**0.754**	**0.759**
Pos. Pagerank	0.637	0.631	0.658	0.644
Neg. Pagerank	0.674	0.684	0.693	0.689

Finally, we choose the 20-newsgroups dataset comprising 18,846 documents organized into 20 different newsgroups, each corresponding to a different topic (vocabulary size = 26k). Here, we manually create a small seed list (25 words) for each newsgroup. Our task is to expand the seed lists. This is done by first constructing a complete graph over all the words in the dataset using the similarity between words to be the cosine similarity between their corresponding SENNA embeddings[3]. To expand the seed list for a particular newsgroup, we assign all seed words to +1 Volt and all remaining seed words to 0 Volt and create a ranking over all nodes as before. Since it is impossible to obtain a ground truth here, we approximate it by ranking the words for each newsgroup by their tf-idf scores in the group. Figure 1(c) shows an improvement over PPR and a drop in MAE for the electrical network as labels increase.

4. CONCLUSION

Our experiments establish the efficacy of electrical networks as a way to provide supervision in random walks. When data can be naturally represented in form of graphs and some labeled data is available, electrical network solving can be an effective way to do semi-supervised learning.

5. REFERENCES

[1] P. G. Doyle and J. L. Snell. *Random walks and electric networks.* Mathematical Association of America, 1984.

[2] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to web, 1999.

[3] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.

[1] www.cs.cornell.edu/people/pabo/movie-review-data/

[2] www.cs.jhu.edu/~mdredze/datasets/sentiment/

[3] ronan.collobert.com/senna/