Mixtures-of-Clusterings by Boosting

Jonathan Chang
Electrical Engineering
Engineering Quadrangle
Princeton, NJ 08544
jcone@princeton.edu

David M. Blei
Computer Science
35 Olden St.
Princeton, NJ 08544
blei@cs.princeton.edu

ABSTRACT

Network data that express connections between nodes in a graph is becoming increasingly ubiquitous. Clustering and latent variable analysis have become invaluable tools for understanding network structure, finding communities, and making predictions. In this paper we present instead a mixture-of-clusterings model which has a simple interpretation. We employ AdaBoost to quickly and effectively train our model and demonstrate that our model can make accurate predictions about new connections between nodes.

1. INTRODUCTION

Network data, such as citation networks of documents, hyperlinked networks of web pages, and social networks of friends, are becoming pervasive in modern machine learning applications. Analyzing network data provides useful predictive models [4], pointing social network members towards new friends, scientific papers towards relevant citations, and web pages towards other related pages.

While techniques using graph cuts [12, 9] have been extensively studied for extracting community structure, much recent research in network data has focused on latent variable models of link structure, models which decompose a network according to hidden patterns of connections between its nodes. Single-membership models such as K-means and more recent latent variable models [7, 6, 8] posit that each node belongs to exactly one latent cluster. Mixed-membership models [2, 1] are powerful extensions to single-membership models which associate a K-dimensional multinomial parameter with each node in the network endowing the node with a partial membership in all K clusters.

In contrast, in this paper we present a *mixture-of-clusterings* model. Our model consists of a mixture of T single-membership clusterings, $x_{1:N}^{(1)} \dots x_{1:N}^{(T)}$, governed by mixture proportions $\alpha^{(1)} \dots \alpha^{(T)}$. In order to learn these parameters of the model, we employ AdaBoost [13]. AdaBoost is an instance of boosting, a class of commonly used techniques for learning a classifier which combines the predictions of several simpler classifiers known as weak hypotheses. Boosting was originally designed for the fully-supervised setting; extensions to unsupervised settings have been proposed [14], though these are not easily applicable here.

Similar in spirit to the technique we propose here is BoostCluster [10] which uses boosting to return a single-membership clustering instead of the mixture-of-clusterings our model produces. Work done by [5] boosts an EM mixture classifier but their focus is on learning classifiers for points. The observation of this paper, however, is that clustering is in essence a classification problem not on the nodes of a graph but rather its adjacency matrix. We show in the sequel that combining simple clusterers which each attempt to capture some aspect of the connectivity structure leads to a model which can be quickly and simply trained, and which can make accurate predictions on new, unseen connections.

Algorithm 1 The mixtures-of-clusterings algorithm

```
Require: Adjacency matrix $\mathbf{A}$ of size $N \times N$.
Require: Number of rounds $T$.
Require: Number of clusters $K$.
  Initialize weights $D_{ij}^{(1)} \leftarrow \frac{1}{N^2} \quad \forall i, j$.
  Initialize prediction matrix $H_{ij}^{(1)} \leftarrow 0 \quad \forall i, j$.
  for all $t$ such that $1 \leq t \leq T$ do
    Use the weak learner to find $x_{1:N}^{(t)} \in \{1 \dots K\}^N$ which optimizes Equation 1.
    Compute the error $\epsilon^{(t)} = \sum_{i,j} D_{ij}^{(t)} \mathbb{1}(x_i^{(t)} \neq x_j^{(t)})$.
    Compute hypothesis weight $\alpha^{(t)} = \frac{1}{2} \ln(\frac{1-\epsilon^{(t)}}{\epsilon^{(t)}})$.
    Update sample weights $D_{ij}^{(t+1)} \leftarrow D_{ij}^{(t)} \exp(-\alpha^{(t)} \mathbb{1}^*(x_i^{(t)} = x_j^{(t)}))$.
    Compute sample weight normalizer $Z^{(t+1)} \leftarrow \sum_{ij} D_{ij}^{(t+1)}$.
    Normalize sample weights $D^{(t+1)} \leftarrow D^{(t+1)} / Z^{(t+1)}$.
    Compute the prediction matrix $H_{ij}^{(t+1)} = H_{ij}^{(t)} + \alpha^{(t)} \mathbb{1}^*(x_i^{(t)} = x_j^{(t)})$.
  end for
  return $H^{(T+1)}$.
```

2. LEARNING THE PARAMETERS

In this section, we present our algorithm for learning the parameters of the mixture of clusterings. Our algorithm is based on AdaBoost. Intuitively, AdaBoost defines a distribution over input labeled examples and then seeks a weak classifier trained on those examples. The algorithm then adjusts the distribution based on the performance of the weak classifier and repeats the procedure for T rounds. Finally, all T weak classifiers are combined to form the final classifier.

Our approach treats the adjacency structure of the network as the labels to be predicted. In each round t of boosting, the algorithm defines a distribution $D_{ij}^{(t)}$ over the edges of the network. Now let $\mathbf{A}$ be the adjacency matrix associated with the network; the entries A_{ij} of the matrix are assumed to be $+1$ is there is a connection between nodes i

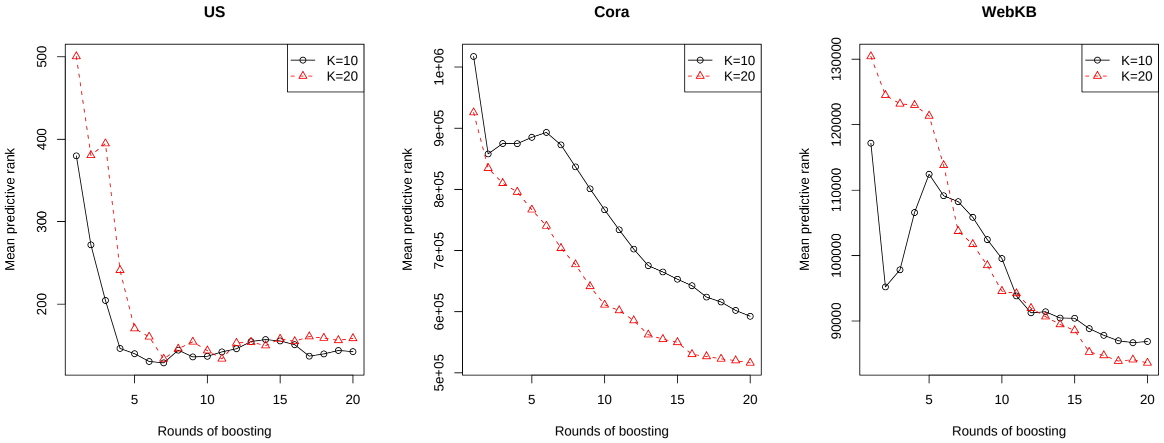

Figure 1: Mean predictive rank for each of the data sets as a function of the number of rounds of boosting. Lower is better. Each plot shows the results for two different values of the number of clusters, K, 10 (black) and 20 (red). The technique significantly improves upon both random guessing and one iteration of clustering.

and j and -1 otherwise. The objective of the weak learner then is to find a cluster assignment x_i for each node i such that the following objective is maximized:

$$\mathcal{L} = \sum_{i,j} A_{ij} D_{ij}^{(t)} \mathbb{1}^*(x_i^{(t)} = x_j^{(t)}), \qquad (1)$$

where $\mathbb{1}^*$ returns $+1$ if its argument is true, and -1 otherwise.

While the algorithm can be used with any procedure which improves the objective $\mathcal{L}$, here we employ a simple greedy algorithm. Starting with a random initial clustering of the nodes, we iteratively select the cluster for each node which maximizes the objective given the current assignment of all the other nodes. The full algorithm is given in Algorithm 1.

3. EXPERIMENTS

We apply the proposed procedure to the adjacency matrix of US states, the Cora [11] data set consisting of 2708 computer science research papers whose connections are determined by citation, and the WebKB [3] data set of webpages whose connections are defined by hyperlinks.

For each of the data sets, we randomly hold out 10% of the links and then apply the parameter estimation procedure described in Section 2. After each iteration of the inner loop of Algorithm 1, we compute the mean predictive rank of the held-out links (predictive ranks are computed by ranking all edges according to their margin). We repeated the process for two choices of the number of latent clusters K: 10 and 20.

The results of these experiments are shown in Figure 1 which plots the mean predictive rank against the number of rounds of boosting. Lower ranks are better. Our proposed technique significantly outperforms both randomly guessing and single-membership clustering (i.e., one round of boosting). Further, the overall procedure is resistant to overfitting. The procedure is easy to implement and quick to train and in future work we hope to extend the objective function to model node attributes as well as network structure.

4. REFERENCES

[1] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *URL http://arxiv. org/abs/0705.4485. Manuscript under review*, 2007.

[2] J. Chang and D. M. Blei. Relational topic models for document networks. *Artificial Intelligence and Statistics*, 2009.

[3] M. Craven, D. DiPasquo, D. Freitag, and A. McCallum. Learning to extract symbolic knowledge from the world wide web. *Proc. AAAI*, 1998.

[4] L. Getoor and C. Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 2005.

[5] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. *Proceedings of the twenty-first international conference on machine learning*, 2004.

[6] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. Dec 2007.

[7] J. Hofman and C. Wiggins. A Bayesian approach to network modularity. *eprint arXiv: 0709.3512*, 2007.

[8] C. Kemp, T. Griffiths, and J. Tenenbaum. Discovering latent classes in relational data. *MIT AI Memo 2004-019*, 2004.

[9] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. *International World Wide Web Conference*, Jan 2008.

[10] Y. Liu, R. Jin, and A. Jain. Boostcluster: boosting clustering by pairwise constraints. *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug 2007.

[11] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 2000.

[12] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23), 2006.

[13] R. Schapire. The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification*, 2003.

[14] M. Welling, R. Zemel, and G. Hinton. Self supervised boosting. *Advances in Neural Information Processing Systems*, pages 681–688, 2003.