Bootstrapped Graph Diffusions: Exposing the Power of Nonlinearity

Extended Abstract

Eliav Buchnik
Tel Aviv University & Google Research
Tel Aviv, Israel
eliavbuh@gmail.com

Edith Cohen
Google Research & Tel Aviv University
CA, USA
edith@cohenwang.com

ABSTRACT

Graph-based semi-supervised learning (SSL) algorithms predict labels for all nodes based on provided labels of a small set of seed nodes. Classic methods capture the graph structure through some underlying diffusion process that propagates through the graph edges. Spectral diffusion, which includes personalized page rank and label propagation, propagates through random walks. Social diffusion propagates through shortest paths. These diffusions are linear in the sense of not distinguishing between contributions of few “strong” relations or many “weak” relations.

Recent methods such as node embeddings and graph convolutional networks (GCN) attained significant gains in quality for SSL tasks. These methods vary on how the graph structure, seed label information, and other features are used, but do share a common thread of nonlinearity that suppresses weak relations and re-enforces stronger ones.

Aiming for quality gain with more scalable methods, we revisit classic linear diffusion methods and place them in a self-training framework. The resulting bootstrapped diffusions are nonlinear in that they re-enforce stronger relations, as with the more complex methods. Surprisingly, we observe that SSL with bootstrapped diffusions not only significantly improves over the respective non-bootstrapped baselines but also outperform state-of-the-art SSL methods. Moreover, since the self-training wrapper retains the scalability of the base method, we obtain both higher quality and better scalability.

CCS CONCEPTS

- Mathematics of computing → Graph algorithms; Bootstrapping.
- Theory of computation → Semi-supervised learning.

KEYWORDS

Graph-based semi-supervised learning; bootstrapping; label propagation

The full version of the paper is available in [7].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS ‘18 Abstracts, June 18–22, 2018, Irvine, CA, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5846-0/18/06.
https://doi.org/10.1145/3219617.3219621

1 SUMMARY

Graph data is prevalent and models entities (nodes) and the strong interactions between them (edges). The source of these graphs can naturally come with the provided interactions (social networks, views, likes, purchases, messages, links) or can be derived from metric data (embedded entities) by retaining only edges that correspond to closest neighbors.

Often, label information is available only for a small set of seed entities \((x_j, y_j) j \leq n_f\) and we are interested in learning the labels of all other entities \(x_i\) for \(i \in (n_f, n_f + n_u)\). See e.g. the surveys [8, 34]. The underlying assumption here is that similarity derived from the graph structure is indicative of similarity of labels. Learning algorithms can use the graph structure and seed labels in isolation or be combined with other available node features.

Classic methods for semi-supervised learning (SSL) and related fundamental graph mining tasks (clustering, centrality, influence) are based on a diffusion process that propagates from a node or set of nodes through the edges of the graph. The diffusion defines dense fine affinity relations between nodes using the provided sparse set of strong interactions. With SSL, the affinity relation guides the label learning, for example, by a weighted aggregation of seed labels to obtain soft labels.

Popular SSL methods use spectral diffusions [9] (the graph Laplacian, graph cuts [5, 6], and random walks). These methods include label propagation [40] and label propagation using the normalized graph Laplacian [25, 38]. They scale well using Jacobi iterations that roughly translate into repeated averaging over neighboring nodes. The methods are used on massive graphs with billions of edges [31] on distributed platforms [26].

Another class of graph diffusions, which we refer to here as social, were used for social and economic models of centrality [3, 4, 14, 18, 29, 32], influence [13, 17, 19, 23], and similarity [12] of nodes. Social diffusion propagate along shortest-paths (distance diffusion) or reachability searches (reach diffusion). A powerful extension defines a generative model from a graph by randomizing the presence (with reach diffusion) or the length (with distance diffusion) of edges [12, 13, 17, 19, 23] and then works with respective expectations. Social diffusion, inspired by the independent cascade model of [23] and the continuous time (distance-based) model of [12, 13, 17, 19]
was recently adapted to SSL [11]. The algorithms scale very well: For the simpler nearest-seed variant which matches each label to the closest seed node in each simulation of the randomized model we simply use small number of graph (Dijkstra) searches. The use of distance or reachability sketching based on [10] allows for highly scalable label learning also over the sketched affinity matrix.

Both spectral and social diffusion based SSL models scale well even with a large number of labels, using heavy hitter sketches with label propagation [31] or naturally with social diffusion using sketches. We interpret these methods as linear in the sense that the diffusion propagates from seeds through edges without amplifying strong signals or suppressing weak ones.

Recently proposed non-linear learning methods, based on node embeddings and graph convolutional networks, had impressive success in improving the quality of the learned labels. In particular, DeepWalk [30] applied the hugely successful word embedding framework of [27] to embed the graph nodes in a way that preserves the affinity relation defined by co-occurrence frequencies of pairs in short random walks. A softmax applied to inner products of embeddings approximates the frequency of the pair. A supervised learning algorithm is then trained on the embedding vectors and labels of seed nodes. Node2vec [20] refined the approach using hyperparameters that tune the depth and breadth of the random walks. Another method, PLANETOID, used a multi-layer neural network instead of a single softmax layer [36]. With these methods, the lower-dimensional embeddings serve as a "low rank" representation of the affinity matrix and the softmax and neural network introduce non-linearities that emphasize larger inner products. Another successful proposal are Graph convolutional networks (GCN) [2, 16, 21, 24], which are neural networks with layers that follow the graph adjacency structure. Each layer applies a non-linear activation function, most often a sigmoid or ReLU, to the aggregate over neighbors.

This abundance of recent work introduced many new components and often at the same time: Low-rank embeddings, non-linear propagations, learning of weights of hidden layers (GCNs), learning of node weights [28]. These recent methods, however, while demonstrating improved labeling quality, do not scale as well as label propagation and methods based on social graph diffusions.

Our aim in this work was to understand which components contributed to the quality gain by the recent methods and to seek methods that combine the scalability advantage of the simpler diffusion-based methods with state of the art labeling quality.

We explore placing these "linear" diffusions in a self-training (bootstrapping) framework. Self-training is arguably the earliest approach to SSL, dating back five decades to Scudder [33], and extensively studied by the NLP community [1, 35, 37]. The self-training framework can be viewed as a wrapper around a base learning algorithm. The base algorithm takes as input a set of labeled examples and makes predictions with associated margins or confidence scores for other examples. The wrapper applies the base algorithm in steps, where at the end of each step, the highest-confidence predictions are converted to become new labeled examples. Our bootstrapped diffusions retain the high scalability of the base diffusions but provide, by allowing for some non-linear propagation, a richer class of models. In particular, with our base algorithms, "seed" examples have a special role that is not amplified by implied high-confidence predictions. Bootstrapping provides such amplification by promoting high-margin predictions to "seed" roles.

We perform experiments using linear diffusion models and their bootstrapped versions. We use classic Label propagation [40], Label propagation using the normalized Laplacian [38], and nearest-seed, which is the simplest distance diffusion model [11]. We apply a very basic bootstrapping wrapper that works with a fixed fraction of highest-margin predictions in each step. We focus on a multi-class setting, where each node is a member of one class, even though most of the method can be extended to the multi-label setting.

We apply the different methods to benchmark data and seed sets used and made available by previous work [24, 36]. In particular, we use social, citation, and knowledge graph data sets. We compare the quality of the learned labels to state of the art baselines, including DeepWalk [30], node2vec [20], PLANETOID [36], and GCNs [24]. We also perform more elaborate experiments on additional data and seed sets and on the well-studied planted partition (stochastic block) model [15] which is often used to understand the performance of clustering and community detection algorithms.

Our main focus is the quality of learning from the graph structure alone using diffused seed node labels. We observe that bootstrapped diffusions consistently improved the quality, by 1% to 12%, over the base diffusion, both for spectral and social and across types of graphs. The most surprising outcome was that prediction quality on benchmark data exceeded that of all recent non-linear methods. The use of additional available node (or edge) features can significantly increase labeling quality. There are multiple ways to integrate such features together with the graph structure in the learning algorithm. We study the direct use of the raw provided node features and the use of smoothed features obtained through a graph diffusion. In both cases, a simple supervised learning algorithm is then trained on the (or diffused) feature vectors and class labels of seed nodes. We applied this method with and without bootstrapping. We observed that both diffusion and bootstrapping significantly enhanced performance. Furthermore, our results dominated those reported (with the use of node features) by the state of the art baselines PLANETOID [36] and GCNs [24]. In particular, GCNs lend themselves to a direct comparison as they can be viewed as a feature diffusion with non-linearities applied after each set of edge traversals and node/layer weight tuned through back propagations. It is interesting that we obtained comparable or better results using bootstrapped linear models and without backprop training.

References

[1] S. Abney. 2004. Understanding the Yarowsky Algorithm. Comput. Linguist. 30, 3 (Sept. 2004).
[2] J. Atwood and D. Towsley. 2016. Diffusion-Convolutional Neural Networks. In NIPS.
[3] A. Bavelas. 1948. A mathematical model for small group structures. Human Organization 7 (1948), 16–30.
[4] F. Bloch and M. O. Jackson. 2007. The formation of networks with transfers among players. Journal of Economic Theory 133, 1 (2007), 83–110.
[5] A. Blum and S. Chawla. 2001. Learning from Labeled and Unlabeled Data Using Graph Mincuts. In ICML.
[6] A. Blum, J. Lafferty, M. R. Rwehbangira, and R. Reddy. 2004. Semi-supervised Learning Using Randomized Mincuts. In ICML.
[7] E. Buchnik and E. Cohen. 2018. Bootstrapped Graph Diffusions: Exposing the Power of Nonlinearity. Proc. ACM Meas. Anal. Comput. Syst. 2, 1 (April 2018).
[8] O. Chapelle, B. Schölkopf, and A. Zien. 2006. Semi-supervised learning. MIT Press.
[9] F. R. K. Chung. 1997. Spectral Graph Theory. American Mathematical Society.
