Semi-supervised Classification in Graphs using Bounded Random Walks

Semi-supervised learning, large graphs, betweenness measure, passage times

Jérôme Callut JEROME.CALLUT@UCLOUVAIN.BE
Kevin Fançoisse KEVIN.FRANCOISSE@UCLOUVAIN.BE
Marco Saerens MARCO.SAERENS@UCLOUVAIN.BE
UCL Machine Learning Group (MLG)
Louvain School of Management, IAG,
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

Pierre Dupont PIERRE.DUPONT@UCLOUVAIN.BE
UCL Machine Learning Group (MLG)
Department of Computing Science and Engineering, INGI,
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

Abstract

This paper describes a novel technique, called $\mathcal{D}$-walks, to tackle semi-supervised classification problems in large graphs. We introduce here a betweenness measure based on passage times during random walks of bounded lengths in the input graph. The class of unlabeled nodes is predicted by maximizing the betweenness with labeled nodes. This approach can deal with directed or undirected graphs with a linear time complexity with respect to the number of edges and the maximum walk length considered. Preliminary experiments on the CORA database show that $\mathcal{D}$-walks outperforms NetKit (Macskassy & Provost, 2007) as well as Zhou et al. algorithm (Zhou et al., 2005), both in classification rate and computing time.

1. Introduction

This paper is concerned with semi-supervised classification of nodes in a graph. Given an input graph with some nodes being labeled, the problem is to predict the missing node labels. This problem has numerous applications such as classification of individuals in social networks, linked documents categorization or protein function prediction, to name a few.

Several approaches have been proposed to tackle semi-supervised classification problems in graphs. Kernel methods (Zhou et al., 2005; Tsuda & Noble, 2004) embed the nodes of the input graph into an Euclidean feature space where a classifier, such as a SVM, can be estimated. Despite of their good predictive performance, these techniques cannot easily scale up to large problems due to their high time complexity. NetKit is an alternative relational learning approach (Macskassy & Provost, 2007). It has a lower computational complexity but is less simple conceptually and may require to fine tune several of its components.

The approach proposed in this paper, called $\mathcal{D}$-walks, relies on random walks performed on the input graph seen as a Markov chain. More precisely, a betweenness measure, based on passage times during random walks of bounded length, is derived for each class (or label category). Unlabeled nodes are assigned to the category for which the betweenness is the highest. The $\mathcal{D}$-walks approach has the following properties: (i) it has a linear time complexity with respect to the number of edges and the maximum walk length considered; such a low complexity allows to deal with very large graphs, (ii) it can handle directed or undirected graphs, (iii) it can deal with multi-class problems and (iv) it has a unique hyper-parameter that can be tuned efficiently.

2. Discriminative random walks

We are given an input graph $\mathcal{G}$ containing a set of nodes $\mathcal{N}$ and edges $\mathcal{E}$. The (possibly weighted) adjacency matrix is denoted A. The graph $\mathcal{G}$ is assumed partially labeled. The nodes in the *labeled set* $\mathcal{L} \subset \mathcal{N}$ are assigned to a category from a discrete set $\mathcal{Y}$. The *unlabeled set* is defined as $\mathcal{U} = \mathcal{N} \setminus \mathcal{L}$.

Random walks in a graph can be modeled by a

Proceedings of the *17th Annual Machine Learning Conference of Belgium and the Netherlands (Benelearn)*, pp. 67–68, Liege, Belgium, May 19-20, 2008.

discrete-time Markov chain (MC) describing the sequence of nodes visited during the walk. Each state of the Markov chain corresponds to a distinct node of the graph. The MC transition probability matrix is simply given by $P = D^{-1}A$, with D the diagonal matrix of node degrees. We consider *discriminative random walks* ($\mathcal{D}$-walks, for short) in order to define a betweenness measure used for classifying unlabeled nodes.

Definition 1 ($\mathcal{D}$-walk) *Given a MC defined on the state set $\mathcal{N}$, a class $y \in \mathcal{Y}$ and a discrete length $l > 1$, a $\mathcal{D}$-walk is a sequence of state $q_0, \ldots, q_l$ such that $y_{q_0} = y_{q_l} = y$ and $y_{q_t} \neq y$ for all $0 < t < l$.*

The notation $\mathcal{D}_l^y$ refers to the set of all $\mathcal{D}$-walks of length l, starting and ending in a node of class y. We also consider $\mathcal{D}_{\leq L}^y$ referring to all $\mathcal{D}$-walks *up to* a given length L. The betweenness function $B_L(q, y)$ measures how much a node $q \in \mathcal{U}$ is located "between" nodes of class $y \in \mathcal{Y}$. The betweenness $B_L(q, y)$ is formally defined as the expected number of times the node q is reached during $\mathcal{D}_{\leq L}^y$-walks.

Definition 2 ($\mathcal{D}$-walk betweenness) *Given an unlabeled node $q \in \mathcal{U}$ and a class $y \in \mathcal{Y}$, the* $\mathcal{D}$-walk betweenness function $\mathcal{U} \times \mathcal{Y} \to \mathbb{R}^+$ *is defined as follows: $B_L(q, y) \triangleq \mathbb{E}\left[\mathrm{pt}(q) \mid \mathcal{D}_{\leq L}^y\right]$, where $\mathrm{pt}(q)$ is the passage times function $\mathcal{N} \to \mathbb{R}^+$ counting the number of times a node q has been visited.*

This betweenness measure is related to the one proposed by Newman in (Newman, 2005). Our measure is however relative to a specific class y rather than to the whole graph. It also considers random walks up to a given length instead of unbounded walks. Bounding the walk length has two major benefits: (i) better classification results are generally obtained with respect to unbounded walks (ii) the betweenness measure can be computed very efficiently (in $\Theta(|\mathcal{E}|L)$) using forward and backward recurrences, similar to those used in the Baum-Welch algorithm for HMM parameter estimation. Finally, an unlabeled node $q \in \mathcal{U}$ is assigned to the class with the highest betweenness.

3. Experiments

We report here preliminary experiments performed on the `Cora` dataset (Macskassy & Provost, 2007) containing 3582 nodes classified under 7 categories. As this graph is fully labeled, node labels were randomly removed and used as test set. More precisely, we have considered 9 different proportions of labeled nodes in the graph: $\{0.1, 0.2, \ldots, 0.9\}$ and for each labeling rate, 10 random deletions were performed. Comparative performances obtained with NetKit (Macskassy & Provost, 2007) and with the approach of Zhou et al. (Zhou et al., 2005) are also provided. The hyper-parameters of each approach have been tuned using ten-fold cross-validation. Figure 1 shows the correct classification rate on test data obtained by each approach for increasing labeling rates. The $\mathcal{D}$-walk approach clearly outperforms its competitors on these data. The $\mathcal{D}$-walks approach is also the fastest method. It requires typically 1.5 seconds of CPU[1] for every graph classification including the auto-tuning of its hyper-parameter L. NetKit takes about 4.5 seconds per graph classification and our implementation of Zhou et al. approach typically takes several minutes. Large graphs (several millions of edges) were also successfully classified in a few minutes with $\mathcal{D}$-walks while neither NetKit nor Zhou et al. methods could be applied on such large graphs.

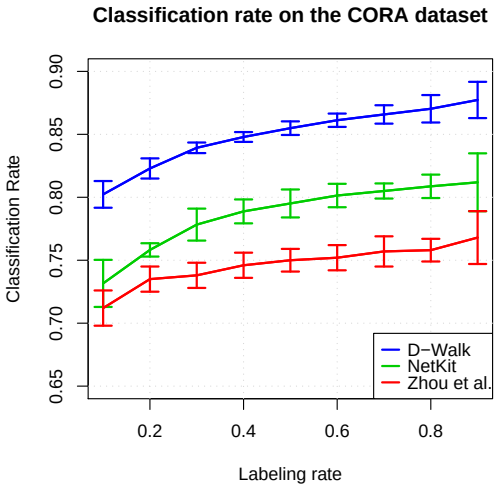

Figure 1. Classification rate of D-walk and two competing methods on the Cora dataset. Error bars report standard deviations over 10 independent runs.

References

Macskassy, S. A., & Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, *8*, 935–983.

Newman, M. (2005). A measure of betweenness centrality based on random walks. *Social networks*, *27*, 39–54.

Tsuda, K., & Noble, W. S. (2004). Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, *20*, 326–333.

Zhou, D., Huang, J., & Schölkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. *ICML '05: Proceedings of the 22nd international conference on Machine learning* (pp. 1036–1043). New York, NY, USA: ACM.

[1] Intel Core 2 Duo 2.2Ghz with 2Gb of virtual memory.