Scalable Network-based Learning Methods for Automated Function Prediction based on the Neo4j Graph-database

Marco Mesiti, Matteo Re and Giorgio Valentini

DI – Dipartimento di Informatica, Università degli Studi di Milano, via Comelico 39/41 - 20135 Milano, Italy

1. INTRODUCTION

A recent class of gene/protein function predictors, based on Graph Semi Supervised Learning (GSSL) [1], is able to exploit the functional relationships between genes to propagate existing annotations to unannotated genes that are topologically related in the network. As the prediction of gene functions using network-based methods is frequently performed at whole genome level, the development of scalable methods is of critical importance to make feasible the analysis of very large graphs.

Unfortunately GSSL methods scale poorly with the size of the graph [1] and usually have time complexity that becomes quickly prohibitive in large graphs, thus preventing their adoption in whole genome applications. This problem is particularly evident with the prediction of the function of genes in high eukaryotes like mammalians or plants.

GSSL methods can be roughly categorized as inductive when they are based on an explicit classification model, or transductive, if the labels are propagated using only information coming from the topology of the network and without training a classification model. Few approaches, aimed at overcoming the scaling limitations of transductive GSSL methods, have been recently proposed: they are based on kernel approximations [2] that trade approximation accuracy for scalability, or on the assumption that graphs have a block structure [3] which is often hard to prove for gene functional networks.

2. METHODS

We propose a novel framework for scalable semi-supervised network-based learning of gene functions that:

- provides a “local implementation” of both classical algorithms (e.g. random walks and random walks with restart) and recently proposed methods (e.g. kernelized score functions [4,5]), based on a “vertex centric” computational model;
- computes a random walk graph kernel without approximation;
- does not make assumptions on the nature of the considered network;
- exploits graph database technologies for the storage of the graph and for efficiently handling nodes and edges in secondary memory.

Our implementation allows the application of the transductive GSSL methods on off-the-shelf machines with limited speed and memory. The execution times are comparable with those obtained with the “global implementation” on main memory with well equipped machines.

In the global implementation, the network is represented through an adjacency matrix and very efficient routines for matrix multiplication are integrated in the implementation of the methods. In the local implementation, by contrast, the algorithms work "vertex-by-vertex" in a native representation of the network that allows us to easily access vertex neighbors in constant time (like in the adjacency-list representation of a graph). Our local implementation allows us to process huge networks, thus overcoming the limitations of current transductive GSSL methods.

Kernelized score functions provide semi-supervised transductive methods that generalize the notion of average distance from a set of core “positive” genes annotated to a specific functional class, and embed a general kernel to model the functional similarity between genes [4,5]. In this contribution we used few variants of kernelized score functions [4] (namely AVG – average, NN – nearest neighbors, and kNN – *k*-nearest neighbors) that can be naturally implemented in local form and that embed a local implementation of the 1-step random walk kernel.

The main contribution of this work is the local implementation of the methods by exploiting the Neo4j graph database [6]. The adoption of Neo4j, that effectively and efficiently handles the network on disk, allows us to overcome the issue of maintaining the entire network in main-memory. Moreover, the Neo4j APIs make the realization of the implementation easy and particularly efficient by exploiting the Neo4j caching facilities. The methods have been implemented in Java using the Neo4j APIs.

GSSL method		AUC	P20R	TIME global impl.	TIME local impl.
Kernelized Score Functions (1 step)	AVG	0.7203	0.1872	1m:20sec	2 m
	NN	0.7125	0.0684	1m:21sec	1m:57sec
	kNN (k=5)	0.7189	0.1359	1m:21sec	1m:55sec
Random Walk	1 step	0.7166	0.1448	44m	33m:25sec
	2 step	0.7849	0.1751	47m:5sec	62m:14s
	3 step	0.7450	0.1280	48m:10sec	91m:57s
Random Walk with Restart	Θ=0.3	0.7720	0.1543	2h:23m	4h:57m
	Θ=0.6	0.7837	0.1566	2h:25m	4h:57m

Table 1: Experimental comparison of the main-memory (global) and graph-database (local) implementations.

3. RESULTS

As a proof of concept of the proposed approach, we predicted the GO BP terms for the genes of the *Arabidopsis thaliana* model organism. To this end we constructed a gene network using the functional relationships encoded in the AraNet [7] functional gene network and we ranked all the 19.647 genes in AraNet (with a total of 1.062.222 edges) according to their likelihood to belong to 40 randomly selected GO BP terms with a number of annotated genes comprised between 20 and 200. The execution time of the local implementation is empirically compared with the global implementation of several network-based transductive rankers (random walk, random walk with restart, kernelized score functions). The global implementations have been executed on an Intel i7 machine with 20 GB RAM (machine M1), whereas, the local ones have been executed on an Intel Core Duo 1.60 machine with 4 GB RAM (machine M2).

Table 1 reports also the average AUC and the precision at 20% recall (P20R) achieved by the different methods across the 40 GO terms, even if the main aim of this work consists in comparing the empirical complexity of the global and local implementation of the same methods. The last two columns of Table 1 report the execution time obtained with the global implementation (which exploits only data structures loaded in main-memory) and the local implementation (in which the network is stored on disk and handled through Neo4j). The execution times with the main-memory and secondary-memory implementations are comparable, even tough the relevant difference between the characteristics of the used machines. We remark that the global implementation on machine M2 cannot be executed because of lack of memory.

4. CONCLUSIONS

Despite the popularity of GSSL methods, the development of purely transductive solutions able to scale to large networks is still an open problem that makes difficult to perform automated function prediction in multiple genomes or in large genomes such the ones of high eukaryotes.

Our general framework allows the application of off-the-shelf machines to the automated gene/protein function prediction in mammalians or plants or other model organisms with a large number of genes, and in perspective could be applied, with well-equipped machines, to multiple-species gene function prediction involving networks with millions of nodes/genes.

We emphasize that our approach could be applied to other GSSL algorithms, if they can be implemented through a vertex-centric computational model using graph database technologies, such as Neo4j.

REFERENCES

1. Liu, W., Wang, J. and Chang, S.F. Robust and Scalable Graph-Based Semisupervised Learning. Proc. of the IEEE, 100.9: 2624-2638, 2012

2. Kang U., Tong, H. and Sun J. Fast Random Walk Graph Kernel. *SIAM Int'l Conf. on Data Mining*. 2012.

3. Kai Y., Yu S., and Tresp. V. Blockwise Supervised Inference on Large Graphs. *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*. 2005.

4. M. Re and G. Valentini, Cancer Module Genes Ranking using Kernelized Score Functions, *BMC Bioinformatics* 13 (Suppl 14): S3, 2012.

5. M. Re, M. Mesiti and G. Valentini, A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks, I*EEE/ACM Trans. Comput. Biology Bioinform.* 9(6) pp. 1812-1818, 2012

6. Jim Webber. A Programmatic Introduction to Neo4j. In *Proc. of Conf. on Systems, Programming, and Applications*: Software for Humanity, 2012.

7. Lee, I., Ambaru, B., Pranjali Thakkar, P., Marcotte, E.M., Rhee, S.Y. Rational association of genes with traits using a genome-scale gene network for Arabidopsis thaliana, *Nat. Biotechnol.* 28,149-156, 2010