A Neural Procedure for Gene Function Prediction

Marco Frasca, Alberto Bertoni, and Andrea Sion

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
via Comelico 39 20135 Milano Italy
{frasca,bertoni}@dsi.unimi.it, andrea.sion@unimi.it

Abstract. The graph classification problem consists, given a weighted graph and a partial node labeling, in extending the labels to all nodes. In many real-world context, such as Gene Function Prediction, the partial labeling is unbalanced: positive labels are much less than negatives. In this paper we present a new neural algorithm for predicting labels in presence of label imbalance. This algorithm is based on a family of Hopfield networks, described by 2 continuous parameters and 1 discrete parameter, and it consists of two main steps: 1) the network parameters are learnt through a cost-sensitive optimization procedure based on local search; 2) a suitable Hopfield network restricted to unlabeled nodes is considered and simulated. The reached equilibrium point induces the classification of unlabeled nodes. An experimental analysis on real-world unbalanced data in the context of genome-wide prediction of gene functions show the effectiveness of the proposed approach.

Keywords: Neural Network, Hopfield Network, Gene Function Prediction.

1 Introduction

Label learning in graphs requires, given a graph with a partial classification of the nodes, to extend the classification to all nodes. Methods for solving this problem are useful in application domains where data are naturally represented as connected nodes, i.e. biological networks [16], social networks [4] and World-Wide-Web [6].

Several methods have been proposed for node classification. First algorithms rely on the *guilt-by-association* principle, which classify unlabeled nodes according to the majority of the labels in their direct neighborhoods [11]. Furthermore, nodes can propagate labels to their neighbors with an iterative process until convergence [17]. Markov Random Walks have been applied to tune the amount of propagation we allow in the graph, by setting the length of the walk across the graph [13]. Other approaches are based on graph regularization [1], on global graph consistency [9], on Markov [5] and Gaussian Random Fields [14].

Unfortunately, these methods suffer a decay in the quality of solutions when input data are unbalanced, that is positive examples are significantly less than those negative. This issue is particularly relevant in Gene Function Prediction (GFP), where the imbalance in data requires to adopt cost-sensitive strategies [7].

B. Apolloni et al. (Eds.): *Neural Nets and Surroundings*, SIST 19, pp. 179–188.
DOI: 10.1007/978-3-642-35467-0_19

For their common characteristics, many of the described approaches can be cast into a common framework where a quadratic cost objective function is minimized [2]. From this point of view, it seems natural a neural approach based on Hopfield networks, that are local optimizers of quadratic functions [9].

In [9] the neural algorithm GAIN is applied to GFP. Neurons represent genes, the connection weights the "similarities" between genes, the activation values are 1, -1 and the thresholds are 0 for each neuron. Fixed a functional class, only a subset neurons are classified (positive or negative), while the classification of the other is unknown. For classifying the unlabelled neurons, an initial state x is given by setting 1 the positive neurons, -1 the negative neurons and 0 those still unclassified. The dynamics of the network is applied to this state until the equilibrium point $\hat{x}$ is reached; a gene k is classified as "positive" iff $\hat{x}_k = 1$.

From a biological standpoint, this approach is motivated by the fact that minimizing the overall energy means maximizing the weighted sum of edges connecting neurons with the same activation value. Nevertheless, this algorithm is affected by the *imbalance problem* in functional classes. Since weights are non negative and thresholds are 0, when the positive examples are less than the negative, the network is likely to converge to a trivial state $(-1,-1,\ldots,-1)$. Observe that, in biological taxonomies, for most of the functional classes only a small number of positive examples is available.

In [3] another neural algorithm, called COSNet, has been proposed for solving the GFP problem on unbalanced data. As in the previous approach, neurons represent genes and connection weights represent the similarities between genes. However, here a class of networks with 2 parameters is considered: each neuron has activation values $\sin\alpha$ and $-\cos\alpha$ and threshold γ. Firstly, the algorithm learns the optimal values of the parameters α and γ, then it runs the subnetwork restricted neurons with unknown classification, that are classified according to the reached equilibrium state.

We point out that in both previous algorithms all the neurons of the network have the same activation values. Since, in principle, each neuron in a Hopfield network might have different activation values, in this work we investigate this case by partitioning the neurons in two classes and assigning to each class different activation values.

Accordingly, in Sect. 3 a family of parametrized Hopfield networks is introduced, whose parameters are the the possible partitions of neurons in 2 classes and the corresponding activation values. In Sect. 5 it is derived an algorithm that firstly learns the optimal values of the 2 continuous parameters (the different activation values) and the discrete parameter (the neuron partition). Then the algorithm runs the subnetwork restricted to neurons with unknown classification , that are classified according to the reached equilibrium state. Finally, in Sect. 6 we describe the experimental procedure adopted to validate the algorithm on the genome-wide prediction of gene functions in a model organism, including around 200 functional classes of the FunCat taxonomy [12], and using 3 different types of biomolecular data.

2 Gene Function Prediction (GFP)

In our setting, GFP is formalized as the problem of label learning in graphs [2]. Genes are represented by a set of nodes $V = \{1,2,\ldots,n\}$ and relationships between genes are

encoded through a symmetric $n \times n$ real weight matrix W, whose elements w_{ij} represent similarities between genes i and j.

For a given functional class c, the nodes V are labeled with $\{+,-\}$, leading to the subsets P and N of positive and negative vertices for class c. For most model organisms, usually the functional labeling is known only for a subset $S \subset V$, while is unknown for $U = V \setminus S$. Let be $S^+ = S \cap P$ and $S^- = S \cap N$: we can refer to S^+, S^- and W as the "prior information" of the GFP problem.

The *Gene Function Prediction problem* consists in finding a bipartition (U^+, U^-) of genes in U on the basis of the prior information. Genes in U^+ are then considered candidates for the class $P \cap U$. From this standpoint, GFP is set as a semi-supervised learning problem on graphs, since gene functions can be predicted by exploiting both labeled and unlabeled nodes/genes and the weighted connections between them.

3 Hopfield Networks for GFP

In this Section we consider a family of Hopfield networks [8] with binary neurons partitioned in two classes G_1 and G_2. The activation values are $\{\sin\alpha_1, -\cos\alpha_1\}$ for neurons in G_1 and $\{\sin\alpha_2, -\cos\alpha_2\}$ for neurons in G_2; the thresholds are set to 0.

Formally, in our setting, a *Hopfield network* H with neurons $V = \{1,2,\ldots,n\}$ is a quadruple $H = \ < W, b, \alpha_1, \alpha_2 >$, where:

- $W = (w_{ij})$ is a $n \times n$ symmetric matrix with null diagonal, whose elements $w_{ij} \in \mathbb{R}$ represent the connection strength between neurons i and j
- $b \in \{0,1\}^n$ is a binary vector partitioning neurons in two classes:

$$G_1 = \{k | b_k = 1\},\ G_2 = \{k | b_k = 0\}$$

- α_1, α_2 are (possibly distinct) real values denoting the neuron activation values: $\{\sin\alpha_1, -\cos\alpha_1\}$ (resp. $\{\sin\alpha_2, -\cos\alpha_2\}$) for neurons k such that $b_k = 1$ (resp. $b_k = 0$)

The dynamics of the network is described as follows:

1. At time 0 an initial value $x_i(0)$ is given for each neuron i
2. At time $t+1$ each neuron is updated asynchronously (up to a permutation) by the following activation rule

$$x_i(t+1) = \begin{cases} b_i \sin\alpha_1 + (1-b_i)\sin\alpha_2 & \text{if } \sum_{j=1}^{i-1} w_{ij}x_j(t+1) + \sum_{k=i+1}^{n} w_{ik}x_k(t) > 0 \\ -b_i \cos\alpha_1 - (1-b_i)\cos\alpha_2 & \text{if } \sum_{j=1}^{i-1} w_{ij}x_j(t+1) + \sum_{k=i+1}^{n} w_{ik}x_k(t) \leq 0 \end{cases} \quad (1)$$

The state of the network at time t is $x = (x_1(t), x_2(t), \cdots, x_n(t))$. The main feature of a Hopfield network is that it admits a Lyapunov function of the dynamics. In particular, consider the following quadratic state function (*energy function*):

$$E(x) = -\frac{1}{2} x^T W x \quad (2)$$

During the dynamics this function is not increasing; this guarantees that the dynamics converges to an equilibrium state $\hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)$, which corresponds to a local minimum of the energy function [8].

4 Subnetwork Property

Let be $H = < W, b, \alpha_1, \alpha_2 >$ a Hopfield network. Fixed $U = \{1, 2, \ldots, h\}$ and $S = \{h+1, h+2, \ldots, n\}$, each network state x can be decomposed in $x = (u, s)$, where u and s are respectively the states of neurons in U and in S. The energy function of H can be written by separating the contributions due to u and s:

$$\begin{aligned} E(u,s) = & -\frac{1}{2}\left(u^T W_{uu} u + s^T W_{ss} s + u^T W_{us} s + s^T W_{us}^T u\right) \\ = & -\frac{1}{2} u^T W_{uu} u + u^T (-W_{us} s) + C \end{aligned} \tag{3}$$

where $W = \begin{pmatrix} W_{uu} & W_{us} \\ W_{us}^T & W_{ss} \end{pmatrix}$ is the weight matrix W decomposed in its submatrices W_{uu} connecting nodes in U, W_{ss} connecting nodes in S, W_{us} connecting each node in U with each node in S, and W_{us}^T its transpose. $C = -\frac{1}{2} s^T W_{ss} s$ is a term constant w.r.t. u.

Suppose now that a state $\tilde{s}$ of neurons in S is given. We are interested in the dynamics obtained by allowing the update just of neurons in U, without updating neurons in S. We denote with $H_{U|\tilde{s}}$ the Hopfield network with neurons U which realizes this dynamics and $E_{|\tilde{s}}$ the corresponding energy; from equation (3) it holds:

Theorem 1. *$H_{U|\tilde{s}} = < W_{uu}, b^u, \alpha_1, \alpha_2 >$, with thresholds $-W_{us}\tilde{s}$ and where b^u is the subvector of b restricted to neurons in U.*

Given a state $\tilde{s}$ of neurons in S, we say that $\tilde{s}$ is part of global minimum of the energy E of H if there is a state u of neurons in U s.t. $(u, \tilde{s})$ is a global minimum of E. The introduction of the network $H_{U|\tilde{s}}$, is motivated by the following property:

Theorem 2. (Subnetwork property) *If $\tilde{s}$ is part of a energy global minimum of H, and $\tilde{u}$ is a global minimum of the energy $E_{|\tilde{s}}(u)$, then $(\tilde{u}, \tilde{s})$ is a energy global minimum of H.*

In our setting, we associate the given bipartition (S^+, S^-) of S with the state $\tilde{s} = x(S^+, S^-)$:

$$x_i(S^+, S^-) = \begin{cases} b_i \sin\alpha_1 + (1-b_i)\sin\alpha_2 & \text{if } i \in S^+ \\ -b_i \cos\alpha_1 - (1-b_i)\cos\alpha_2 & \text{if } i \in S^- \end{cases}$$

for each $i \in S$. Suppose, for suitable b, α_1, α_2, that $x(S^+, S^-)$ is part of a energy global minimum of $H =< W, b, \alpha_1, \alpha_2 >$; then, by the subnetwork property, we can predict the hidden part relative to neurons U by minimizing the energy of $H_{U|x(S^+,S^-)}$.

5 Algorithm for GFP

In this Section we exhibit a procedure based on Hopfield networks for dealing with the *GFP* problem.

For a given similarity matrix W, we consider the class of networks $H =< W, b, \alpha_1, \alpha_2 >$ on neurons $V = \{1, 2, \ldots, n\}$, where α_1, α_2 are real parameters and $b \in \{0,1\}^n$ is a discrete parameter.

Fixed a functional class, an instance of GFP problem is given by the matrix W and the sets S^+ and S^- of positive and negative examples. We hypothesize that there exist a triple $(\hat{b}, \hat{\alpha}_1, \hat{\alpha}_2)$ such that:

1. The solution of the problem corresponds to an energy global minimum of $H =< W, \hat{b}, \hat{\alpha}_1, \hat{\alpha}_2 >$
2. $x(S^+, S^-)$ is part of an energy global minimum of H

Then, by Theorem 2, we can discover the hidden part $\hat{u}$ of the global minimum by minimizing the energy of the network $H_{U|x(S^+,S^-)}$. Accordingly, the procedure for solving the GFP problem can be factorized into two main steps:

Step 1. Determine the parameters $(\hat{b}, \hat{\alpha}_1, \hat{\alpha}_2)$ such that the state $x(S^+, S^-)$ is approximately part of a global minimum by finding the parameters (b, α_1, α_2) for which $x(S^+, S^-)$ is "as close as possible" to a part of an equilibrium state of H.

Step 2. Minimize the energy function of the network $H_{U|x(S^+,S^-)}$ with the estimated parameters $(\hat{b}, \hat{\alpha}_1, \hat{\alpha}_2)$ by reaching an equilibrium state $\hat{u}$ in a dynamics generated by a suitable initial state.

Finally, the solution (U^+, U^-) of GFP is:

$$U^+ = \{i \in U \mid \hat{u}_i > 0\}$$
$$U^- = \{i \in U \mid \hat{u}_i \leq 0\}.$$

In the following we discuss in more details Step 1 (Section 5.1) and Step 2 (Section 5.2) of the algorithm.

5.1 Finding the Optimal Parameters

The main goal of this step is to find the values of the parameters b, α_1 and α_2 such that the state $x(S^+, S^-)$ is "as close as possible" to an equilibrium state.

To this end, we consider the parametrized subnetwork restricted to neurons in S, i.e. $H_S =< W_{ss}, b^s, \alpha_1, \alpha_2 >$, where b^s, α_1 , α_2 are the parameter to be learned.

In the following we describe the objective function adopted for learning the network parameters and the relative optimization procedure.

Objective Function. First of all, we fix b^s, α_1 , α_2. Every neuron i has an "internal energy" A_i, where:

$$\begin{aligned} A_i = \; & \sin\alpha_1 \sum_{k\in S} w_{ik} P_k b_k^s + \sin\alpha_2 \sum_{k\in S} w_{ik} P_k (1 - b_k^s) \\ & - \cos\alpha_1 \sum_{k\in S} w_{ik} (1 - P_k) b_k^s - \cos\alpha_2 \sum_{k\in S} w_{ik} (1 - P_k)(1 - b_k^s) \end{aligned} \tag{4}$$

where P is the characteristic vector of S^+ (i.e. $P_k = 1$ iff $k \in S^+$). By means of A_i, we are able in computing the number of *true positive* TP, *false negative* FN and *false positive* FP:

- $TP(b^s, \alpha_1 , \alpha_2) = \sum_{i\in S} P_i \mathrm{HS}(A_i)$, i.e. the number of positive examples with positive internal energy (true positive)
- $FN(b^s, \alpha_1 , \alpha_2) = \sum_{i\in S} P_i (1 - \mathrm{HS}(A_i))$, i.e. the number of positive examples with negative internal energy (positive misclassification)

- $FP(b^s, \alpha_1, \alpha_2) = \sum_{i \in S}(1 - P_i)\mathrm{HS}(A_i)$, i.e. is the number of negative examples with positive internal energy (negative misclassification)

Here HS denotes the Heaviside function ($\mathrm{HS}(x) = 1$ if $x \geq 0$, 0 otherwise).

The function we want to maximize is the so called F_{score} :

$$F_{score}(b^s, \alpha_1, \alpha_2) = \frac{2TP}{2TP + FP + FN}$$

By observing that $0 \leq F_{score} \leq 1$, this criterion is justified by the following:

Theorem 3. *$F_{score}(b^s, \alpha_1, \alpha_2) = 1$ iff $x(S^+, S^-)$ is an equilibrium state of the sub -network H_S.*

Optimization Procedure. The values of parameters that maximize the F_{score} criterion are:

$$(\hat{b}^s, \hat{\alpha}_1, \hat{\alpha}_2) = \underset{b^s \in \{0,1\}^{|S|}, \alpha_1, \alpha_2}{\operatorname{argmax}} F_{score}(b^s, \alpha_1, \alpha_2). \tag{5}$$

For every $b^s \in \{0,1\}^{|S|}$, we define $F(b^s) = \max\limits_{\alpha_1, \alpha_2} F_{score}(b^s, \alpha_1, \alpha_2)$. Given b^s, an approximation of F(b^s) can be found by applying a standard continuous optimization procedure.

In order to maximize F(b^s), we adopt a simple local search on hypercube $\{0,1\}^{|S|}$, where the neighborhood of b^s is $\{\overline{b}^s | d_H(b^s, \overline{b}^s) = 1\}$, and d_H is the Hamming distance. Once obtained the local optimum $\hat{b}^s$, we determine the optimal values for α_1 and α_2 as $(\hat{\alpha}_1, \hat{\alpha}_2) = \underset{\alpha_1, \alpha_2}{\operatorname{argmax}} F_{score}(\hat{b}^s, \alpha_1, \alpha_2)$.

Having the optimal values $(\hat{b}^s, \hat{\alpha}_1, \hat{\alpha}_2)$, we want to extend the vector $\hat{b}^s$ to $\hat{b} = (\hat{b}^u, \hat{b}^s)$, where the indices of $\hat{b}^u$ are the elements of U.

With regard to this, compute for all k:

$$\Delta_k^+ = \sum_{i \in S^+} w_{ki} \;\; ; \;\; \Delta_k^- = \sum_{i \in S^-} w_{ki}$$

In this way, we associate with each neuron k a point $P_k = (\Delta_k^+, \Delta_k^-)$ in the plane. Consider now the subsets of points C_1 and C_2, where:

$$C_1 = \{P_k : \hat{b}_k^s = 1\} \;\; ; \;\; C_2 = \{P_k : \hat{b}_k^s = 0\}$$

By using C_1, C_2 we learn two bivariate normal distributions $N_2(\mu_1, \Sigma_1), N_2(\mu_2, \Sigma_2)$ where, for $j = 1, 2$, μ_j and Σ_j are respectively the sample mean and the sample covariance of C_j.

Finally, if $k \in U$, we set $\hat{b}_k^u = 1$ if and only if the probability of P_k, according to $N_2(\mu_1, \Sigma_1)$, is greater than the probability of P_k, according to $N_2(\mu_2, \Sigma_2)$.

5.2 Finding the Unknown Labels by Network Dynamics

After the computation of the optimal parameters $(\hat{b}, \hat{\alpha}_1, \hat{\alpha}_2)$, we consider the sub-network $H_{U|x(S^+,S^-)}$:

$$H_{U|x(S^+,S^-)} = < W_{uu}, \hat{b}^u, \hat{\alpha}_1, \hat{\alpha}_2 > \quad (6)$$

with thresholds $-W_{su}^T x(S^+, S^-)$.

Fixed an initial state $u_i = 0$ for each $i \in \{1, 2, \ldots, h\}$, we run the sub-network $H_{U|x(S^+,S^-)}$ to learn the unknown labels of neurons U.

If $\hat{u}$ is the stable state reached by this dynamics, we obtain the final solution (U^+, U^-) by setting:

$$U^+ = \{k|\hat{u}_k > 0\} \ , \ U^- = \{k|\hat{u}_k \leq 0\}$$

6 Algorithm Validation

In this Section we describe the procedure for experimentally evaluate our algorithm and we discuss the results of the comparison of the algorithm with other state-of-the-art methods.

6.1 Experimental Setting

We performed predictions of gene functions at genome-wide level in the *S.cerevisiae* organism (yeast), using the whole FunCat ontology [12] [1]. We predicted functions of genes belonging to three different biomolecular data sets previously adopted in[3]:

- *Pfam* is an enriched representation of Pfam domains by replacing the binary scoring with log E-values obtained with the HMMER software toolkit. This dataset contains 3528 genes and 5724 features.
- *Expr* data contains 250 gene expression measures of 4523 genes
- *SP-sim* is a data set containing pairwise similarities between 3527 yeast genes represented by Smith and Waterman log-E values between all pairs of yeast sequences

As validation procedure we adopt the 10-folds cross validation: genes are randomly divided into 10 equal-sized subsets, and each time the labels for genes in a fold are hidden and predicted using as training data the other nine folds.

6.2 Results

First of all, we compared our method with semi-supervised and supervised machine learning methods proposed in the literature for the Gene Function Prediction problem. We consider: 1) the *GAIN* algorithm [9]; 2) *Zhu-LP*, a popular semi-supervised label propagation learning algorithm based on Gaussian random fields and its class mass

[1] We used the funcat-2.1 scheme with the annotation data funcat-2.1_data_20070316, available from: `ftp://ftpmips.gsf.de/yeast/catalogues/funcat/funcat-2.1_data_20070316`

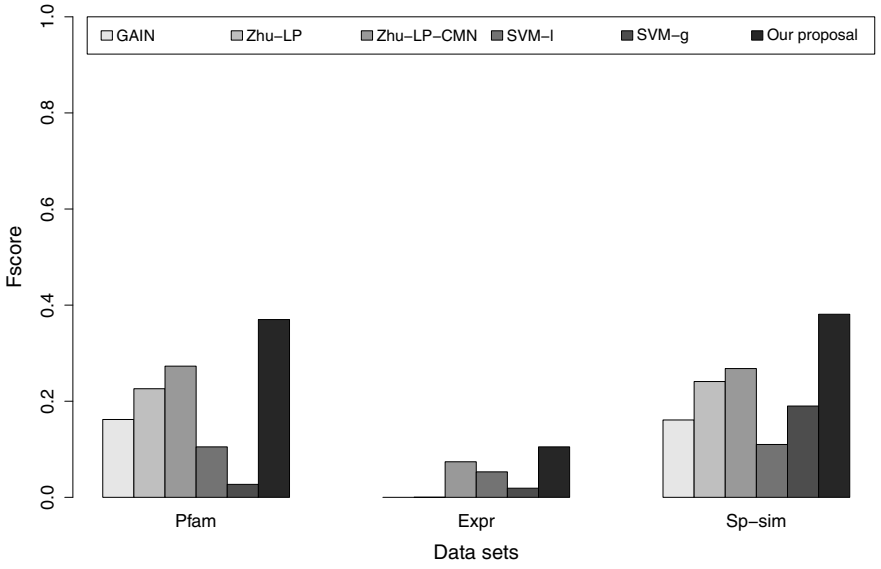

Fig. 1. F-score comparison in terms of averaged F_{score}

normalized version *Zhu-LP-CMN* [17]; 3) Support Vector Machines with linear (*SVM-l*) and Gaussian (*SVM-g*) kernels [10].

In order to take into account the imbalance in positive and negative labels characterizing the GFP context, we adopt the *F-score* performance measure (Sect. 5.1). Figure 1 shows for each dataset and for all the considered methods, the average F-score across all the functional classes.

Our algorithm highly outperforms in terms of average F-score all the other compared methods, and the difference is always significant at 10^{-6} significance level, according to the Wilcoxon signed-ranks test [15]. We think this results are due to the inherent cost-sensitive nature of the algorithm, which is able in automatically finding the parameters that better "re-equilibrate" the imbalance in labels.

Moreover, in order to better analyze the performance of our algorithm, we evaluate also the precision of the algorithm, $precision = \frac{TP}{TP+FP}$, which informally is the probability that a positive prediction corresponds to a true positive. We point out that in GFP context the automatic positive predictions of unknown genes need to be confirmed by expensive experimental laboratory procedures; accordingly, achieving a high precision in predicting functions of unknown genes is central, and provides reliable clues to experimentally check the membership of a gene to a functional class.

In Table 1 we show the averaged *precision* and F_{score} of our algorithm and of another cost-sensitive algorithm, COSNet, proposed in [3].

We can observe that the two algorithms achieve close values of F_{score}; on the other hand, the present algorithm obtains a significant improvement in precision at $\alpha = 5 * 10^{-10}$ significance level (Wilcoxon signed-ranks test).

Table 1. Precision and F-score of COSNet and of our algorithm averaged across the Funcat functional classes

	Pfam	
Method	Precision	F
COSNet	0.445	0.375
Our proposal	0.509	0.370
	Expr	
Method	Precision	F
COSNet	0.057	0.085
Our proposal	0.147	0.105
	SP-sim	
Method	Precision	F
COSNet	0.445	0.376
Our proposal	0.489	0.368

Conclusions

In this paper we propose a new algorithm for predicting node labels in graph in presence of label imbalance. The algorithm is based on a family of Hopfield networks with 2 real parameters and 1 discrete parameter. The parameters are learned by means of a cost-sensitive procedure, which allows to manage the imbalance in data. Then the subnetwork of unlabeled nodes is simulated and the reached equilibrium state provides the classification of unlabeled nodes.

The algorithm has been experimentally validated on the problem of predicting the functions of genes in a model organism; the results, compared with those of the state-of-the-art methods, show the effectiveness of this approach.

In this paper, neurons are bi-partitioned, but in principle we could consider k-partitions, increasing the number of parameters. It should be interesting to evaluate the impact that the number of parameters has on the predicting capabilities of the algorithm, and to define the optimal number of parameters (which ensures to avoid overfitting) through model selection techniques.

References

1. Belkin, M., Matveeva, I., Niyogi, P.: Regularization and Semi-supervised Learning on Large Graphs. In: Shawe-Taylor, J., Singer, Y. (eds.) COLT 2004. LNCS (LNAI), vol. 3120, pp. 624–638. Springer, Heidelberg (2004)
2. Bengio, Y., Delalleau, O., Le Roux, N.: Label Propagation and Quadratic Criterion. In: Chapelle, O., Scholkopf, B., Zien, A. (eds.) Semi-Supervised Learning, pp. 193–216. MIT Press (2006)

3. Bertoni, A., Frasca, M., Valentini, G.: *COSNet*: A Cost Sensitive Neural Network for Semi-supervised Learning in Graphs. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part I. LNCS, vol. 6911, pp. 219–234. Springer, Heidelberg (2011)
4. Borgatti, S., Mehra, A., Brass, D., Labianca, G.: Network Analysis in the Social Sciences. Science 232, 892–895 (2009)
5. Deng, M., Chen, T., Sun, F.: An integrated probabilistic model for functional prediction of proteins. J. Comput. Biol. 11, 463–475 (2004)
6. Dorogovtsev, S., Mendes, J.: Evolution of networks: From biological nets to the Internet and WWW. Oxford University Press, Oxford (2003)
7. Elkan, C.: The foundations of cost-sensitive learning. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, pp. 973–978 (2001)
8. Hopfield, J.: Neural networks and physical systems with emergent collective compatational abilities. Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
9. Karaoz, U., et al.: Whole-genome annotation by using evidence integration in functional-linkage networks. Proc. Natl. Acad. Sci. USA 101, 2888–2893 (2004)
10. Lin, H.T., Lin, C.J., Weng, R.: A note on platt's probabilistic outputs for support vector machines. Machine Learning 68(3), 267–276 (2007)
11. Marcotte, E., Pellegrini, M., Thompson, M., Yeates, T., Eisenberg, D.: A combined algorithm for genome-wide prediction of protein function. Nature 402, 83–86 (1999)
12. Ruepp, A., et al.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Research 32(18), 5539–5545 (2004)
13. Szummer, M., Jaakkola, T.: Partially labeled classification with Markov random walks. In: Advances in Neural Information Processing Systems (NIPS), vol. 14, pp. 945–952. MIT Press (2001)
14. Tsuda, K., Shin, H., Scholkopf, B.: Fast protein classification with multiple networks. Bioinformatics 21(suppl. 2), ii59–ii65 (2005)
15. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics 1, 80–83 (1945)
16. Wuchty, S., Ravasz, E., Barabsi, A.L.: The architecture of biological networks. Complex Systems in Biomedicine 5259, 165–181 (2003)
17. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)