Selection of Negative Examples for Node Label Prediction through Fuzzy Clustering Techniques

Marco Frasca, Dario Malchiodi

Dipartimento di Informatica, Università degli Studi di Milano
via Comelico 39, 20135 Milan, Italy
{frasca,malchiodi}@di.unimi.it

Abstract. Negative examples, which are required for most machine learning methods to infer new predictions, are rarely directly recorded in several real world databases for classification problems. A variety of heuristics for the choice of negative examples have been proposed, ranging from simply under-sampling non positive instances, to the analysis of class taxonomy structures. Here we propose an efficient strategy for selecting negative examples designed for Hopfield networks which exploits the clustering properties of positive instances. The method has been validated in the prediction of protein functions of a model organism.

Keywords: negative selection, fuzzy clustering, Hopfield networks

1 Introduction

In several real-world contexts ranging from text categorization [7] to protein function prediction [8], the notion of negative example is not well defined, and the selection of appropriate sets of negative patterns may sensibly improve the predictive capabilities of supervised and semi-supervised methodologies for inductive inference. This area of machine learning, named Positive-Unlabeled (PU) learning, has seen a surge of interest in latest years [8]. Indeed, in different contexts there is no "gold standard" for negative items and only positive instances are the results of accurate studies (negative examples usually are simply "non positive"). Several approaches have been proposed for selecting negative instances, such as randomly sampling items (assuming the probability of getting a false negative to be low) [10], sampling according to positive-negative similarity measures [5], selecting the items positive for the sibling and/or ancestral categories of the categories of interest as negative examples [18]. Nevertheless, hierarchical methods cannot be applied in contexts where categories are not structured as a hierarchy (for instance in action video detection [5]). Furthermore, even when parent-child relationships are available for the classes being predicted, strategies based on sibling/ancestral categories may often break down, as some items are annotated to more than one sibling category, and many items have few siblings to use [20].

In this work we propose a novel methodology for graph-based semi-supervised learning which is composed of two main steps: Step 1) a novel strategy for PU learning specific for Hopfield networks (HNs) [11], which can be applied both to structured classes and to hierarchy-less contexts; Step 2) a semi-supervised classifier based on a family of parametric Hopfield networks, which embeds the negative selection performed at Step 1 in the dynamics of network.

At Step 1, the approach for detecting negative instances can be summarized as follows:

1.1. The matrix describing symmetric connections between instances is transformed into a feature matrix, in which each instance is associated with a 3-feature vector, obtained through the application of random walks of length $1, 2$ and 3 starting from positive instances.

1.2. The positive points are clustered using a dynamic version of the fuzzy C means algorithm (FCM) [3] exploiting a suitable index [6] in order to decide the optimal number of clusters to summarize data; then, every negative point is assigned a score consisting in the maximum membership it has to the detected clusters of positive items. The low-ranked points according to this score are considered as negatively labeled.

1.3. The remaining points are then further discriminated according to their stability in the Hopfield network constructed from the input graph. In particular, we extend the set of negative instances with those points that locally minimize the energy function.

At Step 2, the Hopfield network is simulated and the items not selected during Step 1 are allowed to change their state along with the items whose label has to be predicted (test set). The final equilibrium state is used to infer the prediction for instances in the test set. In particular, the instances corresponding to neurons fired at equilibrium are considered candidates for the positive class.

We experimentally validated the proposed methodology in the protein function prediction problem, which consists in inferring the biomolecular functions of recently discovered proteins starting from their interactions with already characterized proteins, and in which, apart from rare cases, just the positive annotations are available for the Gene Ontology classes [1]. The comparison with state-of-the-art supervised and semi-supervised label prediction methods and with the vanilla Hopfield network shows the effectiveness of our approach. Moreover, the reduced computational complexity, due to the application of clustering techniques only to positive instances (which usually are a large minority of the data set), allows the application of the proposed methodology to contexts characterized by large-size data.

The paper is organized as follows: Section 2 describes the afforded prediction task, while Section 3 explains why in such context it is difficult to negatively label the available data. Sections 4 and 5 respectively describe and test the proposed methodology. Some concluding remarks close the paper.

2 Node Label Prediction (NLP) in Graphs

Consider a weighted graph $G = (V, \boldsymbol{W})$, where $V = \{1, 2, \ldots, n\}$ is the set of vertices, $\boldsymbol{W}$ is a symmetric $n \times n$ weight matrix, where $W_{ij} \in [0, 1]$ indicates the strength of the evidence of co-functionality between vertices i and j.

For a given class to be predicted, the vertices in V are labeled with $\{+, -\}$, and the labeling is known only for the subset $S \subset V$, whereas it is unknown for $U = V \setminus S$. Furthermore, labeled vertices are partitioned into positive S_+ and negative S_- vertices.

The *Node Label Prediction on graphs* (*NLP*) problem consists in determining a bipartition (U_+, U_-) of vertices in U, such that vertices in U_+ can be considered as candidates for the positive class.

3 The Problem of Selecting Negative Examples for NLP

The problem of selecting negative examples for classification tasks arises in those contexts in which items are classified for the properties they possess, and not for those properties they do not possess. For instance, in text classification it is not practical to label all the topics a document does not contain; hence just the topics a document contains are available [14]. Or in protein function prediction, where the classes to be predicted are the bio-molecular protein functions, proteins are rarely labeled with the functions they do not possess (negative annotation), thus most of all annotations are positive [1]. In our context, we consider unlabeled those vertices for which we want to infer the labels (i.e. U), whereas genes in S which are not positive for the current class (i.e. S_-) are in principle considered as negative. Nevertheless, there is no evidence that vertices in S_- are not positive; some of these instances may be not positive just because not enough studies and analyses have been carried out yet.

To take into account these issues, we assume that there exists a subset $S_-^p \subset S_-$ containing negative instances that are more likely to be classified as positive in future, but such subset is not known. Thus, nodes in $S_- \setminus S_-^p$ can be considered in turn as reliable negative instances. Clearly, the subset S_-^p may vary according to the class to be predicted. In the following we propose a novel method to detect such subset specifically designed for a family of classifiers based on parametric Hopfield networks.

4 Methods

In this section we first describe an algorithm for the *NLP* problem which exploits the properties of a parametric Hopfield model, then we present an extension that embeds in the model a procedure for detecting the negative instances candidates for the set S_-^p.

4.1 An algorithm for NLP problem

COSNet, COst-Sensitive neural Network [2, 9], is a semi-supervised algorithm recently proposed for *NLP* problems characterized by unbalanced labelings. *COSNet* is based on parametric Hopfield networks $H = \langle \boldsymbol{W}, k, \rho \rangle$, where k is the neuron activation threshold and ρ is a real number in $(0, \frac{\pi}{2})$, that determines the two different values $\{-\cos\rho, \sin\rho\}$ for neuron activation. The distinction between neuron labels and neuron activation values allows the method to automatically determine the activation levels for positive and negative neurons in order to counterbalance the prevalence of labels in the majority class (positive or negative class). The optimal parameters $(\hat{k}, \hat{\rho})$ of the sub-network restricted to labeled nodes are learned so as to move the state determined by the bipartition (S_+, S_-) "as close as possible" to an equilibrium state. The authors have shown that the learned activation values move the state provided by know labels closer to a global minimum of the network restricted to S than the classical activation values $\{-1, 1\}$ (see [9] for details). We denote by $\hat{\boldsymbol{l}}$ the state of labeled network after learning, where $\hat{l}_i = -\cos\hat{\rho}$ if $i \in S_-$, and $\hat{l}_i = \sin\hat{\rho}$ when $i \in S_+$.

Then, the network restricted to unlabeled nodes U is simulated by adopting $\{-\cos\hat{\rho}, \sin\hat{\rho}\}$ as activation values and $\hat{k}$ as unique activation threshold. Neurons in S are not updated. The initial state is set to the null vector, and by assuming that $U = \{1, 2, \cdots, h\}$ and $S = \{h+1, h+2, \cdots, n\}$ (up to a permutation), the network evolves according to the following asynchronous update rule:

$$u_i(t) = \begin{cases} \sin\hat{\rho} & \text{if } \sum\limits_{j=1}^{i-1} W_{ij}u_j(t) + \sum\limits_{j=i+1}^{h} W_{ij}u_j(t-1) - \theta_i > 0 \\ -\cos\hat{\rho} & \text{if } \sum\limits_{j=1}^{i-1} W_{ij}u_j(t) + \sum\limits_{j=i+1}^{h} W_{ij}u_j(t-1) - \theta_i \leq 0 \end{cases} \qquad (1)$$

where $u_i(t)$ is the state of neuron $i \in U$ at time t, and $\theta_i = \hat{k} - \sum_{j=h+1}^{n} W_{ij}\hat{l}_j$ is the activation threshold of node i. The state of the network at time t is $\boldsymbol{u}(t) = (u_1(t), u_2(t), \ldots, u_h(t))$, and the main feature of a HN is that it admits a Lyapunov function of the dynamics. In particular, consider the following quadratic state function (energy function):

$$E(\boldsymbol{u}) = -\frac{1}{2} \sum_{\substack{i,j=1 \\ j \neq i}}^{h} W_{ij}u_iu_j + \sum_{i=1}^{h} u_i\theta_i \qquad (2)$$

During the dynamics this function is not increasing and the dynamics converges to an equilibrium state $\hat{\boldsymbol{u}}$, which corresponds to a local minimum of the energy function [11]. The motivation of this approach is that minimizing (2) means maximizing the weighted sum of consistent edges, that is edges connecting neurons at the same state, so as to maximize the coherence with the prior information coded in $\boldsymbol{W}$ and in the labeling $\hat{\boldsymbol{l}}$. The final solution (U_+, U_-) is obtained by setting $U_+ = \{i \in U \mid \hat{u}_i = \sin\hat{\rho}\}$ and $U_- = \{i \in U \mid \hat{u}_i = -\cos\hat{\rho}\}$. In [9] the authors showed that although parameter optimization and label inference are carried out separately, Step 2 and 3 of *COSNet* preserve convergence and optimization properties of the whole HN H.

4.2 A Strategy for Negative Selection

The strategy we propose, able to efficiently identify the subset S_-^p, is composed of three main steps, that we describe in detail in the following.

Node projection onto a feature space. The n by n connection matrix $\boldsymbol{W}$ is transformed into a n by 3 feature matrix $\boldsymbol{F}$, where the i-th row $\boldsymbol{F}_i = (p_i^1, p_i^2, p_i^3)$ is the feature vector associated with node $i \in V$. The j-th feature p_i^j is the probability that a random walk of length j starting from positive instances ends at node i. This choice comes from previous studies, which have proven that such a feature matrix suffices to propagate information coded in the graph labels [17], that is the information coded in the features corresponding to random walks length k, with $k > 3$, is negligible.

Scoring non positive points through fuzzy clustering. Let $\mathcal{F} = \{\boldsymbol{F}_i | i \in V\}$ be the set of projected points, we denote by $\mathcal{F}_+ = \{\boldsymbol{F}_i | i \in S_+\}$ and $\mathcal{F}_- = \{\boldsymbol{F}_i | i \in S_-\}$ the sets of positive and negative projected points. Points in $\mathcal{F}_-$ are scored according to their relation w.r.t. a fuzzy clustering of the set $\mathcal{F}_+$ of positive points. Such clustering is computed through repeated applications of the FCM algorithm [3]. In order to automatically adjust the number of clusters, at each execution of FCM the number of clusters is changed, aiming at maximizing the *fuzzy silhouette* index [6], a fuzzy extension of the crisp silhouette index [12] expressly considering the fuzzy nature of the membership functions outputted by FCM. Being the space dimension fixed to 3, the computational complexity of each execution of the FCM algorithm (including the assessment through the fuzzy silhouette index) will be $\mathcal{O}(Ic^2|S_+|)$, denoting by I the number of iterations of the clustering algorithm. Since the number of positive instances in our context is very small, *a fortiori* also the number of clusters will be low, thus the overall step has a low computational complexity.

Once an optimal clusterization of size c has been found, we assign each point $x \in \mathcal{F}_-$ a *score* $\phi(x)$, obtained considering the maximum membership value of x to the various clusters, that is:

$$\phi(x) = \max_{1 \le k \le c} \left\{ \left(\sum_{j=1}^{c} \left(\frac{d(x, v_k)}{d(x, v_j)} \right)^{\frac{2}{\alpha-1}} \right)^{-1} \right\} \tag{3}$$

where $v_1, \ldots, v_c$ are the cluster centroids, while α and d respectively denote the fuzzification parameter and the distance function used by the FCM algorithm. Therefore, a non-positive point having a score lower than a fixed threshold $\tau > 0$ cannot be reasonably attributed to any cluster grouping positive points, whereas nodes corresponding to points $\mathcal{F}_{-,\tau} = \{x \in \mathcal{F}_- | \phi(x) \ge \tau\}$ are good candidates for the set S_-^p.

Even this step can be efficiently computed: the distance d of 3-feature vectors can be computed in $\mathcal{O}(1)$ time and the time complexity of computing $\phi(x)\ \forall x \in \mathcal{F}_-$ is $\mathcal{O}(|\mathcal{F}_-|)$, since $c \leq |S_+|$ and ${}^{|S_+|}/_{|S_-|} \ll 1$.

Selecting negative instances through local equilibrium. We denote by $S_{-,\tau} = \{i \in S_- | \boldsymbol{F}_i \in \mathcal{F}_{-,\tau}\}$ the set of negative nodes corresponding to points selected at the previous step. We consider the *COSNet* HN $H = \langle \boldsymbol{W}, k, \rho \rangle$ and estimate the optimal parameters $\hat{k}$, $\hat{\rho}$. Then, nodes in $S_{-,\tau}$ are further skimmed according to their stability in the labeled sub-network $H_S = \langle \boldsymbol{W}_S, \hat{k}, \hat{\rho} \rangle$, where $\boldsymbol{W}_S$ is the sub-matrix of connections between nodes in S. In particular, a node $i \in S_-$ is *at equilibrium* if $\beta_S(i) = \sum_{j \in S} W_{ij} \hat{l}_j - \hat{k} \leq 0$. Accordingly, we define $S_-^p = \{i \in S_{-,\tau} | \beta_S(i) > 0\}$. Nodes in S_-^p are thereby such that a possible network update of the sub-network H_S would change their state from negative to positive, thus our choice is coherent with both prior information ($\boldsymbol{W}$ and $\hat{\boldsymbol{l}}$) and network local stability.

4.3 Embedding negative selection into the Hopfield model

After selecting the set S_-^p, we run the dynamics of *COSNet* with labeled and unlabeled nodes defined as $\overline{S} = S \setminus S_-^p$ and $\overline{U} = U \cup S_-^p$, respectively. Labels of nodes in S_-^p are thereby not utilized in computing activation thresholds θ_i in (1); nevertheless, as such information has been considered by the learning procedure, to preserve consistency with the learning criterion, the initial state of neurons in S_-^P is set to $-\cos \hat{\rho}$. This choice allows to avoid the parameter relearning, since $-\cos \hat{\rho}$ is the value assigned to nodes in S_-^P by the learning procedure (see [9] for more details).

Finally, it is worth pointing out that, although the dynamics involves also nodes in S_-^P, the inference is still performed solely on nodes in U.

5 Experimental validation

We validate our approach in predicting the Gene Ontology (GO) functions (release 23-3-13) of the whole genome of the *S.cerevisiae* (a yeast) model organism. In order to predict more specific and unbalanced terms in the ontology, we selected GO Molecular Functions (MF) terms with 30-300 positive annotated genes. The connection network has been obtained by unweighted sum integration on the union genes of 16 networks downloaded from the GeneMANIA website[1] and covering different types of data, ranging from co-expression to physical interactions. The final network has a total of 5775 yeast genes and 127 MF terms.

5.1 Results

We compared our inference methodology with the state-of-the-art supervised and semi-supervised algorithms proposed in the literature for the *NLP* problem. In particular, we considered the *Support Vector Machine* (SVM) algorithm,

[1] `www.genemania.org`

Table 1. Average results. *COSNet-neg* is the our method. The best results are highlighted in bold, whereas methods that are significantly better than all the others according to the Wilcoxon signed-rank sum test ($\alpha = 0.05$) are underlined.

Algorithm	**AUC**	**AUPRC**	**F**
LP	0.947	0.525	0.033
RW	0.941	0.473	0.395
RF	0.602	0.075	0.059
SVM	0.576	0.052	0.025
COSNet-neg	**0.949**	**0.529**	**0.582**

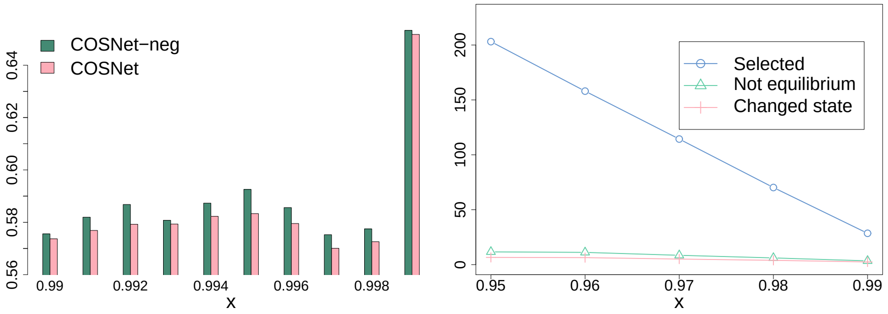
largely applied in computational biology, and more precisely its probabilistic version [13]; the *Random Forest* (RF) method [4]; a *Label Propagation* (LP) algorithm based on Gaussian random fields and harmonic functions [21]; the classical *random walk* (RW) algorithm without restart with at most 100 steps [15]. Moreover, we also considered the original version of *COSNet*, without negative selection, to evaluate the impact of our negative selection strategy on the performance of the model.

The generalization performances have been estimated through a 5-fold cross-validation procedure, and the performances have been assessed using the *Area Under the Precision-Recall Curve* (AUPRC), the *F-measure* (harmonic mean of precision and recall) and Area Under the ROC Curve (AUC). The AUPRC to some extent represents how close the classifier is to a perfect oracle, which would predict no false positives and have an AUC of 1. Indded, the AUC can be interpreted as the probability that a randomly chosen true positive will be ranked higher by the classifier than a randomly chosen true negative. Finally, the F measures the effectiveness of retrieval by taking into account both the precision and the recall of a classifier, that is the probability that a positive predicted is a true positive, and the proportion of true positives the classifier predicts, respectively. Usually, the F-measure is adopted to evaluate the performance in classifying positive instances in contexts where positives are rare.

The results averaged across the 127 GO terms are reported in Table 1. Our method achieves the best results in terms of all the adopted measures, with improvements statistically significant according to the Wilcoxon signed-rank test [19], except for the AUC results. The semi-supervised methods LP and RW have competitive performance in terms of AUC and AUPRC, where they perform as the second and the third best method respectively. Very poor results are obtained by the supervised algorithms SVM and RF, likely due to the fact that they are not well-suited for *NLP* problems with unbalanced labelings.

Finally, the results in terms of F show that our method largely outperforms the compared methodologies as classifier, where only the RW algorithm does not poorly perform w.r.t. remaining methods. F values close to 0 simply means that very few true positives have been predicted.

In order to evaluate the contribution of the negative selection strategy to the performance of the model, in Fig. 1(a) we report the comparison in terms of



(a) (b)

Fig. 1. (a) Comparison of *COSNet* and *COSNet-neg* in terms of average F. x is the selected percentile. (b) *Selected* corresponds to $|\mathcal{F}_{-,\tau}|$ averaged across selected classes, *Not equilibrium* to average $|S^p_-|$ and *Changed state* is the average number of nodes in S^p_- which have changed their initial state after the dynamics of the network.

F between *COSNet-neg* and *COSNet* (for AUC and AUPRC we have a similar behavior). We selected the parameter τ as the x-th percentile of $\phi(\mathcal{F}_-)$, and vary x from 0.99 to 0.999 with step 0.001. The negative selection improves the performance for every choice of τ, whereas the average results considerably vary with x, because the comparison is restricted to classes with at least one negative instance not at equilibrium (i.e. $|S^p_-| > 0$), and clearly the number of such classes varies with the selected percentile x. We chose just these classes for the comparison because when all the neurons in $\mathcal{F}_{-,\tau}$ are at equilibrium, *COSNet* and *COSNet-neg* are identical. After a tuning on a small subset of labeled data, in our experiments we set $x = 0.995$. Finally, to further analyze the negative selection procedure, in Fig. 1(b) we report, for different values of x, the average number of negative instances whose score (3) is greater that τ, the average number of those not at equilibrium (average $|S^p_-|$), and the average number of neurons in S^p_- whose initial and final state differ. As expected, the value of $|\mathcal{F}_{-,\tau}|$ increases linearly when τ decreases. Nevertheless, among neurons in $|\mathcal{F}_{-,\tau}|$, just a small proportion is not at equilibrium in the labeled sub-network, and such proportion decreases quickly when τ decreases, showing on the one hand that the non-equilibrium condition for belonging to S^p_- is highly discriminating, on the other hand that the method is quite robust to variations of parameter τ. Furthermore, among neurons not at equilibrium, a small subset changes the initial state during the dynamics of the unlabeled network, and this may be due to: 1) the unlabeled network tends to predict the proportion of positive instances in the training data, and since such proportion is small, it is expected that a relevant part of neurons in S^p_- is predicted as negative; 2) nodes in S^p_- are not at equilibrium when considering just labeled data, but when we complete the information with the unlabeled part, their initial condition of being negative is restored; 3) some of the negative instances selected for the set S^p_- can be associated to feature vectors characterized by noise, since prior information in this context is still affected by both experimental and biological noise [16].

6 Conclusions

In this work we propose a novel strategy for selecting negative examples in contexts where just positive associations can be considered as reliable. In particular, our methodology is designed explicitly for algorithms based on Hopfield networks, and can be applied in context where classes to be predicted are both structured in a hierarchy or hierarchy-less. The negative selection strategy, based on a fuzzy clustering procedure with low computational impact, has been embedded in the dynamics of a Hopfield network. Its effectiveness has been assessed through the comparison with state-of-the-art approaches in predicting the biological functions of the whole proteome of *S.cerevisiae* organism. The promising results encourage to further extend this approach, in particular for improving the negative selection by exploiting also the hierarchy of functional classes.

References

1. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. Nature genetics 25(1), 25–29 (2000)
2. Bertoni, A., Frasca, M., Valentini, G.: Cosnet: A cost sensitive neural network for semi-supervised learning in graphs. In: Machine Learning and Knowledge Discovery in Databases – European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I. LNAI, vol. 6911, pp. 219–234. Springer-Verlag (2011)
3. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. Computers & Geosciences 10(2), 191–203 (1984)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Burghouts, G.J., Schutte, K., Bouma, H., den Hollander, R.J.M.: Selection of negative samples and two-stage combination of multiple features for action detection in thousands of videos. Machine Vision and Applications 25(1), 85–98 (2014)
6. Campello, R.J.G.B., Hruschka, E.R.: A fuzzy extension of the silhouette width criterion for cluster analysis. Fuzzy Sets and Systems 157(21), 2858–2875 (2006)
7. Fagni, T., Sebastiani, F.: On the selection of negative examples for hierarchical text categorization. In: Proceedings of the 3rd Language & Technology Conference (LTC07). pp. 24–28 (2007)
8. Ferretti, E., Errecalde, M.L., Anderka, M., Stein, B.: On the use of reliable-negatives selection strategies in the PU learning approach for quality flaws prediction in wikipedia. In: Database and Expert Systems Applications (DEXA), 2014 25th International Workshop on. pp. 211–215 (2014)
9. Frasca, M., Bertoni, A., Re, M., Valentini, G.: A neural network algorithm for semi-supervised node label learning from unbalanced data. Neural Networks 43, 84–98 (2013)
10. Gomez, S.M., Noble, W.S., Rzhetsky, A.: Learning to predict protein-protein interactions from protein sequences. Bioinformatics 19(15), 1875–1881 (2003)

11. Hopfield, J.J.: Neural networks and physical systems with emergent collective compatational abilities. Proceedings of the national academy of sciences 79(8), 2554–2558 (1982)
12. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. J. Wiley & Sons, New York (1990)
13. Lin, H.T., Lin, C.J., Weng, R.C.: A note on Platt's probabilistic outputs for support vector machines. Machine Learning 68(3), 267–276 (2007)
14. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on. pp. 179–186 (2003)
15. Lovász, L.: Random walks on graphs: A survey. In: Combinatorics, Paul Erdős is Eighty, pp. 353–397 (1993)
16. Marshall, E.: Getting the noise out of gene arrays. Science 306(5696), 630–631 (2004)
17. Mostafavi, S., Goldenberg, A., Morris, Q.: Labeling nodes using three degrees of propagation. PLoS ONE 7(12), e51947 (2012)
18. Mostafavi, S., Morris, Q.: Using the gene ontology hierarchy when predicting gene function. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp. 419–427 (2009)
19. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics bulletin 1, 80–83 (1945)
20. Youngs, N., Penfold-Brown, D., Drew, K., Shasha, D., Bonneau, R.: Parametric bayesian priors and better choice of negative examples improve protein function prediction. Bioinformatics 29(9), tt10–98 (2013)
21. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML. pp. 912–919 (2003)