# Positive Unlabeled Link Prediction via Transfer Learning for Gene Network Reconstruction

Paolo Mignone and Gianvito Pio
paolo.mignone@uniba.it, gianvito.pio@uniba.it

Department of Computer Science
University of Bari Aldo Moro
Via Orabona, 4 - 70125 Bari (Italy)

**Abstract.** Transfer learning can be employed to leverage knowledge from a source domain in order to better solve tasks in a target domain, where the available data is exiguous. While most of the previous papers work in the supervised setting, we study the more challenging case of positive-unlabeled transfer learning, where few positive labeled instances are available for both the source and the target domains. Specifically, we focus on the link prediction task on network data, where we consider known existing links as positive labeled data and all the possible remaining links as unlabeled data. In many real applications (e.g., in bioinformatics), this usually leads to few positive labeled data and a huge amount of unlabeled data. The transfer learning method proposed in this paper exploits the unlabeled data and the knowledge of a source network in order to improve the reconstruction of a target network. Experiments, conducted in the biological field, showed the effectiveness of the proposed approach with respect to the considered baselines, when exploiting the *Mus Musculus* gene network (source) to improve the reconstruction of the *Homo Sapiens Sapiens* gene network (target).

## 1 Introduction

The link prediction task aims at estimating the probability of the existence of an interaction between two entities on the basis of the available set of known interactions, which belong to the same data distribution, the same context and described according to the same features. However, in many real cases, the identical data distribution assumption does not hold. For example, in the study of biological networks, collecting training data is very expensive and it is necessary to build link prediction models on the basis of data regarding different (even if related) contexts. At this regard, *transfer learning* strategies can be adopted to leverage knowledge from a source domain to improve the performance of a task solved on a target domain, for which we have few labeled data (see Figure 1).

In the literature, we can find several applications where transfer learning approaches have proved to be beneficial. For example, in the classification of Web documents, where the goal is to assign a category to a certain Web document, transfer learning approaches can be exploited to classify newly created Web

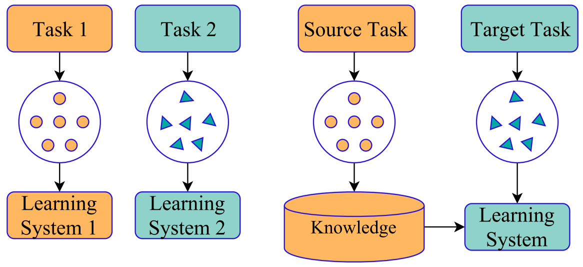

**Fig. 1.** Exploitation of the knowledge acquired on a source task to solve the target task (right-side), compared to solving two machine learning tasks independently (left-side).

sites which follow a different data distribution [3] (e.g., the content is related to new subtopics). Another example is the work proposed in [11], where the authors exploit transfer learning approaches in a situation where data become easily outdated. In particular, the authors aim to adapt a WiFi localization model trained in one time period (source domain) to a new time period (target domain), where the available data possibly follow a different data distribution.

Focusing on the link prediction task, in the literature we can find several works in the biological field, since biological entities and their relationships can be naturally represented as a network. In the specific field of genomics, recent studies have significantly relied on high throughput technologies and on computational methods, which led to an improved understanding of the working mechanisms in several organisms. Such mechanisms are usually modeled through gene interaction networks, where nodes represent genes and edges represent regulation activities. The direct observation of the real structure of these interaction networks would require expensive in-lab experiments. Since gene expression data are easy to obtain, several methods have been proposed in the literature that exploit this kind of data [9]. These approaches analyze the expression level of the genes under different conditions (e.g., with a specific disease or after a treatment with a specific drug) or, alternatively, under a single condition in different time instants. Therefore, most machine learning approaches aiming to solve the link prediction task generally analyze gene expression data. In this context, the goal is to reconstruct the whole network structure (Gene Network Reconstruction - GNR), providing the biologists with a general overview of the interactions among the genes. However, while existing methods generally work effectively on sufficiently large training data, a transfer learning approach could favor the GNR of specific organisms which are not well studied, by exploiting the knowledge acquired about different, related organisms.

The main contribution of this paper is to evaluate the possible benefits that transfer learning techniques can provide to the task of link prediction. In particular, we exploit the available information about a source network for the reconstruction of a target network with poor available data. Moreover, we study the more challenging case of Positive-Unlabeled (PU) transfer learning, where few positive labeled examples are available for both the source and the target domains, and no negative example is available. PU learning setting holds in many real context (e.g., text categorization [8], bioinformatics [5]) where it is very expensive or unfeasible to obtain negative examples for the concept that we intend to model. As described in [16], PU learning methods can be divided into three classes: *a)* the first, called two-step strategy, tries to identify some reliable negative examples in the unlabeled data, and then applies supervised learning algorithms; *b)* the second assigns different weights to positive and unlabeled examples, by estimating the conditional probability of an example of being positive; *c)* the third just treats the unlabeled data as highly noisy negative data.

In this paper, we consider the link prediction task as a PU learning task of class *b)*, that, according to previous studies [2, 13], allows us to avoid the strong assumptions about the negative examples made by methods relying on classes *a)* and *c)*. In particular, we propose a link prediction method which aims at building a binary classifier for all the possible links, where each link $\langle v', v'' \rangle$ between two nodes $v'$ and $v''$ is represented as the concatenation of the feature vectors of $v'$ and $v''$. The training set is built by considering the set of vectors associated to known (i.e., validated) links as positive examples and the vectors associated to all the possible remaining links, excluding self-links, as unlabeled examples. Methodologically, in a first stage we build a clustering model for the source domain and a clustering model for the target domain. In both cases, this is performed only on the positive labeled examples in order to catch several different viewpoints of the underlying concept of positive interactions. In a second stage, the unlabeled data of both source and target domains are weighted according to the similarities with respect to the clusters' centroids. In a third stage, we exploit the positive examples and all the (weighted) unlabeled examples, coming from both the source and the target domains, by training a classifier which is able to handle weights on instances. According to [10][14], our method belongs to the category of *homogeneous* transfer learning approaches, where the source and target domains are described in the same feature space, with possibly different data distributions. This setting is in contrast with the *heterogeneous* transfer setting, which assumes different feature spaces.

In order to evaluate the performance of the proposed method, we performed experiments in the biological domain. In particular, our experiments focused on the reconstruction of the human (Homo Sapiens Sapiens) gene network guided by the gene network of another, related organism, i.e., the mouse (Mus Musculus).

In Section 2, we describe in details our method, while in Section 3 we show the experimental evaluation and report some comments about the results. Finally, in Section 4, we draw some conclusions and outline possible future works.

## 2 The proposed method

In this section, we describe our transfer learning approach to solve link prediction tasks in network data. Before describing it in details, we introduce some useful notions and formally define the link prediction task for a single domain. Let:

- $V$ be the set of nodes of the network;

- $x = \langle v', v'' \rangle \in (V \times V)$ be a (possible) link between two nodes $v'$ and $v''$, where $v' \neq v''$;
- $e(v) = [e_1(v), e_2(v), \ldots, e_n(v)]$ be the vector of features related to the node $v$, where $e_i(v) \in \mathbb{R},\ \forall i \in \{1, 2, \ldots, n\}$;
- $e(x) = [e_1(v'), e_2(v'), \ldots, e_n(v'), e_1(v''), e_2(v''), \ldots, e_n(v'')]$ be the vector of features related to the link $x = \langle v', v'' \rangle$;
- $sim(a, b) \in [0, 1]$ be a similarity function between the vectors $a$ and $b$;
- $l(x)$ a function that returns 1 if the link $x$ is a known existing, and 0 if its existence is unknown;
- $L = \{x \mid x \in (V \times V) \wedge l(x) = 1\}$ be the set of labeled links;
- $U = (V \times V) \setminus L$ be the set of unlabeled links;
- $D = \{\widetilde{X}, P(X)\}$ be the domain described by the feature space $\widetilde{X} = \mathbb{R}^{2n}$, with a specific marginal data distribution $P(X)$, where $X = L \cup U$;
- $w(x)$ $(0 \leq w(x) \leq 1)$ be a computed weight for the link $x \in U$;
- $f(x)$ be an ideal (target) function which returns 1 if $x$ is an existing link, and 0 otherwise.

The task we intend to solve is then defined as follows:
*Given:* a set of training examples $\{\langle e(x), w(x) \rangle\}_x$, each of which described by a feature vector and a weight;
*Find:* a function $f' : \mathbb{R}^{2n} \rightarrow [0, 1]$ which takes as input a vector of features $e(x)$ and returns the probability that the link $x$ exists. Therefore, $f'(e(x)) \approx \mathcal{P}(f(x) = 1)$ or, in other terms, $f'$ approximates the probability distribution over the values of the ideal function $f$.

Our method works with two different domains: the source domain $D_s = \{\widetilde{X_s}, P(X_s)\}$, and the target domain $D_t = \{\widetilde{X_t}, P(X_t)\}$. We remind that our method works with homogeneous feature spaces, that is $\widetilde{X_s} = \widetilde{X_t}$, while the marginal data distributions is generally different, that is $P(X_s) \neq P(X_t)$.

Given the two sets of labeled examples $L_s$ and $L_t$, regarding the source and the target domain respectively, the method consists of three stages, that are summarized in Figure 2 and detailed in the following subsections.

**Stage I - Clustering.** The first stage of our method consists in the identification of a clustering model for the positive examples of each domain (i.e., on $L_s$ and $L_t$). The application of a clustering method is motivated by the necessity to distinguish among possible multiple viewpoints of the underlying concept of positive interactions. Moreover, a summarization in terms of clusters' centroids becomes useful also from a computational viewpoint, since in the subsequent stages we can compare centroids instead of single instances. In this paper, we adopt the classical *k-means* algorithm, since it is well established in the literature. However, any other prototype-based clustering algorithm, possibly able to catch specific peculiarities of the data at hand, could be plugged into our method.

**Stage II - Instance Weighting.** Although an unlabeled link could be either a positive or a negative example, we consider all the unlabeled examples as positive examples and compute a weight representing the degree of certainty in $[0, 1]$ of

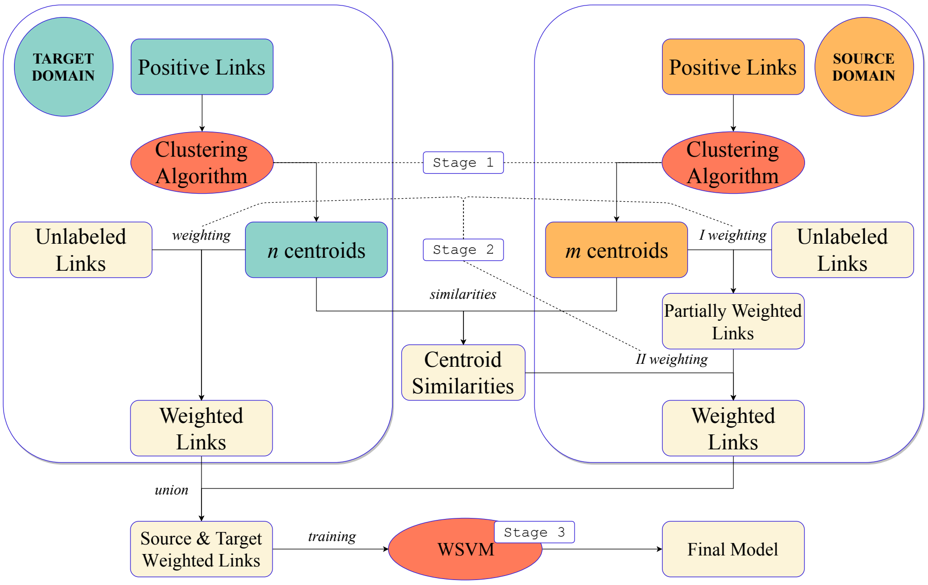

**Fig. 2.** A graphical overview of the proposed transfer learning approach.

being a positive example: a value close to 0 means that the example is likely to be a negative example, while a value close to 1 means that the example is likely to be a positive example. The weight associated to the unlabeled instances of both the source and the target domains are computed according to their similarities with respect to the centroids obtained in the first stage. In particular, we identify a different weighting function for the source and target domains, in order to smooth the contribution provided by instances coming from the source domain.

Specifically, an unlabeled link $x$ belonging to the target network (i.e., $x \in V_t \times V_t$) is weighted according to its similarity with respect to the centroid of its closest cluster, among the clusters identified from the target network. Formally:

$$w(x) = max_{c_t \in C_t}(sim(e(x), c_t)), \tag{1}$$

where $C_t$ are the clusters identified from positive examples of the target network.

On the other hand, an unlabeled link $x'$ belonging to the source network (i.e., $x' \in V_s \times V_s$) is weighted by considering two similarity values: *i)* the similarity with respect to the centroid of its closest cluster, computed among the clusters identified from the source network, and *ii)* the similarity between such a centroid and the closest centroid identified on the target network. Formally, let $c' = argmax_{c_s \in C_s}(sim(e(x'), c_s))$ be the closest centroid with respect to $x'$ among the possible centroids $C_s$ identified in the source network. Then:

$$w(x') = sim(e(x'), c') \cdot max_{c'' \in C_t}(sim(c'', c')). \tag{2}$$

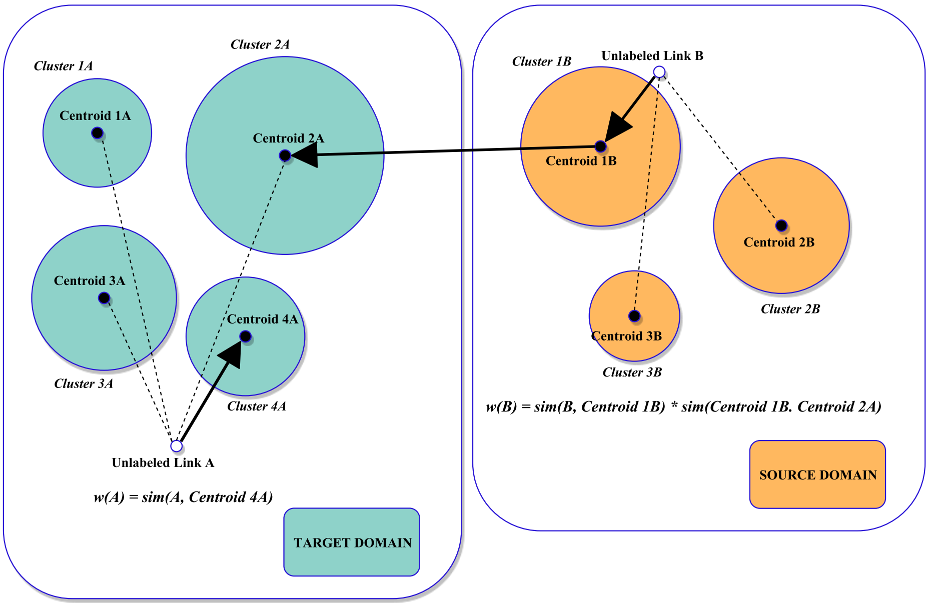

**Fig. 3.** Example of unlabeled link weighting process.

As a similarity function, we exploit the Euclidean distance, after applying a min-max normalization (in the range $[0, 1]$) to all the features of the feature vectors. Formally, $sim(e(x'), e(x'')) = 1 - \sqrt{\sum_{k=1}^{n} \left(e_k(x') - e_k(x'')\right)^2}$. An overview of the weighting strategy can be graphically observed in Figure 3.

**Stage III - Training the classifier.** In the third stage, we train a probabilistic classifier, based on linear Weighted Support Vector Machines (WSVM) [15] with Platt scaling [1], from the weighted unlabeled instances coming from both the source and the target networks. We selected an SVM-based classifier mainly because *i)* it has a (relatively) good computational efficiency, especially in the prediction phase, and *ii)* it already proved to be effective (with Platt scaling) in the semi-supervised setting [4]. At the end of the training phase, the WSVM classifier produces a model in the form of an hyperplane function $h$. In the specific PU setting, while $h$ is not class discriminatory, we can consider as positive examples those appearing close to the identified hyperplane[1]. At this respect, by exploiting the Platt scaling, for each unlabeled link $x$, we compute the probability of being a positive example as $f'(e(x)) = \frac{1}{1+e^{-h(e(x))}}$, where $h(e(x))$ is the score obtained by the learned WSVM. Finally, we rank all the predicted links in descending ordering with respect to the their probability of being positive. A pseudo-code representation of the proposed method is shown in Algorithm 1.

[1] The less the distance between an unlabeled example and the hyperplane, the higher the probability of the existence of the link.

**Algorithm 1:** PU Link Prediction via Transfer Learning

```
Data:
    ·$L_s = \{x_s \mid x_s \in (V_s \times V_s) \wedge l(x_s) = 1\}$: positive links of the source network
    ·$U_s = (V_s \times V_s) \setminus L_s$: unlabeled links of the source network
    ·$L_t = \{x_t \mid x_t \in (V_t \times V_t) \wedge l(x_t) = 1\}$: positive links of the target network
    ·$U_t = (V_t \times V_t) \setminus L_t$: unlabeled links of the target network
    ·$e(x)$: feature vector of the link $x$
    ·$sim(e(x'), e(x'')) = 1 - \sqrt{\sum_{k=1}^{n} (e_k(x') - e_k(x''))^2}$
    ·$k_1, k_2$: number of positive clusters for the source and the target network, respectively
Result:
    ·$ranked\_links$: predicted links ordered according to their likelihood
begin
    $C_s \leftarrow kmeans(L_s, k_1)$; $C_t \leftarrow kmeans(L_t, k_2)$;
    $source\_training\_set \leftarrow \emptyset$; $target\_training\_set \leftarrow \emptyset$; $ranked\_links \leftarrow \emptyset$;
    foreach $x_t \in U_t$ do
        $w(x_t) \leftarrow max_{(c_t \in C_t)}(sim(e(x_t), c_t))$;
        $target\_training\_set \leftarrow target\_training\_set \cup \{\langle e(x_t), w(x_t) \rangle\}$;
    foreach $x_s \in U_s$ do
        $source\_centroid \leftarrow argmax_{(c_s \in C_s)}(sim(e(x_s), c_s))$;
        $partial\_weight \leftarrow sim(e(x_s), source\_centroid)$;
        $centroid\_sim \leftarrow max_{(c_t \in C_t)}(sim(source\_centroid, c_t))$;
        $w(x_s) \leftarrow partial\_weight \cdot centroid\_sim$;
        $source\_training\_set \leftarrow source\_training\_set \cup \{\langle e(x_s), w(x_s) \rangle\}$;
    $training\_set \leftarrow source\_training\_set \cup target\_training\_set$;
    $h(\cdot) \leftarrow WSVM(training\_set)$;
    $f'(\cdot) = \frac{1}{1+e^{-h(\cdot)}}$;
    foreach $x \in U_t$ do
        $ranked\_links \leftarrow ranked\_links \cup \{\langle x, f'(e(x)) \rangle\}$;
    $ranked\_links \leftarrow sort\_by\_score(ranked\_links)$;
    return $ranked\_links$;
```

## 3 Experiments

Our experiments have been performed in the biological field. In particular, the specific task we intend to solve is the reconstruction of the human (*Homo Sapiens Sapiens - HSS*) gene network. As a source task, we will exploit the gene network of another, related organism, i.e., the mouse (*Mus Musculus - MM*).

### 3.1 Dataset

The considered dataset consists of gene expression data related to specific organs. In particular, we analyzed the gene expression levels of 6 organs (lung, liver, skin, brain, bone marrow, heart), obtained by the samples available at Gene Expression Omnibus (GEO), a public functional genomics repository. On overall, 161 and 174 samples were considered respectively for MM and HSS.

All the samples of each organism were processed according to the data acquisition workflow adopted for the DREAM5 challenge [9]. In particular, we processed the samples through the *Affymetrix Expression Console Software*, which led to produce a dataset consisting of a total of 45, 101 genes over the 161 samples for MM and a dataset of 54, 675 genes over the 174 samples for HSS.

Although we originally had a different number of features for the considered organisms, we built two homogeneous datasets by aggregating the features according to the organs. In particular, for each organisms, we represented their genes by means of 6 features (one for each organ), by averaging the expression levels measured within the same organ. Accordingly, the datasets representing the interactions among genes were built by considering all the possible pairs of genes, i.e., by concatenating the feature vectors associated to the genes, leading to 12-dimensional feature vectors. The set of validated gene interactions was extracted from BioGRID[2], which is an interaction repository containing data compiled through comprehensive curation efforts. This set represents our ground truth in terms of positive links. As for the unlabeled instances, we performed a random sampling without replacement from all the other possible links involving at least one gene that appears in the BioGRID ground truth. This procedure led us to build a balanced dataset between positive and unlabeled examples.

### 3.2 Experimental Setting

Since our method exploits the $k$-means clustering algorithm, we performed the experiments with different values for $k_1$ (i.e., the number of clusters for the MM organism) and $k_2$ (i.e., the number of clusters for the HSS organism), in order to evaluate the possible effect of such parameters on the results. In particular, we considered the following parameter values: $k_1 \in \{2, 3\}, k_2 \in \{2, 3\}$.

We remind that we work in the PU learning setting (i.e., the dataset does not contain any negative example). Therefore, inspired by the experiments performed in [12], we evaluated the results in terms of recall@$k$. The adoption of this measure allows us to avoid the estimation of possible negative examples in the ground truth, which could lead to a wrong evaluation of the results. In particular, in order to quantitatively compare the obtained results, we draw the recall@$k$ curve, by varying the value of $k$, and compute the area under the curve.

The experiments have been performed according to the 10 fold cross validation (10 fold CV) on the positive labeled data. In particular, for each iteration of the 10 fold CV, we considered: *a)* a portion of the positive labeled data (9 out of 10 folds) to build the clustering models through $k$-means; *b)* all the unlabeled data and a portion of positive labeled data (1 out of 10 folds) as test set.
We compared our method, indicated as **transfer**, with approaches:
- **no_transfer**, which corresponds the WSVM with Platt scaling learned only from the target network (i.e., from the HSS network). This baseline allows us to evaluate the contribution of the source domain.
- **union**, which is the WSVM with Platt scaling learned from a single dataset consisting of the union of the instances coming from both MM and HSS. This baseline allows us to evaluate the effect of our weighing strategy.

Since we are interested in observing the contribution provided by our approach with respect to the non-transfer approach, results will be evaluated in terms of improvement with respect to the **no_transfer** baseline.

[2] https://thebiogrid.org

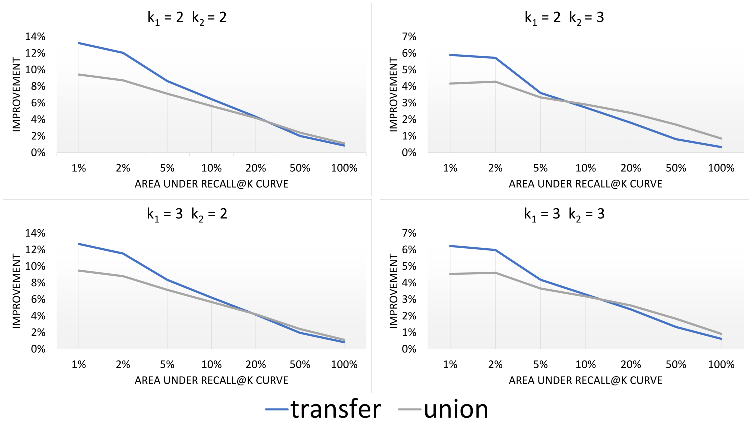

**Fig. 4.** Improvement over no_transfer, with different values of $k_1$ and $k_2$.

### 3.3 Results

In Figure 4, we show the results obtained with different values of $k_1$ and $k_2$. In particular, we considered different percentages of the recall@$k$ curve and measured the area under each sub-curve. This evaluation is motivated by the fact that biologists usually focus their in-lab studies on the analysis of the top-ranked predicted interactions. Therefore, a better result in the first part of the recall@$k$ curve (i.e., at 1%, 2%) appears to be more relevant for the real biological application. The graphs show that both the **union** baseline and the proposed method (**transfer**) are able to obtain a better result with respect to the variant without any transfer of knowledge (**no_transfer**). This confirms that, in this case, the external source of knowledge (the MM gene network) can be exploited to improve the reconstruction of the target network (the HSS gene network).

By comparing the **union** baseline with our method, we can observe that the proposed weighting strategy was effective in assigning the right contribution to each unlabeled instance (coming either from the source or from the target network) in the learning phase. This is even more evident in the first part of the recall@$k$ curve, where our method was able to retrieve about 120 additional true interactions at the top 1% of the ranking with respect to the baseline approaches.

Finally, by analyzing the results with respect to the values of $k_1$ and $k_2$, we can conclude that the highest improvement over the baseline approaches has been obtained with $k_1 = 2$ and $k_2 = 2$. This means that clustering can affect the results, and that even higher improvements could be obtained by adopting smarter clustering strategies that can, for example, catch and exploit the distribution, in terms of density, of the examples in the feature space.

## 4 Conclusion and Future Work

In this paper, we proposed a transfer learning method to solve the link prediction task in the PU learning setting. By resorting to a clustering-based strategy, our method is able to exploit unlabeled data as well as labeled and unlabeled data of a different, related domain, identifying a different weight for each training instance. Focusing on biological networks, we evaluated the performance of the proposed method in the reconstruction of the Human gene network, supported by the knowledge about the mouse gene network. Results show that the proposed method was able to improve the accuracy of the reconstruction, if compared to two baseline approaches. As future work, we plan to implement a distributed version of the proposed method, and to adopt some ensemble-based approaches [6, 7] to exploit multiple clusters in the prediction. We also plan to perform an extensive comparison with state-of-the-art methods in the biological field.

## Acknowledgments

We would like to acknowledge the European project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (ICT-2013-612944).

## References

1. J. c. Platt. Probabilistic outputs for support vector machine and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, 1999.
2. M. Ceci, G. Pio, V. Kuzmanovski, and S. Džeroski. Semi-supervised multi-view learning for gene network reconstruction. *PLOS ONE*, 2015.
3. W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for transfer learning. In *Proc. of ICML*, pages 193–200, 2007.
4. C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proc. of ACM SIGKDD*, pages 213–220, 2008.
5. G. Jowkar and E. Mansoori. Perceptron ensemble of graph-based positive unlabeled learning for disease gene identification. *Comput.Biol.Chem*, 64:263–270, 2016.
6. J. Levatic, M. Ceci, D. Kocev, and S. Dzeroski. Self-training for multi-target regression with tree ensembles. *Knowl.-Based Syst.*, 123:41–60, 2017.
7. J. Levatic, D. Kocev, M. Ceci, and S. Dzeroski. Semi-supervised trees for multi-target regression. *Inf. Sci.*, 450:109–127, 2018.
8. B. Liu, W. S. Lee, P. S. Yu, and X. Li. Partially supervised classification of text documents. In *Proc. of ICML*, pages 387–394, 2002.
9. D. Marbach and J. C. et al. Wisdom of crowds for robust gene network inference. *Nature Methods*, 2016.
10. S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.
11. S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu. Transfer learning for wifi-based indoor localization. *Workshop on Transfer Learning for Complex Task AAAI*, 2008.
12. G. Pio, M. Ceci, D. Malerba, and D. D'Elia. ComiRNet:a web-based system for the analysis of miRNA-gene regulatory networks. *BMC Bioinform*, 16(S-9):S7, 2015.
13. G. Pio, D. Malerba, D. D'Elia, and M. Ceci. Integrating microRNA target predictions for the discovery of gene regulatory networks: a semi-supervised ensemble learning approach. *BMC Bioinform*, 15(S-1):S4, 2014.
14. K. R. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *J. Big Data*, 3:9, 2016.
15. X. Yang, Q. Song, and Y. Wand. A weighted support vector machine for data classification. In *Int J Pattern Recogn, Vol. 21*, 2007.
16. B. Zhang and W. Zuo. Learning from positive and unlabeled examples: A survey. In *ISIP / WMWA*, pages 650–654, 2008.