Link Prediction in Heterogeneous Social Networks

Sumit Negi
Indian Institute of Technology, Delhi
India
negsumit@gmail.com

Santanu Chaudhury
Indian Institute of Technology, Delhi
India
santanuc@ee.iitd.ac.in

ABSTRACT

A *heterogeneous* social network is characterized by multiple *link types* which makes the task of link prediction in such networks more involved. In the last few years *collective link prediction* methods have been proposed for the problem of link prediction in heterogeneous networks. These methods capture the correlation between different types of links and utilize this information in the link prediction task. In this paper we pose the problem of link prediction in heterogeneous networks as a *multi-task, metric learning* (MTML) problem. For each link-type (relation) we learn a corresponding distance measure, which utilizes both network and node features. These *link-type* specific distance measures are learnt in a coupled fashion by employing the Multi-Task Structure Preserving Metric Learning (MT-SPML) setup. We further extend the MT-SPML method to account for task correlations, robustness to non-informative features and non-stationary degree distribution across networks. Experiments on the Flickr and DBLP network demonstrates the effectiveness of our proposed approach vis-à-vis competitive baselines.

1. INTRODUCTION

Predicting link formation between entities is a fundamental problem in social networks. While link prediction is a difficult problem in itself this problem becomes even more involved when multiple entities and heterogeneous interactions[1] are involved. Networks of this type are referred to as *heterogeneous networks*[2]. The complexity due to structural dependency and heterogeneity of links produces obstacles for link prediction in such networks. Well-known topological features designed for homogeneous networks cannot be applied in such situations. Link prediction in these networks has typically been performed by "*treating all relationships equally or by separately studying homogeneous projections of the networks and ignoring dependency patterns across link types*" [7]. Both these approaches are suboptimal and result in a loss of information. Different edge types may have a different topology or link formation mechanisms but nonetheless may be related and mutually influential. *Collective link prediction* methods [5], which have been successfully applied for link prediction in heterogeneous networks, try to quantitatively capture the correlation between different link types and apply this for the link prediction task.

In this paper we formulate the link prediction problem as a *multi-task*, *metric learning* problem[3]. Our end goal is to learn a distance measure for each link-type and use the learnt measure for the corresponding link prediction task. To achieve this the heterogeneous social network is first decomposed into individual homogeneous networks, where each homogeneous network captures interactions of a single type[4]. To illustrate this through an example consider the Flickr[5] network. Users on Flickr can post images, tag them with descriptive keywords, create and join special-interest groups, befriend other users etc. One can represent Flickr as a heterogeneous network composed of several entity types: *users*, *images*, and *groups*, with links between the entities representing different types of relations. A link between users denotes a friendship, a link between a user and a group denotes user's membership in the group etc. For our purpose we decompose the Flickr network into the following three interaction networks namely *User-User*, *User-Flickr Group*, *Flickr Group-Image*. Each network captures interactions of a single type. For instance, the *User-Flickr Group* interaction network encodes the relation *isMemberOf* between *User* and *Flickr Group* entities. Similarly the *User-User* interaction network encodes any user-to-user interaction e.g. friendship and/or other social interactions (e.g. *comments*,*likes* etc). On top of each of these interactions networks we define a corresponding metric learning task (referred to as the *link-type metric learning* task). More precisely, the *link-type metric learning* task is defined as the problem of learning a distance measure between entities of the corresponding interaction network. These distance measures are referred to as the *link-type distance measure*. In the Flickr example, three *link-type distance measures* are learnt, one for each network. To ensure that during the (metric) learning process correlation between different link types is utilized we employ the multi-task learning framework. The multi-task learning framework ensures that the correlations between different link types is modelled and utilized during the link-type metric learning stage. At test time these (learnt) metrics are used to predict the formation of a given link-type between corresponding entities. We illustrate these steps in Figure 1.

[1] we use the terms *link types*,*edge types*, *interaction type* interchangeably

[2] Example, the DBLP network contains *conferences*, *papers*, and *authors* as nodes/entities, with links-types such as *co-author*, *author-writes-paper*, *paper-published-in-conference* etc

[3] In a multi-task, metric learning setup given a set of related tasks, one learns a metric for each task in a coupled fashion in order to improve the performance on individual tasks.

[4] We refer to such networks as *interaction networks*

[5] https://www.flickr.com/

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983722

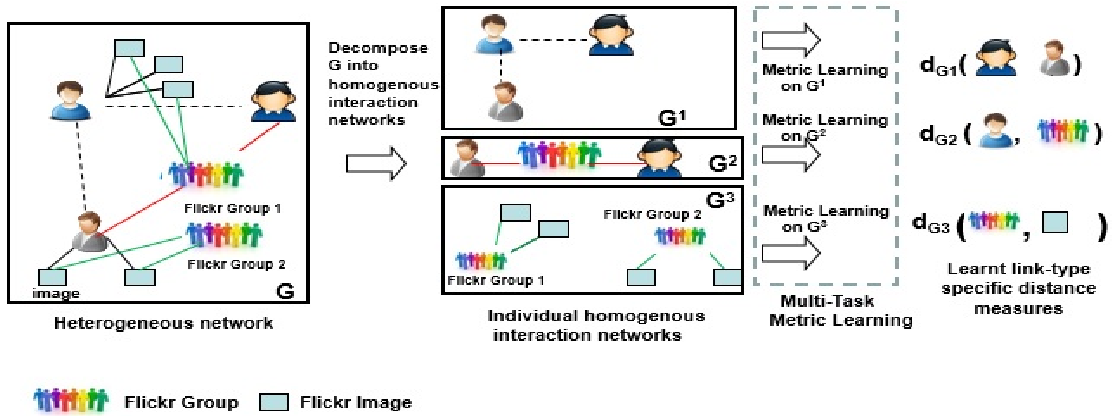

Figure 1: Decompose a heterogeneous network into homogeneous networks (where each network captures interactions of a single type) and then jointly learn *link-type distance measures* for each network. The learnt metrics are then used for the corresponding link prediction task. For example the learnt metric d_{G1} is used to predict which two (existing or new) users are likely to form a link.

This paper is structured in the following way. In Section 2 we describe related work in the area of link prediction and metric learning. Main contributions are detailed in Section 2.4. Notations are introduced in Section 3 followed by a brief description of the structure preserving metric learning approach in Section 4. Our approach is described in Section 5, Section 6 followed by experimental evaluation of the proposed method in Section 7. Conclusion is presented in Section 8.

2. RELATED WORK

In this section we begin by reviewing prior work related to link prediction in both homogenous and heterogeneous networks. This is followed by a brief overview of the metric learning literature.

2.1 Link Prediction in Homogenous Networks

Link prediction in homogeneous information networks has been extensively studied in recent years. Prior work in this space can be broadly categorized as : (1) *unsupervised approaches* which focus on proposing different similarity measures, either based upon graph topology or node attributes [1], [25] etc. (2) *supervised approaches* which exploit different features from the training set and use supervised learning models to predict the potential existence of links [40]. The unsupervised approaches can further be divided into two main categories: *local neighbor* based predictors and *global path* based predicator. Local neighbor based predictors utilize local neighborhood information such as Preferential Attachment Index (PA) [3], Common Neighbor, Adamic/Adar Index [1], Resource Allocation Index [9], Parameter-Dependent [45] etc. In addition to the local neighbor based predictors, many other predictors based on paths in the network have also been proposed to measure the proximity among users. These predictors use features such as Shortest Path, Katz [18], Relation Strength Similarity [6], proximity measures based on Random Walk [12], [2] such as Hitting Time, Commute Time, Random Walk with Restart, SimRank [16], Rooted PageRank (RPR) [25], PropFlow [26] etc. Based on theses features, internal attributes, and external information, many learning (supervised) based link prediction methods have been proposed. These learning based methods can further be be divided into feature-based classification [41], [34] probabilistic graph model [20], [24] and matrix factorization [31] based methods.

2.2 Link Prediction in Heterogenous Networks

Prominent work in the area of link prediction in heterogeneous networks include [7], [28], [5] and [42]. Author's in [7] proposed an unsupervised extension of the Adamic/Adar method to predict heterogeneous relationships in multi-relational networks. Specifically, the proposed multi-relational link prediction (MRLP) method applies a weighting scheme for different edge type combinations. The weights are determined by counting the occurrence of each unique 3-node sub-structure in the network, traditionally termed a triad census. [37] proposed a path-based relationship prediction model to study the co-authorship prediction problem in heterogeneous bibliographic networks. First, the meta path-based topological features are symmetrically extracted from the network using measures such as path count and random walk, around the given meta paths. The meta path captures the composition relation over the heterogeneous networks. Logistic regression is then used to learn the weights associated with different topological features that best predict co-author relationships. In [28] author's design a supervised learning framework that uses a variety of path based features from multiple sources to learn the dynamics of social networks. Recently MRIP or Multi-Relational Influence Propagation (MRIP) [42] was introduced for link prediction in heterogeneous networks. MRIP captures the correlation between different types of links for the link prediction problem. Author's in [5] propose a method where several related link prediction tasks are jointly learned. They propose a bayesian framework for solving the collective link prediction problem, which allows knowledge to be adaptively transferred across heterogeneous tasks.

2.3 Metric Learning

Metric learning has been shown to be effective for classification, clustering and ranking. Metric learning algorithms typically work by optimizing a target distance under various types of constraints (e.g. rank constraints, "must-link" , or "cannot-link"). One class of distance functions that has shown good generalization properties is the Mahalanobis distance. Metric learning has been applied to problems as diverse as link prediction in networks [35], state representation in reinforcement learning [38], music recommendation [29], partitioning problems [21], identity verification [4], web-page archiving [22] etc. Metric learning methods have also been successfully applied in the area of *Computer Vision* for tasks such as image classification [32], object recognition [13], [39], face recognition [14], visual tracking [17], *Information Retrieval* [19], [23] etc. Unfortunately, most metric learning algorithms are not easily applicable when the input is a network instead of points with class labels. This limitation was addressed in [35] where author's proposed SPML *Structure Preserving Metric Learning* - an algorithm for learning a Mahalanobis distance metric from a network such that the learned distances are tied to the inherent connectivity structure of the network. Details of the SPML method and a recently proposed multi-task extension to SPML (referred to as MT-SPML [11]) is described in Section 4.

2.4 Our Contributions

While SPML and its variants (MT-SPML) have been successfully applied for link prediction we observe the following limitations

- Recent work in the metric learning literature [27] indicates that when the input data contains a large portion of non-informative features, existing metric learning methods fail to identify relevant features thus resulting in performance degradation; MT-SPML or SPML are no different. The expectation from a *robust metric* learning method would be that such non-informative features would be automatically detected and their effect suppressed by the metric learning algorithm. Authors in [27] introduce the idea of *Robust Metric Learning* and propose a robust extension to the *metric learning to rank* [30] algorithm. The proposed method imposes a group sparsity penalty on the learned metric. We extend the same idea to the MT-SPML setup by applying a mixed-norm regularization over the learned metric. Our experimental evaluations show that doing so improves the quality of the learnt distance metric, thus positively impacting the link prediction performance metrics.

- As discussed previously individual *link-type distance measures* are learnt in a coupled fashion by posing this as a multi-task (metric) learning problem. An important consideration in a multi-task setup is how task relationships are modeled. In practice some of the tasks might be unrelated (or uncorrelated). Making a simplifying assumption that all tasks are related might impair the performance of other tasks [43]. Take for instance the Flickr example where we are (jointly) learning three different *link-type distance measures*. It is quite possible that the formation of links in the *user-user* network is completely uncorrelated to the link creation in the *flickr_group-image* network. If one naively assumes that all tasks are related this could negatively impact all the metric learning tasks. Considering this, a key requirement would be to automatically learn these correlations (or lack of it) from the data itself. Unlike MT-SPML, our method can automatically infer such task relationships from the data itself thus demonstrating better performance and robustness.

- The degree distribution in the individual interaction network may differ across networks. For example, the *user-user* network has a very different degree distribution as compared to say the *user-flickr_group* network - a user will connect to only a few other users in the network, but a flickr_group could have large number of members/users. Moreover, Flickr Groups with general themes (e.g. art, nature) will have larger group membership as compared to Flickr Groups with a very niche focus. Considering this the multi-task, metric learning setup needs to model for the unique (degree distribution) characteristic of each individual network. The method proposed in this paper incorporates the idea of *Degree Distributional Metric Learning* [35] into the overall multi-task metric learning setting. Doing so allows us to learn how degree distribution depends on node attributes thus yielding more accurate link predictions as shown through our experiments.

3. DEFINITION AND NOTATIONS

In this section we define some common notations that are used through out the paper. Additional notations are introduced in respective sections of the paper. A social network is *heterogeneous* if it contains multiple type of entities and links. We use ℓ to denote the different kinds of entities present in a heterogeneous network. As discussed earlier in the paper for our purpose we decompose the heterogeneous network $G = (V, E)$ into multiple homogeneous interaction networks i.e. G = $\{G^1, G^2, \ldots, G^Q\}$, where each individual network G^q encodes interactions of a specific link-type between a given pair of entity types ($e1$, $e2$). The vertex set and edge set of G^q is denoted as $V^q_{(e1,e2)}$ and $E^q_{(e1,e2)}$. Let n_1 and n_2 denote the vertex count for entity $e1$ and $e2$ respectively. For example consider YouTube, a popular video sharing social multimedia platform. One could model YouTube's heterogeneous network as a collection of users and videos (entities in the heterogeneous network) i.e. ℓ = {"*user*","*video*"}. For illustration, this network could be decomposed into two homogeneous networks namely G^1 and G^2, where $V^1_{(user,user)}$ and $V^2_{(user,video)}$ represents the vertex set for network G^1 and G^2 respectively. The edge set of G^1 and G^2, represented as $E^1_{(user,user)}$ and $E^2_{(user,video)}$ respectively, encodes user-user[6] and user-video interactions [7]. We model each of the homogeneous networks i.e. $G^q = (V^q_{(e1,e2)}, E^q_{(e1,e2)})$[8] as a $d \times n$ and $n_1 \times n_2$ matrix, where n=n_1 + n_2. Here n_1, n_2 refer to the number of nodes in G^1 and G^2 respectively. $V^q_{(e1,e2)} \in \Re^{d\times n}$ represents the node attributes (n nodes in d dimension) and $E^q_{(e1,e2)}$ is represented through a $n_1 \times n_2$ binary adjacency matrix A^q. Entry A^q_{ij} indicates the linkage information between node i and node j i.e. A^q_{ij} = 1 if node i and node j are linked. As discussed in the previous section our end objective is to learn a distance metric for each link type. We achieve this by learning a Mahalanobis distance, which is parameterized by a positive semi-definite (PSD) matrix M[9] $\in \Re^{d\times d}$, for each link type. The corresponding distance function is defined as $D_M(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j)$.

4. SPML

As explained in the previous sections (also see Figure 1) after the heterogeneous network has been decomposed into individual interaction networks a corresponding *link-type distance measure* is learnt for each of the individual networks. Different from i.i.d[10] data, network data not only has attributes (metadata) associated with each entity (node), but rich structural information, mainly encoded in the links. To learn an appropriate distance measure for these networks both attribute and network structure should be exploited during the metric learning stage. In order to achieve this we employ the recently proposed SPML [35] (Structure Preserving

[6] a user is connected with another user if he/she is designated as a *friend* or *contact*, and/or subscribes to his/her channel

[7] a user is linked to a video if he/she upload or shares a video

[8] Also denoted as (V^q, E^q) for brevity

[9] $M \succeq 0$ and d is the dimension of the data

[10] independent and identically distributed

Metric Learning) algorithm. SPML learns a Mahalanobis distance metric for node attributes by using network structure as supervision. Doing so ensures that the learned distance function encodes the structure and can be used to predict link patterns from attributes. The goal of SPML is to learn the Mahalanobis metric M from a network G = (V, E), such that M preserves the adjacency structure.

In other words, given a connectivity algorithm $\hbar$ (*k-nearest neighbors* for instance), SPML learns a metric such that applying $\hbar$ to the input data (i.e. X) using the learned metric M produces the input adjacency matrix (i.e. A) exactly i.e. $\hbar(X,M)=A$. While SPML supports various choices of connectivity algorithm i.e. $\hbar$ including b-matching, k-nearest neighbors, ϵ-neighborhood etc for our purposes we adopt the simple k-nearest neighbors algorithm, meaning each node is connected with its top-k nearest neighbors under the defined metric. For sake of completeness we next provide a brief description of the SPML formulation.

The authors in [35] first formulate the problem of learning a structure preserving metric as a SDP (semi-definite program) which can be solved using any off the shelf SDP solver.

$$\begin{gathered} min_M \ \frac{\lambda}{2}\|M\|_F^2 + \epsilon \\ s.t\, \forall_{i,j}\ d_M(x_i,x_j) \geq (1-A_{ij})\ max_l(A_{il}D_M(x_i,x_l)) + 1 - \epsilon \end{gathered} \tag{1}$$

The constraints enforce that the distance to all disconnected nodes must be larger than the distance to the farthest connected neighbor. The Frobenius norm is a regularizer on M and λ is the corresponding weight parameter. A_{ij} indicates the (i,j) entry in the adjacency matrix A. Considering that the complexity of SDP scales with the number of variables and constraints, yielding a worst-case time of O(d^3 + C^3) where C = O(n^2) it is evident that the SDP formulation of SPML cannot scale for networks larger than a few hundred nodes or for high-dimensional features. By rewriting the constraints in terms of hinge-loss over triplets (i,j,k), each consisting of a node, its neighbor and its non-neighbor the authors are able to propose an efficient stochastic subgradient descent algorithm for solving the above optimization problem. For further details readers are encouraged to refer to [36].

Recently the SPML algorithm was extended by [11] for a multi-task setting[11]. The authors propose a multi-task version of SPML, abbreviated as MT-SPML, which is able to learn across multiple related tasks on multiple networks via shared intermediate parametrization. The input to MT-SPML is a set of networks G = {G^1,... ,G^Q}. Each network is represented as G^q = (V^q,E^q) where nodes of all networks share the same feature space. MT-SPML treats each network as a task and learns a specific metric for each task and a common metric for all the tasks. The task correlation is carried through the common metric and the individual metrics encode task specific information. The end goal of MT-SPML is to learn a task specific metric M_q for each network.

5. ROBUST MULTI-TASK, SPML

As discussed in the previous section MT-SPML combines the benefits of SPML (learning a distance measure which uses both network and node features) and multi-task learning in one setup. However, the MT-SPML approach has some limitations which were discussed in Section 2.4. We briefly summarize them here (1)*sensitiveness to non-informative features* (2) *inability to model non-stationary degree distribution across networks* (3) *assumes fixed task correlation structure*. We first describe how SPML, which is the building block for both MT-SPML and our work, is extended to address the robustness concern. Subsequent to that the problem definition is expanded to account for degree distributional characteristics. Finally, in-order to learn these metrics jointly we adopt the multi-task framework. For this we use a different parametrization than what was used in MT-SPML. Doing so enables our model to learn correlations between link-types.

5.1 Robust Metric Learning

We address the first limitation by using a mixed-norm regularization over the learned metric. The intuition for doing this (as described in [27]) is the following - ideally one would like the metric learning algorithm to learn a metric M which relies only on informative features. More precisely, if some input dimension j is non-informative, then the corresponding row and column of M should suppress this feature, i.e., $M_{j\cdot}$ = $M_{\cdot j}$ = 0. In contrast, sparsity should not be enforced for rows corresponding to informative features. This suggests a natural row (or column) grouping of the entries of M when enforcing sparsity, so that rows corresponding to informative features may be dense, but sparsity is enforced over rows to avoid relying upon to many features. Such row-sparsity can be promoted by using a mixed-norm regularization. Under this mixed-norm ($L_{2,1}$) regularization the SPML problem can be re-written as

$$\begin{gathered} min_M \ \frac{\lambda}{2}\|M\|_F^2 + \beta\|M\|_{2,1} \\ + \frac{1}{|S|}\sum_{i,j,k\in S} max(D_M(x_i,x_j) - D_M(x_i,x_k) + 1, 0) \end{gathered} \tag{2}$$

In this formulation constraints are written in terms of hinge-losses over triplets (i,j,k), each consisting of a node, its neighbor and its non-neighbor. The set of all such triplets is denoted by S = { (i,j,k) | A_{ij} = 1 , A_{ik} = 0}.

5.2 Degree Distributional Metric Learning

Authors in [35] propose an algorithm that learns a similarity metric and a set of degree-based score functions that together provide a structure-aware, distance-based method for link prediction. Applying this idea to our setup is fairly straight forward. Before doing this we first summarize the Degree Distributional Metric Learning (DDML) approach. The connectivity algorithm uses a degree preference function g, which takes a node's feature vector x and a target degree k, and is parameterized by matrix $B \in R^{d\times n}$ ($b_{k'}$ is the k' row of B). The score is then calculated as

$$g(k \mid x; B) = \sum_{k'=1}^{k} x^T b_{k'}$$

Next, score of a graph G is defined as the sum of all edge distances and the degree preference functions for each node. This is represented in the following form (note A denotes the adjacency matrix for graph G)

$$F(A \mid X; M, B) = \sum_{A_{ij}} D_M(x_i,x_j) - \sum_i g\Big(\sum_j A_{ij} \mid x_i; B\Big)$$

Equation 2 is now rewritten in the following form

$$\begin{gathered} min_{M,B} \ \frac{\lambda}{2}\|M\|_F^2 + \beta\|M\|_{2,1} \\ + \frac{1}{|S|}\sum_{i,j,k} max(F(A \mid X; M, B) - F(A^{(i,j,k)} \mid X; M, B) + 1, 0) \end{gathered} \tag{3}$$

$A^{(i,j,k)}$ denotes the false graph produced by toggling the edge between nodes i and j and the edge between nodes i and k. The version presented above is the stochastic version of DDML in which the degree preference function is parameterized only up to a fixed maximum degree, thus eliminating the dependence of the running time on the size of the graph. For further details readers are encouraged to refer to [35].

[11]SPML in its original form is described for a single network

5.3 Learning Task Correlation

For the multi-task setup we assume there are Q tasks (i.e. Q different *link-type distance measures* have to be learnt). The objective of the multi-task, metric learning setup is to learn a metric for each task in a coupled fashion in order to improve the performance on individual tasks. Multi-task methods achieve this by either assuming that the latent data representation is shared by different tasks or the learning models in different tasks have similar model parameter. For our setup we use a different parametrization than what was used in MT-SPML. We use a task covariance matrix $\mathbf{\Sigma}$ to learn the relationships between tasks. We now write Equation 3 in the multi-task form

$$\begin{gathered} min_{M_1,M_2,..,M_Q,B_1,B_2,..,B_Q,\Sigma} \sum_{q=1}^{Q} \frac{\lambda_q}{2}\|M_q\|_F^2 + \sum_{q=1}^{Q}\beta_q\|M_q\|_{2,1} \\ +\frac{\lambda}{2}tr(\mathrm{T}\Sigma^{-1}\mathrm{T}^T) + \sum_{q=1}^{Q} \\ \frac{1}{|S_q|}\sum_{i,j,k\in S_q} max(F(A_q \mid X_q; M_q, B_q) - F(A_q^{(i,j,k)} \mid X_q; M_q, B_q) + 1, 0) \\ s.t.\ M_i \succeq 0\ \forall i = 1...Q, \Sigma \succeq 0, tr(\Sigma) = 1 \\ T = (vec(M_1), ..., vec(M_Q)) \end{gathered} \tag{4}$$

This idea of learning a task covariance matrix Σ in the multi-task metric learning setup is inspired by [43][12]. The vec() symbol denotes the operator which converts a matrix into a vector in a column wise manner. Using this formulation in our setup gives us the following key advantage - Equation 4 decomposes into Q convex problems each having the following structure

$$\begin{gathered} min_{M_q,B_q,\Sigma} \frac{\lambda_q}{2}\|M_q\|_F^2 + \beta_q\|M_q\|_{2,1} + \frac{\lambda}{2}tr(\mathrm{T}\Sigma^{-1}\mathrm{T}^T) \\ +\frac{1}{|S_q|}\sum_{i,j,k\in S_q} max(F(A_q \mid X_q; M_q, B_q) - F(A_q^{(i,j,k)} \mid X_q; M_q, B_q) + 1, 0) \\ s.t.\ M_q \succeq 0, \Sigma \succeq 0, tr(\Sigma) = 1 \\ T = (vec(M_1), ..., vec(M_Q)) \end{gathered} \tag{5}$$

We next describe an efficient method to optimize Equation 5.

6. OPTIMIZATION

To efficiently optimize Equation 5 we use the Alternating Direction Multiplier Method or ADMM. The ADMM method is well suited for distributed convex optimization problems. It takes the form of a *decomposition-coordination* procedure, in which solutions to small local subproblems are coordinated to find a solution to a large global problem. We first optimize w.r.t to M_q assuming all other distance metrics to be fixed. We use the notation M_{-q} = {vec(M_1),...vec(M_{q-1}),vec(M_{q+1}),..vec(M_Q)}. We re-write Equation 5 in the following equivalent format.

$$\begin{gathered} min_{(M_q,V,Z)\in S^d,\Sigma,B_q}\ g(M_q,B_q) + h(V) + f(Z) + \ell(\Sigma,\tilde{T_q}) \\ s.t.\ M_q = V = Z \\ where \\ \tilde{T_q} = (vec(M_q), M_{-q}) \\ g(M_q,B_q) = \frac{\lambda_q}{2}\|M_q\|_F^2 \\ + \sum_{i,j,k\in S_q} max(F(A_q \mid X_q; M_q, B_q) - F(A_q^{(i,j,k)} \mid X_q; M_q, B_q) + 1, 0) \\ h(V) = \beta\|V\|_{2,1}, f(Z) = \begin{cases} 0, Z \in S_+^d \\ \infty, Z \notin S_+^d \end{cases}, \ell(\Sigma,\tilde{T_q}) = \frac{\lambda}{2}tr(\tilde{T_q}\Sigma^{-1}\tilde{T_q}^T) \\ \Sigma \succeq 0, tr(\Sigma) = 1 \end{gathered} \tag{6}$$

The next step is to write the augmented Lagrangian. For this we introduce the Lagrange multipliers $\Gamma_{M_q}, \Gamma_V \in S^d$ [13]. Using these notations the augmented Lagrangian can be written as

$$\begin{gathered} L_\rho(M_q,B_q,V,Z,\Gamma_V,\Gamma_{M_q}) = g(M_q,B_q) + h(V) + f(Z) + \ell(\Sigma,\tilde{T_q}) \\ + < \Gamma_V, V - Z >_F + < \Gamma_{M_q}, M_q - Z >_F + \frac{\rho}{2}(\| M_q - Z \|)_F^2 \\ +\frac{\rho}{2}(\| V - Z \|)_F^2 \end{gathered} \tag{7}$$

where $\rho > 0$ is a scaling parameter. ADMM consists of the following iterations

$$\begin{gathered} M_q^{t+1} = argmin_{M_q\in S^d} g(M_q, B_q^t) + \ell(\Sigma^t, \tilde{T_q^t}) + \frac{\rho}{2}(\| M_q - (Z^t - \Gamma_{M_q}^t) \|)_F^2 \\ \tilde{T_q^{t+1}} = (vec(M_q^{t+1}), M_{-q}) \\ V^{t+1} = argmin_{V\in S^d} h(V) + \frac{\rho}{2}(\| V - (Z^t - \Gamma_V^t) \|)_F^2 \\ Z^{t+1} = argmin_{Z\in S^d} f(Z) + \rho(\| Z - \frac{1}{2}(M_q^{t+1} + V^{t+1} + \Gamma_V^t + \Gamma_{M_q}^t) \|)_F^2 \\ \Sigma^{t+1} = argmin_{\Sigma\in S_+^d, tr(\Sigma)=1} \ell(\Sigma, \tilde{T_q^{t+1}}) \\ B_q^{t+1} = argmin_{B_q} g(M_q^{t+1}, B_q) \\ \Gamma_V^{t+1} = \Gamma_V^t + V^{t+1} - Z^{t+1} \\ \Gamma_{M_q}^{t+1} = \Gamma_{M_q}^t + M^{t+1} - Z^{t+1} \end{gathered} \tag{8}$$

The V and Z update steps are exactly the same as presented in [27]. The V update involves a separate ADMM step which involves a prox-operator ($prox_{l_{2,1}}$) calculation followed by a projection operation $\prod_{S^d}$. The Z update involves a positive semi-definite projection step $\prod_{S_+^d}$. The Σ update step can be solved analytically using the procedure outlined in [44]. The B update step can be borrowed from the stochastic degree distributional metric learning algorithm outlined in [35]. Finally, to write down the update step for M_q we observe the following. Given that the task covariance matrix can be written as Σ^{-1}= $\begin{pmatrix} \gamma_{ii} & \gamma_\mathbf{i}^\mathbf{T} \\ \gamma_\mathbf{i} & \Gamma_{-i} \end{pmatrix}$ one can write $\ell(\Sigma,\tilde{T_q})$ as

$$\ell(\Sigma,\tilde{T_q}) = \frac{\lambda}{2}tr(\tilde{T_q}\Sigma^{-1}\tilde{T_q}^T) = \frac{\lambda}{2}[\gamma_{ii} \parallel M_q \parallel_F^2 + 2tr(AM_q)] \tag{9}$$

where A is a matrix such that $vec(A)$ = $M_{-q}\gamma_\mathrm{i}$[14]. Using Equation 9 we can now write

$$\begin{gathered} M_q^{t+1} = argmin_{M_q\in S^d} g(M_q, B_q^t) + \frac{\lambda}{2}[\gamma_{ii} \parallel M_q \parallel_F^2 + 2tr(AM_q)] \\ +\frac{\rho}{2}(\| M_q - (Z^t - \Gamma_{M_q}^t) \|)_F^2 \end{gathered} \tag{10}$$

Due to symmetric constraint on M_q the update step for M_q is a separate ADMM algorithm which alternates between a gradient update step, a projection $\prod_{S^d}$ and an additive dual update.

6.1 Link Prediction

To predict whether two entities will form a specific link-type we utilize the corresponding learnt distance metric for the link prediction task. For instance, to predict which Flickr Group a given user is likely to join the learnt metric $d_{user-flickr\_group}(x_i, x_j)$ is used. Flickr Group with the smallest $d_{user-flickr\_group}$ value is the most likely candidate for the *user-flickr_group* link formation.

7. EXPERIMENTS

In this section we provide details of the experimental setup, dataset used and the different baselines against which our proposed system is compared. We begin by providing details of the different

[12]As Σ is a task covariance matrix which describes the relationships between tasks it is a PSD matrix. The tr(Σ)=1 constraint is to restrict the scale of Σ and prevent degenerate solution

[13]S^d denotes $d \times d$ symmetric matrix, S_+^d denotes $d \times d$ symmetric positive semi-definite matrix

[14]For the M_q update step A is a known quantity.

Link-Type	ST-SVM	Pooled-SVM	MT-SVM	SPML	Pooled-SPML	MT-SPML	MRIP	Combined Approach
user-user	0.36	0.31	0.39	0.43	0.44	**0.61**	0.54	**0.77**
user-flickr_group	0.29	0.31	0.32	0.48	0.55	0.59	**0.61**	**0.68**
flickr_group-image	0.33	0.36	0.38	0.52	0.59	0.61	**0.64**	**0.72**

Link-Type	ST-SVM	Pooled-SVM	MT-SVM	SPML	Pooled-SPML	MT-SPML	MRIP	Combined Approach
author-author	0.47	0.40	0.49	0.59	0.61	0.63	**0.71**	**0.79**
document-conference	0.49	0.51	0.52	0.59	0.57	0.60	**0.64**	**0.76**
document-document	0.50	0.54	0.58	0.62	0.64	0.65	**0.71**	**0.78**

Table 1: AUC for different link-types – Flickr (top-table), DBLP (bottom-table) dataset

baselines in Section 7.1. Evaluation metric and dataset details are presented in Section 7.2 and Section 7.3 respectively. Results are discussed in Section 7.4.

7.1 Baseline

In order to evaluate the effectiveness of our system we compare our proposed method against the following methods

- **SPML** We apply the SPML method and its variant for the link prediction task.
  - **ST-SPML**: This is the single-task SPML as presented in [35]. A metric is learned for each network independently. This method models network structure but not link-type correlations.
  - **Pooled-ST-SPML**: Training examples from all tasks are pooled together and ST-SPML is applied to the problem. This is a naive way of sharing knowledge between tasks, but it does not respect the differences between tasks.
  - **MT-SPML**: MT-SPML [11] is the multi-task (or multi-network) variant of ST-SPML.
- **SVM-based approaches**: Since SVM-based methods do not model network structure, we need to construct features to encode this piece of information. We adopt the approach proposed in [11] where training examples are constructed by taking the pairwise difference of the attributes between two nodes. The training labels are binary, with 1 representing the existence of a link between a pair of nodes and 0 the absence. The classification score is used to measure the distance (classification score is inversely proportional to the notion of distance).
  - **ST-SVM** : This is the normal single-task SVM. We train a LIBLINEAR SVM for each network independently.
  - **Pooled-SVM**: One SVM is trained for all networks by pooling all data together.
  - **MT-SVM**: This is the multi-task SVM as described in [10]. The MT-SVM jointly learns a common decision boundary for all tasks and a specific boundary for each task. At test time, the common and task specific decision boundary together form the final classification model for each task.
- **Heterogenous Link Prediction Methods**: We benchmark our system against the Multi-Relational Influence Propagation (**MRIP**) algorithm [42]. MRIP is a probabilistic method that models the influence propagating between heterogeneous relationships.
- **Proposed Approach**: We benchmark the performance of the proposed method against the above baselines (Table 1). We also provide results on how different variants of the proposed method perform on the test datasets (Table 2). These variants are
  - **Robust-MT-SPML**: MT-SPML formulation with mixed-norm regularization as described in Section 5.1.
  - **Robust-Degree$_{\text{perf}}$-MT-SPML**: Robust MT-SPML model with degree preference functions as detailed in Section 5.2.
  - **Robust-Covariance-MT-SPML**: Robust MT-SPML with multi-task metric learning with covariance parametrization as described in Section 5.3.
  - **Combined Approach**: Combination of all the above approaches i.e. robust, degree-distributional multi-task metric learning with covariance parametrization.

7.2 Evaluation Metric

We employ AUC or Area Under the ROC Curve metric for comparing the performance of the different methods. The choice of this metric is based on the fact that most link prediction datasets are characterized by extreme imbalance. This issue has motivated the use of AUC as the de facto performance measure for link prediction tasks, as, unlike standard 0-1 accuracy, AUC is not influenced by the distribution of the classes. AUC can be interpreted as the probability that a randomly chosen missing link has higher score than a randomly chosen non-existent link.

7.3 Dataset and Experiment Setup

In this section we provide details of the two datasets that were used to benchmark the performance of our system.

7.3.1 Flickr

The Flickr social network considered in our experiments consists of the following 3 entities : *user*, *flickr_group* and *image*. For our experiments we are interested in predicting the following relations (link-types) : *user-user* i.e. a user will interact with another user (add him as a *friend* or *family*, *comment* or *like* an image), *user-flickr_group* i.e. a user joins a given Flickr Group and *flickr_group-image* i.e. a contributed image will be shared in a Flickr Group. The Flickr data has been obtained by querying Flickr public API in the time window 2013-2014 and then by performing forest fire sampling on the resulting network. We crawled 100 popular Flickr groups based on the *tag-based* group search in Flickr, by selecting 10 popular tags and selecting top 10 groups for each tag. We then collect the information of 10,858 users[15] in these 100 groups, including the images and the associated tags they uploaded as well as

[15] We filter out users that have less than 20 images in a Flickr Group

Link-Type	Robust-MT-SPML	Robust-**Degree$_{\mathbf{perf}}$**-MT-SPML	Robust-Covariance-MT-SPML	Combined Approach
user-user	0.67	0.73	0.69	**0.77**
user-flickr_group	0.61	0.64	0.62	**0.68**
flickr_group-image	0.67	0.69	0.66	**0.72**

Link-Type	Robust-MT-SPML	Robust-**Degree$_{\mathbf{perf}}$**-MT-SPML	Robust-Covariance-MT-SPML	Combined Approach
author-author	0.64	0.72	0.70	**0.79**
document-conference	0.62	0.73	0.68	**0.76**
document-document	0.66	0.74	0.69	**0.78**

Table 2: AUC for different link-types – Flickr (top-table), DBLP (bottom-table) dataset

their profile information. The number of images in our data set is 556,794[16] (approximately $\frac{1}{5}^{th}$ of these images belong in the group pools). For our experiments node features (attributes) for *User*, *Image*, *Flickr Group* are generated from the tags/keywords associated with the *User* (tags associated with the images uploaded by the user), *Image* (image tags/captions) and *Flickr Group* (tags associated with the images uploaded in the Flickr Group). Basic post-processing is done on the collected tags to remove stop-words, tags containing alpha-numeric digits etc followed by *tf-idf*[17] filtering. For the Flick experiments d=6843. The Flickr data-set contains timestamp information on when a link was created, this allows us to perform a chronological split, where older links (80% of the data) are used for learning the model, while the most recent 20% are used as prediction target. The accuracy of link prediction is measured by computing the area under the ROC curve (AUC) over a set of positive and negative examples drawn from the test set. All links in the test-set are treated as positive examples whereas all non-existing links are considered negative example. As noticed in previous work the sparsity of theses networks poses two major concerns. First, the number of non-existing links can be enormous, thus making the computation of the AUC infeasible. Second, missing links do not necessarily represent negative information. Considering these challenges we limit the negative examples to all the 2-hops non-existing links. Link prediction performance is evaluated separately for each link type using AUROC. Table 2 provides the AUC numbers for the different methods and link-types. As the table shows the proposed method out-performs all the baseline methods for all 3 link-type prediction tasks. Our method gives an average[18] AUC improvement of 10.3% over the most competitive baseline.

7.3.2 DBLP

We use the DBLP dataset from [8]. The three entities in this heterogenous network are *documents*, *authors* and *conferences*. The dataset contains $28,569$ documents, $28,702$ authors and 20 conferences. For our experiments we are interested in predicting the following relations (link types) : *author-author* i.e. collaboration, *document-document* i.e. citation, *document-conference*. This network contains a total of $103,201$ links. We choose data between 1990 and 2000 as our training set, and data between 2001 and 2005 as the test set. Even though the network contains other link-types (for e.g author-conference, author-document) we ignore them in our experiments for ease of comparison with prior work [42]. For our experiments node features (terms) for *author*, *document* and *conference* are generated from the documents contributed by the author, *document's* abstract and the documents contributed to the conference respectively. Basic post-processing such as stop-word removal, *tf-idf*[17] filtering is performed on the terms. For the DBLP experiments d=7389. Table 2 provides the AUC numbers for the different methods and link-types. Our method gives an average[18] AUC improvement of 9% over the most competitive baseline.

7.4 Results

As evident from Table 1, our proposed method outperforms all baselines. On average[19] our method gives an AUC improvement of 9.6% over the best performing baseline method (MRIP in most cases). We next discuss some observations.

- The SPML method and it's variants (MT-SPML,Pooled-SPML) outperform our SVM[20] baseline. One key reason for this is that the SVM baseline uses a very naive but commonly used approach to encode network/topological features. This approach is similar to previous work where nonlinear classifiers such as a kernel SVM [15] or decision tree [26] have been used to combine topological and explicit features for the link prediction task. These topological features have limited expressivity when compared to SPML's approach of explicitly modeling the connectivity structure of the network.

- Pooling data (Pooled-SVM or Pooled-SPML) is a naive approach for sharing knowledge across link-types. The performance of the pooled methods is inferior to methods where we explicitly model the relationship between link-types (e.g. MT-SPML, MT-SVM).

- Explicitly modeling for robustness against non-informative features during the metric learning process improves results - Robust-MT-SPML outperforms MT-SPML on both datasets. The gains (in AUC) are more pronounced in the case of the Flickr dataset (4.6% gain over MT-SPML) where the feature vocabulary, which is built from noisy tags, contain more non-informative features as compared to that built from the DBLP dataset (1.3% gain over MT-SPML).

- Incorporating degree-distributional metric learning into the overall metric learning process improves results. As motivated early on in the paper the degree distribution of the individual interaction network varies across networks. For example in Flickr, the *user-user* interaction network has a very different degree distribution as compared to the *user-flickr_group* interaction network - a *user* will connect to only a few other users in the network, but a *flickr_group* could have large number of members/users. Similarly, Flickr Groups with general themes (e.g. art, nature) will have larger group membership as compared to Flickr Groups with very specialized focus [33]. Explicitly modeling for this yields an average AUC improvement of 6.45% over *Robust-MT-SPML*. The average is computed across all link-types and datasets.

[16]images that contain less than 4 tags are discarded

[17]each entity instance is treated as a document

[18]averaged across all link-types

[19]Average computed across all datasets and link-types

[20]We also tried Logistic Regression on the same set of features and found comparable performance

- *Robust-Covariance-MT-SPML* vs. *Robust-MT-SPML*. We get mixed results for this setup. On the Flickr dataset modeling for link-type covariance - i.e. *Robust-Covariance-MT-SPML*- gives us no improvement over *Robust-MT-SPML*. However on the DBLP dataset we see an average[21] 5% AUC improvement over *Robust-MT-SPML*. One reason for this could be that the link-types in the DBLP network (*author-author*, *document-conference*, *document-document*) demonstrate a stronger correlation pattern as compared to link-types in the Flickr network. It would be interesting to observe the performance of this method for networks with a large number of link-types (e.g. Protein-Protein Interaction networks). We leave this as a future exercise.

8. CONCLUSION

In this paper we pose the problem of link prediction in heterogeneous networks as a *multi-task, metric learning* (MTML) problem. For each link-type (interaction type) we learn a corresponding distance measure, which utilizes both network and node features. These distance measures are learnt in a coupled fashion by employing the Multi-Task Structure Preserving Metric Learning (MT-SPML) setup. We further extend the MT-SPML method to account for correlation between link-types, robustness to non-informative features and non-stationary degree distribution across interaction networks. Experiments on the Flickr and DBLP network demonstrates the effectiveness of our proposed approach vis-à-vis competitive baselines. As part of future work we plan to (a) evaluate the model's performance on larger datasets (b) evaluate how the model performs on networks with large number of link-types (c) provide a map-reduce implementation of the current ADMM approach.

9. REFERENCES

[1] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[3] A.L. BarabÂt'asi1. Evolution of the social network of scientific collaborations. 2008.

[4] Xianye Ben, Weixiao Meng, Rui Yan, and Kejun Wang. An improved biometrics technique based on metric learning approach. *Neurocomputing*, 97:44–51, 2012.

[5] Bin Cao, Nathan Nan Liu, and Qiang Yang 0001. Transfer learning for collective link prediction in multiple heterogenous domains. In Johannes FÃijrnkranz and Thorsten Joachims, editors, *ICML*, pages 159–166. Omnipress, 2010.

[6] Hung-Hsuan Chen, Liang Gou, Xiaolong (Luke) Zhang, and C. Lee Giles. Discovering missing links in networks using vertex similarity measures. In Sascha Ossowski and Paola Lecca, editors, *SAC*, pages 138–143. ACM, 2012.

[7] Darcy A. Davis, Ryan Lichtenwalter, and Nitesh V. Chawla. Multi-relational link prediction in heterogeneous information networks. In *ASONAM*, pages 281–288. IEEE Computer Society, 2011.

[8] Hongbo Deng, Jiawei Han, Bo Zhao, Yintao Yu, and Cindy Xide Lin. Probabilistic topic models with biased propagation on heterogeneous information networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1271–1279, New York, NY, USA, 2011. ACM.

[9] Yuxiao Dong, Qing Ke, Jun Rao, and Bin Wu. Predicting missing links via local feature of common neighbors. In *Fuzzy Systems and Knowledge Discovery*, 2011.

[10] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 109–117, New York, NY, USA, 2004. ACM.

[11] Chen Fang and Daniel N. Rockmore. Multi-task metric learning on network data. *CoRR*, abs/1411.2337, 2014.

[12] FranÃğois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.*, 19(3):355–369, 2007.

[13] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, pages 1–8. IEEE Computer Society, 2007.

[14] Matthieu Guillaumin, Jakob J. Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, pages 498–505. IEEE Computer Society, 2009.

[15] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[16] G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. 2002.

[17] Nan Jiang, Wenyu Liu, and Ying Wu. Order determination and sparsity-regularized metric learning adaptive visual tracking. In *CVPR*, pages 1956–1963. IEEE Computer Society, 2012.

[18] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, VOL. 18, NO. 1:39– 43, 1953.

[19] Dor Kedem, Stephen Tyree, Kilian Q. Weinberger, Fei Sha, and Gert R. G. Lanckriet. Non-linear metric learning. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, LÃl'on Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pages 2582–2590, 2012.

[20] Tsung-Ting Kuo, Rui Yan, Yu-Yang Huang, Perng-Hwa Kung, and Shou-De Lin. Unsupervised link prediction using aggregative statistics on heterogeneous social networks. In Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy, editors, *KDD*, pages 775–783. ACM, 2013.

[21] RÃl'mi Lajugie, Francis R. Bach, and Sylvain Arlot. Large-margin metric learning for constrained partitioning problems. In *ICML*, volume 32 of *JMLR Proceedings*, pages 297–305. JMLR.org, 2014.

[22] Marc Teva Law, Carlos Sureda Gutierrez, Nicolas Thome, and StÃl'phane GanÃğarski. Structural and visual similarity learning for web page archiving. In Patrick Lambert, editor, *CBMI*, pages 1–6. IEEE, 2012.

[23] Guy Lebanon. Metric learning for text documents. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):497–508, 2006.

[24] Vincent Leroy, B. Barla Cambazoglu, and Francesco Bonchi. Cold start link prediction. In *Proceedings of the 16th ACM

[21] averaged over link-types

SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, pages 393–402, New York, NY, USA, 2010. ACM.

[25] David Liben-Nowell and Jon M. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2003.

[26] Ryan Lichtenwalter, Jake T. Lussier, and Nitesh V. Chawla. New perspectives and methods in link prediction. In *KDD*, pages 243–252, 2010.

[27] Daryl Lim, Gert R. G. Lanckriet, and Brian McFee. Robust structural metric learning. In *ICML (1)*, volume 28 of *JMLR Proceedings*, pages 615–623. JMLR.org, 2013.

[28] Zhengdong Lu, Berkant Savas, Wei Tang, and Inderjit S. Dhillon. Supervised link prediction using multiple sources. In Geoffrey I. Webb, Bing Liu 0001, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu, editors, *ICDM*, pages 923–928. IEEE Computer Society, 2010.

[29] Brian McFee, Luke Barrington, and Gert R. G. Lanckriet. Learning content similarity for music recommendation. *CoRR*, abs/1105.2344, 2011.

[30] Brian Mcfee and Gert Lanckriet. Metric learning to rank. In *In Proceedings of the 27th annual International Conference on Machine Learning (ICML*, 2010.

[31] Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Proceedings of the ECML/PKDD 2011*, 2011.

[32] Thomas Mensink, Jakob J. Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV (2)*, volume 7573 of *Lecture Notes in Computer Science*, pages 488–501. Springer, 2012.

[33] Radu A. Negoescu and Daniel G. Perez. Analyzing flickr groups. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, CIVR '08, pages 417–426, New York, NY, USA, 2008. ACM.

[34] Rudy Raymond and Hisashi Kashima. Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. In JosÃľ L. BalcÃązar, Francesco Bonchi, Aristides Gionis, and MichÃĺle Sebag, editors, *ECML/PKDD (3)*, volume 6323 of *Lecture Notes in Computer Science*, pages 131–147. Springer, 2010.

[35] Blake Shaw, Bert C. Huang, and Tony Jebara. Learning a distance metric from a network. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 1899–1907, 2011.

[36] Blake Shaw, Bert C. Huang, and Tony Jebara. Learning a distance metric from a network. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 1899–1907, 2011.

[37] Yizhou Sun, Rick Barber, Manish Gupta, Charu C. Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM*, pages 121–128. IEEE Computer Society, 2011.

[38] Matthew E. Taylor, Brian Kulis, and Fei Sha. Metric learning for reinforcement learning agents. In Liz Sonenberg, Peter Stone, Kagan Tumer, and Pinar Yolum, editors, *AAMAS*, pages 777–784. IFAAMAS, 2011.

[39] Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. Learning hierarchical similarity metrics. In *CVPR*, pages 2280–2287. IEEE Computer Society, 2012.

[40] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *ICDM*, pages 322–331. IEEE Computer Society, 2007.

[41] Sen Wu, Jimeng Sun, and Jie Tang. Patent partner recommendation in enterprise social networks. In Stefano Leonardi, Alessandro Panconesi, Paolo Ferragina, and Aristides Gionis, editors, *WSDM*, pages 43–52. ACM, 2013.

[42] Yang Yang, Nitesh V. Chawla, Yizhou Sun, and Jiawei Han. Predicting links in multi-relational and heterogeneous networks. In *ICDM*, 2012.

[43] Yu Zhang and Dit-Yan Yeung. Transfer metric learning by learning task relationships. In Bharat Rao, Balaji Krishnapuram, Andrew Tomkins, and Qiang Yang 0001, editors, *KDD*, pages 1199–1208. ACM, 2010.

[44] Yu Zhang and Dit-Yan Yeung. Transfer metric learning by learning task relationships. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1199–1208, New York, NY, USA, 2010. ACM.

[45] Yu-Xiao Zhu, Linyuan Lu, Qian-Ming Zhang, and Tao Zhou. Uncovering missing links with cold ends. *CoRR*, abs/1104.0395, 2011.