Factorized Similarity Learning in Networks

Shiyu Chang[1], Guo-Jun Qi[2], Charu C. Aggarwal[3], Jiayu Zhou[4], Meng Wang[5], Thomas S. Huang[1]
[1] Beckman Institute, University of Illinois at Urbana-Champaign, IL 61801, USA. {chang87, t-huang1}@illinois.edu.
[2] University of Central Florida, Orlando, FL, 32816, USA. guojun.qi@ucf.edu.
[3] IBM T.J. Watson Research Center, NY, 10598, USA. charu@us.ibm.com.
[4] Arizona State University, Tempe, AZ, 85281, USA. jiayu.zhou@asu.edu.
[5] Hefei University of Technology, Hefei, Anhui, 230009, China. wangmeng@hfut.edu.cn.

***Abstract*—The problem of similarity learning is relevant to many data mining applications, such as recommender systems, classification, and retrieval. This problem is particularly challenging in the context of networks, which contain different aspects such as the topological structure, content, and user supervision. These different aspects need to be combined effectively, in order to create a holistic similarity function. In particular, while most similarity learning methods in networks such as *SimRank* utilize the topological structure, the user supervision and content are rarely considered. In this paper, a *Factorized Similarity Learning (FSL)* is proposed to integrate the link, node content, and user supervision into an uniform framework. This is learned by using matrix factorization, and the final similarities are approximated by the span of low rank matrices. The proposed framework is further extended to a noise-tolerant version by adopting a hinge-loss alternatively. To facilitate efficient computation on large scale data, a parallel extension is developed. Experiments are conducted on the *DBLP* and *CoRA* datasets. The results show that *FSL* is robust, efficient, and outperforms the state-of-the-art.**

I. Introduction

Networks are ubiquitous in the context of data mining and information retrieval applications. Social and technical information systems usually exhibit a wide range of interesting properties and patterns such as interacting physical, conceptual and societal entities. Each individual entity interchanges and influences each other in the context of this interconnected network. Information networks are usually very large and information-rich. A significant amount of research has been done to study various aspects of network analysis, such as search, community detection and collective classification.

A central tenet of network mining research is the notion of similarity between pairs of nodes in a network. In many cases, similarity functions are used as subroutines in different data mining applications. For instance, information retrieval queries use the learned similarities [21][23][30][32][33], and recommender systems model user and item profiles from collaborative similarities [15][26]. However, similarity learning in the network environment differs from traditional approaches, mainly due to the heterogeneous information and sources, including link information, content, and user behaviors. In addition, the noisy nature of the underlying network poses a great challenge to effective learning. For instance, links are not semantically meaningful, especially in online social networks such as *Facebook*. In this context, it is essential to make the network similarity learning algorithms capable of dealing with noisy multi-modality scenarios. We illustrate the problem of similarity learning on networks in Figure 1. The graph demonstrates a generalized network structure, where

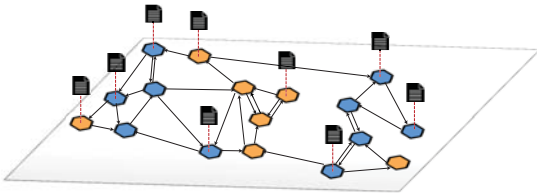

Fig. 1: An example of network structure.

each hexagon indicates a node in the network, and the arrowed dash lines are directed links between different nodes. The color of each node reflects its property. Nodes with the same color indicate that they are similar, or belong to the same group. The nodes also have content associated with them. In the context of networks with noisy links, it is generally hard to learn similarities, with the use of only the linkage structure. In particularly, the impact of cumulative propagation of errors can be very significant in such networks. For example, consider the scientific bibliography networks, in which nodes represent authors, and edges represent collaborations. In many cases, edges represent occasional collaborations between different research domains, in spite of significant differences between the corresponding nodes. On the other hand, the content provides complementary information about authors, but ignores structural relationships among nodes in the network.

In this paper, we propose a *Factorized Similarity Learning (FSL)* approach to transfer and fuse knowledge from different domains. It fuses the information from network structure (links), content, and user supervision, to achieve stable and generalized similarity learning on networks. This is achieved by integrating these heterogeneous facets into an uniform matrix factorization framework. The addition of content information to the network structure resolves the limitation of both local and global similarity measurements. This issue has been widely discussed in information retrieval research [21][35]. The major advantage of matrix factorization is that it provides a seamless way to capture the low rank structure of different aspects of the data, such as content, structure and user supervision. The user supervision is specified in terms of order constraints. The content and order-constraints are leveraged to regularize and reconstruct the network topology by identifying noisy links while enhancing important ones. This provides semantically meaningful similarity functions and effectively prevents the error propagation through the topological links. We further extend FSL to distributed settings, in order to improve the computational efficiency. To verify the proposed FSL algorithm, we conduct several experiments on different data sets, including *DBLP* scientific bibliography [9] and *CoRA* [24] citation data set. The experimental results

evaluated on large-scale data sets verify the effectiveness of our approach.

The remainder of this paper is organized as follows. Section II reviews related work on both link and content based similarity learning, and well-known matrix completion methods. We present the problem formulation and mathematical model for FSL in section III and IV. We then show how the model can handle the case with noisy supervision in section V. We present extensive experiments on a wide range of data sets in section VI. The conclusion and future research directions are presented in section VII.

II. Related Work

In this section, we briefly review existing approaches for learning similarity functions as well as some off-the-shelf matrix completion methods. In general, similarity learning can be done by either using content or network topology.

A. Content-based Similarity Learning

In recent years, there are some emerging research interests in learning content-based similarity in a low-dimensional space such that the regular Euclidean metric is more meaningful in term of reflecting semantic "closeness" [1]. The first category is supervised metric learning, that is learning a distance metric from the training data with explicit class labels. The representative techniques includes the Neighborhood Component Analysis (NCA) [11] and the Large Margin Nearest Neighbor classification (LMNN) [38]. However, the performance of the supervised approaches rely heavily on the number of labeled training data examples. This is a problem, because such labels are usually not available in significant large numbers. Xing et al. [41] proposed to use side information, instead of class labels. The side information is presented as pairwise constraints associated with input data, which provides weaker information than the exact class labels. In particular, each constraint indicates whether a pair of samples is similar or irrelevant to each other. Subsequently, there were several promising research directions, such as Relevance Component Analysis (RCA) [2] and Information Theoretic Metric Learning (ITML) [7].

However, most of the existing metric learning algorithms do not scale well across various high dimensional learning paradigms. The reason is the size of the distance matrix scales with the square of the dimensionality. Sparse Distance Metric Learning (SDML) [31] works under pairwise relevance constraints to produces sparse metrics which significantly reduce the number of parameters, so that the time required for learning reduces dramatically. Another issue, that makes metric-based similarity learning inefficient for real-world applications, is the positive semi-definite (PSD) constraints imposed on the distance matrix. In general, it requires nontrivial PSD programming [4] techniques, and the computational complexity is cubic in the dimensionality of the input data. A recent work proposed by Zhen et al., which is referred to as Locally-Adaptive Decision Learning (LAD) [19] learns a non-isotropic similarity function by a joint model of of a distance metric and a locally adaptive thresholding rule. The LAD algorithm relaxes the PSD constraint so that the learned similarity can be negative, if only the relative order is appreciated.

B. Link-based Similarity Learning

In contrast to content-based similarity learning, link-based methods emphasize network topological structure. The most popular link-based similarity learning method or ranking system is known as the *PageRank* [28] , which is used by the Google search engine. The original Brin and Page model for *PageRank* uses the hyperlink structure of the web to build a Markov process with a primitive transition probability. A lot of link-based similarity learning approaches are motivated by *PageRank* including *SimFusion* [40], *Pagesim* [20] and the Relational like-base ranking [10].

An interesting method, known as *SimRank* [14] which is a iterative *PageRank*-like structure similarity measure in networks. However, *SimRank* only utilizes the in-link relationships for proximity computation while neglecting the information conveyed from out-links. Zhao et al. proposed a *P-Rank* [43] algorithm which extends *SimRank* by considering both in-link and out-link simultaneously. It is worth mentioning that the most of existing link-based methods rely heavily on homophily assumptions [25], which are insufficient for fully capturing the underlying semantics.

C. Matrix Factorization

Matrix factorization is one of the most popular methods in matrix completion and recommendation. Typically, the factorization assumes, that there is low rank distributions in space, and a low rank approximation is utilized to regularize the factorization process. The fundamental problem is to fill out the missing entries of the utility matrix with sparse observations. Traditional approaches include low-rank matrix fitting (LMaFit) [39], nonnegative matrix factorization (NMF) [18] and probabilistic matrix factorization (PMF) [26], which fit a probabilistic distribution for the matrix.

In the domain of collaborative filtering, which learns the similarities between different entries, the social hints are also considered in addition to link structures [22][29]. These approaches are referred to as *social matrix factorization*. Other approaches try to incorporate content similarities into the factorization, and a typical extension is Collaborative Topic Modes [37]. However, all the approaches are unsupervised, and also do not work well in noisy content-centric scenarios.

III. Problem Formulation

The two fundamental components, which define a network topology, are nodes and edges. We model any given network as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents a set of nodes/vertices and $\mathcal{E}$ represents the edges between these nodes. We denote the vertices by $\mathcal{V} = \{v_1, \ldots, v_n\}$ and edges by $\mathcal{E} = \{e_1, \ldots, e_m\}$. Thus, there are a total of n nodes and m directed edges. The directed assumption is without loss of generality, because undirected networks can be easily converted into a directed framework, by simply replacing undirected links by two directed edges. We further assume, that two additional types of information are available. One of them corresponds to link weights and the other one corresponds to content features. The weight of a link indicates the strength of the connection, while the content uniquely describes node characteristics. Let $\mathcal{L} = \{l_1, \ldots, l_m\}$ represent the link weights associated with the corresponding edges $\{e_i\}$ in the network, where each $l_i \in \mathbb{R}$, $\forall i = \{1, \ldots, m\}$. Similarly, let $\mathcal{C} = \{c_1, \ldots, c_n\}$ be the set of content features represented by a vector in some vector space in $\mathbb{R}^d$, so that every $v_i \in \mathcal{V}$ is associated with a content vector denoted by c_i. In addition, supervision information is available about the relative similarity between

nodes. The user supervision (intentional knowledge) is given by triplet constraints of the form:

$$\mathcal{S} = \{(v_i, v_j, v_k) : \ (v_i \text{ and } v_j) \text{ more similar to } (v_i \text{ and } v_k)\}.$$

The triplet setting is generally preferable to the pairwise setting, because comparing two objects in terms of *absolute* similarity is very abstract and subjective [17]. Unlike the traditional pairwise settings, triplet constraints are defined by *comparing* two pairwise similarities. It is worth mentioning that, although we only consider the triplet setup in this paper, our proposed method can be easily extended to other forms of supervision. In summary, we characterize a network, using the representation $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C}, \mathcal{L}, \mathcal{S})$, which includes the graph structure, content and link features, and supervision.

IV. Factorized Similarity Learning on Networks

In this section, we introduce a novel factorization based scheme for learning node-based similarity measures in networks represented as $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C}, \mathcal{L}, \mathcal{S})$ as well as the intuition behind the mathematical abstraction. Our approach models the similarity learning as a matrix completion problem, where it aims at supervised learning the correlation between different nodes using both link and content information so that the completed similarity matrix will correctly reflect the homogeneity between different nodes.

A. Parameterizations and Constraints

In order to model the similarity learning as a matrix completion problem, we formulate $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C}, \mathcal{L}, \mathcal{S})$ in matrix forms. Let $C \in \mathbb{R}^{n \times d}$ and $L \in \mathbb{R}^{n \times n}$ represent the content and link matrices, which are defined as follows. Each row $C_{i\cdot}$ of the content matrix C is the corresponding feature vector $c_i \in \mathcal{C}$. If the link weight $l_p \in \mathcal{L}$ associates with edge $e_p \in \mathcal{E}$ which connects nodes v_i and $v_j \in \mathcal{V}$, then the L_{ij} entry in the link matrix L will be l_p. A nonzero entry L_{ij} in L indicates that a link exists from the node v_i to v_j, with a weight equal to the strength of the link. It is worth pointing out, that both C and L are typically very sparse in practice.

The target of our approach is to learn a matrix $S \in \mathbb{R}^{n \times n}$, which reflects the encoded information in both L and C. The (i, j)th entry of S measures the similarity from node v_i to v_j. The similarity matrix S is not necessarily symmetric, because similarity is usually non-isotropic across the network. Thus, we do not explicitly constrain the symmetry of S, in order to make our model more general. On the other hand, the triplet supervision is modeled as constraints for the space of S, i.e., the similarity matrix S has to obey the user-specified supervision as much as possible. If the supervision suggests that nodes v_i and v_j are more similar to each other, than nodes v_i and v_k, the learned similarity has to reflect the facts by enforcing $S_{ij} > S_{ik}$. However, in term of mathematical abstraction, the strict order relationship is not a compact set regularizing the space of S. Almost all existing optimization approaches do not favor the open set constraints. We leverage the problem by each constraint as a closed half-space. Specifically, we require that S has to be in the set $\mathcal{T}$, which is defined as follows:

$$\mathcal{T} \doteq \{S : S_{ij} \geq S_{ik} + c, \ \forall (v_i, v_j, v_k) \in \mathcal{S}\}. \tag{1}$$

Here, c is the margin controlling the minimal separability of the similar entries. The value of c can be chosen arbitrarily, since the order between candidate nodes is more important than the actual similarity value at each entry of S. Throughout this paper, we set c to be equal to 1 for simplicity. Moreover, it is easy to see that $\mathcal{T}$ is a convex set.

B. Information Encoding

As is generally the case for matrix completion problems, we assume that the rank of S is much less than the number of nodes n in the given network. This is a very natural assumption, because the number of latent factors characterizing different nodes is much smaller than the number of nodes. However, unlike existing matrix completion problems, S also satisfies some partial order constraints. The minimum number of latent topics, that allows S to satisfy all the constraints, indicates the intrinsic rank of the similarity matrix. Both content and link data encoded in the network are traded as side information, to enhance the factorization, followed by intentional knowledge.

To utilize all available information, let S to be a completed matrix using both content information C and link weight matrix L. We factorize S as $S \cong UV$, where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{r \times n}$ are two low-rank matrices such that $r \ll n$. Different terms in the objective function contribute to different aspects of the similarity function. The term $\|S - UV\|_F^2$ penalizes the error by approximating S as two low-rank factors. $\|\cdot\|_F$ is the Frobenius norm of a given matrix, where $\|X\|_F = \sqrt{tr(XX^T)}$ and $tr(\cdot)$ represents the trace of the matrix.

The link information contributes to similarity learning through the following term in the objective function.

$$\|\mathcal{P}_\Omega(S) - \mathcal{P}_\Omega(L)\|_F^2, \tag{2}$$

where Ω is the index set for the observed elements and the projection $\mathcal{P}_\Omega$ is a orthogonal projector defined in [5]: the (i, j)th element of $\mathcal{P}_\Omega(L)$ is equal to L_{ij} if $(i, j) \in \Omega$ and zero otherwise. In other words, we propagate the link information through its non-zero feature weights. This is done, so that the model will have consistent values as suggested by the link features. This term ensures that the similarity matrix S is influenced by the local topological structure.

Furthermore, to encode the content information in our model, we assume that the content matrix C can be factorized as two low-rank matrices that is a shared U and a basis matrix W, where $W \in \mathbb{R}^{r \times d}$. The third term in the objective function contains the sum of errors of two matrix factorizations, among which the matrix U is common. This ensures the propagation of similarity information from C to S.

$$\|S - UV\|_F^2 + \|C - UW\|_F^2. \tag{3}$$

Note that S has already encoded the link information through the objective function term represented by equation (2). The intuition behind these two terms in equation (3), is that the projections from link and content to a common latent space are identical. If we assume that both V and W are orthonormal, then we multiply V^T and W^T on both sides of the equations $S = UV$ and $C = UW$. We obtain the following: $SV^T = U$ and $CW^T = U$. The similarity matrix S, which encodes the link information and the content matrix C, are projected into a common subspace U through projections V^T and W^T.

Therefore, the content and link information can be bridged coherently using the aforementioned scheme, so that the learned similarity matrix S is consistent with both content and link information globally and locally.

C. Integrated Objective Function

According to the discussion in previous sections, we integrate all the aforementioned parts into a coherent learning framework as:

$$\begin{aligned} &\min_{U,V,W,S} \ \|\mathcal{P}_\Omega(S) - \mathcal{P}_\Omega(L)\|_F^2 + \lambda_1\|S - UV\|_F^2 + \lambda_2\|C - UW\|_F^2 \\ &\text{subject to: } S \in \mathcal{T}, VV^T = I_r, WW^T = I_r. \end{aligned} \tag{4}$$

However, the objective in Eq. (4) has two problems, which lead to inefficient optimization algorithms. The first problem is that the first term in the above objective function contains a projection of non-zero entries in the link matrix. $P_\Omega(L)$ can be viewed as indicator function of all non-zero entries of L, which is discrete. Integer programming solvers are usually quite slow. To alleviate these challenges, we introduce a transition variable $T \in \mathbb{R}^{n\times n}$ acting as a bridge to transfer knowledge from L to S. Then, we are able to convert the projection / indicator term in equation (4) to a new set of constraints on T. Another issue is the orthonormal constraints on both V and W. Not only the orthogonal constraints introduce more non-convexity into the objective, they also make the algorithms more complex [44]. Alternatively, we can relax the orthogonal constraint. To prevent overfitting, we introduce Frobenius norms on both V and W. To this end, we reformulate objective function (4) as follows:

$$\begin{aligned} &\min_{U,V,W,T,S} \ \|S - T\|_F^2 + \lambda_1\|S - UV\|_F^2 + \lambda_2\|C - UW\|_F^2 \\ &\qquad + \lambda_3(\|V\|_F^2 + \|W\|_F^2) \\ &\text{subject to: } \mathcal{P}_\Omega(L) = \mathcal{P}_\Omega(T), S \in \mathcal{T}. \end{aligned} \tag{5}$$

D. Optimization

In this subsection, we demonstrate that the optimization problem in equation (5) can be solved efficiently and effectively using the block coordinate descent method [4], which seeks the optimal value for one particular variable, while fixing others. Though the formulation is non-convex, each subproblem in block coordinate descent is convex. The key here is in solving for each of the variable sets U, V, W, T and S, while keeping the others fixed.

1) Solving for U: Fixing parameters V, W, T, S to optimize U, the objective function (5) reduces to a standard convex unconstrained quadratic program as follows:

$$\min_U \ \lambda_1\|S - UV\|_F^2 + \lambda_2\|C - UW\|_F^2. \tag{6}$$

By determining the derivative of the aforementioned objective with respect to U, and setting it to zero, we obtain:

$$-2\lambda_1(S - UV)V^T - 2\lambda_2(C - UW)W^T = 0, \tag{7}$$

We can obtain an analytic solution for the global minimum:

$$U^* = (\lambda_1 SV^T - \lambda_2 CW^T)(\lambda_1 VV^T + \lambda_2 WW^T)^\dagger, \tag{8}$$

where $(\cdot)^\dagger$ indicates the pseudo-inverse for a given matrix.

2) Solving for V: Similar to solving for U, the matrix V can be solved as a standard unconstrained ridge regression problem, and the objective function can be written as follows:

$$\min_V \lambda_1\|S - UV\|_F^2 + \lambda_3\|V\|_F^2. \tag{9}$$

As in the previous case, we can determine the first order derivative of the objective function in equation (9) with respect to V to be zero as follows:

$$-2\lambda_1 U^T(S - UV) + 2\lambda_3 V = 0, \tag{10}$$

The aforementioned equation can be solved in order to obtain a global minimum for V.

$$V^* = (U^T U + \tfrac{\lambda_3}{\lambda_1} I_r)^{-1} U^T S. \tag{11}$$

where I_r is an identity matrix of size $r \times r$.

3) Solving for W: Solving for W is almost identical to solving for V. By fixing U, V, T and S, we can write the objective function and the analytical solution for the optimal value of W as follows:

$$\min_W \lambda_2\|C - UW\|_F^2 + \lambda_3\|W\|_F^2, \tag{12}$$

The optimal value for W is as follows:

$$W^* = (U^T U + \tfrac{\lambda_3}{\lambda_1} I_r)^{-1} U^T C. \tag{13}$$

4) Solving for T: When we solve for T, while keeping the remaining parameters fixed, we obtain a constrained least squares minimization problem:

$$\min_T \ \|S - T\|_F^2 \quad \text{s.t.:} \quad \mathcal{P}_\Omega(L) = \mathcal{P}_\Omega(T). \tag{14}$$

The equality constraints ensures that non-zero entries of the link matrix L are consistent with the corresponding position on T. Since it is a convex problem, the standard technique for solving equation (14) is first sets $T = S$, and then applies the orthogonal projection on T. In particular, we set the the entries of T in Ω to be the same, as the corresponding value of L. The compressed analytical solution for S can be written as $T^* = S + (\mathcal{P}_\Omega(L) - \mathcal{P}_\Omega(S))$.

5) Solving for S: At this point, we can also solve for S, so that equation (5) is minimized. To do so, we obtain the following optimization problem:

$$\min_S \ \|S - T\|_F^2 + \lambda_1\|S - UV\|_F^2 \quad \text{s.t.: } S \in \mathcal{T}. \tag{15}$$

The objective function can be further compressed by a least square term as $\|S - \frac{1}{1+\lambda_1}(T + \lambda_1 UV)\|_F^2$. Since the set $\mathcal{T}$ is a convex set, the problem in Eq. (15) is again a convex constrained optimization problem, which can be solved using projected gradient methods [3], [27]. The proximal operator associated with Eq. (15) is in the form of projecting a point to the intersection of a set of halfspaces $\mathcal{T} = \cap_{i=1}^{|S|} T_i \neq \emptyset$, which can solved using proximal splitting methods [6]. Moreover, we observe that our objective is a simple projection problem, and thus we can use the successive projection algorithm to solve it efficiently [12]. This has the effect of avoiding expensive line search procedures. The optimal S is obtained by first set is as $\frac{1}{1+\lambda_1}(T + \lambda_1 UV)$ then project it onto the convex set $\mathcal{T}$. We now provide a closed form solution to the projection into each set T_i.

Definition 1: A mapping $\Pi_\mathcal{T} : \mathbb{R}^{n\times n} \to \mathcal{T}$ is a projection associated with convex set $\mathcal{T}$, if it satisfies for any $S \in \mathbb{R}^{n\times n}$, $\Pi_\mathcal{T}(S)$ is the unique matrix in $\mathcal{T}$ that is closest to S, i.e.,

$$\|S - \Pi_\mathcal{T}(S)\| \leq \|S - S'\|, \ \ \forall\ S' \in \mathcal{T}, S \in \mathbb{R}^{n\times n}$$

with equality if and only if $S' = \Pi_\mathcal{T}(S)$.

Theorem 1: Suppose that $\mathcal{T}_m = \{S : S_{ij} \geq S_{ik} + 1\}$. Then, for any $S \in \mathbb{R}^{n\times n}$ the projection from S to the convex set $\mathcal{T}_m$ is as follows:

$$\Pi_{\mathcal{T}_m}(S) = S^* = S \;\text{ if }\; S \in \mathcal{T}_m,$$

Furthermore, if $S \notin \mathcal{T}_m$, then the following is true:

$$\Pi_{\mathcal{T}_m}(S) = S^* = \begin{cases} S^*_{ij} = \frac{1}{2}(1 + S_{ij} + S_{ik}) \\ S^*_{ik} = \frac{1}{2}(-1 + S_{ij} + S_{ik}) \\ S^*_{pq} = S_{pq} \;\; \forall\{p,q\} \neq \{i,j\} \text{ and } \{i,k\}. \end{cases}$$

Proof: For any $S \in \mathcal{T}_m$, we have the trivial solution that the projection is itself. For any $S \notin \mathcal{T}_m$, we are seeking the optimal value of S^*, such that the projection error $\|S - S^*\|_F^2$ is minimized. In other words, the solution to the minimization problem of $\min_{S^*\in\mathcal{T}_m} \|S - S^*\|_F^2$ provides the projector. Because the Frobenius norm is decoupled for every element, it follows that $\mathcal{T}_m$ only affects the entries of S^*_{ij} and S^*_{ik}. Therefore, by choosing $S^*_{pq} = S_{pq}$, we obtain zero projection error for S^*_{pq} for all $\{p,q\} \neq \{i,j\}$ and $\{i,k\}$. The minimization problem is further reduced to the following:

$$\min_{S^*_{ij} \geq S^*_{ik}+1} (S_{ij} - S^*_{ij})^2 + (S_{ik} - S^*_{ik})^2.$$

We observe the following property of the optimal solution:

Lemma 1: For any x, y, x' and $y' \in \mathbb{R}$ such that $x' \leq y' - c$, where $c \in \mathbb{R}^+$, $x' = \frac{1}{2}(-c + x + y)$ and $y' = \frac{1}{2}(c + x + y)$ provides the minimal value of the least squares function $f(x, y, x', y') = (x - x')^2 + (y - y')^2$ if $x > y + c$. For $x \leq y - c$, the minimal $f(x, y, x', y')$ is obtained by setting $x' = x$ and $y' = y$.

Apply the above lemma we obtain the optimal least square solution for S^*_{ij} and S^*_{ik} as

$$S^*_{ij} = \tfrac{1}{2}(1 + S_{ij} + S_{ik}) \text{ and } S^*_{ik} = \tfrac{1}{2}(-1 + S_{ij} + S_{ik}).$$

This completes the proof. ■

The proof of Lemma 1 is provided as the followings:

Proof: The problem can be formulated as a constrained convex program as

$$\min_{x',y'} \quad (x' - x)^2 + (y' - y)^2 \quad \text{subject to: } x' \leq y' - c.$$

The optimal solution can be interpreted as numerically solving the KKT system of equations [4]. The Lagrangian dual problem is

$$\max_{\lambda} \;\; \min_{x',y'} \;\; (x' - x)^2 + (y' - y)^2 + \lambda(x' - y' + c),$$

where λ is so called the KKT multiplier. The optimal x'^* and y'^* is achieved if satisfies some regularity conditions such as: the *stationarity*

$$\begin{cases} 2(x' - x) + \lambda = 0 \\ 2(y' - y) - \lambda = 0 \end{cases} \Rightarrow \begin{cases} x' = -\frac{1}{2}\lambda + x \\ y' = \frac{1}{2}\lambda + y \end{cases},$$

the *primal feasibility* $x' - y' + c \leq 0$, the *dual feasibility* $\lambda \geq 0$, and the *complementary slackness* $\lambda(x' - y' + c) = 0$. By solving the system of equations we obtain the optimal solution of x'^* and y'^* as

$$\text{if } \lambda = 0 \text{ then } \begin{cases} x'^* = x \\ y'^* = y \end{cases}, \quad \text{otherwise } \begin{cases} x'^* = (-c + x + y)/2 \\ y'^* = (c + x + y)/2 \end{cases}$$

This thus completes the proof. ■

We conclude this subsection by illustrating the optimization scheme for the proposed FSL method in algorithm 1.

Algorithm 1: Factorized Similarity Learning

Input: Content matrix C, link matrix L and ordered constraint set $\mathcal{T}$
Output: Similarity matrix S

Initialize: U, V, W, T and S
repeat
  $U = (\lambda_1 SV^T - \lambda_2 CW^T)(\lambda_1 VV^T + \lambda_2 WW^T)^\dagger$;
  $V = (U^T U + \frac{\lambda_3}{\lambda_1} I_r)^{-1} U^T S$;
  $W = (U^T U + \frac{\lambda_3}{\lambda_1} I_r)^{-1} U^T C$;
  $T^* = S + (\mathcal{P}_\Omega(L) - \mathcal{P}_\Omega(S))$;
  $S = \frac{1}{1+\lambda_1}(T + \lambda_1 UV)$;
  Slice S in row-wise into $\{S_{i\cdot}\}_{i=1}^n$ to compute parallel;
  for $i = 1 \ldots n$ **do**
    repeat
      if $S_{ij} < S_{ik} + 1 \;\forall(i,j,k) \in \mathcal{S}$ **then**
        $S_{ij} = \frac{1}{2}(1 + S_{ij} + S_{ik})$
        $S_{ik} = \frac{1}{2}(-1 + S_{ij} + S_{ik})$
      end
    until *all constraint satisfied*;
  end
until *converge or maximum iteration exceed*;
return S

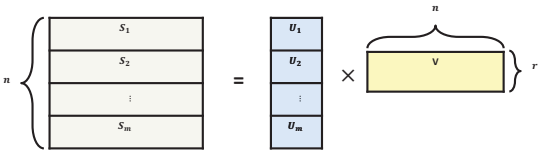

Fig. 2: Large-scale matrix handling.

E. Large-Scale Networks Handling

For a large-scale network, most of commodity hardware cannot hold the similarity matrix S in main memory. This situation is typically arrived at, when the number of nodes exceeds 30,000. In order to alleviate this issue, we will show the proposed method can be easily formulated in a divided and conquer framework.

We first slice the similarity matrix S in row-wise fashion, into different sub-matrices $S_1, \ldots S_m$, where each $S_i \in \mathbb{R}^{(n/m)\times n}$. Then, each S_i can be further expressed as $S_i = U_i V$, where each S_i corresponds to a $(n/m) \times r$ matrix U_i. From the block-wise matrix multiplication, we know if we stack each U_i in column-wise fashion, and multiply by V, the result will be exactly equal to the original $n \times n$ similarity matrix S_i. Figure 2 provides a visual perspective of extending the proposed method into a large-scale framework.

The mathematical abstraction can be directly derived from equation (5) as follows:

$$\begin{aligned} \min_{U_i,V,W,T_i,S_i,\forall i} \;\; & \sum_{i=1}^m \|S_i - T_i\|_F^2 + \lambda_1 \sum_{i=1}^m \|S_i - U_i V\|_F^2 \\ & + \lambda_2 \sum_{i=1}^m \|C_i - U_i W\|_F^2 + \lambda_3(\|V\|_F^2 + \|W\|_F^2) \\ \text{subject to: } & \mathcal{P}_\Omega(L_i) = \mathcal{P}_\Omega(T_i), S_i \in \mathcal{T}_i \;\; \forall i, \end{aligned} \tag{16}$$

Here, C_i, L_i and T_i are the corresponding sliced content, link and bridging matrices. The overall result is that neither the network information, nor the completed similarity matrix S will be stored in main memory as a whole piece, and the memory can be managed much more efficiently.

1) Solving for U_i, T_i and S_i: The process of solving for each U_i, T_i and S_i uses a similar approach. Here, we provide

a detailed optimization scheme for U_i and the similarly idea can be easily extended to solve for T_i and S_i.

Calculating U can been seen as optimizing m sub-problems for each U_i (at a smaller scale), which has no interdependency. Moreover, the solution for U_i is exactly same as before:

$$U_i^* = (\lambda_1 S_i V^T - \lambda_2 C_i W^T)(\lambda_1 V V^T + \lambda_2 W W^T)^\dagger. \quad (17)$$

2) Solving for V and W: Solving for V is slightly different from the case, when we treat matrices S and U as whole. The corresponding Equation (9) is transformed as follows:

$$\min_V \lambda_1 \sum\nolimits_{i=1}^m \|S_i - U_i V\|_F^2 + \lambda_3 \|V\|_F^2, \quad (18)$$

The optimal analytical solution of V is as follows:

$$V^* = \left(\sum\nolimits_i^m U_i^T U_i + \tfrac{\lambda_3}{\lambda_1} I_r\right)^{-1} \left(\sum\nolimits_i^m U_i^T S_i\right). \quad (19)$$

The optimal value of W can be calculated in a similar manner, and that is as follows:

$$W^* = \left(\sum\nolimits_i^m U_i^T U_i + \tfrac{\lambda_3}{\lambda_1} I_r\right)^{-1} \left(\sum\nolimits_i^m U_i^T C_i\right). \quad (20)$$

F. Discussion on Speeding up the Learning

The bottleneck of efficient learning is at the step of updating S or S_i in both conventional and large-scale formulations in equation (15) and (16) respectively. However, the proposed FSL algorithm is able to decouple the row updates of the similarity matrix S, involving supervised projection. Essentially, this can be easily fit into a *MapReduce* framework to significantly boost the training efficiency. Moreover, for the large-scale formulation in equation (16), the low-rank matrices U_i, bridging matrices T_i and the similarity matrix S_i can also be handled in parallel to reduce the running time. While we present these ideas as possibilities for future exploration, a detailed discussion is beyond the scope of this paper. We refer interested readers to [42], and [8] for background on relevant big-data frameworks.

V. Noisy Supervision

Real-world data always contain a significant amount of noise, which could be extremely detrimental to the algorithms. In this section, we explicitly consider the case, where the available supervision is noisy. We show how the proposed method can be integrated with noisy intentional knowledge to yield reliable predictions.

In section III, we model the user intentional knowledge on different samples as a set of triplet constraints $\mathcal{S}$, in which each element in the constraint set is in the form (v_i, v_j, v_k). Specifically, each triplet supervision provides the similarity information on two pairs of nodes with the same query node. When the noise increases, similarity learning could result in poor quality. We illustrate the problem of noisy supervision with a toy example.

Suppose that four different nodes a, b, c, d are given, and the correct underlying similarity order of using a as a query is that $(a, b) > (a, c) > (a, d)$. If $\{(a, b, c), (a, c, d)\}$ is given as the constraint set $\mathcal{S}$, we can order the candidate node b, c, d correctly with respect to reference a. With noisy supervision examples, such as $\{(a, b, c), (a, d, b)\}$ or $\{(a, b, c), (a, d, c), (a, c, d)\}$, the ranking result will either be in an incorrect order, or may have no feasible solution. The inconsistent supervision provides no feasible solution of $S \in \mathcal{T}$ in Equation (5).

The aforementioned toy example suggests that the constraints should be relaxed with the use of slack variables ξ_{ijk}. Intuitively, these slack variables can account for the noise in the objective function. Therefore, the modified optimization problem is as follows:

$$\begin{aligned} \min_{U,V,W,T,S,\xi_{ijk}} \; & \|S - T\|_F^2 + \lambda_1 \|S - UV\|_F^2 + \lambda_2 \|C - UW\|_F^2 \\ & + \lambda_3 (\|V\|_F^2 + \|W\|_F^2) + \lambda_4 \sum\nolimits_{(i,j,k)\in\mathcal{S}} \xi_{ijk} \\ \text{subject to: } & \mathcal{P}_\Omega(L) = \mathcal{P}_\Omega(T), \xi_{ijk} \geq 0, \\ & S_{ij} - S_{ik} \geq 1 - \xi_{ijk} \;\; \forall (i,j,k) \in \mathcal{S}. \end{aligned} \quad (21)$$

It is worth mentioning that the core idea behind such a large-margin relaxation is similar to the formulation of support vector machines (SVM) [36]. It can be solved efficiently using stochastic sub-gradient descent [34] by converting the last two constraints as a penalty term in the objective.

$$\begin{aligned} \min_{U,V,W,T,S} \; & \|S - T\|_F^2 + \lambda_1 \|S - UV\|_F^2 \\ & + \lambda_2 \|C - UW\|_F^2 + \lambda_3 (\|V\|_F^2 + \|W\|_F^2) \\ & + \lambda_4 \sum\nolimits_{(i,j,k)\in\mathcal{S}} \max\{0, 1 - S_{ij} + S_{ik}\} \\ \text{subject to: } & \mathcal{P}_\Omega(L) = \mathcal{P}_\Omega(T), \end{aligned} \quad (22)$$

Here, λ_4 regulates the noise penalty. The term associated with λ_4 is the hinge loss [36].

To solve the optimization problem in equation (22), we follow a similar procedure, as illustrated in algorithm 1 by the block coordinate descent method. The only difference is that we compute the sub-gradient at the step of solving S instead of using the projected gradient methods. By fixing other parameters to compute the optimal value of S, we obtain:

$$\begin{aligned} \min_S \; & f(S) = \|S - T\|_F^2 + \lambda_1 \|S - UV\|_F^2 \\ & + \lambda_4 \sum\nolimits_{(i,j,k)\in\mathcal{S}} \max\{0, 1 - S_{ij} + S_{ik}\}, \end{aligned} \quad (23)$$

This is an unconstrained quadratic programming problem. Furthermore, one of the sub-gradient of $f(S)$ is as follows:

$$\begin{aligned} \tfrac{\partial f(S)}{\partial S} &= 2(S - T) + 2\lambda_1 (S - UV) \\ &+ \lambda_4 \sum\nolimits_{(i,j,k)\in\mathcal{S}} \mathbb{1}\{1 - S_{ij} + S_{ik} \geq 0\}(E_{ik} - E_{ij}), \end{aligned} \quad (24)$$

Here, $\mathbb{1}(\cdot)$ is an indicator function, and $E_{ij} = e_i^T e_j$. Moreover, e_i is the standard unit vector which is a $n \times 1$ vector with only the i^{th} entry set to one, and zero otherwise. We use the line search strategy in our implementation.

VI. Experimental Results

In this section, several experimental results are presented on different data sets in order to validate the effectiveness and efficiency of the proposed *FSL* method. We also present robustness results in terms of parameter sensitivity and noise tolerance. The performance of our *FSL* approach on two real data sets and one synthetic data set outperforms other existing off-the-shelf methods significantly.

TABLE I: The detailed statistics of the data sets.

Data Set	Number of node	Number of edge	Number of node with label	Number of class	Content dimensionality
DBLP	28,702	133,664	4,057	4	13,214
DBLP-clean	2,760	7,636	2,760	4	13,214
CoRA	15,644	59,062	15,644	10	12,313

A. Data sets

The detailed descriptions of the data sets are as follows:

DBLP-Four-Areas Data set: *DBLP* is an online collection of computer science. It is a source of cross-genre information, including content (e.g., keywords of papers) and links (e.g., co-author relationships, and user friendships). In this paper, we use the *DBLP* subset from [9], which contains 28,569 research papers from 28,702 authors, published in 20 conferences. The content information for each paper is extracted from its abstract, and represented using a bag of words. Moreover, 4,057 authors are labeled by four areas, corresponding to database, data mining, information retrieval, and artificial intelligence.

Clean DBLP Data set: A cleaned version of the *DBLP-Four-Areas Data set* [9] is also extracted from the original data set. This cleaned data set, removing all the authors who do not have any connection with others or have any labels, includes 2,760 authors and labeled by four areas. It is utilized to analyze the performance of the proposed algorithm and verify the robustness on parameter selection.

CoRA Data set: This data set is comprised of computer science research papers, and includes full citation graph and the topics (and sub-, sub-subtopics) of each paper[24], resulting in over 80 labels. Instead of using such a huge label space, we used the hierarchical structure of the labels provided by the dataset, and used the higher level labels. In our setting, there are 10 group labels, to identify the class of each paper.

Summary statistics of the data sets are illustrated in Table I.

B. Baseline Methods

We compared our proposed method with a number of state-of-the-art algorithms including the following:

Euclidean Metric: The standard Euclidean distance between content vectors measures the inverse of the similarity between two nodes.

PMF [26]: Probabilistic Matrix Factorization treats the link matrix L as the utility matrix to complete. PMF only utilizes the existing linkage information as observed entries. The stronger a link between a pair of nodes, the greater the similarity between them.

LAD [19]: Locally-Adaptive Decision function learning uses both content and supervision information to learn a local non-isotropic similarity function beyond the traditional generalized Mahalanobis metric.

CFSL: Content-based Factorized Similarity learning is a special case of our *FSL* algorithm by setting $\lambda_4 = 0$ in Equation (22). *CFSL* is still able to incorporate both link and content information in a globally factorized manner.

SSMetric [13]: Semi-supervised Metric learning incorporates knowledge from sparse linkage information and used as neighborhood graph. It is a variant of the originally proposed method, which is modified to allow it to use the linkage structure. The intensional knowledge can be propagated through the link graph L to learn a distance metric on the content vector space.

In summary, the first two baselines learn a similarity measure based only on content or linkage information in an unsupervised manner. *LAD* utilizes both content and supervised knowledge. *CFSL* evaluates the proximity on both contents and links. *SSMetric* is similar to our method in term of incorporating different information sources on content, linkages and supervision.

C. Experimental Settings

In our experiments, we simulated the real-world scenario on similarity learning as a retrieval problem [30], [21]. We start by explaining the experimental settings with an example. As illustrated in figure 3, we divide all pairwise nodes into two disjointed group parameterized by two variables p_v and p_h indicating the level of supervision. For instance, if $p_h = 0.5$ and $p_v = 0.6$, then it means $0.5 + 0.6 \times (1 - 0.5) = 80\%$ of entries are provided supervised knowledge, and the remaining 20% do not have any information about relative ordering. It is worth mentioning that, if we divide the training and testing portions into portions of size 80% and 20%, it does not mean that the full triplet constraints will be given for the training region. Anther hyper-parameter s controls the number of triplet orderings provided for the training region. In our experiments, s is usually set to the range of 5 to 20.

Since the ground truth provided in both the *DBLP* and *CoRA* data sets are explicit multi-class labels, we need to convert them into triplet constraints. One way of achieving this is to generate triplet constraints, is by setting nodes with a same label as similar pair and a different label as dissimilar one. In other words, the triplet constraint $(i, j, k) \in \mathcal{S}$ is generated by randomly choosing two nodes v_i and v_j with the same label. And v_k has a different label with v_i and v_j.

The implementations of *LAD* and *SSMetric* methods use pairwise constraints instead of triplets. Although straightforward conversions exist from pairwise settings to triplet in the most of metric learning based algorithm, we obey their original implementation by converting triplet constraint to pairwise in the following way: each triplet constraint (i, j, k) is split into two different sets that is (v_i, v_j) as a similar pair and (v_i, v_k) as a dissimilar pair. Anther issue for these two baseline is that they are not able to scale-up to a high dimensional setting. Therefore, we perform Principal Component Analysis (PCA) to reduce the dimensionality to $1,000$ as a preprocessing step.

For each data set, we initialize our similarity matrix S by the link matrix L with a small constant value to each entry. The purpose of adding a small constant value in S, is to prevent a row or a column of S without any initial value. Adding a constant value to every entry of the similarity matrix will not affect the performance, since we only emphasize the ordered

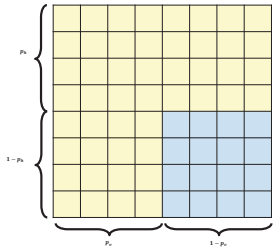

Fig. 3: The experiment settings; yellow region indicates the training while blue is the testing entries.

information instead of the explicit entrywise values. Similar initialization is conducted on the bridging matrix T as well. To initialize the low-rank matrix U, V and W we use a Laplace distribution [16] with zero mean and a scale parameter value of one. In addition, the content matrix C and the link matrix L is normalized to remove the scale variations.

D. Evaluation Measurements

In most recommendation and link prediction applications, the recommended items or the retrieval results are usually presented as the top k most similar candidates to the query. In this case, the accumulated top-k precision and the normalized discounted cumulative gain (NDCG) [23] evaluate the performance effectively among a wide variety of measures. However, in order to compute the NDCG score, we are required to provide a completed ordering information as the ground truth, which is inapplicable to our experimental settings. The precision for a particular value of k, is computed as follows:

$$P@k = \frac{|\text{relevant document} \cap \text{retrieved document}|}{|\text{retrieved document}|}@k.$$

We averaged the precision across different query nodes in the network, and used it as the evaluation metric for our experiments.

E. Results

In this section, we present the results from our proposed *FSL* approach and the aforementioned baseline methods on both *DBLP* and *CoRA* data set. All experimental results were averaged over 10 different runs.

1) DBLP: According to our experimental settings, we provide each node 30 triplet constraints as the intensional knowledge and report the comparative performance with other baseline methods in Figure 4. It is evident that the proposed method achieves the best performance across all ranges of the ranks tested. On the the other hand, link-based *PMF* achieved the poorest performance. The other methods achieved intermediate performance. The *LAD* method achieves the second best performance for learning similarity between authors in the publication network.

An interesting observation is that all methods using linkage information performed worse than the content-based methods, except for the proposed *FSL* scheme. The reason for this is that the noisy links can often hurt the proximity approximation. Predictions from *PMF* methods are based only on the sparse noisy links without any global content bias. *CFSL* utilizes both content and linkage information. However, the noise encoded in the linkage structure prevents good prediction results. *SSMetric* is similar to the proposed *FSL* method which uses linkage, content and supervision simultaneously. However, it is particularly poor at handling noise, because of its inability to prevent similarity propagation along noisy links.

The *LAD* algorithm incorporates the supervised information to learn semantic proximities, which outperform unsupervised content methods. However, the useful information within the linkage structure can not be utilized to enhance the performance. The proposed *FSL* approach is able to identify these unreliable links and eliminate their contributions by transferring and fusing the knowledge from content and supervision. In such a way, influential links can be emphasized, so that *FSL* achieves the best performance.

2) CoRA: Since the *CoRA* data set is somewhat smaller than *DBLP* in terms of the number of nodes and links, we only provided 15 supervised examples per node. We reported the top 50 retrieval results for each baseline method in Figure 5. We obtain similar results to the *DBLP* data set, on which the linkage-based method performed poorly. The *PMF* method obtains the worst result. Although the performance of *CFSL* and *SSMetric* is comparable with the standard Euclidean metric, they are still not quite in the same league as the *LAD* approach.

The proposed method outperforms *LAD* by more than 10%, starting from rank 5, and retains this performance beyond this point. It shows that the proposed *FSL* method not only estimates the proximity of top candidates correctly, but it also retains a very high recall in the retrieval tasks. Our proposed method is very robust, in term of the similarity learning across different data sets.

F. Parameter Sensitivity

The main parameters of the proposed *FSL* algorithm are the weight parameters λ_i, the portion of supervision information s (the number of constraints provided in training for each user), and the rank of matrices U and V (denoted as R). To validate the robustness of parameters and analyze the effect of each parameter on the final result, a group of experiments were conducted on the *Clean DBLP Dataset*. It is a small dataset, obtained by cleaning all the noise from *DBLP*, and contains links, content and four classes. We use the strategy in Section VI-C to generate supervision information.

1) Control Parameters λ_i: The performance with varying λ_1 is shown in Figure 6, in which λ_2 is fixed at 7, $R = 10$ and $s = 12$. λ_1 controls the importance of linkage information considered in factorization. As shown in Figure 6, the performance is stable when $\lambda_1 \geq 1$. The results suggest that as long as sufficient linkage information is provided, the content similarity and supervision can be robustly propagated along the topological structure.

Similarly, the effect of λ_2 is shown in Figure 7, and the performance is robust to parameter setting when $\lambda_2 > 3$. It validates the importance of global (content) information on similarity learning, as hypothesized in Section I. The robustness in parameter choice reflects how optimality is achieved with the help of underlying topological structure spread with linkage information.

A comparison between Figures 6 and 7, yields some interesting observations:

- when λ_1 increases, the performance drops slightly;

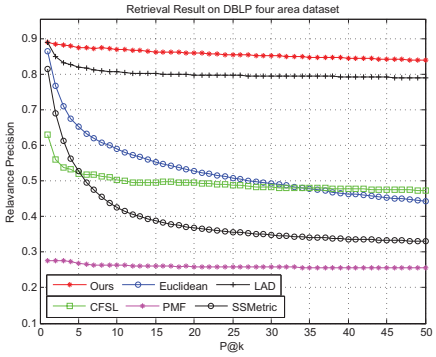

Fig. 4: P@k curve on the *DBLP* data set.

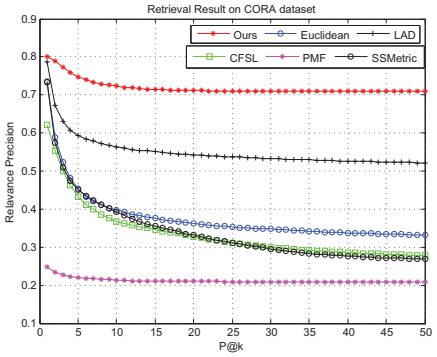

Fig. 5: P@k curve on the *CoRA* data set.

- when λ_2 increases, the performance improves slightly.

This observation is in agreement with our experimental results in Section VI-E. For this particular task assignment, linkage information is not as useful as content similarity.

2) Supervision s: Figure 8 shows the effect of supervision on the *FSL* algorithm, fixing $\lambda_1 = 1.5$, $\lambda_2 = 7$ and $R = 10$. It is obvious that given a certain number of constraints for each user, i.e. $s > 10$, the performance is fairly stable regardless of the value of s. These results suggest the following:

s **increases**: as more supervision is provided, the *FSL* algorithm, will adjust the topological structure of networks relying on trustworthy guidance. In this situation, the information propagation will be more efficient. On the other hand, diminishing returns are achieved for increasing s beyond a certain point.

s **is small**: In this case, the algorithm focuses most of its efforts on fitting a small portion of supervision. This has a detrimental impact on the whole structure of the network. As a result, the performance is not very good in this range.

In this experiment, the percentage of supervision is $p_s = s/N(U)$, which is approximately 4×10^{-4}. This is much smaller than a typical social network, e.g., *Facebook*, where there are hundreds of labeled links (i.e., friendships) on average for each users. Therefore, the algorithm is practical in real-world scenario.

3) Low Rank Approximation: R: Finally, the effect of matrix rank R is shown in Figure 9. As observed from the figure, the performance increases stably after $R \geq 8$. Considering the fact that the samples in the *DBLP* dataset are labeled with 4 classes, it is feasible to assume $R > 4$. Typically, the value assignment of rank R is application-dependent.

G. Noise Tolerance

In this section, we present the performance on error tolerance using the large-margin formulation proposed in equation (21) on the *DBLP-clean* data set. We tested the *FSL* method with different levels of noise in the supervision in figure 10. The color of the histogram indicates the level of noise injection. Furthermore, the different groups in the histogram show the retrieval result at different ranks. We observe that when the noise level is small (1% or 5%) the proposed method maintains very good results, and the retrieval precision decreases very slowly with increasing rank. However, when the noise level becomes high, the *FSL* method obtains a poor recall. Overall, Figure 10 demonstrates that our proposed method is robust to a small-level of error tolerance.

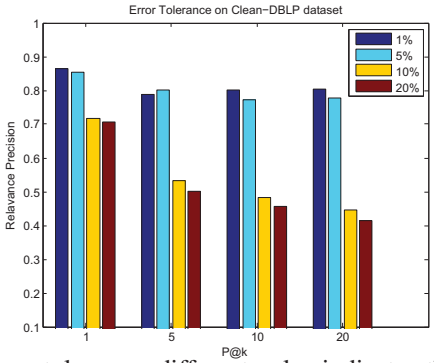

Fig. 10: Error tolerance: different color indicates the percentage of supervision randomly flipped.

VII. Conclusion

In this paper, we proposed a novel learning approach, known as *FSL*, to measure the node-based similarity in networks within a matrix factorization framework. We propose a holistic model, which leverages network topological structure, node content and user supervision. The proposed method is able to ameliorate the impact of noisy linkage structures by fusing and transferring knowledge from other domains. At the same time, the reliable linkages are used effectively in conjunction with content and user-supervision. By embedding content and links into a unified latent space, the supervision can correctly guide the factorization process. We show extensive experiments on real-world data sets. The proposed *FSL* method significantly outperforms other state-of-the-art approaches in node-based retrieval, and is highly robust.

Acknowledgment

This work was founded in part to Shiyu Chang and Thomas S. Huang by the National Science Foundation under Grand No. 1318971 and the Samsung Global Research Program 2013 under Theme "Big Data and Network", Subject "Privacy and Trust Management In Big Data Analysis". This work was partially sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053.

References

[1] C. C. Aggarwal. Towards systematic design of distance functions for data mining applications. In *Proceedings of the ninth ACM SIGKDD*, pages 9–18. ACM, 2003.

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *J. Mach. Learn. Res.*, 6:937–965, Dec. 2005.

[3] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

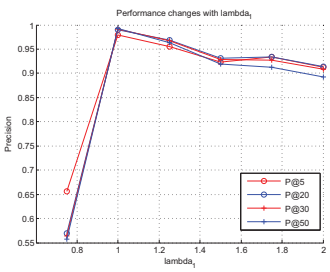

Fig. 6: Parameter testing: λ_1.

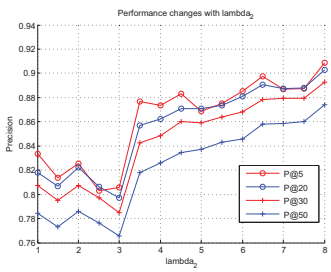

Fig. 7: Parameter testing: λ_2.

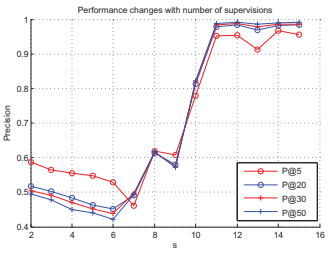

Fig. 8: Parameter testing: number of supervision - s.

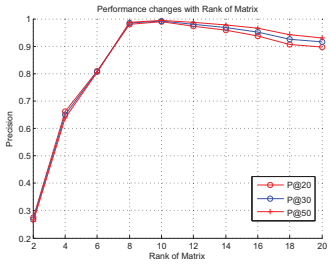

Fig. 9: Parameter testing: number of supervision - R.

[5] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[6] W. Cheney and A. A. Goldstein. Proximity maps for convex sets. *Proc. of the Am. Math. Soc.*, 10(3):448–450, 1959.

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007.

[8] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[9] H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin. Probabilistic topic models with biased propagation on heterogeneous information networks. In *SIGKDD*, pages 1271–1279, 2011.

[10] F. Geerts, H. Mannila, and E. Terzi. Relational link-based ranking. In *VLDB*, pages 552–563, 2004.

[11] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, pages 513–520, 2004.

[12] S.-P. Han. A successive projection method. *Mathematical Programming*, 40(1-3):1–14, 1988.

[13] S. C. H. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. year = 2008. In *CVPR*. IEEE Computer Society.

[14] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *SIGKDD*, pages 538–543, 2002.

[15] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[16] S. Kotz, T. Kozubowski, and K. Podgorski. *The Laplace Distribution and Generalizations: A Revisit With Applications to Communications, Exonomics, Engineering, and Finance*. Progress in Mathematics Series. Birkhäuser Boston, 2001.

[17] N. Kumar, K. Kummamuru, and D. Paranjpe. Semi-supervised clustering with metric learning using relative comparisons. In *Data Mining, Fifth IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[18] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[19] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In *CVPR*, 2013.

[20] Z. Lin, I. King, and M. Lyu. Pagesim: A novel link-based similarity measure for the world wide web. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pages 687–693, 2006.

[21] X. Liu, R. Ji, H. Yao, P. Xu, X. Sun, and T. Liu. Cross-media manifold learning for image retrieval and annotation. In M. S. Lew, A. D. Bimbo, and E. M. Bakker, editors, *Multimedia Information Retrieval*, pages 141–148. ACM, 2008.

[22] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CKIM*, pages 931–940, 2008.

[23] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[24] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[25] M. McPherson, L. Smith-Lovin, and J. M. Cook. *Annual Review of Sociology*, (1):415–444.

[26] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[27] Y. Nesterov and I. E. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

[28] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[29] S. Purushotham, Y. Liu, and C.-C. J. Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. In *ICML*, 2012.

[30] G.-J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. Huang. Exploring context and content links in social media: A latent space method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):850–862, May 2012.

[31] G.-J. Qi, J. Tang, Z.-J. Zha, T.-S. Chua, and H.-J. Zhang. An efficient sparse metric learning in high-dimensional space via l1-penalized log-determinant regularization. In *ICML*, pages 841–848, 2009.

[32] B. Qian, X. Wang, F. Wang, H. Li, J. Ye, and I. Davidson. Active learning from relative queries. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1614–1620. AAAI Press, 2013.

[33] B. Qian, X. Wang, J. Wang, H. Li, N. Cao, W. Zhi, and I. Davidson. Fast pairwise query selection for large-scale active learning to rank. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 607–616. IEEE, 2013.

[34] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, pages 807–814, 2007.

[35] J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua. Inferring semantic concepts from community-contributed images and noisy tags. In *SIGMM*, pages 223–232. ACM, 2009.

[36] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[37] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *SIGKDD*, pages 448 – 456, 2011.

[38] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, June 2009.

[39] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

[40] W. Xi, E. A. Fox, W. Fan, B. Zhang, Z. Chen, J. Yan, and D. Zhuang. Simfusion: Measuring similarity using unified relationship matrix. In *SIGIR*, pages 130–137, 2005.

[41] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, pages 505–512, 2003.

[42] C. Zeng, Y. Jiang, L. Zheng, J. Li, L. Li, H. Li, C. Shen, W. Zhou, T. Li, B. Duan, M. Lei, and P. Wang. Fiu-miner: A fast, integrated, and user-friendly system for data mining in distributed environment. In *SIGKDD*, pages 1506–1509, 2013.

[43] P. Zhao, J. Han, and Y. Sun. P-rank: A comprehensive structural similarity measure over information networks. In *CIKM*, pages 553–562, 2009.

[44] J. Zhou, Z. Lu, J. Sun, L. Yuan, F. Wang, and J. Ye. Feafiner: biomarker identification from medical data through feature generalization and selection. In *SIGKDD*, pages 1034–1042, 2013.