Knowl Inf Syst
DOI 10.1007/s10115-015-0894-8

REGULAR PAPER

Large-scale supervised similarity learning in networks

Shiyu Chang[1] · Guo-Jun Qi[2] · Yingzhen Yang[1] · Charu C. Aggarwal[3] · Jiayu Zhou[4] · Meng Wang[5] · Thomas S. Huang[1]

Received: 10 December 2014 / Revised: 1 July 2015 / Accepted: 10 October 2015
© Springer-Verlag London 2015

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

This paper is an extended journal version of the ICDM 2014 best student paper [6] for the "*Best of ICDM*" special issue.

✉ Shiyu Chang
chang87@illinois.edu

Guo-Jun Qi
guojun.qi@ucf.edu

Yingzhen Yang
yyang58@illinois.edu

Charu C. Aggarwal
charu@us.ibm.com

Jiayu Zhou
jiayuz@msu.edu

Meng Wang
wangmeng@hfut.edu.cn

Thomas S. Huang
t-huang1@illinois.edu

1 Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

2 University of Central Florida, Orlando, FL 32816, USA

3 IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

4 Michigan State University, East Lansing, MI 48824, USA

5 Hefei University of Technology, Hefei 230009, Anhui, China

Published online: 20 October 2015

(FSL) is proposed to integrate the link, node content, and user supervision into a 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* data sets. The results show that *FSL* is robust and efficient and outperforms the state of the art. The code for the learning algorithm used in our experiments is available at http://www.ifp.illinois.edu/~chang87/.

Keywords Supervised network similarity learning · Supervised network embedding · Large-scale network · Supervised matrix factorization · Link content consistency

1 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 [22,24,32,34,35], and recommender systems model user and item profiles from collaborative similarities [16,27]. 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 Fig. 1. The graph demonstrates a generalized network structure, where 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 learn-

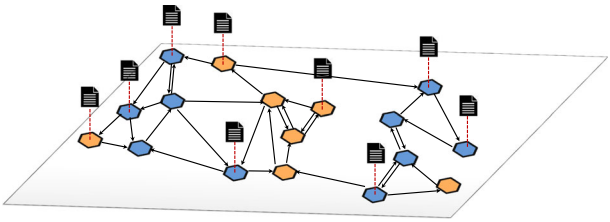


Fig. 1 An example of network structure

ing on networks. This is achieved by integrating these heterogeneous facets into a 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 [22,37]. 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. The enhanced optimization techniques reduce the volume of the parameter space so that fast convergence is assured. To verify the proposed FSL algorithm, we conduct several experiments on different data sets, including *DBLP* scientific bibliography [10] and *CoRA* [25] 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 2 provides an illustrative example to motivate our approach. Section 3 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 Sects. 4 and 5. We then show how the model can handle the case with noisy supervision in Sect. 6. We present extensive experiments on a wide range of data sets in Sect. 7. The conclusion and future research directions are presented in Sect. 8.

2 A motivating example

In this section, we describe a toy example from a real-world scientific author recommendation and retrieval scenario. We show why the direct adoption of content-based or link-based metrics fails to provide good predictions. We consider the top six similar authors calculated from two different metrics of the author *Thomas S. Huang* in the *DBLP-Four-Areas* data set [10], which will be formally introduced in Sect. 7. Table 1 illustrates search results by directly utilizing link weights and content features, respectively. We observed that recommended authors using link information are only Thomas Huang's close collaborators, students, or postdoctoral associates. However, link weights fail to maintain high precisions for a long ranking list because of sparsity issues. The main problem is that of the selection of the proper choice of *indirectly* connected candidates. On the other hand, among authors retrieved from the content source, most of them shared mutual interests for specific scientific topics. One of

Table 1 A motivating example to illustrate the variations in the similarities between nodes from different perspectives

	Link	Content
Rank 1	Shuicheng Yan	Anni R. Bruss
Rank 2	Brendan J. Frey	Qiang Yang
Rank 3	Xiaoou Tang	Takeo Kanade
Rank 4	Ying Wu	Jaime G. Carbonell
Rank 5	Huan Wang	Rong Jin
Rank 6	Antonio Colmenarez	Raghu Ramakrishnan

the drawbacks for such approaches is that each author is usually interested in several research topics or belongs to multiple latent categories. Therefore, the use of a global content measure overlooks the "similarity" in a fine-grid level. From this example, we see that the retrieved results are various a lot from different perspective of similarity measures. Utilizing either of the two along is insufficient to retrieve nodes with similar attributes in networks. Therefore, we seek a unified learning framework that considers both linkage and content information. In addition, we also incorporate the notion of "similar" into model learnings to identify the underlying user intension in this paper.

3 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.

3.1 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 terms 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 include the Neighborhood Component Analysis (NCA) [12] and the Large-Margin Nearest Neighbor classification (LMNN) [41]. However, the performance of the supervised approaches relies 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. [44] proposed to use side information, instead of class labels. The side information is presented as pair-wise 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) [8].

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) [33] works under pair-wise relevance constraints to produce sparse metrics which significantly reduce the number of parameters, so that the time required for learning reduces dramatically. Another

issue, which 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) [20], learns a nonisotropic similarity function by a joint model 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.

3.2 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* [30] , 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* [43], *Pagesim* [21], and the relational like-base ranking [11].

An interesting method, known as *SimRank* [15], is an 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* [46] 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 [26], which are insufficient for fully capturing the underlying semantics.

3.3 Matrix factorization

Matrix factorization is one of the most popular methods in matrix completion and recommendation. Typically, the factorization assumes, that there are 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) [42], nonnegative matrix factorization (NMF) [19] and probabilistic matrix factorization (PMF) [27], 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 [23,31]. 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 [40]. However, all the approaches are unsupervised and also do not work well in noisy content-centric scenarios.

4 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 $d-$dimensional 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 pair-wise setting, because comparing two objects in terms of *absolute* similarity is very abstract and subjective [18]. Unlike the traditional pair-wise settings, triplet constraints are defined by *comparing* two pair-wise 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.

5 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.

5.1 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 nodes v_i to v_j. The similarity matrix S is not necessarily symmetric, because similarity is usually nonisotropic 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 terms 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, the following convexity result holds:

Lemma 1 *The set $\mathcal{T}$, as defined in Eq. (1), is convex.*

Proof $\mathcal{T}$ can be expressed as the intersection of $|\mathcal{S}|$ sets as $\mathcal{T} = T_1 \cap \cdots \cap T_{|\mathcal{S}|}$. Each T_m involves a set of triplet supervision. Without loss of generality, assume $T_m = \{S : S_{ij} \geq S_{ik} + 1\}$. It can be easily verified that T_m is a convex set by the definition of convex sets by assuming $S^1, S^2 \in T_m, \alpha \in [0, 1]$. Then, the following is true:

$$\begin{aligned} \alpha S^1_{ij} + (1-\alpha) S^2_{ij} &\geq \alpha (S^1_{ik} + 1) + (1-\alpha)(S^2_{ik} + 1) \\ &\geq \alpha S^1_{ik} + (1-\alpha) S^2_{ik} + 1. \end{aligned}$$

Therefore, $\alpha S^1 + (1-\alpha) S^2 \in \mathcal{T}_m$ and $\mathcal{T}_m$ is a convex set. Furthermore, $\mathcal{T}$ is an intersection of a finite number of convex sets. Therefore, $\mathcal{T}$ is convex. □

5.2 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, which 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 nonzero 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 Eq. (2). The intuition behind these two terms in Eq. (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 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. A graphical illustration on how different information sources are fused and transferred to contribute to learning node-based similarity is shown in Fig. 2.

5.3 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 nonzero entries in the link matrix. $P_\Omega(L)$ can be viewed as indicator function of all nonzero 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

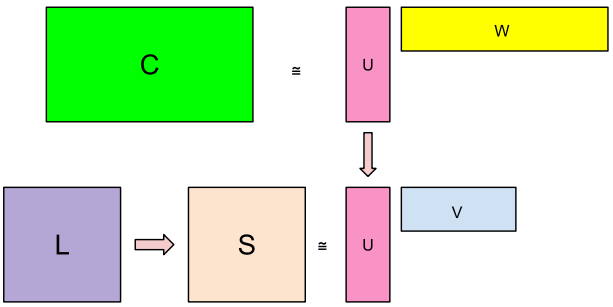


Fig. 2 An idea illustration for integrating different information sources in networks

term in Eq. (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 nonconvexity into the objective, they also make the algorithms more complex [47]. 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:

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

5.4 Optimization

In this subsection, we demonstrate that the optimization problem in Eq. (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 nonconvex, 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.

5.4.1 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.

5.4.2 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 Eq. (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 $r \times r$.

5.4.3 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:

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

5.4.4 Solving for T

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

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

The equality constraints ensure that nonzero 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 Eq. (14) first sets $T = S$ and then applies the orthogonal projection on T. In particular, we set 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 $T^* = S + (\mathcal{P}_\Omega(L) - \mathcal{P}_\Omega(S))$.

5.4.5 Solving for S

At this point, we can also solve for S, so that Eq. (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 [3,28]. The proximal operator associated with Eq. (15) is in the form of projecting a point to the intersection of a set of half-spaces $\mathcal{T} = \cap_{i=1}^{|\mathcal{S}|} T_i \neq \emptyset$, which can solved using proximal splitting methods [7]. Moreover, we observe that our objective is a simple projection problem, and thus, we can use the successive projection algorithm to solve it efficiently [13]. This has the effect of avoiding expensive line search procedures. The optimal S 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 that 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.,

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

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

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

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

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

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

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 2 *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* $y' = \frac{1}{2}(c + x + y)$ *provide 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* $y' = y$.

Applying the above lemma, we obtain the optimal least-squares solution for S^*_{ij} and S^*_{ik} as

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

This completes the proof.

The proof of Lemma 2 is provided as the followings:

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

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

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 the so-called KKT multiplier. The optimal x'^* and y'^* are achieved if it satisfies some regularity conditions such as the *stationarity*

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

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

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.

5.5 Large-scale network 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 submatrices $S_1, \ldots S_m$, 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 that 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. By doing so, it provides significant memory efficiency gain. Instead of storing a $n \times n$ matrix S, we only require $(n/m) \times n$ floating point space. In an extrema case of $n = m$, we achieve the lowest memory cost. Figure 3 provides a visual perspective of extending the proposed method into a large-scale framework.

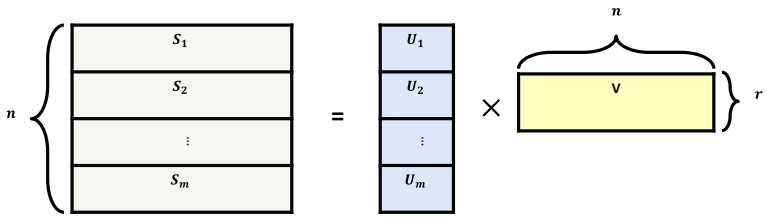


Fig. 3 Large-scale matrix handling

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

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

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.

5.5.1 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 be seen as optimizing m subproblems for each U_i (at a smaller scale), which has no interdependency. Moreover, the solution for U_i is exactly same as before:

$$U_i^* = \left(\lambda_1 S_i V^T - \lambda_2 C_i W^T\right)\left(\lambda_1 V V^T + \lambda_2 W W^T\right)^\dagger. \tag{17}$$

5.5.2 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 Eq. (9) is transformed as follows:

$$\min_V \lambda_1 \sum_{i=1}^{m} \|S_i - U_i V\|_F^2 + \lambda_3 \|V\|_F^2, \tag{18}$$

The optimal analytical solution of V is as follows:

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

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

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

5.6 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 Eqs. (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 Eq. (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 [45], and [9] for background on relevant big data frameworks.

6 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 Sect. 4, 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 Eq. (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:

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

It is worth mentioning that the core idea behind such a large-margin relaxation is similar to the formulation of support vector machines (SVM) [39].

6.1 Optimization

Equation (21) can be solved using stochastic subgradient descent [36] by converting the last two constraints as a penalty term in the objective.

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

Here, λ_4 regulates the noise penalty. The term associated with λ_4 is the hinge loss [39].

To solve the optimization problem in Eq. (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 subgradient 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:

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

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

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

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 $1 \times n$ vector with only the i^{th} entry set to one, and zero otherwise. We use the line search strategy in our implementation.

6.2 An efficient dual solver

Instead of the subgradient method, we derive the dual form of the optimization problem (23) which possesses a efficient solution. Using the slack variables $\{\xi_{ijk}\}$, (23) is equivalent to

$$\begin{aligned} \min_S \ & \|S-T\|_F^2 + \lambda_1 \|S-UV\|_F^2 + \lambda_4 \sum_{(i,j,k)\in\mathcal{S}} \xi_{ijk}, \\ \text{subject to: } & \xi_{ijk} \geq 0, \quad S_{ij} - S_{ik} \geq 1 - \xi_{ijk} \quad \forall (i,j,k) \in \mathcal{S}. \end{aligned} \tag{25}$$

Note that each row of S in the primal problem (25) can be optimized separately, since the rows of S are independent of each other in both the objective function and the constraints. Similar to what we discussed in the Sects. 5.5 and 5.6, solving S in a row-wise manner significantly facilitates large- scale applications and benefits from parallel computing. Let S_i denote the i-th row of S, and then, the optimization problem for each S_i is written as:

$$\begin{aligned} \min_{S_i} \ & \|S_i - T_i\|_2^2 + \lambda_1 \|S_i - U_i V\|_2^2 + \lambda_4 \sum_{(i,j,k)\in\mathcal{S}} \xi_{ijk}, \\ \text{subject to: } & \xi_{ijk} \geq 0, \ S_{ij} - S_{ik} \geq 1 - \xi_{ijk}, \end{aligned} \tag{26}$$

which is a constrained convex optimization problem. It can be solved by its dual problem due to the strong duality according to Slater's condition [4]. In the sequel, we show that the dual problem is a box-constrained quadratic programming problem which can be solved efficiently by the coordinate descent algorithm. As opposed to the subgradient method for the primal problem, the limited inequality constraints lead to a dual problem that can be solved much faster by coordinate descent.

With the dual variables $\alpha_{ijk} \geq 0$ and $\beta_{ijk} \geq 0$ for the inequality constraints, the Lagrangian of the optimization problem (26) is

$$\begin{aligned}\mathcal{L}(S_i, \xi, \alpha, \beta) = \|S_i - T_i\|_2^2 + \lambda_1 \|S_i - U_i V\|_2^2 + \lambda_4 \sum_{(i,j,k)\in\mathcal{S}} \xi_{ijk} \\ - \sum_{(i,j,k)\in\mathcal{S}} \big(\alpha_{ijk}(S_{ij} - S_{ik} + \xi_{ijk} - 1) + \beta_{ijk}\xi_{ijk}\big).\end{aligned} \tag{27}$$

Taking derivative of $\mathcal{L}$ with respect to S_i and $\{\xi\}$, we have

$$\frac{\partial \mathcal{L}}{\partial S_i} = 2(S_i - T_i) + 2\lambda_1(S_i - U_i V) - \sum_{(i,j,k)} \alpha_{ijk}(e_j - e_k), \tag{28}$$

and

$$\frac{\partial \mathcal{L}}{\partial \xi_{ijk}} = \lambda_4 - \alpha_{ijk} - \beta_{ijk}, \quad \forall (i, j, k) \in \mathcal{S}. \tag{29}$$

Letting derivatives in Eqs. (28) and (29) be zero, we have

$$\begin{aligned} S_i^* = &\frac{\sum_{(i,j,k)\in\mathcal{S}} \alpha_{ijk}(e_j - e_k) + 2\lambda_1 U_i V + 2T_i}{2 + 2\lambda_1}, \text{ and} \\ &\alpha_{ijk} + \beta_{ijk} = \lambda_4, \quad \forall (i, j, k) \in \mathcal{S}. \end{aligned} \tag{30}$$

We further denote by α_i, β_i, ξ_i the vectorization of α_{ijk}, β_{ijk}, and ξ_{ijk} with $(i, j, k) \in \mathcal{S}$, respectively. $\mathcal{R}_i = \{(j, k) : (i, j, k) \in \mathcal{S}\}$ is used to represent indices of the elements of S_i that appear in the constraints, and α_i, β_i, ξ_i are of size $1 \times |\mathcal{R}_i|$. Moreover, we define the matrix M_i of size $|\mathcal{R}_i| \times n$ whose rows are comprised of $\{e_j - e_k, (i, j, k) \in \mathcal{S}\}$, and the rows of M are arranged in the order such that $\alpha_i M_i = \sum_{(i,j,k)\in\mathcal{S}} \alpha_{ijk}(e_j - e_k)$. With these new notations, S_i^* can be rewritten as

$$S_i^* = \frac{\lambda_1 U_i V + T_i}{1 + \lambda_1} + \frac{\alpha_i M_i}{2 + 2\lambda_1}. \tag{31}$$

Substituting S_i^* (31) into the Lagrangian (27), we obtain the dual problem below, which is a box-constrained quadratic programming (QP) problem:

$$\min_{\alpha_i} \; P(\alpha_i) = \frac{1}{2}\alpha_i Q_i \alpha_i^T - \alpha_i r_i^T \quad \text{subject to: } 0 \leq \alpha_i \leq \lambda_4, \tag{32}$$

where

$$Q_i = \frac{M_i M_i^T}{2(1 + \lambda_1)}, \quad r_i = \mathbf{1} - \frac{(\lambda_1 U_i V + T_i) M_i^T}{1 + \lambda_1}. \tag{33}$$

$\mathbf{1}$ is an all-ones $1 \times \mathcal{R}_i$ vector, and the inequality in (32) is the element-wise inequality. In addition, according to the KKT conditions, the optimal solution of the primal and dual variables should satisfy:

$$\begin{cases} \alpha_{is}(S_{ij} - S_{ik} + \xi_{ijk} - 1) = 0 \\ \beta_{is}\xi_{is} = 0 \\ \alpha_{is} + \beta_{is} = \lambda_4, \quad \alpha_{is} \geq 0, \quad \beta_{is} \geq 0, \end{cases}$$

where α_{is} is the s-th element of α_i and $1 \leq s \leq |\mathcal{R}_i|$. j, k are the indices that correspond to the s-th constraint. Combined with the primal constraints in Eq. (26), it follows that

$$\begin{cases} \alpha_{is} = 0 \ \Rightarrow \ S_{ij} - S_{ik} \geq 1 \\ 0 < \alpha_{is} < \lambda_4 \ \Rightarrow \ S_{ij} - S_{ik} = 1 \\ \alpha_{is} = \lambda_4, \ \Rightarrow \ S_{ij} - S_{ik} \leq 1. \end{cases} \tag{34}$$

Note that Q_i is positive semi-definite, but it may not be positive definite. Also, it can be verified that the diagonal elements of Q_i are all $\frac{1}{1+\lambda_1}$ since the diagonal elements of $M_i M_i^T$ are all 2. We use coordinate descent method [38] to solve the optimization problem (32). In each iteration of the coordinate descent, the objective function $P(\alpha_i)$ is minimized in a coordinate-wise manner. Suppose $\alpha_i^t = \{\alpha_{i1}^t, \alpha_{i2}^t, \ldots, \alpha_{i|\mathcal{R}_i|}^t\}$ is the value of α_i in t-th iteration for $t \geq 0$, the coordinate descent method minimizes α_{is} for $s = 1, 2, \ldots, |\mathcal{R}_i|$ with other elements fixed:

$$\begin{aligned} \alpha_{i1}^{t+1} &= \arg\min_{\alpha_{i1}} P(\alpha_{i1}, \alpha_{i2}^t, \ldots, \alpha_{i|\mathcal{R}_i|}^t) \\ &\ldots \\ \alpha_{is}^{t+1} &= \arg\min_{\alpha_{is}} P(\alpha_{i1}^{t+1}, \alpha_{i2}^{t+1}, \ldots, \alpha_{is}, \alpha_{i(s+1)}^t, \ldots, \alpha_{i|\mathcal{R}_i|}^t) \\ \alpha_{i|\mathcal{R}_i|}^{t+1} &= \arg\min_{\alpha_{i|\mathcal{R}_i|}} P(\alpha_{i1}^{t+1}, \alpha_{i2}^{t+1}, \ldots, \alpha_{i(|\mathcal{R}_i|-1)}^{t+1}, \alpha_{i|\mathcal{R}_i|}). \end{aligned} \tag{35}$$

To illustrate the coordinate-wise minimization in (35), we show how to optimize over α_{is} with all the remaining elements $\{\alpha_{i1}, \ldots, \alpha_{i(s-1)}, \alpha_{i(s+1)}, \ldots, \alpha_{i|\mathcal{R}_i|}\}$ fixed. In this case, the optimization problem of Eq. (32) is reduced to

$$\min_{\alpha_{is}} \ P(\alpha_{is}) = \frac{1}{2(1+\lambda_1)}\alpha_{is}^2 - R_s\alpha_{is}, \ \text{s.t.:} \ 0 \leq \alpha_{is} \leq \lambda_4, \tag{36}$$

where $R_s = r_{is} - \sum_{u \neq s} \alpha_{iu}(Q_i)_{su}$. Equation (35) is an univariate QP problem, and $P(\alpha_{is})$ achieves its minimum at

$$\alpha_{is}^* = \begin{cases} \lambda_4 & : \ R_s(1+\lambda_1) > \lambda_4 \\ R_s(1+\lambda_1) & : \ 0 \leq R_s(1+\lambda_1) \leq \lambda_4 \\ 0 & : \ R_s(1+\lambda_1) < 0. \end{cases} \tag{37}$$

The coordinate descent algorithm for the dual problem (32) for each $1 \leq i \leq n$ is summarized in Algorithm 2.

6.3 Theoretical guarantees

Let P^* denote the minimum value of the objective function for the dual problem (32), and $\{\alpha_i^t\}_{t=1}^{\infty}$ be the sequence obtained by the coordinate descent Algorithm 2 with $\varepsilon_0 = 0$ and $\tau_{\max} = \infty$. Based on the property of coordinate descent Algorithm [38], Algorithm 2 converges and obtains the globally optimal solution to the dual problem (32). In fact, since the sequence $\{\alpha_i^t\}_{t=1}^{\infty}$ is bounded, it contains a subsequence that converges to the optimal solution to (32) where the optimality condition is met. In practice, the stopping threshold ε_0

Algorithm 2: Coordinate Descent Algorithm for the Dual Problem (32)

Input: U_i, V, λ_4, $\mathcal{R}_i$: the constraint set, α_i^0: the initial value of α_i, ε_0: the stopping threshold, $\tau_{\max}$: the maximum number of iteration

Output: The i-th row of the similarity matrix S_i

Initialize: $t = 0$, α_i^0 is set to be a all zeros vector. Compute Q_i, r_i according to (33).

for $t = 0 \ldots \tau_{max} - 1$ **do**

  for $s = 1 \ldots |\mathcal{R}_i|$ **do**

    $R_s = r_{is} - \sum_{u \neq s} \alpha_{iu}(Q_i)_{su}$, R_s is computed using $\{\alpha_{i1}^{t+1}, \ldots, \alpha_{i(s-1)}^{t+1}, \alpha_{i(s+1)}^{t}, \ldots, \alpha_{i|\mathcal{R}_i|}^{t}\}$.

    $$\alpha_{is}^{t+1} = \begin{cases} \lambda_4 & : \ R_s(1+\lambda_1) > \lambda_4 \\ R_s(1+\lambda_1) & : \ 0 \leq R_s(1+\lambda_1) \leq \lambda_4 \\ 0 & : \ R_s(1+\lambda_1) < 0 \end{cases}$$

  end

  if $\|\alpha_i^{t+1} - \alpha_i^t\|_2 < \varepsilon_0$ **then**

    The algorithm converges and break

  end

  $t = t + 1$

end

Compute $S_i = \frac{\lambda_1 U_i V + T_i}{1+\lambda_1} + \frac{\alpha_i^* M_i}{2+2\lambda_1}$ using the obtained optimal solution α_i^*.

return S_i

is a small positive number and $\tau_{\max}$ is finite. For $\varepsilon_0 > 0$, Theorem 2 gives the upper bound for the number of iterations required for the convergence of Algorithm 2.

Theorem 2 *The coordinate descent Algorithm 2 converges after at most* $\left\lceil \frac{2|\mathcal{R}_i|(P_0 - P^*)(1+\lambda_1)}{\varepsilon_0^2} \right\rceil$ *iterations, where* $P_0 = P(\alpha_i^0)$ *is the initial value of the objective function.*

Proof First of all, since P is a contentious function that defined on a compact set specified by $0 \leq \alpha_i \leq \lambda_4$, the range of P is also a compact set, and it follows that P^* exists and $-\infty < P^* < \infty$. In the following text, we will prove that by each iteration of coordinate descent described in Algorithm 2, the decline of the value of the objective function is bounded from above, from which both the convergence and the bound for the number of iterations required for convergence are established.

After the t-th ($t \geq 0$) iteration, if the algorithm goes on to the $(t+1)$-th iteration, then $\|\alpha_i^{t+1} - \alpha_i^t\|_2 \geq \varepsilon_0$. Let $s' = \arg\max_s |\alpha_{is}^{t+1} - \alpha_{is}^t|$, and it follows that $(\alpha_{is'}^{t+1} - \alpha_{is'}^t)^2 \geq \frac{\varepsilon_0^2}{|\mathcal{R}_i|}$, which is the lower bound for the maximum change in the elements of α_i in t-th iteration. Now, we will consider three cases in the updating formula (37) to get the bound for the change in the objective function given the change in the s'th element of α_i, i.e., $\alpha_{is'}$.

According to the Taylor's Theorem, we obtain

$$\begin{aligned} P(\alpha_{is'}^{t+1}) - P(\alpha_{is'}^t) = & \frac{\alpha_{is'}^t - R_{s'}(1+\lambda_1)}{1+\lambda_1}(\alpha_{is'}^{t+1} - \alpha_{is'}^t) \\ & + \frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^t)^2 \end{aligned} \tag{38}$$

When $0 \leq R_{s'}(1+\lambda_1) \leq \lambda_4$, $\alpha_{is'}^{t+1} = R_{s'}(1+\lambda_1)$, thereby the change in the objective function is

$$
\begin{aligned}
P(\alpha_{is'}^{t+1}) - P(\alpha_{is'}^{t}) &= \frac{\alpha_{is'}^{t} - \alpha_{is'}^{t+1}}{1+\lambda_1}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t}) + \frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2 \\
&= -\frac{(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2}{1+\lambda_1} + \frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2 \\
&= -\frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2 \leq -\frac{\varepsilon_0^2}{2|\mathcal{R}_i|(1+\lambda_1)}
\end{aligned} \tag{39}
$$

When $R_{s'}(1+\lambda_1) > \lambda_4$, $\alpha_{is'}^{t+1} = \lambda_4$. Also, since $0 \leq \alpha_{is'}^{t} \leq \lambda_4$, $\alpha_{is'}^{t} \leq \alpha_{is'}^{t+1}$. The change in the objective function is

$$
\begin{aligned}
P(\alpha_{is'}^{t+1}) - P(\alpha_{is'}^{t}) =& \frac{\alpha_{is'}^{t} - \alpha_{is'}^{t+1} + \alpha_{is'}^{t+1} - R_{s'}(1+\lambda_1)}{1+\lambda_1}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t}) \\
&+ \frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2 \\
=& -\frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2 + \frac{\alpha_{is'}^{t+1} - R_{s'}(1+\lambda_1)}{1+\lambda_1}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t}) \\
\leq& -\frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2 \leq -\frac{\varepsilon_0^2}{2|\mathcal{R}_i|(1+\lambda_1)},
\end{aligned} \tag{40}
$$

since $(\alpha_{is'}^{t+1} - R_{s'}(1+\lambda_1))(\alpha_{is'}^{t+1} - \alpha_{is'}^{t}) \leq 0$.

Similarly, when $R_{s'}(1+\lambda_1) < 0$, $\alpha_{is'}^{t+1} = 0$, and $\alpha_{is'}^{t} \geq \alpha_{is'}^{t+1}$. We still have $(\alpha_{is'}^{t+1} - R_{s'}(1+\lambda_1))(\alpha_{is'}^{t+1} - \alpha_{is'}^{t}) \leq 0$, and it follows that the change in the objective function is

$$
P(\alpha_{is'}^{t+1}) - P(\alpha_{is'}^{t}) \leq -\frac{1}{2(1+\lambda_1)}(\alpha_{is'}^{t+1} - \alpha_{is'}^{t})^2 \leq -\frac{\varepsilon_0^2}{2|\mathcal{R}_i|(1+\lambda_1)}. \tag{41}
$$

Based on (39), (40), and (41), the change in the objective function given the change in $\alpha_{is'}$ is bounded from above by $-\frac{\varepsilon_0^2}{2|\mathcal{R}_i|(1+\lambda_1)}$ after the t-th iteration.

Moreover, let $P_0 = P(\alpha_i^0)$ be the initial value of the objective function, and then, the difference between the initial value and the optimal value of the objective function is $P_0 - P^* < \infty$. Therefore, after at most $\left\lceil \frac{P_0 - P^*}{\frac{\varepsilon_0^2}{2|\mathcal{R}_i|(1+\lambda_1)}} \right\rceil = \left\lceil \frac{2|\mathcal{R}_i|(P_0 - P^*)(1+\lambda_1)}{\varepsilon_0^2} \right\rceil$ iterations, Algorithm 2 converges. □

We run Algorithm 2 for $i = 1 \ldots n$ to obtain the entire S, and the computation of S can be parallelized by computing $\{S_i\}$ separately. Moreover, according to Theorem 2, let ε_0 be the stopping threshold of the coordinate descent method in Algorithm 2, $|\mathcal{R}_i|$ is the number of constraints in S_i, Algorithm 2 converges after at most $\left\lceil \frac{2|\mathcal{R}_i|(P_0 - P_i^*)(1+\lambda_1)}{\varepsilon_0^2} \right\rceil$ iterations, where $P_0 = P(\alpha_i^0) = 0$, P_i^* is minimum value of the objective function for the dual problem (32). Therefore, the time complexity for computing S_i is $\mathcal{O}(|\mathcal{R}_i|^2 \left\lceil \frac{2|\mathcal{R}_i||P_i^*|(1+\lambda_1)}{\varepsilon_0^2} \right\rceil + rn)$. Let $\mathcal{R}_{\max} = \max_{1 \leq i \leq n} \mathcal{R}_i$ be the maximum number of constraints across all the rows of S, $P_{\min}^* = \min_{1 \leq i \leq n} P_i^*$, then the time complexity for completing the entire S sequentially is $\mathcal{O}(n|\mathcal{R}_{\max}|^2 \left\lceil \frac{2|\mathcal{R}_{\max}||P_{\min}^*|(1+\lambda_1)}{\varepsilon_0^2} \right\rceil + rn^2)$.

Table 2 The detailed statistics of the data sets

	DBLP	DBLP-clean	CoRA
Number of node	28,702	2760	15,644
Number of edge	133,664	7636	59,062
Number of node with label	4057	2760	15,644
Number of class	4	4	10
Content dimensionality	13,214	13,214	12,313

7 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.

7.1 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 [10], 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, 4057 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* [10] 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 2760 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 [25], resulting in over 80 labels. Instead of using such a huge label space, we used the hierarchical structure of the labels provided by the data set 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 2.

7.2 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 [27]**:** 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.

NMF [29]**:** Nonnegative Matrix Factorization is similar to PMF in which the link matrix L is used to be completed.
LAD [20]**:** Locally Adaptive Decision function learning uses both content and supervision information to learn a local nonisotropic 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 Eq. (22). *CFSL* is still able to incorporate both link and content information in a globally factorized manner.
SSMetric [14]**:** 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 terms of incorporating different information sources on content, linkages, and supervision.

7.3 Experimental settings

In our experiments, we simulated the real-world scenario on similarity learning as a retrieval problem [22,32]. We start by explaining the experimental settings with an example. As illustrated in Fig. 4, we divide all pair-wise 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 bef given for the training region. Another hyperparameter s controls the number of triplet orderings provided for the training region. In our experiments, s is usually set to the range of 5–20.

Since the ground truth provided in both the *DBLP* and *CoRA* data sets is explicit multiclass labels, we need to convert them into triplet constraints. One way of achieving this is to generate triplet constraints, 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 pair-wise constraints instead of triplets. Although straightforward conversions exist from pair-wise settings to triplet in the most of metric learning-based algorithms, we obey their original implementation by converting triplet constraint to pair-wise 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. Another issue for these two baselines 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

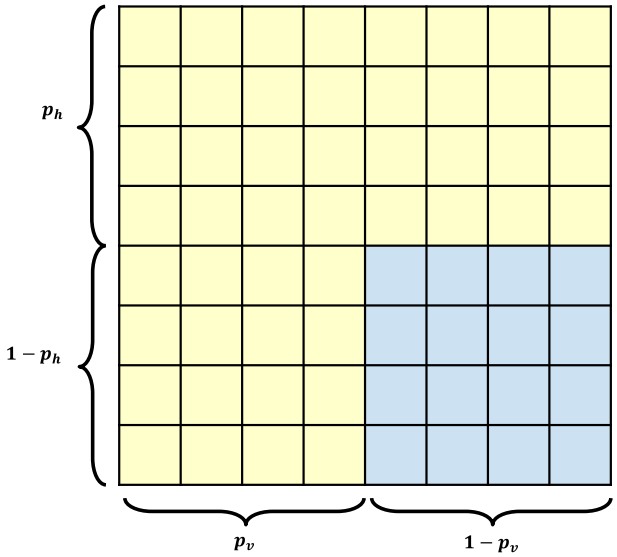


Fig. 4 The experiment settings: *yellow* region indicates the training, while *blue* is the testing entries

information instead of the explicit entry-wise 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 [17] with zero mean and a scale parameter value of one. In addition, the content matrix C and the link matrix L are normalized to remove the scale variations.

7.4 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) [24] 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.

7.5 Results

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

7.5.1 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 Fig. 5. It is evident that the proposed method achieves the best performance across all ranges of the ranks tested. On the other hand, link-based methods as *PMF* and *NMF* 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* and *NMF* 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.

7.5.2 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 Fig. 6. We obtain similar results to the *DBLP* data set, on which the linkage-based method performed poorly. The *PMF* and *NMF* methods obtain 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 terms of the similarity learning across different data sets.

7.5.3 Discussion

Comparing the experimental results we obtained from Figs. 5 and 6, we discover that the precision decreases much slower with k increases for the DBLP data set. Specifically, the precision of our proposed method at 50 for the DBLP data set still remains around 0.85, while the CoRA data set only has 0.7 left. Similar observations are also reflected from other baselines. This is due to the number of classes for the CoRA data set is significantly larger than the one in the DBLP data set. In addition, the labels' granularity is much finer for the CoRA data set, which imposes huge challenges to correctly retrieve other "similar" nodes.

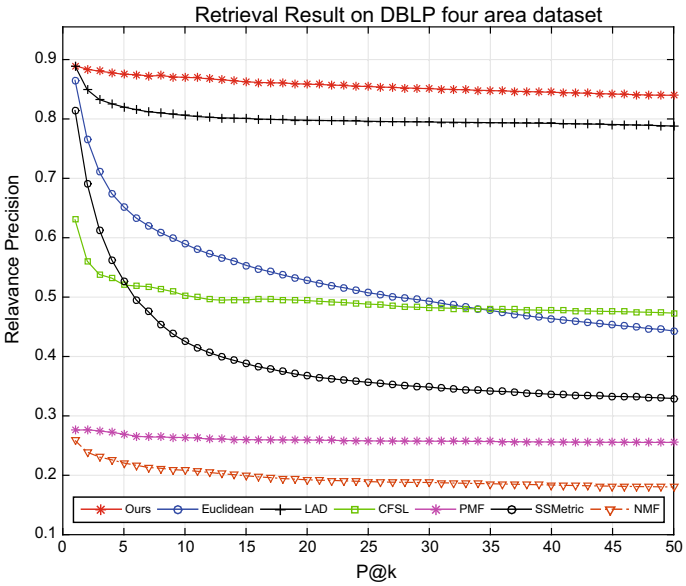


Fig. 5 P@k curve on the *DBLP* data set

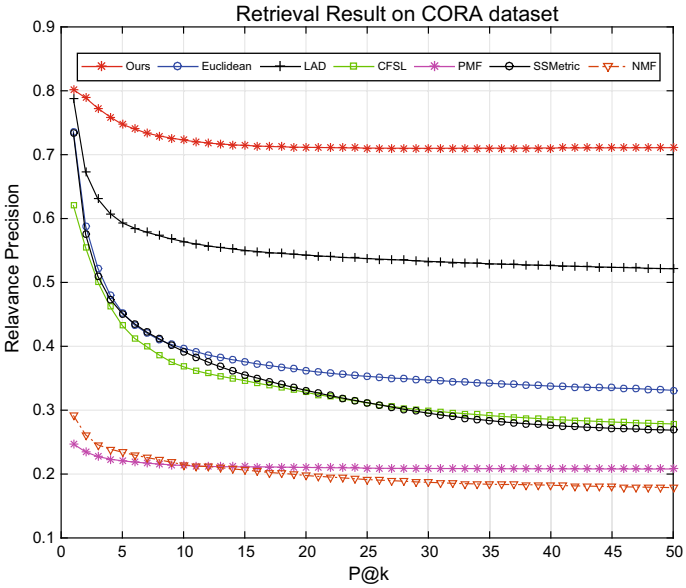


Fig. 6 P@k curve on the *CoRA* data set

Although the absolute precision of our model for the CoRA data set is lower than the one in the DBLP data set, the relative performance between our algorithm and the second best performed method is much better. This implies that our performance drops much slower comparing to other baselines when the retrieval task getting much harder.

7.6 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 Data set*. It is a small data set, obtained by cleaning all the noise from *DBLP*, and contains links, content, and four classes. We use the strategy in Sect. 7.3 to generate supervision information.

7.6.1 Control parameters λ_i

The performance with varying λ_1 is shown in Fig. 7, 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 Fig. 7, 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 Fig. 8, 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 Sect. 1. The robustness in parameter choice reflects how optimality is achieved with the help of underlying topological structure spread with linkage information.

A comparison between Figs. 7 and 8 yields some interesting observations:

- when λ_1 increases, the performance drops slightly;
- when λ_2 increases, the performance improves slightly.

This observation is in agreement with our experimental results in Sect. 7.5. For this particular task assignment, linkage information is not as useful as content similarity.

7.6.2 Supervision s

Figure 9 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 user. Therefore, the algorithm is practical in real-world scenario.

7.6.3 Low-rank approximation: R

Finally, the effect of matrix rank R is shown in Fig. 10. As observed from figure, the performance increases stably after $R \geq 8$. Considering the fact that the samples in the *DBLP* data

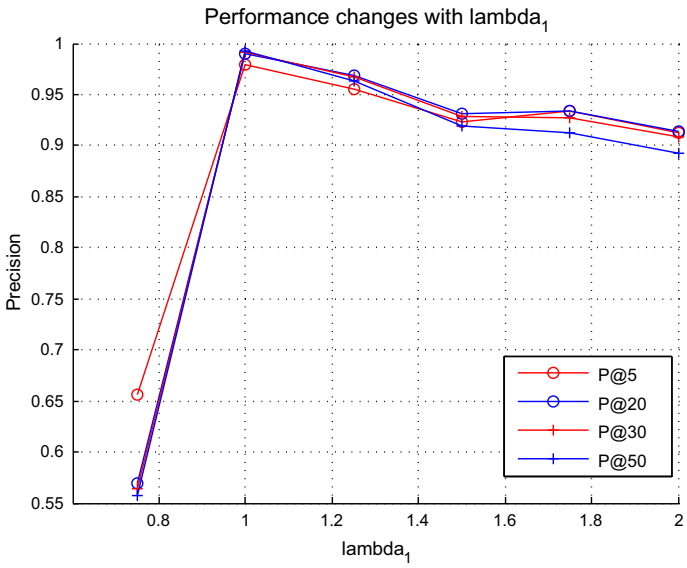


Fig. 7 Parameter testing: λ_1

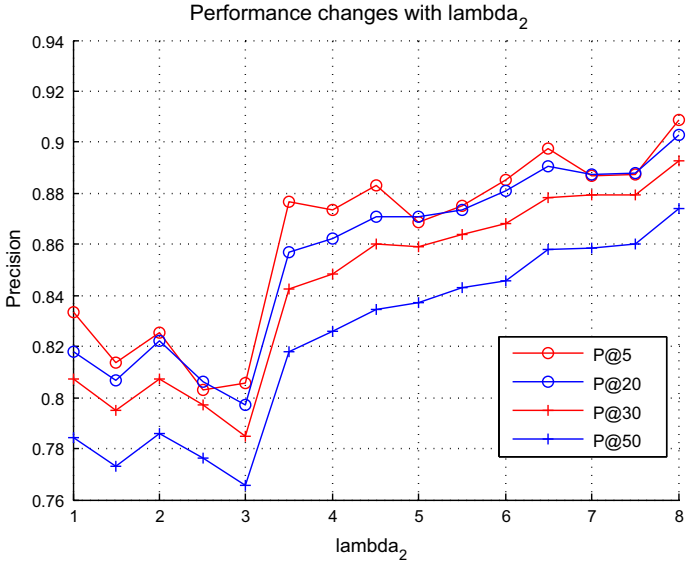


Fig. 8 Parameter testing: λ_2

set are labeled with 4 classes, it is feasible to assume $R > 4$. Typically, the value assignment of rank R is application dependent.

7.7 Noise tolerance

In this section, we present the performance on error tolerance using the large-margin formulation proposed in Eq. (21) on the *DBLP-clean* data set. We tested the *FSL* method with

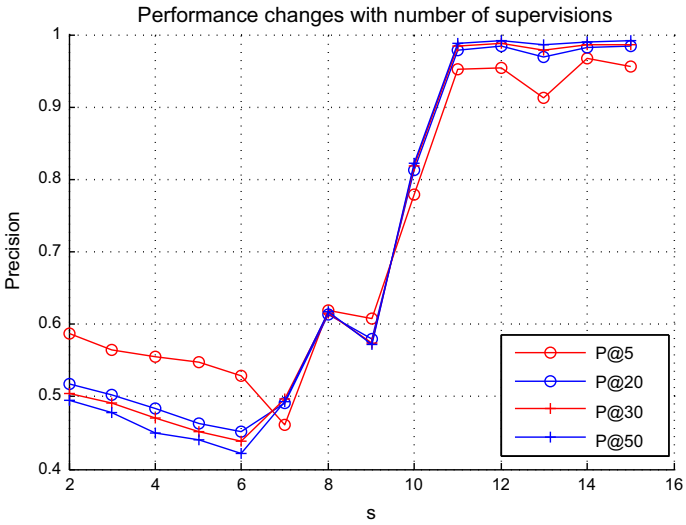


Fig. 9 Parameter testing: number of supervision—*s*

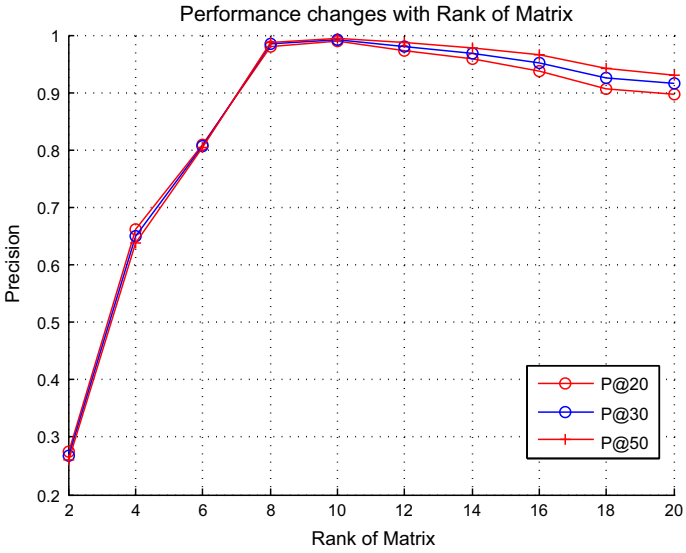


Fig. 10 Parameter testing: number of supervision—*R*

different levels of noise in the supervision in Fig. 11. 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, Fig. 11 demonstrates that our proposed method is robust to a small level of error tolerance.

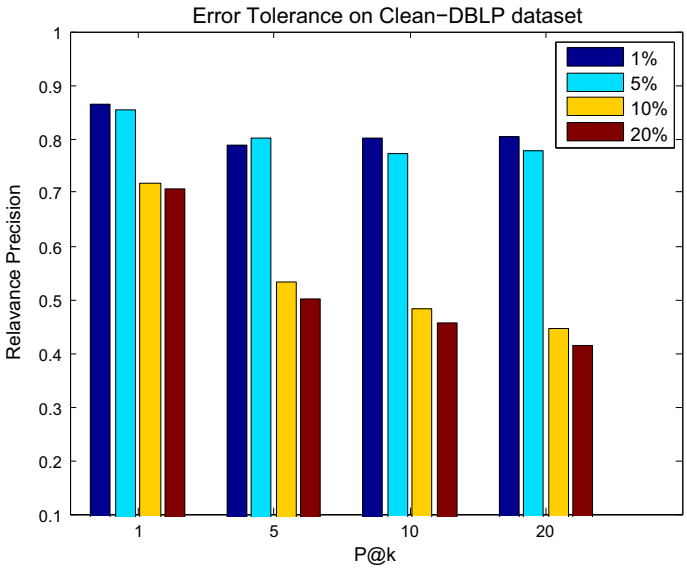


Fig. 11 Error tolerance: *Different color* indicates the percentage of supervision randomly flipped

7.7.1 Efficient solution by the dual

Directly solving the primal problem (23) needs to handle n^2 elements of S by the subgradient method. In contrast, the efficient dual solver only deals with $|\mathcal{R}_i|$ variables for each row of S, with a total of $n|\mathcal{R}_i|$ variables, and it leads to a much more efficient solution. We perform the comparison of computational time between the optimization of (23) in the primal form using subgradient versus the dual form using quadratic programming by coordinate descent. Figure 12 shows the comparison of the computational time using fixed number of users $n = 10^4$, with the number of constraints varies within $\{1, 100, 200, \ldots, 1000\}$. It is observed that the dual method always needs less time than the primal method. In addition, both of them take more time with the increasing number of constraints. With more constraints, more computational cost arises when computing the subgradient for the primal method, and there are more variables in the dual method. Figure 13 illustrates the comparison of the computational time using fixed number of constraints, i.e., $|\mathcal{R}_i| = 100$ for all rows of S, with the number of users varies within $\{10^4, 10^5, 2 \times 10^5, \ldots, 10^6\}$. In this case, the number of variables for the dual method is fixed, and the number of variables for the primal method increases quadratically with the number of users. We can see that the dual method is significantly faster than the primal method. Also, the computational time of the dual method increases with more and more users, since the dual method still needs to compute Q_i, r_i, and S_i for each row of S (please refer to Sect. 6.2).

Note that the rows of S can be computed separately. In both comparisons, the first 300 rows of S are computed, and the Frobenius norm of the difference of S computed using the primal and the dual is always less than 10^{-7}. The maximum number of iterations for the subgradient method in the primal and the coordinate descent in the dual is 200. We perform the comparisons on a Desktop with 16 GB memory and Intel i7-4770 3.4 GHz CPU.

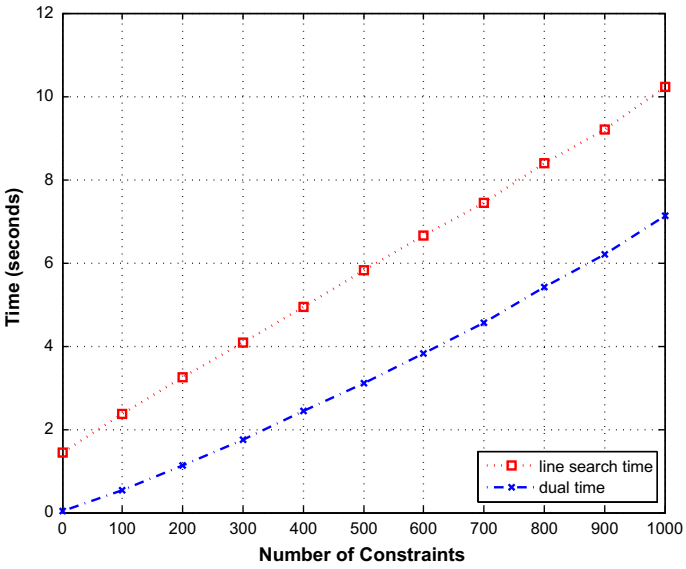


Fig. 12 Comparison of computation time with varying number of constraints and fixed number of users

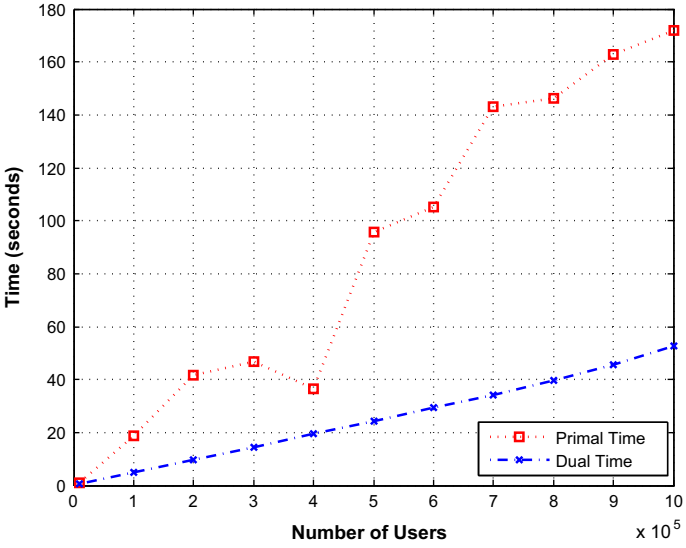


Fig. 13 Comparison of computation time with varying number of users and fixed number of constraints

8 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

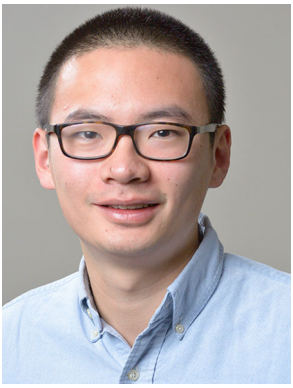
used effectively in conjunction with content and user supervision. By embedding content and links into a unified latent space, the supervision can correctly guide the factorization process. We show extensive experiments on real-world data sets. The proposed *FSL* method significantly outperforms other state-of-the-art approaches in node-based retrieval and is highly robust and efficient.

Acknowledgments The work of Shiyu Chang and Thomas S. Huang was funded in part by the National Science Foundation under Grant Number 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.

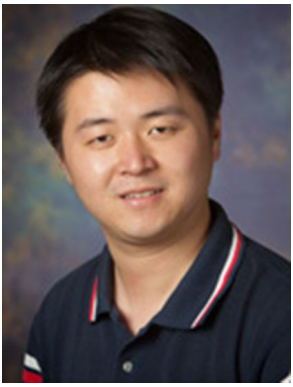
References

1. Aggarwal CC (2003) Towards systematic design of distance functions for data mining applications. In: Proceedings of the ninth ACM SIGKDD, ACM, pp 9–18
2. Bar-Hillel A, Hertz T, Shental N, Weinshall D (2005) Learning a mahalanobis metric from equivalence constraints. J Mach Learn Res 6:937–965
3. Birgin EG, Martínez JM, Raydan M (2000) Nonmonotone spectral projected gradient methods on convex sets. SIAM J Optim 10(4):1196–1211
4. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, New York
5. Ca JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM J Optim 20(4):1956–1982
6. Chang S, Qi G, Aggarwal C, Zhou J, Wang M, Huang T (2014) Factorized similarity learning in networks. In: ICDM, pp 60–69
7. Cheney W, Goldstein AA (1959) Proximity maps for convex sets. Proc Am Math Soc 10(3):448–450
8. Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. In: ICML, pp 209–216
9. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
10. Deng H, Han J, Zhao B, Yu Y, Lin CX (2011) Probabilistic topic models with biased propagation on heterogeneous information networks. In: SIGKDD, pp 1271–1279
11. Geerts F, Mannila H, Terzi E (2004) Relational link-based ranking. In: VLDB, pp 552–563
12. Goldberger J, Roweis S, Hinton H, Salakhutdinov R (2004) Neighbourhood components analysis. In: NIPS, pp 513–520
13. Han SP (1988) A successive projection method. Math Progr 40(1–3):1–14
14. Hoi SCH, Liu W, Chang SF (2008) Semi-supervised distance metric learning for collaborative image retrieval. In: CVPR, IEEE computer society
15. Jeh G, Widom J (2002) Simrank: a measure of structural-context similarity. In: SIGKDD, pp 538–543
16. KorenY Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
17. Kotz S, Kozubowski T, Podgorski K (2001) The laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance. Progress in mathematics series. Birkhäuser, Boston
18. Kumar N, Kummamuru K, Paranjpe D (2005) Semi-supervised clustering with metric learning using relative comparisons. In: Fifth IEEE international conference on data mining, p 4
19. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401(6755):788–791
20. Li Z, Chang S, Liang F, Huang TS, Cao L, Smith JR (2013) Learning locally-adaptive decision functions for person verification. In: CVPR, 2013
21. Lin Z, King I, Lyu M (2006) Pagesim: a novel link-based similarity measure for the world wide web. In: IEEE/WIC/ACM international conference on web intelligence, 2006. WI 2006, pp 687–693
22. Liu X, Ji R, Yao H, Xu P, Sun X, Liu T (2008) Cross-media manifold learning for image retrieval and annotation. In: Lew MS, Bimbo AD, Bakker EM (eds) Multimedia information retrieval. ACM, New York, pp 141–148
23. Ma H, Yang H, Lyu MR, King I (2008) Sorec: social recommendation using probabilistic matrix factorization. In: CKIM, pp 931–940

24. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York
25. McCallum AK, Nigam K, Rennie J, Seymore K (2000) Automating the construction of internet portals with machine learning. Inf Retr 3(2):127–163
26. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: homophily in social networks. Annu Rev Sociol 27:415–444
27. Mnih A, Salakhutdinov R (2007) Probabilistic matrix factorization. In: NIPS, pp 1257–1264
28. Nesterov Y, Nesterov IE (2004) Introductory lectures on convex optimization: a basic course, vol 87. Springer, Berlin
29. Paatero P, Tapper U (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5(2):111–126
30. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. Technical report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120
31. Purushotham S, Liu Y, Kuo CCJ (2012) Collaborative topic regression with social matrix factorization for recommendation systems. In: ICML, 2012
32. Qi GJ, Aggarwal C, Tian Q, Ji H, Huang T (2012) Exploring context and content links in social media: a latent space method. IEEE Trans Pattern Anal Mach Intell 34(5):850–862
33. Qi GJ, Tang J, Zha ZJ, Chua TS, Zhang HJ (2009) An efficient sparse metric learning in high-dimensional space via l1-penalized log-determinant regularization. In: ICML, pp 841–848
34. Qian B, Wang X, Wang F, Li H, Ye J, Davidson I (2013) Active learning from relative queries. In: Proceedings of the twenty-third international joint conference on artificial intelligence. AAAI Press, pp 1614–1620
35. Qian B, Wang X, Wang J, Li H, Cao N, Zhi W, Davidson I (2013) Fast pairwise query selection for large-scale active learning to rank. In: IEEE 13th international conference on data mining (ICDM), 2013, pp 607–616
36. Shalev-Shwartz S, Singer Y, Srebro N (2007) Pegasos: primal estimated sub-gradient solver for svm. In: ICML, pp 807–814
37. Tang J, Yan S, Hong R, Qi GJ, Chua TS (2009) Inferring semantic concepts from community-contributed images and noisy tags. In: SIGMM. ACM, pp 223–232
38. Tseng P (2001) Convergence of a block coordinate descent method for nondifferentiable minimization. J Optim Theory Appl 109(3):475–494
39. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
40. Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. In: SIGKDD, pp 448–456
41. Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10:207–244
42. Wen Z, Yin W, Zhang Y (2012) Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Math Progr Comput 4(4):333–361
43. Xi W, Fox EA, Fan W, Zhang B, Chen Z, Yan J, Zhuang D (2005) Simfusion: measuring similarity using unified relationship matrix. In: SIGIR, pp 130–137
44. Xing EP, Ng AY, Jordan MY, Russell S (2003) Distance metric learning, with application to clustering with side-information. In: NIPS, pp 505–512
45. Zeng C, Jiang Y, Zheng L, Li J, Li L, Li L, Shen C, Zhou W, Li T, Duan B, Lei M, Wang P (2013) Fiu-miner: a fast, integrated, and user-friendly system for data mining in distributed environment. In: SIGKDD, pp 1506–1509
46. Zhao P, Han J, Sun Y (2009) P-rank: a comprehensive structural similarity measure over information networks. In: CIKM, pp 553–562
47. Zhou J, Lu Z, Sun J, Yuan L, Wang F, Ye J (2013) Feafiner: biomarker identification from medical data through feature generalization and selection. In: SIGKDD, pp 1034–1042



Shiyu Chang is a Ph.D. student at University of Illinois at Urbana-Champaign, supervised by Prof. Thomas S. Huang. He received his B.S. and M.S. degrees from the University of Illinois at Urbana-Champaign in 2011 and 2014, respectively. He has a wide range of research interests in data exploratory and analytics. His current directions lie on building high-performance and reliable systems with the help of large-scale multimodality information to solve complex computational tasks in real world. He is the recipient of the best student paper award in IEEE ICDM 2014.



Guo-Jun Qi received the PhD degree from the University of Illinois at Urbana-Champaign, in December 2013. His research interests include pattern recognition, machine learning, computer vision, multimedia, and data mining. He received twice IBM PhD fellowships, and Microsoft fellowship. He is the recipient of the Best Paper Award at the 15th ACM International Conference on Multimedia, Augsburg, Germany, 2007. He is currently a faculty member with the Department of Electrical Engineering and Computer Science at the University of Central Florida and has served as program committee member and reviewer for many academic conferences and journals in the fields of pattern recognition, machine learning, data mining, computer vision, and multimedia.

Yingzhen Yang received the B.Eng. degree and the M.Eng. degree in College of Computer Science and Technology from Zhejiang University of China, Beijing, China, in 2006 and 2008, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. His current research interests include machine learning with focus on statistical learning theory, large-scale probabilistic graphical models, sparse coding, manifold learning and nonparametric methods, and computer vision with focus on image classification using deep learning methods and image/video enhancement.

Charu C. Aggarwal received the BS degree from IIT Kanpur in 1993 and the PhD degree from Massachusetts Institute of Technology in 1996. He is a research scientist at the IBM T.J. Watson Research Center in Yorktown Heights, New York. He has since worked in the field of performance analysis, databases, and data mining. He has published more than 155 papers in refereed conferences and journals and has been granted over 50 patents. He has served on the program committees of most major database/data mining conferences and served as program vice-chairs of the SIAM Conference on Data Mining, 2007, the IEEE ICDM Conference, 2007, the WWW Conference 2009, and the IEEE ICDM Conference, 2009. He served as an associate editor of the IEEE Transactions on Knowledge and Data Engineering Journal from 2004 to 2008. He is an associate editor of the ACM TKDD Journal, an action editor of the Data Mining and Knowledge Discovery Journal, an associate editor of the ACM SIGKDD Explorations, and an associate editor of the Knowledge and Information Systems Journal. He is the recipient of the IEEE ICDM Research Contributions Award (2015) and a Fellow of the ACM, SIAM, and the IEEE.

Jiayu Zhou is an assistant professor at Department of Computer Science and Engineering, Michigan State University. Before joining MSU, Jiayu was a staff research scientist at Samsung Research America, leading the industrial research on recommender systems and deep learning algorithms. Jiayu received his Ph.D. degree in computer science at Arizona State University in 2014. Jiayu has a broad research interest in large-scale machine learning and data mining, and biomedical informatics. He has served as Senior Program Committee of IJCAI 2015. He also served as program committee members in premier conferences such as NIPS, ICDM, SDM, WSDM, ACML, and PAKDD. Jiayu currently serves as an Associate Editor of Neurocomputing. Most of Jiayu's research has been published in top machine learning and data mining venues including NIPS, SIGKDD, ICDM, and SDM. One of his papers has been selected for the best student paper award in ICDM 2014.

Meng Wang is a professor at the Hefei University of Technology, China. He received his B.E. and Ph.D. degrees in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, respectively. His current research interests include multimedia content analysis, search, mining, recommendation, and large-scale computing. He received the best paper awards successively from the 17th and 18th ACM International Conference on Multimedia, the best paper award from the 16th International Multimedia Modeling Conference, the best paper award from the 4th International Conference on Internet Multimedia Computing and Service, and the best demo award from the 20th ACM International Conference on Multimedia.

Thomas S. Huang received the ScD from MIT in 1963. He is a William L. Everitt Distinguished professor in the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. Since 2001, he has also been the member of National Academy of Engineering. Before he joined the University of Illinois, he was a professor at Purdue University from 1973 to 1980, and an assistant and then an associate professor at MIT from 1963 to 1973, both in the Department of Electrical Engineering. His professional interests are computer vision, image processing, pattern recognition, and multimodal signal processing. He is a life fellow of the IEEE.