2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Network Completion via Joint Node Clustering and Similarity Learning

Dimitrios Rafailidis* and Fabio Crestani†
*Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
†Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, Switzerland
draf@csd.auth.gr, fabio.crestani@usi.ch

***Abstract*—In this study, we investigate the problem of network completion by considering the similarities between the node attributes. Given a sample of observed nodes with their incident edges, how can we efficiently reconstruct the network by completing the missing edges of unobserved nodes? Apart from the missing edges, in real settings the node attributes may be partially missing, as well as they may introduce noise when completing the network. We propose a network completion method based on joint clustering and similarity learning. The proposed approach differs from competitive strategies, which consider attribute-based similarities at the node-level. First we generate clusters based on the node attributes, thus reducing the noise and the sparsity in the case that the attributes may be missing. We design a joint objective function to jointly factorize the adjacency matrix of the observed edges with the cluster-based similarities of the node attributes, while at the same time the clusters are adapted, accordingly. In addition, we propose an optimization algorithm to solve the network completion problem via alternating minimization. Our experiments on two real world social networks from Facebook and Google+ show that the proposed approach achieves high completion accuracy, compared to other state-of-the-art methods.**

***Index Terms*—Network completion, joint matrix factorization, clustering nodes with attributes.**

I. Introduction

With the advent of social networks, such as Twitter, Facebook, Myspace and so on, researchers have focused on how to collect reliable large scale network data for analyzing the user behavior [1]. The main problem is that often the collected network data is incomplete with missing nodes and edges [2]. Given that only a part of the network can be observed, the question is how can we complete the unobserved part of the network. This issue is also known as the *network completion problem* [1], [3], where the goal is to reconstruct the entire underlying network topology given partially observed nodes with their edges. The network completion problem has several real world applications; for example, given a subset of observed nodes/users how can we infer the relationships of unobserved ones, so as to improve advertising techniques in social networks, accounting for the fact that social friends have similar taste [4].

Given a sample of observed nodes, several network completion methods have been introduced, which infer the missing relationships/edges, like in the studies reported in [1], [3], [5]. However, these network completion strategies ignore the fact that nodes may also have a set of attributes; for example in a real world social network users interact with each other, thus forming relationships, while at the same time the profile of a user may be available, including its set of attributes, such as age, gender, hometown, occupation, and so on.

Provided that users with similar personal attributes tend to form relationships, the node attributes play an important role when completing the network [6]. Recently, a few approaches have been introduced to complete networks with attributes based on different strategies [4], [7]. For example, in the method presented in [4] a weighted sum is computed by aggregating the edges' weights of observed nodes and the node attributes of both the observed and unobserved nodes. The method in [7], first factorizes only the sampled network of the observed nodes and then exploits the similarities of the node attributes to complete the network. Nonetheless, in real settings, the node attributes may contain noise and/or be partially missing as well [8]. Therefore, a pressing challenge resides on how to complete a network with missing edges and node attributes, by also taking into account that the similarities between the node attributes may degrade the completion accuracy, as they may contain noise.

In this paper, we propose a network completion method, by performing Joint node Clustering and Similarity Learning, namely JCSL. Instead of calculating pairwise attribute-based similarities between the nodes, in our approach we compute the similarities based on the node attributes at a cluster-level, thus handling the issue that the node attributes may be partially missing, as well as reducing the noise that attributes may introduce. To achieve this, we formulate a *joint objective function* to co-factorize the adjacency matrix of the observed nodes with the cluster-based similarities, while adapting the clusters, accordingly. Furthermore, we present an efficient *optimization algorithm based on alternating minimization* to complete the network. In our experiments with two real world social networks, we show that the proposed approach significantly outperforms competitive network completion strategies.

II. Related Work

Network Completion. Network completion methods observe a partial sample of the network, and based on that sample, infer what the rest of the network looks like, trying to complete the missing nodes or edges. Hanneke and Xing

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Francisco, CA, USA
978-1-5090-2846-7/16/$31.00 © 2016 IEEE

[5] present a learning method from a set of observed nodes and compute confidence bounds on the number of differences between the learned and the true topologies, as well as they introduce a variant of the network completion problem based on a stochastic block model. Kim and Leskovec [1] combine the Expectation Maximization algorithm with Kronecker graphs model and design a Gibbs sampling approach to estimate the model parameters as well as the inference about the missing nodes and edges of the network. The study in [3] embeds the nodes, which should be clustered, in a graph structure representing the affinity between each pair of nodes based on the structure of the network. However, these methods do not consider the node attributes, when completing the network. Sina et al. [4] extend the network completion method of [3], where they introduce two algorithms to combine the node attributes with the graph structure in different ways. In the first algorithm, a weighted sum is calculated between the network structures of the observed nodes and the node attributes; while in their second algorithm, the node attributes are aggregated into a unified network and then a weighted sum is used within the unified network to compute the affinities. Masrour et al. [7] generate a subnetwork based on the observed nodes with their edges, and then perform matrix completion/factorization on the adjacency matrix of the subnetwork. The factorized matrix is combined with the similarity information of the node attributes to transduct the knowledge to the unobserved nodes, thus completing the entire network. However, all the aforementioned network completion strategies that consider the information of the node attributes ignore the fact that node attributes may be noisy or partially missing.

Clustering Nodes with Attributes. Representative approaches of clustering nodes with attributes are roughly divided into two main categories: *distance-based* [9], [10] and *model-based* [11], [12]. For example, [10] is a *distance-based* clustering method for nodes with attributes, which computes a unified distance measure to capture both the structural and attribute similarities of the nodes. Meanwhile, *model-based* methods enforce the intra-cluster similarity by modelling the attribute values and the edges of nodes in a cluster by various distributions. For instance, the study in [13] uses Bayesian inference, considering that the nodes in the same cluster should follow a common multinomial distribution for each of their attributes and a common Bernoulli distribution for their connections. In addition, methods for clustering graphs with attributes identify clusters either in the full space of the network [11], [10], [13] or in multiple sub-spaces [6], [14]. However, the above methods focus on partitioning the network into clusters and not on the network completion problem.

Joint/Collective Matrix Factorization. In several applications with multiple relations, represented as multiple matrices, the matrix completion/factorization accuracy may be improved by exploiting information from one relation, while predicting the other. Such factorization techniques are applicable on condition that the multiple matrices share at least one dimension, for example, the user or node dimension. The co-factorization of multiple matrices is called *joint or collective matrix factorization*, having a variety of real world applications, such as multi-view clustering [15], link prediction [16], and recommendation with side information e.g., user profiles or item attributes [17], [18]. Nonetheless, these joint matrix factorization techniques do not consider the network completion problem with node attributes.

III. Problem Formulation

In our problem, we assume that the network has a set of $\mathcal{N}$ nodes, which is composed of subsets of observed $\mathcal{N}_o$ and unobserved $\mathcal{N}_u$ nodes, with $\mathcal{N}_o \cup \mathcal{N}_u \equiv \mathcal{N}$. Let $\mathbf{A} \in \Re^{n \times n}$ be the *incomplete adjacency matrix*, where $n = |\mathcal{N}|$ is the number of nodes and $[\mathbf{A}]_{i,j}$ is set to 1, if the edge between two observed nodes i and j does exist, with $i, j \in \mathcal{N}_o$. For the unobserved nodes $[\mathbf{A}]_{i,j}$ is set to 0, if $i, j \in \mathcal{N}_u$. This means that we only consider the observed nodes with their edges, while the rest of edges that are incident on all the unobserved nodes in $\mathcal{N}_u$ have to be completed. In addition, we assume that a node (both observed and unobserved) might have x attributes, that are either categorical e.g., gender, country and so on, or numerical e.g., age. In our setting, except for the missing edges of unobserved nodes, a subset of attributes might be missing as well, reflecting on the real world scenario.

First, for each pair of nodes we compute x similarities based on the x attributes. In the case of a categorical attribute, we set the similarity equal to 1, if two nodes have the same attribute and 0 otherwise; while for a numerical attribute we normalize the similarities in $[0, 1]$. Then, we aggregate the x similarities into a *unified attribute-based similarity matrix* $\mathbf{S} \in \Re^{n \times n}$.

Following the notation of matrix factorization techniques [19], we assume that the *complete adjacency matrix* $\widetilde{\mathbf{A}} \in \Re^{n \times n}$ can be derived by factorizing the incomplete adjacency matrix $\mathbf{A}$, trying to minimize the following d *low rank approximation error* $||\mathbf{A} - \widetilde{\mathbf{A}}||_F^2$, where d is the number of latent factors and $||\cdot||_F$ denotes the Frobenius norm. However, in the network completion problem we have an entire column or row of $\mathbf{A}$ filled with zeros for the unobserved nodes in $\mathcal{N}_u$, as we do not consider any edge for these nodes. This problem can significantly degrade the completion accuracy of matrix factorization techniques [17]. Our problem is formally defined as follows:

Definition (Problem)**.** *"The goal of the proposed approach is to compute $\widetilde{\mathbf{A}}$ with the missing edges of the unobserved nodes, by co-factorizing the incomplete adjacency matrix $\mathbf{A}$ with the side information of the attribute-based similarity matrix $\mathbf{S}$."*

IV. Proposed Approach

A. Joint Objective Function

Given the *attribute-based similarity matrix* $\mathbf{S}$ (Section III), first we have to define the objective function for performing node clustering, that is, to calculate the cluster assignment matrix $\mathbf{C} \in \Re^{k \times n}$, where k is the number of clusters. According to the Laplacian method of [20], the clustering problem is equivalent to the following minimization problem:

$$\min_{\mathbf{C}} Tr\left[\mathbf{C}\mathbf{L}\mathbf{C}^T\right] \tag{1}$$

where Tr is the trace operator. Matrix $\mathbf{L} \in \Re^{n \times n}$ is the Laplacian of $\mathbf{S}$, which is computed as follows: $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D} \in \Re^{n \times n}$ is a diagonal matrix (degree matrix), whose entries are calculated as $\mathbf{D}_{ii} = \sum_{ij} \mathbf{S}_{ij}$. The term $Tr\big[\mathbf{C}\mathbf{L}\mathbf{C}^T\big]$ is the matrix form of $\sum_{ij} \mathbf{S}_{ij} ||\mathbf{c}_i - \mathbf{c}_j||^2$, which expresses if two nodes are similar based on $\mathbf{S}$, then they should be assigned to the same clusters $\mathbf{c}_i$ and $\mathbf{c}_j$.

In our approach, we consider the factorization of matrices $\mathbf{A}$ and $\mathbf{C}$, resulting in the complete adjacency matrix $\widetilde{\mathbf{A}}$ and the cluster-level similarity matrix $\widetilde{\mathbf{C}}$ of the node attributes, respectively. Given d latent factors, in total we have to minimize the following two individual d *low-rank approximation errors*:

$$||\mathbf{A} - \mathbf{U}_A \mathbf{V}_A||_F^2 \tag{2}$$

$$||\mathbf{C} - \mathbf{U}_C \mathbf{V}_C||_F^2 \tag{3}$$

where $\mathbf{U}_A \in \Re^{n \times d}$ and $\mathbf{U}_C \in \Re^{k \times d}$ are the basis matrices; $\mathbf{V}_A \in \Re^{d \times n}$ and $\mathbf{V}_C \in \Re^{d \times n}$ are the coefficient matrices based on matrices $\mathbf{A}$ and $\mathbf{C}$ respectively [19]; with $\tilde{\mathbf{A}} = \mathbf{U}_A \mathbf{V}_A$ and $\tilde{\mathbf{C}} = \mathbf{U}_C \mathbf{V}_C$. In order to perform joint matrix factorization, we force the coefficient matrices $\mathbf{V}_A$ and $\mathbf{V}_C$ to a *consensus matrix* $\mathbf{V}^* \in \Re^{d \times n}$ [15], which results in the following minimization problem:

$$\min_{\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{V}^*} ||\mathbf{A} - \mathbf{U}_A \mathbf{V}_A||_F^2 + ||\mathbf{C} - \mathbf{U}_C \mathbf{V}_C||_F^2 + \lambda(||\mathbf{V}_A - \mathbf{V}^*||_F^2 + ||\mathbf{V}_C - \mathbf{V}^*||_F^2) \tag{4}$$

with $\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{V}^* \geq 0$. The third term of (4) denotes the *disagreement measurements* between $\mathbf{V}_A - \mathbf{V}^*$ and $\mathbf{V}_C - \mathbf{V}^*$; where $\mathbf{V}_A$, $\mathbf{V}_C$ and $\mathbf{V}^*$ are comparable, as they are $(d \times n)$ matrices. Parameter $\lambda > 0$ controls the impact of the *disagreement measurements* on (4), which can be interpreted as follows, high λ values influence more the approximation errors, thus forcing more the coefficient matrices $\mathbf{V}_A$ and $\mathbf{V}_C$ to the consensus matrix $\mathbf{V}^*$.

To jointly compute the clustering matrix $\mathbf{C}$ of (1) when minimizing (4), we define the following *joint objective function*:

$$\min_{\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{V}^*, \mathbf{C}} \mathcal{F} = ||\mathbf{A} - \mathbf{U}_A \mathbf{V}_A||_F^2 + ||\mathbf{C} - \mathbf{U}_C \mathbf{V}_C||_F^2 + \lambda(||\mathbf{V}_A - \mathbf{V}^*||_F^2 + ||\mathbf{V}_C - \mathbf{V}^*||_F^2) + \alpha Tr\big[\mathbf{C}\mathbf{L}\mathbf{C}^T\big] \tag{5}$$

where parameter $\alpha > 0$ controls the influence of the clustering error on the joint objective function in (5).

B. Alternating Minimization

As the joint objective function $\mathcal{F}(\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{V}^*, \mathbf{C})$ in (5) is not convex with respect to the six variables/matrices, we propose an alternating minimization algorithm, where we update one variable, while keeping the remaining five fixed. The cluster assignment matrix $\mathbf{C}$ is initialized by performing power iteration [21], using the attribute-based similarities in $\mathbf{S}$; while $\mathbf{U}_A$ $\mathbf{V}_A$, $\mathbf{U}_C$ and $\mathbf{V}_C$ are initialized by random matrices, with $\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C \geq 0$. The consensus matrix $\mathbf{V}^*$ is initialized by averaging the two coefficient matrices $\mathbf{V}_A$ and $\mathbf{V}_C$. Next, we present the updating steps for each matrix/variable.

Step 1, fix $\mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{V}^*, \mathbf{C}$ and update $\mathbf{U}_A$. By considering the optimality condition $\partial F / \partial \mathbf{U}_A = 0$, we have to calculate the partial derivative of $\mathcal{F}$ with respect to $\mathbf{U}_A$. The partial derivative of $\mathcal{F}$ with respect to $\mathbf{U}_A$ is equal to:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{U}_A} = -2\mathbf{A}{\mathbf{V}_A}^T + 2\mathbf{U}_A \mathbf{V}_A {\mathbf{V}_A}^T \tag{6}$$

By setting the partial derivative equal to zero and solving with respect to $\mathbf{U}_A$, we have the following *updating rule* for $\mathbf{U}_A$:

$$\mathbf{U}_A = \mathbf{A}{\mathbf{V}_A}^T (\mathbf{V}_A {\mathbf{V}_A}^T)^{-1} \tag{7}$$

where matrix $(\mathbf{V}_A {\mathbf{V}_A}^T)$ is positive definite, thus invertible. To speed up the computation in each step, we use the Cholesky decomposition to approximate the inverse of a matrix [22].

Step 2, fix $\mathbf{U}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{V}^*, \mathbf{C}$ and update $\mathbf{V}_A$. Similarly, the partial derivative of $\mathcal{F}$ with respect to $\mathbf{V}_A$ equals:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{V}_A} = -2{\mathbf{U}_A}^T \mathbf{A} + 2{\mathbf{U}_A}^T \mathbf{U}_A \mathbf{V}_A + 2\lambda \mathbf{V}_A - 2\lambda \mathbf{V}^* \tag{8}$$

Provided that $({\mathbf{U}_A}^T \mathbf{U}_A + \lambda \mathbf{I})$ is positive definite, we set $\partial \mathcal{F} / \partial \mathbf{V}_A$ equal to zero, resulting in the following *updating rule* for $\mathbf{V}_A$:

$$\mathbf{V}_A = ({\mathbf{U}_A}^T \mathbf{U}_A + \lambda \mathbf{I})^{-1} ({\mathbf{U}_A}^T \mathbf{A} + \lambda \mathbf{V}^*) \tag{9}$$

Step 3, fix $\mathbf{U}_A, \mathbf{V}_A, \mathbf{V}_C, \mathbf{V}^*, \mathbf{C}$ and update $\mathbf{U}_C$. Given the partial derivative of $\mathcal{F}$ with respect to $\mathbf{U}_C$

$$\frac{\partial \mathcal{F}}{\partial \mathbf{U}_C} = -2\mathbf{C}{\mathbf{V}_C}^T + 2\mathbf{U}_C \mathbf{V}_C {\mathbf{V}_C}^T \tag{10}$$

we have the following *update rule* for $\mathbf{U}_C$

$$\mathbf{U}_C = \mathbf{C}{\mathbf{V}_C}^T (\mathbf{V}_C {\mathbf{V}_C}^T)^{-1} \tag{11}$$

where $(\mathbf{V}_C {\mathbf{V}_C}^T)$ is positive definite.

Step 4, fix $\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}^*, \mathbf{C}$ and update $\mathbf{V}_C$. The partial derivative of $\mathcal{F}$ with respect to$\mathbf{V}_C$ is equal to:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{V}_C} = -2{\mathbf{U}_C}^T \mathbf{C} + 2{\mathbf{U}_C}^T \mathbf{U}_C \mathbf{V}_C + 2\lambda \mathbf{V}_C - 2\lambda \mathbf{V}^* \tag{12}$$

Given that $({\mathbf{U}_C}^T \mathbf{U}_C + \lambda \mathbf{I})$ is a positive definite matrix, the *updating rule* for $\mathbf{V}_C$ is:

$$\mathbf{V}_C = ({\mathbf{U}_C}^T \mathbf{U}_C + \lambda \mathbf{I})^{-1} ({\mathbf{U}_C}^T \mathbf{C} + \lambda \mathbf{V}^*) \tag{13}$$

Step 5, fix $\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{C}$ and update $\mathbf{V}^*$. The partial derivative of $\mathcal{F}$ with respect to $\mathbf{V}^*$ is calculated as follows:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{V}^*} = -2\lambda \mathbf{V}_A - 2\lambda \mathbf{V}_C + 4\lambda \mathbf{V}^* \tag{14}$$

Then, the *updating rule* for the consensus matrix $\mathbf{V}^*$ is:

$$\mathbf{V}^* = \frac{1}{2}(\mathbf{V}_A + \mathbf{V}_C) \tag{15}$$

Step 6, fix $\mathbf{U}_A, \mathbf{V}_A, \mathbf{U}_C, \mathbf{V}_C, \mathbf{V}^*$ and update $\mathbf{C}$. By taking the partial derivative of $\mathcal{F}$ with respect to $\mathbf{C}$, we have

$$\frac{\partial \mathcal{F}}{\partial \mathbf{C}} = 2\mathbf{C} - 2\mathbf{U}_C \mathbf{V}_C + 2\alpha \mathbf{L}\mathbf{C} \tag{16}$$

By setting the partial derivative equal to zero, the *updating rule* for the cluster assignment matrix $\mathbf{C}$ is computed as follows:

$$\mathbf{C} = (\mathbf{I} + \alpha\mathbf{L})^{-1}\mathbf{U}_C\mathbf{V}_C \tag{17}$$

with $(\mathbf{I} + \alpha\mathbf{L})$ being a positive definite matrix. In this step, the clustering assignment matrix is recalculated based on the values that matrices $\mathbf{U}_C$ and $\mathbf{V}_C$ have acquired at steps 3 and 4, thus *adapting the clusters over the iterations*, accordingly.

Finally, after the alternating minimization has finished, the complete adjacency matrix is computed by setting $\tilde{\mathbf{A}} = \mathbf{U}_A\mathbf{V}_A$.

V. Experiments

A. Evaluation Setup

Datasets. In our experiments, we use two publicly available social networks from the Stanford Network Analysis Project[1] (SNAP), the social networks of ***Facebook*** and ***Gplus***. The ***Facebook*** dataset consists of 10 ego-networks, having in total 4,039 users, 88,234 social relationships, and 26 attributes of user profiles, such as hometowns, birthdays, colleagues, political affiliations, and so on. The ***Gplus*** dataset, crawled from Google+, consists of 133 ego-networks, with 106,674 users, 13,673,453 edges, and 6 attributes, that is, gender, last name, job titles, institutions, universities and places lived. In both datasets, each attribute is represented as a feature vector in $\{0, 1\}$, and the set of attributes is represented by concatenating the respective feature vectors. More details for the evaluation networks and how the feature vectors have been generated can be found at [23].

Evaluation Protocol. To evaluate the network completion accuracy, we randomly select a percentage of edges $< i, j >$ as a test set $\mathcal{M}$, while the remaining edges form the incomplete adjacency matrix to train the network completion method. We evaluate the performance in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which are formally defined as follows [1], [7]:

$$\text{MAE} = \frac{\sum_{<i,j>\in\mathcal{M}} |\mathbf{A}_{i,j} - \widetilde{\mathbf{A}}_{i,j}|}{|\mathcal{M}|}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{<i,j>\in\mathcal{M}} \left(\mathbf{A}_{i,j} - \widetilde{\mathbf{A}}_{i,j}\right)^2}{|\mathcal{M}|}}$$

We repeated our experiments ten times and we averaged our results over the ten trials.

Compared Methods.

- ***NMF*** [19]: is a baseline Non-negative Matrix Factorization method, which does not consider the side information of the node attributes.
- ***CMF*** [17]: is a state-of-the-art Collective Matrix Factorization technique, which uses a learning strategy to complete the network by co-factorizing the incomplete adjacency matrix with the attribute-based similarity matrix.
- ***MCDT*** [7]: is a Matrix Completion with Decoupled Transduction strategy, which completes the adjacency sub-matrix of the observed nodes and then utilizes the attribute-based similarities to complete the missing edges of the unobserved nodes.
- ***JSL***: is a variant of the proposed approach, which performs Joint Similarity Learning, without computing the cluster-based similarities. JSL uses directly the attribute-based similarity matrix $\mathbf{S}$, when performing joint factorization. Given the approximation error $||\mathbf{S} - \mathbf{U}_S\mathbf{V}_S||_F^2$ to factorize the attribute-based similarity matrix $\mathbf{S}$, JSL tries to minimize the following objective function:

$$\begin{aligned}\mathcal{F} &= ||\mathbf{A} - \mathbf{U}_A\mathbf{V}_A||_F^2 + ||\mathbf{S} - \mathbf{U}_S\mathbf{V}_S||_F^2 \\ &+ \lambda(||\mathbf{V}_A - \mathbf{V}^*||_F^2 + ||\mathbf{V}_S - \mathbf{V}^*||_F^2)\end{aligned} \tag{18}$$

  The optimization algorithm is similar to the alternating minimization of Section IV-B.
- ***JCSL***: is the proposed network completion method, which performs Joint node Clustering and Similarity Learning.

B. Impact of Observed Nodes

To evaluate the impact of the missing edges on the performance of the examined methods, we randomly select a percentage of observed nodes, and consider their edges as training set; while the remaining edges of unobserved nodes are considered as test set, that is, the missing edges that have to be completed [7]. In the first set of experiments, we assume that the node attributes information is available for all nodes, both for the observed and unobserved ones. Table I shows the effect on MAE and RMSE by varying the percentage of observed nodes in 25, 50 and 75% for Facebook and Gplus, respectively. We observe that when the percentage of observed nodes increases, all methods achieve lower completion error metrics, in terms of MAE and RMSE. The baseline NMF method has poor performance, compared to the rest approaches. This happens because NMF ignores the side information of node attributes, thus having rows and columns with only zeros when factorizing the incomplete adjacency matrix; which as a consequence results in a high completion error. Meanwhile, CMF and JSL achieve comparable performance, by following different joint/collective factorization strategies, when co-factorizing the incomplete adjacency matrix with the attribute-based similarity matrix. The competitive MCDT method achieves a relatively low completion error, by following a different strategy, where it first factorizes the submatrix of the observed network and then exploits the similarity information of the node attributes to complete the network. For all the different percentages of observed nodes, the proposed JSCL method is superior over the competitive strategies. To verify this, we ran the paired t-test and found that the differences are statistically significant for $p < 0.05$. Compared with the state-of-the art methods, JCLS exploits the side information of the node similarities at a cluster-level, by co-factorizing the incomplete

[1] http://snap.stanford.edu/data/

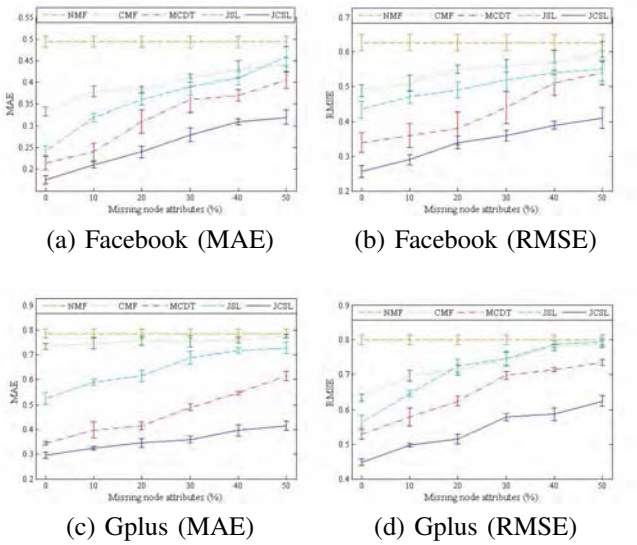
adjacency matrix with the attribute-based generated clusters, thus achieving a low completion error in both datasets.

TABLE I
EFFECT ON MAE AND RMSE BY VARYING THE PERCENTAGE OF OBSERVED NODES. ALL NODE ATTRIBUTES ARE USED. BOLD VALUES DENOTE THE BEST SCORES, FOR $^*p < 0.05$.

Facebook		method				
metric	observed nodes (%)	NMF	CMF	MCDT	JSL	JCSL
MAE	25	0.622	0.372	0.333	0.358	**0.274***
	50	0.494	0.334	0.215	0.242	**0.176***
	75	0.465	0.231	0.149	0.188	**0.104***
RMSE	25	0.681	0.576	0.524	0.550	**0.477***
	50	0.627	0.488	0.340	0.434	**0.256***
	75	0.536	0.394	0.237	0.291	**0.164***
Gplus		method				
metric	observed nodes (%)	NMF	CMF	MCDT	JSL	JCSL
MAE	25	0.874	0.807	0.484	0.653	**0.448***
	50	0.786	0.734	0.344	0.527	**0.296***
	75	0.615	0.522	0.216	0.345	**0.176***
RMSE	25	0.963	0.846	0.742	0.781	**0.677***
	50	0.801	0.634	0.529	0.566	**0.448***
	75	0.641	0.466	0.347	0.37	**0.239***

C. Impact of Missing Node Attributes

In the next set of experiments, we evaluate the performance of the examined network completion methods by varying the percentage of the missing node attributes. We randomly sampled and removed a subset of attributes from both the observed and unobserved nodes. In this set of experiments, we fix the percentage of observed nodes to 50% and we vary the the percentage of the missing node attributes from 0 to 50%, by a step of 10%. Figure 2 shows the effect on MAE and RMSE for the evaluation datasets. The baseline NMF method is not affected when varying the missing node attributes, as it does not exploit the auxiliary similarity information. We observe that the performance of all methods significantly degrades, having higher completion errors when the percentage of missing node attributes increases. For example, in the case of 50% missing node attributes, CMF and JSL have similarly low performance as the baseline NMF method. This occurs because both CMF and JSL co-factorize the incomplete adjacency matrix with the attributed-based similarity matrix, which is extremely sparse in the case of 50% missing node attributes, and as a consequence the co-factorization with the incomplete adjacency matrix cannot perform well in the completion task. Similarly, MCDT is negatively affected when increasing the percentage of missing node attributes, as in its second step it exploits directly the attribute-based similarity matrix to complete the missing edges. The proposed JCSL method keeps the completion error relatively low, even in the case of a large percentage of missing node attributes. This happens because JCSL co-factorizes the adjacency matrix of the incomplete edges with the attribute-based clusters, thus becoming less sensitive when more node attributes are missing, compared to the other approaches.



(a) Facebook (MAE) (b) Facebook (RMSE)

(c) Gplus (MAE) (d) Gplus (RMSE)

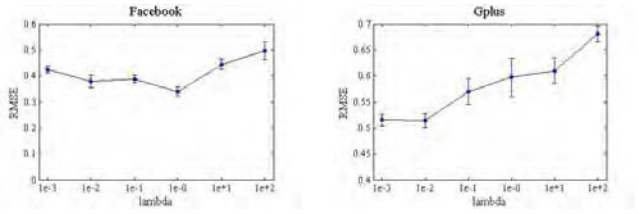
Fig. 1. Effect on MAE and RMSE by varying the percentage of missing node attributes, using a 50% of observed nodes.

D. Parameter Analysis

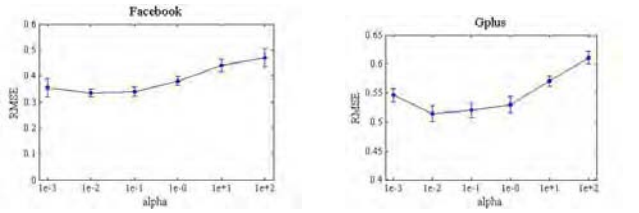
In Figure 2, we evaluate the influence of the parameters λ and α of the joint objective function in (5), by fixing the number of latent factors d to 10, and the percentages of observed nodes and missing node attributes to 50% and 20%, respectively. Figures 2(a)-(b) present the experimental results by varying the parameter λ for Facebook and Gplus, respectively. We observe that the lowest completion error is achieved when λ is equal to 1e-0 and 1e-2 for Facebook and Gplus, accordingly, which means a more conservative selection of λ is required for the larger dataset of Gplus. In both evaluation datasets, the completion error is significantly increased for larger values of λ. This can be explained as follows, parameter λ controls the impact of the approximation errors between the coefficient matrices $\mathbf{V}_A$ and $\mathbf{V}_C$, and the consensus matrix $\mathbf{V}^*$ when minimizing (5) via the proposed alternating minimization algorithm. This means that high λ values influence more the respective approximation errors, thus forcing more the coefficient matrices $\mathbf{V}_A$ and $\mathbf{V}_C$ to the consensus matrix $\mathbf{V}^*$. Similarly, Figures 2(c)-(d) show the influence of α for the Facebook and Gplus datasets, respectively, where α controls the impact of the clustering reconstruction error on (5). The values of α that achieve the lowest RMSE are 1-e01 and 1-e02 for Facebook and Gplus, respectively, with larger values of α denoting more influence of the clustering error on the joint objective function.

E. Convergence Analysis

Let $\mathcal{F}^{(t)}$ and $\mathcal{F}^{(t+1)}$ denote the value of $\mathcal{F}$ after t and $t+1$ iterations are finished in the alternating optimization algorithm. At the end of each $t+1$ iteration, we compute the normalized difference between iterations t and $t+1$ as follows: $|\mathcal{F}^{(t)} - \mathcal{F}^{(t+1)}|/\mathcal{F}^{(t)}$. Figure 3 shows the normalized differences of the joint objective function in (5) for 200 iterations over the ten trials. We observe that the algorithm converges after 40 and



(a) Facebook-λ (b) Gplus-λ



(c) Facebook-α (d) Gplus-α

Fig. 2. Analysis of parameters λ and α of the joint objective function in (5), using 50% of observed nodes and 20% of missing node attributes.

160 iterations in the Facebook and Gplus datasets, respectively. In the Facebook dataset, the algorithm converges close to a 1e-4 normalized difference between two consecutive iterations t and $t+1$, while in the larger dataset of Gplus the lower bound is approximately at 1e-3 normalized difference of the joint objective function.

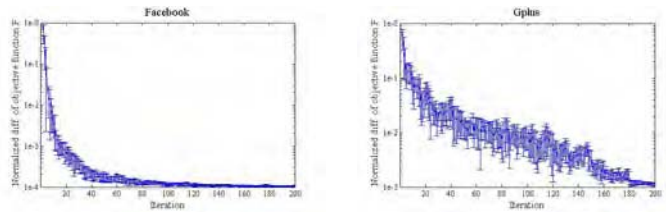


Fig. 3. Normalized differences of the objective function $\mathcal{F}$ in (5) versus the number of iterations for the ten trials.

VI. Conclusion

In this paper, we presented an efficient network completion strategy based on joint node clustering and similarity learning, namely JCSL. Different from related studies, the proposed approach first computes the attribute-based similarities at the node-level, based on which the clustering assignment matrix is initialized. Then, the proposed JCSL method formulates a joint objective function to adapt the clustering solution, while at the same time performs joint matrix factorization of the incomplete adjacency matrix with the cluster-based similarities. Finally, we presented an optimization algorithm via alternating minimization with convergence guarantees to complete the network. Our experiments on two benchmark datasets confirmed the superiority of the proposed JCSL method over other state-of-the-art methods, by preserving a low completion error in all settings, achieving a relative improvement of 20% approximately, compared to the second best method. An interesting direction of future work is to extend the proposed approach to perform clustering in partially co-aligned social networks, that is, to generate clusters in multiple networks that have a subset of common (anchor) nodes [24].

References

[1] M. Kim and J. Leskovec, "The network completion problem: Inferring missing nodes and edges in networks," in *SDM*, 2011, pp. 47–58.

[2] G. Kossinets, "Effects of missing data in social networks," *Social Networks*, vol. 28, no. 3, pp. 247–268, 2006.

[3] R. Eyal, S. Kraus, and A. Rosenfeld, "Identifying missing node information in social networks," in *AAAI*, 2011.

[4] S. Sina, A. Rosenfeld, and S. Kraus, "Solving the missing node problem using structure and attribute information," in *ASONAM*, 2013, pp. 744–751.

[5] S. Hanneke and E. P. Xing, "Network completion and survey sampling," in *AISTATS*, 2009, pp. 209–215.

[6] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, and E. Müller, "Focused clustering and outlier detection in large attributed graphs," in *KDD*, 2014, pp. 1346–1355.

[7] F. Masrour, I. Barjasteh, R. Forsati, A. Esfahanian, and H. Radha, "Network completion with node similarity: A matrix completion approach with provable guarantees," in *ASONAM*, 2015, pp. 302–307.

[8] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *NIPS*, 2011, pp. 1413–1421.

[9] H. Cheng, Y. Zhou, and J. X. Yu, "Clustering large attributed graphs: A balance between structural and attribute similarities," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 2, pp. 12:1–12:33, 2011.

[10] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *VLDB*, vol. 2, no. 1, pp. 718–729, 2009.

[11] Xu, Zhiqiang and Ke, Yiping and Wang, Yi and Cheng, Hong and Cheng, James, "GBAGC: A general bayesian framework for attributed graph clustering," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 1, pp. 5:1–5:43, 2014.

[12] J. Yang, J. J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *ICDM*, 2013, pp. 1151–1156.

[13] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "A model-based approach to attributed graph clustering," in *SIGMOD*, 2012, pp. 505–516.

[14] S. Günnemann, I. Färber, S. Raubach, and T. Seidl, "Spectral subspace clustering for graphs with feature vectors," in *ICDM*, 2013, pp. 231–240.

[15] J. Gao, J. Han, J. Liu, and C. Wang, "Multi-view clustering via joint nonnegative matrix factorization," in *SDM*, 2013, pp. 252–260.

[16] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *TKDD*, vol. 5, no. 2, p. 10, 2011.

[17] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *KDD*, 2008, pp. 650–658.

[18] D. Rafailidis and A. Nanopoulos, "Repeat consumption recommendation based on users preference dynamics and side information," in *WWW - Companion Volume*, 2015, pp. 99–100.

[19] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000, pp. 556–562.

[20] S. Gao, I. W. Tsang, and L. Chia, "Laplacian sparse coding, hypergraph laplacian sparse coding, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, 2013.

[21] F. Lin and W. W. Cohen, "Power iteration clustering," in *ICML*, 2010, pp. 655–662.

[22] G. H. Golub and C. F. V. Loan, *Matrix computations (3. ed.)*. Johns Hopkins University Press, 1996.

[23] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *NIPS*, 2012, pp. 548–556.

[24] J. Zhang and P. S. Yu, "PCT: partial co-alignment of social networks," in *WWW*, 2016, pp. 749–759.