Predicting Directed Links using Nondiagonal Matrix Decompositions

Jérôme Kunegis

University of Koblenz–Landau, Germany
kunegis@uni-koblenz.de

Jörg Fliege

The University of Southampton, UK
j.fliege@soton.ac.uk

Abstract—We present a method for trust prediction based on nondiagonal decompositions of the asymmetric adjacency matrix of a directed network. The method we propose is based on a nondiagonal decomposition into directed components (DEDICOM), which we use to learn the coefficients of a matrix polynomial of the network’s adjacency matrix. We show that our method can be used to compute better low-rank approximations to a polynomial of a network’s adjacency matrix than using the singular value decomposition, and that a higher precision can be achieved at the task of predicting directed links than by undirected or bipartite methods.

Keywords—trust; decomposition into directed components

I. INTRODUCTION

In online social networks, a trust relationship is a unidirectional connection between two persons that denotes the trust of one person in another. Trust relationships are found for instance on dedicated sites such as the product rating site Epinions [1]. In trust networks, a commonly studied problem is the problem of trust prediction. From a machine learning perspective, the trust prediction problem is a link prediction problem on a social graph with directed edges. The classical link prediction problem, as studied for instance in [2] applies to undirected networks. Instead, what needs to be addressed in social recommender systems is the *directed* link prediction problem. To do that, we will use spectral graph theory and consider decompositions of that matrix.

While network mining for undirected networks is a well-developed area, the same cannot be said of directed networks. For instance, the directed link prediction problem is itself rarely addressed, let alone using algebraic methods. Thus, our contribution in this paper is a structural trust prediction algorithm, based on a little-known family of matrix decompositions, the decomposition into directed components (DEDICOM), which we combine with a novel technique to learn a trust prediction function, and show that the resulting trust predictions outperform other structural trust prediction methods. Our evaluation is performed on five trust network datasets.

II. BACKGROUND

A directed network consists of vertices connected by edges. We denote a directed network as $G = (V, E)$ where V is the set of vertices and E is the set of directed edges.

An edge connecting the vertices $i, j \in V$ will be denoted (i, j) . A trust network is then a directed network in which nodes are persons and edges represent a trust relationship.

A major method for solving the link prediction problem is by representing a network as a matrix, which is then manipulated to compute link prediction values. Algebraic graph theory uses (among others) the adjacency matrix $\mathbf{A}$ to study a graph, defined as $\mathbf{A}_{ij} = 1$ when (i, j) is an edge and $\mathbf{A}_{ij} = 0$ otherwise. In the general case of undirected networks, the adjacency matrix $\mathbf{A}$ is symmetric, and we can consider its eigenvalue decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, in which $\mathbf{U}$ is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix. The reason one often considers the eigenvalue decomposition of the adjacency matrix is that allows to compute a power of the adjacency matrix as $\mathbf{A}^k = \mathbf{U}\mathbf{\Lambda}^k\mathbf{U}^T$, which can be used to define trust prediction methods such as triangle closing, rank reduction, the matrix exponential and the Neumann kernel. This useful property is not found in other decomposition such as nonnegative matrix factorization and probabilistic latent semantic analysis.

A. Symmetric Case

If, in a given directed network with adjacency matrix $\mathbf{A}$, edge directions are not important, we may map the network to an undirected network. The resulting graph is undirected and its adjacency matrix equals $\mathbf{A} + \mathbf{A}^T$. This new adjacency matrix is normal and thus has the eigenvalue decomposition $\mathbf{A} + \mathbf{A}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, in which the matrix $\mathbf{\Lambda}$ is diagonal and $\mathbf{U}$ is orthogonal. Many link prediction functions can be constructed from the eigenvalue decomposition using so-called spectral transformations [3]. Given the eigenvalue decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, a spectral transformation of $\mathbf{A}$ is a function f that can be written as $f(\mathbf{A}) = \mathbf{U}f(\mathbf{\Lambda})\mathbf{U}^T$, such as the matrix exponential $e^{\mathbf{A}}$. In particular, any power sum $p(\mathbf{A})$ can be written as

$$p(\mathbf{A}) = p(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T) = \mathbf{U}p(\mathbf{\Lambda})\mathbf{U}^T.$$

This can be exploited to learn a link prediction function p using curve fitting [3]. Taking the set of edges in a network, we split it into two disjoint sets. The first set represents known edges, and the second set represents edges that we want to predict. Let both edge sets be represented by the adjacency matrices $\mathbf{A}$ and $\mathbf{B}$. We now want to find a power

sum p mapping $\mathbf{A}$ to $\mathbf{B}$. Using the eigenvalue decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, we can write $p(\mathbf{A}) = p(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)$. Since for every k we can write $\mathbf{A}^k = (\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)^k = \mathbf{U}\mathbf{\Lambda}^k\mathbf{U}^T$, it follows that $p(\mathbf{A}) = \mathbf{U}p(\mathbf{\Lambda})\mathbf{U}^T$ and finally

$$p(\mathbf{\Lambda}) = \mathbf{U}^T\mathbf{B}\mathbf{U}.$$

Since $p(\mathbf{\Lambda})$ is necessarily a diagonal matrix we can ignore the nondiagonal entries of $\mathbf{U}^T\mathbf{B}\mathbf{U}$. Therefore, a good power sum p is one that maps the eigenvalues $\mathbf{\Lambda}_{kk}$ to the diagonal elements $(\mathbf{U}^T\mathbf{B}\mathbf{U})_{kk}$. Finding such a p is a one-dimensional curve fitting problem whose size equals the rank of the eigenvalue decomposition, and can thus be solved efficiently.

B. Bipartite Case

The bipartite double cover of a directed graph is an undirected bipartite graph that has twice as many nodes and the same number of edges as the directed graph.

Formally, the directed graph $G = (V, E)$ has the bipartite double cover $H = (V \times \{1\} \cup V \times \{2\}, F)$ such that

$$F = \{((i, 1), (j, 2)) \mid (i, j) \in E\}.$$

If $\mathbf{A}$ is the adjacency matrix of G , then

$$\text{bip}(\mathbf{A}) = \begin{bmatrix} \mathbf{A} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{A} \end{bmatrix}$$

is the adjacency matrix of H . We can interpret $\mathbf{A}$ as the biadjacency matrix of a bipartite graph whose adjacency matrix is $\text{bip}(\mathbf{A})$. The eigenvalue decomposition of $\text{bip}(\mathbf{A})$ is then equivalent to the singular value decomposition of $\mathbf{A}$. Given the singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the eigenvalue decomposition of $\text{bip}(\mathbf{A})$ is given by

$$\text{bip}(\mathbf{A}) = \begin{bmatrix} \bar{\mathbf{U}} & \bar{\mathbf{U}} \\ \bar{\mathbf{V}} & -\bar{\mathbf{V}} \end{bmatrix} \begin{bmatrix} +\mathbf{\Sigma} & \\ & -\mathbf{\Sigma} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{U}} & \bar{\mathbf{U}} \\ \bar{\mathbf{V}} & -\bar{\mathbf{V}} \end{bmatrix}^T,$$

with $\bar{\mathbf{U}} = \mathbf{U}/\sqrt{2}$ and $\bar{\mathbf{V}} = \mathbf{V}/\sqrt{2}$. In this decomposition, each singular value σ corresponds to the eigenvalue pair $\{\pm\sigma\}$. Odd powers of $\text{bip}(\mathbf{A})$ then have the form

$$\text{bip}(\mathbf{A})^{2k+1} = \begin{bmatrix} & (\mathbf{A}\mathbf{A}^T)^k\mathbf{A} \\ (\mathbf{A}^T\mathbf{A})^k\mathbf{A}^T & \end{bmatrix},$$

where the alternating power $(\mathbf{A}\mathbf{A}^T)^k\mathbf{A}$ can be explained by the fact that in the bipartite double cover, a path will follow edges from one vertex set to the other in alternating directions. The same learning method as for the eigenvalue decomposition can be used with the singular value decomposition of the asymmetric matrix $\mathbf{A}$. However, it will not find power sums but instead alternating power sums [4]. For instance taking the cube of the singular values in the decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ leads to the following alternating

power of $\mathbf{A}$: $\mathbf{U}\mathbf{\Sigma}^3\mathbf{V}^T = \mathbf{A}\mathbf{A}^T\mathbf{A}$, which can be interpreted as the act of following an edge in the forward direction, then following an edge in the backward direction, and the following an edge in the forward direction again. In this way, every odd power series applied to the singular values gives the corresponding alternating power of the adjacency matrix. Thus, instead of using the matrix exponential, one may instead use its odd component, the hyperbolic sine [4].

III. RELATED ASYMMETRIC MATRIX DECOMPOSITIONS

To compute link prediction functions of the asymmetric adjacency matrix $\mathbf{A}$, the methods described in the previous section cannot be applied because they either ignore the direction of edges or reduce the network to its bipartite double cover. Instead, this section reviews simple but ultimately ineffective methods for achieving this: Direct computations of link prediction functions and nonorthogonal eigenvalue decompositions, including personalized PageRank. An effective method for computing link prediction functions of the asymmetric adjacency matrix is given in the next section.

A. Direct Computation of the Matrix Exponential

Instead of using matrix decompositions, we may think of computing a link prediction function directly. For instance, since the matrix exponential is a valid link prediction function, we may try to compute it directly. An overview of methods for achieving this is given in [5]. In short, the known methods are not scalable to large, sparse matrices, since the exponential will be dense. Therefore, any scalable computation of the matrix exponential must result in a low-rank or otherwise compressed form. According to the state of the art in the cited paper, this can only be achieved by actually computing a matrix decomposition.

B. Nonorthogonal Eigenvalue Decomposition

The adjacency matrix $\mathbf{A}$ of an undirected graph is not symmetric in the general case, and therefore the normal eigenvalue decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ is not guaranteed to exist. However, as long as $\mathbf{A}$ is diagonalizable, we can write $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$, in which $\mathbf{U}$ is a matrix whose columns have unit norm (but are not necessarily orthogonal) and $\mathbf{\Lambda}$ contains complex eigenvalues.

This model works well as long as the asymmetric matrix $\mathbf{A}$ is diagonalizable. In practice, this is not the case. For instance, if a network is acyclic, then all the eigenvalues of $\mathbf{A}$ are zero, and the eigenvalue decomposition is not defined. Even though networks such as scientific citation networks are not acyclic, because pairs of papers may cite each other. However, the networks are almost acyclic, and accordingly their eigenvalues are small and their eigenvalue decomposition cannot be used. Another problem with the nonorthogonal eigenvalue decomposition are complex eigenvalues. Because they are complex, it is hard to find a spectral transformation mapping them to new, real eigenvalues, since

all usual link prediction functions map complex values to complex values. Thus, while the nonorthogonal eigenvalue decomposition can be used in theory to compute spectral transformations of directed matrices, this works badly in practice.

C. Personalized PageRank

A personalized variant of PageRank can be defined that gives, for each pair of nodes in a directed network, a score which can be used for link prediction [6]. Using the diagonal degree matrix $\mathbf{D}$ defined as $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, we can define the right-stochastic transition probability matrix as $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ to give the trust prediction function

$$(\mathbf{I} - \beta\mathbf{P})^{-1}\mathbf{P} = \mathbf{P} + \beta\mathbf{P} + \beta^2\mathbf{P}^2 + \dots$$

with $0 < \beta < 1$. To compute this power sum, we now have the same problem as with the computation of power sums of $\mathbf{A}$ itself. Since the matrix $\mathbf{P}$ is not symmetric, its eigenvalue decompositions is complex in the general case, and is not suited for the computation of matrix powers.

IV. PROPOSED METHOD

We now present our new method for predicting directed links in trust networks. Our method is based on DEDICOM, a decomposition type of asymmetric matrices, and adds to it a novel way of learning a transformation of it that can be used for link prediction. DEDICOM (Decomposition into Directed Components [7]) refers to a class of matrix decompositions of the form $\mathbf{A} = \mathbf{U}\mathbf{X}\mathbf{U}^T$ in which the matrix $\mathbf{U}$ is orthogonal and the matrix $\mathbf{X}$ is not necessarily diagonal. The idea behind DEDICOM is to map all asymmetry of $\mathbf{A}$ into the central matrix $\mathbf{X}$, and leave the outer factors $\mathbf{U}$ and $\mathbf{U}^T$ symmetric in the decomposition. This has the advantage that powers of $\mathbf{A}$ can be computed easily. Note also that if a decomposition $\mathbf{A} = \mathbf{Y}\mathbf{X}\mathbf{Y}^T$ is given in which $\mathbf{Y}$ is not orthogonal, it can be transformed to orthogonal form by using the QR decomposition $\mathbf{Y} = \mathbf{Q}\mathbf{R}$, giving $\mathbf{A} = \mathbf{Q}(\mathbf{R}\mathbf{X}\mathbf{R}^T)\mathbf{Q}^T$, which is in the desired form. DEDICOM was originally developed in 1979 to study directed relations in social groups [7]. Individual DEDICOM algorithms are described in [8], [9], [10].

A. Computation of DEDICOM

Unlike eigenvalue problems, there is no single simple algorithm for DEDICOM that returns a globally optimal result. Instead, several algorithms are used in the literature, each giving different results that are local optima to the underlying optimization problem.

A useful property of the singular value decomposition is that a truncation of it gives the best possible rank- r approximation to the original matrix for all r . Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the asymmetric adjacency matrix of a directed graph,

and $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ its singular value decomposition. The following problem seeks a rank- r approximation of $\mathbf{A}$:

$$\min_{\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times r}} \|\mathbf{A} - \mathbf{X}\mathbf{Y}^T\|_F$$

An optimal solution to this problem is given by

$$\mathbf{X} = \mathbf{U}_{1:r} \mathbf{\Sigma}_{1:r} \mathbf{U}_{1:r}^T, \quad \mathbf{Y} = \mathbf{V}_{1:r} \mathbf{\Sigma}_{1:r} \mathbf{V}_{1:r}^T.$$

The ordering of the singular values in $\mathbf{\Sigma}$ must be chosen such that the largest singular values are kept, to give the best rank- r approximation. On the other hand, decompositions into directed components are not as well-behaved. In fact, a full-rank DEDICOM can be trivially written as

$$\mathbf{A} = \mathbf{U}\mathbf{X}\mathbf{U}^T, \quad \mathbf{X} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T,$$

in which $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the singular value decomposition of $\mathbf{A}$, i.e., a decomposition such that $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Sigma}$ is a diagonal matrix. This construction suggests that a DEDICOM can be computed based on the singular value decomposition. However, a truncation of this decomposition is not the best rank- r approximation to $\mathbf{A}$, because in the general case we have the inequality

$$\mathbf{\Sigma}_{1:r} \mathbf{U}_{1:r} \mathbf{V}_{1:r}^T \mathbf{U}_{1:r} \neq (\mathbf{\Sigma}\mathbf{V}^T\mathbf{U})_{1:r} \mathbf{U}_{1:r}.$$

This is true because we can split the right into the sum

$$\begin{aligned} (\mathbf{\Sigma}\mathbf{V}^T\mathbf{U})_{1:r} \mathbf{U}_{1:r} &= \mathbf{\Sigma}_{1:r} \mathbf{U}_{1:r} (\mathbf{V}^T\mathbf{U})_{1:r} \\ &= \mathbf{\Sigma}_{1:r} \mathbf{U}_{1:r} (\mathbf{V}^T\mathbf{U})_{1:r} + \mathbf{\Sigma}_{1:r} (r+1:n) (\mathbf{V}^T\mathbf{U})_{(r+1):n} \mathbf{U}_{(r+1):n} \mathbf{U}_{1:r}, \end{aligned}$$

whose right side is not zero in the general case. In fact, since the central matrix $\mathbf{X}$ is not diagonal, it is not even evident which of the latent dimensions should be kept. For these reasons, decompositions into directed components must be computed for individual values of r separately.

Left and Right Closed-form Solutions: Approximate solutions to DEDICOM can be obtained from the rank- r singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ [7]:

$$\begin{aligned} \mathbf{A} &= \mathbf{U}(\mathbf{\Sigma}\mathbf{V}^T\mathbf{U})\mathbf{U}^T \\ \mathbf{A} &= \mathbf{V}(\mathbf{V}^T\mathbf{U}\mathbf{\Sigma})\mathbf{V}^T \end{aligned}$$

We will call these the left and right closed-form DEDICOMs (LEFT and RIGHT).

Symmetric Closed-form Solution: A closed-form solution is given by first computing the rank- r singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and then computing the rank- r eigenvalue decomposition of the following symmetric matrix [7]:

$$\mathbf{U}\mathbf{\Sigma}\mathbf{U}^T + \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

The matrix $\mathbf{Q}$ is then used to construct a decomposition into directed components:

$$\mathbf{A} = \mathbf{Q}\mathbf{X}\mathbf{Q}^T, \quad \mathbf{X} = \mathbf{Q}^T\mathbf{A}\mathbf{Q}$$

We will call this the symmetric closed-form solution (CLO), because it is based on the symmetric eigenvalue decomposition.

Iterative Solution: Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and a rank $1 \leq r \leq n$, an optimal decomposition into directed components is given by solutions to the following minimization problem.

$$\min_{\mathbf{U} \in \mathbb{R}^{n \times r}, \mathbf{X} \in \mathbb{R}^{r \times r}} \|\mathbf{A} - \mathbf{UXU}^T\|_F$$

This problem can be solved in various ways using iterative algorithms [7], [8], [9]. These algorithms are in alternating least-squares form, i.e. they choose a $\mathbf{U}$ and a $\mathbf{X}$ and then alternatively find a new $\mathbf{U}$ and a new $\mathbf{X}$ that minimizes the Frobenius norm.

In our method, we used the algorithm described in [8], as given in Algorithm 1. In this algorithm, $\text{svd}(\mathbf{A})$ returns the rank- r singular value decomposition of $\mathbf{A}$, and $\text{qr}(\mathbf{U})$ orthogonalizes the columns of $\mathbf{U}$. The term $2a\|\mathbf{D}\|_F \mathbf{U}$ can be omitted to give faster convergence, although the resulting iteration is not guaranteed to converge monotonically anymore. In practice, we found that omitting the term did not make convergence faster. Therefore, all subsequent tests are computed with the term. The precision parameter ε determines the precision of the computed decomposition. We used $\varepsilon = 10^{-7}$ in our experiments.

Input: matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, rank $1 \leq r < n$
Output: matrices $\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{X} \in \mathbb{R}^{r \times r}$
 $\mathbf{U}, \mathbf{D}, \mathbf{V} \leftarrow \text{svd}(\mathbf{A}, r)$
 $a \leftarrow \mathbf{D}_{11}$
repeat
 $\mathbf{U} \leftarrow \mathbf{A}\mathbf{U}\mathbf{U}^T \mathbf{A}^T \mathbf{U} + \mathbf{A}^T \mathbf{U}\mathbf{U}^T \mathbf{A} \mathbf{U} + 2a\|\mathbf{D}\|_F \mathbf{U}$
 $\mathbf{U} \leftarrow \text{qr}(\mathbf{U})$
 $\mathbf{D}' \leftarrow \mathbf{D}$
 $\mathbf{D} \leftarrow \mathbf{U}^T \mathbf{A} \mathbf{U}$
until $\|\mathbf{D} - \mathbf{D}'\|_F \leq \varepsilon/n$

Algorithm 1: The iterative algorithm (ITER) to compute the decomposition into directed components.

B. Nondiagonal Spectral Transformations

In the case of the eigenvalue and singular value decompositions, we have seen that spectral transformations can be learned by curve fitting. In the case of the nonorthogonal eigenvalue decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$ we can use the same technique to learn a power series. In this case however, eigenvalues are complex and special care must be taken.

A given decomposition into directed components $\mathbf{A} = \mathbf{UXU}^T$ can be used to compute a power series p of $\mathbf{A}$ as $p(\mathbf{A}) = \mathbf{U}p(\mathbf{X})\mathbf{U}^T$. Here, $\mathbf{X}$ is a full $r \times r$ matrix. For the values of r for which a DEDICOM can reasonably be computed (< 100), any power series such as the exponential of $\mathbf{X}$ can be easily computed. In order to learn an optimal function $p(\mathbf{A})$, we must thus solve $p(\mathbf{X}) = \mathbf{U}^T \mathbf{B} \mathbf{U}$. Since $\mathbf{X}$ is not diagonal, the problem does not reduce to a one-dimensional curve fitting problem. As a solution, we propose the following method.

Our method can be applied to learn a polynomial p , and does not generalize to other power series such as the

Table II
THE NETWORKS USED IN OUR EVALUATION. THE RANK r IS USED IN ALL MATRIX DECOMPOSITIONS OF EACH NETWORK. D: INCLUDES DISTRUST EDGES. T: INCLUDES EDGE ARRIVAL TIMES.

	Network	Vertices	Edges	r	Type
[11]	Advogato	6,535	51,397	25	Trust
[12]	DBLP	12,591	49,793	14	Citation
[13]	Email	265,214	420,045	10	Communication
[1]	Epinions	131,828	841,372	11	Trust D T
[14]	Slashdot	79,120	515,581	11	Trust D
[15]	Twitter	465,017	835,423	10	Trust
[16]	Wikipedia	8,297	107,071	19	Trust D T

matrix exponential. Let the polynomial p have degree d with coefficients a_i :

$$p(\mathbf{X}) = a_0 \mathbf{I} + a_1 \mathbf{X} + a_2 \mathbf{X}^2 + \dots + a_d \mathbf{X}^d$$

Thus we have

$$a_0 \mathbf{I} + a_1 \mathbf{X} + a_2 \mathbf{X}^2 + \dots + a_d \mathbf{X}^d = \mathbf{U}^T \mathbf{B} \mathbf{U}.$$

In vectorized form, we get the linear system

$$[a_0 a_1 \dots a_d] \begin{bmatrix} \text{vec}(\mathbf{I}) \\ \text{vec}(\mathbf{X}) \\ \dots \\ \text{vec}(\mathbf{X}^d) \end{bmatrix} = \text{vec}(\mathbf{U}^T \mathbf{B} \mathbf{U}),$$

in which $\text{vec}(\mathbf{X})$ transforms the matrix $\mathbf{X}$ into a row vector. Since the powers of $\mathbf{X}$ and $\mathbf{U}^T \mathbf{B} \mathbf{U}$ are known, this is a linear system of n^2 equations and $d + 1$ variables. This system is over-specified, and can be solved using the usual methods for systems of linear equations. In analogy with the diagonal case, we can find the best weights a_k by solving the following minimization problem:

$$\min_{a \in \mathbb{R}^{\{0, \dots, d\}}} \left\| \left[\sum_{k=0}^d a_k \mathbf{X}^k \right] - \mathbf{U}^T \mathbf{B} \mathbf{U} \right\|_F$$

The result are the weights a_k of a link prediction function.

V. EVALUATION

Table I gives the list of link prediction functions we use in the evaluations. We use the networks given in Table II for evaluation. All networks are available at konect.uni-koblenz.de.

A. Approximation of Prediction Functions

As mentioned before, the best rank- r approximation to a given matrix is given by a truncation of its singular value decomposition. In this regard the singular value decomposition is optimal. However, finding low-rank approximations of a network's adjacency matrix is not a typical task. Instead, a typical task is to compute functions of the adjacency matrix to predict links, given by power series of the adjacency matrix, such as the matrix exponential. If we now evaluate various matrix decompositions at the task of

Table I
SPECTRAL TRUST PREDICTION METHODS USED IN THE EVALUATION.

	Name	Decomposition	Spectral transformation
SYM	Eigenvalue decomposition	$\mathbf{A} + \mathbf{A}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$	Polynomial [3]
SVD	Singular value decomposition	$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$	Polynomial [3]
COMP	Complex eigenvalue decomposition	$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$	Polynomial [3]
PPR	Personalized PageRank	$\mathbf{D}^{-1}\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$	$(1 - \beta\lambda)^{-1}\lambda$ [6]
LEFT	Left closed-form DEDICOM	$\mathbf{A} = \mathbf{U}(\mathbf{\Sigma}\mathbf{V}^T\mathbf{U})\mathbf{U}^T$	Nondiagonal polynomial (Section IV-B)
RIGHT	Right closed-form DEDICOM	$\mathbf{A} = \mathbf{V}(\mathbf{V}^T\mathbf{U}\mathbf{\Sigma})\mathbf{V}^T$	Nondiagonal polynomial (Section IV-B)
CLO	Symmetric closed-form DEDICOM	$\mathbf{A} = \mathbf{U}\mathbf{X}\mathbf{U}^T$	Nondiagonal polynomial (Section IV-B)
ITER	Iterative DEDICOM	$\mathbf{A} = \mathbf{U}\mathbf{X}\mathbf{U}^T$	Nondiagonal polynomial (Section IV-B)

approximating the exponential of a given matrix, we find that the DEDICOMs perform better than the singular value decomposition. Assume that we want to approximate a link prediction function of $\mathbf{A}$ such as the matrix exponential. We are thus solving

$$\min_{\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times r}} \|e^{\alpha\mathbf{A}} - \mathbf{X}\mathbf{Y}^T\|_F.$$

In the case where $\mathbf{A}$ is symmetric, the eigenvalue decomposition and the singular value decomposition coincide, and we can use the eigenvalue decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ to write $e^{\alpha\mathbf{A}} = e^{\alpha\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T} = \mathbf{U}e^{\alpha\mathbf{\Lambda}}\mathbf{U}^T$ which itself is the eigenvalue decomposition of $e^{\alpha\mathbf{A}}$ and therefore a solution to Equation (1) is given by

$$\mathbf{X} = \mathbf{U}e^{\alpha\mathbf{\Lambda}}, \quad \mathbf{Y} = \mathbf{U}.$$

If $\mathbf{A}$ is asymmetric however the two decompositions do not coincide. The reason for this is that it is wrong in the general case that $e^{\mathbf{A}} = \mathbf{U}e^{\mathbf{\Sigma}}\mathbf{V}^T$ when $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the singular value decomposition of $\mathbf{A}$. Instead, we must look for decompositions of the form $\mathbf{A} = \mathbf{U}\mathbf{X}\mathbf{U}^{-1}$. Since the nonorthogonal eigenvalue decomposition and the DEDICOM-type decompositions are of this type, we may use them both. Figure 1 shows the root mean squared error on the approximation to a graph kernel of the asymmetric adjacency matrix of Advogato for the decompositions given in Table I. As expected the singular value decomposition gives the best approximation to $\mathbf{A}$ itself. For the exponential of $\mathbf{A}$ however, the singular value decomposition is very imprecise and the other decompositions are better. The smallest error is achieved by the iterative DEDICOM.

B. Trust Prediction

We evaluate the performance of our method on the task of trust prediction. Given a directed trust network, the task we are implementing is to predict which directed edges will be added in the future. Given a directed trust network $G = (V, E)$, we split its edge set E into a training set E_{training} and a test set E_{test} . Then, we compute the largest weakly connected component in the training set. All nodes that are not in it are then removed from both the training and the test set. The training set is then split into a source E_{source}

Figure 1. Approximation of the adjacency matrix $\mathbf{A}$ and of the link prediction function $e^{0.1\mathbf{A}}$ for the Advogato trust network.

and a target set E_{target} of edges, which are used to learn the different link prediction functions. When the edges in the networks have known arrival times, we compute the split such that the source set contains edge arrived earlier than the target set, and both of these arrived earlier than the test set. For networks that are unweighted, we augment the test set by a set of vertex pairs (i, j) for which (i, j) is not an edge. As a comparison with non-spectral methods, we include preferential attachment and the friend of a friend count. Let $d_{\text{in}}(i)$ and $d_{\text{out}}(i)$ be the indegree and outdegree of vertex i . Then, the preferential attachment prediction score for the vertex pair (i, j) is given by $d_{\text{out}}(i)d_{\text{in}}(j)$. The friend of a friend count is defined as the number of directed paths of length two between any two nodes.

The other methods are based on matrix decompositions. Let $\mathbf{U}\mathbf{Y}\mathbf{V}^T$ be one of the matrix decompositions given in Table I, in which $\mathbf{U}$ may be equal to $\mathbf{V}$. Then predictions using the corresponding method are computed as $\mathbf{U}p(\mathbf{Y})\mathbf{V}^T$. In the case of the nonorthogonal eigenvalue decomposition (COMP), we use the real part of the resulting complex prediction values. For the decompositions SYM, SVD, COMP, we use the method described in [3] to learn a polynomial. For personalized PageRank (PPR), we use the spectral transformation $(1 - \beta\lambda)^{-1}\lambda$, using the method of [3] to learn β . For the DEDICOM methods, we use the method described in Section IV-B. The link prediction accuracy is

Table III
ALL AVERAGE PRECISION VALUES. THE BEST METHOD FOR EACH NETWORK IS HIGHLIGHTED IN BOLD.

Network	PREF	FOAF	SYM	SVD	COMP	PPR	LEFT	RIGHT	CLO	ITER
Advogato	0.8988	0.8547	0.8765	0.8348	0.7806	0.7138	0.8785	0.8917	0.9290	0.9343
DBLP	0.9406	0.6873	0.9483	0.9594	0.5945	0.5058	0.8288	0.8013	0.9501	0.9162
Email	0.9330	0.8547	0.9971	0.8098	0.9106	0.8133	0.9242	0.9546	0.9850	0.9840
Epinions	0.7885	0.8869	0.8789	0.8408	0.8150	0.8350	0.8174	0.8635	0.8644	0.8903
Slashdot	0.6826	0.8565	0.9044	0.8429	0.8232	0.8249	0.8361	0.8429	0.8844	0.9013
Twitter	0.9849	0.5199	0.9776	0.9169	0.3415	0.5143	0.3378	0.8169	0.9859	0.9107
Wikipedia	0.7699	0.8484	0.8535	0.8250	0.7806	0.7973	0.8427	0.7801	0.8540	0.8453

measured using the average precision in the following way. We sort all vertex pairs (i, j) in the test set by decreasing link prediction scores, and then compute the average precision, based on whether a pair (i, j) represents an edge or a non-edge. The results of the evaluation are given in Table III.

1) *Discussion:* As expected, the naive methods LEFT and RIGHT do not perform well. The closed-form method CLO and the iterative solution ITER both perform well for different trust datasets. For the Slashdot network, the symmetric eigenvalue decomposition SYM performs better, indicating that Slashdot is more a friendship network than a trust network. The complex eigenvalue decompositions (COMP and PPR) do not give accurate results in any network. We interpret this as being due to the complex value of the eigenvalues, which cannot be accurately mapped to other complex values. There are exceptions to these observations: In the Twitter network, the singular value decomposition SVD works better. We interpret this as an indication that the Twitter follower graph has a rather bipartite structure. This is in line with previous results describing Twitter as news media rather than a social network [17].

2) *Other Network Types:* In addition to trust networks, two other important types of directed networks exist: citation networks and communication networks. In citation networks such as DBLP, the bipartite approach is best. In communication networks such as the Email dataset, the symmetric approach is best. An explanation of why DEDICOM works in trust networks and not in citation and communication networks is that trust networks display actual directed transitivity in their edge structure. On the other hand, citation networks follow the co-citation models, which leads to a bipartite structure, and communication networks are essentially symmetric. A conclusion of these results is that we are able to recommend our method for link prediction in trust networks. The resulting trust prediction algorithm gives better prediction values than either the symmetric, bipartite or complex variant.

Acknowledgments: We thank Yoshio Takane, Tamara G. Kolda, Charles Van Loan and Matthias Thimm. The research was funded by the European Community's Seventh Frame Programme under the project ROBUST.

REFERENCES

[1] P. Massa and P. Avesani, "Controversial users demand local trust metrics: an experimental study on epinions.com com-

munity," in *AAAI*, 2005, pp. 121–126.

[2] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *CIKM*, 2003, pp. 556–559.

[3] J. Kunegis and A. Lommatzsch, "Learning spectral graph transformations for link prediction," in *ICML*, 2009, pp. 561–568.

[4] J. Kunegis, E. W. De Luca, and S. Albayrak, "The link prediction problem in bipartite networks," in *IPMU*, 2010, pp. 380–389.

[5] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Review*, vol. 45, no. 1, pp. 1–46, 2003.

[6] F. E. Walter, S. Battiston, and F. Schweitzer, "Personalised and dynamic trust in social networks," in *RecSys*, 2009, pp. 197–204.

[7] R. A. Harshman, "Models for analysis of asymmetrical relationships among n objects or stimuli," in *Proc. First Meeting of the Psychometric Soc. and the Soc. for Mathematical Psychology*, 1978.

[8] H. A. L. Kliers, J. M. F. ten Berge, Y. Takane, and J. de Leeuw, "A generalization of Takane's algorithm for DEDICOM," *Psychometrika*, vol. 55, no. 1, pp. 151–158, 1990.

[9] Y. Takane and Z. Zhang, "Algorithms for DEDICOM: Acceleration, deceleration, or neither?" *J. of Chemometrics*, vol. 23, pp. 364–370, 2009.

[10] R. A. Harshman, P. E. Green, Y. Wind, and M. E. Lundy, "A model for the analysis of asymmetric data in marketing research," *Marketing Science*, vol. 1, no. 2, pp. 205–242, 1982.

[11] D. Stewart, "Social status in an open-source community," *American Sociological Rev.*, vol. 70, no. 5, pp. 823–842, 2005.

[12] M. Ley, "The DBLP computer science bibliography: Evolution, research issues, perspectives," in *Proc. Int. Symp. on String Processing and Information Retrieval*, 2002, pp. 1–10.

[13] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1–40, 2007.

[14] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The Slashdot Zoo: Mining a social network with negative edges," in *WWW*, 2009, pp. 741–750.

[15] M. D. Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Kelliher, "How does the data sampling strategy impact the discovery of information diffusion in social media?" in *ICWSM*, 2010, pp. 34–41.

[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Governance in social media: A case study of the Wikipedia promotion process," in *ICWSM*, 2010, pp. 98–105.

[17] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *WWW*, 2010, pp. 591–600.