A Fuzzy Based Link Analysis for Mining Relational Databases

Sudhakar Krishnan [1*] Rubini Pandu [2] Mangai [3]
1.Assistant Professor, Department of CSE, Sengunthar College of Engineering, Namakkal.
2.Assistant Professor, Department of CSE, Sengunthar Engineering College, Namakkal.
3.Professor, Department of ECE, Velalar College of Engg and Tech, Erode
*E-mail: ksudhakar.cs@gmail.com

Abstract
This work introduces a link analysis procedure for discovering relationships in a relational database or a graph, generalizing both simple and multiple correspondence analysis. It is based on a random walk model through the database defining a Markov chain having as many states as elements in the database. Suppose we are interested in analyzing the relationships between some elements (or records) contained in two different tables of the relational database. To this end, in a first step, a reduced, much smaller, Markov chain containing only the elements of interest and preserving the main characteristics of the initial chain, is extracted by stochastic complementation. This reduced chain is then analyzed by projecting jointly the elements of interest in the diffusion map subspace and visualizing the results. This two-step procedure reduces to simple correspondence analysis when only two tables are defined, and to multiple correspondence analysis when the database takes the form of a simple star-schema. On the other hand, a kernel version of the diffusion map distance, generalizing the basic diffusion map distance to directed graphs, is also introduced and the links with spectral clustering are discussed. Several data sets are analyzed by using the proposed methodology, showing the usefulness of the technique for extracting relationships in relational databases or graphs.

Keywords: Graph mining, link analysis, kernel on a graph, diffusion map, correspondence analysis, dimensionality reduction, statistical relational learning.

1. INTRODUCTION

Traditional statistical, machine learning, pattern recognition, and data mining approaches usually assume a random sample of independent objects from a single relation. Many of these techniques have gone through the extraction of knowledge from data (typically extracted from relational databases), almost always leading, in the end, to the classical double-entry tabular format, containing features for a sample of the population. These features are therefore used in order to learn from the sample, provided that it is representative of the population as a whole. However, real-world data coming from many fields (such as World Wide Web, marketing, social networks, orbiology) are often multi relational and interrelated. The work recently performed in statistical relational learning aiming at working with such data sets, incorporates research topics, such as link analysis, web mining, social network analysis, or graph mining. All these research fields intend to find and exploit links between objects (in addition to features -as is also the case in the field of spatial statistics, which could be of various types and involved in different kinds of relationships. The focus of the techniques has moved over from the analysis of the features describing each instance belonging to the population of interest (attribute value analysis) to the analysis of the links existing between these instances (relational analysis), in addition to the features.

This paper precisely proposes a link-analysis-based technique allowing to discover relationships existing between elements of a relational database or, more generally, a graph. More specifically, this work is based on a random walk through the database defining a Markov chain having as many states as elements in the database. Suppose, for instance, we are interested in analyzing the relationships between elements contained in two different tables of a relational database. To this end, a two-step procedure is developed. First, a much smaller, reduced, Markov chain, only containing the elements of interest—typically the elements contained in the two tables—and preserving the main characteristics of the initial chain, is extracted by stochastic complementation. An efficient algorithm for extracting the reduced Markov chain from the large, sparse, Markov chain representing the database is proposed. Then, the reduced chain is analyzed by, for instance, projecting the states in the subspace spanned by the right eigenvectors of the transition matrix; called the basic diffusion map in this paper), or by computing a kernel principal component analysis on a diffusion map kernel computed from the reduced graph and visualizing the results. Indeed, a valid graph kernel based on the diffusion map distance, extending the basic diffusion map to directed graphs, is introduced. The motivations for developing this two-step procedure are twofold. First, the computation would be cumbersome, if not impossible, when dealing with the complete database. Second, in many situations, the analyst is not interested in studying all the relationships between all elements of the database, but only a subset of them. Moreover, if the whole set of elements in the database is analyzed, the resulting mapping would be averaged out by the numerous relationships and elements we are not interested in—for instance, the principal axis would be completely different. It would therefore not exclusively reflect the relationships between the elements of interest. Therefore, reducing the Markov chain by stochastic complementation allows focusing the analysis on the elements and relationships we are interested in. Interestingly enough, when dealing with a bipartite graph (i.e., the database only contains two tables linked by one relation), stochastic complementation followed by a basic diffusion map is exactly equivalent to simple correspondence analysis. On the other hand, when dealing with a star schema database (i.e., one central table linked to several tables by different relations), this two-step procedure reduces to multiple correspondence analysis. The proposed methodology therefore extends correspondence analysis to the analysis of a relational database. In short, this paper has three main contributions:

- A two-step procedure for analyzing weighted graphs or relational databases is proposed.
- It is shown that the suggested procedure extends correspondence analysis.
- A kernel version of the diffusion map distance, applicable to directed graphs, is introduced.

The paper is organized as follows: Section 2 introduces the basic diffusion map distance and its natural kernel on a graph. Section 3 introduces some basic notions of stochastic complementation of a Markov chain. Section 4 presents the two-step procedure for analyzing the relationships between elements of different tables and establishes the equivalence between the proposed methodology and correspondence analysis in some special cases. Section 5 presents some illustrative examples involving several data sets, while Section 6 gives the conclusion.

2. THE DIFFUSION MAP DISTANCE AND ITS NATURAL KERNEL MATRIX

The basic diffusion map distance is briefly reviewed and some of its theoretical justifications are detailed. Then, a natural kernel matrix is derived from the diffusion map distance, providing a meaningful similarity measure between nodes.

2.1 Notations and Definitions

Let us consider that we are given a weighted, directed, graph G possibly defined from a relational database in the following, obvious, way: each element of the database is a node and each relation corresponds to a link (for a detailed procedure allowing building a graph from a relational database. The associated adjacency matrix A is defined in a standard way as $a_{ij}= [A]_{ij}=w_{ij}$ if node i is connected to node j and $a_{ij} = 0$ otherwise (say G has n nodes in total). The weight $w_{ij} > 0$ of the edge connecting node i and node j is set to have larger value if the affinity between i and j is important. If no information about the strength of relationship is available, we simply set $w_{ij} = 1$(unweighted graph). We further assume that there are no self-loops ($w_{ii} = 0$ for i = 1,…., n) and that the graph has a single connected component; that is, any node can be reached from any other node. If the graph is not connected, there is no relationship at all between the different components and the analysis has to be performed separately on each of them. It is therefore to be hoped that the graph modeling the relational database does not contain too many disconnected components—this can be considered as a limitation of our method. Partitioning a graph into connected components from its adjacency matrix can be done in $O(n^2)$ .Based on the adjacency matrix, the Laplacian matrix L of the graph is defined in the usual manner: L = D - A, where D =Diag(a_i) is the generalized out degree matrix with diagonal entries

$$d_{ii} = [\mathbf{D}]_{ii} = a_{i.} = \sum_{j=1}^{n} a_{ij}.$$

The column vector d =diag(a_i) is simply the vector containing the out degree of each node. Furthermore, the volume of the graph is defined as

$$v_g = vol(G) = \sum_{i=1}^{n} d_{ii} = \sum_{i,j=1}^{n} a_{ij}.$$

From this graph, we define a natural random walk through the graph in the usual way by associating a state to each node and assigning a transition probability to each link. Thus, a random walker can jump from element to element, and each element therefore, represents a state of the Markov chain describing the sequence of visited states. A random variable s(t) contains the current state of the Markov chain at time step t: if the random walker is in state i at time t, then s(t)= i. The random walk is defined by the following single-step transition probabilities of jumping from any state i = s(t) to an adjacent state: $j = s(t=1):P(s(t+1)=j|s(t)=i)=a_{ij}/a_i=P_{ij}$. The transition probabilities only depend on the current state and not on the past ones (first-order Markov chain). Since the graph is completely connected, the Markov chain is irreducible, that is, every state can be reached from any other state. If we denote the probability of being in state i at time t by $x_i(t)=P(s(t)=i)$ and we define P as the transition matrix with entries p_{ij}, the evolution of the Markov chain is characterized by $x(t+1)=P^T x(t)$, with $x(0)=x_0$, and T being the matrix transpose. This provides the state probability distribution $x(t)=\{x1(t),x2(t),\ldots x_n(t)]^T$ at time t once the initial distribution x(0) is known. Moreover, we will denote as $x_i(t)$ the column vector containing the probability distribution of finding the random walker in each state at time t when starting from state i at time t = 0. That is, the entries of the vector $x_i(t)$ are $x_{ij}(t)= P(s(t)=j|s(0)=i), j=1,\ldots n$.

Since the Markov chain represents a random walk on the graph G, the transition matrix is simply P = D$^-$

[1]A. Moreover, if the adjacency matrix A is symmetric, the Markov chain is reversible and the steady-state vector, π, is simply proportional to the degree of each state, d (which has tobe normalized in order to obtain a valid probability distribution). Moreover, this implies that all the eigen values (both left and right) of the transition matrix are real.

2.2 The Diffusion Map Distance

In our two-step procedure, a diffusion map projection, based on the so-called diffusion map distance, will be performed after stochastic complementation. Now, since the original definition of the diffusion map distance deals only with undirected, a periodic, Markov chains, it will first be assumed in Section 2 that the reduced Markov chain, obtained after stochastic complementation, is indeed undirected, aperiodic, and connected—in which case the corresponding random walk defines an irreducible reversible Markov chain. Notice, that it is not required that the original adjacency matrix is irreducible and reversible; these assumptions are only required for the reduced adjacency matrix obtained after stochastic complementation (see the discussion. Moreover, some of these assumptions will be relaxed in Section 2.3, when introducing the diffusion map kernel that is well-defined, even if the graph is directed. The original derivation of the diffusion map, introduced independently by Nadler et al., and Pons and Latapy [42], [43], [46], [47], is detailed in Section 2, but other interpretations of this mapping appeared in the literature (see the discussion at the end of Section 2). Moreover, the basic diffusion map is closely related to correspondence analysis, as detailed in Section 4. For an application of the basic diffusion map to dimensionality reduction, see [35]. Since P is aperiodic, irreducible, and reversible, it is well known that all the eigenvalues of P are real and the eigenvectors are also real (see, e.g., [7], p. 202). Moreover, all its eigenvalues Є[-1,+1], and the eigenvalue 1 has multiplicity one [7]. With these assumptions, Nadler et al. and Pons and Latapy [42], [43], [46], [47] proposed to use as distance between states i and j

$$d_{ij}^2(t) = \sum_{k=1}^{n} \frac{(x_{ik}(t) - x_{jk}(t))^2}{\pi_k}$$
$$\propto (\mathbf{x}_i(t) - \mathbf{x}_j(t))^{\mathrm{T}} \mathbf{D}^{-1} (\mathbf{x}_i(t) - \mathbf{x}_j(t)),$$

since, for a simple random walk on an undirected graph, the entries of the steady-state vector _ are proportional (the / sign) to the generalized degree of each node (the total of the elements of the corresponding row of the adjacency matrix [48]). This distance, called the diffusion map distance, corresponds to the sum of the squared differences between the probability distribution of being in any state after t transitions when starting (i.e., at time t = 0) from two different states, state i and state j. In other words, two nodes are similar when they diffuse through the network—and thus influence the network—in a similar way. This is a natural definition which quantifies the similarity between two states based on the evolution of the states' probability distribution. Of course, when i = j; $d_{ij}(t)$=0. Nadler et al. [42], [43] showed that this distance measure has a simple expression in terms of the right eigenvectors of P:

$$d_{ij}^2(t) = \sum_{k=1}^{n} \lambda_k^{2t} (u_{ki} - u_{kj})^2,$$

where u_{ki} = $[uk]_i$ is component i of the kth right eigenvector, u_k, of P and λ_k is its corresponding eigenvalue. Asusual, the λ_k are ordered by decreasing modulus, so that the contributions to the sum in (3) are decreasing with k. On the other hand, xi(t) can easily be expressed [42], [43] in the
space spanned by the left eigenvectors of P, the v_k,

$$\mathbf{x}_i(t) = (\mathbf{P}^{\mathrm{T}})^t \mathbf{e}_i = \sum_{k=1}^{n} \lambda_k^t \mathbf{v}_k \mathbf{u}_k^{\mathrm{T}} \mathbf{e}_i = \sum_{k=1}^{n} (\lambda_k^t u_{ki}) \mathbf{v}_k,$$

where e_i is the ith column of I, e^i=$[0,\ldots,0,1,0,\ldots,0]^T$; with the single 1 in position i:

2.3 A Kernel View of the Diffusion Map Distance

We now introduce1 a variant of the basic "diffusion map" model introduced by Nadler et al. and Pons and Latapy [42], [43], [46], [47], which is still well-defined when the original graph is directed. In other words, we do not assume that the initial adjacency matrix A is symmetric in
this section. This extension presents several advantages in comparison with the original basic diffusion map:

- The kernel version of the diffusion map is applicable to directed graphs while the original model is restricted to undirected graphs,

- The extended model induces a valid kernel on a graph,
- The resulting matrix has the nice property of being symmetric positive definite—the spectral decomposition can thus be computed on a symmetric positive definite matrix, and finally
- The resulting mapping is displayed in a Euclidean space in which the coordinate axes are set in the directions of maximal variance by using (uncentered if the kernel is not centered) kernel principal component analysis [54], [57] or multidimensional scaling [6], [12].

This kernel-based technique will be referred to as the diffusion map kernel PCA or the KDM PCA. Let us defineW= $(\mathrm{Diag}(\pi))^{-1}$, where π is the stationary distribution of the finite Markov chain. Remember that if the adjacency matrix is symmetric, the stationary distribution
of the natural random walk is proportional to the degree of the nodes, W ά D^{-1} [48].
The diffusion map distance is therefore redefined as

$$d^2_{ij}(t)=x_i(t)-x_j(t))^TW(x_i(t)-x_j(t))$$

since $x_i(t)=(P^T)^te_i$, becomes

$$d^2_{ij}(t)=(e_i-e_j)^TP^tW(P^T)^t(e_i-e_j)$$
$$=(e_i-e_j)^TK_{DM}(e_i-e_j)$$
$$=[K_{DM}]_{ii}+[K_{DM}]_{jj}-[K_{DM}]_{ij}-[K_{DM}]_{ji,}$$

Where we defined

$$K_{DM}(t)=P^tW(P^T)^t$$

treferred to as the diffusion map kernel. Thus, the matrix K_{DM} is the natural kernel (inner product matrix) associated to the squared diffusion map distances [6], [12]. It is clear that this matrix is symmetric positive semidefinite and contains inner products in a euclidean space, where the node vectors are exactly separated by $d_{ij}(t)$ (the proof is straightforward and can be found in [17]—appendix D— where the same reasoning was applied to the commute time kernel). It is therefore a valid kernel matrix.

2.4 Links between the Basic Diffusion Map and the Kernel Diffusion Map

While both representing the graph in a euclidean space, where the nodes are exactly separated by the distances defined by (2), and thus providing exactly the same embedding, the mappings are, however, different for each method. Indeed, the coordinate system in the embedding space differs for each method.

In the case of the basic diffusion map, the eigenvector u_k represents the kth coordinate of the nodes in the embedding space. However, in the case of the diffusion map kernel, since a kernel PCA is performed, the first coordinate axis corresponds instead to the direction of maximal variance in terms of diffusion map distance (2). Therefore, the coordinate system used by the diffusion map kernel is actually different than the one used by the basic diffusion map.

Putting the coordinate system in the directions of maximal variance, and thus computing a kernel PCA, is probably more natural. We now show that there is a close relationship between the two representations. Indeed, from (4), we easily observe that the mapping provided by the basic diffusion map remains the same in function of the parameter t, up to a scaling of each coordinate/dimension (only the scaling changes). This is, in fact, not the case for the kernel diffusion map. In fact, the mapping provided by the diffusion map kernel tends to be the same as the one provided by the basic diffusion map for growing values of t in the case of an undirected graph. Indeed, it can be shown that the kernel matrix can be rewritten as K_{DM} ά $U\Lambda^{2t}U^T$, where U contains the right eigenvectors of P; u_k, as columns. In this case, when t is large, every additional dimension has a very small contribution in comparison with the previous ones. This fact will be illustrated in the experimental section. In practice, we observed that the two mappings are already almost identical when t is equal to 5 or 6.

3. ANALYZING RELATIONS BY STOCHASTIC COMPLEMENTATION

In Section 3, the concept of stochastic complementation is briefly reviewed and applied to the analysis of a graph through the random-walk-on-a-graph model. From the initial graph, a reduced graph containing only the nodes of interest, and which is much easy to analyze, is built.

3.1 Computing a Reduced Markov Chain by Stochastic Complementation

Suppose we are interested in analyzing the relationship between two sets of nodes of interest. A reduced Markov chain can be computed from the original chain, in the following manner: First, the set of states is partitioned into two subsets, S1—corresponding to the nodes of interest to be analyzed—and S2—corresponding to the remaining nodes, to be hidden. We further denote by n1 and n2 (with n1 + n2 = n) the number of states in S1 and S2, respectively; usually n2 >> n1. Thus, the transition matrix is repartitioned as

$$\mathbf{P} = \begin{array}{c c} & \begin{array}{cc} S_1 & S_2 \end{array} \\ \begin{array}{c} S_1 \\ S_2 \end{array} & \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \end{array}.$$

The idea is to censor the useless elements by masking them during the random walk. That is, during any random walk on the original chain, only the states belonging to S1 are recorded; all the other reached states belonging to subset S2 being censored, and therefore, not recorded. One can show that the resulting reduced Markov chain obtained by censoring the states S2 is the stochastic complement of the original chain [41]. Thus, performing a stochastic complementation allows to focus the analysis on the tables and elements representing the factors/features of interest. The reduced chain inherits all the characteristics from the original chain; it simply censors the useless states. The stochastic complement Pc of the chain, partitioned as in (9), is defined as

$$P_c = P_{11}+P_{12}(I-P_{22})^{-1}P_{21.}$$

It can be shownI-P22 that the matrix Pc is stochastic, that is, the sum of the elements of each row is equal to 1; it therefore corresponds to a valid transition matrix between states of interest. We will assume that this resulting stochastic matrix is aperiodic and irreducible, that is, primitive. Indeed, Meyer showed in that if the initial chain is irreducible or aperiodic, so is the reduced chain. Moreover, even if the initial chain is periodic, the reduced chain frequently becomes aperiodic by stochastic complementation. One way to ensure the aperiodicity of the reduced chain is to introduce a small positive quantity on the diagonal of the adjacency matrix A, which does not fundamentally change the model. Then, P has nonzero diagonal entries and the stochastic complement, Pc, is primitive.

4. ANALYZING THE REDUCED MARKOV CHAIN WITH THE BASIC DIFFUSION MAP: LINKS WITH CORRESPONDENCE ANALYSIS

Once a reduced Markov chain containing only the nodes of interest has been obtained, one may want to visualize the graph in a low-dimensional space preserving as accurately as possible the proximity between the nodes. This is the second step of our procedure. For this purpose, we propose to use the diffusion maps introduced in Sections 2.2 and 2.3. Interestingly enough, computing a basic diffusion map on the reduced Markov chain is equivalent to correspondence analysis in two special cases of interest: a bipartite graph and a star-schema database. Therefore, the proposed two step procedure can be considered as a generalization of correspondence analysis.

Correspondence analysis is a widely used multivariate statistical analysis technique which still is the subject of much research efforts simple correspondence analysis aims to provide insights into the dependence of two categorical variables. The relationships between the attributes of the two categorical variables are usually analyzed through a biplot [23]—a 2D representation of the attributes of both variables. The coordinates of the attributes on the biplot are obtained by computing the eigenvectors of a matrix. Many different derivations of simple correspondence analysis have been developed, allowing for different interpretations of the technique, such as maximizing the correlation between two discrete variables, reciprocal averaging, categorical discriminant analysis, scaling and quantification of categorical variables, performing a principal component analysis based on the chi-square distance, optimal scaling, dual scaling, etc. Multiple correspondence analysis is the extension of simple correspondence analysis to a larger number of categorical variables.

4.1 Simple Correspondence Analysis

As stated before, simple correspondence analysis aims to study the relationships between two random variables x1 and x2 (the features) having each mutually exclusive, categorical, outcomes, denoted as attributes. Suppose the variable x1 has n1 observed attributes and the variable x2 has n2 observed attributes, each attribute being a possible outcome value for the feature. An experimenter makes a series of measurements of the features x1, x2 on a sample of v_g individuals and records the outcomes in a frequency (also called contingency) table, f_{ij}, containing the number of individuals having both attribute x1 = i and attribute x2 = j. In our relational database, this corresponds to two tables, each table corresponding to one variable, and containing the set of observed attributes (outcomes) of the variable. The two tables are linked by a single relation This situation can be modeled as a bipartite graph, where each node corresponds to an attribute and links are only defined between attributes of x1 and attributes of x2. The weight associated to each link is set to w_{ij} = f_{ij}, quantifying the strength of the relationship between i and j.

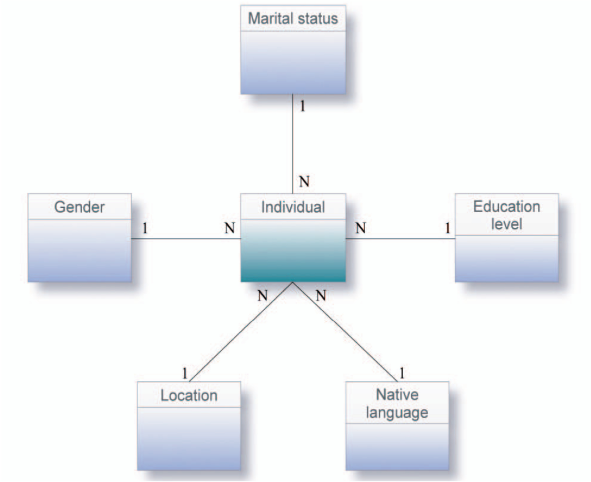
4.2 Multiple Correspondence Analysis

Multiple correspondence analysis assigns a numerical score to each attribute of a set of p > 2 categorical variables Suppose the data are available in the form of a star-schema: the individuals are contained in a main table and the categorial features of these individuals, such as education level, gender, etc., are contained in p auxiliary, satellite, tables. The corresponding graph is built naturally by defining one node for each individual and for each attribute while a link between an individual and an attribute is defined when the individual possesses this attribute. This configuration is known as a star-schema in the data warehouse or relational database fields

Let us first renumber the nodes in such a way that the attribute nodes appear first and the individuals nodes last. Thus, the attributes-to-individuals matrix will be denoted by A_{12}; it contains a 1 on the (i,j) entry when the individual j has attribute i, and 0 otherwise. The individuals-to-attributes matrix, the transpose of the attributes-to-individuals matrix, is A_{21}. Thus, the adjacency matrix of the graph is

$$\mathbf{A} = \begin{bmatrix} \mathbf{O} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{O} \end{bmatrix}.$$

Now, the individuals-to-attributes matrix exactly corresponds to the data matrix $A_{21} = X$ containing, as rows, the individuals and, as columns, the attributes. Since them different features are coded as indicator (dummy) variables, m a row of the X matrix contains a 1 if the individual has the corresponding attribute and 0 otherwise. We thus have $A_{21} =X$ and $A_{12} = X^T$.



Assuming binary weights, the matrix D1 contains on its diagonal the frequencies of each attribute, that is, the number of individuals having this attribute. On the other hand, D2 contains p on each element of its diagonal, since each individual has exactly one attribute for each of the p features (attributes corresponding to a feature are mutually exclusive).

Thus, $D_2 = p$ I and $P_{12} = D_1^{-1}A_{12}$, $P_{21} = D_2^{-1}A_{21}$

Suppose we are first interested in the relationships between attribute nodes, thereby hiding the individual nodes contained in the main table. By stochastic complementation (10), the corresponding attribute-attribute transition matrix is

$$\mathbf{P}_c = \mathbf{D}_1^{-1}\mathbf{A}_{12}\mathbf{D}_2^{-1}\mathbf{A}_{21} = \frac{1}{p}\mathbf{D}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{21}$$
$$= \frac{1}{p}\mathbf{D}_1^{-1}\mathbf{X}^T\mathbf{X} = \frac{1}{p}\mathbf{D}_1^{-1}\mathbf{F},$$

where the element f_{ij} of the frequency matrix $F = X^TX$, also called the Burt matrix, contains the number of co-occurences of the two attributes i and j, that is, the number of individuals having both attribute i and attribute j. The largest nontrivial right eigenvector of the matrix Pc represents the scores of the attributes in a multiple correspondence analysis. Thus, computing the eigenvalues and eigenvectors of Pc and displaying the nodes with coordinates proportional to the eigenvectors, weighted by the corresponding eigenvalue, exactly corresponds to multiple correspondence analysis. This is precisely what we obtain when computing the basic diffusion map on Pc with t = 1. Indeed, as for simple correspondence analysis, it can easily be shown that Pc has real nonnegative eigenvalues, and thus, ordering the eigenvalues by modulus is equivalent to ordering by value.

5 EXPERIMENTS

This experimental section aims to answer four research questions.

- Howdoes the graph mappings provided by the kernel PCA based on the diffusion map kernel (K_{DM} PCA) compares with the basic diffusion map projection?
- Does the proposed two-step procedure (stochastic complementation + diffusion map) provide realistic

subgraph drawings?

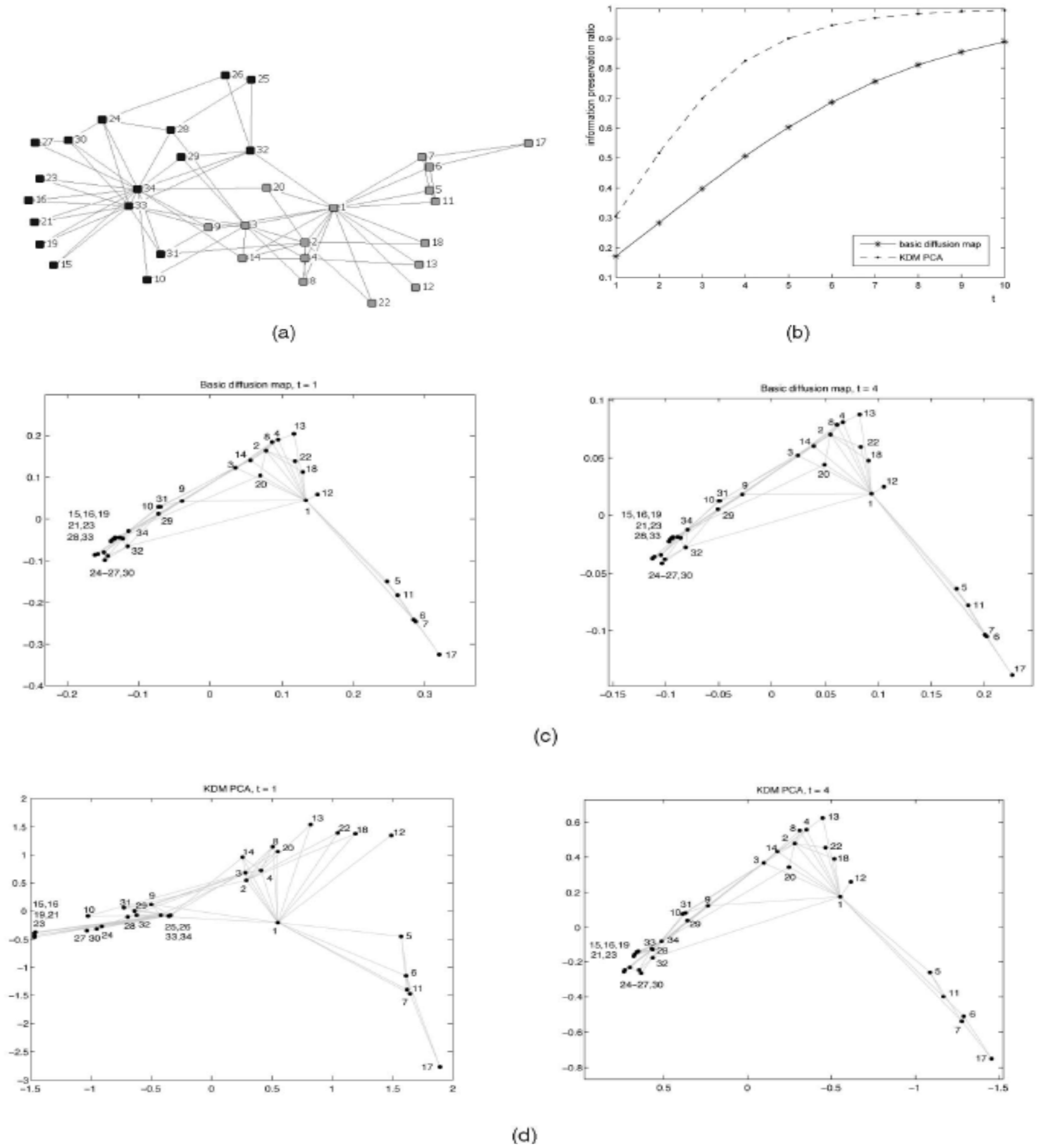
- How the diffusion map kernel combined with stochastic complementation does compares to other popular dimensionality reduction techniques?
- Does stochastic complementation accurately preserve the structural information?

5.1 Graph Embedding

Two simple graphs are studied in order to illustrate the visualization of the graph structure by diffusion maps alone (without stochastic complementation): the Zachary karate club [71] and the dolphins social network.

5.2 Analyzing the Effect of Stochastic

Complementation on a Real-World Data Set This second experiment aims to illustrate the two-step mapping procedure, i.e., first applying a stochastic complementation and then computing the K_{DM} PCA, on a real-world data set—the newsgroups data. However, for illustrative purposes, the procedure is first applied on a toy example.

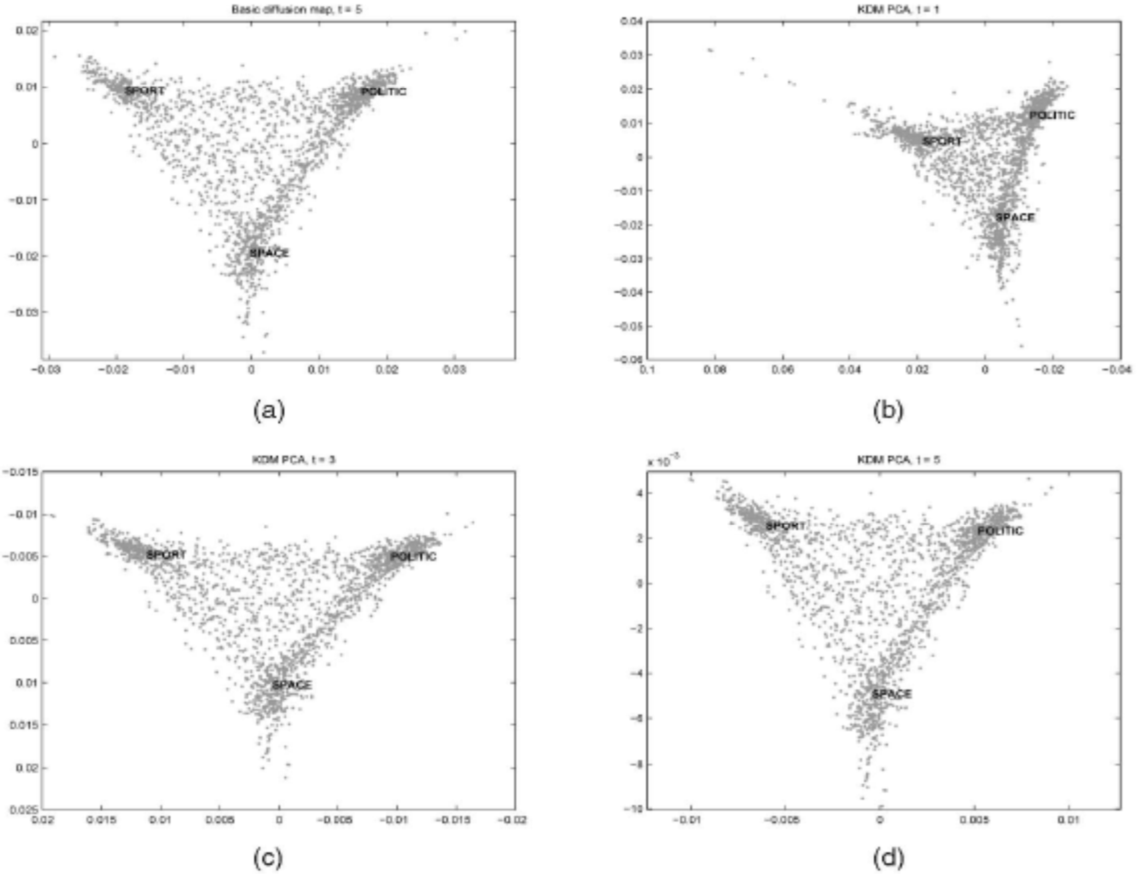


5.3 Graph Reduction Influence and Embedding Comparison

The objective of this experiment is twofold. The first aim is to study the influence of stochastic complementation on graph mapping. The second one is to compare five popular dimensionality reduction methods, namely, the diffusion map kernel PCA (K_{DM} PCA or simply KDM), the Laplacian Eigenmap (LE) [3], the Curvilinear Component Analysis (CCA) [14], Sammon's nonlinear Mapping (SM) [52], and the classical Multidimensional Scaling [6], [12], based on geodesic distances (MDS). For CCA, SM, and MDS, the distance matrix is given by the shortest path distance computed on the reduced graph whose weights are set to the inverse of the entries of the adjacency matrix obtained by stochastic complementation. Notice that the MDS method computed from the geodesic distance on a graph is also known as the ISOMAP method after [61]. Provided that the resulting reduced Markov chain is usually dense, the time complexity of each algorithm is as follows: For K_{DM} PCA, LE, and MDS, the problem is to compute the d dominant eigenvectors of a square matrix since the graph is mapped on a d-dimensional space,

5.4 Discussion of the Results

Let us now come back to our research questions. As a first observation, we can say that the two-step procedure (stochastic complementation followed by a diffusion map projection) provides an embedding in a low-dimensional subspace from which useful information can be extracted. Indeed, the experiments show that highly related elements are displayed close together while poorly related elements tend to be drawn far apart. This is quite similar to correspondence analysis to which the procedure is closely related. Second, it seems that stochastic complementation reasonably preserves proximity information, when combined with a diffusion map (K_{DM} PCA) or an ISOMAP projection (MDS). For the diffusion map, this is normal, since both stochastic complementation and the diffusion map distance are based on a Markov chain model—stochastic complementation is the natural technique allowing censoring states of a Markov chain. On the contrary, stochastic complementation should not be combined with a Laplacian Eigenmap, a curvilinear component analysis, or a Sammon nonlinear mapping—the resulting mapping is not accurate. Finally, the KDM PCA provides exactly the same results as the basic diffusion map when t is large. However, when the parameter t is low, the resulting projection tends to highlight the outlier nodes and to magnify the relative differences between nodes. It is therefore recommended to display a whole range of mappings for several different values of t.



6 CONCLUSIONS AND FURTHER WORK

This work introduced a link-analysis-based technique allowing to analyze relationships existing in relational databases. The database is viewed as a graph, where the nodes correspond to the elements contained in the tables and the links correspond to the relations between the tables. A two-step procedure is defined for analyzing the relationships between elements of interest contained in a table, or a subset of tables. More precisely, this work 1) proposes to use stochastic complementation for extracting a subgraph containing the elements of interest from the original graph and 2) introduces a kernel-based extension of the basic diffusion map for displaying and analyzing the reduced subgraph. It is shown that the resulting method is closely
related to correspondence analysis.

Several data sets are analyzed by using this procedure,showing that it seems to be well-suited for analyzing relationships between elements. Indeed, stochastic complementation considerably reduces the original graph and allows to focus the analysis on the elements of interest, without having to define a state of the Markov chain for each element of the relational database. However, one fundamental limitation of this method is that the relational database could contain too many disconnected components, in which case our link analysis approach is almost useless. Moreover, it is clearly not always an easy task to extract a graph from a relational database, especially when the database is huge. These are the two main drawbacks of the proposed two-step procedure. Further work will be devoted to the application of this methodology to fuzzy SQL queries or fuzzy information retrieval. The objective is to retrieve not only the elements strictly complying with the constraints of the SQL query, but also the elements that almost comply with these constraints and are therefore close to the target elements. We will also evaluate the proposed methodology on real relational databases.

REFERENCES

[1] D.J. Cook and L.B. Holder, Mining Graph Data. Wiley and Sons, 2006.

[2] T. Cox and M. Cox, Multidimensional Scaling, second ed. Chapman and Hall, 2001.

[3] N. Cressie, Statistics for Spatial Data. Wiley, 1991.

[4] P. Demartines and J. Herault, “Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets,” IEEE Trans. Neural Networks, vol. 8, no. 1, pp. 148-154,Jan. 1997.

[5] C. Ding, “Spectral Clustering,” Tutorial presented at the 16thEuropean Conf. Machine Learning (ECML ’05), 2005.

[6] P. Domingos, “Prospects and Challenges for Multi-Relational DataMining,” ACM SIGKDD Explorations Newsletter, vol. 5, no. 1,pp. 80-83, 2003.

[7] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, “Random-Walk Computation of Similarities between Nodes of a Graph,with Application to Collaborative Recommendation,” IEEE Trans.Knowledge and Data Eng, no. 3, pp. 355-369, Mar. 2007.

[8] F. Fouss, J.-M. Renders, and M. Saerens, “Links between Kleinberg’s Hubs and Authorities, Correspondence Analysis and Markov Chains,” Proc. Third IEEE Int’l Conf. Data Mining (ICDM), pp. 521-524, 2003.

[9] F. Fouss, L. Yen, A. Pirotte, and M. Saerens, “An Experimental Investigation of Graph Kernels on a Collaborative Recommendation Task,” Proc. Sixth Int’l Conf. Data Mining (ICDM ’06), pp. 863-868, 2006.

[10] F. Geerts, H. Mannila, and E. Terzi, “Relational Link-Based Ranking,” Proc. 30th Very Large Data Bases Conf. (VLDB), pp. 552-563, 2004.