# A Data-Analysis Technique of Multiple Correspondence Analysis for Discovering Relationships in a Graph

[1]K Phalguna Rao, [2]Ramudu Jonnalagadda

[1]Dept. of IT, Sri Vasavi Engineering College, Pedatadepalli, Tadepalligudem, AP, India

[1]Dept. of CSE, Sri Vasavi Engineering College, Pedatadepalli, Tadepalligudem, AP, India

## Abstract

It introduces a data-analysis procedure for discovering relationships in a graph, generalizing both simple and multiple correspondence analysis. It is based on a random-walk model through the graph defining a Markov chain having as many states as nodes in the graph. Suppose we are interested in analyzing the relationships between some nodes (or records) contained in a graph. 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 nodes of interest in the diffusion-map subspace and visualizing the results. 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 datasets are analyzed by using the proposed methodology, showing the usefulness of the technique for extracting relationships in graphs or relational databases.

## Keywords

Include at least 5 keywords or phrases

## I. Introduction

This work useful in statistical relational learning [4], aiming at working with such data sets, incorporates research topics such as link analysis web mining [6] social-network [5] analysis or graph mining. All these research fields intend to find and exploit links between objects, 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, 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 this two-step procedure are two-fold. 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.

Therefore, reducing the Markov chain by stochastic complementation allows to focus 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 (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.

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.

## II. Stochastic Complementation

From the initial graph, a reduced graph containing only the nodes of interest, and which is much more easy to analyze, is built.A stochastic matrix describes a Markov chain $X_t$ over a f finite state space S. If the probability of moving from i to j in one time step is $P_r(j \mid i) = P_{i,j}$, the stochastic matrix P is given by using $P_{i,j}$ as the ith row and jth column element, e.g.,

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,j} & \cdots \\ p_{2,1} & p_{2,2} & \cdots & p_{2,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ p_{i,1} & p_{i,2} & \cdots & p_{i,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}.$$

Since the probability of transitioning from state i to some state must be 1, this matrix is a right stochastic matrix, so that

$$\sum_i P_{i,j} = 1$$

The probability of transitioning from i to j in two steps is then given by the (i,j)th element of the square of P: $(P^2)_{i,j}$

Two main subgroups can be identified from the mapping

Example: The Cat and Mouse

Suppose you have a timer and a row of five adjacent boxes, with a cat in the first box and a mouse in the fifth box at time zero. The cat and the mouse both jump to a random adjacent box when the timer advances. E.g. if the cat is in the second box and the mouse in the fourth one, the probability is one fourth that the cat will be in the fifth after the timer advances.

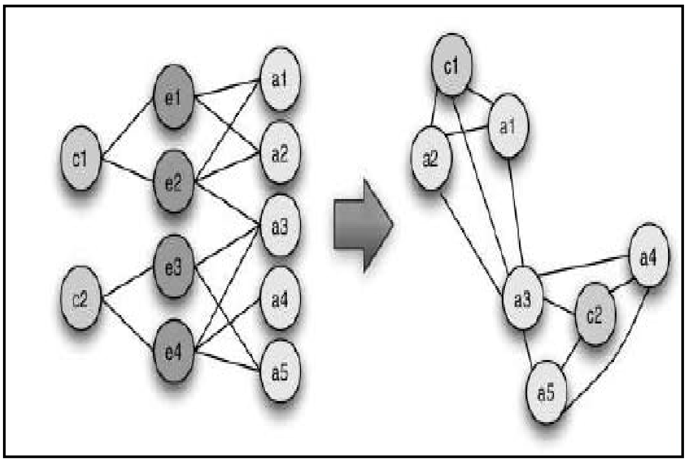

Fig. 1: Toy Example Illustrating Our Two-Step Procedure (Stochastic Complementation Followed By $K_{DM}$ PCA)

If the cat is in the first box and the mouse in the fifth one, the probability is one that the cat will be in box two and the mouse will be in box four after the timer advances. The cat eats the mouse if both end up in the same box, at which time the game ends. The random variable K gives the number of time steps the mouse stays in the game.

The Markov chain that represents this game contains the following five states

State 1: cat in the first box, mouse in the third box: (1, 3)
State 2: cat in the first box, mouse in the fifth box: (1, 5)
State 3: cat in the second box, mouse in the fourth box: (2, 4)
State 4: cat in the third box, mouse in the fifth box: (3, 5)
State 5: the cat ate the mouse and the game ended: F.

We use a stochastic matrix to represent the transition probabilities of this system,

Examples of Markov Chains

$$P = \begin{bmatrix} 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

### A. Compute A Reduced Markov Chain

Formally, a Markov chain [2] is a discrete (discrete-time) random process with the Markov property. Often, the term "Markov chain" is used to mean a Markov process which has a discrete (finite or countable) state-space. Usually a Markov chain is defined for a discrete set of times (i.e., a discrete-time Markov chain) although some authors use the same terminology where "time" can take continuous values. The use of the term in Markov chain Monte Carlo methodology covers cases where the process is in discrete time (discrete algorithm steps) with a continuous state space. The following concentrates on the discrete-time discrete-state-space case.

A discrete-time random process involves a system which is in a certain state at each step, with the state changing randomly between steps. The steps are often thought of as moments in time, but they can equally well refer to physical distance or any other discrete measurement; formally, the steps are the integers or natural numbers, and the random process is a mapping of these to states. The Markov property states that the conditional probability distribution for the system at the next step (and in fact at all future steps) depends only on the current state of the system, and not additionally on the state of the system at previous steps.

Since the system changes randomly, it is generally impossible to predict with certainty the state of a Markov chain at a given point in the future. However, the statistical properties of the system's future can be predicted. In many applications, it is these statistical properties that are important.

The changes of state of the system are called transitions, and the probabilities associated with various state-changes are called transition probabilities. The set of all states and transition probabilities completely characterizes a Markov chain. By convention, we assume all possible states and transitions have been included in the definition of the processes, so there is always a next state and the process goes on forever.

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, $S_1$ -corresponding to the nodes of interest to be analyzed and $S_2$ corresponding to the remaining nodes, to be hidden. We further denote by $n_1$ and $n_2$ (with $n_1 + n_2 = n$) the of states in $S_1$ and $S_2$, respectively; usually $n_2 » n_1$.

Thus, the transition matrix is repartitioned as

$$P = \begin{array}{cc} & \begin{array}{cc} S_1 & S_2 \end{array} \\ \begin{array}{c} S_1 \\ S_2 \end{array} & \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \end{array} \tag{9}$$

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 $S_1$ are recorded; all the other reached states belonging to subset $S_2$ being censored and therefore not recorded. One can show that the resulting reduced Markov chain obtained by censoring the states $S_2$ is the stochastic complement of the original chain. Thus, performing a stochastic complementation [1] 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 $P_c$ of the chain, partitioned as in Equation (9), is defined as,

$$P_c = P_{11} + P_{12}(I - P_{22})^{-1} P_{21} \tag{10}$$

It can be shown 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 a periodic and irreducible, that is, primitive. Indeed, Meyer showed in that if the initial chain is irreducible or a periodic, so is the reduced chain. Moreover, even if the initial chain is periodic, the reduced chain frequently becomes a periodic 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, $P_c$, is primitive.

Let us show that the reduced chain also represents a random walk on a reduced graph $G_c$ containing only the nodes of interest. We therefore partition the matrices A, D, L, as,

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}; \; D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix};$$

$$L = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix};$$

from which we easily find

$P_C = D_{11}(A_{11} + A_{12} \; (D_2 - A_{22})^{-1} A_{21}) = D_{11} A_C D_{11} A_C$, where we defined Ac=$(A_{11} + A_{12} \; (D_2 - A_{22})^{-1} A_{21})$. Notice that if A is symmetric (the graph $G_C$ is undirected), $A_C$ is symmetric as well. Since $P_C$ is stochastic, we deduce that the diagonal matrix $D_1$contains the row sums of $A_C$ and that the entries of $A_C$are positive. The reduced chain thus corresponds to a random walk on the graph $G_C$ whose adjacency matrix is $A_C$.

Moreover, the corresponding Laplacian matrix of the graph $G_C$ can be obtained by

$$L_C = D_1 - A_C = (D_1 - A_{11}) - A_{12}\ (D_2 - A_{22}) - 1A_{21}$$
$$= L_{11} - L_{12}L_2L_{21} \qquad (11)$$

since $L_{12} = -A_{12}$ and $L_{21} = -A_{21}$. If the adjacency matrix A is symmetric, $L_{11}$ ($L_{22}$) is positive definite since it is obtained from the positive semi definite matrix L by deleting the rows associated to $S_2$ ($S_1$) and the corresponding columns, therefore eliminating the linear relationship. Notice that $L_C$ is simply the Schur complement of $L_{22}$. Thus, for an undirected graph G, instead of directly computing $P_C$, it is more interesting to compute $L_C$ , which is symmetric positive definite, from which Pc can easily be deduced: $P_C = I - D_1^{-1}L_C$, directly following from $L_C = D_1 - A_C$; see the next section for a proposition of iterative computation of $L_C$.

A Very simple Weather Model

The probabilities of weather conditions (modeled as either rainy or sunny), given the weather on the preceding day, can be represented by a transition matrix:

$$P = \begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix}$$

The matrix P represents the weather model in which a sunny day is 90% likely to be followed by another sunny day, and a rainy day is 50% likely to be followed by another rainy day. The columns can be labelled "sunny" and "rainy" respectively, and the rows can be labelled in the same order. $(P)_{ij}$ is the probability that, if a given day is of type i, it will be followed by a day of type j.

Notice that the rows of P sum to 1: this is because P is a stochastic matrix.

Predicting The Weather

The weather on day 0 is known to be sunny. This is represented by a vector in which the "sunny" entry is 100%, and the "rainy" entry is 0%:

$$X^{(0)} = (1 \quad 0)$$

The weather on day 1 can be predicted by:

$$X^{(1)} = X^{(0)}P = [1 \quad 0]\begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix} = [0.9 \quad 0.1]$$

Thus, there is an 90% chance that day 1 will also be sunny.

The weather on day 2 can be predicted in the same way:

$$X^{(2)} = X^{(1)}P = X^{(1)}P^2 = [1 \quad 0]\begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix}^2 = [0.86 \quad 0.14]$$

General rules for day n are:

$$X^{(n)} = X^{(n-1)}P$$
$$X^{(n)} = X^{(0)}P^n$$

The Diffusion-Map Distance And Its Natural Kernel Matrix

### A. 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 to build 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 $d_{ii} = [D]_{ii} = a_i = \sum_{j=1}^{n} a_{ij}$. The column vector d= diag$(a_{i.})$ is simply the vector containing the outdegree of each node. Further more, the volume of the graph is defined as ($v_g$) = vol(G) $=\sum_{i=1}^{n} d_{ii} = \sum_{i,j=1}^{n} a_{ij}$. Usually, we are dealing with symmetric adjacency matrices, in which case L is symmetric and positive semi definite.

It defines 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 xi (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 is the matrix transpose. This provides the state probability distribution x(t) = $[x_1(t), x_2(t), \ldots, x_n(t)]^T$ 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, . . . n.

Since the Markov chain represents a random walk on the graph G, the transition matrix is simple P= D −1A. 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 to be normalized in order to obtain a valid probability distribution). Moreover, this implies that all the eigenvalues (both left and right) of the transition matrix are real.

### B. The Diffusion-Map Distance

Diffusion maps[3] are a non-linear technique. It achieves dimensionality reduction by re-organising data according to parameters of its underlying geometry.

The connectivity of the data set, measured using a local similarity measure, is used to create a time-dependent diffusion process. As the diffusion progresses, it integrates local geometry to reveal geometric structures of the data-set at different scales. Defining a time-dependent diffusion metric, we can then measure the similarity between two points at a specific scale (or time), based on the revealed geometry.

A diffusion map [8] embeds data in (transforms data to) a lower-dimensional space, such that the Euclidean distance between points approximates the diffusion distance in the original feature space. The dimension of the diffusion space is determined by the geometric structure underlying the data, and the accuracy by which the diffusion distance is approximated. The rest of this section discusses different aspects of the algorithm in more detail.

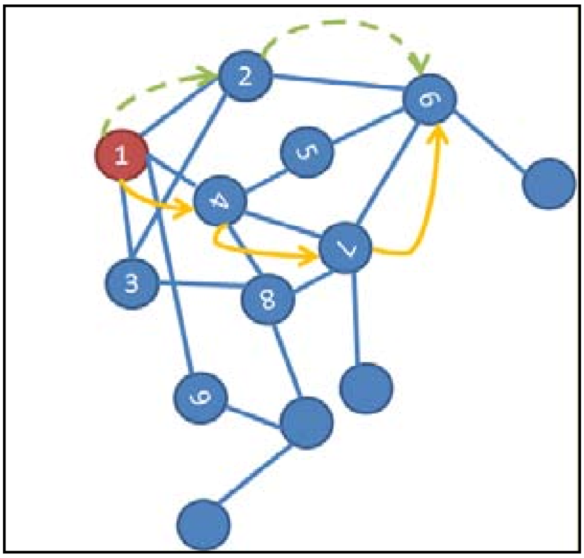

Fig. 2: A random Walk on a Data Set. Each "Jump" has a Probability Associated with it. The Dashed Path Between Nodes 1 and 6 Requires Two Jumps (i.e., Two Time Units) with the Probability Along the Path Being p(Node 1, Node 2) p(Node 2, Node 6)

### 1. Connectivity

Suppose we take a random walk on our data, jumping between data points in feature space (see fig. 1). Jumping to a nearby data-point is more likely than jumping to another that is far away. This observation provides a relation between distance in the feature space and probability.

The connectivity between two data points, x and y, is defined as the probability of jumping from x to y in one step of the random walk, and is connectivity(x, y) = p(x, y).

It is useful to express this connectivity in terms of a nonnormalised likelihood function, k, known as the diffusion kernel:

connectivity(x, y) $\propto$ k(x, y).

The diffusion kernel satisfies the following properties:

1. k is symmetric: k(x, y) = k(y, x)
2. k is positivity preserving: k(x, y) $\geq$ 0

The relation between the kernel function and the connectivity is then

connectivity(x,y)=p(x,y)= $\frac{1}{d_x}$ k(x; y).

with $\frac{1}{d_x}$ the normalisation constant.

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

Each element, Pi,j , is the probability of jumping between data points i and j. When P is squared, it becomes

$$P^2 = \begin{pmatrix} p_{11}p_{11} + p_{12}p_{21} & p_{12}p_{22} + p_{11}p_{12} \\ p_{21}p_{12} + p_{22}p_{21} & p_{22}p_{22} + p_{21}p_{12} \end{pmatrix}$$

Note that $p_{11} = p_{11}p_{11} + p_{12}p_{21}$, which sums two probabilities: that of staying at point 1, and of moving from point 1 to point 2 and back. When making two jumps, these are all the paths from point i to point j. Similarly, Ptij sum all paths of length t from point i to point j.

### 2. Diffusion Process

As we calculate the probabilities Pt for increasing values of t, we observe the data-set at different scales. This is the diffusion process2, where we see the local connectivity integrated to provide the global connectivity of a data-set.

With increased values of t (i.e. as the diffusion process "runs forward"), the probability of following a path along the underlying geometric structure of the data set increases. This happens because, along the geometric structure, points are dense and therefore highly connected Pathways form along short, high probability jumps. On the other hand, paths that do not follow this structure include one or more long, low probability jumps, which lowers the path's overall probability.

In Fig 2, the red path becomes a viable alternative to the green path as the number of steps increases. Since it consists of short jumps, it has a high probability. The green path keeps the same, low probability, regardless of the value of t.

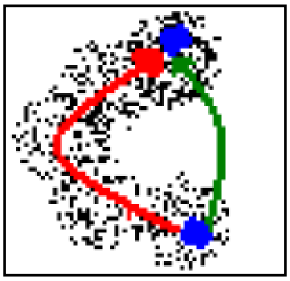

Fig. 3: Paths Along the True Geometric Structure of the Data Set Have High Probability

### 3. Diffusion Distance

The previous section showed how a diffusion process reveals the global geometric structure of a data set. Next, we define a diffusion metric based on this structure. The metric measures the similarity of two points in the observation space as the connectivity (probability of "jumping") between them.

The diffusion distance is small if there are many high probability paths of length t between two points. Unlike isomap's approximation of the geodesic distance, the diffusion metric is robust to noise perturbation, as it sums over all possible paths of length t between points.

As the diffusion process runs forward, revealing the geometric structure of the data, the main contributors to the diffusion distance are paths along that structure.

Consider the term $p_t(x, u)$ in the diffusion distance. This is the probability of jumping from x to u (for any u in the data set) in t time units, and sums the probabilities of all possible paths of length t between x and u.

The diffusion metric manages to capture the similarity of two points in terms of the true parameters of change in the underlying geometric structure of the specific data set.

### 4. Diffusion Map

Low-dimensional data is often embedded in higher dimensional spaces. The data lies on some geometric structure or manifold, which may be non-linear. In the previous section, we found a metric, the diffusion distance that is capable of approximating distances along this structure. Calculating diffusion distances is computationally expensive. It is therefore convenient to map data points into a Euclidean space according to the diffusion metric.

The diffusion distance in data space simply becomes the Euclidean distance in this new diffusion space.

A diffusion map, which maps coordinates between data and diffusion space[9], aims to re-organise data according to the diffusion metric. We exploit it for reducing dimensionality.

Algorithm: Basic Diffusion Mapping Algorithm

INPUT: High dimensional data set $X_i, i = 0..N-1$

Define a kernel, k(x, y) and create a kernel matrix, K, such that $K_{i,j} = k(X_i,X_j)$.

Create the diffusion matrix by normalising the rows of the kernel matrix.

Calculate the eigenvectors of the diffusion matrix.

Map to the d-dimensional diffusion space at time t, using the d dominant eigenvectors.

OUTPUT: Lower dimensional data set Yi, i = 0..N- 1.

Now, since the original definition of the diffusion-map distance[10] deals only with undirected, a periodic, Markov chains, it will first be assumed in the reduced Markov chain, obtained after stochastic complementation, is indeed undirected, a periodic 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 [11] obtained after stochastic complementation. Since P is aperiodic, irreducible and reversible, it is well-known that all the eigen values of P are real and the eigen-vectors are also real. Moreover, all its eigen values ∈ [−1, +1], and the eigen value 1 has multiplicity one. with these assumptions, proposed to use as distance between states i and j,

$$d_{i,j}^2(t) = \sum_{k=1}^{n} \frac{(x_{ik}(t) - x_{jk}(t))^2}{\pi_k} \qquad (1)$$

$$\propto \alpha(x_i(t) - x_j(t))^T D^{-1} x_i(t) - x_j(t) \qquad (2)$$

Since, for a simple random walk on an undirected graph, the entries of the steady-state vector π are proportional (the $\propto$ sign) to the generalized degree of each node. 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. showed that this distance measure has a simple expression in terms of the right eigenvectors of P:

$$d_{i,j}^2(t) = \sum_{k=1}^{n} \lambda_k^{2t} (u_{ki} - u_{kj})^2 \qquad (3)$$

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

$$x_i(t) = \left(P^T\right)^t e_i = \sum_{k=1}^{n} \lambda_k^t v_k u_k u_k^T e_i = \sum_{k=1}^{n} (\lambda_k^t u_{ki}) v_k \qquad (4)$$

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.

The resulting mapping aims to represent each state i in a n-dimensional Euclidean space with coordinates ($|\lambda_2^t|u_{2i}$, $|\lambda_3^t|u_{3i}$, . . ., $|\lambda_n^t|u_{ni}$), as in Equation(4) (the first right eigenvector is trivial and is therefore discarded). Dimensions are ordered by decreasing modulus, $|\lambda_k^t|$. This original mapping introduced by Nadler and coauthors will be referred to as the basic diffusion map in this paper, in contrast with the diffusion-map kernel ($K_{DM}$ ) that will be introduced in the next section.

The weighting factor, D − 1, in Equation (2) is necessary to obtain Equation (3) since the $v_k$ are not orthogonal. Instead, it can easily be shown that we have $v_i^T$ D – 1 $v_j$ =$\delta_{ij}$, which aims to redefine the inner product as <x, y> = $x^T$ D − 1y, where the metric of the space is D $^{-1}$.

Notice also that there is a close relationship between spectral clustering (the mapping provided by the normalized Laplacian matrix;) see for instance and the basic diffusion map. Indeed, a common embedding of the nodes consists of representing each node by the coordinates of the smallest non-trivial eigenvectors (corresponding to the smallest eigen values) of the normalized Laplacian matrix, L= $D^{-1/2}$ L $D^{-1/2}$.More precisely, if $u_k$ is the kth largest right eigenvector of the transition matrix P and $I_K$ ithe kth smallest non-trivial eigenvector of the normalized Laplacian matrix L,

$$u_k = D^{-1/2} I_K \qquad (5)$$

and $I_K$ is associated to eigen value (1 − $\lambda_K$).

A subtle, still important, difference between this mapping and the one provided by the basic diffusion map concerns the order in which the dimensions are sorted. Indeed, for the basic diffusion map, the eigen values of the transition matrix P are ordered by decreasing modulus value. For this spectral-clustering model, the eigen values are sorted by decreasing value(and not modulus), which can result in a different representation if P has large negative eigen values. This shows that the mappings provided by spectral clustering and by the basic diffusion map are closely related.

More generally, these mappings are, of course, also related to graph embedding and nonlinear dimensionality reduction, which have been highly studied topics in recent years, especially in the manifold learning community.

### C. A Kernel View of the Diffusion-Map Distance

We now introduce a variant of the basic "diffusion-map" model introduced by Nadler et al. and Pons & Latapy 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 [12] while the original model is restricted to undirected graphs.
- The extended model induces a valid kernel on a graph, and
- 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 axis are set in the directions of maximal variance by using (uncentered if the kernel is not centered) kernel principal component analysis or multidimensional scaling. This kernel-based technique will be referred to as the diffusion-map kernel PCA or the $K_{DM}$ PCA.

Let us define W = (Diag(π))−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 ∝ D − 1. The diffusion-map distance is therefore redefined as

$$d_{ij}^2(t) = (x_i(t) - x_j(t))^T \mathbf{W} (x_i(t) - x_j(t) \quad (6)$$

Since $x_i(t) = \left(P^T\right)^t e_i x_i(t) = \left(P^T\right)^t e_i$, (6) becomes

$$d_{ij}^2(t) = (e_i - e_j)^T P^t W (P^T)^t (e_i - e_j)$$

$$= (e_i - e_j)^T \mathbf{K}_{\mathrm{DM}} (e_{i-} e_j)$$

$$= [\mathbf{K}_{\mathrm{DM}}]_{ii} + [\mathbf{K}_{\mathrm{DM}}]_{jj} - [\mathbf{K}_{\mathrm{DM}}]_{ij} - [\mathbf{K}_{\mathrm{DM}}]_{ji} \quad (7)$$

where we defined

$$\mathrm{K}_{\mathrm{DM}}(t) = P^t W (P^T)^t \quad (8)$$

referred to as the diffusion-map kernel [13]. Thus, the matrix $\mathrm{K}_{\mathrm{DM}}$ is the natural kernel (inner-product matrix) associated to the squared diffusion-map distances. It is clear that this matrix is symmetric positive semi definite and contains inner products in a Euclidean space where the node vectors are exactly separated by $d_{ij}(t)$. It is therefore a valid kernel matrix.

Notice that the resulting kernel matrix can easily be centered by H$\mathbf{K}_{\mathbf{DM}}$H with H = I − (e $e^T$/n), where e is a column vector all of whose elements are “1” (i.e., e = $[1, 1, \ldots, 1]^T$).H is called the centering matrix. This aims to place the origin of the coordinates of the diffusion map at the center of gravity of the node vectors.

## V. 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. Interesting 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. As stated for instance in 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 a two-dimensional 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 components analysis based on the chi-square distance, optimal scaling, dual scaling. Multiple correspondence analysis is the extension of simple correspondence analysis to a larger number of categorical variables.

### A. Simple Correspondence Analysis

As stated before, simple correspondence analysis (see for instance aims to study the relationships between two random variables $x_1$ and $x_2$ (the features) having each mutually exclusive, categorical, outcomes, denoted as attributes. Suppose the variable $x_1$ has $n_1$ observed attributes and the variable $x_2$ has $n_2$ observed attributes, each attribute being a possible outcome value for the feature. An experimenter makes a series of measurements of the features $x_1$, $x_2$ on a sample of $v_g$ individuals and records the outcomes in a frequency table, $f_{ij}$ , containing the number of individuals having both attribute $x_1$ = i and attribute $x_2$ = 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.

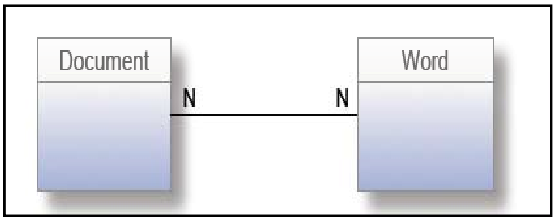

Fig. 4: Trivial example of a single relation between two variables, Document and Word. The Document table contains outcomes of documents while the Word table contains outcomes of words. where O is a matrix full of zeroes.

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. The associated n×n adjacency matrix and the corresponding transition matrix can be factorized as

$$\mathrm{A} = \begin{pmatrix} 0 & \mathrm{A}_{12} \\ \mathrm{A}_{21} & 0 \end{pmatrix}, \quad \mathrm{P} = \begin{pmatrix} 0 & \mathrm{P}_{12} \\ \mathrm{P}_{21} & 0 \end{pmatrix} \quad (14)$$

Suppose we are interested in studying the relationships between the attributes of the first variable x1 which corresponds to the n1 first elements. By stochastic complementation (see Equation (10)), we easily obtain $\mathrm{P_c} = \mathrm{P}_{12}\,\mathrm{P}_{21} = \mathrm{D}_{11}\mathrm{A}_{12}\,\mathrm{D}_{21}\mathrm{A}_{21}$ . Computing the diffusion map for t=1 aims to extract the subdominant right-hand eigenvectors of $\mathrm{P_c}$, which exactly corresponds to correspondence analysis. Moreover, it can easily be shown that Pc has only real non-negative eigen values and thus ordering the eigen values by modulus is equivalent to ordering them by value. In correspondence analysis, eigen values reflect the relative importance of the dimensions: each eigen value is the amount of inertia a given attribute explains in the frequency table. The basic diffusion map after stochastic complementation on this bipartite graph therefore leads to the same results as simple correspondence analysis.

### B. 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 attributes nodes appear first and the individuals nodes last. Thus, the attributes-to-individuals matrix will be denoted by $\mathrm{A}_{12}$ ; 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 $\mathrm{A}_{21}$ . Thus, the adjacency matrix of the graph is

$$\mathrm{A} = \begin{pmatrix} 0 & \mathrm{A}_{12} \\ \mathrm{A}_{21} & 0 \end{pmatrix} \quad (15)$$

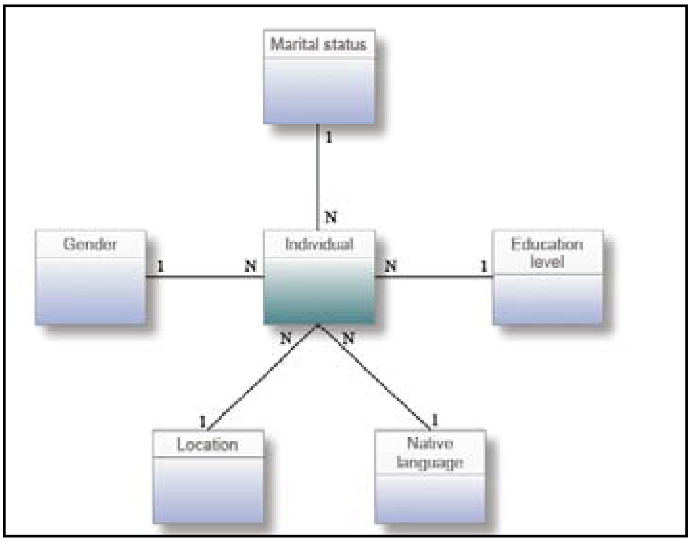

Fig. 5: Trivial example of a star-schema relation between a main variable, Individual, and auxiliary variables, Gender, Education level, etc. Each table contains outcomes of the corresponding random variable

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 the different are coded as indicator (dummy) variables, 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 $A_{12} = X^T$.

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

$$P_c = D_1^{-1}A_{12}D_2^{-1}A_{21} = \frac{1}{p}D_1^{-1}A_{12}A_{21}$$

$$= \frac{1}{p}D_1^{-1}X^TX = \frac{1}{p}D_1^{-1}F \qquad (16)$$

where the element $f_{ij}$ of the frequency matrix F = $X^T$X, 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.

Thus, computing the eigen values and eigen-vectors of $P_c$ and displaying the nodes with coordinates proportional to the eigenvectors, weighted by the corresponding eigen value, exactly corresponds to multiple correspondence analysis. This is precisely what we obtain when computing the basic diffusion map on $P_c$ with t = 1. If we are interested in the relationships between elements of the main table (the individuals) instead of the attributes, we obtain

$$P_c = \frac{1}{p}A_{21}D_1^{-1}A_{12} = \frac{1}{p}XD_1^{-1}X^T \qquad (17)$$

## VI. Experimental Results

Does the proposed two-step procedure (stochastic complementation + diffusion map) provide realistic sub graph drawings

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

### A. Zachary Karate Club

Zachary has studied the relations between the 34 members of a karate club. A disagreement between the club instructor (node 1) and the administrator (node 34) resulted in the split of the club into two parts. Each member of the club is represented by a node of the graph and the weight between nodes (i, j) is set to be the number of time member i and member j met outside the club. The Ucinet drawing of the network Split along administrator/ instructor minimum cut

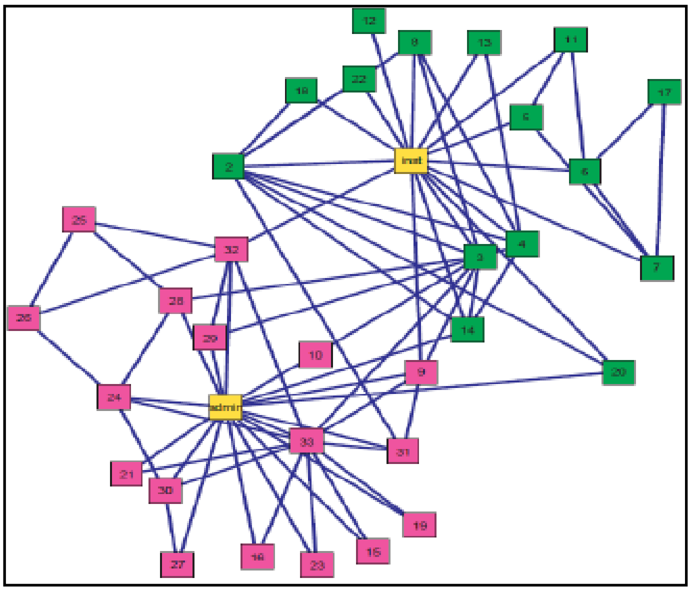

Fig. 5:

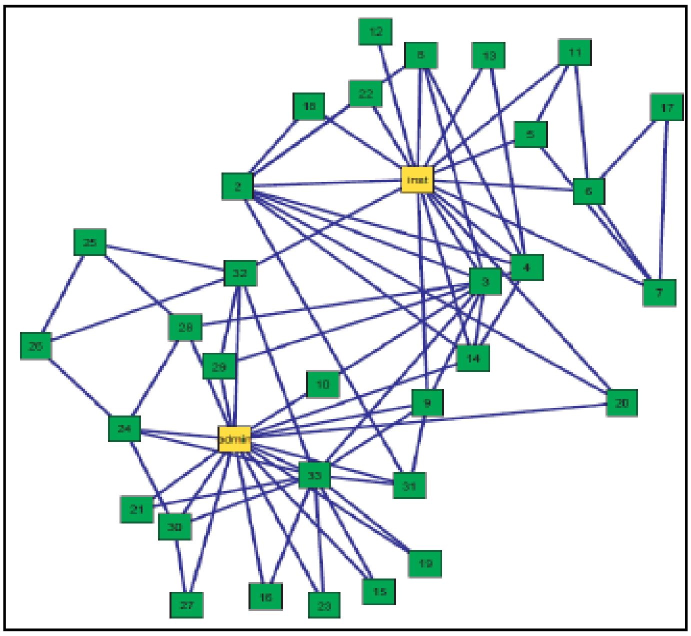

Fig. 6:

The diffusion map reduces the dimension of the graph 40% comparative to general graphs.

## VII. Conclusions And Further Work

This Paper 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

- Proposes to use stochastic complementation for extracting a subgraph containing the elements of interest from the original graph and
- 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 datasets 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] C. D. Meyer,"Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems", SIAM Review, 31(2), pp. 240–272, 1989.

[2] F. Fouss, J.-M. Renders, M. Saerens,"Links between Kleinberg's hubs and authorities, correspondence analysis and Markov chains", In Proceedings of the 3th IEEE International Conference on Data Mining (ICDM), pp. 521–524, 2003.

[3] B. Nadler, S. Lafon, R. Coifman, I. Kevrekidis,"Diffusion maps, spectral clustering and eigen functions of Fokker-Planck operators", Advances in Neural Information Processing Systems 18, pp. 955–962, 2005.

[4] B. Nadler, S. Lafon, R. Coifman, I. Kevrekidis,"Diffusion maps, spectral clustering and reaction coordinate of dynamical systems", Applied and Computational Harmonic Analysis, 21, pp. 113–127, 2006.

[5] C. Blake, E. Keogh, C. Merz.,"UCI repository of machine learning databases", Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[6] S. Chakrabarti,"Mining the Web: Discovering Knowledge from Hypertext Data", Elsevier Science, 2003.

[7] R.R. Coifman, S. Lafon,"Diffusion maps. Applied and Computational Harmonic Analysis", 21(1), pp. 5–30, 2006.

[8] R.R. Coifman, S. Lafon,"Diffusion maps. Applied and Computational Harmonic Analysis", 21(1), pp. 5–30, 2006.

[9] A. Ihler,"Nonlinear Manifold Learning (MIT 6.454 Summary)", 2003.

[10] I.T. Jolliffe,"Principal component analysis", Springer- Verlag New York, 1986.

[11] S.S. Lafon,"Diffusion Maps and Geometric Harmonics", PhD thesis, Yale University, 2004.

[12] Joshua B. Tenenbaum, Vin de Silva, John c Langford, "A Global Geometric Framework ".

K Phalguna rao: recived the M.Tech in Informations Technology(IT) from Andhra University Visakhpatnam pursuing Ph.D from Andhra university His main research Areas include data mining graph mining, Informations Technology(IT) as a Associative Professor computer science Department of C S E – Sri Vasavi Engineering College, Pedatadepalli, Tadepalligudem. W.G.Dt. A.P., India

Ramudu J: recived the BSC dergee in computer science And master dergee in computer science from Acharya Nagarjuna University.(ANU) and Pursuing M.Tech –CSE (II Semester) Sri Vasavi Engineering College, Pedatadepalli. Tadepalligudem.