Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MARCUS WEBER*

Clustering by using a simplex structure

*supported by DFG Research Center "Mathematics for Key Technologies" in Berlin

ZIB-Report 04-03 (February 2004)

Clustering by using a simplex structure

Marcus Weber

Abstract

In this paper we interpret clustering as a mapping of data into a simplex. If the data itself has simplex struture this mapping becomes linear. Spectral analysis is an often used tool for clustering data. We will show that corresponding singular vectors or eigenvectors comprise simplex structure. Therefore they lead to a cluster algorithm, which consists of a simple linear mapping. An example for this kind of algorithms is the Perron cluster analysis (PCCA). We have applied it in practice to identify metastable sets of molecular dynamical systems. In contrast to other algorithms, this approach provides an a priori criterion to determine the number of clusters. In this paper we extend the ideas to more general problems like clustering of bipartite graphs.

Keywords: cluster algorithms, Perron cluster analysis, stochastic matrices, bipartite graphs

MSC: 62H30, 65F15

Contents

1 Introduction - Clustering as a mapping of data into a simplex

Spectral graph partitioning. Graph partitioning with spectral methods is a widely used concept. The papers of Froyland et al.[10, 11, 9], Kamavar et al.[18], Ng et al.[1] and Dhillon [7] are methods for spectral graph partitioning and bipartite graph partitioning similar to the method we present in this paper.

The basic idea of spectral graph partitioning is the classification of points, dynamical states, or molecular configurations etc. by using eigenvectors or singular vectors of matrices derived from the data. In these methods, the vectors are used as input data instead of the original point set. The methods described in the papers above classify the eigenvector or singular vector data with a k-means routine or similar algorithms. Taking a closer look at the vector data turning up in these cluster methods one can make an astonishing observation, which led us to a new and very simple cluster algorithm.

The observation of an astonishing structure of the data. Our starting point for an examination of cluster algorithms were the stochastic transition matrices T of reversible Markov processes [5] arising in molecular dynamics. Here, clustering is used to find a hidden block structure in T.

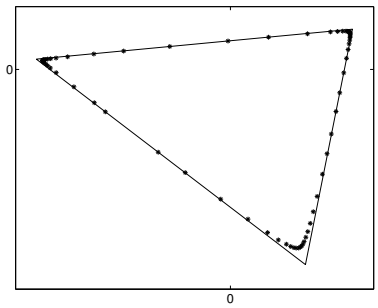

Figure 1: *Simplex structure of 72 data points in the case of 3 clusters.*

To give an example, a spectral analysis of a transition matrix T turning up from an HIV protease inhibitor dynamic simulation led us to three clusters. As we plotted the components of the eigenvectors corresponding to the three highest eigenvalues of T as a three dimensional point set (details are discribed later), we found an amazing structure, see a two dimensional projection in Fig. 1. The data points nearly span a simplex. Other examples can be found in Fig. 4 and Fig. 5 in [18]. Especially this structure implies a very simple cluster method.

Soft concept. Often the solution of a cluster problem is understood as a decomposition of the original discrete and finite data set $\Omega, n = |\Omega|$, into subsets with similar structure. In a vector space model, Ω consists of row-vectors $X(i,:) \in \mathbb{R}^m, i = 1, \ldots, n$ [1].

[1] Throughout this paper, we use matrix indices in MatLab style, i.e. $M(\mathrm{i},\mathrm{j})$ instead of M_{ij}, and also the colon style, i.e. $M(:,\mathrm{j})$ and $M(\mathrm{i},:)$, for the columns and the rows of M.

In fuzzy clustering [14, 13], for every cluster $i = 1, \ldots, s$ there is an n-dimensional vector $\chi(:, \mathrm{i}) \geq 0$, whose components are the *grades of membership* of every data point $j \in \Omega \simeq \{1, \ldots, n\}$ to the cluster i [20].

Since altogether every data point should be assigned to one cluster, we have

$$\sum_{i=1}^{s} \chi(:, \mathrm{i}) = \mathbb{1}_\Omega, \tag{1}$$

where $\mathbb{1}_\Omega \in \mathbb{R}^n$ is the constant vector $\mathbb{1}_\Omega = (1, \ldots, 1)^T$.The solution of an s-cluster problem is therefore a positive stochastic (n, s)-matrix χ.

Clustering as a mapping into a simplex. On input we have an (n, m)-matrix X, on output we have the (n, s)-matrix χ. Each data point $j = 1, \ldots, n$, i.e. a row of X, is therefore mapped into the s-dimensional space, i.e a row of χ.

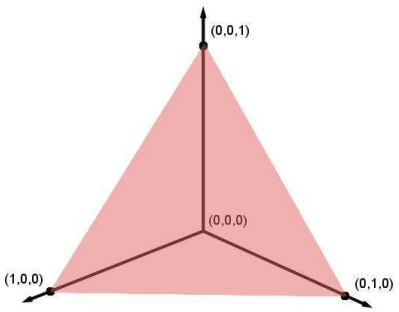

Figure 2: *2-Simplex with vertices (1,0,0), (0,1,0), and (0,0,1).*

But not every point in $\mathbb{R}^s$ lies in the range of this mapping. Since χ is positive and (1) holds, only points out of the standard simplex σ_{s-1}, spanned by the s unit vectors of $\mathbb{R}^s$ are possible, e.g. see Fig. 2 for $s = 3$. Therefore, fuzzy clustering can be understood as a mapping of data points into a simplex.

Clustering as a linear mapping. Things become easy, if $m = s$ and the original data points $X(1, :), \ldots, X(\mathrm{n}, :)$ already lie inside a general $(s-1)$-simplex $\widetilde{\sigma}_{s-1}$, as it is the case in Fig. 1. If in addition the s vertices of this simplex $X(\pi_1, :), \ldots, X(\pi_s, :)$ can be found among the data points, then the row indices $\pi_1, \ldots, \pi_s$ are called *representative* and the mapping $\Omega \to \sigma_{s-1}$ becomes linear, i.e. $\chi = X\,\mathcal{A}$ (see [6, 19]), with an (s, s)-transformation matrix $\mathcal{A}$, where

$$\mathcal{A}^{-1} = \begin{pmatrix} X(\pi_1, 1) & \ldots & X(\pi_1, \mathrm{s}) \\ \vdots & & \vdots \\ X(\pi_s, 1) & \ldots & X(\pi_s, \mathrm{s}) \end{pmatrix}. \tag{2}$$

In practice, we do not know the representatives a priori, but a simple algorithm described in section 3.1 can find them since they are vertices of a simplex-like

data set. We only have to know, that our data set has a simplex-like structure and that the vertices of the simplex are among the data points.

Outline of this paper. In this paper we show that special matrices turning up from dynamical clustering and biclustering problems lead to eigenvectors or singular vectors having nearly simplex structure. This is a new theoretical result. The main work is done in Lemma 2.2 and equation (14) for transition matrices and in equations (18) and (19) for adjacency matrices of bipartite graphs.

We also present some new results on how to fix the number of clusters a priori. The main work on this topic is done in section 2.1.3 and in section 2.2.3.

For the reason of illustration of the ideas of this paper, we wrote some non-sophisticated software code in MatLab style in section 3.

In spectral s-partitioning methods often a smaller subset of m eigenvectors $m < s$, is used as input data, e.g. $m = \lceil \log_2 s \rceil$ in [7, 9]. In section 2.3 we show that this may lead to a wrong classification result.

2 Special matrices and simplex structures

In this section two different matrices T and A are described. The first one is an (n, n)-transition matrix T turning up from reversible Markov chains, e.g. for the determination of almost invariant sets in conformational dynamics [5]. This matrix leads to an eigenvalue problem

$$TX = X\Lambda \tag{3}$$

with an (n, n)-eigenvector matrix X and a diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ containing the eigenvalues $\lambda_1, \ldots, \lambda_n$ in order along its diagonal. For a reversible Markov chain the stochastic transition matrix $T = D^{-1}S$ is the product of a positive (n, n)-diagonal weight matrix D with a nonnegative symmetric (n, n)-matrix S. This is the so called *detailed balance* condition [5]. In this case the eigenvalues and eigenvectors are real and the eigenvectors are orthogonal w.r.t. the weighted inner product $\langle x, y\rangle = x^T\, D\, y$.

The second matrix is an (n, m)-adjacency matrix A, $n > m$, of a bipartite graph, i.e. a rectangular matrix with entries 1 and 0. This matrix leads to a singular value decomposition

$$AY = X\Sigma, \quad A^T X = Y\Sigma^T \tag{4}$$

with an orthogonal (n, n)-singular vector matrix X, an orthogonal (m, m)-singular vector matrix Y, and a (n, m)-singuar value matrix Σ containing the singular values $\sigma_1 \geq \ldots \geq \sigma_m \geq 0$ in order along its diagonal. Such matrices turn up, e.g., in search engines [15, 3] or document clustering [7, 3] as term-document-matrices.

2.1 Stochastic transition matrix

Imagine a dynamical system in which we want to find sets of initial states that are almost invariant with regard to the equation of motion evaluated for a certain time span.

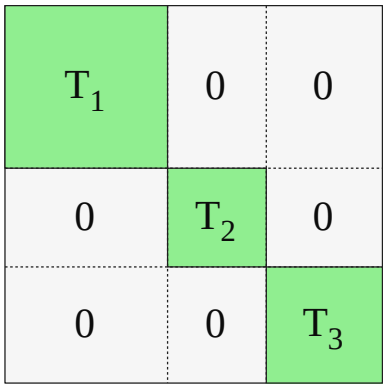

Figure 3: *Structure of a transition matrix in the case of $s = 3$ uncoupled Markov chains after a suitable permutation of state indices.*

For this purpose we decompose the space of interest into a finite set Ω of n states. Then we sample the dynamics via computer experiments and count the transitions between these states. From this Markov experiment we get an (n, n)-matrix S of transitions. By dividing each row of S by its rowsum, we get a stochastic matrix $T = D^{-1}S$. Searching almost invariant sets is equivalent to discovering a hidden block structure in T, as shown in Fig. 3.

Since in molecular dynamics the underlying Hamiltonian differential equation is reversible, the matrix S is symmetric. In the case where the matrix S is symmetric, the Markov process is also reversible. Therefore, the stochastic transition matrix T has a real-valued spectrum. For more details see [17].

2.1.1 Eigenvectors of a reducible transition matrix

If $T(0)$ is the transition matrix of a Markov process having $s \in \mathbf{N}$ distinct invariant index subsets, i.e. if it is realized by s uncoupled Markov chains, then $T(0)$, after suitable permutation, is a block diagonal matrix having s blocks, see Fig. 3 for $s = 3$. Each of these blocks again is a stochastic matrix.

Eigenpairs of $T(0)$. For a decomposition $C_1 \cup \ldots \cup C_s = \{1, \ldots, n\}$ according to the block structure of $T(0)$ we define the characteristic vectors $\mathbb{1}_1, \ldots, \mathbb{1}_s \in \mathbb{R}^n$ as

$$\mathbb{1}_i(j) = \begin{cases} 1, & \text{if } j \in C_i \\ 0, & \text{else.} \end{cases}$$

For the reducible transition matrix $T(0)$ having s blocks we get an s-dimensional eigenspace for the eigenvalues $\lambda_1, \ldots, \lambda_s = 1$ spanned by the characteristic vectors $\mathbb{1}_1, \ldots, \mathbb{1}_s$.

This means, that if we solve the eigenvector problem for the *Perron eigenvalue* [16, 5] $\lambda_1 = \ldots = \lambda_s = 1$ of $T(0)$ we get a basis of eigenvectors, which can be linearly transformed into the solution $\mathbb{1}_1, \ldots, \mathbb{1}_s$ of the cluster problem.

Perturbation. In practice we do not have reducible transition matrices with perfect block structure. But we have slightly perturbed matrices $T(\epsilon)$ with a

perturbation parameter ϵ and an expansion

$$T(\epsilon) = T(0) + \epsilon\, T^{(1)} + O(\epsilon^2)$$

with a constant first order error matrix $T^{(1)}$. The s-fold Perron eigenvalue 1 of $T(0)$ degenerates into a single eigenvalue $\lambda_1 = 1$ with a constant eigenvector $\mathbb{1}_\Omega$, and $s-1$ eigenvalues $\lambda_2, \ldots, \lambda_s \leq 1$ of $T(\epsilon)$ near 1. This is called the Perron cluster of eigenvalues and therefore we call the corresponding cluster algorithm PCCA — Perron Cluster Cluster Analysis.

In this situation we are interested in *almost* invariant index subsets, i.e. an ϵ-perturbation $\chi(:,1), \ldots, \chi(:,\mathrm{s}) \in \mathbb{R}^n$ of the characteristic vectors $\mathbb{1}_1, \ldots, \mathbb{1}_s$, such that the result $\chi \geq 0$ is a clustering in the sense of equation (1).

For a detailed examination of the perturbed eigenvectors see [6] and Lemma 2.2 below. In the following we only consider perturbed transition matrices and therefore write T instead of $T(\epsilon)$.

2.1.2 Simplex structure of perturbed eigenvectors

A spacial case of a perturbed transition matrix is an (n,n)-matrix T with s row indices $\pi_1, \ldots, \pi_s$ and positive factors α_{ki} for $k = 1, \ldots, n$ and $i = 1, \ldots, s$, such that each row $T(\mathrm{k},:)$ is a linear combination of the s representative rows, i.e.

$$T(\mathrm{k},:) = \sum_{i=1}^{s} \alpha_{ki}\, T(\pi_i,:).$$

The matrix T is stochastic, i.e. the sum of each row is 1. Therefore we get

$$\begin{aligned}
\sum_{i=1}^{s} \alpha_{ki} &= \sum_{i=1}^{s} \alpha_{ki} \cdot 1 \\
&= \sum_{i=1}^{s} \alpha_{ki} \sum_{j=1}^{n} T(\pi_i, \mathrm{j}) \\
&= \sum_{j=1}^{n} \sum_{i=1}^{s} \alpha_{ki}\, T(\pi_i, \mathrm{j}) \\
&= \sum_{j=1}^{n} T(\mathrm{k}, \mathrm{j}) \\
&= 1. \qquad (5)
\end{aligned}$$

The sum of the linear combination factors is 1, i.e. they are *convex combination factors*.

The main idea of this paper is that the convex combination of representative rows of T leads to the same convex combination of the components of the eigenvectors. For $k, l = 1, \ldots n$,with $\lambda_l \neq 0$, and convex combination factors α_{ki} , $i = 1, \ldots, s$ this can be shown by

$$\begin{aligned}
X(\mathrm{k},\mathrm{l}) &= \lambda_l^{-1}\, T(\mathrm{k},:)\, X(:,\mathrm{l}) \\
&= \lambda_l^{-1}\, (\sum_{i=1}^{s} \alpha_{ki}\, T(\pi_i,:))\, X(:,\mathrm{l}) \\
&= \lambda_l^{-1} \sum_{i=1}^{s} \alpha_{ki}\, T(\pi_i,:)\, X(:,\mathrm{l}) \qquad (6) \\
&= \sum_{i=1}^{s} \alpha_{ki}\, X(\pi_i, \mathrm{l}).
\end{aligned}$$

Equation (6) shows that the data points $X(1, \mathrm{ind}), \ldots, X(\mathrm{n}, \mathrm{ind})$ lie inside a simplex spanned by the vertices $X(\pi_1, \mathrm{ind}), \ldots, X(\pi_s, \mathrm{ind})$, where "ind" is the set of indices l with $\lambda_l \neq 0$.

In general, the rows of T are not convex combinations of some representative ones. In the following we extend equation (6) to more general matrices (see equation (13)).

ps-transition matrices. Conformation dynamics of biomolecules is the application field in which we use Perron Cluster Analysis.

The dynamics of a molecule is driven by a potential energy landscape. Molecule geometries with low potential energy have a vanishing propability for a transition into another conformation. These states are called *pure states* in the transition matrix. We now define a *pure state transition matrix*. A special structure which can often be found in conformation dynamics.

Definition 2.1 (ps-transition matrix) *An (n, n)-transition matrix is called a* pure state *transition matrix (ps-transition matrix), if there exists a disjoint decomposition $C_1 \cup \ldots \cup C_s = \{1, \ldots, n\}$ of the index set such that there are s row indices $\pi_1, \ldots, \pi_s \in \{1, \ldots, n\}$, such that for all $i, j = 1, \ldots, s$*

$$\mathbb{1}_j \; T(\pi_i, :) = \delta_{ij}.$$

$\pi_1, \ldots, \pi_s$ are called pure states*.*

Lemma 2.2 *Each row $T(\mathrm{k}, :)$ of a ps-transition matrix T can be written as a convex combination of s representative rows with the exception of an additive error vector $X_k^{\perp} \in \mathbb{R}^n$ which is* first order orthogonal *w.r.t. the space spanned by the eigenvectors $X(:, 1), \ldots, X(:, \mathrm{s})$, i.e. there exist α and π such that*

$$T(\mathrm{k}, :) = X_k^{\perp} + \sum_{i=1}^{s} \alpha_{ki} \, T(\pi_i, :) \tag{7}$$

with

$$X_k^{\perp} \, X(:, \mathrm{l}) = O(\epsilon^2), \quad l = 1, \ldots, s.$$

Proof: Since T is a ps-transition matrix there exist representative rows $\pi_1, \ldots,$ π_s of T with

$$\mathbb{1}_j \, T(\pi_i, :) = \delta_{ij} \tag{8}$$

as in Definition 2.1, where $C_1 \cup \ldots \cup C_s$ is a decomposition of the index set and defines the unperturbed case. Further, let the linear combination factors α_{ki} equal the probability of a transition from state k into the index subset C_i, i.e.

$$\alpha_{ki} = \mathbb{1}_i \, T(\mathrm{k}, :). \tag{9}$$

Since these factors are positive and they meet the partition-of-one property, i.e.

$$\sum_{i=1}^{s} \alpha_{ki} = 1,$$

they are convex combination factors.

It is left to show that with this choice of α and π the error vectors in (7) meet the first order orthogonality. Equations (8) and (9) lead to the orthogonality condition

$$X_k^{\perp}\, \mathbb{1}_j = (T(\mathrm{k},:) - \sum_{i=1}^{s} \alpha_{ki}\, T(\pi_i,:))\, \mathbb{1}_j = 0. \tag{10}$$

Since $C_1 \cup \ldots \cup C_s$ is the decomposition of the index set in the unperturbed case, a perturbation result from Deuflhard et al. [6] (Lemma 1.1) is

$$X(:,\mathrm{l}) - \sum_{j=1}^{s} b_{jl}\, \mathbb{1}_j = O(\epsilon^2), \quad l = 1, \ldots, s \tag{11}$$

with suitable $b_{jl} \in \mathbb{R}, \quad j, l = 1, \ldots, s$.

From equations (11) and (10) we get the first error orthogonality result

$$X_k^{\perp}\, X(:,\mathrm{l}) = O(\epsilon^2), \quad l = 1, \ldots, s. \tag{12}$$

This completes the proof. □

Simplex structure. With this modification, equation (6) for $l = 1, \ldots, s$ and $k = 1, \ldots, n$ becomes

$$\begin{aligned} X(\mathrm{k},\mathrm{l}) &= \lambda_l^{-1}\, T(\mathrm{k},:)\, X(:,\mathrm{l}) \\ &= \lambda_l^{-1}\, (X_k^{\perp} + \sum_{i=1}^{s} \alpha_{ki}\, T(\pi_i,:))\, X(:,\mathrm{l}) \\ &= \lambda_l^{-1} \sum_{i=1}^{s} \alpha_{ki}\, T(\pi_i,:)\, X(:,\mathrm{l}) + O(\epsilon^2) \\ &= \sum_{i=1}^{s} \alpha_{ki}\, X(\pi_i,\mathrm{l}) + O(\epsilon^2) \end{aligned} \tag{13}$$

i.e. the data points $X(1,1{:}\mathrm{s}), \ldots, X(\mathrm{n},1{:}\mathrm{s})$ nearly span a simplex with the vertices $X(\pi_1,1{:}\mathrm{s}), \ldots, X(\pi_s,1{:}\mathrm{s})$.

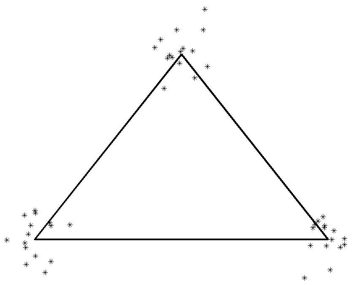

Figure 4: *Unlikely simplex structure of data for $s = 3$, because equation (14) contradicts this situation.*

Better than $O(\epsilon^2)$. In [6] Deuflhard et al. have shown by equation (11), that the eigenvectors nearly span a simplex $\widetilde{\sigma}_{s-1}$, and the corresponding data points $X(1, 1:\mathrm{s}), \ldots, X(\mathrm{n}, 1:\mathrm{s})$ lie in its vertices with a deviation $O(\epsilon^2)$. This could mean, that the data points are distributed around the vertices, as in Fig. 4. In that case a k-means method would be preferable to cluster the data points. However, equation (10) can be used to show that in molecular dynamics the deviation of the data points from the simplex facets is much smaller. Therefore Fig. 1 for $s = 3$ points up the true picture, which favours the soft concept described in section 1.

This can be shown as follows. If we define the error vector $a_l \in \mathbb{R}^n$ via

$$a_l := X(:,\mathrm{l}) - \sum_{j=1}^{s} b_{jl}\, \mathbb{1}_j = O(\epsilon^2), \quad l = 1, \ldots, s,$$

then $\|a_l\|_\infty = O(\epsilon^2)$ is the maximal deviation of the data points from the vertices of $\widetilde{\sigma}_{s-1}$. For an irreducible stochastic matrix T it is true that $\|T(\mathrm{k},:)\, a_l\|_\infty < \|a_l\|_\infty$, for any $k = 1, \ldots, n$. From equation (10) we conclude

$$\|X_k^\perp\, X(:,\mathrm{l})\|_\infty = \|X_k^\perp\, a_l\|_\infty < \|a_l\|_\infty,$$

which leads to

$$\|X(\mathrm{k},:) - \sum_{i=1}^{s} \alpha_{ki}\, X(\pi_i,:)\|_\infty < \|a_l\|_\infty = O(\epsilon^2)$$

for the deviation of data from the simplex facets in equation (13). Since the deviation from constant level pattern of the eigenvectors of T is maximal for "transition states", and transition states have high fluctuation in molecular dynamics (Bolzmann distribution), small values of $T(\mathrm{k},\mathrm{i})$ coincide with high values of $a_l(i)$. Therefore we have $\|T(\mathrm{k},:)\, a_l\|_\infty \ll \|a_l\|_\infty$ in that case, i.e.

$$\|X(\mathrm{k},:) - \sum_{i=1}^{s} \alpha_{ki}\, X(\pi_i,:)\|_\infty \ll \|a_l\|_\infty = O(\epsilon^2). \tag{14}$$

Cluster algorithm. We do not know the representative rows $\pi_1, \ldots, \pi_s$ of a transition matrix T a priori. And we do not know α. But from Lemma 2.2 and especially from equation (14) we know that the corresponding data points $X(\pi_1, 1{:}\mathrm{s}), \ldots, X(\pi_s, 1{:}\mathrm{s})$ of the eigenvector matrix X are nearly the vertices of a simplex including all other data points, if we assume T to be a ps-transition matrix.

As we have seen in section 1 above, this is enough information, because we can apply an algorithm that finds the vertices of a simplex-like data set and then compute the solution χ of the cluster problem via $\chi = X\mathcal{A}$, see section 3.1 and equation (2).

χ may have negative elements, because the simplex structure is a first order result. Therefore the indicator

$$-\min \chi \geq 0 \tag{15}$$

can be used to get the magnitude of deviation $O(\epsilon^2)$. And perhaps as an a posteriori criterion for the quality of the clustering [20, 19]. In addition, Theorem 2.1 in [6] describes the correspondence between uniqueness of the clustering and $\min \chi = 0$.

2.1.3 Eigenvalues of T and the number of clusters

To fix the number of almost invariant sets a priori, we could e.g. count the number of eigenvalues of T which are greater than a preset lower bound MINVALUE.

Note, that there are other heuristics to determine the number of clusters [4].

Since the detailed balance condition holds for T, we can write $T = D^{-1}S$ with a diagonal (n,n)-matrix D and a symmetric (n,n)-matrix S. The following lemma may be useful for a presetting of MINVALUE.

Lemma 2.3 *For a stochastic transition matrix $T = D^{-1}S$ of a reversible Markov chain we have $\lambda_s(T) \geq 1 - \rho(T - \overline{T})$, where $\overline{T} = D^{-1}\bar{S}$ is a stochastic, reducible matrix having s blocks with a corresponding index set decomposition $C_1, \ldots, C_s$. $\bar{S}$ is defined as follows*

$$\bar{S}(\mathrm{i},\mathrm{j}) = \begin{cases} S(\mathrm{i},\mathrm{j}) & ,\mathrm{i} \neq \mathrm{j} \\ S(\mathrm{i},\mathrm{i}) + \sum_{k=1;i\notin C_k}^{s} \sum_{l \in C_k} S(\mathrm{i},\mathrm{l}) & ,\mathrm{i} = \mathrm{j}. \end{cases}$$

Proof. By construction, $\overline{T}$ is detailed balanced, stochastic and has s blocks, i.e. $\lambda_s(\overline{T}) = 1$. From [19] equation (1) we know that

$$D^{-0.5}SD^{-0.5} \quad \text{and} \quad D^{-0.5}\bar{S}D^{-0.5}$$

are symmetric matrices with the same eigenvalues as T and $\overline{T}$, respectively. From [12] Corollary 8.1.6 we get

$$\begin{aligned} 1 - \lambda_s(T) &= |\lambda_s(T) - \lambda_s(\overline{T})| \\ &= |\lambda_s(D^{-0.5}SD^{-0.5}) - \lambda_s(D^{-0.5}\bar{S}D^{-0.5})| \\ &\leq \|D^{-0.5}SD^{-0.5} - D^{-0.5}\bar{S}D^{-0.5}\|_2 \\ &= \|D^{-0.5}(S - \bar{S})D^{-0.5}\|_2 \\ &= \max\{|\lambda_1(D^{-0.5}(S - \bar{S})D^{-0.5})|, |\lambda_n(D^{-0.5}(S - \bar{S})D^{-0.5})|\} \\ &= \max\{|\lambda_1(D^{-1}(S - \bar{S}))|, |\lambda_n(D^{-1}(S - \bar{S}))|\} \\ &= \max\{|\lambda_1(T - \overline{T})|, |\lambda_n(T - \overline{T})|\} \\ &= \rho(T - \overline{T}). \end{aligned}$$

This completes the proof. □

Corollary 2.4 *A direct consequence of Lemma 2.3 is*

$$\lambda_s(T) \geq 1 - \|T - \overline{T}\|$$

for any submultiplicative matrix norm $\|\cdot\|$.

Example 2.5 *If we want the error $\epsilon = \|T - \overline{T}\|$ to be small, then a necessary condition is $\lambda_s(T) \geq 1 - \epsilon$.*

2.2 Adjacency matrix of a bipartite graph

We now consider the clustering of bipartite graphs. An undirected bipartite graph $G = (V_1, V_2, E)$ consists of two sets of vertices $V_1 = \{1, \ldots, m\}$, $V_2 = \{1, \ldots, n\}$ and a set of edges $E \subseteq V_1 \times V_2$ only connecting points of V_1 with points of V_2. The corresponding (n, m)-adjacency matrix A is rectangular.

In this case, clustering or bi-clustering means, that we search for subgraphs $G' = (V_1', V_2', E')$ of G with $V_1' \subset V_1$, $V_2' \subset V_2$ and $E' = E \cap (V_1' \times V_2')$, such that these graphs G' are nearly bi-cliques.

In bi-cliques every vertex of V_1' is conected with every vertex of V_2'.

2.2.1 Reducible adjacency matrix

The ideal case is a bipartite graph with s disjoint bi-cliques. In this case the adjacency matrix has got s distinct rectangular blocks where each element of these blocks is 1 and the other elements are 0. For example $s = 3$ and

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Obviously we have $\mathrm{rank}(A) = s$. In this case the singular value decomposition (SVD) of A, see eq. (4), has the following property

$$A = X(:, 1:\mathrm{s})\, \Sigma(1:\mathrm{s}, 1:\mathrm{s})\, Y(:, 1:\mathrm{s})^T,$$

because we compute the singular values $\sigma_i = 0$ for $i > s$, see [15] pp. 53-57. This means, that the whole information for A is contained is the first s singular vectors.

In particular, the matrix A_k computed from the first k singular vectors and the (k, k)-matrix Σ_k having the highest singular values on its diagonal is the best rank-k approximation of A, i.e.

$$\|A - A_k\|_F = \min_{\mathrm{rank}(B)=k} \|A - B\|_F = \sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2} \tag{16}$$

for the Frobenius norm $\|\cdot\|_F$, this follows from [15] (4.7).

2.2.2 Adjacency matrix with overlapping bi-cliques

Linear combination of representative rows. In this section we consider adjacency matrices having s blocks and $\mathrm{rank}(A) = s$, but the blocks in A may overlap, see example 2.6 and Fig. 5. In that case some row or column indices correspond to more then one bi-clique. If we take an arbitrary row index k then there exist representative rows $\pi_{11}, \ldots, \pi_{1s}$, such that

$$A(\mathrm{k}, :) = \sum_{i=1}^{s} \alpha_{ki}\, A(\pi_{1i}, :), \tag{17}$$

with $\alpha_{ik} = 1$ whenever the row k is a member of the bi-clique i and $\alpha_{ik} = 0$ whenever k does not belong to bi-clique i. An analog equation can be found for the columns of A.

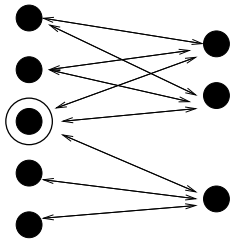

Figure 5: *A bipartite graph with "overlapping" bi-cliques.*

Simplex structure. The factors α in equation (17) are not convex combination factors. But if we reweight the rows of A with a diagonal (n, n)-matrix such that $A_1 = D_1^{-1}A$ is a stochastic matrix, then a short calculation as in equation (5) shows, that the rows of A_1 are convex combinations of the representative ones. A similar approach is used in [7]. However, in [7] Dhillon does not make use of transforming the simplex structure of the matrix A_1 to its singular vectors.

As we have shown, the rows of A_1 are convex combinations

$$A_1(\mathrm{k},:) = \sum_{i=1}^{s} \alpha_{ki}^{(1)}\, A_1(\pi_{1i},:)$$

of some representative ones with the row indices $\pi_{11}, \ldots, \pi_{1s}$. For the singular vector matrices X_1 and Y_1 of A_1 according to equation (4) and the singular values $\sigma_{11}, \ldots, \sigma_{1s} \neq 0$ we get

$$\begin{aligned} X_1(\mathrm{k},\mathrm{l}) &= \sigma_{1l}^{-1}\, A_1(\mathrm{k},:)\, Y_1(:,\mathrm{l}) \\ &= \sigma_{1l}^{-1}\, (\sum_{i=1}^{s} \alpha_{ki}^{(1)}\, A_1(\pi_{1i},:))\, Y_1(:,\mathrm{l}) \\ &= \sigma_{1l}^{-1} \sum_{i=1}^{s} \alpha_{ki}^{(1)}\, A_1(\pi_{1i},:)\, Y_1(:,\mathrm{l}) \\ &= \sum_{i=1}^{s} \alpha_{ki}^{(1)}\, X_1(\pi_{1i},\mathrm{l}). \end{aligned} \tag{18}$$

Since $\alpha_{ij}^{(1)}$ are convex combination factors, the data points $X_1(1, 1\!:\!\mathrm{s}), \ldots,$ $X_1(\mathrm{n}, 1\!:\!\mathrm{s})$ lie inside a simplex spanned by the vertices $X_1(\pi_{11}, 1\!:\!\mathrm{s}), \ldots,$ $X_1(\pi_{1s}, 1\!:\!\mathrm{s})$.

With an analog argument we reweight the columns of A such that $A_2^T = D_2 A^T$ is a stochastic matrix. The columns of A_2 are now convex combinations

$$A_2(:,\mathrm{k}) = \sum_{i=1}^{s} \alpha_{ki}^{(2)}\, A_2(:,\pi_{2i})$$

of some representative ones with the column indices $\pi_{21}, \ldots, \pi_{2s}$. For the singular vector matrices Y_2 and X_2 of A_2 according to equation (4) and the singular values $\sigma_{21}, \ldots, \sigma_{2s} \neq 0$ we get

$$
\begin{aligned}
Y_2(\mathrm{k},\mathrm{l}) &= \sigma_{2l}^{-1}\, A_2(:,\mathrm{k})^T\, X_2(:,\mathrm{l}) \\
&= \sigma_{2l}^{-1}\, (\sum_{i=1}^{s} \alpha_{ki}^{(2)}\, A_2(:,\pi_{2i})^T)\, X_2(:,\mathrm{l}) \\
&= \sigma_{2l}^{-1} \sum_{i=1}^{s} \alpha_{ki}^{(2)}\, A_2(:,\pi_{2i})^T\, X_2(:,\mathrm{l}) \\
&= \sum_{i=1}^{s} \alpha_{ki}^{(2)}\, Y_2(\pi_{2i},\mathrm{l}).
\end{aligned}
\tag{19}
$$

The data points $Y_2(1,1{:}\mathrm{s}),\ldots,Y_2(\mathrm{n},1{:}\mathrm{s})$ lie inside a simplex spanned by the vertices $Y_2(\pi_{21},1{:}\mathrm{s}),\ldots,Y_2(\pi_{1s},1{:}\mathrm{s})$.

Cluster algorithm. In both cases (18) and (19) we get a simplex structure of the corresponding singular vectors. We can compute the indices of the representative rows and columns via a routine searching the vertices of this simplex, see section 3.1. But in contrast to equation (9) the convex combination factors $\alpha_{kl}^{(i)}$ in the above equations have no stochastic interpretation anymore.

Therefore we use X_1 and Y_2 only to compute the indices of the representatives and then apply the linear transformations $\chi^{(1)} = X\mathcal{A}_1$ and $\chi^{(2)} = Y\mathcal{A}_2$ to the singular vectors of A with

$$
\mathcal{A}_1^{-1} = \begin{pmatrix} X(\pi_{11},1) & \ldots & X(\pi_{11},\mathrm{s}) \\ \vdots & & \vdots \\ X(\pi_{1s},1) & \ldots & X(\pi_{1s},\mathrm{s}) \end{pmatrix} \tag{20}
$$

and

$$
\mathcal{A}_2^{-1} = \begin{pmatrix} Y(\pi_{21},1) & \ldots & Y(\pi_{21},\mathrm{s}) \\ \vdots & & \vdots \\ Y(\pi_{2s},1) & \ldots & Y(\pi_{2s},\mathrm{s}) \end{pmatrix}. \tag{21}
$$

According to equation (17) and the fact that linear combinations of rows and columns of A transfer to its singular vectors X and Y, we get the solution $\chi^{(1)}(\mathrm{i},\mathrm{j}) = 1$ if the row index i belongs to bi-clique j, else $\chi^{(1)}(\mathrm{i},\mathrm{j}) = 0$. And analog, $\chi^{(2)}(\mathrm{i},\mathrm{j}) = 1$ if the column index i belongs to bi-clique j, else $\chi^{(2)}(\mathrm{i},\mathrm{j}) = 0$.

Example 2.6 *As an example we use a* $(6,5)$*-matrix* A *with perfect and overlapping block structure having* 2 *bi-cliques*

$$
A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}
$$

This matrix leads to the following row and column normalization:

$$
A_1 = \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix},\ A_2 = \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 \end{pmatrix}.
$$

The singular value decomposition of A gives 2 positive singular values $\sigma_1 = 3.3166, \sigma_2 = 2.4495$. Therefore $s = 2$. The singular vectors used for the index search algorithm are

$$X_1 = \begin{pmatrix} 0.0553 & 0.6511 \\ 0.0553 & 0.6511 \\ 0.2557 & 0.3519 \\ 0.5563 & -0.0970 \\ 0.5563 & -0.0970 \\ 0.5563 & -0.0970 \end{pmatrix}, Y_2 = \begin{pmatrix} -0.5438 & -0.1938 \\ -0.5438 & -0.1938 \\ -0.5438 & -0.1938 \\ -0.2374 & 0.6661 \\ -0.2374 & 0.6661 \end{pmatrix}.$$

If we interpret the rows of X_1 and Y_2 as points in the 2-dimensional space, then Y_2 only consists of two points. One can chose e.g $\pi_{21} = 1, \pi_{22} = 4$. X_1 consists of 3 points, where the point corresponding to the third row is a convex combination of the other two points e.g. $X_1(3,:) = 0.6 * X_1(1,:) + 0.4 * X_1(4,:)$, i.e. $\pi_{11} = 1, \pi_{12} = 4$.

With the equations (20) and (21) and the singular value decomposition of A one computes the matrices $\chi^{(1)}$ and $\chi^{(2)}$ as

$$\chi^{(1)} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \chi^{(2)} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix},$$

where $\chi^{(1)}$ gives the row clustering and $\chi^{(2)}$ the column clustering.

Incomplete blocks. In practice, the adjacency matrix A has not this perfect structure, some edges of the bipartite graph are missing and some may be inserted between different clusters. The result matrices $\chi^{(1)}$ and $\chi^{(2)}$ of the cluster algorithm are perturbed and may have elements $\chi^{(i)}(\mathrm{l}, \mathrm{j}) > 1$. Since this does not make sense, we normalize the rows of $\chi^{(i)}$ such that the maximum of each row is 1.

Again as in equation (15), the minimal element $\min \chi^{(i)} \leq 0$ can be used as an indicator for the deviation of the data from simplex structure and therefore as an a posteriori indicator for the quality of the solution.

Latent Semantic Indexing. A very similar method compared with the above algorithm is the Latent Semantic Indexing (LSI) routine using SVD, see e.g. Berry [2] pp. 7-15 or [3] pp. 103-122. Therein, also a set of representative indices for rows and columns of A is computed and bi-clustering is done according to the singular vector structure. But the LSI algorithm does not use the simplex structure of X_1 and Y_2. Whereas our report focuses on this special structure.

2.2.3 Singular values of A and the number of clusters

How can we compute the number of clusters a priori? A simple method is to count the number of singular values of A which are above a preset lower bound MINVALUE. The new question is, how can we fix MINVALUE a priori?

The following lemmas can be used for a heuristical determination of MINVALUE.

Lemma 2.7 *Let $\widetilde{A}$ be an (n,m)-adjacency matrix with s different complete blocks and $d \in \mathbf{N}$. Let the (n,m)-adjacency matrix A differ from $\widetilde{A}$ at most in d elements per row and column. Then we have $\sigma_{s+1} \leq d$ for the singular values of A.*

Proof: From the construction of $\widetilde{A}$ we get $\mathrm{rank}(\widetilde{A}) = s$. Since A differs from $\widetilde{A}$ in at most d elements per row and column, for each row of matrix $E = (A-\widetilde{A})^T(A-\widetilde{A})$ the sum of the absolute values of the row elements is less than or equal to d^2. From a Gershgorin estimate of eigenvalues of E the spectral norm of the difference is therefore $\|A-\widetilde{A}\|_2 \leq d$. For real (n,m)-matrices B, we further get

$$d \geq \|A-\widetilde{A}\|_2 \geq \min_{rank(B)=s} \|A-B\|_2 = \sigma_{s+1},$$

where the last equality is shown in [12](Theorem 2.5.3). This completes the proof. $\square$

Lemma 2.8 *Let again $\widetilde{A}$ be an (n,m)-adjacency matrix with s different complete blocks. Let the (n',m')-matrix $\widetilde{A}'$ be constructed from $\widetilde{A}$ by eleminating all rows and columns, where the blocks in $\widetilde{A}$ overlap. Then*

$$\sigma_s(\widetilde{A}) \geq \sqrt{c_{min} r_{min}},$$

where c_{min} and r_{min} are the column size resp. row size of the minimal block in $\widetilde{A}'$. Furthermore, let $d \in \mathbf{N}$ and let the (n,m)-adjacency matrix A differ from $\widetilde{A}$ in at most d elements per row and column. Then

$$\sigma_s(A) \geq \sqrt{c_{min} r_{min}} - d.$$

Proof: It is known, that adding columns to $\widetilde{A}$ may only increase the singular value σ_s [12] (Corollary 8.6.3). Therefore eleminating columns may only decrease σ_s. Since the transpose of $\widetilde{A}$ has the same singular values, there is an analog argument for the rows, i.e. the singular value $\sigma_s(\widetilde{A}')$ is not higher than that one of $\widetilde{A}$.

Let $S' \subset \mathbb{R}^{m'}$ and $\mathbb{1}_1, \ldots, \mathbb{1}_s \in \mathbb{R}^{m'}$ be the (orthogonal) characteristic vectors of the column indices of the blocks $1, \ldots, s$ in $\widetilde{A}'$. For computation of σ_s we can use the fact [12] (Theorem 8.6.1) that

$$\sigma_s(\widetilde{A}) = \max_{\dim(S)=s} \min_{0 \neq x \in S} \frac{\|\widetilde{A}x\|_2}{\|x\|_2},$$

where S is a subspace of $\mathbb{R}^m$. We get the following estimate

$$\begin{aligned} \sigma_s(\widetilde{A}) &\geq \sigma_s(\widetilde{A}') \\ &= \max_{\dim(S')=s} \min_{0 \neq x \in S'} \frac{\|\widetilde{A}'x\|_2}{\|x\|_2} \\ &\geq \min_{0 \neq x \in \mathrm{span}(\mathbb{1}_1, \ldots, \mathbb{1}_s)} \frac{\|\widetilde{A}'x\|_2}{\|x\|_2} \end{aligned}$$

$$= \min_{\theta_1,\ldots,\theta_s \in \mathbb{R}} \sqrt{\frac{\sum_{i=1}^{s} \theta_i r_i c_i^2}{\sum_{i=1}^{s} \theta_i c_i}}$$

$$= \sqrt{c_{min} r_{min}},$$

where c_i and r_i is the column size resp. row size of block i in $\widetilde{A}'$. The second inequality of the lemma can be shown as follows. From [12] (Corollary 8.6.2) we know that

$$|\sigma_s(A) - \sigma_s(\widetilde{A})| \leq \|A - \widetilde{A}\|_2$$

and from the proof of Lemma 2.7 that $\|A - \widetilde{A}\|_2 \leq d$. $\square$

These two lemmas have shown, that we can choose e.g. MINVALUE= d for the computation of the number of biclusters, if the expected number d of maximal deviation from perfect block structure has an upper bound $d < 0.5 * \sqrt{c_{min} r_{min}}$.

Example 2.9 *If we assume, that A has non-overlapping blocks with at least size* 3×2 *and that A differs from perfect block structure only in* 1 *element per row and column (like in Example 3.1), then* $1 \leq$MINVALUE$< \sqrt{6} - 1$ *are possible presettings.*

Another possibility to compute the number of clusters with the Frobenius norm instead of the 2-norm can be derived from equation (16).

2.3 Less input vectors lead to a wrong data classification

In spectral k-partitioning methods often a smaller number of eigenvectors or singular vectors, $s < k$, is used as input data, e.g. $s = \lceil \log_2 k \rceil$ in [7, 9]. If the simplex-like data is well separated as in Fig. 4 then a projection of the point set into a low dimensional subspace may be successful. But by means of a counter-example we show that the projection into lower subspaces may cause failures in classification.

E.g. with transition matrices in molecular dynamics we do not get data spread around the vertices of a simplex. We have "full" simplices, see Fig. 1. A projection of this figure into a one-dimensional subspace conceils the difference between transition states ($A \leftrightarrow B$) and states corresponding to another representative (C), see Fig. 6.

That this situation is the rule and not an exception can be seen by equation (9), because therein the convex combination factors determining the simplex structure are equal to certain transition probabilities. Since, in general, transition states occur in molecular dynamics, there are also points at the facets of the corresponding simplex.

Therefore especially successive algorithms working only on the eigenvector corresponding to the 2nd largest eigenvalue may fail. A famous algorithm of this type is the Fiedler Cut for the Laplacian of a graph, see e.g. [8].

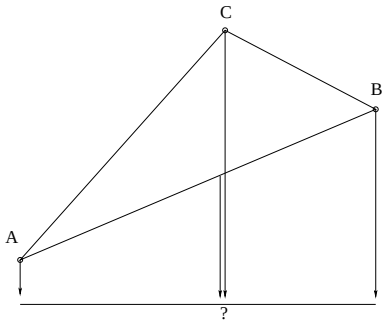

Figure 6: *Projection leads to a wrong data classification. No difference between transition states and representative states.*

3 Cluster algorithms

3.1 Search routine for the vertices of a simplex

For a detailed description of this algorithm see [6]. The input of this algorithm is an eigenvector or a singular vector matrix, and the number of vertices that should be produced. The output is a vector of the indices of the vertices. All algorithms are written in MatLab style.

```
function [ind]=indsearch(Evs, NoOfClus)
    C=Evs(:, 1:NoOfClus);
    OrthoSys=C;
    maxdist=0.0;
    ind=zeros(NoOfClus,1);

    % First vertex having maximal norm
    for i=1:size(Evs,1)
       if norm(C(i,:)) > maxdist
                maxdist = norm(C(i,:));
                ind(1)=i;
       end;
    end;
    for i=1:size(Evs,1)
        OrthoSys(i,:)=OrthoSys(i,:)-C(ind(1),:);
    end;

    % Further vertices via Gram-Schmidt orthogonalization
    for k = 2:NoOfClus
        maxdist=0.0;
        temp=OrthoSys(ind(k-1),:);
        for i=1:size(Evs,1)
            OrthoSys(i,:)=OrthoSys(i,:)-(temp*transpose(OrthoSys(i,:)))*temp;
            distt=norm(OrthoSys(i,:));
            if distt > maxdist
                maxdist = distt;
                ind(k)=i;
            end;
        end;
        OrthoSys=OrthoSys./norm(OrthoSys(ind(k),:));
    end;
```

3.2 Almost invariant sets and biclustering

For a better understanding we present two simple cluster algorithms for the above problems: finding almost invariant sets, and biclustering.

In both algorithms the number of clusters may be unknown a priori. A suggestion for determining the number of clusters is to count the eigenvalues or singular values which are above a preset lower bound MINVALUE.

There are more sofisticated methods [6, 4] to identify the number of clusters. Furthermore, for conformation analysis there has to be a post-processing routine to compute a clustering χ with only positive entries [6], which leads to an optimization problem.

The presented algorithms are simple to implement and gives the main idea for clustering as a linear transformation of eigenvectors or singular vectors.

3.2.1 Almost invariant sets

The input parameters of the **almostinvar** algorithm are the transition matrix of a reversible Markov chain and, optionally, the number of clusters. The outputs are the matrix χ and a vector of the s highest eigenvalues. If you want to find out if the cluster method works well, the smallest entry of χ should be near 0 [19, 20].

```
function [Chi, Lambda]=almostinvar(matrix,NoOfClus)
    % solve the eigenvalue problem
    % sort the eigenvectors (NoOfClus may be unknown)
    if nargin == 1
        [X Lambda]=eig(eye(size(matrix,1))-matrix);
        [Lambda sind]=sort(diag(Lambda)); Lambda=1-Lambda;
        X=X(:,sind);
    else
        [X Lambda]=eigs(matrix+eye(size(matrix,1)),NoOfClus);
        [Lambda sind]=sort(diag(-Lambda)); Lambda=-Lambda-1;
        X=X(:,sind);
    end

    % determine the number of clusters
    if nargin==1
        NoOfClus=0;
        for i=1:size(Lambda,1)
            if Lambda(i) > MINVALUE
                NoOfClus = NoOfClus +1;
            end;
        end;
        X=X(:,1:NoOfClus);
        Lambda=Lambda(1:NoOfClus);
    end;

    % linear transformation of eigenvectors
    ind = indsearch(X, NoOfClus);
    Chi=X*inv(X(ind,:));
```

3.2.2 Clustering of bipartite graphs

The input parameters of the **bipartclus** algorithm are the adjacency matrix of a bipartite graph and, optionally, the number of clusters. The output are the matrices $\chi^{(1)}$ and $\chi^{(2)}$ and a vector of the s highest singular values.

```
function [Chi1, Chi2, S] = bipartclus(matrix,NoOfClus)
    % singular value decomposition of matrix
    % (number of clusters may be unknown)
    if nargin==1
        S=svd(matrix);
    else
        [X S Y]= svds(matrix,NoOfClus);
    end;

    % determination of NoOfClus
    if nargin==1
        NoOfClus=0;
        for i=1:size(S,1)
            if S(i) > MINVALUE
                NoOfClus = NoOfClus +1;
            end;
        end;
        [X S Y]=svds(matrix, NoOfClus);
    end;
    S = diag(S);

    % special matrices for index searching routine
    matrixL = matrix*inv(diag(sum(matrix)));
    matrixR = inv(diag(sum(transpose(matrix))))*matrix;
    [XL SL YL]=svds(matrixL, NoOfClus);
    [XR SR XR]=svds(matrixR, NoOfClus);

    % clustering is a linear transformation
    % followed by a special normalization
    ind = indsearch(XR,NoOfClus);
    X = X(:,1:NoOfClus);
    Chi1 = X * inv(X(ind,:));
    Chi1 = inv(diag(max(transpose(Chi1))))*Chi1;
    ind = indsearch(YL,NoOfClus);
    Y = Y(:,1:NoOfClus);
    Chi2 = Y * inv(Y(ind,:));
    Chi2 = inv(diag(max(transpose(Chi2))))*Chi2;
```

Example 3.1 *We present an example for a biclustering of a* $(12, 9)$*-adjacency matrix* A *with the bipartclus algorithm from above and* $\mathrm{MINVALUE} = 1.0$.

```
>> A

A =

     1     0     1     1     0     1     1     0     1
     0     1     0     0     1     0     0     1     0
     1     0     0     1     0     1     0     0     0
     0     0     1     0     0     0     0     0     1
     0     0     1     0     0     0     1     0     1
     0     1     1     0     1     0     1     1     1
     1     0     0     1     0     0     0     0     0
     0     0     1     0     0     0     1     0     1
     1     0     0     1     0     1     0     0     0
     1     0     0     1     0     1     0     0     0
     0     0     1     0     0     0     1     0     0
     0     1     0     0     1     0     0     1     0

>> [Chi1,Chi2]=bipartclus(A)

Chi1 =

    0.0031    1.0000    0.9666
```

```
    1.0000    0.0000   -0.0000
   -0.0029    0.0124    1.0000
         0    1.0000    0.0000
    0.0060    1.0000    0.0057
    0.6901    1.0000    0.0057
   -0.0000   -0.0000    1.0000
    0.0060    1.0000    0.0057
   -0.0029    0.0124    1.0000
   -0.0029    0.0124    1.0000
    0.0000    1.0000    0.0000
    1.0000   -0.0000    0.0000


Chi2 =

    1.0000    0.0134   -0.0380
   -0.0000    1.0000    0.0000
   -0.0366   -0.0492    1.0000
    1.0000    0.0134   -0.0380
   -0.0000    1.0000    0.0000
    1.0000    0.0000    0.0000
    0.0000   -0.0000    1.0000
   -0.0000    1.0000         0
   -0.0000         0    1.0000
```

The result is $s = 3$. $\chi^{(1)}$ and $\chi^{(2)}$ lead to a clustering of row and column indices, the permutation of indices in A according to this clustering shows the three overlapping, incomplete blocks in A.

```
>> ind1=[2,12,    6,4,5,8,11,1,    3,7,9,10]; ind2=[1,4,6,   2,5,8,   3,7,9];
>> A(ind1, ind2)

ans =

     0     0     0     1     1     1     0     0     0
     0     0     0     1     1     1     0     0     0
     0     0     0     1     1     1     1     1     1
     0     0     0     0     0     0     1     0     1
     0     0     0     0     0     0     1     1     1
     0     0     0     0     0     0     1     1     1
     0     0     0     0     0     0     1     1     0
     1     1     1     0     0     0     1     1     1
     1     1     1     0     0     0     0     0     0
     1     1     0     0     0     0     0     0     0
     1     1     1     0     0     0     0     0     0
     1     1     1     0     0     0     0     0     0
```

4 Conclusion

In this paper we described a spectral graph partitioning method for dynamical cluster problems and for bipartite graphs. Despite the fact that neither spectral clustering nor the "soft concept", i.e. fuzzy clustering, are new, there are several new constributions of this paper:

- First, we proved that the Perron eigenvector data turning up in molecular dynamics really has simplex structure, which had not been shown in our last papers [20, 19, 6].

- Second, we have shown that the idea of using a simplex structure in cluster algorithms can be extended to adjacency matrices of bipartite graphs, which leads to routines that are easy to implement, because clustering becomes a linear mapping.
- Third, we have shown that the heuristics of using less singular or eigenvectors than the number of clusters in spectral graph partitioning may lead to wrong data classification.

We have given some simple examples for the presented cluster algorithms. However, it is left to show, that these routines work well for larger problems, too. The algorithms have proved to be suitable in conformational dynamics and analysis [4]. Here, they give additional information about transition states of the examined biomolecules.

Acknowledgements. The author wants to thank Daniel Baum for carefully reading this paper and helpful discussions. I gratefully acknowledge the valuable contributions concerning cluster analysis of the Algorithmics Group of the Max Planck Institute for Molecular Genetics, especially Alexander Schliep and Wasinee Rungsarityotin.

References

[1] Andrew Y. Ng, Michael I. Jordan and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14*, 2002.

[2] M.W. Berry, editor. *Computational Information Retrieval*. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

[3] M.W. Berry, editor. *Survey of Text Mining,* Clustering, Classification and Retrieval. Springer, 2004.

[4] F. Cordes, M. Weber, and J. Schmidt-Ehrenberg. Metastable conformations via successive perron-cluster cluster analysis of dihedrals. Technical Report ZIB 02-40, Zuse Institute Berlin, 2002.

[5] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Lin. Alg. Appl.*, 315:39–59, 2000.

[6] P. Deuflhard and M. Weber. Robust Perron Cluster Analysis in Conformation Dynamics. Technical Report ZIB 03-19, Zuse Institute Berlin, 2003.

[7] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. *Dept. of Computer Sciences, University of Texas, Austin, TX78712*, 2001.

[8] Miroslav Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech. Math. J.*, 25(100):619–633, 1975.

[9] G. Froyland and M. Dellnitz. Detecting and locating almost-invariant set and cycles. *Dept. of Mathematics and Statistics, University Paderborn*, 2001.

[10] Gary Froyland. Approximating physical invariant measures of mixing dynamical systems in higher dimensions. *Nonlinear Anal.*, 32(7):831–360, 1998.

[11] Gary Froyland and Michael Dellnitz. Statistically optimal almost–invariant sets: Efficient detection and adaptive resolution. Preliminary work, 2001.

[12] G.H. Golub and C.F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[13] Heather L. Gordon and Rajmund L. Somorjai. Fuzzy cluster analysis of molecular dynamics trajectories. *Proteins*, 14:249–264, 1992.

[14] J.C. Bezdek and S.K. Pal. Fuzzy models for pattern recognition. *IEEE Press, New York*, 1992.

[15] Michael W. Berry and Muray Browne. *Understanding Search Engines*. Society for Industrial and Applied mathematics, 1999.

[16] R.B. Bapat and T.E.S. Raghavan. *Nonegative Matrices and Applications*. Cambridge University Press, 1997.

[17] Ch. Schütte. *Conformational Dynamics: Modelling, Theory, Algorithm, and Application to Biomolecules*. Habilitation Thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin, 1998.

[18] Sepandar D. Kamvar, Dan Klein, Christopher D. Manning. Spectral learning. *http://www.stanford.edu/ sdkamavar/papers/spectral.ps*, 2003.

[19] M. Weber. Improved Perron Cluster Analysis. Technical Report ZIB 03-04, Zuse Institute Berlin, 2003.

[20] M. Weber and T. Galliat. Characterization of transition states in conformational dynamics using Fuzzy sets. Technical Report Report 02–12, Konrad–Zuse–Zentrum (ZIB), Berlin, March 2002.