Journal of Machine Learning Research 1 () Submitted ; Published

An Exact Relaxation of the Hard Assignment Problem in Clustering

Morten Mørup MM@IMM.DTU.DK
Lars Kai Hansen LKH@IMM.DTU.DK
DTU Informatics
Richard Petersens Plads, bld. 321
2800 Kgs. Lyngby

Editor: n/a

Abstract

Continuous relaxation of hard assignment clustering problems can lead to better solutions than greedy iterative refinement algorithms. However, the validity of existing relaxations is contingent on problem specific *fuzzy* parameters that quantify the level of similarity between the original combinatorial problem and the relaxed continuous domain problem. Equivalence of solutions obtained from the relaxation and the hard assignment is guaranteed only in the limit of vanishing 'fuzzyness'. This paper derives a new exact relaxation without such a fuzzy parameter which is applicable for a wide range of clustering problems such as the K-means objective and pairwise clustering as well as graph partition problems, e.g., for community detection in complex networks. In particular we show that a relaxation to the simplex can be given for which the extreme solutions are stable hard assignment solutions and vice versa. Based on the new relaxation we derive the SR-*clustering* algorithm that has the same complexity as traditional greedy iterative refinement algorithms but leading to significantly better partitions of the data. A Matlab implementation of the SR-*clustering* algorithm is available for download.

Keywords: Clustering, Complex Networks, Community Detection, Simplicial Relaxation, SR-clustering.

1. Introduction

Clustering - the problem of grouping a set of objects according to a given similarity measure - is at the core of human cognition and therefore unrivalled as the most important, challenging, and well studied problem in unsupervised machine learning. While we as humans can produce a *hard* clustering, i.e., assign all relevant objects to one and only one group, the approach towards such hard clustering can involve an intermediate soft assignment state in which multiple group memberships are hypothesized for given objects (Kemp and Tenenbaum, 2008). In this paper we discuss the interplay between hard and soft assignments and show that by proper choice of representation soft and hard assignments can coexist for the clustering problem.

From an engineering point of view clustering is important for the analysis of data both when represented as points in vector spaces or more abstractly as graphs. In fact, there is a recent surge of interest in clustering of graph structured data including the worldwide web, metabolic networks, food webs, neural networks, communication and distribution networks and social networks (Wasserman and Faust, 1994; Albert and Barabási, 2002; Newman, 2003, 2006). An aim of this research is to identify communities of densely connected groups of vertices with relatively less connections running between groups (Newman, 2006). For point clustering of data in $\mathbb{R}^n$ the aim is to form

groups of neighboring objects. Each group consists of objects that are closer than objects in the other groups (Berkhin, 2002). Such vector space clustering of high-dimensional data is an important problem in many fields of science ranging from text information retrieval (Larsen and Aone, 1999), bio-informatics (Eisen et al., 1998), and marketing (Punj and Stewart, 1983).

We note that since a graph can be embedded in $\mathbb{R}^n$ and data in $\mathbb{R}^n$ can be represented as a weighted graph based on pairwise distances – graph clustering and point clustering are highly interrelated problems.

By definition grouping is by *hard assignment* of objects to classes and the many hard clustering problems have been shown to be NP-complete (Megiddo and Supowit, 1984). Therefore a large number of heuristics have been proposed for the hard clustering problem, including the widely used iterative refinement algorithm due to Lloyd which is often simply referred to as the K-*means* algorithm (Lloyd, 1982; Hartigan and , 1979).

An alternative strategy is to map the hard clustering problem on to a continuous domain and apply efficient solvers available for continuous optimization. In fact, it has been shown that such *relaxations* can lead to powerful methods for the clustering problem (Hofmann and Buhmann, 1997; Pal and Bezdek, 1995; Hathaway and Bezdek, 1988; Slonim et al., 2005). A main concern with the available relaxations is that they rely on a *fuzzy* parameter which controls the approximation of the original hard clustering problem. For example, in the papers of (Hofmann and Buhmann, 1997; Slonim et al., 2005; Lehmann and Hansen, 2007), a Lagrange multiplier is introduced which plays the role of an equivalent *temperature* and only in the low temperature limit are soft assignments equivalent to hard clustering. The quality of the solutions obtained depends critically on the way temperature is annealed from high to low and this path may even involve phase transitions.

Here we will aim for an exact relaxation, i.e., a relaxation that does not rely on a temperature parameter. We will establish a quite generic formalism which enables us to analyze a wide range of clustering problems including the K-means objective and pairwise clustering as well as community detection in complex network based on Hamiltonian or Modularity optimization (Fu and Anderson, 1986; Reichardt and Bornholdt, 2004, 2006; Newman and Girvan, 2004; Newman, 2006). The main contributions of this paper are:

i We demonstrate that the discrete combinatorial constraints of hard assignments can be replaced exactly by convex non-negativity and linear constraints to the simplex for both point clustering in vector spaces and graph clustering. The equivalence is obtained by showing that locally optimal configurations of the cost function in the continuous domain are equivalent to valid hard assignments, and that these hard assignments are in fact stable against single node changes in assignment, in the sense that any such change leads to a higher cost.

ii Based on the exact relaxation to the simplex we produce new efficient algorithms for clustering that has the same complexity as traditional greedy iterative refinement algorithms for clustering but results in significantly better partitions of the data.

2. Clustering approaches

It is out of this paper's scope to give a full summary of clustering approaches, instead we refer to the comprehensive survey given in (Berkhin, 2002). In general, clustering methods can be divided into hierarchical and partitional approaches. The hierarchical approaches are again split into divisive or agglomerative approaches. In the divisive and agglomerative approaches the observations initially

belong to respectively one or J clusters and are recursively split respectively joined to form smaller or larger clusters. As such the graph max and min cut algorithms based on the graph Laplacian falls into this framework (Fiedler, 1973; Pothen. et al., 1990) as does the linkage approaches forming dendrograms based on pairwise similarity measures. The partitional approaches on the other hand, attempts to estimate K clusters at once by inferring a $K \times N$ assignment matrix $\boldsymbol{S}$ with components $s_{k,j}$.

We note that a graph/network is an ordered pair $G = (V, E)$ comprising a set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V. We presently consider undirected graphs which are often represented by the sparse symmetric adjacency matrix $\boldsymbol{A}^{J \times J}$ such that $a_{i,j} = 1$ if there is an edge between vertex i and j and zero otherwise. Among partitional approaches, perhaps the most well-known frameworks for community detection in undirected graphs/networks is based on maximizing graph clustering (GC) objectives of the following form (Fu and Anderson, 1986; Reichardt and Bornholdt, 2004, 2006; Newman and Girvan, 2004; Newman, 2006)

$$\begin{aligned} P_{\text{BC}}^{\text{GC}}: \quad & \arg\max_{\boldsymbol{S}} \operatorname{trace}[\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top] \\ s.t. \quad & s_{k,j} \in \{0,1\} \; and \; \sum_k s_{k,j} = 1 \end{aligned}$$

with $\boldsymbol{B}$ being a $J \times J$ symmetric matrix. The constraints $s_{k,j} \in \{0,1\}$ and $\sum_k s_{k,j} = 1$ enforce $\boldsymbol{S}$ to be optimized under binary combinatorial (BC) constraints (i.e. optimized such that there is only one non-zero element of each column of $\boldsymbol{s}_j$ with value 1) enforcing $\boldsymbol{S}$ to form a clustering assignment matrix. We denote this problem $P_{\text{BC}}^{\text{GC}}$. In particular, the above constraint enforces that the cardinality of each column of $\boldsymbol{S}$ be one. Since $\boldsymbol{B}$ is generally not positive semi-definite (PSD) the problem is non-convex. Furthermore, the constraints on $\boldsymbol{S}$ indicate that the optimization problem has a non-continuous combinatorial nature. Traditionally, $\boldsymbol{B} = \boldsymbol{A} - \boldsymbol{P}$ where $\boldsymbol{P}$ constitutes an inherent null-hypothesis such that regions more connected than this theoretical hypothesis level will be favored to cluster together. As such, setting $\boldsymbol{P} = \rho \mathbf{1}\mathbf{1}^\top$ where $\rho = \sum_{i,j} a_{i,j}/J^2 = \|\boldsymbol{A}\|/J^2$ is the average density of the graph (Fu and Anderson, 1986; Reichardt and Bornholdt, 2004) the above objective favors clustering vertices more densely connected than average which follows ones natural notion that clusters constitute regions of above average link-density. By setting $\boldsymbol{P} = \frac{1}{\sum_j k_j} \mathbf{k}\mathbf{k}^\top$ where $k_i = \sum_j a_{i,j}$ the Modularity objective (Newman and Girvan, 2004; Newman, 2006; Reichardt and Bornholdt, 2004) is derived. Under this null-hypothesis, $P_{\text{HARD}}^{\text{GC}}$ measures the deviation of the fraction of edges within communities from the expected fraction of such edges based on their degree distribution. Thus, communities should link more than expected in terms of their degree product. A benefit of these inherent null-hypothesis are that they work as a contrast for clustering and automatically selects for an optimal number of clusters K. Finally, we note that if $\boldsymbol{P} = diag(\boldsymbol{k})$ then $\boldsymbol{B}$ corresponds to the well studied graph Laplacian (Fiedler, 1973) enforcing equilibrium between the out and in-flow of each vertex in the graph.

For point clustering in $\mathbb{R}^n$ the perhaps most well known clustering objective is the K-means objective given by $\|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{S}\|_F^2$ where $\boldsymbol{S}$ is constrained as in $P_{\text{HARD}}^{\text{GC}}$ to be a clustering assignment matrix and $\boldsymbol{c}_k$ the k^{th} cluster centroid which is given as the average of the data points belonging to the cluster, i.e. $\boldsymbol{c}_k = \frac{1}{\sum_j s_{k,j}} \sum_j \boldsymbol{x}_j s_{k,j}$. For data characterized by a pairwise symmetric similarity matrix $\boldsymbol{B}$ where $b_{i,j}$ denotes the similarity between data point i and j the following clustering objective is commonly maximized (Buhmann and Hofmann, 1994; Fischer et al., 2001) forming the

pairwise clustering (PC) objective

$$P_{\text{BC}}^{\text{PC}}: \quad \arg\max_{\boldsymbol{S}} \text{trace}[\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top \boldsymbol{D}]$$
$$s.t. \quad s_{k,j} \in \{0,1\} \; and \; \sum_k s_{k,j} = 1.$$

where $\boldsymbol{D}$ is a diagonal matrix given by $d_{k,k'} = \delta_{k,k'}(\sum_j s_{k,j})^{-1}$. The K-means objective is a special case of this approach formed by measuring similarity in terms of inner products between observations, i.e. $\boldsymbol{B} = \boldsymbol{X}^\top \boldsymbol{X}$, for details on this consult theorem 7. Notice, rather than incorporating a null-hypothesis as in $P_{\text{BC}}^{\text{GC}}$ through the formulation of $\boldsymbol{B}$, $\boldsymbol{D}$ balances the clusters such that the contribution of each cluster to the objective is weighted by the number of observations in the cluster. Hence this framework is, apart from various types of similarity measures, also useful for weighted graphs where a reasonable null-hypothesis can be difficult to derive. The objective is non-convex even if $\boldsymbol{B}$ is PSD due to the introduction of $\boldsymbol{D}$. Furthermore, the constraints on $\boldsymbol{S}$ again indicate that the optimization problem has a binary combinatorial nature.

To solve for these binary combinatorial constraints the well known greedy iterative refinement algorithms also known as Lloyds algorithm or simply as K-means were derived for the K-means objective (Lloyd, 1982; Hartigan and , 1979). Given an initial set of cluster centroids the iterative refinement algorithm assigns each observation to the centroid that is closest (for K-means this is in terms of euclidian norm) and updates the centroids to the center of their assigned data points. The algorithm converges when no assignments change. Within combinatorial optimization simulated annealing, deterministic annealing and mean field methods are widely used (Hofmann and Buhmann, 1997; Slonim et al., 2005; Lehmann and Hansen, 2007). Common to these approaches is that a temperature parameter T is controlled such that at high temperatures the clustering objective is smoothed and assignments can be made more or less arbitrarily while for $T \to 0$ the original clustering problem is recovered. In fuzzy clustering the binary combinatorial assignments are relaxed to soft clustering membership (Pal and Bezdek, 1995; Hathaway and Bezdek, 1988) and the extend of fuzzyness controlled by a parameter. This is related to the expectation maximization (EM) algorithm where each centroid is described by its location (mean) and a distribution around this mean (variance). Data points are then assigned a probability to each centroid according to the distributions imposed (these distributions are most often based on Gaussian mixtures but alternative distribution have also been proposed, see (Banerjee et al., 2005)). A major benefit of the EM approach being that the clustering problem is stated in a formal Bayesian framework. The parameter controlling the overlap in fuzzy clustering as well as the variance of the distributions in the EM approach has a similar interpretation as the temperature in the annealing approaches and can as such also potentially be annealed to recover binary combinatorial assignments. Methods from binary programming has also been proposed to solve clustering problems based on branch and bound algorithms to handle the combinatorial clustering constraints (du Merle et al., 2000). Finally, the clustering problems have been relaxed to convex objectives in the so-called spectral clustering approaches (Bach and Jordan, 2004; Ng et al., 2001; Peng and Wei, 2007) that can be solved by semi-definite programming. Here the combinatorial constraints on $\boldsymbol{S}$ are relaxed such that $\boldsymbol{S}$ is orthonormal, i.e. $\boldsymbol{S}\boldsymbol{S}^\top = \boldsymbol{I}$ while the similarity matrix $\boldsymbol{B}$ is turned PSD (i.e. for graphs by considering the graph Laplacian). The benefit being that the optimization problem becomes convex (for graph clustering this holds when analyzing the graph Laplacian which is a diagonal dominant matrix) and solutions can be derived through the singular value decompositions (SVD) (Golub and Van Loan, 1996) while the eigenvectors give

insight in the hard clustering problem. As such, the second smallest singular value of the graph Laplacian is related to the optimal cut in the graph. For an excellent review of spectral methods see also (von Luxburg, 2007).

To summarize, the main challenge when optimizing for the clustering problem is the handling of the binary combinatorial constraints without resorting to greedy approaches. This has previously been achieved by relaxations that are either not exact or dependent on some problem specific annealing parameter in order to recover the original binary combinatorial constraints.

3. Soft Clustering on the simplex

We will presently demonstrate how the binary combinatorial (BC) nature of the above clustering problems can be circumvented completely by relaxing the clustering problems $P_{\text{BC}}^{\text{GC}}$ and $P_{\text{BC}}^{\text{PC}}$ to continuous optimization problems over the simplex (Δ^n) such that $\boldsymbol{S}$ admits soft clustering.

Definition 1 *The simplex will presently refer to the standard n-simplex or unit n-simplex given by*

$$\Delta^n = \{\mathbf{s} \in \mathbb{R}^{n+1} | \sum_{k=1}^{n+1} s_k = 1 \text{ and } s_k \geq 0 \forall k\}.$$

Although this can be considered a relaxation between the original problem and the spectral clustering approaches we will demonstrate that this relaxation turns out to be an exact proxy for the original binary combinatorial clustering problem. The following definition formalizes the simplicial relaxation,

Definition 2 *A* simplicial relaxation*, (*SR*), is given by relaxing the binary combinatorial (*BC*) constraints to Δ^{K-1}, i.e. relaxing the constraints $s_{k,j} \in \{0,1\}$ and $\sum_k s_{k,j} = 1$ to $s_{k,j} \geq 0$ and $\sum_k s_{k,j} = 1$.*

Thus, contrary to BC there always exists a continuous path from any solution to another for the SR. As such, the graph and pairwise clustering problems $P_{\text{BC}}^{\text{GC}}$ and $P_{\text{BC}}^{\text{PC}}$ have the following simplicial relaxations

$$\begin{aligned} P_{\text{SR}}^{\text{GC}}: \quad & \arg\max_{\boldsymbol{S}} \operatorname{trace}[\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top] \\ P_{\text{SR}}^{\text{PC}}: \quad & \arg\max_{\boldsymbol{S}} \operatorname{trace}[\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top\boldsymbol{D}] \\ s.t. \quad & s_{k,j} \geq 0 \; and \; \sum_k s_{k,j} = 1. \end{aligned}$$

Notice, the simplicial relaxation is similar to previous soft clustering approaches in that nodes/observations can potentially belong to several clusters, however, contrary to previous approaches we will demonstrate that the problems formed by the above simplicial relaxation is in fact equivalent to the original binary combinatorial problems. Thus, we first formalize what we mean by two optimization problems being equivalent.

Definition 3 *Two optimization problems P_1 and P_2 are* equivalent*, i.e. $P_1 \sim P_2$, if a (local) optimum of P_1 is also a (local) optimum of P_2 and a (local) optimum of P_2 is a (local) optimum of P_1.*

Hence, two problems are equivalent when a solution of one problem is also a solution of the other problem and vice versa. The following two definition clarifies what is meant by (local) optimum under BC and SR constraints.

Definition 4 *A (local) optimum of an optimization problem under* BC *constraints is given by a configuration where any change of assignment in one column of $\boldsymbol{S}$ decrease the objective function relative to the current assignment.*

This definition of an optimal configuration is also referred to as 1-spin-stable (Waclaw and Burda, 2008). According to the above definition any solution obtained under BC by the greedy iterative refinement algorithm such as Lloyd's K-means algorithm leads to a local optimal solution, i.e a solution in which no single change of assignment is better than the current assignment.

Definition 5 *A (local) optimum of an optimization problem under* SR *constraints is given by a configuration where any infinitesimal change of a column of $\boldsymbol{S}$ over the simplex decrease the objective function relative to the current position on the simplex.*

Notice in particular how the above definition of a (local) optimum for SR requires that the Karush-Kuhn-Tucker (KKT) conditions for the constrained optimization problem have to be satisfied.

Theorem 6

$$P_{\text{BC}}^{\text{GC}} \sim P_{\text{SR}}^{\text{GC}} \tag{1}$$

Proof Without loss of generality we can assume $b_{i,i} = \delta > 0$ as the contribution of the diagonal elements $b_{i,i}$ are independent of $\boldsymbol{S}$ when $\boldsymbol{S}$ is a binary cluster indicator matrix. To enforce the equality constraint and the non-negativity constraint in $P_{\text{SR}}^{\text{GC}}$ we optimize the following objective

$$\begin{aligned} \arg\max_{\boldsymbol{S}} \mathfrak{L}(\boldsymbol{S}) &= \text{trace}[\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top] \\ &+ \sum_j \lambda_j (\sum_k \boldsymbol{s}_{k,j} - 1) + \sum_k \theta_{k,j} s_{k,j} \end{aligned}$$

Where λ_j is the lagrange multiplier for the j^{th} equality constraint and $\theta_{k,j}$ the lagrange multiplier for the non-negativity constraint imposed on $s_{k,j}$. The optimal solution has to satisfy the following Karush-Kuhn-Tucker (KKT) conditions at a (local) optimum of $P_{\text{SR}}^{\text{GC}}$

$$\begin{aligned} g_{k,j} + \lambda_j + \theta_{k,j} &= 0 \\ \theta_{k,j} &\geq 0 \\ \theta_{k,j} s_{k,j} &= 0 \end{aligned}$$

where $g_{k,j} = 2\sum_i s_{k,i} b_{i,j}$ is the gradient of the objective function and λ_j and $\theta_{k,j}$ the lagrange multipliers of the equality and non-negativity constraint of the lagrange function $\mathfrak{L}$.

We first prove that a (local) optimum of $P_{\text{BC}}^{\text{GC}}$ is a (local) optimum of $P_{\text{SR}}^{\text{GC}}$. Consider a (local) optimum $\boldsymbol{S}^*$ to $P_{\text{BC}}^{\text{GC}}$. For $s^*_{k',j} = 1$ we must have that $g_{k',j} \geq g_{k,j} \quad \forall k \neq k'$ otherwise a change of assignment in a column of $\boldsymbol{S}^*$ would be more optimal. Setting the lagrange multipliers $\lambda_j = -g_{k',j}$ and $\theta_{k,j} = -\lambda_j - g_{k,j}$ the solution obey the required KKT conditions for $P_{\text{SR}}^{\text{GC}}$.

We next want to prove that a (local) optimum of $P_{\text{SR}}^{\text{GC}}$ is a (local) optimum of $P_{\text{BC}}^{\text{GC}}$. At the optimum we have either that $s_{k,j} = 0$ such that $\theta_{k,j} \geq 0$ or that $s_{k,j}$ is larger than zero such that $\theta_{k,j} = 0$. For all k for which $s_{k,j}$ is larger than zero we find according to the KKT condition

$$g_{k,j} = -\lambda_j. \tag{2}$$

and for zero entries k^0 we have $\theta_{k^0,j} = -\lambda_j - g_{k^0,j} = g_{k,j} - g_{k^0,j} \geq 0$. Thus, the gradient of $s_{k^0,j}$ given by $g_{k^0,j}$ has to be smaller than the gradients of the non-zero elements, i.e. $g_{k^0,j} \leq g_{k,j}$. As a result, if $s_{k,j} = 1$ this is also an optimum of $P^{\text{GC}}_{\text{BC}}$. Assume $s_{k,j} < 1$, since a non-binary solution has at least two non-zero elements there will exist at least pairs k', k'' for which (2) holds. We therefore have for potential non-binary solutions

$$g_{k',j} = g_{k'',j}. \tag{3}$$

We want to prove that a non-binary solution does not form a (local) optimum. Assume we make an infinitesimal change of two or more elements constituting the non-binary solution in the column of $\boldsymbol{s}_j$ given by the vector $\boldsymbol{\epsilon}$ such that $\mathbf{1}^\top \boldsymbol{\epsilon} = 0$, i.e. the changed solution resides on the simplex according to definition 5. Let $f(\boldsymbol{S}) = \text{trace}[\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top]$ and $\boldsymbol{E}$ be the matrix indicating the corresponding change given by $\boldsymbol{\epsilon}$ of $\boldsymbol{s}_j$. From Taylor expanding f we find

$$\begin{aligned} f(\boldsymbol{S}+\boldsymbol{E}) &= f(\boldsymbol{S}) + \boldsymbol{g}_j^\top \boldsymbol{\epsilon} + \frac{1}{2}\epsilon^\top \boldsymbol{H}\epsilon \\ &= f(\boldsymbol{S}) + \frac{1}{2}\epsilon^\top \boldsymbol{H}\epsilon > f(\boldsymbol{S}). \end{aligned}$$

Second equality holds since $\boldsymbol{g}^\top \boldsymbol{\epsilon} = 0$ due to (3). The last inequality follows since the Hessian $\boldsymbol{H}$ is a diagonal matrix with entries $h_{k,k} = 2b_{k,k} = 2\delta > 0$ hence is positive definite. Thus, since the diagonal elements of $\boldsymbol{B}$ are positive any infinitesimal change over the simplex of a non-binary column of $\boldsymbol{S}$ will converge to a binary configuration since a non-binary solution form a suboptimal configuration. ■

Theorem 7

$$P^{\text{PC}}_{\text{BC}} \sim P^{\text{PC}}_{\text{SR}} \tag{4}$$

Proof We will prove the theorem by first considering the K-means objective function given by minimizing $\|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{S}\|_F^2$ where we have that the centroids $\boldsymbol{C}$ is given by $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{S}^\top\boldsymbol{D}$, $d_{k,k'} = \delta_{k,k'}(\sum_j s_{k,j})^{-1}$ and $\boldsymbol{S}$ is defined as in $P^{\text{PC}}_{\text{BC}}$, i.e. is a clustering assignment matrix. We note that this objective can be rewritten as

$$\begin{aligned} \|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{S}\|_F^2 &= trace[\boldsymbol{X}^\top\boldsymbol{X}] \\ &+ trace[\boldsymbol{S}^\top\boldsymbol{D}\boldsymbol{S}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{S}^\top\boldsymbol{D}\boldsymbol{S}] \\ &- 2trace[\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{S}^\top\boldsymbol{D}\boldsymbol{S}] \\ &= \|\boldsymbol{X}\|_F^2 - trace[\boldsymbol{S}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{S}^\top\boldsymbol{D}] \end{aligned}$$

using the fact that $\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{D} = \boldsymbol{I}$ for binary $\boldsymbol{S}$. This is identical to the objective in $P^{\text{PC}}_{\text{BC}}$ for $\boldsymbol{B} = \boldsymbol{X}^\top\boldsymbol{X}$ up to the constant $\|\boldsymbol{X}\|_F^2$. Thus, we will in the following w.l.g. assume $\boldsymbol{B} = \boldsymbol{X}^\top\boldsymbol{X}$.

To enforce the equality constraint and non-negativity constraint in $P^{\text{PC}}_{\text{SR}}$ we need to optimize the following objective

$$\begin{aligned} \arg\max_{\boldsymbol{S}} \mathfrak{L}(\boldsymbol{S}) &= \text{trace}[\boldsymbol{S}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{S}^\top\boldsymbol{D}] \\ &+ \sum_j \lambda_j(\sum_k \boldsymbol{s}_{k,j} - 1) + \sum_k \theta_{k,j}s_{k,j}, \end{aligned}$$

where λ_j is the lagrange multiplier for the j^{th} equality constraint and $\theta_{k,j}$ the lagrange multiplier for the non-negativity constraint imposed on $s_{k,j}$. Again, the optimal solution has to satisfy the following Karush-Kuhn-Tucker (KKT) conditions

$$\begin{aligned} g_{k,j} + \lambda_j + \theta_{k,j} &= 0 \\ \theta_{k,j} &\geq 0 \\ \theta_{k,j} s_{k,j} &= 0 \end{aligned}$$

where

$$\begin{aligned} g_{k,j} &= 2(\boldsymbol{D}\boldsymbol{S}\boldsymbol{X}^\top \boldsymbol{x}_j)_k - (\boldsymbol{S}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{S}^\top)_{k,d}\boldsymbol{D}^2_{k,d} && (5) \\ &= -\|\boldsymbol{x}_j - \boldsymbol{c}_k\|_F^2 + \|\boldsymbol{x}_j\|_F^2, && (6) \end{aligned}$$

is the gradient of the objective.

We first prove that a (local) optimum of $P^{\text{PC}}_{\text{BC}}$ is a (local) optimum of $P^{\text{PC}}_{\text{SR}}$. Consider a (local) optimal solution $\boldsymbol{S}^*$ to $P^{\text{PC}}_{\text{BC}}$. For $s^*_{k',j} = 1$ we must have that $g_{k',j} \geq g_{k,j} \quad \forall k \neq k'$ otherwise a change of assignment in a column of $\boldsymbol{S}^*$ would be more optimal. Again setting the lagrange multipliers $\lambda_j = -g_{k',j}$ and $\theta_{k,j} = -\lambda_j - g_{k,j}$ the solution obey the required KKT conditions for $P^{\text{PC}}_{\text{SR}}$.

We next want to prove that a (local) optimum of $P^{\text{PC}}_{\text{SR}}$ is a (local) optimum of $P^{\text{PC}}_{\text{BC}}$. At the optimum we have either that $s_{k,j} = 0$ such that $\theta_{k,j} \geq 0$ or that $s_{k,j}$ is non-zero. If $s_{k,j}$ is non-zero we find according to the KKT condition

$$g_{k,j} = -\lambda_j. \tag{7}$$

and since $\theta_{k^0,j} = -\lambda_j - g_{k^0,j} \geq 0$ for zero entries we have that the gradient of $s_{k^0,j}$ given by $g_{k^0,j}$ is smaller than the gradients of non-zero elements, i.e. $g_{k^0,j} \leq g_{k,j}$. Thus if $s_{k,j} = 1$ this is also an optimum of $P^{\text{PC}}_{\text{BC}}$. Assume $s_{k,j} < 1$, since a non-binary solution has at least two non-zero elements there will exist at least pairs k', k'' for which (7) holds. We therefore have

$$g_{k',j} = g_{k'',j}, \tag{8}$$

i.e. the minimal distance between two centroids and the j^{th} data point must be the same. Now assume there are two centroids $\boldsymbol{c}_{k'}$ and $\boldsymbol{c}_{k''}$ for which (8) holds. Then if the only data point belonging to $\boldsymbol{c}_{k'}$ and $\boldsymbol{c}_{k''}$ is $\boldsymbol{x}_j$, j can be assigned strictly to k' or k'' forming an equivalent binary solution. Assume now that (8) holds and more than one point belong to $\boldsymbol{c}_{k'}$ and $\boldsymbol{c}_{k''}$ such that $\boldsymbol{c}_{k'} \neq \boldsymbol{c}_{k''}$. We now have that there will only be overlap if the distance from $\boldsymbol{x}_j$ to $\boldsymbol{c}_{k'}$ and $\boldsymbol{c}_{k''}$ are identical. The Hessian of the objective function is given by

$$\begin{aligned} h_{(k,j),(k',j')} &= 2d_{k,k'}(x_j - c_k)^\top (x_{j'} - c_{k'}) \\ &= d_{k,k'}(\|\boldsymbol{c}_k - \boldsymbol{x}_j\|_F^2 + \|\boldsymbol{c}_k - \boldsymbol{x}_{j'}\|_F^2 - \|\boldsymbol{x}_j - \boldsymbol{x}_{j'}\|_F^2). \end{aligned}$$

Notice, the Hessian is block diagonal within each entry j, j' since $d_{k,k'} = 0$ for $k \neq k'$. Furthermore for non-binary j the diagonals are positive due to the triangular inequality $\|\boldsymbol{c}_k - \boldsymbol{x}_j\|_F^2 + \|\boldsymbol{c}_k - \boldsymbol{x}_{j'}\|_F^2 \geq \|\boldsymbol{x}_j - \boldsymbol{x}_{j'}\|_F^2$ and the fact that $\boldsymbol{c}_k \neq \boldsymbol{x}_j$ as this would imply $\boldsymbol{c}_k = \boldsymbol{c}_{k'}$. We want to prove that a non-binary solution does not form a (local) optimum. Assume we make an infinitesimal

change of two or more elements constituting the non-binary solution in the column of $\boldsymbol{s}_j$ given by the vector $\boldsymbol{\epsilon}$ such that $\mathbf{1}^\top \boldsymbol{\epsilon} = 0$, i.e. the changed solution resides on the simplex according to definition 5. let $f(\boldsymbol{S}) = \mathrm{trace}[\boldsymbol{S}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{S}\boldsymbol{D}^\top]$ be the K-means objective defined by the solution $\boldsymbol{S}$ and $\boldsymbol{E}$ be the matrix indicating the corresponding change in $\boldsymbol{S}$ given by $\boldsymbol{\epsilon}$. Define $q_{k,k'} = h_{k,j,k',j}$. From Taylor expanding f we find

$$\begin{aligned} f(\boldsymbol{S}+\boldsymbol{E}) &\approx f(\boldsymbol{S}) + \boldsymbol{g}_j^\top \boldsymbol{\epsilon} + \frac{1}{2}\epsilon^\top \boldsymbol{Q}\epsilon \\ &= f(\boldsymbol{S}) + \frac{1}{2}\epsilon^\top \boldsymbol{Q}\epsilon > f(\boldsymbol{S}). \end{aligned}$$

Second equality holds since $\boldsymbol{g}^\top \boldsymbol{\epsilon} = 0$ due to (8). The last inequality follows since $\boldsymbol{Q}$ is a diagonal matrix with $q_{k,k} > 0$ hence positive definite. Thus, any infinitesimal change over the simplex of a non-binary column of $\boldsymbol{S}$ will converge to a binary configuration since a non-binary solution form a suboptimal configuration.

The rest of the proof follows by noting that any symmetric similarity matrix has an imbedding in $\mathbb{R}^n$ given by $\boldsymbol{B} = \boldsymbol{\Phi}^\top \boldsymbol{\Phi}$ where $\boldsymbol{\Phi}$ denotes the embedding. In particular, the evaluation of distance between data points and cluster centers in the embedding only requires evaluations of the kernel function $\boldsymbol{B}$, i.e. according to equation (5) we have

$$\begin{aligned} &\|\boldsymbol{\phi}_j - \boldsymbol{c}_k\|_F^2 + \|\boldsymbol{\phi}_j\|_F^2 = (\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top)_{k,d}\boldsymbol{D}_{k,d}^2 - 2(\boldsymbol{D}\boldsymbol{S}\boldsymbol{b}_j)_k \Rightarrow \\ &\|\boldsymbol{\phi}_j - \boldsymbol{c}_k\|_F^2 = (\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top)_{k,d}\boldsymbol{D}_{k,d}^2 - 2(\boldsymbol{D}\boldsymbol{S}\boldsymbol{b}_j)_k - b_{j,j}. \end{aligned}$$

Thus, the pairwise clustering objective corresponds to the regular K-means objective in the space of the embedding $\boldsymbol{\Phi}$ ■

Due to theorem 6 and 7 we can perform continuous optimization over the simplex, rather than binary combinatorial optimization to solve for the hard clustering assignments. This admits novel types of algorithms for clustering. In fact, non-negative optimization problems with linear constraints form some of the most well studied problems in optimization. In particular $P_{\mathrm{SR}}^{\mathrm{GC}}$ form the well studied non-negative quadratic programming problem with linear constraints. Thus, the above formulation admits the use of standard continuous optimization rather than annealing approaches, fuzzy and spectral methods while guaranteeing binary combinatorial cluster assignments.

3.1 The SR-clustering algorithm

We presently derive a simple algorithm based on projected gradient ascent that directly enforces the simplicial constraints by recasting the problem in the l_1-normalization invariant variables $\widetilde{\boldsymbol{s}}_j = \frac{\boldsymbol{s}_j}{\sum_k s_{k,j}}$. Notice, $\frac{\partial \widetilde{s}_{k',j}}{\partial s_{k,j}} = \frac{\delta_{k',k}}{\sum_k s_{k,j}} - \frac{s_{k',j}}{(\sum_k s_{k,j})^2}$. Hence, differentiating by parts, we find the following updates for $\boldsymbol{S}$ for the clustering objectives recast in the normalization invariant variables $\widetilde{\boldsymbol{S}}$

$$s_{k,j} \leftarrow \max\{\widetilde{s}_{k,j} + \mu(\frac{g_{k,j}}{\sum_k \widetilde{s}_{k,j}} - \sum_{k'} g_{k',j}\frac{\widetilde{s}_{k',j}}{(\sum_k \widetilde{s}_{k,j})^2}), 0\}, \quad \widetilde{s}_{k,j} = \frac{s_{k,j}}{\sum_k s_{k,j}}$$

where the updating is performed on all elements simultaneously and μ is a step-size parameter that is tuned by line-search. We recall that the gradients $g_{k,j}$ of the updates for GC and PC are given by

$$\begin{aligned} g_{k,j}^{\text{GC}} &= 2\sum_i s_{k,i} b_{i,j} \\ g_{k,j}^{\text{PC}} &= 2(\boldsymbol{DSB}_j)_k - (\boldsymbol{SBS}^\top)_{k,d}\boldsymbol{D}_{k,d}^2. \end{aligned}$$

Since the matrix $\boldsymbol{B}$ in P^{GC} is normally given by the sparse adjacency matrix $\boldsymbol{A}$ subtracted a rank one term the computational complexity per iteration for the $P_{\text{SR}}^{\text{GC}}$ optimization according to the above updates is $\mathcal{O}(K\|\boldsymbol{A}\|)$ where $\|\boldsymbol{A}\|$ denotes the number of non-zero entries in $\boldsymbol{A}$ whereas the computational complexity for the $P_{\text{SR}}^{\text{PC}}$ updates for the general pairwise clustering by $\boldsymbol{SB}$ having complexity $\mathcal{O}(KJ^2)$. In particular, for the K-means objective this calculation is given $\boldsymbol{SX}^\top\boldsymbol{X}$ which has complexity $\mathcal{O}(min(KIJ, KJ^2))$. This is the same computational cost per iteration as the original greedy iterative refinement algorithms that evaluates each nodes/observations membership to each cluster.

Notice, after completing each iteration $\sum_k \widetilde{s}_{k,j} = 1$. As a result, the above updates reduce to

$$s_{k,j} \leftarrow \max\{\widetilde{s}_{k,j} + \mu(g_{k,j} - \sum_{k'} g_{k',j}\widetilde{s}_{k',j}), 0\}, \quad \widetilde{s}_{k,j} = \frac{s_{k,j}}{\sum_k s_{k,j}}.$$

Thus, the update has the following simple interpretation: $g_{k,j}$ gives the preference strength of cluster k to node/observation j. $\widetilde{s}_{k,j}$ denotes node/observation j's soft assignment to cluster k. Thus, $\sum_{k'} g_{k',j}\widetilde{s}_{k',j}$ weights how each cluster prefer node j with the nodes soft assignment, i.e forming what we will denote the node's/observation's preference. Thus, if a cluster prefers a node/observation more than the node's/observation's preference the soft assignment for this cluster will increase relative to some of the remaining clusters and vice versa if the cluster prefer the node/observation less than the node's/observation's preference. We will denote the above algorithm SR-*clustering*, details of the algorithm can be found in the Matlab implementation available for download at `www.mortenmorup.dk`.

Similarly to the fuzzy clustering and annealing approaches an important property of the above algorithm is that initially node/observation j identifies itself to some degree with all the clusters by random initialization. As such, the clusters will have to compete for all the data points such that regions where there are many data points will draw more attention than regions with few data points. As a result, the above SR-*clustering* algorithm will tend to place more clusters at dense regions than the regular greedy iterative refinement algorithm where each node/observation is assigned to its closest cluster and as such the densities of the true underlying clusters are not explicitly taken into account. Furthermore, the SR-*clustering* algorithm has a lesser tendency to generate empty clusters as there is no initial hard assignment causing clusters that happens to not be the closest to any data point become empty. Instead, the clusters compete over all the data points and eventually each cluster is highly likely to win at least a few data points, see also figure 1. However, contrary to the fuzzy clustering and annealing approaches no temperature has to be controlled in order to achieve binary solutions. The solution simply has to converge and the binary combinatorial assignments are guaranteed according to theorem 6 and 7.

4. Simplex Point Clustering

In figure 1 we investigate the derived SR-clustering algorithm on a simulated data set. The data set consists of 5 clusters residing in 2-dimensional space. Three of the clusters have 250 data points and two of the clusters have 1000 data points. From the figure it can be seen that the SR-*clustering* algorithm ($\boldsymbol{B} = \boldsymbol{X}^\top \boldsymbol{X}$) works better than the greedy iterative refinement algorithm for K-means as the algorithm tends to place more clusters at dense regions of the data.

In figure 2 we compare the standard iterative refinement algorithm for K-means with the corresponding SR-*clustering* algorithm on the USPS handwritten image data set (Cun et al., 1990) containing 7,291 images in a 256 dimensional space, i.e $\boldsymbol{X}^{256\times 7291}$ as well as CBCL Face Database ♯1 (MIT Center For Biological and Computation Learning `http://www.ai.mit.edu/projects/cbclcontaining`) containing 2,429 facial images in 361 dimensional space, i.e $\boldsymbol{X}^{361\times 2429}$. We further compare the performance to a random dataset generated such that 10,000 observations resided uniformly within the unit hyper-cube of a 1,000 dimensional space, i.e. $\boldsymbol{X}^{1000\times 10000}$. While the SR-*clustering* algorithm performs somewhat better than the iterative refinement algorithm for the USPS data, the algorithm performed significantly better on the Face Database and random data using about the same number of iterations as the greedy iterative refinement algorithm.

5. Simplex Graph Clustering

Based on the SR-*clustering* algorithm we analyzed three networks, a biological network of protein interaction, a text-mining network of word association, and a social network of co-authorship respectively.

Yeast Protein Interaction Network (Yeast): The yeast protein interaction network described in (Sun et al., 2003) quantifies the interaction between proteins. The size of the data is $2,361$ vertices containing $13,828$ edges.

Reuters terror news network (Reuters 911) Reuters terror news network is based on all stories released during 66 consecutive days by the news agency Reuters concerning the September 11 attack on the U.S. The vertices of a network are words (terms); there is an edge between two words if and only if they appear in the same text unit (sentence). The network has $13,332$ vertices (different words in the news) and $243,447$ edges (Corman et al., 2002).

Condensed matter collaborations 2005 (CondPhys2005): Network of co-authorship between scientists posting preprints on the Condensed Matter E-Print Archive. This version includes all preprints posted between Jan 1, 1995 and March 31, 2005 (Newman, 2001) the size of the network is $40,421$ vertices containing $351,386$ edges. To fit the framework of Modularity and Hamiltonian optimization we converted the undirected weighted graph into an unweighted graph by setting an edge to one where authors had co-authored a paper with another author (i.e. disregarding the weights).

The result of the SR-clustering analysis can be seen in figure 3. Clearly, the algorithm has detected regions with densely connected groups of vertices with relatively less connections running between groups. Furthermore, the performance of the SR-clustering analysis is equivalent to the performance of the best cooling schedules found by Gibbs sampling.

6. Discussion

We have demonstrated how the combinatorial constraints in clustering can be relaxed to a continuous optimization problem over the simplex. Contrary to previous clustering relaxations that are dependent on some problem specific annealing parameter in order to recover the original combinatorial constraints simplicial relaxation constitutes an exact proxy for the original hard assignment clustering problems such that the combinatorial problem is equivalent to the problem formed by the simplicial relaxation. This opens new doorways from optimization to solve for the generally difficult clustering problems and demonstrates that the binary combinatorial constraints can, in fact, be reformulated as continuous optimizations over the simplex.

We proposed the SR-*clustering* algorithm which constitutes a simple gradient ascent based method that optimizes the clustering assignment matrix $\boldsymbol{S}$. We demonstrated how this method performs better than the greedy iterative refinement algorithm for K-means as the method favors to put emphasis on regions that are dense while empty clusters are less likely to occur. For clustering of graphs the method gave equivalent performance to previous reported annealing based approaches (Lehmann and Hansen, 2007). Contrary to the annealing approaches, however, the method does not rely on controlling a cooling scheme but is guaranteed to converge to a binary configuration.

References

R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1): 47–97, Jan 2002. doi: 10.1103/RevModPhys.74.47.

F. R. Bach and M. I. Jordan. Learning spectral clustering. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, 2005. ISSN 1533-7928.

P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

J.M. Buhmann and T. Hofmann. A maximum entropy approach to pairwise data clustering. *Proceedings of the 12th IAPR International. Conference on Pattern Recognition*, 2:207–212, 1994.

S. R. Corman, T. Kuhn, R. D. McPhee, and Dooley K. J. Studying complex discursive systems: Centering resonance analysis of communication. *Human communication research*, 28(2):157–206, 2002.

Y. L. Cun, J. S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pages 598–605. Morgan Kaufmann, 1990.

O. du Merle, P. Hansen, B. Jaumard, and N. Mladenovic. An interior point algorithm for minimum sum-of-squares clustering. *SIAM J. Sci. Comput.*, 21(4):1485–1505, 2000.

M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, 1998.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

B. Fischer, T. Zöller, J. M. Buhmann, R. Friedrich, and W. Universität. Path based pairwise data clustering with application to texture segmentation, 2001.

Y. Fu and P.W. Anderson. Application of statistical mechanics to np-complete problems in combinatorial optimisation. *J. Phys. A: Math. Gen.*, 19:1605–1620, 1986.

Gene H. Golub and Charles F. Van Loan. *Matrix Computation*. Johns Hopkins Studies in Mathematical Sciences, 3 edition, 1996.

J. A. Hartigan and M. A. Wong (1979). A k-means clustering algorithm. *Applied Statistics*, 28(1): 100–108, 1979.

R. J. Hathaway and J. C. Bezdek. Recent convergence results for the fuzzy c-means clustering algorithms. *Journal of Classification*, 5:237–247, 1988.

T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1–14, 1997.

C. Kemp and J. B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences of the United States of America*, 105(31):1068710692, 2008.

B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22, New York, NY, USA, 1999. ACM.

S. Lehmann and L. K. Hansen. Deterministic modularity optimization. *The European Physical Journal B*, 60(1):83–88, 2007.

S. P. Lloyd. Least square quantization in pcm. *Special issue on quantization, IEEE Trans. Inform. Theory*, 28:129–137, 1982.

N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

M. E. J. Newman. The structure of scientific collaboration networks. *PNAS*, 98(2):404–409, 2001.

M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 103 (23):8577–8582, 2006.

M. E. J. Newman. The structure and function of complex networks, March 2003.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113–1–15., 2004.

A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Adv. in Neural Inform. Process. Systems (NIPS'01)*, 14:83–88, 2001.

N. R. Pal and J. C. Bezdek. On cluster validity for the fuzzy c-means model. *Fuzzy Systems, IEEE Transactions on*, 3(3):370–379, 1995.

J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM J. on Optimization*, 18(1):186–205, February 2007. ISSN 1052-6234.

A. Pothen., H.D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11:430–452, 1990.

G. Punj and D. W. Stewart. Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research*, 20(2):134–148, 1983.

J. Reichardt and S. Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett.*, 93(21):218701, Nov 2004.

J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(1):016110, 2006.

N. Slonim, G. S. S. Atwal, G. Tkacik, and W. Bialek. Information-based clustering. *Proceedings of the National Academy of Sciences of the United States of America*, 102(51):18297–18302, 2005.

S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450, 2003.

U. von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17:395–416, 2007.

B. Waclaw and Z. Burda. Counting metastable states of Ising spin glasses on arbitrary graphs. *Physical Review E*, 77(4):041114–+, 2008.

S. Wasserman and K. Faust. *Social network analysis*. Cambridge University Press, Cambridge, 1994.

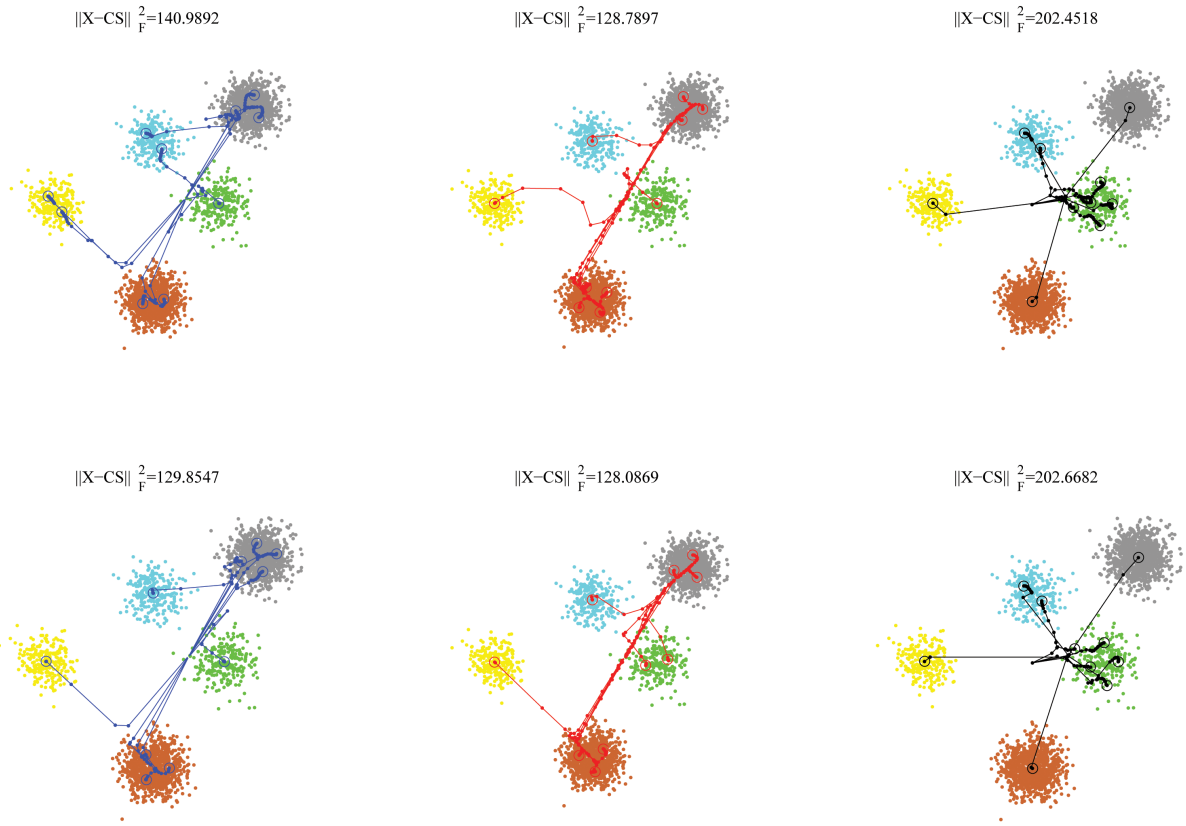

Figure 1: The results for K-means based on the proposed SR-*clustering algorithm* for initial step size $\mu = 1$ given to the left in blue and $\mu = 0.01$ given in the middle in red as well as the greedy iterative refinement algorithm to the right given in black. $K = 10$ clusters were fitted for two different random initializations (top panel and bottom panel). Whereas the brown and grey clusters each contain 1000 data-points in $\mathbb{R}^2$ the remaining clusters each have 250 data-points. The iteration path is indicated by the blue, red and black lines and dots indicate position of the various centroids at given iterations, colored circles indicate the final optimal configuration of the centroids found by each algorithm while at the top of each plot is given the value of the K-means objective function. While the greedy iterative refinement algorithm given by the black paths send the centroids to the regions where they happen to be closest to data points, the SR-*clustering* approach send the centroids towards the regions with most data points (i.e. most of the blue and red paths initially go towards the two dense clusters). Centroids that on their way to the dense regions come close to the less dense clusters change direction and take over responsibility of these regions. This results in dense regions being assigned more clusters than less dense regions overall resulting in better partitions of the data. Notice how small initial step-sizes (red paths vs. blue paths) result in better clustering assignments as the centroids are more carefully moved to the dense data regions.

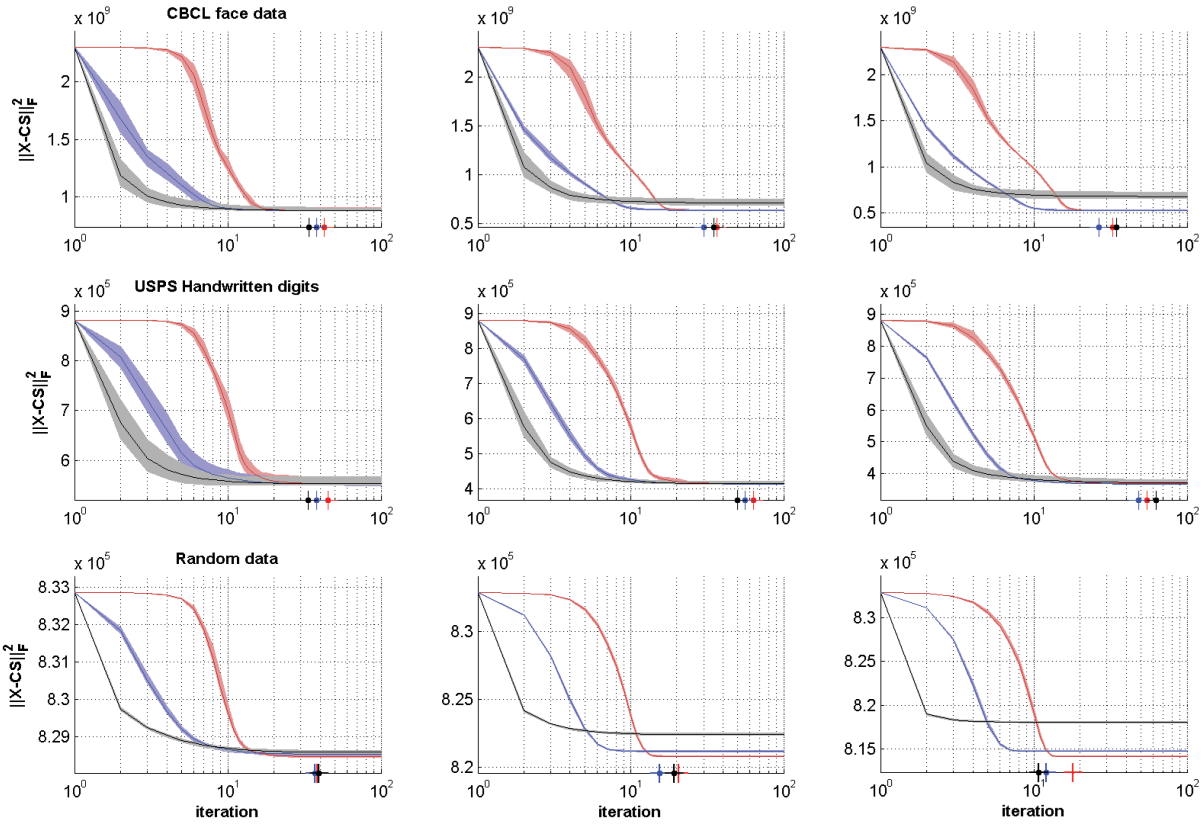

Figure 2: Clustering result for the CBCL Face Database ♯1, USPS handwritten image data and randomly generated data. SR-*clustering* results are given by red curves ($\mu = 0.01$), blue curve ($\mu = 1$) and black curves show the performance of the greedy iterative K-means refinement algorithm. 100 models based on $K = 10$, $K = 50$ and $K = 100$ clusters were inspected. Shaded region indicates best and worst of the 100 solutions for each method. Clearly, the SR-*clustering* method outperforms the greedy iterative refinement algorithm, particularly for the Face Database and random data. The mean number of iterations required for convergence for each approach is indicated on the x-axis by the colored dots with vertical bars. Clearly, all approaches use more or less the same number of iterations. In particular, the greedy iterative refinement algorithms sometimes uses more iterations than the proposed SR-*clustering* method. All methods were based on the same random initializations.

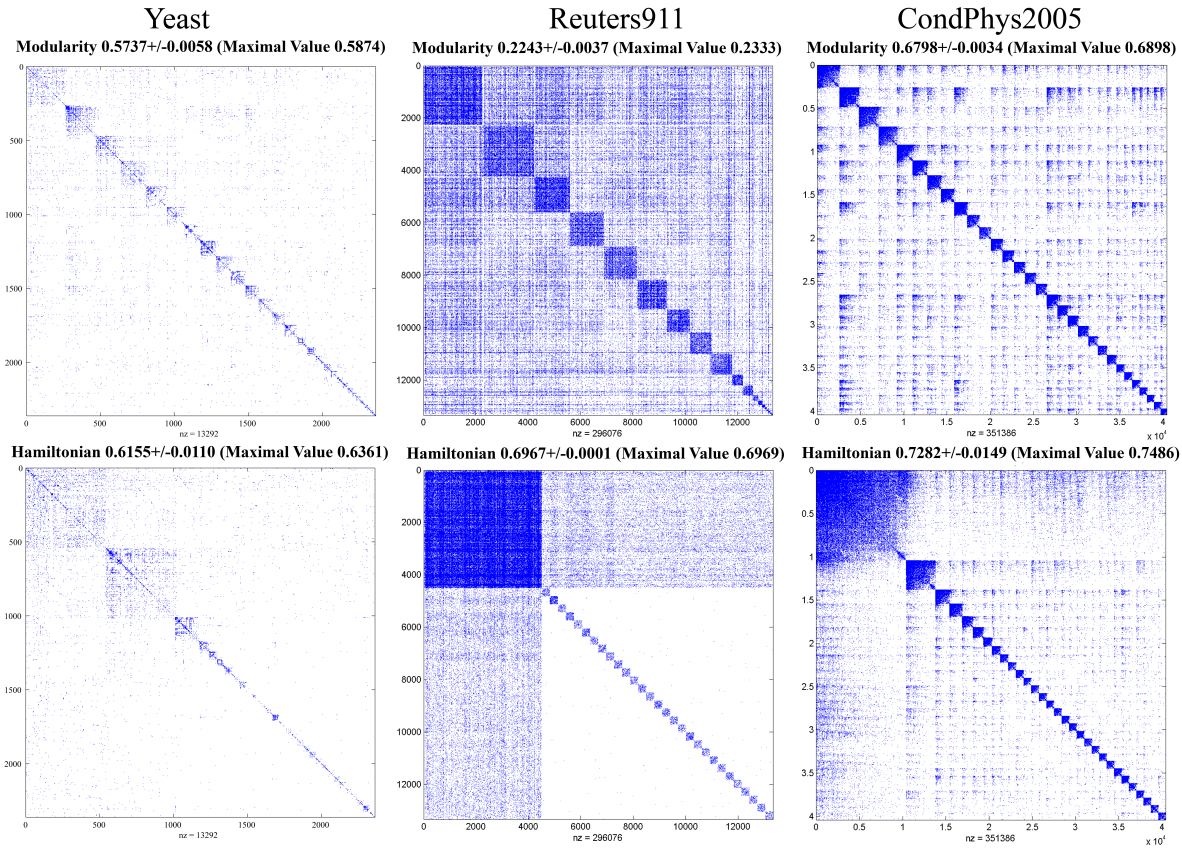

Figure 3: SR-*clustering* analysis of the yeast, Reuters 911 and Condensed Physics 2005 networks for $K = 30$. Clearly, the method has clustered the data into regions of high intra cluster connectivity with low inter cluster connectivity. Reported are the mean of the estimated Modularity and Hamiltonian values as well as the standard deviation of these values for 100 estimated models as well as the value of the best estimated model. Below is given the corresponding permuted graph for the best estimated model. The modularity values are in accordance with the reported values found by Gibbs sampling and simulated annealing in (Lehmann and Hansen, 2007). Notice how the effect of null-hypothesis greatly impact the type of clustering achieved. Hence, the modularity tend to cluster nodes more connected than indicated by their degree product whereas the Hamiltonian approach is set to cluster vertices that are connected more than average. The reported values for the Modularity approach is given by $\frac{1}{\|\boldsymbol{A}\|} trace[\boldsymbol{S}(\boldsymbol{A} - \frac{1}{\|\boldsymbol{A}\|}\boldsymbol{k}\boldsymbol{k}^\top)\boldsymbol{S}^\top]$ where $k_i = \sum_j a_{i,j}$ whereas the reported Hamiltonian values are calculated as $\frac{1}{\|\boldsymbol{A}\|} trace[\boldsymbol{S}(\boldsymbol{A} - \frac{\|\boldsymbol{A}\|}{J^2}\mathbf{1}\mathbf{1}^\top)\boldsymbol{S}^\top]$. Using Gibbs sampling we found for the best performing cooling schedules Modularity values of 0.2270 for Yeast, 0.2310 for Reuters 911 and 0.6853 for CondPhys2005 networks giving comparable performance to the obtained SR-clustering results for the Reuters 911 and CondPhys2005 networks. We were however unable to recover the Modularity values of 0.5737 obtained on average by SR-clustering for the Yeast network.