Constrained K-means with General Pairwise and Cardinality Constraints

Adel Bibi, Baoyuan Wu*, Bernard Ghanem
(* Corresponding Author)
Visual Computing Center, King Abdullah University of Science and Technology
adel.bibi@kaust.edu.sa, wubaoyuan1987@gmail.com, bernard.ghanem@kaust.edu.sa

Abstract

In this work, we study constrained clustering, where some constraints are utilized to guide the clustering process. In existing work on this topic, two main categories of constraints have been explored, namely pairwise and cardinality constraints. Pairwise constraints enforce that the cluster labels of two instances be the same (must-link constraints) or different (cannot-link constraints). Cardinality constraints force cluster sizes to satisfy a user-specified distribution, such as a balanced distribution. However, most constrained clustering methods focus on only one category of constraints at a time. In this paper, we enforce both categories in a single unified clustering model. As these two categories provide useful information at different levels, utilizing both of them is expected to allow for better clustering performance than using only one of them. Specifically, we rewrite the discrete optimization problem into an equivalent continuous one, using the ℓ_p-box ADMM framework (Wu and Ghanem 2016). Pairwise/cardinality constraints are incorporated into the model as quadratic/linear constraints. The resulting constrained continuous optimization can be efficiently solved using the ADMM algorithm. Extensive experiments on both synthetic and real data demonstrate: **(1)** when utilizing only one constraint category, the proposed model is superior to or competitive with state-of-the-art constrained clustering models, and **(2)** when utilizing both categories of constraints simultaneously, the proposed model shows better performance than when only one category is used.

Introduction

Clustering is the task of partitioning data into different clusters, based on some specific cluster assumptions. For example, K-means and Gaussian mixture models (GMM) assume each cluster is sampled from a Gaussian distribution. In contrast, density-based clustering assumes that the densities of data points in different clusters should be different, such as Chameleon (Karypis, Han, and Kumar 1999) and AITC (Wu and Hu 2011), or clusters should be partitioned at low density regions (Chapelle and Zien 2005). However, if the adopted cluster assumption is not suited to the target dataset, the clustering performance may be poor. To avoid such performance instability, prior knowledge or constraints on the data can be used to guide the clustering process. These constraints are independent of cluster assumptions, and they provide weak supervision to reflect user preferences. Thus, clustering with constraints, called *constrained clustering* (Von Luxburg 2007; Lu and Ip 2010; Lu and Leen 2007; Wu et al. 2013b; Ng et al. 2002; Wagstaff et al. 2001; Klein, Kamvar, and Manning 2002; Wagstaff and Cardie 2000; Höppner and Klawonn 2008; Klawonn and Höppner 2006; Bradley, Bennett, and Demiriz 2000), is expected to give better and more stable performance than unconstrained clustering.

Two main categories of constraints have been widely studied in the field of constrained clustering, namely pairwise and cardinality constraints. *Pairwise constraints* include must-link and cannot-link constraints. Must-link constraints enforce that a set of pairs of instances should be in the same cluster, while cannot-link constraints enforce that they belong to different clusters. This category represents the relationship between cluster labels of two instances, thus it can be viewed as instance-level constraints. *Cardinality constraints* provide a prior/preference on the size distribution of all clusters. For example, if the user expects a uniform partition, then it is incorporated as balancing constraints, which are a special case of the more general cardinality constraints. This category can be viewed as cluster-level constraints.

Many clustering methods have been proposed to utilize one of the two categories of constraints listed above, such as the ones with pairwise constraints (Lu and Ip 2010; Von Luxburg 2007; Lu and Leen 2007; Dai et al. 2003; Wu et al. 2013b; Ng et al. 2002; Wagstaff et al. 2001; Klein, Kamvar, and Manning 2002; Wagstaff and Cardie 2000), and the ones with cardinality constraints (Höppner and Klawonn 2008; Klawonn and Höppner 2006; Bradley, Bennett, and Demiriz 2000; Shi and Malik 2000). However, to the best of our knowledge, there is no existing work that can seamlessly incorporate both categories jointly. We believe that a more reasonable approach is to jointly embed both categories into one clustering model, if both of them are available. Firstly, for the same clustering task, both constraints can be provided simultaneously, as they are derived from different sources. For example, pairwise constraints are usually obtained from an oracle query, while cardinality constraints can be obtained from experience or user preference. Secondly, they represent supervision at different levels. Each of them can provide particularly useful information that is not covered by the other. Thus, better performance is ex-

pected when both constraint categories are used. This point will be supported in our experiments.

For existing constrained clustering methods that handle one constraint category, it is non-trivial to directly add the other. For example, cardinality constraints cannot be embedded into the COP-KMEANS (Wagstaff et al. 2001) or CCL (Dai et al. 2003) models, where pairwise constraints have been utilized. Moeover, it is also not easy to embed the pairwise constraints into normalized/ratio-cut (Shi and Malik 2000), which exploits balanced distribution constraints. In short, existing models are specifically designed to exploit one category of constraints at a time.

In this work, we propose a unified model to incorporate both categories of constraints to guide the clustering process. We reformulate the standard discrete K-means model into an equivalent continuous optimization problem. Specifically, a recent technique in integer programming, called ℓ_p-box ADMM (Wu and Ghanem 2016), is exploited to replace the discrete variables (cluster labels) with the intersection-set of two continuous constraints. In this framework, pairwise constraints are modeled as quadratic functions and cardinality constraints as linear ones.

Our contributions are two-fold. **(i)** We embed both pairwise and cardinality constraints into one unified clustering model. To the best of our knowledge, this is the first attempt in the field of constrained clustering. **(ii)** We incorporate both categories of hard constraints based on an equivalent continuous reformulation of standard K-means.

Related Work

Here, we briefly review existing clustering models that utilize pairwise or cardinality constraints.

Pairwise Constraints. They were first introduced into clustering in (Wagstaff and Cardie 2000) and (Wagstaff et al. 2001). In (Wagstaff and Cardie 2000), a method called COP-COBWEB inserted the pairwise constraints into the clustering process of the incremental clustering method COBWEB (Fisher 1987), which utilizes four operators (i.e. add, new, merge, and split) to maximize the intra-cluster similarity and the inter-cluster dissimilarity. In each operator of COP-COBWEB, the given pairwise constraints are checked to ensure the satisfaction of all constraints. In (Wagstaff et al. 2001), the method COP-KMEANS checks the pairwise constraints in each assignment step of K-means. Both COP-COBWEB and COP-KMEANS treat the pairwise constraints as hard constraints (i.e. all constraints must be satisfied), and the constraints are somewhat independent of the original objective. A common limitation of these two methods is that the processing order of instances influences clustering performance, and sometimes they may even fail to output a feasible partition.

To avoid this limitation, some methods propose to treat the pairwise constraints as soft constraints (i.e. a subset of these constraints could be violated), and developed more flexible approaches to embed constraints. For example, in constrained complete-link (CCL) (Klein, Kamvar, and Manning 2002), pairwise constraints are used to modify the instance proximity computed in the original feature space. Then, standard complete-link clustering is applied using the modified proximity matrix. Penalized probabilistic clustering (PPC) (Lu and Leen 2007) used pairwise constraints as a prior term w.r.t. the cluster labels within the underlying GMM-based model. Clustering configurations not satisfying the constraints have a lower probability. Moreover, HMRF-KMEANS (Basu, Bilenko, and Mooney 2004) embeds pairwise constraints as correlations between the cluster label variables in a hidden Markov random field (HMRF). A metric learning step is added into the standard K-means algorithm to encourage the gradual satisfaction of pairwise constraints. Other methods propagate pairwise constraints via instance similarity to obtain soft constraints, such as constrained spectral clustering (Lu and Ip 2010) and HMRF-pc (Wu et al. 2013b) (Wu et al. 2013a).

Cardinality Constraints. They are widely used to guide the clustering process. Balancing constraints are a special type that encourage all clusters to be balanced in size or connecting weights. For example, normalized cuts (Shi and Malik 2000) divides the standard cut cost (sum of edge weights connecting the two clusters) by the sum of edge weights between each cluster and all other instances. Hence, each cluster is encouraged to have similar edge weights connecting to other clusters. Similarly, ratio cut (Dhillon, Guan, and Kulis 2004) normalizes the cut function by the size of each cluster to encourage similar sized clusters. Equi-sized Fuzzy c-means (FCM) (Klawonn and Höppner 2006) formulates the balancing constraints as equality constraints, where the size of each cluster equals to the average cluster size (i.e. the total number of instances divided by the number of clusters). More general constraints on the cluster sizes have also been explored. For example, a constrained K-means method (Bradley, Bennett, and Demiriz 2000) sets a lower bound on the cluster's size, to avoid very small or empty clusters that occur in standard K-means. An extension of the Equi-sized FCM is proposed in (Höppner and Klawonn 2008), where the size of one single cluster is set to a specific size.

Many methods have been proposed to utilize one category of constraints; however, to the best of our knowledge, combining both categories into a unified clustering model has not been explored in any existing work.

Proposed Method

Unlike previous methods that can only handle either pairwise or cardinality constraints, we show, in this section, a detailed derivation of our framework that embeds both constraints simultaneously. In fact, this formulation is flexible and generic enough to handle any other linear equality or inequality constraints like cardinality bounds. In our framework, we adopt the K-means integer program formulation expressed as follows:

$$\begin{aligned} \min_{\{x_{ij}\}_{i=1,j=1}^{n,k}} \quad & \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij}\left\|\mathbf{s}_i - \frac{\sum_{p=1}^{n} x_{pj}\mathbf{s}_p}{\sum_{l=1}^{n} x_{lj}}\right\|_2^2 \\ \text{s.t.} \ & \sum_{j=1}^{k} x_{ij} = 1, \ \ \forall \ i \ \ x_{ij} \in \{0,1\} \ \forall \ i,j \end{aligned} \tag{1}$$

where $\mathbf{s}_p \in \mathbb{R}^d$ is the p^{th} data point to be clustered and k is the number of clusters. The variable x_{ij} defines the binary association between data point i and cluster j. The constraint $\sum_{j=1}^{k} x_{ij} = 1$ enforces data point i to belong to one and only one cluster. This constraint can be simply written as a matrix vector multiplication: $\Psi^\top \mathbf{x} = \mathbf{1}_n$, where $\Psi^\top \in \mathbb{R}^{n \times nk}$ is a binary matrix that has in each row a vector $\mathbf{1}_k^\top$ that sums all the binary labels for a given data point while the rest are 0.

To simplify the fractional objective, we introduce variable w_{pj}, such that: $x_{pj} = w_{pj} \sum_{l=1}^{n} x_{lj}$. For ease of notation, we concatenate all the binary labels x_{ij} into one vector ordered by the data points one at a time as follows: $\mathbf{x}^\top = [(x_{11} \ \cdots \ x_{1k}) \ \cdots \ (x_{n1} \ \cdots \ x_{nk})]^\top$. We also concatenate and reorder the w_{pj} values one cluster at a time: $\mathbf{w}^\top = [(w_{11} \ \cdots \ w_{n1}) \ \cdots \ (w_{1k} \ \cdots \ w_{nk})]^\top$. A matrix $\mathbf{P} \in \mathbb{R}^{nk \times nk}$ is used to swap the order of the binary vectors from a cluster based order to a data point order and vice versa. Note that the matrix $\mathbf{P}$ is a proper permutation matrix that is symmetric and it satisfies: $\mathbf{P}\mathbf{P}^\top = \mathbf{P}^2 = \mathbf{I}_{nk}$. Therefore, the compact form of unconstrained K-means can be re-written as follows:

$$\begin{aligned} \min_{\{x_{ij}\}_{i=1,j=1}^{n,k}, \{w_{pj}\}_{p=1,j=1}^{n,k}} \quad & \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij} \Big\| \mathbf{s_i} - \sum_{p=1}^{n} w_{pj} \mathbf{s_p} \Big\|_2^2 \\ \text{s.t.} \quad & \Psi^\top \mathbf{x} = \mathbf{1}_n, \ \ \mathbf{x} = \mathbf{P}\mathbf{w} \odot \mathbf{C}\mathbf{x}, \ \ \mathbf{x} \in \{0,1\}^{nk} \end{aligned} \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{nk \times nk}$ sums the binary labels of each cluster. Next, we discuss how we incorporate cardinality and pairwise (must-link and cannot-link) constraints. Details of this mathematical treatment are in the **supplementary material**.

Cardinality Constraints. They are enforced by a linear constraint $\sum_{i=1}^{n} x_{ij} = u_j \ \ \forall j$, or in vector form as $\mathbf{Q}\mathbf{P}^\top \mathbf{x} = \mathbf{u}$, where u_j is the size of cluster j and $\mathbf{Q} \in \mathbb{R}^{k \times nk}$ sums the binary labels of each cluster for all data points.

Must-Link Constraints. We define $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{kv \times nk}$ as selection matrices that choose the two sets of data points ($\mathbf{E}_1\mathbf{x}$ and $\mathbf{E}_2\mathbf{x}$) involved in the v must-link constraints. Requiring $\mathbf{E}_1\mathbf{x} = \mathbf{E}_2\mathbf{x}$ is equivalent to $\|\mathbf{E}_1\mathbf{x} - \mathbf{E}_2\mathbf{x}\|_2^2 = \|\mathbf{E}_1\mathbf{x}\|_2^2 + \|\mathbf{E}_2\mathbf{x}\|_2^2 + 2\mathbf{x}^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} = 0$. Since $\Psi^\top\mathbf{x} = \mathbf{1}_n$ and $\mathbf{x} \in \{0,1\}^{nk}$, then $\|\mathbf{E}_1\mathbf{x}\|_2^2 = \|\mathbf{E}_2\mathbf{x}\|_2^2 = v$. Therefore, the must-link constraints can be encoded using one additional quadratic constraint: $\mathbf{x}^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} = v$.

Cannot-Link Constraints. We define $\mathbf{E}_3, \mathbf{E}_4 \in \mathbb{R}^{ke \times nk}$ as selection matrices for the two sets of data points ($\mathbf{E}_3\mathbf{x}$ and $\mathbf{E}_4\mathbf{x}$) involved in the e cannot-link constraints. We incorporate these constraints by adding one additional quadratic constraint: $\mathbf{x}^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x} = 0$. Similar to the must-link case, the quadratic constraint follows naturally from: $\|\mathbf{E}_3\mathbf{x} + \mathbf{E}_4\mathbf{x}\|_2^2 = 2e$ (enforces dissimilar labels), $\Psi^\top\mathbf{x} = \mathbf{1}_n$, and $\mathbf{x} \in \{0,1\}^{nk}$.

Therefore, constrained K-means is formulated as follows:

$$\begin{aligned} \min_{\{x_{ij}\}_{i=1,j=1}^{n,k}, \{w_{pj}\}_{p=1,j=1}^{n,k}} \quad & \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij} \Big\| \mathbf{s_i} - \mathbf{S}\Lambda_j\mathbf{w} \Big\|_2^2 \\ \text{s.t.} \quad & \Psi^\top\mathbf{x} = \mathbf{1}_n, \ \ \mathbf{x} \in \{0,1\}^{nk}, \ \ \ \mathbf{Q}\mathbf{P}^\top\mathbf{x} = \mathbf{u} \\ & \mathbf{x} = \mathbf{P}\mathbf{w} \odot \mathbf{C}\mathbf{x}, \ \ (\mathbf{E}_1\mathbf{x})^\top\mathbf{E}_2\mathbf{x} = v, \ \ (\mathbf{E}_3\mathbf{x})^\top\mathbf{E}_4\mathbf{x} = 0 \end{aligned} \tag{3}$$

where $\mathbf{S} \in \mathbb{R}^{d \times n}$ contains all the data points in its columns and $\Lambda_j \in \mathbb{R}^{n \times nk}$ is zero everywhere except for the j^{th} block that is identity, i.e. $\Lambda_j = [\mathbf{0} \cdots \ \ \mathbf{I}_j \ \ \cdots \mathbf{0}]$. To solve problem (3), we use the ℓ_p-box ADMM method proposed in (Wu and Ghanem 2016) to achieve a local solution to the equivalent continuous reformulation of the clustering problem. For simplicity, we use the ℓ_2 version of this method. In this framework, binary constraints are replaced with an intersection of the ℓ_2-sphere (defined by set S_2) and box constraints (defined by set S_b). To apply ℓ_2-Box ADMM, we introduce auxiliary variables ($\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{z}_3$, and $\mathbf{z}_4$) to separate the binary constraints and change the quadratic constraints into bi-linear ones, as follows:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{w},\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3,\mathbf{z}_4} \quad & \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij} \Big\| \mathbf{s_i} - \mathbf{S}\Lambda_j\mathbf{w} \Big\|_2^2 \\ \text{s.t.} \quad & \Psi^\top\mathbf{x} = \mathbf{1}_n, \ \ \mathbf{z}_1 \in S_b, \ \ \mathbf{z}_2 \in S_2, \ \ \mathbf{Q}\mathbf{P}^\top\mathbf{x} = \mathbf{u} \\ & \mathbf{x} = \mathbf{P}\mathbf{w} \odot \mathbf{C}\mathbf{x}, \ \ (\mathbf{E}_1\mathbf{z}_3)^\top\mathbf{E}_2\mathbf{x} = v, \ \ (\mathbf{E}_3\mathbf{z}_4)^\top\mathbf{E}_4\mathbf{x} = 0 \\ & \mathbf{x} = \mathbf{z}_1, \mathbf{x} = \mathbf{z}_2, \mathbf{x} = \mathbf{z}_3, \mathbf{x} = \mathbf{z}_4 \end{aligned} \tag{4}$$

Let $\mathcal{L}_{\rho_{1-9}}$ be the augmented Lagrangian function of problem (4). We define it as follows:

$$\begin{aligned} & \mathcal{L}(\mathbf{x}, \mathbf{w}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5, y_6, \mathbf{y}_7, y_8, \mathbf{y}_9) := \\ & \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij} \Big\| \mathbf{s_i} - \mathbf{S}\Lambda_j\mathbf{w} \Big\|_2^2 + \mathbf{y}_1^\top(\Psi^\top\mathbf{x} - \mathbf{1}_n) + \frac{\rho_1}{2}\|\Psi^\top\mathbf{x} - \mathbf{1}_n\|_2^2 + \\ & \mathbb{I}_{\{\mathbf{z}_1 \in S_b\}} + \mathbf{y}_2^\top(\mathbf{x} - \mathbf{z}_1) + \frac{\rho_2}{2}\|\mathbf{x} - \mathbf{z}_1\|_2^2 + \mathbb{I}_{\{\mathbf{z}_2 \in S_2\}} + \mathbf{y}_3^\top(\mathbf{x} - \mathbf{z}_2) + \\ & \frac{\rho_3}{2}\|\mathbf{x} - \mathbf{z}_2\|_2^2 + \mathbf{y}_4^\top(\mathbf{Q}\mathbf{P}^\top\mathbf{x} - \mathbf{u}) + \frac{\rho_4}{2}\|\mathbf{Q}\mathbf{P}^\top\mathbf{x} - \mathbf{u}\|_2^2 + \\ & \mathbf{y}_5^\top\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x} + \frac{\rho_5}{2}\Big\|\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x}\Big\|_2^2 + \\ & y_6(\mathbf{z}_3^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} - v) + \frac{\rho_6}{2}\|\mathbf{z}_3^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} - v\|_2^2 + \mathbf{y}_7^\top(\mathbf{x} - \mathbf{z}_3) + \\ & \frac{\rho_7}{2}\|\mathbf{x} - \mathbf{z}_3\|_2^2 + y_8(\mathbf{z}_4^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x}) + \frac{\rho_8}{2}\|\mathbf{z}_4^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x}\|_2^2 + \\ & \mathbf{y}_9^\top(\mathbf{x} - \mathbf{z}_4) + \frac{\rho_9}{2}\|\mathbf{x} - \mathbf{z}_4\|_2^2 \end{aligned} \tag{5}$$

where the $\mathbf{y}$ variables are the Lagrange multipliers of the corresponding constraints, $\mathbb{I}$ is the indicator function that penalizes infeasible $\mathbf{z}_1$ and $\mathbf{z}_2$, and $\rho_{1-9} \geq 0$ are the penalty parameters. In our experiments, we set all the ρ coefficients to the same value. The iterative ADMM steps for problem (4) are described in Algorithm 1. ADMM updates are performed by optimizing for the set of primal variables one at a time, while keeping the rest of the primal and dual variables fixed. Then, the dual variables are updated using gradient ascent on the corresponding dual problem. For more details on this method, we refer the reader to (Wu and Ghanem 2016).

Now, we focus on the subproblems of ℓ_2-Box ADMM that need to be solved at each iteration. We only show the final update rules for each subproblem here, but the exact derivations can be found in the **supplementary material**.

Algorithm 1: ADMM for Solving Problem (4)

Input : Set $\mathbf{S} \in \mathbb{R}^{d\times n}$. Set ρ_{1-9}, $\mathbf{y}_{1-5,7,9} = \mathbf{0}$, $y_{6,8} = 0$, $\mathbf{x}_{\text{kmeans}}$, $\mathbf{w} = \mathbf{P}^\top\mathbf{x} \odot \text{diag}^{-1}(\mathbf{C}\mathbf{x})\mathbf{1}_{nk}$.
Output: $\mathbf{x}$
while not converged **do**
  update: $\mathbf{x}$ by solving Eq (6).
  update: $\mathbf{w}$ by solving Eq (7).
  update: $\mathbf{z}_{1-4}$ via Eqs (8,9, 10,11).
  update: $\mathbf{y}_{1-5,7,9}, y_{6,8}$ via Eqs (12).
end

Update $\mathbf{x}$: We need to solve the following linear system using the conjugate gradient method.

$$\Big(\rho_1\Psi\Psi^\top + (\rho_2+\rho_3+\rho_7+\rho_9)\mathbf{I}_{nk} + \rho_4\mathbf{P}\mathbf{Q}^\top\mathbf{Q}\mathbf{P}^\top + \rho_5\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)^\top\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big) + \rho_6\mathbf{E}_2^\top\mathbf{E}_1\mathbf{z}_3\mathbf{z}_3^\top\mathbf{E}_1^\top\mathbf{E}_2 + \rho_8\mathbf{E}_4^\top\mathbf{E}_4\mathbf{z}_4\mathbf{z}_5^\top\mathbf{E}_3^\top\mathbf{E}_4\Big)\mathbf{x} = -\Big(\text{vect}(\mathbf{B}) + \Psi\mathbf{y}_1 + \mathbf{y}_2 + \mathbf{y}_3 - \rho_1\Psi\mathbf{1}_n - \rho_2\mathbf{z}_1 - \rho_3\mathbf{z}_2 - \mathbf{C}^\top\text{diag}(\mathbf{P}\mathbf{w})\mathbf{y}_5 + y_6\mathbf{E}_2^\top\mathbf{E}_1\mathbf{z}_4 - \rho_6 v\mathbf{E}_2^\top\mathbf{E}_1\mathbf{z}_3 + \mathbf{y}_7 - \rho_7\mathbf{z}_3 + y_8\mathbf{E}_4^\top\mathbf{E}_3\mathbf{z}_4 + \mathbf{y}_9 - \rho_9\mathbf{z}_4 + \mathbf{P}\mathbf{Q}^\top\mathbf{y}_4 - \rho_4\mathbf{P}\mathbf{Q}^\top\mathbf{u}\Big) \quad (6)$$

where $\mathbf{B}(i,j) = \|\mathbf{s_i} - \mathbf{S}\Lambda_j\mathbf{w}\|_2^2$, and $\text{vect}(\mathbf{B})$ is simply a columnwise vectorization of the matrix $\mathbf{B}$.

Update $\mathbf{w}$: We need to solve the following linear system using the conjugate gradient method.

$$\Big[\sum_j^k\sum_i^n 2x_{ij}\Lambda_j^\top\mathbf{S}^\top\mathbf{S}\Lambda_j + \rho_5\mathbf{P}^\top\text{diag}\Big(\mathbf{C}\mathbf{x}\odot\mathbf{C}\mathbf{x}\Big)\mathbf{P}\Big]\mathbf{w} = \sum_j^k\sum_i^n 2x_{ij}\Lambda_j^\top\mathbf{S}^\top\mathbf{s}_i + \mathbf{P}^\top\mathbf{C}\mathbf{x}\odot\mathbf{y}_5 + \rho_5\mathbf{P}^\top\mathbf{C}\mathbf{x}\odot\mathbf{x} \quad (7)$$

Update $\mathbf{z}_1$: Here, we need to perform a simple projection onto the box: $S_b = \{\mathbf{a} : \mathbf{0} \le \mathbf{a} \le \mathbf{1}\}$. This projection is an elementwise clamping between 0 and +1.

$$\mathbf{z}_1 = \mathbf{P}_{S_b}\left(\mathbf{x} + \frac{\mathbf{y}_2}{\rho_2}\right) = \min\left(\max\left(\mathbf{x} + \frac{\mathbf{y}_2}{\rho_2}, \mathbf{0}\right), \mathbf{1}\right) \quad (8)$$

Update $\mathbf{z}_2$: We need to perform a simple projection onto the ℓ_2-sphere: $S_2 = \{\mathbf{a} \in \mathbb{R}^{nk} : \|\mathbf{a} - \frac{1}{2}\mathbf{1}\|_2^2 = \frac{nk}{4}\}$. This involves an elementwise shift and ℓ_2 vector normalization.

$$\mathbf{z}_2 = \mathbf{P}_{S_2}\left(\mathbf{x} + \frac{\mathbf{y}_3}{\rho_3}\right) = \frac{\sqrt{nk}}{2}\frac{\left(\mathbf{x} + \frac{\mathbf{y}_3}{\rho_3}\right) - \frac{1}{2}\mathbf{1}}{\left\|\left(\mathbf{x} + \frac{\mathbf{y}_3}{\rho_3}\right) - \frac{1}{2}\mathbf{1}\right\|_2} + \frac{1}{2}\mathbf{1} \quad (9)$$

Update $\mathbf{z}_3$: We need to solve the following linear system using the conjugate gradient method.

$$\Big[\rho_6\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x}\mathbf{x}^\top\mathbf{E}_2^\top\mathbf{E}_1 + \rho_7\mathbf{I}_{nk}\Big]\mathbf{z}_3 = \mathbf{y}_7 + \rho_7\mathbf{x} - y_6\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} + \rho_6 v\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} \quad (10)$$

Update $\mathbf{z}_4$: We need to solve the following linear system using the conjugate gradient method.

$$\Big[\rho_8\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x}\mathbf{x}^\top\mathbf{E}_4^\top\mathbf{E}_3 + \rho_9\mathbf{I}_{nk}\Big]\mathbf{z}_4 = \mathbf{y}_9 + \rho_9\mathbf{x} - y_8\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x} \quad (11)$$

Update $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5, y_6, \mathbf{y}_7, y_8, \mathbf{y}_9$: Lastly, we need to perform dual ascent on the dual variables as follows:

$$\begin{aligned}
&\mathbf{y}_1 \leftarrow \mathbf{y}_1 + \rho_1\Big(\Psi^\top\mathbf{x} - \mathbf{1}_n\Big), \quad \mathbf{y}_2 \leftarrow \mathbf{y}_2 + \rho_2\Big(\mathbf{x} - \mathbf{z}_2\Big)\\
&\mathbf{y}_3 \leftarrow \mathbf{y}_3 + \rho_3\Big(\mathbf{x} - \mathbf{z}_3\Big), \quad \mathbf{y}_4 \leftarrow \mathbf{y}_4 + \rho_4\Big(\mathbf{Q}\mathbf{P}^\top\mathbf{x} - \mathbf{u}\Big)\\
&\mathbf{y}_5 \leftarrow \mathbf{y}_5 + \rho_5\Big(\mathbf{x} - \mathbf{P}\mathbf{w}\odot\mathbf{C}\mathbf{x}\Big), \quad y_6 \leftarrow y_6 + \rho_6\Big(\mathbf{z}_3^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} - v\Big)\\
&\mathbf{y}_7 \leftarrow \mathbf{y}_7 + \rho_7\Big(\mathbf{x} - \mathbf{z}_3\Big), \quad y_8 \leftarrow y_8 + \rho_8\Big(\mathbf{z}_4^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x}\Big)\\
&\mathbf{y}_9 \leftarrow \mathbf{y}_9 + \rho_9\Big(\mathbf{x} - \mathbf{z}_4\Big)
\end{aligned} \quad (12)$$

The ℓ_2-Box ADMM iterations are run until convergence (i.e. when the relative change in $\mathbf{x}$ between iterations is below a threshold). When the method converges, all the primal variables ($\mathbf{x}$ and $\mathbf{z}_{1-4}$) converge to the same feasible binary vector. The convergence properties/guarantee for this method are detailed in (Wu and Ghanem 2016).

Experiments

In this section, we conduct extensive experiments to motivate and evaluate our proposed clustering method, both on synthetic and real datasets. We also compare our method against other constrained clustering methods on well-known benchmarks, thus, demonstrating superior performance and flexibility, as well as, superior gain that can be achieved when both categories of constraints (cardinality and pairwise) are combined in our framework.

1. Datasets and Implementation Details. The datasets used in this section vary from synthetic to real. As for the synthetic ones, we construct two datasets, one is cluster balanced (denoted as `Balanced`) and another is imbalanced (denoted as `ImBalanced`) as shown in Figure 1. Each dataset comprises 700 data points with 2 clusters. In `Balanced`, each cluster has exactly 350 data points, while in `ImBalanced` one cluster has 600 data points and the other contains 100. As for the real datasets, we make use of various popular UCI datasets (Bradley, Bennett, and Demiriz 2000), e.g. iris, wine, glass, ionoshphere, hehpitates/hehpitates1, and Breast Cancer Wisconsin-Diagnostic. Following convention, data points are normalized to have a value in $[-1, +1]$. For hehpitates and hehpitates1, we remove all points with missing or none categorical features.

As for the implementation details, none of the selection matrices used in the proposed framework (i.e. $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4, \mathbf{P}, \mathbf{C}, \mathbf{Q}, \Psi, \Psi^T, \Lambda_j$) are actually constructed. Only element indexing within vectors is used, thus, keeping the necessary computation cost minimal. We set all ρ parameters to the same value 20 and we increase it every 5 iterations by 10% for all real datasets. Moreover, we initialize all the optimization variables using zero vectors, while

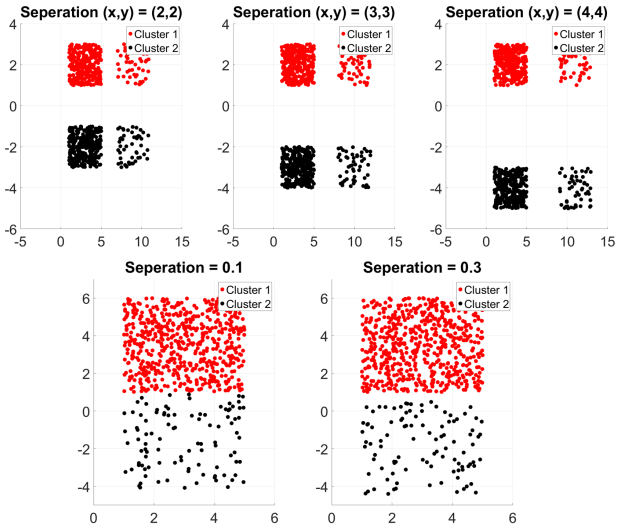

Figure 1: The *Balanced* and *ImBalanced* in the two consecutive rows respectively. They comprise two clusters (red/black) with an increasing separation between clusters.

$\mathbf{x}$ is initialized to random (i.e. random assignment of data points to clusters) if the comparison is against K-means. When comparing against other clustering methods, we use the same K-means initialization as other methods. In all comparisons, $\mathbf{w}$ is initialized to a feasible point as given in Algorithm 1. MATLAB is used to implement our method. The most expensive operation in our framework is the $\mathbf{x}$ and $\mathbf{w}$ updates, which involve solving an $n \times k$ linear system. This is the bottleneck of our framework causing it to have a computational complexity $\mathcal{O}(n^3k^3)$ per iteration. In the final experiment, we report the runtime of our framework on different sized datasets with a variety of constraint choices.

As for the evaluation metric, we adopt a common criterion used in the clustering community to compare different clustering methods, namely *RandInd*), which calculates a measure of agreement between two partitions of a dataset. In all experiments, clustering is repeated 10 times with different initializations and we report the average and standard deviation *RandInd*.

2. Comparing Different Constraint Design Choices. We apply our flexible constrained K-means (FCKm) on the same clustering task with several choices of constraints: no constraints, only cardinality constraints, only pairwise constraints, and both types jointly. We refer to each as FCKm, FCKm-Car, FCKm-Pair and FCKm-Mix respectively.

(i) Traditional K-means versus FCKm. Here, we show that our FCKm method attains a very similar, if not better, performance than traditional K-means (builtin MATLAB function). Experiments are conducted on some of the UCI datasets (Bradley, Bennett, and Demiriz 2000) (wine, iris, and glass). Table 1 reports the K-means objective value and the *RandInd*(%) metric for both methods.

(ii) Traditional K-means versus FCKm-Car. Here, we demonstrate that our framework coupled with only cardinality constraints outperforms traditional unconstrained K-means on a variety of synthetic data. This highlights the importance of having this prior information available and harnessing it in the clustering process. In these experiments, the cardinality constraints are generated from ground truth labels. To show that cardinality does in fact help clustering performance, we apply FCKm-Car on the two synthetic datasets (*Balanced* and *ImBalanced*) and report their *RandInd* results in Table 2.

Table 1: Comparison between traditional K-means and FCKm on real UCI datasets using both K-means objective value and *RandInd*(%).

Datasets	K-mean (Obj Value)	FCKm (Obj Value)
wine	195.903 ± 0.071	**195.816** ± 0.002
iris	30.0624 ± 4.768	**28.2641** ± 0
glass	**78.1381** ± 3.301	84.3469 ± 1.405
	K-means (*RandInd%*)	FCKm (*RandInd%*)
wine	93.90 ± 0.74	**94.42** ± 0.35
iris	85.31 ± 4.69	**87.97** ± 0
glass	67.74 ± 0.18	**68.07** ± 1.5

Table 2: Comparison between K-means and FCKm-Car on synthetic datasets using *RandInd(%)*.

Datasets	K-means	FCKm-Car
Balanced (x,y)=(2,2)	74.96 ± 26.39	**100** ± 0
Balanced (x,y)=(3,3)	84.98 ± 24.19	**100** ± 0
Balanced (x,y)=(4,4)	74.96 ± 26.39	**100** ± 0
Imbalanced y=0.1	69.18 ± 0	**98.89** ± 2.34
Imbalanced y=0.3	67.78 ± 0	**100** ± 0

For the *Balanced* dataset, the separation between the four groups of points increases. In fact, K-means tends to cluster points together such that each cluster has a similar variance as other clusters. Consequently, K-means clusters the high-density points of the *Balanced* dataset together and groups the remaining less dense points into another cluster. In comparison, our framework exploits the cardinality constraints to achieve perfect clustering performance. Similarly, the *ImBalanced* dataset contains two imbalanced clusters with very different densities, where the separation between them is increased. In this case, K-means often mixes data points between clusters, since the cardinality constraints are not used. On the other hand, FCKm-Car can almost perfectly predict the ground truth clustering labels. Interestingly, the variance of our results is much lower than that of K-means even though they both use the same clustering initialization. This indicates that the cardinality constraints afford our method robustness to the initialization.

To the best of our knowledge, all previous work that handles generic cardinality constraints do not have readily available code for comparison. Therefore, we only compare our method with traditional unconstrained K-means, so as to demonstrate the effectiveness of adding cardinality constraints to an unconstrained clustering method.

(iii) Pairwise Constrained Clustering Methods versus FCKm-Pair. Here, we compare our FCKm-Pair method against several pairwise constrained methods from the literature, namely Spectral Clustering (Lu and Ip 2010), Penalized probabilistic Clustering (PPC) (Lu and Leen 2007), and CCL (Dai et al. 2003). Among these methods, only CCL ex-

Table 3: Comparison of several pairwise constrained clustering methods against FCKm-Pair using *RandInd(%)* (RI), as well as, must-link (MLV) and cannot-link (CLV) violations in the constraints. See text for more details.

	Constraints	Spectral Clustering			PPC			CCL			FCKm-Pair		
		RI	MLV	CLV	RI	MLV	CLV	RI	MLV	CLV	RI	MLV	CLV
wine	20ml, 20cl	**97.71** ± 0	3	1	94.26 ± 0.6	4.7	0.9	71.07 ± 0	1	0	95.17 ± 2.7	0	0
	40ml, 40cl	97.71 ± 0	3	1	96.49 ± 0.6	6.2	0.8	99.18 ± 0	0	0	**97.96** ± 0.6	0	0
	60ml, 60cl	97.71 ± 0	4	1	94.06 ± 0.6	12	0.2	83.29 ± 0	2	0	**100** ± 0	0	0
	80ml, 80cl	96.98 ± 0	5	3	96.06 ± 0.3	10.4	4	100 ± 0	0	0	**100** ± 0	0	0
	100ml, 100cl	96.98 ± 0	5	3	96.19 ± 0	8.1	2.1	78.10 ± 0	7	0	**100** ± 0	0	0
iris	20ml, 20cl	86.23 ± 0	0	10	**97.40** ± 0	2	0	70.58 ± 0	2	3	89.15 ± 7.1	0	0
	40ml, 40cl	85.68 ± 0	0	15	**98.25** ± 2	1	0	76.03 ± 0	3	1	94.61 ± 1.9	0	0
	60ml, 60cl	85.68 ± 0	0	20	99.11 ± 0	0	1	78.48 ± 0	2	0	**99.66** ± 1.1	0	0
	80ml, 80cl	85.68 ± 0	0	20	99.11 ± 0	0	1	75.19 ± 0	2	0	**98.68** ± 2.8	0	0
	100ml, 100cl	85.68 ± 0	0	20	99.11 ± 0	0	1	71.47 ± 0	6	0	**100** ± 0	0	0

Table 4: Effect of using different types of clustering constraints as compared to traditional K-means on several UCI datasets using *RandInd(%)*.

Datasets	K-means	FCKm-Car	Time	FCKm-Pair	Time	FCKm-Mix	Time
iris	85.31 ± 5	87.97 ± 0	1.21 sec	96.85 ± 7.3	1.96 sec	**100** ± 0	2.33 sec
wine	92.99 ± 0.7	95.34 ± 0	1.14 sec	98.86 ± 0.1	2.99 sec	**100** ± 0	4.19 sec
ionoshpere	58.80 ± 0.1	54.52 ± 0	1.50 sec	63.61 ± 0.3	5.00 sec	**67.98** ± 0	6.57 sec
hepatitis	56.62 ± 5	72.24 ± 6	0.55 sec	67.25 ± 3	0.99 sec	**97.53** ± 3	1.09 sec
hepatitis1	58.94 ± 0	64.23 ± 0.8	0.91 sec	64.46 ± 0.9	0.70 sec	**86.81** ± 0	1.26 sec
breast cancer diag	86.60 ± 0	87.51 ± 0	3.36 sec	89.68 ± 0	3.27 sec	**90.63** ± 0	9.27 sec

actly enforces the constraints, while the others incorporate them as soft pairwise constraints in their clustering framework. Consequently, Spectral Clustering and PPC may result in clustering violations. We run all four methods on two UCI datasets (wine, iris) and ensure that all methods receive the same randomly chosen pairwise constraints. Table 3 reports the performance of these methods using the *RandInd* criterion. For each experiment, we also report the number of must-link (ml) and cannot-link (cl) constraints, as well as, the number of must-link violations (MLV) and the cannot-link violations (CLV). It is clear that FCKm-Pair outperforms all other methods, while satisfying all the constraints.

(iv) K-means versus FCKm-Mix. Here, we demonstrate the main motivation behind our flexible framework, namely its ability to incorporate both cardinality and pairwise constraints simultaneously in the clustering optimization. Firstly, we demonstrate that increasing the number of pairwise constraints (either must-link or cannot-link) with the same cardinality constraints consistently improves performance. We conduct this experiment on two different datasets: one synthetic (*ImBalanced* `y=0.1`) and one real (wine). Figure 2 compares our method against traditional K-means in such a setup. Obviously, K-means does not benefit from the constraints while ours consistently improves in performance. Secondly, we compare all three variants of our framework, i.e. cardinality only constraints (FCKm-Car), pairwise only constraints (FCKm-Pair), and both (FCKm-Mix), on several UCI datasets (iris, wine, ionoshpere, hepatitis, hepatitis1, and Breast Cancer Wisconsin-Diagnostic). Results in Table 4 show that our method performs increasingly better (and becomes more robust to different K-means initializations), when more constraint categories are used simultaneously. This improvement reaches as high as 40% in *RandInd* for some datasets. We did not compare against other methods here, since there is no existing work that combines both categories of constraints in a unified framework and extending the pairwise constrained methods to cardinality constraints is not trivial.

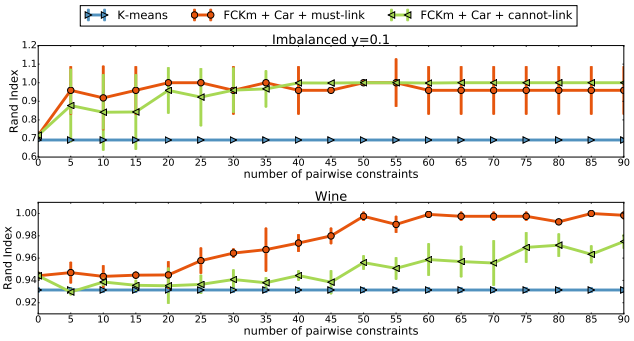

Figure 2: Effect of increasing must-link and cannot-link constraints separately, as compared to unconstrained K-means.

Conclusions

In this paper, we propose a new flexible framework to handle both pairwise and cardinality constraints for K-means clustering. The resulting integer program is transformed into an equivalent continuous reformulation that is solved using ADMM. Extensive experiments have been conducted on both synthetic and real datasets to demonstrate the competitive performance of our method under different constraint choices. Also, we show that our framework achieves state-of-art performance on popular datasets when both types of constraints are used simultaneously.

References

Basu, S.; Bilenko, M.; and Mooney, R. J. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 59–68. ACM.

Bradley, P.; Bennett, K.; and Demiriz, A. 2000. Constrained k-means clustering. *Microsoft Research, Redmond* 1–8.

Chapelle, O., and Zien, A. 2005. Semi-supervised classification by low density separation. In *AISTATS*, 57–64.

Dai, B.-R.; Lin, C.-R.; Chen, M.-S.; et al. 2003. On the techniques for data clustering with numerical constraints. *Age* 23(13):10.

Dhillon, I. S.; Guan, Y.; and Kulis, B. 2004. *A unified view of kernel k-means, spectral clustering and graph cuts*. Citeseer.

Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine learning* 2(2):139–172.

Höppner, F., and Klawonn, F. 2008. Clustering with size constraints. In *Computational Intelligence Paradigms*, 167–180. Springer.

Karypis, G.; Han, E.-H.; and Kumar, V. 1999. Chameleon: Hierarchical clustering using dynamic modeling. volume 32, 68–75. IEEE.

Klawonn, F., and Höppner, F. 2006. Equi-sized, homogeneous partitioning. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 70–77. Springer.

Klein, D.; Kamvar, S. D.; and Manning, C. D. 2002. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. Stanford.

Lu, Z., and Ip, H. H. 2010. Constrained spectral clustering via exhaustive and efficient constraint propagation. In *European Conference on Computer Vision*, 1–14. Springer.

Lu, Z., and Leen, T. K. 2007. Penalized probabilistic clustering. *Neural Computation* 19(6):1528–1567.

Ng, A. Y.; Jordan, M. I.; Weiss, Y.; et al. 2002. On spectral clustering: Analysis and an algorithm. volume 2, 849–856. MIT; 1998.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. volume 22, 888–905. IEEE.

Von Luxburg, U. 2007. A tutorial on spectral clustering. volume 17, 395–416. Springer.

Wagstaff, K., and Cardie, C. 2000. Clustering with instance-level constraints. volume 1097.

Wagstaff, K.; Cardie, C.; Rogers, S.; Schrödl, S.; et al. 2001. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, 577–584.

Wu, B., and Ghanem, B. 2016. ℓ_p-box admm: A versatile framework for integer programming. volume abs/1604.07666.

Wu, B., and Hu, B. 2011. Density and neighbor adaptive information theoretic clustering. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 230–237. IEEE.

Wu, B.; Lyu, S.; Hu, B.-G.; and Ji, Q. 2013a. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2856–2863.

Wu, B.; Zhang, Y.; Hu, B.-G.; and Ji, Q. 2013b. Constrained clustering and its application to face clustering in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3507–3514.

Constrained K-means with General Pairwise and Cardinality Constraints (Supplementary Material)

Because of limited space, we could not include all details (e.g. derivations, reformulations, and auxiliary results) in the main manuscript. In this section, we provide these details and clarify the main algorithmic steps done in the paper.

Reformulation of Constrained K-means

We start with our formulation of constrained K-means in Equation (1) below (or Equation (4) in the manuscript).

$$\begin{aligned}&\min_{\mathbf{x},\mathbf{w},\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3,\mathbf{z}_4} \quad \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij}\Big\|\mathbf{s_i}-\mathbf{S}\Lambda_j\mathbf{w}\Big\|_2^2\\ &\text{s.t. } \Psi^\top\mathbf{x}=\mathbf{1}_n,\ \ \mathbf{z}_1\in S_b,\ \ \mathbf{z}_2\in S_2,\ \ \mathbf{Q}\mathbf{P}^\top\mathbf{x}=\mathbf{u}\\ &\mathbf{x}=\mathbf{P}\mathbf{w}\odot\mathbf{C}\mathbf{x},\ \ (\mathbf{E}_1\mathbf{z}_3)^\top\mathbf{E}_2\mathbf{x}=v,\ \ (\mathbf{E}_3\mathbf{z}_4)^\top\mathbf{E}_4\mathbf{x}=0\\ &\mathbf{x}=\mathbf{z}_1,\mathbf{x}=\mathbf{z}_2,\mathbf{x}=\mathbf{z}_3,\mathbf{x}=\mathbf{z}_4\end{aligned} \tag{1}$$

We define $\mathbf{X}\in\mathbb{R}^{k\times n}$ and therefore $\mathbf{x}=vect(\mathbf{X})$. The $vect$ operator simply vectorizes the matrix one column at a time (i.e. one data point at a time). The order of concatenation is reverse for the label vector: $\mathbf{w}=vect(\mathbf{W})$, where $\mathbf{W}\in\mathbb{R}^{n\times k}$ (i.e. the vectorization is done one cluster at a time). Therefore, it is important to point out that the order of the binary labels $\mathbf{x}$ and that of $\mathbf{w}$ is swapped. Reordering these vectors based on data points or clusters is controlled by the permutation matrix $\mathbf{P}\in\mathbb{R}^{nk\times nk}$. Therefore, $\mathbf{P}\mathbf{x}=\mathbf{P}^\top\mathbf{x}=vect(\mathbf{X}^\top)$ and $\mathbf{P}\mathbf{w}=\mathbf{P}^\top\mathbf{w}=vect(\mathbf{W}^\top)$. Throughout the main manuscript and the supplementary material, we use $\mathbf{P}$ and $\mathbf{P}^\top$ to change the order of $\mathbf{w}$ to the same order as $\mathbf{x}$ and vice versa. Matrix $\mathbf{S}\in\mathbb{R}^{d\times n}$ in Equation (1) simply concatenates all the data points in its columns, where $\Lambda_j\in\mathbb{R}^{n\times nk}$ is zero everywhere except for the j^{th} block that is identity, i.e. $\Lambda_j=\begin{bmatrix}\mathbf{0} & \cdots & \mathbf{I}_k^j & \cdots & \mathbf{0}\end{bmatrix}$.

As for $\Psi^\top\in\mathbb{R}^{n\times nk}$, it is a binary matrix that has in each row a vector $\mathbf{1}_k^\top$ that sums all the binary labels for a given data point while the rest are zeros. We write this matrix in blockwise form as follows:

$$\Psi^\top=\begin{bmatrix}\mathbf{1}_k^\top & \mathbf{0}_k & \cdots\\ \mathbf{0}_k^\top & \mathbf{1}_k^\top & \mathbf{0}_k\\ \vdots & \vdots & \vdots\\ \mathbf{0}_k & \cdots & \mathbf{1}_k\end{bmatrix} \tag{2}$$

As for $\mathbf{Q}\in\mathbb{R}^{k\times nk}$, it sums all the binary labels of each cluster at a time for all the data points and its blockwise matrix form is given as follows:

$$\mathbf{Q}=\begin{bmatrix}\mathbf{1}_n^T & \mathbf{0}_{nk-n}^T & \cdots\\ \mathbf{0}_n^T & \mathbf{1}_n^T & \mathbf{0}_{nk-n}^T & \cdots\\ \cdots\end{bmatrix} \tag{3}$$

As for $\mathbf{C}\in\mathbb{R}^{nk\times nk}$, it sums the binary labels for each cluster at a time and its blockwise matrix form is as follows:

$$\mathbf{C}=\begin{bmatrix}\mathbf{I}_k & \cdots & \mathbf{I}_k\\ \vdots & \ldots & \vdots\\ \mathbf{I}_k & \cdots & \mathbf{I}_k\end{bmatrix} \tag{4}$$

As for the box and ℓ_2-sphere constraints (whose intersection is the binary vector space), we define two sets: $S_b\coloneqq\{\mathbf{a}:\mathbf{0}\le\mathbf{a}\le\mathbf{1}\}$ and $S_2\coloneqq\{\mathbf{a}\in\mathbb{R}^n:\|\mathbf{a}-\frac{1}{2}\mathbf{1}\|_2^2=\frac{n}{2}\}$, respectively. It is shown in (Wu and Ghanem 2016) that: $\{0,1\}^n=S_b\cap S_2$.

Lastly, $\mathbf{E}_1,\mathbf{E}_2\in\mathbb{R}^{kv\times nk}$ and $\mathbf{E}_3,\mathbf{E}_4\in\mathbb{R}^{ke\times nk}$ are selection matrices for the must-link and cannot-link constraints respectively. They select the data points that are involved in both types of constraints.

Applying ADMM to Equation (1)

Following the conventional treatment of an optimization problem using ADMM, we first formulate the augmented Lagrangian function for problem (1) as follows:

$$\begin{aligned}&\mathcal{L}(\mathbf{x},\mathbf{w},\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3,\mathbf{z}_4,\mathbf{y}_1,\mathbf{y}_2,\mathbf{y}_3,\mathbf{y}_4,\mathbf{y}_5,y_6,\mathbf{y}_7,y_8,\mathbf{y}_9)\coloneqq\\ &\sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij}\Big\|\mathbf{s_i}-\mathbf{S}\Lambda_j\mathbf{w}\Big\|_2^2+\mathbf{y}_1^\top(\Psi^\top\mathbf{x}-\mathbf{1}_n)+\frac{\rho_1}{2}\|\Psi^\top\mathbf{x}-\mathbf{1}_n\|_2^2+\\ &\mathbb{I}_{\{\mathbf{z}_1\in S_b\}}+\mathbf{y}_2^\top(\mathbf{x}-\mathbf{z}_1)+\frac{\rho_2}{2}\|\mathbf{x}-\mathbf{z}_1\|_2^2+\mathbb{I}_{\{\mathbf{z}_2\in S_2\}}+\mathbf{y}_3^\top(\mathbf{x}-\mathbf{z}_2)+\\ &\frac{\rho_3}{2}\|\mathbf{x}-\mathbf{z}_2\|_2^2+\mathbf{y}_4^\top(\mathbf{Q}\mathbf{P}^\top\mathbf{x}-\mathbf{u})+\frac{\rho_4}{2}\|\mathbf{Q}\mathbf{P}^\top\mathbf{x}-\mathbf{u}\|_2^2+\\ &\mathbf{y}_5^\top\Big(\mathbf{I}-\mathrm{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x}+\frac{\rho_5}{2}\Big\|\Big(\mathbf{I}-\mathrm{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x}\Big\|_2^2+\\ &y_6(\mathbf{z}_3^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x}-v)+\frac{\rho_6}{2}\|\mathbf{z}_3^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x}-v\|_2^2+\mathbf{y}_7^\top(\mathbf{x}-\mathbf{z}_3)+\\ &\frac{\rho_7}{2}\|\mathbf{x}-\mathbf{z}_3\|_2^2+y_8(\mathbf{z}_4^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x})+\frac{\rho_8}{2}\|\mathbf{z}_4^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x}\|_2^2+\\ &\mathbf{y}_9^\top(\mathbf{x}-\mathbf{z}_4)+\frac{\rho_9}{2}\|\mathbf{x}-\mathbf{z}_4\|_2^2\end{aligned} \tag{5}$$

ADMM updates steps tend to update each primal variable ($\mathbf{x}$, $\mathbf{w}$, and $\mathbf{z}_{1-4}$) sequentially, while keeping the rest of these variables and the dual variables ($\mathbf{y}_{1-5}$, $\mathbf{y}_{6,8}$, and $\mathbf{y}_{6,9}$) set to their most recent values. After the primal variables are updated, the dual variables are updated via a single gradient ascent step. Next, we detail each update step and the underlying optimization sub-problem that needs to be solved.

Update $\mathbf{x}$:

$$\begin{aligned}
&\mathbf{x} \leftarrow \arg\min_{\mathbf{x}} \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij}\Big\|\mathbf{s_i} - \mathbf{S}\Lambda_j\mathbf{w}\Big\|_2^2 + \mathbf{y}_1^\top(\Psi^\top\mathbf{x} - \mathbf{1}_n) + \\
&\frac{\rho_1}{2}\|\Psi^\top\mathbf{x} - \mathbf{1}_n\|_2^2 + \mathbf{y}_2^\top(\mathbf{x} - \mathbf{z}_1) + \frac{\rho_2}{2}\|\mathbf{x} - \mathbf{z}_1\|_2^2 + \mathbf{y}_3^\top(\mathbf{x} - \mathbf{z}_2) + \\
&\frac{\rho_3}{2}\|\mathbf{x} - \mathbf{z}_2\|_2^2 + \mathbf{y}_4^\top(\mathbf{Q}\mathbf{P}^\top\mathbf{x} - \mathbf{u}) + \frac{\rho_4}{2}\|\mathbf{Q}\mathbf{P}^\top\mathbf{x} - \mathbf{u}\|_2^2 + \\
&\mathbf{y}_5^\top\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x} + \frac{\rho_5}{2}\|\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x}\|_2^2 + \\
&y_6(\mathbf{z}_4^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} - v) + \frac{\rho_6}{2}\|\mathbf{z}_4^\top\mathbf{E}_1^\top\mathbf{E}_2\mathbf{x} - v\|_2^2 + \mathbf{y}_7^\top(\mathbf{x} - \mathbf{z}_4) + \\
&\frac{\rho_7}{2}\|\mathbf{x} - \mathbf{z}_4\|_2^2 + y_8(\mathbf{z}_5^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x}) + \frac{\rho_8}{2}\|\mathbf{z}_5^\top\mathbf{E}_3^\top\mathbf{E}_4\mathbf{x}\|_2^2 + \mathbf{y}_9^\top(\mathbf{x} - \mathbf{z}_5) + \\
&\frac{\rho_9}{2}\|\mathbf{x} - \mathbf{z}_5\|_2^2 \qquad (6)
\end{aligned}$$

Problem (6) is strongly convex quadratic in $\mathbf{x}$. Therefore, a stationary point is necessary and sufficient for optimality. By equating the gradient of problem (6) to zero, we get:

$$\begin{aligned}
&\Big(\rho_1\Psi\Psi^\top + (\rho_2 + \rho_3 + \rho_7 + \rho_9)\mathbf{I}_{nk} + \rho_4\mathbf{P}\mathbf{Q}^\top\mathbf{Q}\mathbf{P}^\top + \\
&\rho_5\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)^\top\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big) + \\
&\rho_6\mathbf{E}_2^\top\mathbf{E}_1\mathbf{z}_3\mathbf{z}_3^\top\mathbf{E}_1^\top\mathbf{E}_2 + \rho_8\mathbf{E}_4^\top\mathbf{E}_4\mathbf{z}_4\mathbf{z}_5^\top\mathbf{E}_3^\top\mathbf{E}_4\Big)\mathbf{x} = \\
&-\Big(\text{vect}(\mathbf{B}) + \Psi\mathbf{y}_1 + \mathbf{y}_2 + \mathbf{y}_3 - \rho_1\Psi\mathbf{1}_n - \rho_2\mathbf{z}_1 - \\
&\rho_3\mathbf{z}_2 - \mathbf{C}^\top\text{diag}(\mathbf{P}\mathbf{w})\mathbf{y}_5 + y_6\mathbf{E}_2^\top\mathbf{E}_1\mathbf{z}_4 - \rho_6 v\mathbf{E}_2^\top\mathbf{E}_1\mathbf{z}_3 + \mathbf{y}_7 - \\
&\rho_7\mathbf{z}_3 + y_8\mathbf{E}_4^\top\mathbf{E}_3\mathbf{z}_4 + \mathbf{y}_9 - \rho_9\mathbf{z}_4 + \mathbf{P}\mathbf{Q}^\top\mathbf{y}_4 - \rho_4\mathbf{P}\mathbf{Q}^\top\mathbf{u}\Big) \qquad (7)
\end{aligned}$$

where $\mathbf{B}(i,j) = \|\mathbf{s_i} - \mathbf{S}\Lambda_j\mathbf{w}\|_2^2$.

Update $\mathbf{w}$:

$$\begin{aligned}
&\mathbf{w} \leftarrow \arg\min_{\mathbf{w}} \sum_{j=1}^{k}\sum_{i=1}^{n} x_{ij}\Big\|\mathbf{s_i} - \mathbf{S}\Lambda_j\mathbf{w}\Big\|_2^2 + \\
&\mathbf{y}_5^T\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x} + \frac{\rho_5}{2}\|\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x}\|_2^2 \qquad (8)
\end{aligned}$$

Similar to the $\mathbf{x}$-update step, the problem is strongly convex quadratic and finding a stationary point is necessary and sufficient for a global solution. Thus, the gradient of (8) is given by:

$$\begin{aligned}
&-\sum_{j=1}^{k}\sum_{i=1}^{n} 2x_{ij}(\mathbf{S}\Lambda_j)^T(\mathbf{s}_i - \mathbf{S}\Lambda_j\mathbf{w}) - \mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{y}_5 - \\
&\rho_5\Big(\mathbf{P}^T\text{diag}(\mathbf{C}\mathbf{x})\Big)\Big(\mathbf{I} - \text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\Big)\mathbf{x} = 0 \qquad (9)
\end{aligned}$$

Then:

$$\begin{aligned}
&-\sum_{j}^{k}\sum_{i}^{n} 2x_{ij}\Lambda_j^T\mathbf{S}^T\mathbf{s}_i + \sum_{j}^{k}\sum_{i}^{n} 2x_{ij}\Lambda_j^T\mathbf{S}^T\mathbf{S}\Lambda_j\mathbf{w} - \mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{y}_5 \\
&- \rho_5\mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{x} + \rho_5\mathbf{P}^T\text{diag}(\mathbf{C}\mathbf{x})\text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\mathbf{x} = 0
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\Big[\sum_{j}^{k}\sum_{i}^{n} 2x_{ij}\Lambda_j^T\mathbf{S}^T\mathbf{S}\Lambda_j\mathbf{w} + \rho_5\mathbf{P}^T\text{diag}(\mathbf{C}\mathbf{x})\text{diag}(\mathbf{P}\mathbf{w})\mathbf{C}\mathbf{x}\Big] \\
&= \sum_{j}^{k}\sum_{i}^{n} 2x_{ij}\Lambda_j^T\mathbf{S}^T\mathbf{s}_i + \mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{y}_5 + \rho_5\mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{x}
\end{aligned}$$

And finally, we have:

$$\begin{aligned}
&\Big[\sum_{j}^{k}\sum_{i}^{n} 2x_{ij}\Lambda_j^T\mathbf{S}^T\mathbf{S}\Lambda_j + \rho_5\mathbf{P}^T\text{diag}\Big(\mathbf{C}\mathbf{x} \odot \mathbf{C}\mathbf{x}\Big)\mathbf{P}\Big]\mathbf{w} \\
&= \sum_{j}^{k}\sum_{i}^{n} 2x_{ij}\Lambda_j^T\mathbf{S}^T\mathbf{s}_i + \mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{y}_5 + \rho_5\mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{x}
\end{aligned}$$

In this derivation, we use the fact that $\nabla_{\mathbf{w}}\Big(\mathbf{y}_5^T\mathbf{P}\mathbf{w} \odot \mathbf{C}\mathbf{x}\Big) = \mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{y}_5$. We discuss why this is true next. Note the following identities: $\mathbf{a} \odot \mathbf{b} = \mathbf{b} \odot \mathbf{a} = \text{diag}(\mathbf{a})\mathbf{b} = \text{diag}(\mathbf{b})\mathbf{a}$. Therefore,

$$\begin{aligned}
\nabla_{\mathbf{w}}\Big(\mathbf{y}_5^T\mathbf{P}\mathbf{w} \odot \mathbf{C}\mathbf{x}\Big) &= \nabla_{\mathbf{w}}\Big(\mathbf{y}_5^T\text{diag}(\mathbf{C}\mathbf{x})\mathbf{P}\mathbf{w}\Big) \\
&= \mathbf{P}^T\text{diag}(\mathbf{C}\mathbf{x})\mathbf{y}_5 \\
&= \mathbf{P}^T\mathbf{C}\mathbf{x} \odot \mathbf{y}_5
\end{aligned}$$

Update $\mathbf{z}_1$:

$$\begin{aligned}
&\mathbf{z}_1 \leftarrow \arg\min_{\mathbf{z}_1 \in S_b} \ \mathbf{y}_2^\top(\mathbf{x} - \mathbf{z}_1) + \frac{\rho_2}{2}\|\mathbf{x} - \mathbf{z}_1\|_2^2 \\
\Leftrightarrow\ &\mathbf{z}_1 \leftarrow \arg\min_{\mathbf{z}_1 \in S_b} \ \Big\|\mathbf{z}_1 - \Big(\mathbf{x} + \frac{\mathbf{y}_2}{\rho_2}\Big)\Big\|_2^2 \\
\Leftrightarrow\ &\mathbf{z}_1 = \mathbf{P}_{S_b}\Big(\mathbf{x} + \frac{\mathbf{y}_2}{\rho_2}\Big) \qquad (10)
\end{aligned}$$

Here, we need to perform a simple projection onto the box set S_b. The projection $\mathbf{P}_{S_b}(.)$ is an elementwise clamping between 0 and +1. In fact, $\mathbf{P}_{S_b}(a) = \min(\max(a, 0), 1)$ for a scalar value a.

Update $\mathbf{z}_2$:

$$\begin{aligned}
&\mathbf{z}_2 \leftarrow \arg\min_{\mathbf{z}_2 \in S_2} \ \mathbf{y}_3^\top(\mathbf{x} - \mathbf{z}_2) + \frac{\rho_3}{2}\|\mathbf{x} - \mathbf{z}_2\|_2^2 \\
\Leftrightarrow\ &\mathbf{z}_2 \leftarrow \arg\min_{\mathbf{z}_2 \in S_2} \ \Big\|\mathbf{z}_2 - \Big(\mathbf{x} + \frac{\mathbf{y}_3}{\rho_3}\Big)\Big\|_2^2 \\
\Leftrightarrow\ &\mathbf{z}_2 = \mathbf{P}_{S_2}\Big(\mathbf{x} + \frac{\mathbf{y}_3}{\rho_3}\Big) \qquad (11)
\end{aligned}$$

We need to perform a simple projection onto the ℓ_2-sphere: $S_2 = \{\mathbf{a} \in \mathbb{R}^n : \|\mathbf{a} - \frac{1}{2}\mathbf{1}\|_2^2 = \frac{n}{4}\}$. The projection $\mathbf{P}_{\mathcal{S}_2}(.)$ involves an elementwise shift and ℓ_2 vector normalization. In fact, $\mathbf{P}_{\mathcal{S}_2}(\mathbf{a}) = \frac{\sqrt{n}}{2}\Big(\frac{\mathbf{a} - \frac{1}{2}\mathbf{1}}{\|\mathbf{a} - \frac{1}{2}\mathbf{1}\|_2}\Big) + \frac{1}{2}\mathbf{1}$, for any vector $\mathbf{a} \in \mathbb{R}^n$.

Update $\mathbf{z}_4$:

$$\mathbf{z}_4 \leftarrow \arg\min_{\mathbf{z}_4} \; y_6 \mathbf{z}_4^\top \mathbf{E}_1^\top \mathbf{E}_2 \mathbf{x} + \frac{\rho_6}{2} \|\mathbf{z}_4^\top \mathbf{E}_1^\top \mathbf{E}_2 \mathbf{x} - v\|_2^2 - \mathbf{y}_7^\top \mathbf{z}_4 + \frac{\rho_7}{2} \|\mathbf{x} - \mathbf{z}_4\|_2^2 \tag{12}$$

The problem is strongly convex quadratic in $\mathbf{z}_4$, so we obtain the unique global minimizer by equating the gradient to zero.

$$\Big[\rho_6 \mathbf{E}_1^\top \mathbf{E}_2 \mathbf{x}\mathbf{x}^\top \mathbf{E}_2^\top \mathbf{E}_1 + \rho_7 \mathbf{I}_{nk}\Big]\mathbf{z}_4 = \mathbf{y}_7 + \rho_7 \mathbf{x} - y_6 \mathbf{E}_1^\top \mathbf{E}_2 \mathbf{x} + \rho_6 v \mathbf{E}_1^\top \mathbf{E}_2 \mathbf{x}$$

Update $\mathbf{z}_5$:

$$\mathbf{z}_5 \leftarrow \arg\min_{\mathbf{z}_5} \; y_8 \mathbf{z}_5^\top \mathbf{E}_3^\top \mathbf{E}_4 \mathbf{x} + \frac{\rho_8}{2} \|\mathbf{z}_5^T \mathbf{E}_3^\top \mathbf{E}_4 \mathbf{x}\|_2^2 - \mathbf{y}_9^\top \mathbf{z}_5 + \frac{\rho_9}{2} \|\mathbf{x} - \mathbf{z}_5\|_2^2$$

The problem is also strongly convex quadratic in $\mathbf{z}_5$, so we obtain the unique global minimizer by equating the gradient to zero.

$$\Big[\rho_8 \mathbf{E}_3^\top \mathbf{E}_4 \mathbf{x}\mathbf{x}^\top \mathbf{E}_4^\top \mathbf{E}_3 + \rho_9 \mathbf{I}_{nk}\Big]\mathbf{z}_5 = \mathbf{y}_9 + \rho_9 \mathbf{x} - y_8 \mathbf{E}_3^\top \mathbf{E}_4 \mathbf{x}$$

Update $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5, y_6, \mathbf{y}_7, y_8, \mathbf{y}_9$: Lastly, we need to perform dual gradient ascent to update the dual variables as follows:

$$\begin{aligned}
&\mathbf{y}_1 \leftarrow \mathbf{y}_1 + \rho_1\Big(\Psi^\top \mathbf{x} - \mathbf{1}_n\Big), \quad \mathbf{y}_2 \leftarrow \mathbf{y}_2 + \rho_2\Big(\mathbf{x} - \mathbf{z}_2\Big) \\
&\mathbf{y}_3 \leftarrow \mathbf{y}_3 + \rho_3\Big(\mathbf{x} - \mathbf{z}_3\Big), \quad \mathbf{y}_4 \leftarrow \mathbf{y}_4 + \rho_4\Big(\mathbf{Q}\mathbf{P}^\top \mathbf{x} - \mathbf{u}\Big) \\
&\mathbf{y}_5 \leftarrow \mathbf{y}_5 + \rho_5\Big(\mathbf{x} - \mathbf{P}\mathbf{w} \odot \mathbf{C}\mathbf{x}\Big), \quad y_6 \leftarrow y_6 + \rho_6\Big(\mathbf{z}_3^\top \mathbf{E}_1^\top \mathbf{E}_2 \mathbf{x} - v\Big) \\
&\mathbf{y}_7 \leftarrow \mathbf{y}_7 + \rho_7\Big(\mathbf{x} - \mathbf{z}_3\Big), \quad y_8 \leftarrow y_8 + \rho_8\Big(\mathbf{z}_4^\top \mathbf{E}_3^\top \mathbf{E}_4 \mathbf{x}\Big) \\
&\mathbf{y}_9 \leftarrow \mathbf{y}_9 + \rho_9\Big(\mathbf{x} - \mathbf{z}_4\Big)
\end{aligned} \tag{13}$$

Auxiliary Results

Here, we present some additional experimental results that augment the discussion made in the manuscript. Primarily, we provide empirical evidence that our FCKm method and its constrained variants converge to binary solutions that satisfy different constraints (pairwise and cardinality).

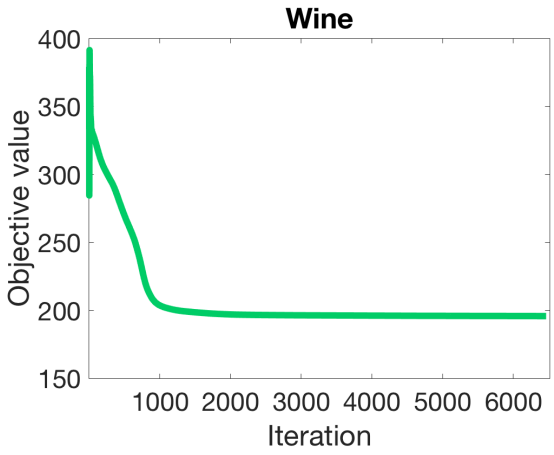

Figure 1: Convergence of the K-means objective value using FCKm with random initialization on the Wine dataset. Note the decreasing nature of the objective and its smooth convergence to the solution.

Convergence for Unconstrained Clustering. In Figure 1, we plot the K-means objective value at each ADMM iteration for an unconstrained clustering task with three clusters. Note that the initialization to this problem is a random three-way clustering. The objective decreases monotonically (after the first 2-3 iterations) and converges to a minimum value in approximately 6000 iterations. The optimization is stable with no perturbations at the onset of convergence.

In Figure 2, we plot the cluster vector for each of the three clusters being optimized at different iterations (1, 500, 2500, and 6530), i.e. we plot the three pieces of the label vector $\mathbf{x}$. In the first iteration, the initial clustering is done randomly, so it is binary but it does not lead to a good objective. As ADMM progresses, the continuous solution $\mathbf{x}$ becomes more and more binary, until it converges to a feasible binary solution where the three clusters are completely disjoint.

Convergence for Constrained Clustering. In Figure 3, we plot the K-means objective value at each ADMM iteration for a constrained clustering task with three clusters. In this task, we enforce cardinality, must-link, and cannot-link constraints onto the optimization. The initialization is taken to be a random assignment between the three disjoint clusters. In this case, the objective tends to be monotonically increasing after the first few iterations. This might seem counter-intuitive, since we are trying to minimize the objective. However, it must be noted that the continuous solution vector $\mathbf{x}$ in each ADMM iteration tends not to be feasible

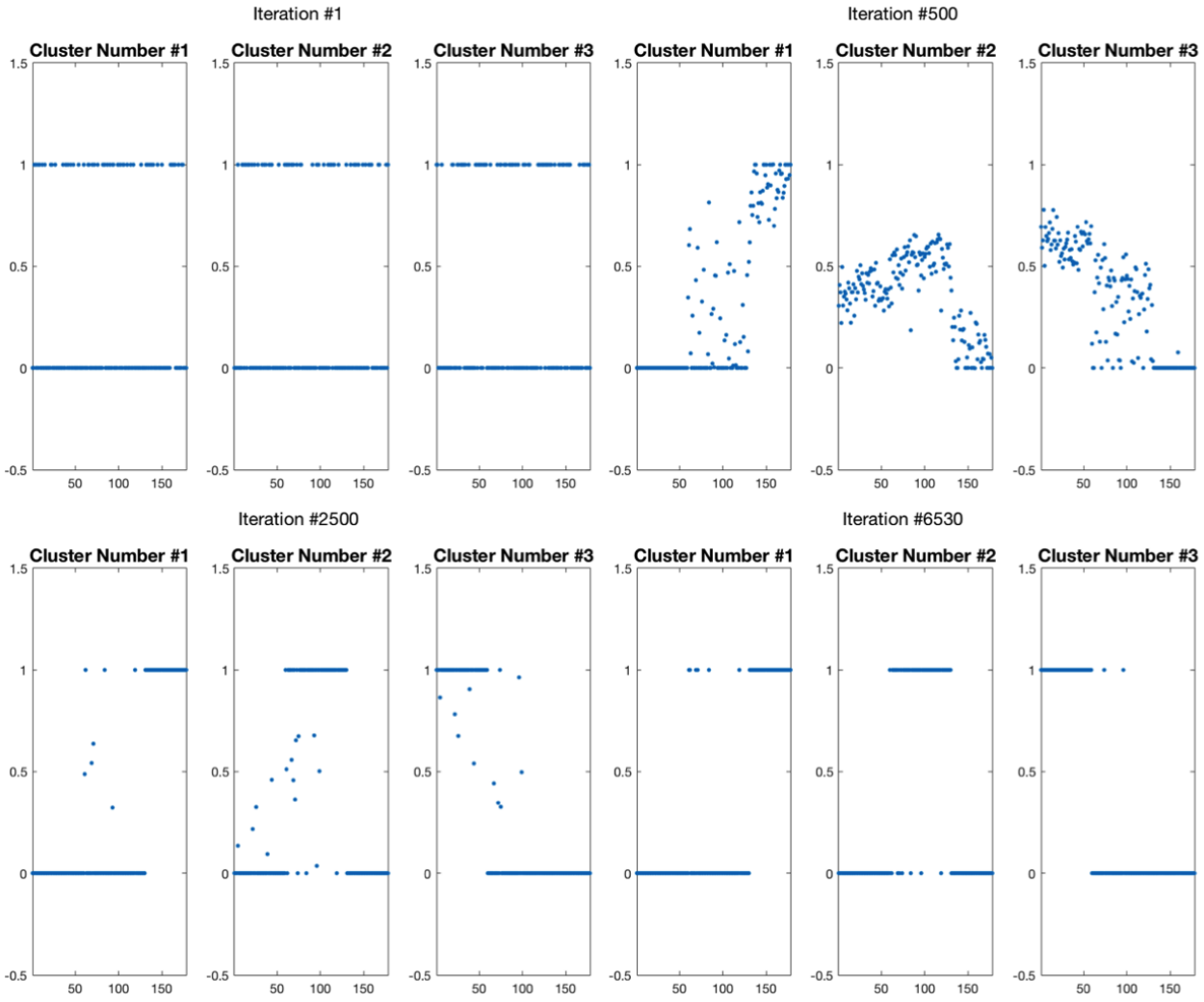

Figure 2: Convergence of the solution $\mathbf{x}$ using FCKm with random initialization on the Wine dataset.

w.r.t. the enforced constraints. In other words, these constraints are being enforces more and more as the ADMM process proceeds, which forces the tradeoff between objective and feasibility. But, similar to the unconstrained case, the variation in objective is smooth and no perturbations are exhibited when ADMM begins to converge.

Moreover, Figure 4 plots the solution vectors for each cluster at four different iterations (1, 15, 25, and 200). A similar behavior to the unconstrained case is encountered, where disjoint binary solution vectors are converged to. However, the notable difference is that we also report the number of cardinality (CardV), must-link (MLV), and cannot-link (CLV) violations at each of these iterations. We see that these violations gradually decrease until convergence occurs, when no violations persist.

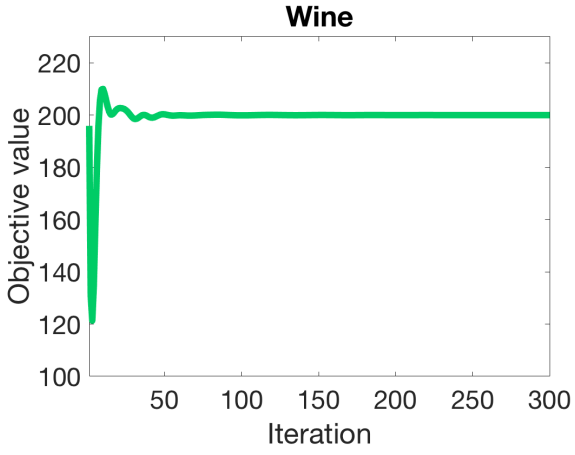

Figure 3: Convergence of the K-means objective value using FCKm-Mix with a random initialization on Wine dataset.

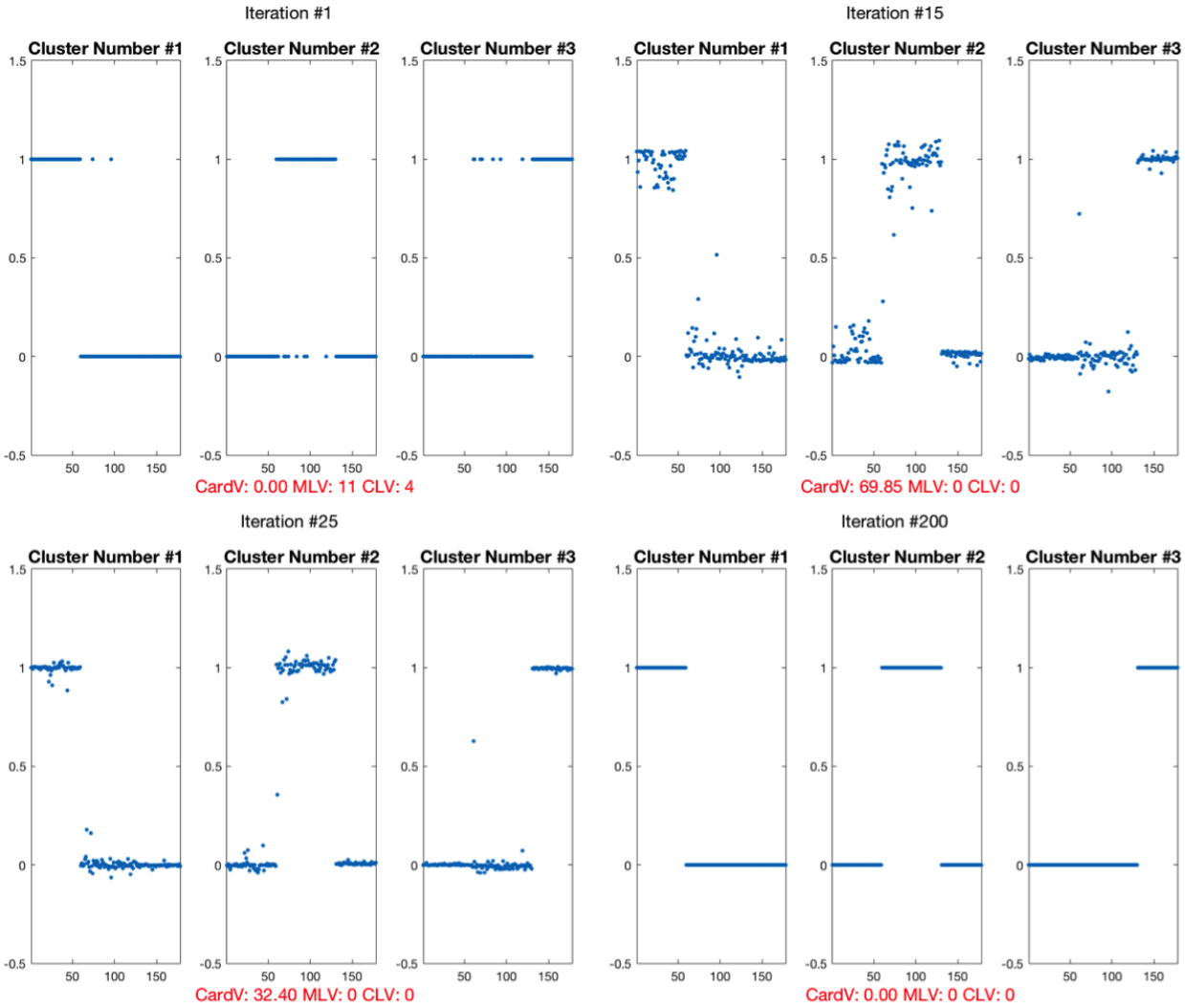

Figure 4: Convergence of the solution $\mathbf{x}$ using FCKm-Mix with random initialization on the Wine dataset

References

Wu, B., and Ghanem, B. 2016. ℓ_p-box admm: A versatile framework for integer programming. volume abs/1604.07666.