Chapter 9

Using Assignment Constraints to Avoid Empty Clusters in k-means Clustering

A. Demiriz
Dept. of Industrial Engineering, Sakarya University, Sakarya, 54187, Turkey
ademiriz@gmail.com

K. P. Bennett
Dept. of Mathematical Sciences, Rensselaer Polytechnic Inst., Troy, NY 12180 bennek@rpi.edu

P. S. Bradley
Apollo Data Technologies, 12729 Northup Way, Ste. 7, Bellevue, WA 98005
paul@apollodatatech.com

Abstract We consider practical methods for adding constraints to the k-means clustering algorithm in order to avoid local solutions with empty clusters or clusters having very few points. We often observe this phenomena when applying k-means to datasets where the number of dimensions is $d \geq 10$ and the number of desired clusters is $k \geq 20$. Moreover, recent studies have shown successful formulations of various other types of constraints. Particularly, must-link and cannot-link types constraints have been studied in several papers. An appropriate objective function needs to be constructed to find clusters that satisfy minimum capacity, must-link and cannot-link pairwise constraints at the same time. Obviously, it requires an analysis of the applicability and the level of complexity of the constraint types.

We propose explicitly adding k constraints to the underlying clustering optimization problem requiring that each cluster have at least a minimum number of points in it i.e. minimum capacity. We then investigate the resulting cluster assignment step. Numerical tests on real datasets indicate that the constrained approach is less prone to poor local solutions, producing a better summary of the underlying data. We also successfully formulate extended optimization models to cover other types of assignment constraints, specifically pairwise assignment constraints as well.

0-8493-0052-5/00/$0.00+$.50

9.1 Introduction

The k-means clustering algorithm [16] has become a workhorse for the data analyst in many diverse fields. One drawback to the algorithm occurs when it is applied to datasets with n data points in $d \geq 10$ dimensional real space $\mathbb{R}^d$ and the number of desired clusters is $k \geq 20$. In this situation, the k-means algorithm often converges with one or more clusters which are either empty or summarize very few data points (i.e. one data point). Preliminary tests on clustering sparse 300-dimensional web-browsing data indicate that k-means frequently converges with truly empty clusters. For $k = 50$ and $k = 100$, on average 4.1 and 12.1 clusters are empty.

Incorporating prior knowledge, whether in the form of firmly defining the number of non-empty clusters or pairwise relationships, is very essential in partially supervised clustering. Like the general clustering problem, the partially supervised clustering problem can also be posed as an optimization problem. With partial supervision, the underlying clustering model can be used to prevent poor local solutions.

We propose explicitly adding k constraints to the underlying clustering optimization problem requiring that cluster h contain at least τ_h points. We focus on the resulting changes to the k-means algorithm and compare the results from standard k-means and the proposed constrained k-means algorithms. Empirically, for modest values of τ_h, solutions are obtained that better summarize the underlying data.

Since clusters with very few or no data points may be artifacts of poor local minima, typical approaches to handling them within the standard k-means framework include re-running the algorithm with new initial cluster centers or checking the cluster model at algorithm termination, resetting empty clusters, and re-running the algorithm. Our approach avoids the additional computation of these heuristics which may still produce clusters with too few points. In addition to providing a well-posed mathematical way to avoid small clusters, this work can be generalized to other constraints ensuring desirable clustering solutions (e.g. outlier removal or specified groupings) and to Expectation-Maximization probabilistic clustering.

Alternatively, empty clusters can be regarded as desirable "natural" regularizer of the cluster model. This heuristic argument states that if the data does not "support" k clusters, then allowing clusters to go empty, and hence reducing the value of k, is a desirable side effect. But there are applications in which, given a value of k, one desires to have a cluster model with k non-empty clusters. These include the situation when the value of k is known *a priori* and applications in which the cluster model is utilized as a compressed version of a specific dataset [5, 19].

A significant part of this chapter is based on our earlier work in [8]. However we extend our formulations in this chapter to cover pairwise assignment

constraints and a new constraint on minimum capacity on labeled points assigned to each cluster. The remaining portion of the chapter is organized as follows. Section 9.2 formalizes the constrained clustering optimization problem and outlines the algorithm computing a locally optimal solution. The sub-problem of computing cluster assignments so that cluster h contains at least τ_h points is discussed in Section 9.3. Section 9.4 presents numerical evaluation of the algorithm in comparison with the standard k-means implementation on real datasets. We report results on both small and large datasets in Section 9.4. In addition to constrained k-means results, we report also constrained k-median results and compare them. In Section 9.5, we provide a wide variety of extensions to our base model to incorporate new types of assignment constraints and Section 9.6 concludes the chapter.

9.2 Constrained Clustering Problem and Algorithm

Given a dataset $\mathcal{X} = \{x_i\}_{i=1}^n$ of n points in $\mathbb{R}^d$ and a number k of desired clusters, the k-means clustering problem is as follows. Find cluster centers $\mu_1, \mu_2, \ldots, \mu_k$ in $\mathbb{R}^d$ such that the sum of the 2-norm distance squared between each point x_i and its *nearest* cluster center μ_h is minimized. Specifically:

$$\min_{\mu_1,\ldots,\mu_k} \sum_{i=1}^{n} \min_{h=1,\ldots,k} \left(\frac{1}{2} \|x_i - \mu_h\|^2 \right). \tag{9.1}$$

By [10, Lemma 2.1], Problem (9.1) is equivalent to the following problem where the min operation in the summation is removed by introducing "selection" variables $T_{i,h}$.

$$\begin{aligned} \underset{\mu,T}{\text{minimize}} \quad & \sum_{i=1}^{n} \sum_{h=1}^{k} T_{i,h} \cdot \left(\frac{1}{2} \|x_i - \mu_h\|^2 \right) \\ \text{s.t.} \quad & \sum_{h=1}^{k} T_{i,h} = 1,\ i = 1, \ldots, n, \\ & T_{i,h} \geq 0,\ i = 1, \ldots, n,\ h = 1, \ldots, k. \end{aligned} \tag{9.2}$$

Note that $T_{i,h} = 1$ if data point x_i is closest to center μ_h and zero otherwise.

Problem (9.2) or, equivalently (9.1), is solved by the k-means algorithm iteratively. In each iteration, Problem (9.2) is solved first for $T_{i,h}$ with the cluster centers μ_h fixed. Then, (9.2) is solved for μ_h with the assignment variables $T_{i,h}$ fixed. The stationary point computed satisfies the Karush-Kuhn-Tucker (KKT) conditions [17] for Problem (9.2), which are necessary for optimality.

k-means Clustering Algorithm Given a database $\mathcal{X}$ of n points in $\mathbb{R}^d$ and cluster centers $\mu_{1,t}, \mu_{2,t}, \ldots, \mu_{k,t}$ at iteration t, compute $\mu_{1,t+1}, \mu_{2,t+1}, \ldots, \mu_{k,t+1}$ at iteration $t+1$ in the following 2 steps:

1. **Cluster Assignment.** For each data record $x_i \in \mathcal{X}$, assign x_i to cluster $h(i)$ such that center $\mu_{h(i),t}$ is nearest to x_i in the 2-norm.
2. **Cluster Update.** Compute $\mu_{h,t+1}$ as the mean of all points assigned to cluster h.

Stop when $\mu_{h,t+1} = \mu_{h,t}$, $h = 1, \ldots, k$, else increment t by 1 and go to step 1.

Suppose cluster h is empty when Algorithm 9.2 terminates, i.e., $\sum_{i=1}^{n} T_{i,h} = 0$. The solution computed by Algorithm 9.2 in this case satisfies the KKT conditions for Problem (9.2). Hence, it is plausible that the standard k-means algorithm may converge with empty clusters. In practice, we observe this phenomenon when clustering high-dimensional datasets with a large number of clusters.

The KKT conditions [17] for Problem (9.2) are:

$$\sum_{h=1}^{k} T_{i,h} = 1 \,\forall i, \;\; T_{i,h} \geq 0 \,\forall i, h,$$

$$\|x_i - \mu_h\|^2 = \min_{\tilde{h}=1,\ldots,k} \|x_i - \mu_{\tilde{h}}\|^2 \Leftrightarrow T_{i,h} \geq 0,$$

$$\sum_{i=1}^{n} T_{i,h} > 0 \Rightarrow \mu_h = \frac{\sum_{i=1}^{n} T_{i,h} x_i}{\sum_{i=1}^{n} T_{i,h}}$$

$$\sum_{i=1}^{n} T_{i,h} = 0 \Rightarrow \mu_h \text{ arbitrary.}$$

To avoid solutions with empty clusters, we propose explicitly adding constraints to Problem (9.2) requiring that cluster h contain at least τ_h data points, where $\sum_{h=1}^{k} \tau_h \leq n$. This yields the following constrained k-means

problem:

$$\begin{aligned} \underset{\mu,T}{\text{minimize}} \quad & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right) \\ & \sum_{i=1}^{n} T_{i,h} \geq \tau_h,\ h=1,\ldots,k \\ \text{s.t.} \quad & \sum_{h=1}^{k} T_{i,h} = 1,\ i=1,\ldots,n, \\ & T_{i,h} \geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k. \end{aligned} \tag{9.3}$$

Like the classic k-means algorithm, we propose an iterative algorithm to solve (9.3).

Constrained k-means Clustering Algorithm Given a database $\mathcal{X}$ of n points in $\mathbb{R}^d$, minimum cluster membership values $\tau_h \geq 0$, $h = 1,\ldots,k$ and cluster centers $\mu_{1,t},\mu_{2,t},\ldots,\mu_{k,t}$ at iteration t, compute $\mu_{1,t+1},\mu_{2,t+1},\ldots,\mu_{k,t+1}$ at iteration $t+1$ in the following 2 steps:

1. **Cluster Assignment.** Let $T^t_{i,h}$ be a solution to the following linear program with $\mu_{h,t}$ fixed:

$$\begin{aligned} \underset{T}{\text{minimize}} \quad & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_{h,t}\|^2\right) \\ & \sum_{i=1}^{n} T_{i,h} \geq \tau_h,\ h=1,\ldots,k \\ \text{s.t.} \quad & \sum_{h=1}^{k} T_{i,h} = 1,\ i=1,\ldots,n, \\ & T_{i,h} \geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k. \end{aligned} \tag{9.4}$$

2. **Cluster Update.** Update $\mu_{h,t+1}$ as follows:

$$\mu_{h,t+1} = \begin{cases} \dfrac{\sum_{i=1}^{n} T^t_{i,h} x_i}{\sum_{i=1}^{n} T^t_{i,h}} & \text{if } \sum_{i=1}^{n} T^t_{i,h} > 0, \\ \mu_{h,t} & \text{otherwise.} \end{cases}$$

Stop when $\mu_{h,t+1} = \mu_{h,t}$, $h = 1,\ldots,k$, else increment t by 1 and go to step 1.

Like the traditional k-means approach, the constrained k-means algorithm

iterates between solving (9.3) in $T_{i,h}$ for fixed μ_h, then solving (9.3) in μ_h for fixed $T_{i,h}$. We end this section by with a finite termination result similar to [9, Theorem 7].

PROPOSITION 9.1
The Constrained k-means Algorithm 9.2 terminates in a finite number of iterations at a cluster assignment that is locally optimal. Specifically, the objective function of (9.3) cannot be decreased by either reassignment of a point to a different cluster, while maintaining $\sum_{i=1}^{n} T_{i,h} \geq \tau_h$, $h = 1, \ldots, k$, *or by defining a new cluster center for any of the clusters.*

PROOF At each iteration, the cluster assignment step cannot increase the objective function of (9.3). The cluster update step will either strictly decrease the value of the objective function of (9.3) or the algorithm will terminate since

$$\mu_{h,t+1} = \arg\min_{\mu} \sum_{i=1}^{n} \sum_{h=1}^{k} T_{i,h}^{t} \cdot \left(\frac{1}{2} \|x_i - \mu_h\|^2 \right)$$

is a strictly convex optimization problem with a unique global solution. Since there are a finite number of ways to assign n points to k clusters so that cluster h has at least τ_h points, since Algorithm 9.2 does not permit repeated assignments, and since the objective of (9.3) is strictly non-increasing and bounded below by zero, the algorithm must terminate at some cluster assignment that is locally optimal. □

Although our problem formulation is given for the constrained k-means algorithm, by utilizing a 1-norm cost function and using a 1-norm distance metric for the cluster assignment and update steps we can readily extend our formulation to run constrained k-median algorithm. In the next section we discuss solving the linear program sub-problem in the cluster assignment step of Algorithm 9.2 as a minimum cost network flow problem.

9.3 Cluster Assignment Sub-problem

The form of the constraints in the cluster assignment sub-problem (9.4) make it equivalent to a Minimum Cost Flow (MCF) linear network optimization problem [6]. This is used to show that the optimal cluster assignment will place each point in exactly one cluster and can be found using fast network simplex algorithms. In general, a MCF problem has an underlying graph

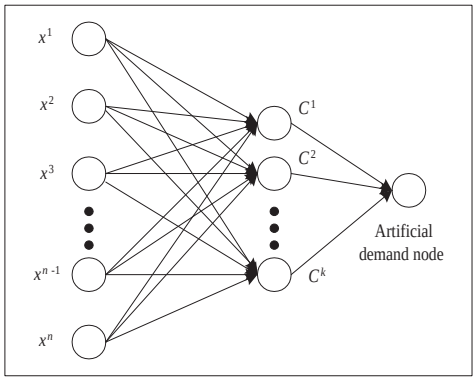


FIGURE 9.1: Equivalent Minimum Cost Flow formulation of (9.4).

structure. Let $\mathcal{N}$ be the set of nodes. Each node $i \in \mathcal{N}$ has associated with it a value b_i indicating whether it is a supply node ($b_i > 0$), a demand node ($b_i < 0$), or a transshipment node ($b_i = 0$). If $\sum_{i \in \mathcal{N}} b_i = 0$, the problem is feasible (i.e. the sum of the supplies equals the sum of the demands). Let $\mathcal{A}$ be the set of directed arcs. For each arc $(i, j) \in \mathcal{A}$, the variable $y_{i,j}$ indicates amount of flow on the arc. Additionally, for each arc (i, j), the constant $c_{i,j}$ indicates the cost of shipping one unit flow on the arc. The MCF problem is to minimize $\sum_{(i,j) \in \mathcal{A}} c_{i,j} \cdot y_{i,j}$ subject to the sum of the flow leaving node i minus the sum of flow incoming is equal to b_i. Specifically, the general MCF is:

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & \sum_{(i,j) \in \mathcal{A}} c_{i,h} \cdot y_{i,j} \\ \text{s.t.} \quad & \sum_j y_{i,j} - \sum_j y_{j,i} = b_i, \forall i \in \mathcal{N} \\ & 0 \leq y_{i,j} \leq u_{i,j}, \forall (i,j) \in \mathcal{A}. \end{aligned}$$

Let each data point x_i correspond to a supply node with supply = 1 ($b_{x_i} = 1$). Let each cluster μ_h correspond to a demand node with demand $b_{\mu_h} = -\tau_h$. Let there be an arc in $\mathcal{A}$ for each (x_i, μ_h) pair. The cost on arc (x_i, μ_h) is $\|x_i - \mu_h\|^2$. To satisfy the constraint that the sum of the supplies equals the sum of the demands, we need to add an artificial demand node a with demand $b_a = -n + \sum_{h=1}^{k} \tau_h$. There are arcs from each cluster node μ_h to a with zero cost. There are no arcs to or from the data point nodes x_i to the artificial node a. See Figure 9.1. Specifically, let $\mathcal{N} = \{x_i,\, i = 1, \ldots, n\} \cup \{\mu_h,\, h = 1, \ldots, k\} \cup \{a\}$. Let $\mathcal{A} = \{(x_i, \mu_h),\, x_i, \mu_h \in \mathcal{N}\} \cup \{(\mu_h, a),\, \mu_h \in \mathcal{N}\}$. With these identifications and the costs, supplies and demands above, (9.4) has an

equivalent MCF formulation. This equivalence allows us to state the following proposition that integer values of $T_{i,h}$ are optimal for (9.4).

PROPOSITION 9.2
If each τ_h, $h = 1, \ldots, k$ is an integer, then there exists an optimal solution of (9.4) such that $T_{i,h} \in \{0, 1\}$.

PROOF Consider the equivalent MCF formulation of (9.4). Since $b_{x_i} = 1, \forall x_i \in \mathcal{N}$, $b_{\mu_h} = -\tau_h$, and $b_a = -n + \sum_{h=1}^{k} \tau_h$ are all integers, it follows from [6, Proposition 2.3] that an optimal flow vector y is integer-valued. The optimal cluster assignment values $T_{i,h}$ correspond y_{x_i,μ_h} and, since each node x_i has 1 unit of supply, the maximum value of $T_{i,h}$ at a solution is 1. □

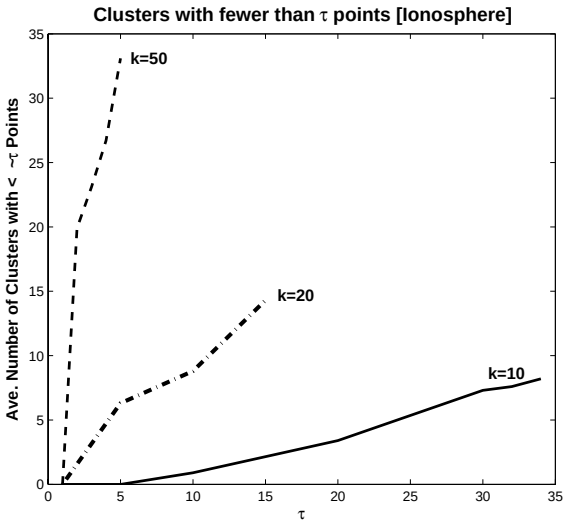
Hence, we are able to obtain optimal $\{0, 1\}$ assignments without having to solve a much more difficult integer programming problem. In addition to deriving the integrality result of Proposition 9.2, the MCF formulation allows one to solve (9.4) via codes specifically tailored to network optimization [6]. These codes usually run 1 or 2 orders of magnitude faster than general linear programming (LP) codes.

9.4 Numerical Evaluation

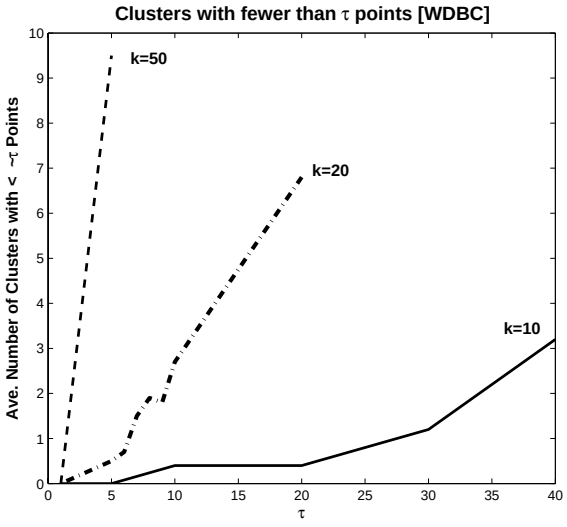
We conducted two different sets of experiments on machine learning benchmark datasets provided in [1]. In the first set of experiments, we report results using two real datasets: the Johns Hopkins Ionosphere dataset and the the Wisconsin Diagnostic Breast Cancer dataset (WDBC) [1]. The results from the first set of experiments are also reported in [8].

The Ionosphere dataset contains 351 data points in $\mathbb{R}^{33}$ and values along each dimension were normalized to have mean 0 and standard deviation 1. The WDBC dataset subset used consists of 683 normalized data points in $\mathbb{R}^9$. The values of τ_h (denoted by τ) were set equally across all clusters. The ILOG CPLEX 6.5 LP solver was used for cluster assignment. For initial cluster centers sampled uniformly on the range of the data, k-means produced at least 1 empty cluster in 10 random trials on WDBC for $k \geq 30$ and on Ion for $k \geq 20$. Figures 9.2 and 9.3 give results for initial clusters chosen randomly from the dataset. This simple technique can eliminate many empty clusters. Figure 9.2 shows the frequency with which the standard k-means algorithm 9.2 converges to clusters having fewer than τ points.

The effect on the quality of the clustering by the constraints imposed by



(a) Ionosphere, $K = 10, 20, 50$



(b) WDBC, $K = 10, 20, 50$

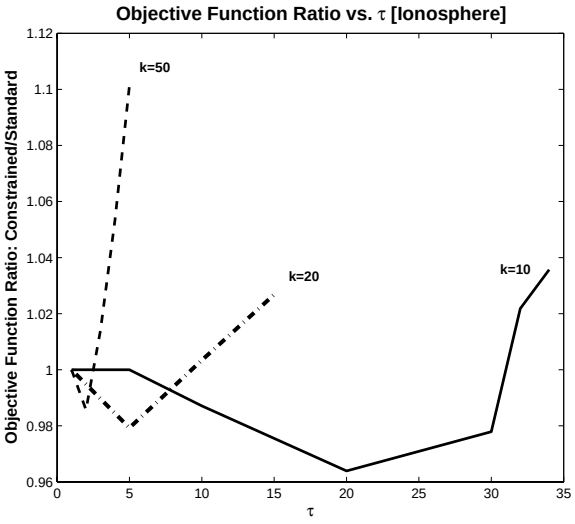
FIGURE 9.2: Average number of clusters with fewer than τ data points computed by the standard k-means algorithm 9.2

the constrained k-means Algorithm 9.2 is quantified by the ratio of the average objective function of (9.1) computed at the constrained k-means solution over that of the standard k-means solution. Adding constraints to any minimization problem can never decrease the **globally** optimal objective value. Thus we would expect this ratio to be greater than 1. Surprisingly the constrained k-means algorithm frequently found better local minima (ratios less than 1) than did the standard k-means approach. This might be due to the a local solution with a large cluster, some other clusters with few points and/or even empty clusters. Note that the same starting points were used for both algorithms. Results are summarized in Figure 9.3. Notice that for a fixed k, solutions computed by constrained k-means are equivalent to standard k-means for small τ-values. For large τ-values, the constrained k-means solution is often inferior to those of standard k-means. In this case, to satisfy the τ-constraints, the algorithm must group together points which are far apart resulting in a higher objective value. For a given dataset, superior clustering solutions are computed by the constrained k-means algorithm when τ is chosen in conjunction with k. For small values of k (e.g. $k = 5$) we observe ratios < 1 up to $\tau = 50$ (maximum tested) on Ionosphere. For $k = 20$, we begin to see ratios > 1 for $\tau = 10$. Similar results are observed on WDBC.

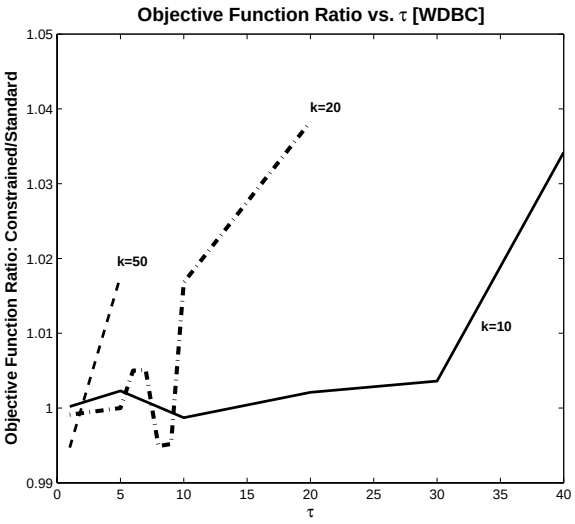
For a given values of k and τ_h, $h = 1, \ldots, k$, effort is made so that the τ_h constraints are satisfies by the initial cluster centers and the final cluster centers computed by k-means. Initial cluster centers where chosen by randomly selecting k data points. If the number of points in cluster h is $< \tau_h$, then a new set of initial cluster centers are chosen. This is repeated until the thresholds τ_h, $h = 1, \ldots, k$ are satisfied or until 50 sets of initial centers have been tried. The k-means Algorithm 9.2 is applied. If, at convergence, the τ_h thresholds are not satisfied, the entire initialization procedure is repeated (at most 10 times). The initial centers used for k-means are then also used to initialize constrained k-means. With this initialization strategy, for all values of k and $\tau_h > 1$ tested, k-means often converges with clusters violating the τ_h constraints.

The second set of experiments was run over a higher-dimensional dataset derived from web-browsing behavior to a large internet portal. The browsing history for a group of 10144 randomly selected users to 300 of the most popular news category stories was generated. This dataset can be viewed as 10144 data points in $\mathbb{R}^{300}$. We refer to this dataset as the "Web Dataset". In order to handle this larger dataset, we modify our original MATLAB code and utilize MOSEK 4.0 as the linear programming solver [18], which can be seamlessly integrated with MATLAB.

In addition to running k-means and constrained k-means algorithms, we also report results from k-median [10] and constrained k-median algorithms by using the 1-norm distance metric as mentioned in Section 9.2. The k-median clustering algorithm uses the median value in updating the cluster centers instead of using the average in the case of the k-means algorithm. Since we used a larger dataset, we modified the definition of the empty cluster to be



(a) Ionosphere, $K = 10, 20, 50$



(b) WDBC, $K = 10, 20, 50$

FIGURE 9.3: Average ratio of objective function (9.1) computed at the constrained k-means solution over that of the standard k-means solution versus τ.

TABLE 9.1: k-means and Constrained k-means Results on Web Dataset for k=20

	k-means			Constrained k-means	
τ	Objective ($\pm$ σ)	Time (Sec.)	No. of Empty Clusters ($\pm$ σ)	Objective ($\pm$ σ)	Time (Sec.)
10	396037 $\pm$ 59297	104.9	8.7 $\pm$ 1.89	574555 $\pm$ 13209	154.2
20	424178 $\pm$ 31575	102.1	8.5 $\pm$ 1.72	661215 $\pm$ 6367	140.5
30	377261 $\pm$ 59321	90.1	9.3 $\pm$ 2.31	710156 $\pm$ 8086	154.9

TABLE 9.2: k-median and Constrained k-median Results on Web Dataset for k=20

	k-median			Constrained k-median	
τ	Objective ($\pm$ σ)	Time (Sec.)	No. of Empty Clusters ($\pm$ σ)	Objective ($\pm$ σ)	Time (Sec.)
10	37783 $\pm$ 539	77.2	2.8 $\pm$ 1.48	38091 $\pm$ 1166	137.62
20	37989 $\pm$ 709	69.2	1.9 $\pm$ 1.79	38389 $\pm$ 1258	135.90
30	38140 $\pm$ 748	70.6	2.0 $\pm$ 1.49	38811 $\pm$ 878	133.46

one with 5 or fewer points. We ran the experiments on a Pentium M 1.60 GHz notebook with 768 MB of memory running under the Windows XP operating system. For brevity, we only set k equal to 20. We set τ to be 10, 20 and 30. Initial cluster centers were randomly picked from the dataset. Thus, the initial starting point consists of clusters that contain at least 1 point. At algorithm termination, clusters containing 5 or fewer points are considered "empty" per the modified definition mentioned above. For each τ value, we ran 10 random realizations of the dataset. We report average values over these 10 runs in the following tables.

Average objective values and times in seconds for both regular and constrained clustering methods and also number of empty clusters are reported for k-means and k-median clustering in Tables 9.1 and 9.2 respectively. Corresponding standard deviations are reported after the $\pm$ operator. Notice that the k-means clustering algorithm ends up with approximately 9 empty clusters on average out of 20 initial clusters. On the other hand, k-median clustering algorithm results in around 2 empty clusters on average. Changing τ does not seriously affect the running time for both constrained clustering methods. Although the objective values of both constrained and regular k-median methods do not differ, we see a significant change in constrained k-means probably due to the empty or near empty clusters found in regular k-means methods. From the comparisons of standard deviations of the objective values from both regular and constrained k-means algorithms, we can conclude that although standard k-means algorithm has lower average objective values, it has higher variations. This result directly indicates the volatility of the local solutions of the regular k-means algorithm.

These results may be a result of the following observations: i) a data point

might be closer to any other data point as the dimensionality of the space becomes very large; ii) the k-means algorithm is more prone to be affected by "outliers" in the dataset than the k-median algorithm since k-means minimizes the 2-norm squared distance, whereas k-median minimizes the 1-norm distance [7].

9.5 Extensions

Around the same time that our earlier work [8] was published, Wagstaff and Cardie proposed using pairwise constraints in clustering problems [20]. More specifically they proposed the usage of must-link and cannot-link types of constraints in a clustering framework. From an optimization point of view, it might be more challenging to add pairwise constraints into clustering problems in general since it might jeopardize convexity and the smoothness of the solution. The work of Wagstaff and Cardie was later applied to GPS lane finding problem [21]. Another constraint type was first studied in [2]. The aim in [2] was to utilize a sampling based scalable clustering algorithm with balancing constraints to produce balanced clusters which is important in some commercial applications. Chapter 8 of this book is also on balancing constraints. In this section, we basically review some prior work and develop certain optimization models to tackle with new types of constraints.

Kleinberg and Tardos proposed some linear programming relaxations of the metric labeling problem in [14, 15]. Specifically they used pairwise relationships in assigning k `labels (classes)` to each of n `objects`. In their approach to metric labeling problem, they utilized a Markov Random Fields framework [14, 15].

We can easily extend their uniform metric labeling formulation to a 2-norm cost function as follows in the following optimization model. Approximations to Kleinberg and Tardos' model for the general metrics are studied in [11].

$$\begin{aligned} \underset{T}{\text{minimize}} \; & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right) + \sum_{(u,v)\in\mathcal{X}} w(u,v)\cdot\frac{1}{2}\sum_{h=1}^{k}|T_{u,h}-T_{v,h}| \\ \text{s.t.} \; & \sum_{h=1}^{k} T_{i,h} = 1,\; i=1,\ldots,n, \\ & T_{i,h}\geq 0,\; i=1,\ldots,n,\; h=1,\ldots,k. \end{aligned} \tag{9.5}$$

The major difference in Problem 9.5 with the original clustering problem defined in Problem 9.2 is the fact that there is a cost w associated with pairing two objects u and v. Technically, we can easily incorporate both must-link and cannot-link pairwise constraints with an appropriate cost structure with

this formulation. Intuitively, appropriate positive terms should be assigned to $w(u,v)$'s. Assigning a negative value would make the objective non-convex and more difficult to solve with ordinary linear programming approach. Since each point is assigned exactly to one cluster, the term $w(u,v)\cdot\frac{1}{2}\sum_{h=1}^{k}|T_{u,h}-T_{v,h}|$ will be equal to 0 when both points are assigned to the same cluster and non-zero otherwise.

Although a minimum might exist, an algorithm like Algorithm 9.2 may not be sufficient to find a solution and the convergence of such an algorithm may not be guaranteed. Therefore a near zero cost value should be assigned to w for a cannot-link pairwise relationship (constraint). We can assign prohibitively large cost values for the must-link constraints. In this case, we can argue that there exists an extreme point solution, yet we need to show that Algorithm 9.2 converges. However, a more elegant way of introducing constraints is needed. In the following model, we first introduce our constraints on the number of points assigned to each cluster to Kleinberg and Tardos' model proposed in [14, 15].

$$\begin{aligned}
\underset{T}{\text{minimize}} \quad & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right) + \sum_{(u,v)\in\mathcal{X}} w(u,v)\cdot\frac{1}{2}\sum_{h=1}^{k}|T_{u,h}-T_{v,h}| \\
\text{s.t.} \quad & \sum_{h=1}^{k} T_{i,h} = 1,\ i=1,\ldots,n, \\
& \sum_{i=1}^{n} T_{i,h} \geq \tau_h,\ h=1,\ldots,k, \\
& T_{i,h} \geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k.
\end{aligned}$$

Certainly, pairwise relationships can be introduced to Markov random fields models in various ways. Basu *et al.* used hidden Markov models in [3, 4] in a probabilistic way to introduce such constraints. In the following model, we introduce such pairwise assignment constraints in our mathematical programming model. Notice that cannot-link constraints can be added without violating the convexity. However care is needed for the must-link type constraints since they are in the form of absolute value that is non-convex.

$$
\begin{aligned}
\underset{T,\varepsilon}{\text{minimize}} \; & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right) + \sum_{(u,v)\in\mathcal{X}} w(u,v)\cdot\frac{1}{2}\sum_{h=1}^{k}|T_{u,h}-T_{v,h}| \\
& \sum_{h=1}^{k} T_{i,h} = 1,\ i=1,\ldots,n, \\
& \sum_{i=1}^{n} T_{i,h} \geq \tau_h,\ h=1,\ldots,k, \\
\text{s.t.} \quad & T_{i,h}+T_{j,h}\leq 1,\ \forall\ i,j\in C_{\neq}, h=1,\ldots,k, \\
& -\varepsilon_{i,j,h}\leq T_{i,h}-T_{j,h}\leq\varepsilon_{i,j,h},\ \forall\ i,j\in C_{=}, \\
& \sum_{h=1}^{k}\varepsilon_{i,j,h}=0,\ \forall\ i,j\in C_{=}, \\
& T_{i,h}\geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k.
\end{aligned}
\tag{9.6}
$$

In the previous model, Problem 9.6, we basically introduce a new variable, ε, for the each must-link constraint. From a practical point of view, Problem 9.6 needs to be solved by introducing a regularizer as given below model. Our aim in introducing the regularizer, ρ, is just to simplify the objective function and speed-up the solution. By doing this, we basically soften the must-link constraints. They are no longer hard constraints meaning that some violations of this type of constraints are permitted given that they are below certain associated costs.

$$
\begin{aligned}
\underset{T,\varepsilon}{\text{minimize}} \; & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right) + \rho\sum_{(i,j)\in C_{=}}\sum_{h=1}^{k}\varepsilon_{i,j,h} \\
& \sum_{h=1}^{k} T_{i,h} = 1,\ i=1,\ldots,n, \\
& \sum_{i=1}^{n} T_{i,h} \geq \tau_h,\ h=1,\ldots,k, \\
\text{s.t.} \quad & T_{i,h}+T_{j,h}\leq 1,\ \forall\ i,j\in C_{\neq}, h=1,\ldots,k, \\
& -\varepsilon_{i,j,h}\leq T_{i,h}-T_{j,h}\leq\varepsilon_{i,j,h},\ \forall\ i,j\in C_{=}, \\
& T_{i,h}\geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k.
\end{aligned}
\tag{9.7}
$$

After removing the cost function associated with $w(u,v)$ from Problem 9.6 and introducing a regularizer in Problem 9.7, the resulting mathematical programming model has become numerically more stable and an algorithm, such as Algorithm 9.2, can be devised to solve this problem. Considering the cannot-link constraints, such an algorithm will converge. By adding transshipment nodes, we can show that the problem is equivalent to MCF. Thus we will have an integer solution i.e. the integrality constraints are satisfied

too. On the other hand, considering the must-link constraints, we can show that the algorithm will converge but we no more have the integrality.

Our proposed framework in this section enables us to introduce new constraints to the clustering problem in general. Assume that we face a situation that point x_i must be in the same cluster with point x_j or in the same cluster with point x_g but not in the same cluster with both points x_j and x_g. This situation might arise in analyzing social networks data. Imagine one chooses to be friend of another person from among two persons but cannot be friend of both persons at the same time. We call this type of constraints as `OR` type constraints and denote it by C_{OR}. We show in the following model how to represent such constraints.

$$
\begin{aligned}
\underset{T,\varepsilon}{\text{minimize}} \quad & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right)+\rho\sum_{(i,j)\in C_=}\sum_{h=1}^{k}\varepsilon_{i,j,h} \\
& \sum_{h=1}^{k} T_{i,h}=1,\ i=1,\dots,n, \\
& \sum_{i=1}^{n} T_{i,h}\geq \tau_h,\ h=1,\dots,k, \\
\text{s.t.} \quad & T_{i,h}+T_{j,h}\leq 1,\ \forall\ i,j\in C_{\neq}, h=1,\dots,k, \\
& -\varepsilon_{i,j,h}\leq T_{i,h}-T_{j,h}\leq \varepsilon_{i,j,h},\ \forall\ i,j\in C_=, \\
& \sum_{h=1}^{k}|T_{i,h}-T_{j,h}|+|T_{i,h}-T_{g,h}|\leq 1,\ \forall\ i,j,g\in C_{OR}, \\
& T_{i,h}\geq 0,\ i=1,\dots,n,\ h=1,\dots,k.
\end{aligned}
\tag{9.8}
$$

In Problem 9.8, C_{OR} constraints are convex. However, we can still propose a relaxed form. Since C_{OR} constraints are also convex, the algorithm to find a solution for this problem will converge but we will not have the integrality.

Adding must-link and cannot-link types of constraints into the clustering model may decrease the quality of solution. Unexpected or even unwanted results may occur. In [12], two measures, namely informativeness and coherence are proposed to understand the underlying effects of adding constraints to the clustering problem. Such measures surely help to evaluate the importance of the semi-supervised approach through constrained clustering. Certain types of clustering approaches can be deployed for the transduction problem as well such as graph cut methods. However, it is reported that after deploying such methods for the two-class transduction problem, the algorithm might very well result in one very small cluster [13]. Such results may require a new type of constraint, precisely the minimum number of labeled points falling into each cluster. We can readily add such constraints to above Problem 9.8 as in the following formulation.

$$\begin{aligned}
\underset{T,\varepsilon}{\text{minimize}} \quad & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right)+\rho\sum_{(i,j)\in C_=}\sum_{h=1}^{k}\varepsilon_{i,j,h} \\
& \sum_{h=1}^{k} T_{i,h}=1,\ i=1,\ldots,n, \\
& \sum_{i=1}^{n} T_{i,h}\geq\tau_h,\ h=1,\ldots,k, \\
\text{s.t.} \quad & \sum_{i\in l} T_{i,h}\geq\pi_h,\ h=1,\ldots,k, \\
& T_{i,h}+T_{j,h}\leq 1,\ \forall\ i,j\in C_{\neq}, h=1,\ldots,k, \\
& -\varepsilon_{i,j,h}\leq T_{i,h}-T_{j,h}\leq\varepsilon_{i,j,h},\ \forall\ i,j\in C_=, \\
& T_{i,h}\geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k.
\end{aligned}$$

To simplify the model, we can just omit the other types of constraint and just focus on the minimum number of points (minimum capacity) for the each cluster whether labeled or unlabeled. Following formulation is provided for that reason.

$$\begin{aligned}
\underset{T}{\text{minimize}} \quad & \sum_{i=1}^{n}\sum_{h=1}^{k} T_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mu_h\|^2\right) \\
& \sum_{h=1}^{k} T_{i,h}=1,\ i=1,\ldots,n, \\
& \sum_{i=1}^{n} T_{i,h}\geq\tau_h,\ h=1,\ldots,k, \\
\text{s.t.} \quad & \sum_{i\in l} T_{i,h}\geq\pi_h,\ h=1,\ldots,k, \\
& T_{i,h}\geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k.
\end{aligned} \tag{9.9}$$

We can easily show that Problem 9.9 is equivalent to MCF by adding transshipment nodes. Therefore solution will converge and we will have the integrality constraints satisfied. From a practical point of view, Problem 9.9 is simple yet has potential to be very useful in the area of semi-supervised learning.

9.6 Conclusion

The k-means algorithm can be extended to insure that every cluster contains at least a given number of points. Using a cluster assignment step with constraints, solvable by linear programming or network simplex methods, can guarantee sufficient population within each cluster. A surprising result was

that constrained k-means was less prone to local minima than traditional k-means. Thus adding constraints may be beneficial to avoid local minima even when empty clusters are permissible. Constrained clustering suggests many research directions. Robust clustering can be done by simply adding an "outlier" cluster with high fixed distance that gathers "outliers" far from true clusters. Constraints forcing selected data into the same cluster could be used to incorporate domain knowledge or to enforce consistency of successive cluster solutions on related data.

We show in this chapter that it is feasible to solve constrained clustering problems by using efficient linear programming based algorithms even for the large datasets. We extend our solution to solve the constrained k-median algorithm. Results from real datasets are reported.

In addition to our original constraints on the number of points assigned to each cluster, we propose some extensions to represent pairwise assignment constraints via mathematical programming models in this chapter. Further investigations are still needed for these extensions to prove that they converge and the results satisfy the integrality constraints. Notice that such integrality constraints are expected to be satisfied without using more complex mixed-integer models. Our aim in this chapter was to show that linear programming and network simplex models can be efficiently used in solving constrained clustering problems.

Acknowledgement

Some parts of the work for this chapter were completed when Ayhan Demiriz was visiting University College of London through funding from EU PASCAL Network of Excellence.

References

[1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[2] A. Banerjee and J. Ghosh. Scalable clustering algorithms with balancing constraints. *Journal of Data Mining and Knowledge Discovery*, 13(3):365–395, 2006.

[3] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM Interna-*

0-8493-0052-5/00/$0.00+$.50

tional Conference on Data Mining (SDM-2004), pages 333–344, Lake Buena Vista, FL, April 2004.

[4] S. Basu, M. Bilenko, A. Banerjee, and R. J. Mooney. Probabilistic semi-supervised clustering with constraints. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 73–102. The MIT Press, 2006.

[5] K. P. Bennett, U. M. Fayyad, and D. Geiger. Density-based indexing for approximate nearest neighbor queries. In *Proceedings of 5th International Conference on Knowledge Discovery and Data Mining (KDD99)*, pages 233–243, New York, 1999. ACM Press.

[6] D. P. Bertsekas. *Linear Network Optimization*. MIT Press, Cambridge, MA, 1991.

[7] Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Database Theory - ICDT '99, 7th International Conference*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235, Jerusalem, Israel, January 1999. Springer.

[8] P. S. Bradley, K. P. Bennett, and A. Demiriz. Constrained k-means clustering. Technical Report MSR-TR-2000-65, Microsoft Research, May 2000.

[9] P. S. Bradley and O. L. Mangasarian. k-Plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[10] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Clustering via concave minimization. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems -9-*, pages 368–374, Cambridge, MA, 1997. MIT Press.

[11] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for metric labeling problem. *SIAM Journal of Discrete Mathematics*, 18(3):608–625, 4 2005.

[12] I. Davidson, K. L. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *Proceedings of the Tenth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 115–126, September 2006.

[13] T. De Bie and N. Cristianini. Fast sdp relaxations of graph cut clustering, transduction, and other combinatorial problems. *Journal of Machine Learning Research*, 7:1409–1436, 7 2006.

[14] J. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *Proceedings of the 40th Annual IEEE Symposium on*

the Foundations of Computer Science, Los Alamitos, CA, October 1999. IEEE Computer Society Press.

[15] J. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM*, 49(5):616–639, September 2002.

[16] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*, volume 1, pages 281–297, Berkeley, CA, 1967. University of California Press.

[17] O. L. Mangasarian. *Nonlinear Programming*. McGraw–Hill, New York, 1969. Reprint: SIAM Classic in Applied Mathematics 10, 1994, Philadelphia.

[18] MOSEK, 2007. http://www.mosek.com.

[19] J. Shanmugusundaram, U. M. Fayyad, and P. S. Bradley. Compressed data cubes for olap aggregate query approximation on continuous dimensions. In *Proceedings of 5th International Conference on Knowledge Discovery and Data Mining (KDD99)*, pages 223–232, New York, 1999. ACM Press.

[20] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1103–1110, 2000.

[21] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 577–584, 2001.