k-quantiles: L1 distance clustering under a sum constraint

Dennis Wei*

Thomas J. Watson Research Center, IBM Research, 1101 Kitchawan Road, Yorktown Heights, NY 10598, USA

A R T I C L E   I N F O

Article history:
Received 23 July 2016
Available online 1 April 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Proportional data
Compositional data
Centroid
k-means**
Workforce analytics

A B S T R A C T

This paper considers centroid-based clustering under $\ell_1$ distance in which both data points and cluster centers are subject to a sum constraint on their components. A closed-form solution is derived for the cluster center optimization problem, enabling an interpretation as a sample quantile of the cluster. An adaptive sampling initialization step is also adopted to provide a guarantee on expected clustering cost as well as empirical improvements. Experiments on synthetic data indicate that the advantages of the proposed algorithms increase as clusters become more concentrated and as the dimension increases. An application to clustering employee job role profiles highlights the utility of $\ell_1$ distance in promoting sparse, interpretable cluster centers.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Centroid-based clustering, in which each cluster is represented by a cluster center, remains a popular approach to partitioning a dataset into a given number of clusters [15]. The classic instance of this approach is $k$-means [17,18], which minimizes the squared Euclidean distance between cluster centers and data points and results in centers that are means of the points assigned to them. Other variations generalize Euclidean distance in various ways: to Mahalanobis distance [19], using kernels to yield kernel $k$-means [11], to the class of Bregman distances [3], and to other $\ell_p$ norms, notably $\ell_1$ and the associated $k$-medians problem.

This paper considers a variation in the last category, i.e. using $\ell_1$ distance, in which both points and cluster centers are further constrained to have non-negative components that sum to a constant, i.e. $\sum_{j=1}^d x_j = C$. Data points with the above properties are referred to as proportional or compositional data [1]. They commonly arise as instances of empirical distributions, e.g. normalized histograms, and discrete probability distributions in general. Real-world examples include word and topic distributions in document analysis [6] and color histograms for image segmentation [20]. Compositional data, as the name suggests, also arise in representing compositions and allocations of various kinds, extending into scientific domains such as geology [23]. This paper highlights a lesser-known application to allocations of expertise, specifically in characterizing job roles of employees in workforce analytics [26].

The $\ell_1$ distance is of interest for several reasons as well. First, it is often preferred to Euclidean ($\ell_2$) distance when robustness to outliers is a concern. Second, it produces more interpretable results for end users [16]. In a literal sense, the $\ell_1$ distance between compositions or allocations can be understood intuitively in terms of surpluses and deficits in each component. Furthermore, when applied to sparse data, using $\ell_1$ distance tends to yield sparse and hence more interpretable cluster centers, as will be illustrated in Section 5. Third, in the case of discrete probability distributions, the $\ell_1$ distance is proportional to the total variation distance, a fundamental divergence measure between distributions [24].

The problem of $\ell_1$ norm clustering under a sum constraint has been studied before in Kashima et al. [16]. In that paper, Lloyd’s alternating minimization algorithm [17,18] (popularly known as the $k$-means algorithm) was adopted, and the main innovation was in determining cluster centers to minimize $\ell_1$ distances while respecting the sum constraint. For this purpose, Kashima et al. [16] proposed an iterative pairwise-update algorithm that has lower complexity than a standard linear programming solution.

The present paper’s contributions go beyond Kashima et al. [16] in several ways. First, in Section 3.1 it is shown that the cluster center optimization problem can be solved in closed form. This results in improved computational complexity over the iterative algorithm of Kashima et al. [16], as discussed in Section 3.3. Moreover, unlike in Kashima et al. [16], the closed-form solution per-

* E-mail address: dwei@us.ibm.com

http://dx.doi.org/10.1016/j.patrec.2017.03.028
0167-8655/© 2017 Elsevier B.V. All rights reserved.
mints an interpretation as a sample quantile of the cluster. The overall proposed algorithm is thus referred to as \(k\)-quantiles, in analogy with \(k\)-means and \(k\)-medians. Second, it is observed in Section 3.2 that an adaptively sampled initialization similar to that in the \(k\)-means++ algorithm [2] can be applied to the present problem. Again in analogy with \(k\)-means++, the proposed algorithm with enhanced initialization is called \(k\)-quantiles++. Although simple, this initialization ensures that \(k\)-quantiles++ yields an expected clustering cost that is no more than a factor of \(O(\log k)\) higher than the optimal cost. Empirical gains over standard uniform initialization are demonstrated as well. Third, Section 4 herein describes experiments on synthetic data, an aspect missing from Kashima et al. [16]. Using samples generated from Dirichlet mixture distributions, these experiments allow manipulation of data dimensions and distribution parameters to explore different regimes. It is seen that the proposed algorithms offer greater advantages as the cluster concentration parameter increases and when the dimension is high compared to the number of points. Lastly in Section 5, an application to clustering employee job role profiles is presented, highlighting in particular the value of sparse cluster centers.

The remainder of the paper is organized as follows. In Section 2, the clustering problem of interest is formally stated. Section 3 describes the proposed algorithm, focusing on cluster center optimization, initialization, and computational complexity. Experiments on synthetic data are discussed in Section 4, and in Section 5, an application to clustering expertise profiles is presented. The paper concludes in Section 6.

### 2. Problem formulation

We consider the \((d - 1)\)-dimensional simplex \(S^{d-1} \subset \mathbb{R}^d\) consisting of points \(x\) with non-negative components \(x_j\) satisfying the sum constraint \(\sum_{j=1}^{d} x_j = C\) for \(C > 0\). Such points may be referred to as proportional or compositional vectors. For the most common normalized case of \(C = 1\), \(S^{d-1}\) is the standard \((d - 1)\)-simplex. Given two elements \(x, y \in S^{d-1}\), the distance between them is measured using the \(\ell_1\) norm:

\[
\|x - y\|_1 = \sum_{j=1}^{d} |x_j - y_j|.
\]

This is equal to twice the total variation distance [24] between \(x\) and \(y\) if they are viewed as discrete probability distributions over the same domain.

Given \(n\) points \(x_i \in S^{d-1}\) and an integer \(k \geq 2\), the clustering problem at hand is to partition the points into \(k\) clusters (i.e. subsets) \(A_l\), each characterized by a cluster center \(c_l\) that must also lie in \(S^{d-1}\). The cost to be minimized is the sum of the \(\ell_1\) distances between cluster centers and the points assigned to them,

\[
\phi = \sum_{l=1}^{k} \sum_{i \in A_l} \|x_i - c_l\|_1.
\]  

(1)

The desired minimum cost is denoted as \(\phi^*\).

Relaxing the constraints \(c_l \in S^{d-1}\) results in the \(k\)-medians problem, so-called because the optimal centers \(c_l\) are componentwise medians of the points in \(A_l\). These medians however will generally lie outside of \(S^{d-1}\). Alternatively, if the \(\ell_1\) norm is changed to the squared \(\ell_2\) (i.e. Euclidean) norm, then we have the \(k\)-means problem in which the optimal \(c_l\) are means of \(A_l\). In this case, the means do belong to \(S^{d-1}\) because it is a convex set, but the cost metric is different.

This paper adopts the viewpoint that the process generating the points to be clustered is unknown. Indeed, irrespective of the generating process, the objective is to obtain an optimal clustering in the sense of minimal \(\ell_1\) distances (1). This approach based on clustering cost contrasts with alternatives in which a random process that generates and labels points is assumed. In the latter setting, note that a clustering minimizing (1) may very well differ in its labelling from the generating process; for example, the labels of two points may be switched if doing so places them closer to their respective centers. For more on the generating process approach, the reader is referred to works in the vein of Bouguila et al. [8], which posit and learn mixture models for compositional data based on the Dirichlet and related distributions, and Dalton et al. [9], which develops “Bayes clusterers” that minimize clustering error directly given a random labeled point process.

A related possibility is to apply parametric and semi-parametric mixture modelling to the cost-based clustering problem at hand (1). Section 4 reports on one example of this approach in which a Dirichlet mixture model learned using expectation-maximization (EM) is compared to the proposed algorithms in terms of clustering cost. More sophisticated mixture models not considered here include ones based on the generalized Dirichlet distribution [7] and independent component analysis [25], the latter of which can adapt to unknown underlying distributions.

### 3. \(k\)-Quantiles and \(k\)-Quantiles++ algorithms

As with \(k\)-medians and \(k\)-means, we follow the alternating minimization approach of Lloyd’s algorithm [17], also known simply as the \(k\)-means algorithm. By analogy, the basic algorithm proposed herein is referred to as \(k\)-quantiles, a name that will be explained in Section 3.1. An initial set of cluster centers \(c_1, \ldots, c_k\) is obtained as described later in Section 3.2. Then the following two steps are iterated until convergence:

1. Given cluster centers \(c_l\), form clusters \(A_l\) by assigning points \(x_i\) to the nearest cluster center \(c_l\), where

   \[
   l_i \in \arg \min_{l=1,\ldots,k} \|x_i - c_l\|_1.
   \]

   If the nearest center is not unique then one of them is chosen arbitrarily.

2. Given clusters \(A_l\), update cluster centers \(c_l \in S^{d-1}\) via the optimization

   \[
   c_l = \arg \min_{c} \sum_{i \in A_l} \|x_i - c\|_1
   \]

   \[
   \text{s.t. } \sum_{j=1}^{d} c_j = C, \quad c_j \geq 0 \quad \forall j. \quad (2)
   \]

The first step above entails straightforward computation and comparison of distances. The optimization (2) in the second step is of primary interest in this paper and is discussed next.

#### 3.1. Cluster center optimization

The cluster center optimization problem (2) is convex. It can be reformulated as a linear program or solved using the specialized pairwise-update algorithm in Kashima et al. [16]. The present work goes one step further by providing an analytical expression for an optimal solution to (2), given in Theorem 1 below. Section 3.3 shows that this analytical solution is computationally advantageous compared to the iterative algorithm of Kashima et al. [16].

For convenience, we re-index the points in cluster \(A_l\) by \(i = 1, 2, \ldots, n_l\), where \(n_l\) is the number of points in \(A_l\). For each component \(j\), let \(x_{ij}\) denote the \(i\)th order statistic such that \(x_{(1)j} \leq x_{(2)j} \leq \cdots \leq x_{(n_l)j}\). Also define \(x_{(0)j} = 0 \leq x_{(1)j} \leq x_{(n_l+1)j} = C\) as a convention.
Theorem 1. Let \( i \in \{1, 2, \ldots, n_i - 1\} \) be an index such that
\[
\sum_{j=1}^{d} x_{ij} \leq C \leq \sum_{j=1}^{d} x_{i+1,j}.
\]
Then an optimal solution to the cluster center optimization problem (2) is given by
\[
c_{ij} = \theta x_{ij} + (1 - \theta) x_{i+1,j}, \quad j = 1, \ldots, d,
\]
where
\[
\theta = \frac{\sum_{j=1}^{d} x_{ij} - C}{\sum_{j=1}^{d} x_{i+1,j} - \sum_{j=1}^{d} x_{ij}} \in [0, 1].
\]

According to Theorem 1, an optimal solution to (2) may be obtained by linearly interpolating between the \( i \)-th and \((i+1)\)-th order statistics of the cluster, for each component \( j \). The index \( i \) and interpolation coefficient \( \theta \) are chosen to ensure that \( c_i \) satisfies \( \sum_{j=1}^{d} c_{ij} = C \). Since the order statistics are also sample quantiles, the optimal center \( c_i \) can be interpreted as a sample quantile of the points in cluster \( A_i \). This is similar to \( k \)-means and a generalization of \( k \)-medians in which optimal centers are means and medians respectively of the corresponding clusters. For this reason, we refer to the basic version of the proposed algorithm as \( k \)-quantiles.

Proof of Theorem 1. Define the Lagrangian function
\[
L(c, \lambda) = \sum_{i=1}^{n_i} \|x_i - c_i\|_1 + \lambda \left( \sum_{j=1}^{d} c_{ij} - C \right),
\]
obtained by relaxing only the sum constraint in (2) and assigning it the Lagrange multiplier \( \lambda \in \mathbb{R} \). Since problem (2) has a convex cost function and linear constraints, strong duality holds [4, Prop. 5.2.1]. In terms of the Lagrangian (4), this implies the existence of at least \( \lambda \) such that \( c_i \) is an optimal solution of (2) if and only if \( c_i \) is feasible and \( c_i = \arg\min_{c_i \geq 0} L(c_i, \lambda) \) [4, Prop. 5.1.1]. Since (4) is separable over \( j = 1, \ldots, d \), this Lagrangian minimization condition reduces to
\[
c_{ij} = \arg\min_{c_{ij} \geq 0} \sum_{i=1}^{n_i} \|x_i - c_{ij}\|_1 + \lambda c_{ij} = \arg\min_{c_{ij} \geq 0} L(c_{ij}, \lambda),
\]
\[
j = 1, \ldots, d.
\]
It follows that problem (2) can be solved by finding a multiplier \( \lambda \) such that the \( c_{ij} \) resulting from (5) satisfy the sum constraint in (2) and hence are feasible.

We derive the optimal solutions to the convex one-dimensional minimization in (5). The constraint \( c_{ij} \geq 0 \) is neglected for now; it will be shown that the solutions of interest are guaranteed to be non-negative. Then a necessary and sufficient condition of optimality is that the subdifferential of \( L(c_{ij}, \lambda) \) contain zero [4, Prop. B.24(f)]. Recalling the subdifferential of the absolute value function,
\[
\partial |y| = \begin{cases} [1, 1], & y \neq 0, \\ \text{sgn}(y), & y = 0, \end{cases}
\]
the subdifferential of \( L(c_{ij}, \lambda) \) is seen to be
\[
\partial L(c_{ij}, \lambda) = [2(i - 1) - n_i + \lambda, 2i - n_i + \lambda],
\]
\[
c_{ij} = x_{i(i)}, \quad i = 1, \ldots, n_i,
\]
\[
\partial L(c_{ij}, \lambda) = [2i - n_i + \lambda, 2i - n_i + \lambda],
\]
\[
c_{ij} = x_{i(i), i(i+1)}, \quad i = 0, 1, \ldots, n_i.
\]
It follows that the optimal solutions to (5) are given by
\[
c_{ij}(\lambda) \in \{x_{i(i), i(i+1)}\}, \quad \lambda = n_i - 2i, \quad i = 0, 1, \ldots, n_i,
\]
\[
c_{ij}(\lambda) = x_{i(i)}, \quad \lambda = (n_i - 2(i - 1), n_i - 2i), \quad i = 1, \ldots, n_i.
\]

Input: Data points \( x_1, \ldots, x_n \), number of clusters \( k \)
Set \( i = 1 \) and select first cluster center \( c_1 \) uniformly at random from \( x_1, \ldots, x_n \).
Compute \( \phi(x_i) \) for \( i = 1, \ldots, n \).
for \( i = 2 \) to \( k \) do
Select \( i \)-th cluster center \( c_i = x_i \) with probability \( \phi(x_i)/\sum_{i=1}^{n} \phi(x_i) \).
Update \( \phi(x_i) \) for \( i = 1, \ldots, n \).
end for

Fig. 1. D sampling algorithm.

3.2. Initialization

A common way of initializing cluster centers for Lloyd’s algorithm is to select a \( k \)-subset uniformly at random from the given points \( x_1, \ldots, x_n \). This work proposes using an alternative adaptive sampling method in which the probability of selecting a point \( x_i \) is proportional to its current contribution to the cost. Such a method is known as \( D' \) sampling in general and as \( k \)-means++ when applied to the \( k \)-means problem specifically [2]. The \( D' \) (i.e. \( D' \) for \( i = 1 \)) sampling algorithm used herein is detailed in Fig. 1, where the cost of \( x_i \) in iteration \( t \) is defined as
\[
\phi(x_i) = \min_{i=1, \ldots, d} \|x_i - c_i\|.
\]
In analogy with \( k \)-means++, we use the name \( k \)-quantiles++ to refer to the combination of \( D' \) sampling followed by the foregoing \( k \)-quantiles algorithm. The name \( k \)-quantiles will henceforth refer specifically to the version employing non-adaptive uniform initialization.

As shown in Arthur and Vassilvitskii [2], the \( D' \) sampling algorithm has the virtue of guaranteeing that the resulting cost \( \phi \) (after \( k \) iterations) is within a factor of \( O(\log k) \) of the optimal cost \( \phi^* \). This bound holds in expectation with respect to the randomness in the algorithm. We thus have the following corollary for the \( k \)-quantiles++ algorithm.
Theorem 2. The k-quantiles++ algorithm yields an expected cost that is within a logarithmic factor of the optimum.

$$\mathbb{E}[\phi] \leq 4(\ln k + 2)\phi^*.$$ 

Proof. The $\ell_1$ norm satisfies the triangle inequality by definition and appears in the cost function (1) with exponent $\ell = 1$. Hence Theorem 5.1 in Arthur and Vassilvitskii [2] applies with $\ell = 1$, yielding the stated bound for the $D$ sampling initialization alone. Since the subsequent minimizations in Lloyd's algorithm (steps 1 and 2 at the beginning of Section 3) either decrease the cost or leave it unchanged, the same bound holds for the overall algorithm. □

3.3. Computational complexity

To assess computational complexity, we consider first the cluster center optimization problem (2) for a single cluster $A_i$, and compare the analytical solution in Theorem 1 to the iterative algorithm of Kashima et al. [16]. Both solutions require sorting the values $x_{ij}, i \in A_i$ separately for each component $j = 1, \ldots, d$. This sorting generally takes $O(n_i \log n_i)$ time per component, for a total of $O(d n_i \log n_i)$. Once the order statistics $X_{(i)}$ are obtained, finding an interval $[\sum_{j=1}^{d-1} X_{(ij)}, \sum_{j=1}^{d-1} X_{(i+1,j)}]$ that contains C can be done via bisection search, computing sums as needed, in time $O(\log n_i)$. The final interpolation to compute $c_i$ requires $O(d)$ operations, so the total complexity after sorting is $O(d n_i \log n_i)$. In contrast, the algorithm of Kashima et al. [16] assumes a heap data structure and takes $O(d \log d) \log n_i$ time after sorting, which is a factor of $O(n_i \log d(\log n_i)$ higher. Including sorting, the complexity of the solution in Theorem 1 is $O(d n_i \log n_i)$ versus $O(d n_i \log(\log n_i))$ for Kashima et al. [16].

Considering all clusters $\ell = 1, \ldots, k$ together, for the proposed algorithm, the complexity of the cluster center optimizations (step 2 in Lloyd's algorithm) is $O(d \sum_i n_i \log n_i)$. This can be further bounded by $O(d n_k \log n_k)$, a worst case that is approached if almost all of the $n$ points are concentrated in one cluster. On the other hand, cluster reassignments (step 1 in Lloyd's algorithm) require $O(d n_k)$ time to compute distances between all possible pairs. Thus, the complexity of each iteration, $O(d n_k \log n_k)$, may actually be dominated by the reassignment step unless the number of clusters is small (i.e. $k = O(\log n_k)$).

Lastly for D sampling, the running time for all $k$ iterations is also $O(\log n_k)$ [2]. This can be seen from Fig. 1, noting that the minima in (8) are updated incrementally.

4. Experiments on synthetic data

Numerical experiments were conducted on synthetic data to evaluate the proposed k-quantiles and k-quantiles++ algorithms, the former using uniform initialization, the latter using D sampling.

The proposed algorithms were compared against several alternatives. First, recall from Section 3.1 that k-quantiles and the algorithm of Kashima et al. [16] solve the same cluster center optimization problem (2) within Lloyd's algorithm, and hence both should yield the same clustering results. However, k-quantiles is a more efficient algorithm as discussed in Section 3.3. Next we consider two approximate algorithms, also from Kashima et al. [16], that approximate the cluster center optimization (2). The first approximation compiles cluster centers by taking the median of points in a cluster and then normalizing to satisfy the sum constraint. The second approximation uses means for cluster centers. These two algorithms will be called k-medians and k-means for short. Note however that for all algorithms above, cluster assignments (step 1 in Lloyd’s algorithm) are made based on $\ell_1$ distance. All algorithms other than k-quantiles++ use uniform initialization.

In addition to the above, four other algorithms were tested: fuzzy c-means [5], expectation-maximization (EM) [10], agglomerative hierarchical clustering, and random clustering, the last intended as a baseline. For fuzzy c-means, the fuzzifier parameter $m$ was set to 2. The EM algorithm assumes the same Dirichlet mixture model (below in (9)) used to generate the data and alternates in the usual way for mixture models between updating cluster membership probabilities and updating mixture parameters. To estimate Dirichlet parameters, algorithms from Minka [21] were used to estimate concentration and mean parameters separately, corresponding to $\alpha$ and $\mu_i$ in (9). For both fuzzy c-means and EM, points were assigned at the end to the clusters with the largest membership probabilities. For agglomerative clustering, MATLAB’s clustertree function was run until a desired number of clusters was obtained. The $\ell_1$ (“city-block”) distance metric and centroid linkage method were selected, in keeping with the clustering objective herein. Lastly for random clustering, points are randomly partitioned into approximately equal-sized clusters. Then cluster centers are found by solving (2) as in Section 3.1.

To generate datasets $[x_i^n]_{i=1}^n$ on the standard simplex $S^{d-1}$ that are suitable for clustering, the following mixture of Dirichlet distributions' was used:

$$x_i \mid p, \alpha, \mu_1, \ldots, \mu_k \sim \sum_{l=1}^k p_l \text{Dir}(\alpha \mu_l), \quad (9)$$

where the mixture coefficients $p_1, \ldots, p_k$ sum to 1 and each Dirichlet distribution Dir($\alpha$) with parameters $\alpha = (\alpha_1, \ldots, \alpha_k)$ corresponds roughly to a true cluster. In (9), the Dirichlet parameters are factored into a scalar concentration parameter $\alpha$, common to all mixture components, and a mean vector $\mu_i$ that sums to $1$, $\sum_{j=1}^d \mu_{ij} = 1$. Given parameters $p$, $\alpha$, and $\mu_1, \ldots, \mu_k$, a dataset was formed by generating samples $x_i$, $i = 1, \ldots, n$ i.i.d. according to (9). In turn, for each dataset, $p$ was drawn from a Dir(2,2,...,2) distribution, while the means $\mu_1, \ldots, \mu_k$ were drawn i.i.d. from a uniform distribution over $S^{d-1}$ (i.e. Dir(1,1,...,1)). The distribution for $p$ was chosen to yield some diversity in the mixture coefficients and hence in cluster sizes. The concentration parameter $\alpha$ was set deterministically to values greater than $d$, the corresponding value for a uniform distribution, to ensure that the within-cluster concentration is higher than the concentration of the cluster means $\mu_1, \ldots, \mu_k$.

With the above setup, the parameters that can be controlled for each dataset are the number of points $n$, the dimension $d$, the number of mixture components $K$, and the concentration $\alpha$. One hundred (100) datasets were generated for each such combination of parameters and all results were averaged over these datasets. For each dataset, assumed number of clusters $k$, and algorithm (except for agglomerative clustering), 10 runs were executed starting from different initializations. The minimum and average costs from these 10 runs were recorded. For agglomerative clustering, which is a deterministic algorithm, only a single run is required.

The results from these experiments were observed to depend most strongly on the concentration parameter $\alpha$. Fig. 2a and b depict the clustering costs $\phi$ achieved by all the algorithms (minimum over runs then averaged over 100 datasets) for two contrasting values of $\alpha$. To facilitate comparison over a range of $k$ values, the costs are scaled by dividing by the corresponding cost for $k$-means. It is clear that in the less concentrated case, any improvements over $k$-means are no more than $2\%$ or $3\%$, whereas they can be much larger in the more concentrated case. In addition, the EM
algorithm is uniformly outperformed by k-means for all values of k. Agglomerative clustering performs well for k greater than the true number of mixture components K (which would not be known with non-synthetic data) and is the best algorithm for k ≈ K. However for k < K, performance declines markedly as may be expected for bottom-up agglomeration. Fuzzy c-means and random clustering are much worse than the other algorithms; the largest relative costs attained by these two methods are 2.1 and 2.2 in Fig. 2a (beyond the plot boundaries) and 2.3 and 4.8 in Fig. 2b. Since similar patterns for EM, agglomerative clustering, fuzzy c-means, and random clustering are observed in other experimental settings, we omit these four algorithms from the remaining plots in this section in order to focus on the more consistently competitive ones, namely k-medians, k-quantiles, and k-quantiles++. Fig. 2c and d show average clustering costs in addition to minimum costs for k-medians, k-quantiles, and k-quantiles++ under the same settings as in Fig. 2a and b. Fig. 3d shows a similar plot for an intermediate value of α/d = 10. These figures illustrate patterns that were consistently observed throughout the experiments. Namely, both k-medians and k-quantiles outperform k-means for k < K, with k-quantiles holding a slight edge over k-medians. However, as k approaches and exceeds K, the differences relative to k-means diminish and in many cases reverse. In contrast, k-quantiles++ is consistently the best-performing algorithm for all settings tested and maintains its advantage for k > K. Moreover, the improvement in average cost for k-quantiles++ is even greater than in minimum cost, suggesting that the D sampling initialization not only leads to lower costs but does so regularly as reflected in the average.

Fig. 3 illustrates performance in the same manner as in Fig. 2c and d but for different choices of the number of points n and dimension d. Increasing n closer to d in Fig. 3b results in slight increases in the margins for k-quantiles++ for all k and for k-quantiles at low k. Decreasing d from 50 to 10 reduces the differences among the algorithms. These results suggest that k-quantiles++ has a slightly greater advantage in higher-dimensional settings.

As for the number of mixture components K, Fig. 4 shows that as K increases, the patterns for k-medians and k-quantiles shift accordingly (better than k-means for k < K, similar or slightly worse for k ≥ K). The relative improvement of k-quantiles++ increases weakly with K.
5. Application to employee expertise profiles

This section discusses the application of the proposed clustering algorithms to compositional data representing the expertise of employees. Many large organizations maintain data on employee expertise, ranging in specificity from job category (e.g. sales) to detailed skills and competencies (e.g. selling a particular company product), and often structured in the form of a taxonomy [14]. The data support a variety of uses in workforce management, from locating experts in a given area [12] to assigning workers to projects [22], generating staffing plans for service engagements [13], and identifying internal candidates for job openings [26].

For this experiment, we consider employees in an IBM business division that provides information technology (IT) security services. Attention is focused on the intermediate job role level of IBM’s expertise taxonomy. The raw data consist of the job roles in the taxonomy currently held by each employee (almost always multiple), whether the job role is considered primary or not, and an assessment of proficiency on a 5-point scale. From this data, a proportional vector $x_i$ is derived to summarize the job role distribution of each employee, where $x_{ij}$ corresponds to capability in job role $j$, and total capacity (i.e. $\sum x_{ij}$) is assumed to be constant across employees.

In this work, $x_i$ is computed as described below, generalizing the method in Wei et al. [26] to incorporate assessment scores. For employee $i$, let $a_{ij} \in \{0,1,\ldots,5\}$ denote her/his assessment on job role $j$ (zero if not present). The vector components $x_{ij}$ are proportional to $a_{ij}$. To determine the normalization constant, define $p_j$ to be the set of primary job roles (occasionally more than one) and $N_j^*$ to be the set of non-primary job roles. Also define

$$A_{P_i} = \sum_{j \in P_i} a_{ij}, \quad A_{N_j^*} = \sum_{j \in N_j^*} a_{ij},$$

$$m_{P_i} = \min_{j \in P_i} a_{ij}, \quad M_{N_j^*} = \max_{j \in N_j^*} a_{ij}.$$  

Then $x_{ij}$ is given by

$$x_{ij} = \begin{cases} \frac{a_{ij}}{A_{P_i} + \min(A_{N_j^*}, m_{P_i}/M_{N_j^*} A_{N_j^*})}, & j \in P_i, \\ \frac{a_{ij}}{A_{N_j^*} + \max(A_{P_i}, m_{P_i}/m_{P_i} A_{P_i})}, & j \in N_j^*; \end{cases}$$

(10)

The reason for the form of the denominators above is to satisfy two business assumptions made in Wei et al. [26], namely (1) expertise in a primary job role should never be less than that in a non-primary job role, i.e. $x_{ij} \geq x_{ij'}$ for $j \in P_i, j' \in N_j^*$, and (2) at least 50% of the distribution should be in primary job roles, $\sum_{j \in P_i} x_{ij} \geq 1/2$.

The $n = 5810$ employees in the IT security business division collectively hold $d = 282$ job roles. However, since each employee has at most a handful of job roles, the data is very sparse (0.75% non-zeros). As a consequence, the $k$-medians algorithm from Section 4, which uses normalized medians, cannot be applied here since the median of a cluster is often all zeros and cannot be normalized. Similarly, since the domain of the Dirichlet distribution is restricted to strictly positive values, the EM algorithm based on the Dirichlet mixture model in (9) encounters numerical difficulties, even after truncating zero values to a small positive number. For the remaining algorithms ($k$-means, $k$-quantiles, $k$-quantiles++, fuzzy $c$-means, random clustering), 10 trials with different initializations were performed for each assumed number of clusters $k$ and the minimum and average costs are recorded, as before. Agglomerative clustering is also included with a single trial for each $k$.

Fig. 5a shows the resulting minimum clustering costs for all algorithms while Fig. 5b shows minimum and average costs for $k$-means, $k$-quantiles, and $k$-quantiles++. The costs for fuzzy $c$-means, agglomerative clustering, and random clustering all fail to decrease appreciably as $k$ increases; the first two algorithms are in fact worse than random clustering. In the case of agglomerative clustering, this situation is suggestive of the $k < K$ regime in Fig. 2a and b; it should be noted of course that the true number of clusters for this real-world dataset is unknown (and difficult to define). In Fig. 5b, it can be seen that the advantage of the $k$-quantiles and $k$-quantiles++ algorithms over $k$-means is generally 15%–20%, which is larger than on the synthetic Dirichlet mixtures in Section 4.

Given the patterns observed in Section 4, this could be because of the higher dimension and strong clustering of the expertise data. On the other hand, there is little difference between $k$-quantiles and $k$-quantiles++. As seen in Fig. 5c, both proposed algorithms also converge in roughly half the number of iterations on average as $k$-means.

Fig. 6 takes a closer look at the clustering results by plotting the cluster centers and distributions of employees over clusters resulting from the lowest-cost trials of the $k$-quantiles algorithm. The most apparent benefit of the proposed approach is the sparsity of the cluster centers. This sparsity was also noted in Kashima et al. [16] and is a consequence of using $\epsilon_l$ distance and extracting sample quantiles from sparse data. All but one of the cluster centers depicted have only one or two non-zero components, with correspondingly simple values of 1, 1/2, or 1/4, and collectively they span only 14 job roles (these result from the algorithm and are not pre-defined). In contrast, the centers returned by $k$-means are dense (although not fully so).

As the number of clusters increases, progressively more structure is revealed in the composition of the workforce. With only $k = 2$ clusters, the basic split is between software developers and security-related job roles. With $k = 3$ and $k = 5$ clusters, security consultants are split from security specialists, and technical support professionals and managers emerge. Additional clusters identify security delivery specialists as distinct from other security specialists, software testers, and smaller sales and client-facing job roles.
6. Conclusion

This paper has proposed a new algorithm for \( \ell_1 \) distance-based clustering of data points that obey a sum constraint. The algorithm is distinguished by a closed-form solution to the cluster center optimization problem, which allows an intuitive interpretation as a sample quantile. This interpretability was reinforced in an application involving employee job role profiles. An adaptive sampling initialization was also incorporated, providing a theoretical guarantee and further empirical improvements. Experiments on synthetic data showed how the advantages of the proposed algorithms depend on data concentration parameters, dimensions, and the number of clusters. Future work could include the application of mixture models beyond Dirichlet to the distance minimization problem treated here, for example those in Bougueda and Ziou [7], Salazar et al. [25] or new models tailored to the \( \ell_1 \) distance.

Acknowledgments

The author acknowledges Kush R. Varshney for discussions related to earlier collaboration [26] that led to the present work. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References