

K-means Clustering of Proportional Data Using L1 Distance

Hisashi Kashima

IBM Tokyo Research Laboratory, 1623-14 Shimotsuruma

Yamato 242-8502

hkashima@jp.ibm.com

Jianying Hu

Bonnie Ray

Moninder Singh

IBM T.J. Watson Research Center, 1101 Kitchawan Road, Route 134

Yorktown Heights, NY 10598

{jyhu,bonnier,moninder}@us.ibm.com

Abstract

We present a new L1-distance-based k-means clustering algorithm to address the challenge of clustering high-dimensional proportional vectors. The new algorithm explicitly incorporates proportionality constraints in the computation of the cluster centroids, resulting in reduced L1 error rates. We compare the new method to two competing methods, an approximate L1-distance k-means algorithm, where the centroid is estimated using cluster means, and a median L1 k-means algorithm, where the centroid is estimated using cluster medians, with proportionality constraints imposed by normalization in a second step. Application to clustering of projects based on distribution of labor hours by skill illustrates the advantages of the new algorithm.

1. Introduction

The k-means clustering algorithm is one of the most popular algorithms for unsupervised data partitioning [5]. The classic k-means algorithm uses the squared Euclidean distance as the distortion measure, and can be viewed as fitting a mixture of multivariate Gaussian distributions using the EM algorithm. However, in many real world applications, the squared Euclidean distance is not a good measure of distortion. Thus multiple variations of the k-means algorithm have been proposed to allow different distortion measures [4, 3].

In this paper, we propose a new variation of the k-means algorithm designed for the clustering of *proportional* vectors using L1 distance as distortion measure. Denoting a vector $\mathbf{x}_i \in \mathbb{R}^d$ by $(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)})$, we consider \mathbf{x}_i to be proportional if it describes an

allocation, and hence satisfies the following *proportionality constraints*: 1) $x_i^{(j)} > 0$ for each j ; and 2) $\sum_{j=1}^d x_i^{(j)} = 1$. Such proportional vectors are encountered in many domains, for example to capture skill allocation for different types of projects in workforce management [6], to characterize distributions of customer buying patterns across products in marketing studies, as topic distributions in text mining [1], and for color representation in video analysis [8].

In many business applications involving proportional data, the L1 distance is the preferred distortion metric because it offers intuitive and actionable interpretations. However it also leads to difficulty in the implementation of the k-mean algorithm, since there is no closed-form solution for centroid estimation under L1 distance. We formulate this problem as constrained optimization, and propose an efficient algorithm for solving it. The resulting algorithm is called *Constrained L1 k-means*. We present results of experiments demonstrating the advantages of this new algorithm compared with other variations of k-means that use approximate closed-form solutions.

The rest of the paper is organized as follows. Section 2 describes a motivating application for L1 distance-based clustering of proportional data. In Section 3 we formalize the problem specification and describe the constrained L1 k-means algorithm. Section 4 provides experimental results using real world data for the workforce management application. Section 5 concludes.

2. Skill Allocation-Based Project Clustering

A large professional service company typically has multiple active service engagements at different stages

of completion at any given time, with competing demands on its human resource supply. Therefore a methodology of generating project categorizations (taxonomy) that can be directly linked to resource requirements is crucial for ensuring timely staffing actions. However, mapping service engagements to staffing needs is in general a difficult task because these engagements are often at least partially customized, and the services portfolio is constantly evolving. Thus much effort has been devoted to developing automatic ways of inferring project categorization based on historical staffing patterns [6].

The first step of such automation is the clustering analysis of skill allocations of past engagements [6]. In this setting, each engagement is represented as a proportional vector describing allocation of total project hours over a set of predefined skills (e.g., software architect, project manager, Web portal specialist, etc.). The goal is to cluster a given set of engagements into groups with similar skill allocations. Each group is then assigned a unique engagement type, and the centroid of the corresponding cluster is considered the *standard staffing profile* for this engagement type.

This particular clustering problem has many challenging characteristics. First the data is often high dimensional - a typical IT service provider tends to utilize tens to hundreds of different skills. Secondly, the derived cluster centroids need to satisfy the proportionality constraints since they will be used as standard staffing profiles. Finally, while various distortion metrics are applicable to clustering of proportional vectors in general, L1 distance is the preferred metric here because it leads to intuitive and actionable business interpretations. Since different skills are associated with different costs, the deviation from standard staffing profiles in terms of L1 distance can be directly translated into cost differences. Furthermore, the L1 formulation lends itself naturally to a weighted extension, where the absolute difference in each component skill is weighed by its cost rate. The skill-based engagement clustering problem is thus mapped to that of L1 distance based clustering of proportional vectors.

3. Constrained L1 k-means Clustering

3.1. Problem Formulation

The classic k-means algorithm is defined using the squared Euclidean distance [5]. Here we give a generalized description of the algorithm using a generic distortion function \mathcal{D} . Given a set of data points $\mathbf{x}_i \in \mathbb{R}^d$, the k-means algorithm seeks to create a k-partitioning of the data $\{\pi_j\}_{j=1}^k$, such that if $\{\xi_j\}_{j=1}^k$ represent the

cluster centroids, the following objective function:

$$\mathbb{Q} = \sum_{j=1}^k \sum_{\mathbf{x}_i \in \pi_j} \mathcal{D}(\mathbf{x}_i, \xi_j) \quad (1)$$

is locally minimized. The algorithm finds the local minimum by starting from an initial set of centroids, and then iteratively improving upon the objective function by repeating the following two steps:

- Assign each data point \mathbf{x}_i to cluster $j^* = \operatorname{argmin}_j \mathcal{D}(\mathbf{x}_i, \xi_j)$.
- Estimate new centroids:

$$\xi_j^* \leftarrow \operatorname{argmin}_{\xi_j} \sum_{\mathbf{x}_i \in \pi_j} \mathcal{D}(\mathbf{x}_i, \xi_j). \quad (2)$$

Clearly, the centroid estimation step of the algorithm itself involves solving an optimization problem, as defined in Eqn. 2. In the classic k-means algorithm, for which \mathcal{D} is the squared Euclidean distance, the solution to Eqn. 2 is simply the sample mean. Since in many applications the squared Euclidean is not the best metric, various extensions to the classic k-mean algorithm have been developed for other types of distances, including cosine distance [4] and mutual information based metrics [3]. However, all previous extensions involve scenarios where a closed form solution exists for Eqn. 2.

For L1 distance, when the observations are unconstrained, it can be shown that the solution to Eqn. 2 is the component-wise median. However, for proportional vectors a closed form solution does not exist. Thus a new extension to the k-means algorithm is needed, where the centroid estimation step involves solving the following constrained optimization problem:

$$\begin{aligned} \xi_j^* &\leftarrow \operatorname{argmin}_{\xi_j} \sum_{\mathbf{x}_i \in \pi_j} |\mathbf{x}_i - \xi_j| \\ \text{s.t. } &|\xi_j| = 1, \quad \xi_j^{(\ell)} > 0, \ell = 1 \dots d. \end{aligned} \quad (3)$$

Since centroid estimation is invoked for each cluster during each iteration of the k-means algorithm, it is essential that solution be efficient for the overall algorithm to be practical.

3.2. Sequential Optimization for Centroid Estimation

Although the optimization problem defined by Eqn. 3 can be cast as a Linear Programming (LP) problem, the LP formulation involves $n = (2|\pi_j| + 1)d$ variables. Since even the most advanced LP solvers have a complexity of over n^3 , this can quickly become intractable when data sample size and dimension are large.

We propose a specialized algorithm that can solve this optimization problem more efficiently. The algorithm uses the fact that there is only one equality constraint, and that the objective function can be decomposed into parts, each of which involving only one variable. The basic idea behind our algorithm is the same as that of the *sequential minimal optimization* algorithm [7] for solving quadratic programming with one equality constraint for support vector machines. However, the decomposability of the objective function allows us to obtain reduced complexity.

We start by decomposing the objective function of the constrained optimization problem as follows.

$$J(\xi_j) := \sum_{\mathbf{x}_i \in \pi_j} |\mathbf{x}_i - \xi_j| = \sum_{\ell=1}^d J^{(\ell)}(\xi_j^{(\ell)}) \quad (4)$$

$$J^{(\ell)}(\xi_j^{(\ell)}) := \sum_{\mathbf{x}_i \in \pi_j} |x_i^{(\ell)} - \xi_j^{(\ell)}| \quad (5)$$

$J^{(\ell)}(\xi_j^{(\ell)})$ is the objective function for the ℓ -th dimension, so the original objective function is decomposed into the objective functions for each dimension. Note that Eqn. 5 is piecewise linear and convex, so we can obtain the optimal solution by iteratively updating the current solution in the direction leading to lower objective function. To maintain the equality constraint $|\xi_j| = 1$, we have to change at least two variables at a time.

Let us define the right gradient and the left gradient of $J^{(\ell)}(\xi_j^{(\ell)})$ as

$$g^+(\xi_j^{(\ell)}) := \lim_{\delta \rightarrow 0^+} \frac{J^{(\ell)}(\xi_j^{(\ell)} + \delta) - J^{(\ell)}(\xi_j^{(\ell)})}{\delta} \quad (6)$$

$$g^-(\xi_j^{(\ell)}) := \lim_{\delta \rightarrow 0^-} \frac{J^{(\ell)}(\xi_j^{(\ell)} + \delta) - J^{(\ell)}(\xi_j^{(\ell)})}{\delta}, \quad (7)$$

respectively.

At each step of the iterations, we find the pair of variables $\xi_j^{(\ell)}$ and $\xi_j^{(m)}$ that improve the solution the most. Assume that we move $\xi_j^{(\ell)}$ to the left and $\xi_j^{(m)}$ to the right while maintaining the equality constraint. The ratio of improvement is measured by

$$g^-(\xi_j^{(\ell)}) - g^+(\xi_j^{(m)}). \quad (8)$$

Thus we find ℓ which maximizes $g^-(\xi_j^{(\ell)})$ and m which minimizes $g^+(\xi_j^{(m)})$ to obtain the largest improvement of the objective function.

Once the best pair (ℓ, m) is determined, we move the corresponding two variables $(\xi_j^{(\ell)}, \xi_j^{(m)})$ in the direction of $(-1, 1)$ until either of them reaches the point at which

the gradient changes. In other words, the solution is updated by

$$\xi_j^{(\ell)} := \xi_j^{(\ell)} - \Delta, \quad (9)$$

$$\xi_j^{(m)} := \xi_j^{(m)} + \Delta, \quad (10)$$

where Δ is determined as

$$\Delta := \min \left\{ \xi_j^{(\ell)} - \max\{x_i^{(\ell)} | \xi_j^{(\ell)} > x_i^{(\ell)}\}, \right. \\ \left. \min\{x_i^{(m)} | \xi_j^{(m)} < x_i^{(m)}\} - \xi_j^{(m)} \right\} \quad (11)$$

The above algorithm is summarized as follows.

1. Initialize ξ_j satisfying $|\xi_j| = 1$.
2. Find the best pair (ℓ, m) which maximizes Eqn. 8. If its value is not positive, stop.
3. Update the solution by using Eqn. 9 and Eqn. 10.
4. Return to Step 2.

The computational complexity of the algorithm can be analyzed easily. Since the values of the gradients can be computed by counting the number of $x_i^{(\ell)}$ lying at the right hand side of $\xi_j^{(\ell)}$, we sort $\{x_i^{(\ell)}\}_{\mathbf{x}_i \in \pi_j}$ for $\ell = 1, \dots, d$, which requires $O(d|\pi_j| \log |\pi_j|)$ computational steps. At Step 2 in the algorithm, we find ℓ and m , respectively. This can be implemented with $O(\log d)$ complexity by using an appropriate data structure such as heaps. Also, since the objective function is piecewise linear and convex, it is easy to see that the gradients found $g^-(\xi_j^{(\ell)})$ and $g^+(\xi_j^{(m)})$ are always better than the ones that would be found in the future. Therefore the number of updates is at most $|\pi_j|$ for each dimension, resulting in $d|\pi_j|$ in total. Summarizing the above discussion, the total complexity for solving Eqn. 3 becomes $O(d|\pi_j| \log |\pi_j| + d|\pi_j| \log d) = O(d|\pi_j| \log \min\{|\pi_j|, d\})$.

4. Experiments

In this section we report results on two real world data sets representing skill allocations for past projects in two service areas within IBM. Data set 1 contains 1604 projects with 16 skill categories, and data set 2 contains 302 projects with 67 skill categories.

We compare the result of the proposed Constrained L1 k-means (*CLI k-means*) algorithm against two alternative approaches of centroid estimation in L1 distance based k-means:

1. Median L1 k-means (*MLI k-means*): The centroid is first computed as the component-wise median, then normalized to satisfy the proportionality constraints.

- Approximate L1 k-means (*AL1 k-means*): The centroid is computed as the sample mean, automatically satisfying the proportionality constraints.

Experiments were run with the predetermined number of clusters $k = 2, \dots, 10$. For each value of k , the algorithms are compared using the overall L1 distortion, computed using 10-fold validation. For each fold each clustering algorithm was run 10 times with different randomly generated initial clusters; and the one resulting in the lowest objective function was selected. The results are shown in Fig. 1 and Fig. 2. As seen in the plots, the proposed algorithm consistently outperforms both alternative approaches at all values of k .

A closer look at the cluster centroids produced by the different algorithms reveals another advantage of our approach: it leads to more interpretable cluster centroids. As is often the case with high dimensional data, our data sets exhibit a fair degree of sparsity (i.e., each group of projects typically involves only a subset of all possible skills). The approximate L1 k-means (assigning means as cluster centroids as is the case with classic k-means) failed to preserve the sparsity in the data and produced highly fragmented cluster centroids, which are difficult to interpret and use as staffing profiles. The median L1 k-means, on the other hand, exaggerated the sparsity and produced many centroids that have only a single non-zero dimension. The proposed constrained L1 k-means algorithm was able to most faithfully capture and represent the the sparsity structure in the original data, producing centroids that are directly usable as staffing profiles.

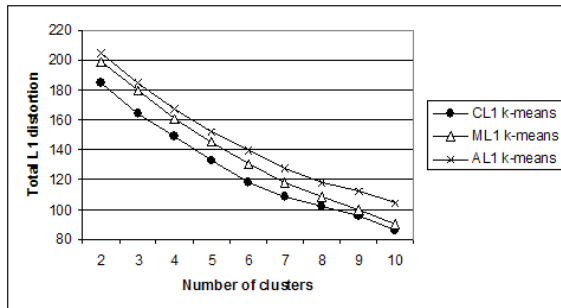


Figure 1. Results on data set 1

5. Conclusion

We have presented a new algorithm for clustering vectors of proportions using a variation of the k-means algorithm that explicitly accounts for proportionality constraints in the centroid estimation step while using L1 distance. The new algorithm provides improved L1

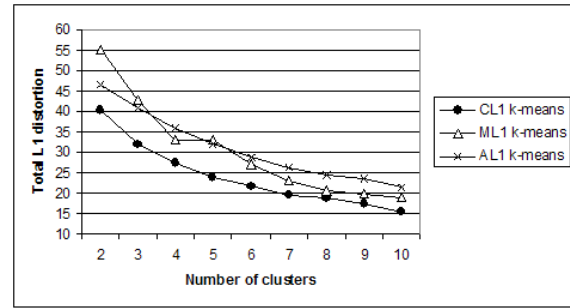


Figure 2. Results on data set 2

performance relative to other variations on L1 k-means clustering in the case of high-dimensional proportional vectors. Empirical evidence suggests that it also leads to more interpretable cluster centroids. While we have illustrated the algorithm in the context of a single domain, that of skills-based project clustering, many other domains present data for which the algorithm is pertinent, including document clustering based on topic distributions, video analysis based on color distributions, and customer segmentation based on category revenue distributions. Future work will explore application of the algorithms to these other domains, as well as formal analysis of issues of sparsity preservation. Additionally, comparisons to methods for model-based clustering of proportional vectors, such as proposed in [2], will be explored.

References

- D. Blei and J. Lafferty. Dynamic topic models. In *Proc. 2006 Int. Conf. on Machine Learning*, 2006.
- R. Datta, J. Hu, and B. Ray. Building project taxonomies for resource demand forecasting. In *Proc. Workshop on Data Mining for Business, PAKDD*, 2007.
- I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, (3):1265–1287, 2003.
- I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, (42):143–175, 2001.
- R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.
- J. Hu, B. Ray, and M. Singh. Statistical methods for automated generation of service engagement staffing plans. *IBM Journal of Research and Development*, (3), March 2007.
- J. Platt. *Sequential Minimal Optimization: a fast algorithm for training support vector machines (Technical Report MSR-TR-98-14)*. Microsoft Research, US, 1999.
- D. Zhong, H. Zhang, and S. Chang. Clustering methods for video browsing and annotation. In *Proc. IS&T/SPIE Symposium on Storage and Retrieval for Image and Video Databases*, 1996.