Generating Diverse Clusterings

Anonymous Author(s)

Abstract

Traditional clustering algorithms search for an optimal clustering based on a pre-specified clustering criterion, e.g. the squared error distortion of Kmeans. However, most real-world data sets allow alternate clusterings that are potentially useful to different users. We propose a clustering algorithm called *Diverse-Kmeans* that automatically generates a diverse set of alternate and informative clusterings. Experiments show that our proposed clustering algorithm thoroughly explores the space of clusterings, allowing it to find multiple high-quality that are missed by traditional clustering algorithms such as Kmeans.

1 Introduction

One objective of clustering is help users gain insight into their data. Most clustering algorithms try to find the optimal clustering according to specific clustering criteria. For example, Kmeans is optimized to find local minima of the squared error distortion. Many real-world data sets, however, allow multiple useful clusterings. Consider a database containing information about people's age, gender, education, job history, spending patterns, debts, medical history, etc. where clustering is used to find groups of similar people. A user who wants to find groups of consumers who will buy a car probably wants different clusters than a medical researcher looking for groups with high risk of heart disease. Different users may want different clusterings of the same data.

One way to find the best clustering for each user is to modify the clustering algorithm to incorporate expert knowledge [13, 2, 5, 9, 14]. This is done by reformulating the clustering objective function so that it satisfies constraints posed by the user. The user's background knowledge is presented either as class labels of few data points, or must-link and cannot-link constraints between pairs of points. This information guides the clustering algorithm towards appropriate partitioning of the data. However, although gathering a large amount of unlabeled data is cheap and easy, expert knowledge is limited, labor-intensive, and expensive to collect. Furthermore, clustering is often applied early during data exploration, before users understand the data well enough to define suitable clustering criteria or constraints.

In this paper our goal is to automate the process of finding multiple informative clusterings of the same data set. We propose a clustering algorithm for automatically generating multiple informative clusterings, called *Diverse-Kmeans*. The algorithm starts with an initial set of clusterings, then generates additional clusterings that are both "reasonable" in term of the squared error distortion and different from the initial set of clusterings. The paper proceeds as follows. Section 2 introduces related work. Section 3 describes our algorithm for generating a diverse set of clusterings. Section 4 describes criteria used to evaluate our algorithm. Section 5 describes three data sets used for evaluation, and Section 6 presents the empirical results. Finally, conclusions are given in Section 7.

2 Related Work

Our algorithm is an EM algorithm built upon Kmeans. Before describing the new algorithm in detail, it will be useful to first review related work in this area. Kmeans is a widely used clustering algorithm that is easy to understand and implement. In Kmeans, a data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ for $\mathbf{x}_i \in \mathbf{R}^d$ is partitioned into K clusters by iteratively relocating the cluster centers, $\{\mu_k\}_{k=1}^K$. Effectively, it minimizes the squared error distortion,

$$J_{Kmeans} = \sum_{i=1}^{N} \|\mathbf{x}_i - \mu_{\pi(\mathbf{x}_i)}\|^2, \tag{1}$$

where $\pi = \mathbf{X} \mapsto \{1, 2, ..., K\}$ is the clustering which maps each point $\mathbf{x} \in \mathbf{X}$ to its corresponding cluster. Note that Kmeans seeks local minima rather than the global minimum solutions.

In [2], the Kmeans algorithm is shown to be an EM algorithm on a mixture of K Gaussians under assumptions of identity covariance of the Gaussian, uniform priors of the mixture components and expectation under a particular conditional distribution.

In semi-supervised clustering, user's background knowledge about a data set is usually presented as a few labeled data points or as must-link and cannot-link constraints to guide the clustering algorithm to an appropriate partition. CKmeans (Constraint Kmeans) [13] is a semi-superivised variant of Kmeans. The objective function of CKmeans is reformulated to incorporate the cost incurred by violating any pairwise constraints specified by the user. Let $\mathcal{ML}$ be the set of must-link pairs where $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{ML}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in the same cluster; and $\mathcal{CL}$ be the set of cannot-link pairs where $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{CL}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in different clusters. Let $W_{\mathcal{ML}} = \{w_{ij}\}$ and $\overline{W}_{\mathcal{CL}} = \{\overline{w}_{ij}\}$ be the penalty cost for violating the constraints in $\mathcal{ML}$ and $\mathcal{CL}$ respectively. Therefore, the goal of CKmeans is to minimize the following objective function,

$$\begin{aligned} J_{CKmeans} &= \sum_{i=1}^{N} \|\mathbf{x}_i - \mu_{\pi(\mathbf{x}_i)}\|^2 + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{ML}} w_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \mathcal{I}[\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j)] \\ &+ \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{CL}} \overline{w}_{ij} [d^2_{max} - \|\mathbf{x}_i - \mathbf{x}_j\|^2] \mathcal{I}[\pi(\mathbf{x}_i) = \pi(\mathbf{x}_j)], \end{aligned} \tag{2}$$

where d_{max} is the maximum distance between any pairs of data points. Similarly to Kmeans, CKmeans seeks local minima of the above objective function.

While pairwise constraints can guide a clustering algorithm towards an appropriate partition of a data set, they can also be used to learn the underlying distance metric. In previous word on adaptive metrics for clustering [5, 1, 14], a symmetric positive-definite matrix $\mathbf{A}$ is used as a linear transformation of the features of data points. The matrix $\mathbf{A}$ is trained to minimize the distance between must-linked instances and maximize the distance between cannot-linked instances. The objective function of MKmeans (metric learning Kmeans) is modify to use the learned metric,

$$J_{MKmeans} = \sum_{i=1}^{N} (\mathbf{x}_i - \mu_{\pi(\mathbf{x}_i)})^T \mathbf{A} (\mathbf{x}_i - \mu_{\pi(\mathbf{x}_i)}), \tag{3}$$

where the matrix $\mathbf{A}$ is learned by optimized the following convex function,

$$min_{\mathbf{A}} \quad \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{ML}} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}} \quad \text{s.t.} \quad \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{CL}} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}} \geq 1, \text{ and } \mathbf{A} \succeq 0. \tag{4}$$

If $\mathbf{A}$ is restricted to a diagonal matrix, it scales each dimension by a different weight, and corresponds to feature weighting; otherwise new features are linear combinations of the original ones. Noted that MKmeans also seeks local minima of the objective function in equation 3.

In [3], both metric learning and the use of pairwise constraints are utilized in the clustering process. In MCKmeans (metric learning and constraints Kmeans), a separate weight matrix for each cluster, denoted $\mathbf{A}_k$ for cluster k, is learned to minimize the distance between must-linked instances and maximize the distance between cannot-link instances. Hence, the objective function of MCKmeans minimizes cluster dispersion under the learned metrics while reducing constraint violations,

$$\begin{aligned} J_{MCKmeans} &= \sum_{i=1}^{N} \left[\|\mathbf{x}_i - \mu_{\pi(\mathbf{x}_i)}\|^2_{\mathbf{A}_{\pi(\mathbf{x}_i)}} - \log(\det(\mathbf{A}_{\pi(\mathbf{x}_i)})) \right] \\ &+ \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{ML}} w_{ij} \left[\frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}_{\pi(\mathbf{x}_i)}} + \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}_{\pi(\mathbf{x}_j)}} \right] \mathcal{I}[\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j)] \\ &+ \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{CL}} \overline{w}_{ij} \left[(d^2_{\mathbf{A}_{\pi(\mathbf{x}_i)}})max + \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{A}_{\pi(\mathbf{x}_i)}} \right] \mathcal{I}[\pi(\mathbf{x}_i) = \pi(\mathbf{x}_j)], \end{aligned} \tag{5}$$

where $(d_{\mathbf{A}_{\pi(\mathbf{x}_i)}})max$ is the maximally distance between two points in the data set according to the $\mathbf{A}_{\pi(\mathbf{x}_i)}$ metric. We also notice that MCKmeans seeks local minima of the above objective function.

Previous work on generating a diverse set of clusterings can be found in [4]. The first approach is running Kmeans with different random initializations which leads to different local minima of the squared error distortion. However, the space of Kmeans local minima is very limited compared to the space of reasonable

clusterings. Hence, the authors proposed a feature weighting scheme to the feature vector of each data point before applying Kmeans, called FWKmeans (Feature Weighting Kmeans). The modified objective function of FWKmeans is defined as follow,

$$J_{FWKmeans} = \sum_{i=1}^{N} \|\mathbf{w} \odot \mathbf{x}_i - \mu_{\pi(\mathbf{x}_i)}\|^2, \tag{6}$$

where $\odot$ is a element-wise multiplication and the weight vector $\mathbf{w}$ is generated using the Zipf distribution [15] with a shape parameter α.

3 Diverse-Kmeans: A new algorithm for generating alternate clusterings

Instead of forcing users to define appropriate constraints to guide the clustering process to a suitable partition of a data, we propose a clustering algorithm that is able to automatically generate multiple informative clusterings of the data set.

Before describing the new algorithm, we define some measures of clustering similarity/distance and the normalized mutual information (NMI) among clusterings. First, we introduce some notation. A clustering π is a partition of the data points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ into sets $\{C_1, C_2, ..., C_K\}$ called clusters such that $C_k \cap C_l = \emptyset$ and $\bigcup_{k=1}^K C_k = \mathbf{X}$. Let the number of data points in cluster C_k be n_k. Hence, we have that $N = \sum_{k=1}^K n_k$. Let the second clustering of the same data points $\mathbf{X}$ be $\pi' = \{C'_1, C'_2, ..., C'_{K'}\}$, with cluster size $n'_{k'}$. Next, we compute the so-called confusion matrix, or contingency table M of the clustering pair (π, π'), whose the $m_{kk'}$ element is the number of points in the intersection of clusters C_k of π and $C'_{k'}$ of π', $m_{kk'} = |C_k \cap C'_{k'}|$. Next, we define the entropy associated with clustering π as

$$H(\pi) = -\sum_{k=1}^{K} P(k) \log P(k), \tag{7}$$

where $P(k) = \frac{n_k}{N}$; and the mutual information between the pair of clusterings (π, π') as

$$MI(\pi, \pi') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P(k, k') \log \frac{P(k, k')}{P(k)P(k')}, \tag{8}$$

where $P(k, k') = \frac{|C_k \cap C'_{k'}|}{N}$.

The new algorithm depends on the two clustering similarities: the normalized mutual information and the Rand Index. The first measure of the clustering similarity between the two clustering that we use is the normalized mutual information,

$$NMI(\pi, \pi') = \frac{MI(\pi, \pi')}{\sqrt{H(\pi)H(\pi')}}. \tag{9}$$

The second clustering similarity is based on counting pairs of points on which two clusterings agree or disagree. The (adjusted) Rand Index was introduced by [8] of Rand's [11] criterion,

$$Rand(\pi, \pi') = \frac{N_{00} + N_{11}}{N_{11} + N_{10} + N_{01} + N_{00}}, \tag{10}$$

where N_{11} and N_{00} are the number of point pairs that are in the same cluster, in the different clusters respectively under both π and π'; N_{10} and N_{01} are the number of point pairs that are in the same cluster under π but not under π', and the same cluster under π' but not under π respectively.

The *Diverse-Kmeans* algorithm is given in 1. The algorithm starts with an initial set of seed clusterings which we call constraint clusterings, $\Pi = \{\pi_i\}_{i=1}^m$. The algorithm will iteratively find a cluster assignment to each data point (a clustering) that has both low squared error distortion (similar to Kmeans) and is different from all clusterings in the set of constraint clusterings. Therefore, the goal of Diverse-Kmeans is to optimize the following objective function:

$$J_{Diverse\text{-}Kmeans} = \sum_{i=1}^{N} \|\mathbf{x}_i - \mu_{\pi(\mathbf{x}_i)}\|^2$$

$$+ \quad w_{\textit{diversity}} \sum_{i=1}^{m} (\textit{NMI}(\pi, \pi_i) + \textit{Rand}(\pi, \pi_i))$$

$$+ \quad w_{\textit{prior}} \sum_{k=1}^{K} (P(k) - \frac{1}{K})^2], \tag{11}$$

which includes three factors: the squared error distortion, the diversity and the prior on the size of each cluster in π. In our experiment, we use a uniform prior which prevents the clustering π to degenerate to one large cluster containing all data points. The costs $w_{\textit{diversity}}$, $w_{\textit{prior}}$ associated with the diversity factor and the cluster size prior factor are to adjust how significant each factor will affect the outcome clustering. Diverse-Kmeans seeks local minima of the objective function above. To initialize the clustering, we use one of the constraint clusterings, $\pi \in \Pi$, as the starting point. In our initial experiments, we observe that random partition initialization does not seem to work as well as using one of the constraint clusterings. Hence, better initializations usually lead to better partitions of the data.

Algorithm 1: Diverse-Kmeans

Input : $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ for $\mathbf{x}_i \in \mathbf{R}^d$; K is the number of clusters; $\Pi = \{\pi_i\}_{i=1}^{m}$ is the set of constraint clusterings; and $w_{\textit{diversity}}$, $w_{\textit{prior}}$ are the costs associated with the diversity factor and the cluster size prior factor respectively

Output: $\pi = \mathbf{X} \mapsto \{1, 2, ..., K\}$ is the clustering which maps each point $\mathbf{x} \in \mathbf{X}$ to its corresponding cluster

Initializes the clustering π

repeat

Assigns each data point $\mathbf{x}_i$ to its new corresponding cluster:

$$\pi(\mathbf{x}_i) \quad \leftarrow \quad \textit{argmin}_{k \in \{1,2,...,K\}} [\|\mathbf{x}_i - \mu_k\|^2$$

$$+ \quad w_{\textit{diversity}} \sum_{i=1}^{m} (\textit{NMI}(\pi, \pi_i) + \textit{Rand}(\pi, \pi_i))$$

$$+ \quad w_{\textit{prior}} \sum_{k=1}^{K} (P(k) - \frac{1}{K})^2]$$

Estimates new means:

$$\{\mu_k\}_{k=1}^{K} \leftarrow \{\frac{1}{n_k} \sum_{\pi(\mathbf{X})=k} \mathbf{x}\}_{k=1}^{K}$$

until *the clustering π does not change*

4 Evaluation Criteria

Clustering performance is difficult to evaluate [12]. In supervised learning, model performance is assessed by comparing model predictions to targets. In clustering we do not have targets and usually do not know *a priori* what groupings of the data are best. This hinders discerning when one clustering is better than another, or when one clustering algorithm outperforms another.

Our goal is to generate a large diverse set of candidate clusterings which are potentially useful to different users. To evaluate how well different clustering algorithms explore the space of clusterings, we use two performance metrics: clustering diversity, and clustering quality. (Clustering quality is an objective proxy that we use in place of subjective user preferences.)

Given a set of clusterings $\Pi = \{\pi_i\}_{i=1}^{m}$, *clustering diversity* measures the average clustering distance between any pairs of clusterings.

$$\textit{Diversity}(\Pi) = \frac{2}{m(m-1)} \sum_{i<j} \frac{\textit{VIDist}(\pi_i, \pi_j) + \textit{RandDist}(\pi_i, \pi_j)}{2}, \tag{12}$$

where the Rand distance is the complement of the Rand Index defined in equation 10,

$$\textit{RandDist}(\pi_i, \pi_j) = \frac{N_{01} + N_{10}}{N_{11} + N_{10} + N_{01} + N_{00}}, \tag{13}$$

and the variation of information (VI) distance [10] measures the amount of information that is lost or gained in changing from clustering π_i to clustering π_j,

$$\textit{VIDist}(\pi_i, \pi_j) = H(\pi_i) + H(\pi_j) - 2MI(\pi_i, \pi_j). \tag{14}$$

The diversity measure estimates the volume of the clustering space explored by an algorithm. Higher diversity indicates better exploration of the clustering space.

Given a set of clusterings $\Pi = \{\pi_i\}_{i=1}^m$, *clustering quality* measures the averaged quality of each clusterings which is the ratio of inter-class cluster and intra-class cluster.

$$\textit{Quality}(\Pi) = \frac{1}{m}\sum_{i=1}^{m} \frac{\textit{Inter-Cluster}(\pi_i)}{\textit{Intra-Cluster}(\pi_i)}, \tag{15}$$

where the inter-class cluster measures the averaged separation among clusters,

$$\textit{Inter-Cluster}(\pi) = \frac{\sum_{k_1<k_2}^{K}(n_{k_1} + n_{k_2})\frac{\sum_{\pi(\mathbf{X}_i)=k_1, \pi(\mathbf{X}_j)=k_2} \|\mathbf{X}_i - \mathbf{X}_j\|}{n_{k_1} n_{k_2}}}{2N}, \tag{16}$$

and the intra-class cluster measures the averaged compactness of each cluster,

$$\textit{Intra-Cluster}(\pi) = \frac{\sum_{k=1}^{K} n_k \frac{\sum_{i<j} \|\mathbf{X}_i - \mathbf{X}_j\|}{n_k(n_k-1)/2}}{N}. \tag{17}$$

The clustering quality is an indication of how well the data is partitioned into their natural groupings. Higher clustering quality indicates better groupings of the data.

We combine the two performance measures into an overall score of the set of clusterings by taking the harmonic mean.

$$F(\Pi) = \frac{2\textit{Diversity}(\Pi)\textit{Quality}(\Pi)}{\textit{Diversity}(\Pi) + \textit{Quality}(\Pi)} \tag{18}$$

5 Data Sets

We evaluate the new clustering algorithm on three data sets: Phoneme, LetterFont, and Yale Face. The Phoneme data is from the UCI Machine Learning Repository [6]. It records 11 phonemes of 15 speakers. The LetterFont data is composed of 10 different letters under 10 different fonts. The Yale Face data is a subset of the Yale Face Databse B [7] which includes portraits of 10 different subjects seen under 9 poses and 8 illumination conditions. The LetterFont and a small sample of Yale Face data set are shown in Figure 1.

In practice, users may have only a vague idea of the desired clustering and may not be able to provide the constraints needed for semi-supervised clustering. To help demonstrate that different clusterings may be useful to different users, we classify points in each data set using two or more sets of auxiliary labels that are *external* to the clustering: phoneme spoken or the speaker identity for the Phoneme data; letters or fonts for the LetterFont data; and face identity, pose, or illumination for the Yale Face data. These labels are intended as an objective proxy for what users might consider to be good clusterings for their particular application. The auxiliary labels are meant to represent clusterings users might find useful. In no way are they intended to represent an exclusive ground truth for clustering. If such a classification existed, supervised learning would be more appropriate.

Given auxiliary labels $L = \mathbf{X} \mapsto \{1, 2, ..., C\}$, accuracy measures how a clustering performs in comparison to the auxiliary labels,

$$\textit{Accuracy}(L) = \frac{\sum_{k=1}^{K} \textit{majority}(C_k|L)}{N}, \tag{19}$$

where *majority*$(C_k|L)$ is the number of points with the plurality label in the k^{th} cluster (if label l appeared in cluster k more often than any other label, then *majority*$(C_k|L)$ is the number of points in C_k with the label l).

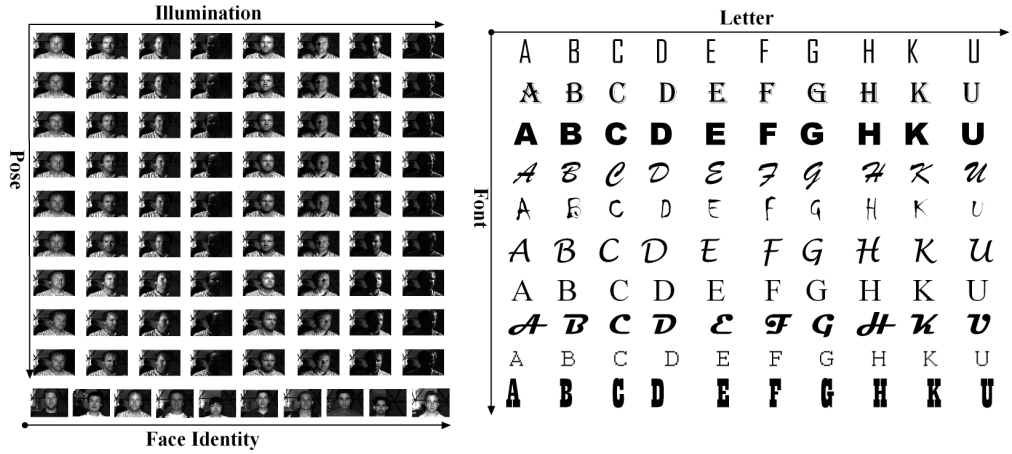


Figure 1: Yale Face and LetterFont Data Set

Table 1: Evalulation Measures of the data sets

Data Set	Phoneme			Yale Face			LetterFont		
Algorithms	Diver.	Qual.	F	Diver.	Qual.	F	Diver.	Qual.	F
Kmeans	0.298	1.168	0.475	0.144	1.444	0.262	0.132	1.816	0.245
FWKmeans	0.386	1.109	0.573	0.251	1.352	0.423	0.299	1.497	0.499
Diverse-Kmeans	0.418	1.108	0.607	0.329	1.207	0.516	0.379	1.163	0.572

6 Experiment

In this section, we present empirical results on three data sets using the new algorithm, Diverse-Kmeans, in comparison to Kmeans, CKmeans (Constraint Kmeans [13]), and FWKmeans (Feature Weighting Kmeans [4]) described in section 2. In all of our experiments, the initial set of constraint clusterings used in Diverse-Kmeans contains only one clustering, $|\Pi| = 1$.

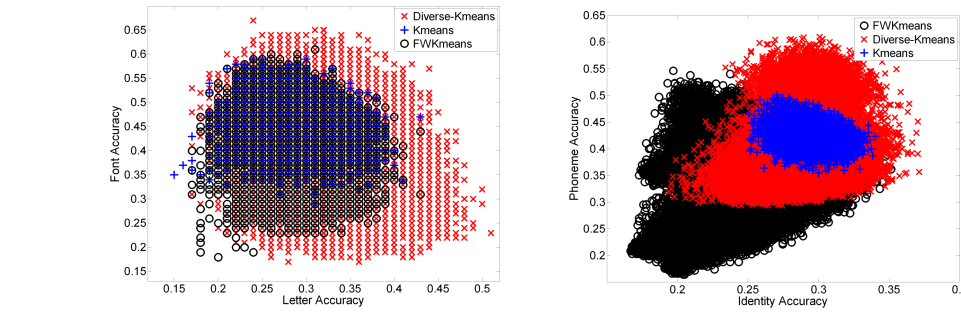
6.1 Diverse-Kmeans from Kmeans

The purpose of this experiment is to compare how well the algorithms: Kmeans, CKmeans, and FWKmeans, explore the space of alternative clusterings of the same data. First, we generate different clusterings using Kmeans with random initializations. Then Diverse-Kmean uses each clustering generated by Kmeans as its constraint clustering. Each clustering generated by the Diverse-Kmeans is different from a given Kmeans clustering. The costs $w_{diversity}$ and w_{prior} are chosen from the set $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. With FWKmeans, we vary the shape parameter of the Zipf distribution from the set, $\{0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$ [1].

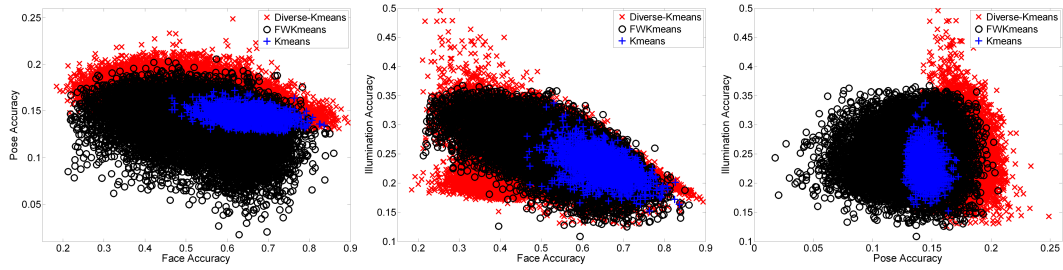
As seen in Figure 2, the clustering space explored by Kmeans with random initializations is very limited. FWKmeans explores a larger clustering space than that of Kmeans with random initializations, however it also tends to generate many clusterings in regions of low accuracy. For example, in the Phoneme data set, FWKmeans finds clusterings having low phoneme and identity accuracy. Similarly, Diverse-Kmeans also explores a larger clustering space than that of Kmeans with random initializations. However, Diverse-Kmeans is able to find clusterings with higher accuracy (using different auxiliary labels) in comparison to FWKmeans.

Table 1 shows the evaluation measures of the three data sets. We notice that there is a inverse relationship between diversity and quality, i.e. if diversity increases then clustering quality decreases. Clusterings generated by Diverse-Kmeans have high diversity, but lower average clustering quality in comparison to those generated by Kmeans and FWKmeans. However, at the expensive of sometimes generating clusterings of low quality, Diverse-Kmeans is able to much more throughly explore the space of good clusterings. Also, low quality measure does not always mean poor clustering since [4] showed that clusterings of most interest to users often are not very compact clusterings. In the overall F-score, the Diverse-Kmeans leads the other two algorithms.

[1] See [4] for the significance of different Zipf shape parameters in generating feature weights



a) LetterFont b) Phoneme



c) Yale Face

Figure 2: Exploring the Space of Clusterings: Kmeans, FWKmeans, Diverse-Kmeans

6.2 Diverse-Kmeans from CKmeans

Figure 2 shows that clusterings that are best for one criterion often have mediocre or poor performance on other criteria. Performance on different auxiliary criteria is poorly correlated, and sometimes negatively correlated. In this section we demonstrate that if Diverse-Kmeans is initialized with clusterings that excel on one dimension (e.g. clusterings with excellent accuracy predicting speaker identity on the phoneme problem), because Diverse-Kmeans is driven to find different clusterings, it will find clusterings with poor performance on this initial dimension but much better performance on an alternate dimension.

First, we generate many different clusterings using CKmeans biased to find excellent clusterings for one criterion by using the class labels for that criterion. This "cheating" yields exceptionally high quality clusterings for this criterion (clusterings colored black in Figure 3). Diverse-Kmeans is then initialized with these exceptional clusterings. Figure 3 shows that Diverse-Kmeans generates very different, yet reasonable, clusterings that tend to have high performance on the alternate criteria. Particularly on the Phoneme data, given accurate phoneme clusterings Diverse-Kmeans finds accurate identity clustering, and vice versa. This confirms our assumption that clusterings different from each other on auxiliary criteria tend also to be different from each other in terms of the clustering distance which is used internally by Diverse-Kmeans.

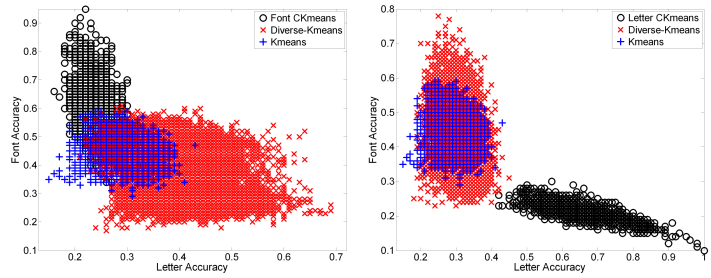
7 Conclusion

In this paper, we address the problem that for many real-world data sets there are different clusterings that potentially are useful for different users. We propose a clustering algorithm, called *Diverse-Kmeans*, that automatically generates a diverse set of alternate clusterings. Our experimental results show that Diverse-Kmeans is able to find many quality clusterings that are missed by traditional clustering algorithms such as random restart Kmeans. In addition, in comparison to FWKmeans (Feature Weighting Kmeans [4]), a recently developed clustering algorithm for generating a diverse set of clusterings, the new algorithm is able to explore a more diverse clustering space that includes better clusterings.

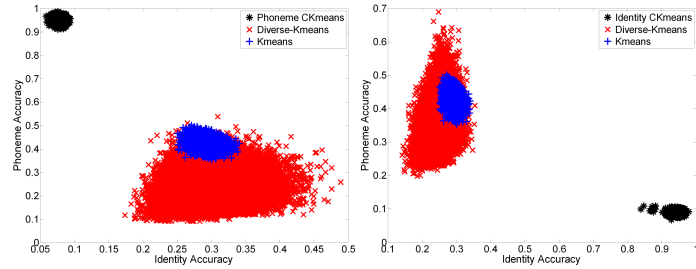
References

[1] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *Proceedings of 20th International Conference on Machine Learning*, 2003.

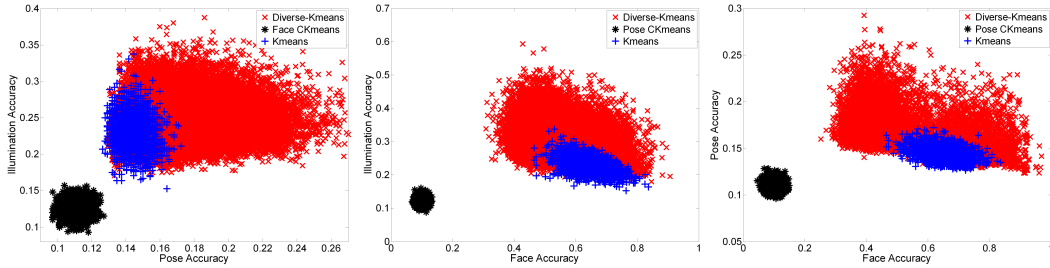
[2] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Semi-supervised clustering by seeding. In *Proceedings of 19th International Conference on Machine Learning*, 2002.



a) LetterFont



b) Phoneme



c) Yale Face

Figure 3: Diverse-Kmeans from CKmeans

[3] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of 21th International Conference on Machine Learning*, 2004.

[4] Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. Meta clustering. In *Proceedings of the Sixth International Conference on Data Mining*, 2006.

[5] David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised clustering with user feedback. In *Technical Report TR2003-1892*, Cornell University.

[6] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[7] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[8] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[9] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of 19th International Conference on Machine Learning*, 2002.

[10] Marina Meila. Comparing clusterings by the variation of information. In *Proceedings of the Sixteenth Annual Conference of Computational Learning Theory (COLT)*, 2003.

[11] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[12] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Advances in Neural Information Processing System 12*, 1999.

[13] Kiri Wagsta, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of 18th International Conference on Machine Learning*, 2001.

[14] Eric P. Xing, Andew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, 2003.

[15] George K. Zipf. *The Psychobiology of Language*. Houghton-Mifflin, New York, NY, 1935.