Linear Manifold Clustering

Robert Haralick and Rave Harpaz

Pattern Recognition Laboratory
The Graduate Center, City University of New York,
365 Fifth Avenue New York, NY 10016, USA
haralick@ptah.gc.cuny.edu,rbharpaz@sci.brooklyn.cuny.edu

Abstract. In this paper we describe a new cluster model which is based on the concept of linear manifolds. The method identifies subsets of the data which are embedded in arbitrary oriented lower dimensional linear manifolds. Minimal subsets of points are repeatedly sampled to construct trial linear manifolds of various dimensions. Histograms of the distances of the points to each trial manifold are computed. The sampling corresponding to the histogram having the best separation between a mode near zero and the rest is selected and the data points are partitioned on the basis of the best separation. The repeated sampling then continues recursively on each block of the partitioned data. A broad evaluation of some hundred experiments over real and synthetic data sets demonstrates the general superiority of this algorithm over any of the competing algorithms in terms of stability, accuracy, and computation time.

1 Introduction

The problem of clustering can be loosely defined as the partitioning of a set of points in a multidimensional space into groups (*clusters*) such that the points in each group are similar to one another. Finding these clusters is important because their points correspond to observations of different classes of objects that may have been previously unknown. A second kind of latent information that may be of interest, are correlations in a data set. A correlation is a linear dependency between two or more attributes of the data set. Knowing about the existence of a relationship between attributes may enable us to learn hidden causalities. For example, the influence of the age of a patient and the dose rate of medication on the length of his disease.

Due to recent technology advances in data collection many applications of clustering are now characterized by high dimensional data, some of whose dimensions are non-information carrying. Thus, clusters or correlations may be visible only in linear combinations of subsets of the dimensions. Conventional clustering algorithms such as K-means [10], and DBSCAN [8] are "full-dimensional" in the sense that they give equal relevance to all dimensions, and therefore are likely to fail when applied to such high-dimensional data. Subspace clustering is an extension to traditional clustering in that it attempts to find clusters embedded in different subspaces of the same data set. A subspace cluster consists of a subset of points and a corresponding subset of attributes (or linear combinations

of attributes), such that these points form a dense region in a subspace defined by the set of corresponding attributes. Most subspace clustering methods such as CLIQUE [3], MAFIA [11], and PROCLUS [1] are restricted to finding clusters in subspaces spanned by some subset of the original measurement features. However, examination of real data often shows that points tend to get aligned along arbitrarily oriented subspaces. ORCLUS [2] the most relevant algorithm to our problem is an extension to PROCLUS which allows clusters to exist in arbitrarily oriented subspaces. Dasgupta [5] presents two important results related to *random projections* which have implications to clustering in high dimensional spaces. These results show that it is possible to project high dimensional data into substantially lower dimensions while still retaining the approximate level of separation between clusters. In a recent paper Haralick at el. [6] use random projections in the context of projection pursuit to search for interesting one dimensional projections that reveal inter-cluster separations. Their algorithm, called HPCluster, uses an hierarchical approach that repeatedly bi-partitions the data set using interesting one-dimensional projections.

In this paper we describe a new cluster model that is based on the concept of linear manifolds. It takes into account both linear dependencies among features and distances betweens points. In section 2 we formalize our model of a cluster. Based on this model, we present in section 3 the algorithm-LMCLUS. In section 4 we present a broad evaluation of LMCLUS applied on synthetic and real data sets, and in section 5 we conclude the paper giving hints on future work.

2 The Cluster Model

The goal is to find clusters with an intrinsic dimensionality that is much smaller than the dimensionality of the data set, and that exhibit correlation among some subset of attributes or linear combinations of attributes. The cluster model which we propose has the following properties: the points in each cluster are embedded in a lower dimensional linear manifold [1]. The intrinsic dimensionality of the cluster is the dimensionality of the linear manifold. The manifold is arbitrarily oriented. The points in the cluster induce a correlation among two or more attributes (or linear combinations of attributes) of the data set. In the orthogonal complement space to the manifold the points form a compact densely populated region.

Definition 1 (Linear Manifold). *L is a **linear manifold** of vector space V if and only if for some subspace S of V and translation $t \in V$, $L = \{x \in V | for\ some\ s \in S, x = t + s\}$. The dimension of L is the dimension of S.*

Definition 2 (Linear Manifold Cluster Model). *Let D be a set of d-dimensional points, $C \subseteq D$ a subset of points that belong to a cluster, x some point in C, $b_1, \ldots, b_d$ an orthonormal set of vectors that span $\mathbb{R}^d$, $(b_i, \ldots, b_j)$ a matrix whose*

[1] A linear manifold is a translated subspace. A subspace is a subset of points closed under linear combination.

columns are the vectors $b_i, \ldots, b_j$*, and* μ *some point in* $\mathbb{R}^d$*. Then each* $x \in C$ *is modeled by,*

$$x = \mu + (b_1, \ldots, b_l)\lambda + (b_{l+1}, \ldots, b_d)\psi, \tag{1}$$

where μ *is the cluster mean,* λ *is a zero mean random* $l \times 1$ *vector whose entries are i.i.d.* $U(-R/2, +R/2)$*,* ψ *is a zero mean random vector with small variance independent of* λ*, and* R *is the range of the data.*

The idea is that each point in a cluster lies close to an l-dimensional linear manifold, which is defined by $\mu + span\{b_1, \ldots, b_l\}$. It is easy to see that μ is the cluster mean since

$$E[x] = E[\mu + (b_1, \ldots, b_l)\lambda + (b_{l+1}, \ldots, b_d)\psi] =$$

$$\mu + (b_1, \ldots, b_l)E[\lambda] + (b_{l+1}, \ldots, b_d)E[\psi] = \mu + (b_1, \ldots, b_l)\mathbf{0} + (b_{l+1}, \ldots, b_d)\mathbf{0} = \mu$$

Classical clustering algorithms such as K-means take $l = 0$ and therefore omit the possibility that a cluster has a non-zero dimensional linear manifold associated with it. In the manifold we assume the points are uniformly distributed in each direction according to $U(-R/2, +R/2)$. It is in this manifold that the cluster is embedded, and therefore the intrinsic dimensionality of the cluster will be l. The third component models a small disturbance, or error factor associated with each point in the manifold. The idea is that each point may be perturbed in directions that are orthogonal to the manifold, i.e., the vectors $b_{l+1}, \ldots, b_d$. We model this behavior by requiring that ψ be a $(d-l) \times 1$ random vector, normally distributed according to $N(\mathbf{0}, \Sigma)$, where the largest eigenvalue of Σ is much smaller than R. Since the variance along each of these directions is much smaller than the range R of the embedding, the points are likely to form a compact and densely populated region, which can be used to cluster the data.

Figure 1 is an example of data set modeled by eq. (1). The data set contains three non-overlapping clusters, where C_1, C_2 which are almost planner are embedded in 2D manifolds. Their points are uniformly distributed in the manifold and they include a random error element in the orthogonal complement space to the manifold. Similarly, C_3 an elongated line like cluster, is embedded in a 1D linear manifold.

3 The Algorithm

LMCLUS can be viewed as an hierarchical-divisive procedure, which marries the ideas of random projection via sampling and histogram thresholding, in order to detect clusters embedded in lower dimensional linear manifolds. It expects three inputs: L, an estimate of the highest dimension of the manifolds in which clusters may be embedded. $\hat{K}$, an estimate of the largest number of clusters expected to be found, which is used to compute the number of trial manifolds of a given dimensionality that will be examined in order to reveal the best possible partitioning of the data set. Γ, a sensitivity threshold which is used to determine whether or not a partitioning should take place. We note that unlike

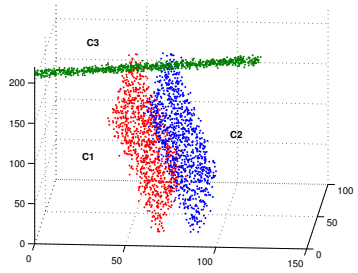


Fig. 1. A data set demonstrating the concept of linear manifold clusters.

related methods $\hat{K}$ does not impose a restriction on the number of clusters the algorithm actually finds. The output of LMCLUS is a set of labeled clusters together with the intrinsic dimensionality of each cluster. Knowing the dimensionality associated with each cluster can then be used with methods such as PCA to model the data in each cluster. The algorithm operates by detecting one cluster at a time and successively reapplying itself on the remaining set of points. It iterates over a range of manifold dimensionalities, in an a priori fashion, starting from the lowest-1, and terminating with the highest-L. For each dimensionality the algorithm invokes a procedure which we call *FindSeparation* in an attempt to reveal separations among subsets of the data. Its underlying idea is to successively randomly sample subsets of points that can define a linear manifold of a given dimension. Of the linear manifolds constructed, the one closest to a substantial number of data points is selected. The proximity of the input data points to the manifold is captured by a distance histogram. If the manifold indeed has some subset of points near it, then the distance histogram will reveal a mixture of two distributions. One of the distributions has a mode near zero and is the distribution of distances of points that potentially belong to a cluster, and the other is the distribution of the remaining points in the data set. The problem of separating the cluster points from the rest is then cast into a histogram thresholding problem, which is solved using Kittler and Illingworth minimum error thresholding technique [9]. *FindSeparation* returns four values γ- which is a measure of the "goodness" of the separation, τ- a proximity threshold that is computed from the histogram and is used to split the data, B- the basis of the manifold which exposed the separation, and x_0-the origin of the manifold. When γ exceeds the value of the sensitivity threshold Γ, indicating that a worthy separation has been found, then the data set is split according to τ. This split corresponds to the partitioning of all the points which are located close enough to a manifold, i.e. all points that potentially belong to a given cluster, and those that belong to other clusters. In addition the dimensionality of the manifold which revealed the separation, corresponding to the intrinsic dimensionality of the cluster is recorded. An attempt to further partition the cluster which may

consist of sub-clusters is executed by reapplying *FindSeparation* until the cluster can not be further separated. At this point the algorithm will attempt to partition the cluster in higher dimensions, a process which will continue until the dimension limit L is reached. When L is reached we have a subset of points that cannot be partitioned any more, and declare that a cluster is found. We note that if outliers exist then the last partition will contain this set of points. By definition outliers do not belong to any cluster and therefore will remain the last group of points to be associated to any other group. Moreover, since they are unlikely to form any clusters the algorithm will not be able to partition them, and therefore will all be grouped together.

Sampling Linear Manifolds. To construct an l-dimensional linear manifold by sampling points from the data we need to sample $l+1$ points. Let $x_0, \ldots x_l$ denote these points. We choose one of the points x_0 as the origin. Then the l vectors spanning the manifold are obtained by $x'_i = x_i - x_0$ where $i = 1 \ldots l$. Assuming each of these sampled points came from the same cluster, then according to eq. (1)

$$x'_i = (\mu_0 + B\lambda_i + B_c\psi_i) - (\mu_0 + B\lambda_0 + B_c\psi_0) = B(\lambda_i - \lambda_0) + B_c(\psi_i - \psi_0)$$

where $B = (b_1, \ldots, b_l)$ and $B_c = (b_{l+1}, \ldots, b_d)$. If the cluster points did not have an error component, that is, they all lie at distance zero from the manifold, then sampling any $l+1$ which are linearly independent, and belong to the same cluster would enable us to reconstruct B. Therefore in order to get a good approximation of B we would like each of the sampled points to come from the same cluster, and to be as close as possible to the linear manifold spanned by the column vectors of B. In other words we would like each of the $l+1, \ldots, d$ components of each x'_i to be close to zero, and this occurs when $\psi_i - \psi_0 \approx \mathbf{0}$. A good indication as to why this is likely to occur when the sampled points come from the same cluster, is given by the fact that $E[\psi_i - \psi_0] = \mathbf{0}$. Therefore the problem of sampling a linear manifold that will enable us to separate a cluster from the rest of the data basically reduces to the problem of sampling $l+1$ points that all come from the same cluster.

Assuming the data set contains K clusters all having approximately the same number of points. Then the probability that a sample of $l+1$ points all come from the same cluster is approximately $\left(\frac{1}{K}\right)^l$. The probability that out of n samples of $l+1$ points, none come from the same cluster, is approximately $(1-(1/K)^l)^n$ and $1-(1-(1/K)^l)^n$ will be the probability that at least for one of the samples all of its $l+1$ points come from the same cluster. Therefore the sample size n required such that this probability is greater than some value $1-\epsilon$ is given by

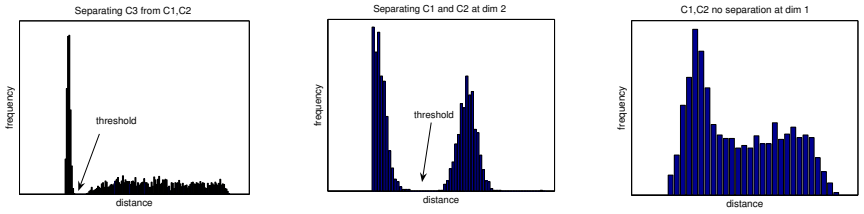
$$n \geq \frac{log\ \epsilon}{log(1-(1/K)^l)} \tag{2}$$

Thus, by computing n given ϵ, and $K = \hat{K}$ we can approximate a lower bound on the number samples required or trial manifolds that will be examined. Note that by varying $\hat{K}$ we can tradeoff accuracy with efficiency.

For each sample of points $(x'_1, \ldots, x'_l)$ we construct an orthonormal basis B of a linear manifold, measure distances to it, and then using Kittler and Illingworth's method we compute a threshold τ. Of all possible thresholds corresponding to different linear manifolds we prefer the one which induces the best separation. That is to say, the one which induces the largest *discriminability* given by $\frac{(\mu_1(\tau)-\mu_2(\tau))^2}{\sigma_1(\tau)^2+\sigma_2(\tau)^2}$, and the one which causes the deepest broadest minimum in the Kittler and Illingworth criterion function J [9]. This can be measured by the difference/depth of the criterion function evaluated at τ and the value evaluated at the closest local maxima τ', i.e., $depth = J(\tau') - J(\tau)$. Thus, our composite measure of the "goodness" of a separation is given by

$$\gamma = discriminability \times depth \tag{3}$$

A set of typical histograms generated during the clustering process are depicted in Fig. 2, corresponding to a subset of the histograms used to cluster the data set given in Fig. 1.



(a) (b) (c)

Fig. 2. Histograms used to separate the clusters from Fig. 1. (a) C_3 is separated from C_2 and C_3 by sampling 1D linear manifolds. (b) C_1 is separated from C_2 by sampling 2D linear manifolds. (c) a histogram for which no separation can be found.

4 Empirical Evaluation

LMCLUS as well as three other related methods: DBSCAN a representative of the full-dimensional clustering methods, ORCLUS a representative of subspace clustering methods, and HPCluster a random projection based clustering method, were implemented in C++. The aim of the experiment was to evaluate LMCLUS's performance in comparison to the other methods with respect to accuracy, efficiency, scalability and its stability as a stochastic algorithm.

4.1 Synthetic Data Generation

In order to generate clusters embedded in different arbitrary oriented linear manifolds of various dimensions, and following the model given by eq. (1) we

used a method similar to one described in the ORCLUS paper. The underlying idea is to first generate the clusters in an axis parallel manner and then randomly translate and rotate each cluster to achieve the effect of an arbitrary orientation in the space. A candidate data set that we would like to produce is one in which the clusters are relatively close in some subspace with minimal overlap, and yet sparse enough that canonical clustering algorithms would not be able to detect. We also used the *cluster sparsity coefficient* proposed in the ORCLUS paper to measure the relative hardness of clustering a given data set, and selected only data sets which yielded a measure within a specific range.

4.2 Accuracy

To measure the accuracy of LMCLUS we have generated several dozen synthetic data sets of various sizes, space/manifold dimensionalities, and number of clusters. Table 1 summarizes the properties of fifteen representative data sets, along with the performance of each of the algorithms when applied to these data sets. The ones marked with a star ($'*'$) denote *star* data sets due to their star like geometry, which are likely to present difficulties to many clustering algorithms. Accuracy was measured by means of a *Confusion Matrix*. In addition the amount of time (in hours, minutes, and seconds) required by each algorithm to cluster a data set was recorded. These results clearly demonstrate LMCLUS's superiority over the other clustering algorithms. LMCLUS was the only one able to discover (over 85% accuracy) all the clusters. DBSCAN's poor performance emphasizes the ineffectiveness of distance metrics that utilize the full space. Note that only LMCLUS and ORCLUS were able to handle the *star* clusters. However requiring the number of clusters and the dimensionality of the subspaces in which the clusters are embedded makes ORCLUS impractical for real data sets. The fact that HPCluster was not able to cluster the *star* data sets also comes at no surprise since it searches for 1D projections, and any 1D projection of these type of data sets will only reveal unimodal distributions. However its ability to cluster well the first type of data sets supports the concept of random projections which LMCLUS also implements. In terms of running time, LMCLUS ranked second after HPCluster. The remarkable low running times of HPCluster can be attributed to the fact that it is based on a stochastic procedure which tries a constant number of 1D projections to discover inter-cluster separations, and thus invariant to the size of the data set. Nonetheless LMCLUS runs faster than the other algorithms on seven of the fifteen data sets, and when compared to ORCLUS and DBSCAN only, demonstrates a significant gain in efficiency, especially when applied on large or high dimensional data sets.

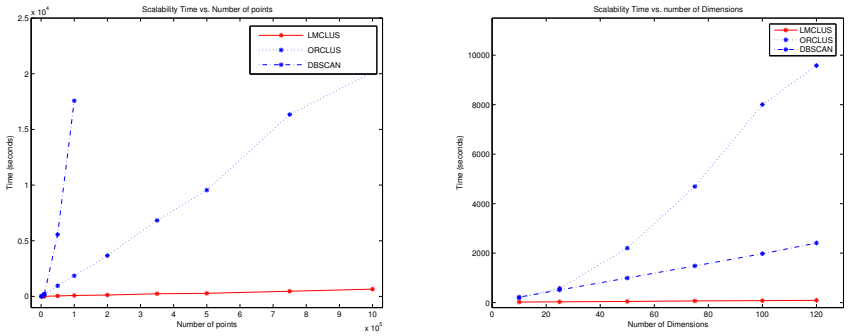
4.3 Scalability

We measured the scalability of LMCLUS in terms of size and dimensions. In the first set of tests, we fixed the number of dimensions at ten, and the number of clusters to three, each of which was embedded in a three-dimensional manifold. We then increased the number of points from 1,000 to 1,000,000. In the second

Table 1. Data set properties along with accuracy and running time results used for the Accuracy benchmark.

	size	clusters	dim	LM dim	LMCLUS	ORCLUS	DBSCAN	HPCluster
D_1	3000	3	4	2-3	95% / 0:0:08	80% / 0:0:22	34.6% / 0:0:9	72% / 0:0:51
D_2	3000	3	20	13-17	98.4% / 0:0:33	58.8% / 0:2:18	65.5% / 0:0:36	97.4% / 0:1:39
D_3	30000	4	30	1-4	100% / 0:15:38	64.9% / 1:5:30	100% / 1:31:52	99.3% / 0:1:32
D_4	6000	3	30	4-12	99.9% / 0:9:22	98.3% / 0:8:20	66.5% / 0:3:49	97.1% / 0:0:12
D_5	4000	3	100	2-3	100% / 0:0:20	87.9% / 0:54:30	65.3% / 0:5:24	99% / 0:3:54
D_6	90000	3	10	1-2	99.99% / 0:0:29	100% / 0:29:02	66.7% / 4:58:49	100% / 0:1:23
D_7	5000	4	10	2-6	99.24% / 0:2:05	99.3% / 0:2:41	74.1% / 0:0:54	96% / 0:0:35
D_8	10000	5	50	1-4	99.9% / 0:1:42	63.64% / 1:33:52	100% / 0:17:00	99.2% / 0:3:43
D_9	80000	8	30	2-7	99.9% / 3:12:46	96.9% / 13:30:30	100% / 10:51:15	99.9% / 0:4:57
D_{10}	5000	5	3	1-2	86.5% / 0:0:48	68.2% / 0:0:45	59.6% / 0:0:5	78% / 0:0:33
$*D_{11}$	1500	3	3	1	98.5% / 0:0:01	99.6% / 0:0:10	42.6% / 0:0:02	33.3% / 0:0:52
$*D_{12}$	1500	3	3	2	97% / 0:0:02	99% / 0:0:11	33.8% / 0:0:02	33.3% / 0:0:26
$*D_{13}$	1500	3	7	3	97.7% / 0:0:05	99.1% / 0:0:17	33.9% / 0:0:04	33.3% / 0:0:34
$*D_{14}$	5000	5	20	4	99.9% / 0:5:46	100% / 0:10:42	21.1% / 0:1:39	20% / 0:1:30
$*D_{15}$	4000	4	50	3	99% / 0:9:14	100% / 0:25:52	25% / 0:2:34	25% / 0:3:20

set of tests we fixed the number of points, and clusters as before, but increased the number of dimensions from 10 to 120. Fig. 3 is a plot of the running times of LMCLUS in comparison to the other algorithms. The figure shows that in practice, for data sets with a small number of clusters which are embedded in low dimensional manifolds, LMCLUS, like ORCLUS scales linearly with respect to the size of the data set. This can be attributed to the sampling scheme it uses and to the fact that each cluster that is detected is removed from the data set. We note however that as the dimensionality of manifolds increases, performance is likely to degrade. The figure also shows that LMCLUS, like DBSCAN scales linearly with respect to the dimensionality of the data set. Combined together, linearity in both the size and dimensionality of the data set makes LMCLUS one of the fastest algorithms in its class.



(a) (b)

Fig. 3. Scalability, (a) running time vs. data size. (b) running time vs. dimensionality.

4.4 Real Data and Applications

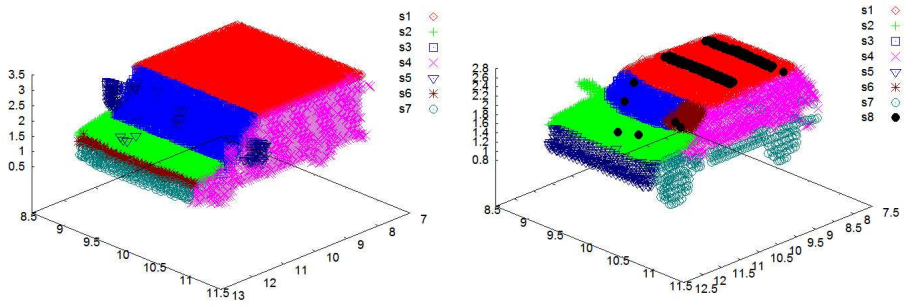
Time Series Clustering/Classification. In this experiment we applied LMCLUS on a Time Series data set obtained from the UCI KDD Archive [7] consisting of 600 examples with 60 attributes each, divided into 6 different classes. The donors of this data set claim the this is a good data set to test time series clustering because Euclidean distance measures will not be able to achieve good accuracy. LMCLUS was able to achieve an average of 87% with a high of 89% accuracy. ORCLUS was only able to achieve a high of 50% accuracy, while DBSCAN with extensive tuning of its parameters achieved a high of 68% accuracy, and HPCluster a high of 63.5%.

Handwritten Digit Recognition. The data used in this experiment consists of 3823 handwritten digit bitmaps of 64 attributes each, obtained from the UCI Machine Learning Repository [4]. We divided the data into the even and odd digits, and clustered each separately. LMCLUS was able to achieve an average of 95% and 82% for the even and odd digits respectively, whereas DBSCAN 82% and 58%, ORCLUS 84.7% and 82.9%, and HPCluster 50.3% and 93%.

E3D Point Cloud Segmentation. DARPA's "Exploitation of 3D Data" identification system must take as input a 3D point cloud of a military target and then compare it to a database of highly detailed 3D CAD models. The first step to accomplish this task usually involves segmenting the targets into their constituent parts. In this experiment we demonstrate LMCLUS's usefulness as a segmentation procedure. Specifically, LMCLUS was applied on 3D vehicle point cloud CAD models obtained from ALPHATECH Inc., as these provide a similar level of complexity, to that of military vehicles. The applicability of LMCLUS to this problem results from the fact that the surfaces constituting the vehicles closely correspond to 2D linear manifolds embedded in a 3D space. The results of this experiment applied on two different vehicles are depicted in Fig. 4. These results clearly demonstrate LMCLUS's ability to identify with high precision 2D linear manifolds.

5 Conclusion

In this paper we explored the concept of linear manifold clustering in high dimensional spaces. We proposed a new cluster model and showed its relevance to subspace and correlation clustering. Based on this model we presented our algorithm LMCLUS, and demonstrated its superiority over methods such as ORCLUS, DBSCAN, and HPCluster for linear manifold clustering. In addition we presented a successful application of our algorithm to the problem of 3D point cloud segmentation. In the feature we plan to investigate the applicability of linear manifold clustering to microarray gene expression clustering, and its usefulness as a tool for modeling high dimensional probability distributions.



(a) (b)

Fig. 4. (a) 2D view of a segmented Aeromate delivery van 3D point cloud (b) 2D view of a segmented Ford Explorer 3D point cloud.

References

1. Charu C. Aggarwal, Joel L. Wolf, Philip S. Yu, Cecilia Procopiuc, and Jong Soo Park. Fast algorithms for projected clustering. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 61–72. ACM Press, 1999.
2. Charu C. Aggarwal and Philip S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 70–81, 2000.
3. Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 94–105, 1998.
4. C.L. Blake and C.J. Merz. UCI Repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html (1998).
5. S. Dasgupta. Learning mixtures of gaussians. In *In Proc. of the 40th Ann. IEEE Symp. on Foundations of Computer Science*, 1999.
6. R.M. Haralick, J.E.Rome, and Alexei Miasnikov. A hierarchical projection pursuit clustering algorithm. In *17th International Conference on Pattern Recognition (ICPR)*, 2004.
7. S. Hettich and S. D. Bay. The UCI KDD archive. http://kdd.ics.uci.edu (1999).
8. Ester M.and Kriegel H.P., Sander J., and Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf.on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, Portland, OR, 1996.
9. J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recogn.*, 19(1):41–47, 1986.
10. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Berkeley, University of California Press, 1967.
11. H. Nagesh, S. Goil, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets, 1999.