A Uniformity Criterion and Algorithm for Data Clustering

Sanketh Shetty and Narendra Ahuja
Beckman Institute for Advanced Science and Technology
University of Illinois, Urbana-Champaign, Urbana IL 61801, USA
{sshetty2,n-ahuja}@uiuc.edu

Abstract

We propose a novel multivariate uniformity criterion for testing uniformity of point density in an arbitrary dimensional point pattern . An unsupervised, nonparametric data clustering algorithm, using this criterion, is also presented. The algorithm relies on a relatively general notion of cluster so that it is applicable to clusters of relatively unrestricted shapes, densities and sizes. We define a cluster as a set of contiguous interior points surrounded by border points. We use our uniformity test to differentiate between interior and border points. We group interior points to form cluster cores, and then identify cluster borders as formed by the border points neighboring the cluster cores. The algorithm is effective in resolving clusters of different shapes, sizes and densities. It is relatively insensitive to outliers. We present results for experiments performed on artificial and real data sets.

1 Introduction

This paper is about detecting clusters defined by uniformly distributed points in a D-dimensional space. A cluster may have the dimensionality D of the parent space, or it may occupy a lower, d-dimensional subspace. The volume occupied by the cluster may have an arbitrary shape, but the points within it have an unknown, uniform density. We achieve this by determining whether the neighborhood of a point is characterized by a uniformly dense point placement, in any d-dimensional, linear subspace, $d \leq D$. If so, we group sets of contiguous points, with the same density and within the same subspace, into clusters. The main contributions of this algorithm and their advantages over most existing ones are as follows. First, we present a new criterion to test the uniformity of point placement in a given neighborhood. Second, unlike our approach, most existing algorithms are sensitive to the shapes of the clusters to be detected. Third, most existing algorithms require the user to specify certain input parameters whose values are critical to the performance but require familiarity with the data being analyzed. The clustering performance of the algorithm we present is stable with respect to the few input parameters the user needs to provide.

Our approach to clustering is based on the following notion of a cluster: (1) A cluster as a set of contiguous points having similar local structures, defined by the relative spatial distribution of points, which is in contrast with the distributions in its immediate surround. In this paper, we limit the definition of the local distribution to local spatial density of points. (2) A cluster is composed of two kinds of points. **Interior** points are characterized by the property that local neighborhoods centered on them have uniform point density. **Border** points are themselves members of such uniform neighborhoods, but the part of the neighborhood centered on a border point outside the cluster does not contain points from the cluster; it may contain points from zero or more, other, nearby clusters. Thus, the distribution of points in the neighborhood is characterized by a piecewise uniform density. (3) The difference in densities across a border point is referred to as the density contrast at that border point. For a cluster to be perceived, the density contrast across its border points must be greater than its internal density variation.

Overview of Algorithm: There are two major parts to the proposed clustering algorithm. First, we determine the dimensionality of the point distribution in the vicinity of a given point. This determines whether in the vicinity of a point the cluster is confined to a linear subspace of the multidimensional parent space of the data. Second, we develop a uniformity test for whether the spatial distribution in the neighborhood of a given point, is uniform in its (sub)space (sec 3), and use this test to differentiate between interior and border points. We group contiguous interior points to identify cores of the clusters (sec 4). The cluster membership of bor-

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

der points is determined by determining the cluster core they are adjacent to. To this end, we let each interior point identify each border point it finds by the label of its core cluster. The border points then accumulate border votes, one from each interior point whose neighborhood they belong to along with the core cluster label of the voting interior point.Since border points mostly clearly belong to one cluster, the vote label counts decisively to support one cluster to which they are then assigned. These steps are independet of cluster shape, making our algorithm invariant to it. In section 5 we present results of our algorithm on artificial datasets, with outliers, as well as demonstrate its performance on one application area, that of image segmentation by clustering of pixels in a feature space.

2 Background

Clustering Algorithms: Clustering algorithms are extremely diverse in their definition of clusters and approaches to finding them. Recent surveys of clustering algorithms are present in papers by Jain et al. [5] and Xu and Wunsch [9]. Popular clustering algorithms such as those based on the Expectation Maximization framework[1] and K-Means are plagued with problems such as a tendency to converge to suboptimal solutions, sensitivity to outliers and a bias towards hyperellipsoidal shapes [2, 5, 9].

Our approach to clustering is philosophically similar to density based techniques[9]. Density based techniques such as DBSCAN [3] and DENCLUE [4] are based on approximation of the local density. In DBSCAN, the user defines an ϵ-neighborhood and the minimum number of points(*minpts*) that should be present in that neighborhood. Points that satisfy this criterion are labeled interior and others are labeled boundary. Clusters are found as sets of density-connected points [3]. The output of the algorithm is sensitive to the neighborhood size and *minpts*. DENCLUE [4] involves estimating the underlying probability density of the data by superimposing kernels. Clusters are identified by associating points with the nearest mode. This algorithm suffers from the lack of a principled method of determining kernel parameters. While these algorithms explicitly model the densities present in the data to differentiate cluster interior and border, we utilize more general cues that are independent of density and, as discussed earlier, reflect the piecewise uniformity of density in the nighborhoods of the border points vs. the complete uniformity characteristics of interior points.

Multivariate Uniformity Testing: Tests such as the Kolmogorov-Smirnov (KS) test, Cramer von Mises test and Watson's U^2 test are used to test uniformity in one-dimensional data sets [8]. However, a limited amount of work has been done on tests for multivariate uniformity. Liang et al. in [7] present a multivariate uniformity test for $[0, 1]^d$, a d-dimensional unit hypercube, based on asymptotic number theoretic properties, called discrepancies, of uniformly distributed points. Jain et al.[6] propose a test based on minimum spanning tree that relies on resampling to calculate a test statistic. The power of this test was found to degrade with increasing dimensionality. More recently, Petrie and Willemain [8] proposed a test for uniformity based on the method of snakes. A snake is essentially a Hamiltonian path through a set of points. The method represents the edge lengths of the snake as a time series and models this as a low order autoregressive process. This test is known to be weaker than the test based on discrepancies. However, it is less restrictive as it also applies to non-convex neighborhoods. In our work we use the 1-d KS test in higher dimensions by obtaining 1-d distributions of local point patterns. This is done by estimating the principal axes of the (sub)space associated with a neighborhood point distribution, and projecting all the points in the neighborhood on these axes. The empirical 1-d axial distributions are then compared to the theoretically predicted distributions discussed in the next section.

3 Uniformity Criterion

Consider a d-dimensional hyperspherical neighborhood, centered at the origin, and any one of its diametrical chords. Assuming points in this neighborhood are distributed uniformly, we analyze the cumulative distribution function (cdf) of the projections of all points in this neighborhood on the chord. If the x-axis is assumed to be aligned with the (e.g., horizontal) chord, with the origin at one (e.g., the left) end, then the value of the cdf $F_{axial}(x)$ (equations 1-2) at a point x on the chord is equal to the ratio of the volume of the hyperspherical cap covering the chord from position 0 through x_i, to the total volume of the hypersphere.

$$F_{axial}(x) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{n}{2}+1)}{\Gamma(\frac{n+1}{2})} \int_0^{\phi} \sin^n \theta d\theta \quad (1)$$

$$\phi = \cos^{-1} \frac{x}{r}, -r \leq x \leq r \quad (2)$$

Here Γ is the *Gamma* function and r is the radius of the neighborhood. This distribution of projections is a necessary condition for uniformity. This property is true for any diametrical chord of a points neighborhood as long as the neighborhood is contained inside a uniform cluster. For clusters in a lower dimensional (d) subspace of the D-dimensional parent space, this cdf holds only for

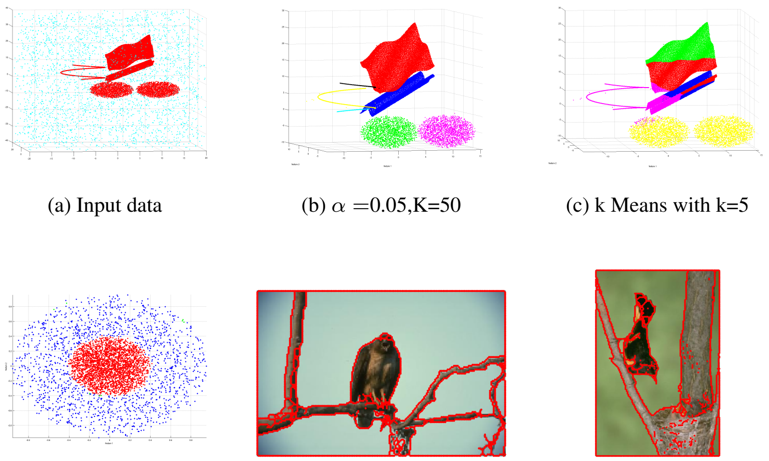
the linear subspace containing the cluster. In particular, it holds true for chords coaligned with the d-principal components that span the cluster subspace within this local neighborhood. Therefore, the test for multidimensional uniformity is as follows:
(1) Determine intrinsic dimensionality (d) of the neighborhood by performing PCA on the local covariance matrix. Scale eigenvalues (λ_i) by dividing them by the largest eigenvalue. Take d to be the number of eigenvalues greater than an input threshold ($\frac{\lambda_i}{\lambda_{max}} \geq t_{eig}$).
(2) Select the top d-eigenvectors corresponding to the d largest eigenvalues as the diametrical chords.
(3) Project the neighborhood points on each of the d-eigenvectors.
(4) Compare the distribution of projections with the known distributions (from equations 1-2) using the 1D Kolmogorov-Smirnov (KS) 1-sample test to verify if they satisfy this property of uniformity at some given significance level α.
(5) The neighborhood is uniform if all d distributions return a positive on the KS test.

4 Algorithm

We specify a point neighborhood by K, the number of nearest neighbors it must contain, making the algorithm independent of scale. Given a data set X=$\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, the algorithm requires as user inputs: the level of significance to be used for the 1-d uniformity test α, K, and t_{eig}, the parameter to determine the dimensionality of a local neighborhood. Our clustering algorithm then proceeds as follows:
1. **Determination of local coordinate system:** For each point $\mathbf{x}_i$, we use its K-nearest neighbors to determine the dimensionality d and the d principal eigenvectors that best span this d-dimensional subspace. The d-eigenvectors are used to determine the uniformity of this local K neighborhood. Empirically, we found K between 50-100 points to be sufficient for a good approximation of the empirical axial projection cdf.
2. **Classification of interior and border points:** Given the K neighborhood of a point we now apply the multidimensional uniformity test (sec 3) on the axial projections of the points in this neighborhood after the chord length (neighborhood diameter) is normalized to unity ($0 \leq x \leq 1$). These tests are conducted at a significance level α. Points which pass the test are labeled interior and points that fail are labeled border.
3. **Agglomeration of interior points into clusters:** We next pool the interior points into cluster cores and assign each core a distinct cluster label. To this end, we associate with each interior point a core neighborhood. The core neighborhood of an interior point is defined as the largest neighborhood that can be grown around an interior point without encountering any border points. An interior point transfers its cluster label to all interior points present in its core neighborhood. Clustering proceeds by including overlapping core neighborhoods into the same cluster, in a way similar to connected component labeling in binary images by propagating component labels across adjacent 1s. The output of this stage is a set of labeled cluster cores.
4. **Assignment of border points to clusters:** The labels of the border points of each cluster are the same as the label of the cluster cores they are adjacent to. Every interior point that finds a specific border point in its K-neighborhood casts a vote for the label of the cluster to which it belongs. Border points are assigned to the cluster with the largest vote.
The output of the algorithm is a (K,α,t_{eig}) clustering of the data.

5 Results

We present results on artificial data and real images from the Berkeley dataset. We generated 1-3 dimensional point patterns and embedded them in a 10-d space. The output of the clustering (projected into 3-d space) for one such point pattern (fig 1(a)) is shown in figure 1(b). Additionally, we tested the algorithm's performance on clusters of upto 15 dimensions. We also tested its sensitivity to inter-cluster separation and overlap (fig 1(d)). The latter case is interesting because both model based methods (e.g. k-Means) and density based methods (DBSCAN) fail under these conditions. This is because a purely density based definition of interior and border points as used by DBSCAN makes it difficult to choose fixed density parameters that resolve clusters of arbitrary density and overlap. Our definition of interior and border points is independent of the exact value of cluster density. Our experiments demonstrate the algorithm's ability to differentiate between clusters of many dimensionalities, shapes and densities in the presence of noise and outliers.
Next, we used our clustering algorithm to segment color images from the Berkeley segmentation data set. We treat each pixel in the image as a point in 5D feature space (x,y,L,a,b). We obtain a segmentation for user defined input of α. The input images and their corresponding segmentations are shown in figures 1(e)-1(f). Our goal here is to test the clustering algorithm while the segmentation quality depends on the choice of features and other details. The clustering appears to qualitatively work as, based on the intensity and color features alone, the algorithm produces segmentations of regions with constant or gradually varying intensity. In all



(a) Input data

(b) α =0.05,K=50

(c) k Means with k=5

(d) α =0.01,K=50

(e) α =0.01,K=50

(f)
α =0.01,K=50

Figure 1. Clustering results for (a) artificial data with noise (light blue),(b) output of our algorithm, (c) output for k-Means with k=5. (d) Output of our algorithm for overlapping clusters of different densities. (f) Color image segmentation using our clustering method. Region boundaries are shown in bold.

our experiments, K was fixed at 50 and t_{eig} at 0.2. The results did not vary to any significant degree if we varied K between 50 and 100 and t_{eig} in the range 0.1 and 0.3. For α, we also found that values between 0.01-0.05 gave nearly the same clustering results.

6 Conclusions

We have presented a novel unsupervised, nonparametric clustering algorithm based on a multivariate uniformity test which we have also described. Our algorithm can identify clusters of varying density, shape, size and intrinsic dimensionality. Our main contribution is the novel uniformity criterion to identify interior and border points, and its use to define a clustering algorithm whose performance is not very sensitive to the input parameters. This obviates the need to explicitly model the underlying density. Furthermore, the clustering is robust to noise and outliers. In future work, we plan to identify more cues for interior and border point identification and extend this framework for identification of clusters with modes (e.g. Gaussian distributed data).

Acknowledgement

The support of the Office of Naval Research under grant N00014-06-1-0101 is gratefully acknowledged.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[3] M. Este, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Intl. Conf. Knowledge Discovery and Data Mining*, 1996.

[4] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Knowledge Discovery and Data Mining*, pages 58–65, 1998.

[5] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *Surveys*, 31(3):264–323, 1999.

[6] A. K. Jain, X. Fan, and T. Ho. Uniformity testing using minimal spanning tree. In *Proc. of IEEE Conference on Pattern Recognition*, 2002.

[7] J.-J. Liang, K.-T. Fang, F. J. Hickernell, and R. Li. Testing multivariate uniformity and its applications. *Mathematics of Computation*, 70(233):337–355, 2001.

[8] A. Petrie and T. R. Willemain. Spanning trees as data analysis tools. In *Proc. of JSM*, 2006.

[9] R. Xu and D. Wunsch. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.