A Fast Clustering Algorithm with Application to Cosmology

Woncheol Jang

May 5, 2004*

Abstract

We present a fast clustering algorithm for density contour clusters (Hartigan , 1975) that is a modified version of the Cuevas, Febrero and Fraiman (2000) algorithm. By Hartigan's definition, clusters are the connected components of a level set $S_c \equiv \{f > c\}$ where f is the probability density function. We use kernel density estimators and orthogonal series estimators to estimate f and modify the Cuevas, Febrero and Fraiman (2000) Algorithm to extract the connected components from level set estimators $\widehat{S}_c \equiv \{\widehat{f} > c\}$. Unlike the original algorithm, our method does not require an extra smoothing parameter and can use the Fast Fourier Transform (FFT) to speed up the calculations. We show the cosmological definition of clusters of galaxies is equivalent to density contour clusters and present an application in cosmology.

Key Words: Density contour cluster; clustering; Fast Fourier Transform.

1 Introduction

Clustering is an important subject in statistics and has recently received a great deal of attention in the field of machine learning under the name

*The author would like to thank Larry Wasserman and Bob Nichol for their helpful comments and suggestions

unsupervised learning (Hastie et al , 2001). The usual tools for clustering are similarities or distances between objects.

In most case, the objectives of clustering are to find the locations and the number of clusters. Although these two problems are separate, it is tempting to solve both of them simultaneously. For the first step in clustering, we shall define clusters precisely from statistical point of view.

From one point of view, a cluster is a mode associated with a *location carrying high probability over a neighborhood* rather than a *local maximum of the density*. To capture this concept, several definitions of clusters have been introduced in statistics, for example, density contour clusters (Hartigan , 1975), modes of given width (Hartigan , 1977) and bumps (Good and Gaskins , 1980).

We adapt Hartigan's definition of clusters : clusters are connected components of level sets $S_c \equiv \{f > c\}$ where f is the probability density function on $\mathbb{R}^d$. Therefore clustering is equivalent to estimating level sets.

Then we face the following two problems immediately :

- How to estimate the level set?
- How to extract the connected components of the estimated level set?

A naive estimator for the level set is the plug-in estimator $\widehat{S}_c \equiv \{\widehat{f} > c\}$ where $\widehat{f}$ is a nonparametric density estimator. For example, kernel density estimators and orthogonal series estimators can be used. The consistency of the plug-in estimator was proved by Cuevas and Fraiman (1997) in terms of a set metric such as the symmetric difference d_μ and the Hausdorff metric d_H:

$$d_\mu \equiv \mu(T \Delta S), \quad d_H(T, S) \equiv \inf\{\epsilon > 0 : T \subset S^\epsilon, T^\epsilon \subset S\},$$

where Δ is symmetric difference, μ is Lebesgue measure and S^ϵ is the union of all open balls with a radius ϵ around points of S.

Baíllo et al (2001) showed that the convergence rates of the plug-in estimator are at most the order of $n^{-1/(d+2)}$.

While the plug-in estimator is conceptually simple, it is not easy to extract the connected components of the estimated level set in practice. Instead of using the plug-in estimator, Cuevas, Febrero and Fraiman (2000) proposed a different method which we will refer to as the CFF algorithm.

The key idea of the CFF algorithm is first to find the subset of data belonging to the level set and then find clusters by agglomerating the data

points. Unlike other clustering algorithms such as mixture models and hierarchical single linkage clustering, the CFF algorithm performs well even with a noisy background (Wong and Moore , 2002).

2 Clustering Algorithm

The CFF algorithm consists of two key steps.

- Find the data points Y_i's which belong to estimated level set $\widehat{S}_c$.
- Join every given pair of Y_i's with a path consisting of a finite number of edges with length smaller than $2\epsilon_n$.

In other words, the CFF algorithm provides a method to approximate $\widehat{S}_c$ by

$$\widetilde{S}_c^1 = \bigcup_{i=1}^{k_n} B(Y_i, \epsilon_n)$$

where $B(Y_i, \epsilon_n)$ is a closed ball centered at Y_i with radius ϵ_n and k_n is the number of the observations which belong to $\widehat{S}_c$. Note that k_n is random.

While the CFF algorithm is simple and outperforms the other clustering algorithms for noisy background cases, it is also computationally expensive. Even for the first step, we need to evaluate the density estimates at every data point. Especially in high dimension, the task could be daunting even with today's high computing power. Furthermore, the CFF algorithm require an extra smoothing parameter ϵ_n in addition to the smoothing parameter of the density estimator such as the bandwidth in the kernel density estimator.

Gray and Moore (2003) addressed the issue in the first step. They evaluated density estimates by cutting off the search early without computing exact densities.

The second step is equivalent to finding Minimum Spanning Tree and (Wong and Moore , 2002) proposed an alternative implementation based on the GeoMS2 algorithm (Narasimhan et al , 2000). Though Wong and Moore showed the improvement of the CFF algorithm, their algorithm still requires ϵ_n as an input.

To avoid choosing another smoothing parameter and save computing time, we propose a modified version of the CFF algorithm. The key idea

is to replace data points with grid points. In other words, we approximate $\widehat{S}_c$ by

$$\widetilde{S}_c^2 \equiv \bigcup_{i=1}^{k'_m} B(t_i, \epsilon'_m)$$

where t_i's are equally spaced grid points which belong to $\widehat{S}_c$, k'_m is the total number of the grid points belonging to $\widehat{S}_c$ and ϵ'_m is the grid size.

Having used the size of grid as the radius of the ball, one can avoid an extra smoothing parameter. Moreover, one can use the Fast Fourier Transform (FFT) to evaluate density estimates at grid points to speed up the calculations. Since grid points are equally spaced, one can also use information of coordinate systems of grid points to calculate the distance of any pairs. We us the following steps as described in (Cuevas, Febrero and Fraiman , 2000).

Let T be the number of connected components and set the initial value of T as 0.

Step 1 Evaluate $\widehat{f}$ at every gird point using the FFT to find the set $\{t_i : t_i \in \widehat{S}_c\}$.

Step 2 Start with any grid point of the set and call it t_1. Compute the distance r_1 between t_1 and the nearest grid point, (say t_2).

Step 3 If $r_1 > 2\epsilon'_m$, the ball $B(t_1, \epsilon'_m)$ is a connected component of $\widehat{S}$. Put $T = T + 1$ and repeat step 1 with any grid point in $\widehat{S}_c$ except t_1.

Step 4 If $r_1 \leq 2\epsilon'_m$, find another grid point (denote t_3) closest to the set $\{t_1, t_2\}$ and compute

$$r_2 = \min\{\|t_3 - t_1\|, \|t_3 - t_2\|\}$$

Step 5 If $r_2 > 2\epsilon'_m$, put $T = T + 1$ and repeat step 1 with any grid point in $\widehat{S}_c$ except t_1 and t_2.

Step 6 If $r_2 \leq 2\epsilon'_m$, compute, by recurrence,

$$r_K = \min\{\|t_{K+1} - t_i\|, i = 1, \ldots, K\},$$

where t_{K+1} is the grid point closest to the set $\{t_1, \ldots, t_K\}$.

Continue in this way until we get , for the first time, $r_K > 2\epsilon'_m$. Then put $T = T + 1$ and return to step 1.

Step 7 Repeat Step 2 - 6 until every grid point is considered, then the total number of clusters, connected components of $\widehat{S}_c$ is T.

3 Application in Cosmology

In cosmology, clusters of galaxies play an important role in tracing the large-scale of the universe. However, the availability of high quality of astronomical sky survey data for such studies was limited until recently.

The power of modern technology is opening a new era of massive astronomical data that is beyond the capabilities of traditional methods for galaxy clustering. For example, Figure 1 show the Mock 2dF catalogue. The Mock 2dF catalogue has been built to develop faster algorithms to deal with the very large numbers of galaxies involved and the development of new statistics (Cole et al , 1998). The catalogue contains 202,882 galaxies and each galaxies has 4 attributes : right ascension (RA) , declination (DEC), redshift and apparent magnitude. RA and DEC are the longitude and latitude with respect to the Earth and the redshift can be considered as a function of time.

Cosmological theory assume that clusters of galaxies are virialized objects which means that they have come into dynamical equilibrium. To reach dynamical equilibrium, a cluster must satisfy the following geometric condition,

$$C = \Big\{ x \Big| \rho(x|t) > \delta \Big\},$$

where δ is given from cosmological theory and $\rho(x|t)$ is the mass density function at time t.

Estimating ρ is equivalent to estimating a probability density (Jang , 2003). Therefore, from cosmological point of view, clusters of galaxies is the same as density contour clusters.

Our goal is to find the spatial distribution of the locations of clusters as a function of time. In other words, we want to estimate the joint distribution of RA and DEC given redshift. To do so, the data were divided into 10 slices by equally spaced redshift and then, a bivariate kernel density estimator was fitted. Figure 2 (a) shows a slice of the 2dF data with $0.10 < z < 0.125$ and the contour plot by the density estimates is given in Figure 2 (b).

To keep the original scale of the data, a spherically symmetric kernel was used, which means the bandwidth matrix is a constant times the identity matrix. The bandwidth was selected by cross-validation and density estimates

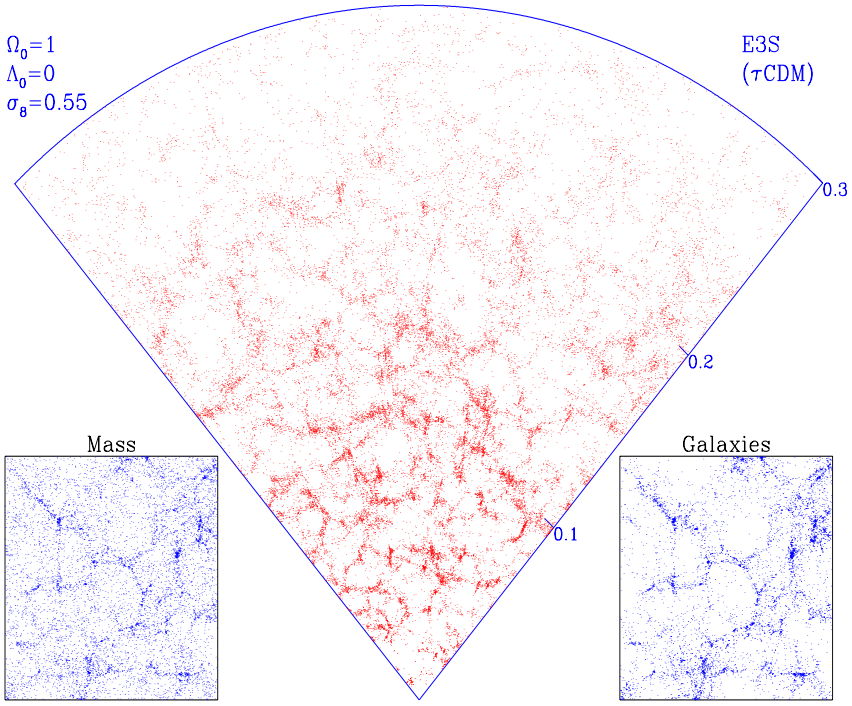


Figure 1: Mock 2dF catalogue

at the grid points were evaluated by the FFT. A Newton-Raphson type optimizer was used to find the optimal bandwidth and the plug-in method was used to provide the starting point in the Newton-Raphson method. The FFT and the plug-in method were implemented by the R library "KernSmooth" developed by Matt Wand.

After finding the sub set of grid points belonging to the level set, the modified CFF algorithm was applied for galaxy clustering. Figure 2 (c) shows the grids point which belongs to the estimated level set $\{\widehat{f} > \delta\}$. In Figure 2 (d), each color represents a different cluster and 1,945 clusters were found out of 33,157 galaxies.

4 Nonparametric Confidence Sets

To address uncertainty of the level set estimators or clustering results, one consider constructing the confidence sets for clusters. While there is a substantial literature on making confidence statements about a curve f in the context of nonparametric regression and nonparametric density estimation, most of them produce confidence bands for f. Therefore, it is not easy to construct confidence statements about features of f such as density contour clusters from the band.

Beran and Dümbgen (1998) developed a method for constructing confidence sets for nonparametric regression which can be used to extract confidence sets for features of f. The confidence set C_n is asymptotically uniform over certain functional classes. Thus,

$$\liminf_{n\to\infty} \inf_{f\in\mathcal{F}} P(f \in C_n) \geq 1-\alpha. \tag{1}$$

As a result, a confidence set for a functional $T(f)$ is

$$\left(\inf_{f\in\mathcal{C}_n} T(f),\ \sup_{f\in\mathcal{C}_n} T(f) \right).$$

These confidence sets are uniform as in (1), simultaneously over all functionals.

The theory in Beran and Dümbgen (1998) doesn't not carry over directly due to some technical reasons. (Jang et al , 2004) provides a method to construct uniform confidence sets for densities and density contour clusters.

5 Conclusion

The explosion of data in scientific problems provides a better opportunity where nonparametric methods can be applied for solving the problems. Our algorithm shows the improvement of the original CFF algorithm in terms of computation expense with the FFT. We also address the issue of the extra smoothing parameter ϵ_n by using the grid space as the size of the balls.

Constructing confidence sets for clusters can be used to address the uncertainty of the clustering results. While the theory has been developed, it is computationally challenging to extract the confidence sets for clusters from the confidence sets for densities.

From practical point of view, it is desirable to develop a stand alone R library for our clustering method. Another possible improvement is to combine our method with Gray and Moore's method which can be used to speed up the density estimation part in the first step.

References

Baíllo, A., Cuesta-Albertos, J. and Cuevas, A. (2001). Convergence rates in nonparametric estimation of level sets. *Statistics and Probability Letters*, **53**, 27–35.

Beran, R. (2000). REACT Scatterplot Smoothers: Superefficiency through Basis Economy. *Journal of American Statistical Association*, **63**, 155–171.

Beran R. and Dümbgen, L. (1998). Modulation of Estimators and Confidence Sets. *Annals of Statistics*, **26**, 155–171.

Cole,S., Hatton,S., Weinberg, D. and Frenk, C. (1998). Mock 2dF and SDSS Galaxy Redshift surveys. *Monthly Notices of the Royal Astronomical Society*, **300**, 945–966.

Cuevas, A. and Fraiman, R. (1997). A Plugin approach to support estimation. *Annals of Statistics*, **25**, 2300–2312.

Cuevas, A., Febrero, M. and Fraiman, R. (2000). Estimating the number of clusters. *The Canadian Journal of Statistics*, **28**, 367–382.

Good, I.J. and Gaskins R.A. (1980). Density Estimation and Bump Hunting by the Penalized Likelihood Method Exemplified by Scattering and Meteorite Data. *Journal of American Statistical Association*, **75**, 42–73.

Gray, A.G. and Moore, A. W. (2003). Very Fast Multivariate Kernel Density Estimation via Computational Geometry. Unpublished manuscript.

Hartigan, J.A. (1975). *Clustering Algorithms*. Wiley, New York.

Hartigan, J.A. (1977). Distribution Problems in Clustering. In *Classification and Clustering*. Academic Press, New York, 45–72.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, New York.

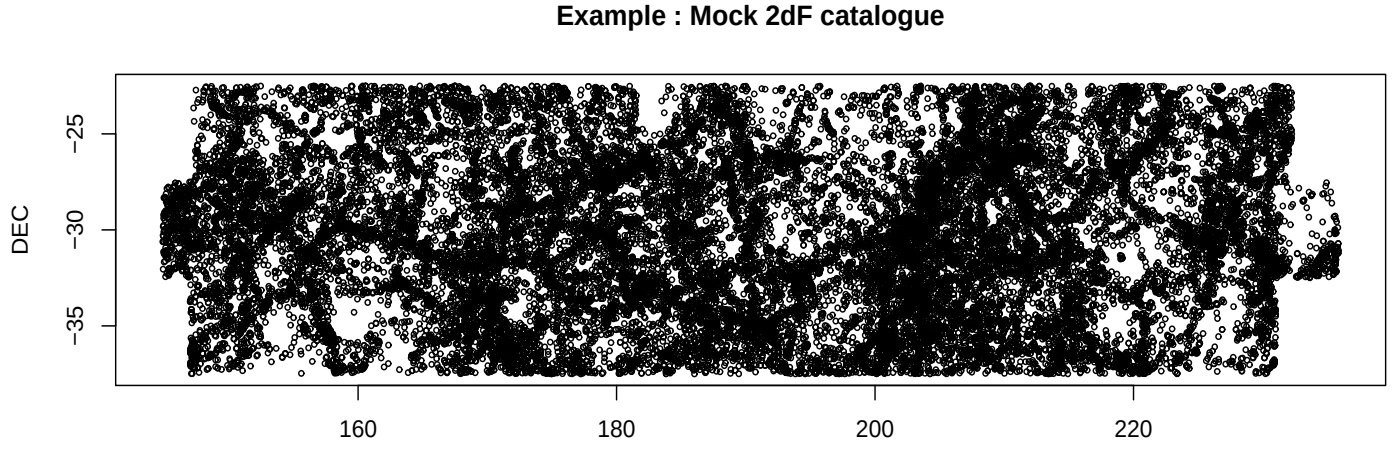
Jang, W. (2003). Nonparametric Density Estimation and Clustering with Application to Cosmology. unpuplished Ph.D. dissertation, Department of Statistics, Carnegie Mellon University.

Jang, W., Genovese, C. and Wasserman, L. (2004). Nonparametric Confidence Sets for Densities and Clusters. Technical Report **795**, Carnegie Mellon University.

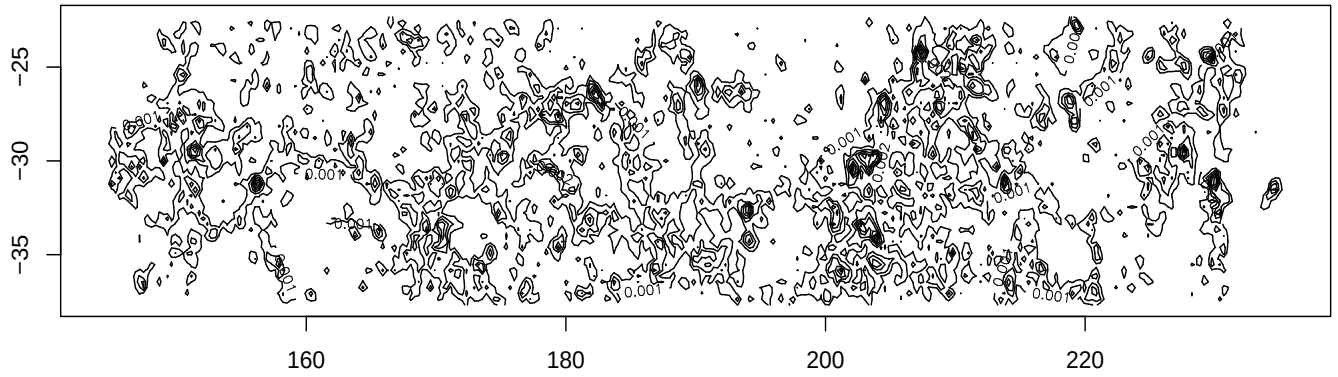
Martínez, V. and Saar, E. (2002). *Statistics of the Galaxy Distribution*. Chapman and Hall, London.

Narasimhan, G., Zhu, J., and Zachariasen, M. (2000). Experiments with computing geometric minimum spanning trees. In *Proceedings of ALENEX'00, Lecture Notes in Computer Science*. Springer-Verlag.

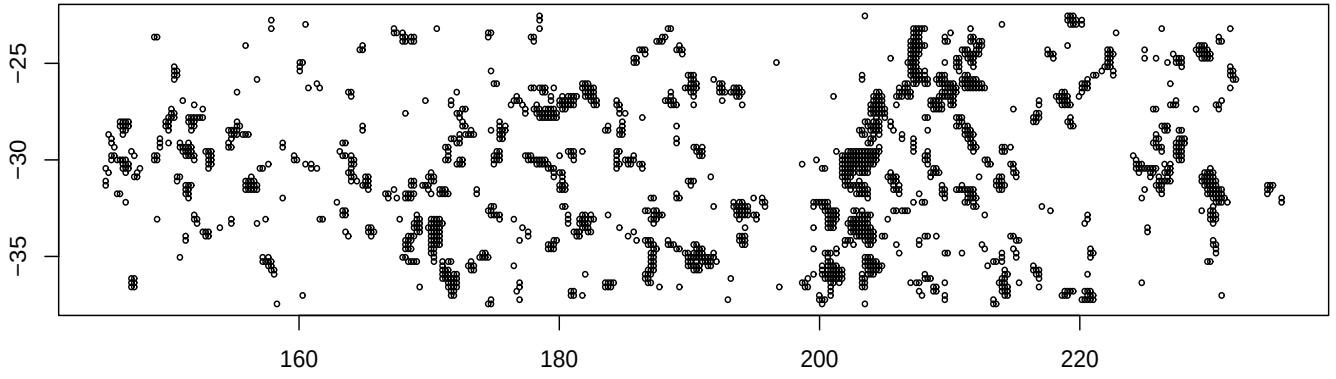
Wong, W-K. and Moore, A.W. (2002). Efficient algorithms for nonparametric clustering with clutter. In *Proceeding of the Interface 2002 conference*.



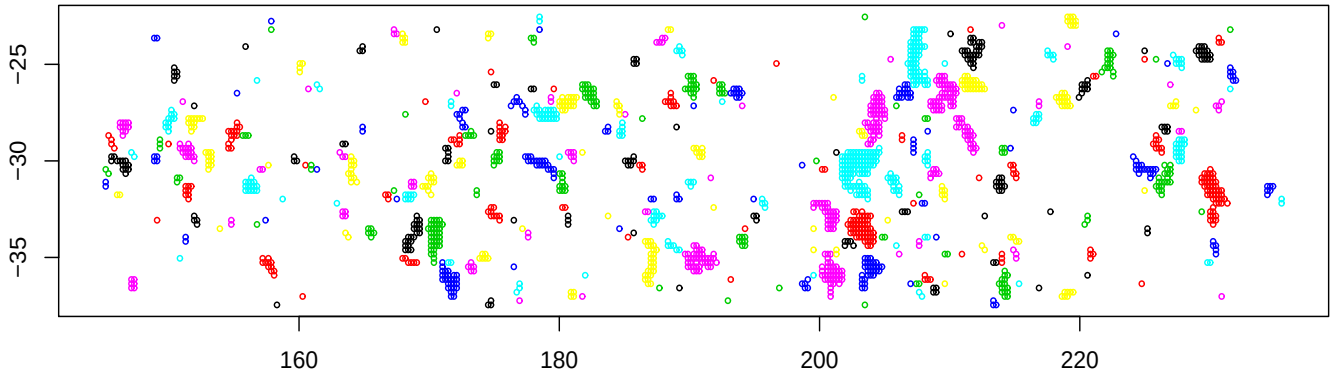
(a) Mock 2dF catalogue with 0.1 < z < 0.125



(b) contour plot by kernel density estimation



(c) Grid points belong to level sets



(d) Clustering with modified Cuevas algorithm − Each color presents a different level set

Figure 2: Mock 2dF catalogue with $0.10 < \text{redshift} < 0.125$