CGCA-py-astro-stat study group – 05/08

Chapter 6 – searching for structure in point data

- Exploratory data analysis
- Given N points in D-dimensional space there are three classes of problem that we want to address: 1) density estimation 2) cluster finding 3) statistical description of observed structure
- Density estimation – to infer pdf from a sample of data (also known as data smoothing).
  *) Given a pdf estimated from point data, can generate simulated distributions and compare with observations. If can identify regions of low probability within the pdf, we can find anomalous or unusual sources.
  *) If we have subsamples, can separate our data, estimate pdf for each subsample and use resulting set of pdfs to classify new points
- Clustering – finding concentrations of multivariate points (also known as overdensities). Clustering can refer to separate objects (gravitationally bound clusters of galaxies) or loose groups of sources with common properties (quasars based on their colour property).
- Statistical description – clusters can have specific meaning (such as hot stars or quasars) but on the large-scale clustering of galaxies, clusters carry info only in a statistical sense. E.g. can test cosmological models of lareg scale structure by comparing clustering statistics in observed and simulated data (correlation functions).

6.1 Nonparametric density estimation

- Chapter 3 – estimating the underlying density of data using a parametric model, chapter 4/5 – estimating parameters from frequentist and Bayesian perspectives. Now look at how to estimate density nonparametrically, i.e. without specifying a specific model.
- Nonparametric methods capture every aspect of density's shape; data rarely follows simple distribution
- Downsides – convenience, computational simplicity, easy interpretability

6.1.1 Kernel density estimation

- Modelling underlying distribution is the model of **kernel density estimation** (KDE)
- Recall a problem – in a standard histogram the location of bins can make a big difference; not clear in advance how to pick the bins (see fig 6.1 in top 2 panels – bimodal vs extended flat distribution).
- Each point in histogram contributes one unit to height at the position of its bin. Allow each point to have its own bin (rather than regular grid) and to

overlap. Meaning: each point is replaced by box of unit height and predefined width (see fig 6.1 middle left)

- Data drives bin positioning
- Example of KDE is this above histogram – the kernel is a **top-hat distribution** centered on each individual point
- Slight problem is that the rectangular kernel does not give a smooth distribution and give suspicious spikes, hence why **Gaussian** kernel is often used (middle right and bottom of fig. 6.1 are Gaussian kernel).
- It matters what kernel width you use (see bottom of fig 6.1); too narrow leads to noise, too wide leads to smoothing
- Give eqn and Gaussian/top hat/exponential kernels and draw shapes
- Bandwidth or h defines the width of the kernel. This is more important to get right than actual kernel. Must use **cross-validation**
- Look at likelihood cross-validation or the mean integrated square error
- Optimal kernel function in terms of variance is eqn 6.9
- Fig 6.3 shows example of KDE applied to a 2D data set.

6.1.2 KDE with measurement errors

- All data have some error attached
- Need to deconvolve the KDE (see steps at pg 256)

6.2 Nearest Neighbour Density Estimation

- For each point we can find the distance to the Kth nearest neighbor d.
- Point density at an arbitrary position x is eq 6.13
- This is so simple because we assume the underlying density field is locally constant.
- Technique improves when consider distances to all nearest neighbours rather than just the kth nearest neighbor.
- When looking for local overdensities in sparse data should look at all neighbours rather than just the kth.
- Fig 6.4 compares KDE and k nearest neighbours. For small K fine structure in the galaxy distribution is preserved, but as k increases the density distribution becomes smoother.
- Fig 6.5 shows that for a small number of points both KDE and nearest neighbours produce noisy distributions compared to Bayesian blocks. But if in crease the number of points then both do well

6.3 Parametric Density Estimation

- KDE esimates the density of a set of points by affixing a kernel to each point.
- Can use fewer kernels and fit for the kernel locations as well as widths – mixture model
- Mixture model – density estimation model like KDE but also a clustering algorithm

6.3.1 Gaussian mixture model

- GMM models the underlying density (pdf) of points as a sum of gaussians (see eqn 6.17)
- AIC – aitake information criterion BIC- Bayesian information criterion
- See figure 6.6 for example of 2d mixture model of gaussians and 4d.
- Misconception of GMM: information criterion prefer an N component peak does not necessarily mean that there are N components. If clusters are not near Gaussian or there's a strong background then the mixture will not correspond to number of clusters.
- Shown with great wall (fig 6.7). Not a one to one mapping between gaussians and positions of clusters. Better as a density estimator as opposed to cluster identification.
- When samples are small (fig 6.8) GMM solution found using BIC isn't quite right. It improves when add more points, re-evaluate the BIC and redo.
- BIC is a good tool to find how many statistically significant clusters are supported by data
- With a sufficiently large number of components, mixture models approach flexibility of nonparametric estimation methods.
- Determining the number of components is relatively poor in astronomy. Rare to find distinct, isolated clusters; almost all are continuous.
- For clustering studies, it is useful to fit a mixture model with many components to divide components into 'clusters' and 'background'

6.3.2 Cloning data in D>1 dimensions

- Figure 6.10 shows 1000 data points, then the GMM and extending the number of points based on the model.

6.3.3 GMM with errors – extreme deconvolution

- Pdf in eqn 6.18
- Aim of xd is to find parameters in eqn
- Given errors you can still find the true distribution (look at figures 6.11)
- Xd can treat cases of missing data

6.4 – finding clusters in data

- Concentration of points

6.4.2 – clustering by sum of squares minimization: K-means

- K means seeks a partitioning of the points in to K disjoint subsets
- Minimizing the sum of the square errors
- Eqn 6.28
- In practice k means is run multiple times with different starting values for the centroids
- Fig 6.13 assume 4 clusters we find background pulls the centroid of two of the clusters away from peak. Contrast to earlier GMM.

6.4.3 – max radius minimization

- Minimize the maximum radius of a cluster
- Gonzalez algorithm starts with no clusters and progressively add one cluster at a time

6.4.4 clustering by nonparametric density estimation – mean shift

- To find arbitrary shaped clusters should define clusters in terms of modes or peaks of the nonparametric density estimate, associate each data point with its closest peak
- mean shift algorithm is a technique to find local modes in a KDE
- eqn 6.30 / fig 6.14

6.4.5 – clustering procedurally – hierarchal clustering

- procedural method is a method not been formally related to some function of the underlying density
- this relaxes the need to specify the number of clusters by finding all clusters at all scales
- start my dividing data in N clusters one for each point. Then join two clusters resulting in N-1 clusters. Repeat until Nth partition contains one cluster. If two points are in the same cluster at level m and remain together at all subsequent levels this is known as hierarchal clustering and can see it in a tree diagram (fig 6.15)
- top down procedure – progressively divide the data, bottom up where we merge the nearest pairs of clusters
- can use a distance minimum to cluster results in a hierarchal way known as minimum spanning tree (see fig. 6.15)

6.5 – correlation functions

- how well does a distribution of points differ from a random distribution
- metrics for testing models of structure formation and evolution directly against data
- eqn 6.38 shows correlation function
- 2 point correlation function describes the excess probability of finding a pair of points as a function of separation compared to random
- positive – correlated, negative – anticorrelated, zero – random
- 2 point relates directly to power spectrum through FT
- see fig 6.16 for examples of 2,3,4 point correlation functions

6.5.1 - computing nth correlation function

- 2 point correlation function estimated y calculating the excess or deficit of pair of points with a distance r and r+dr compared to random distribution
- computational cost of estimating the correlation function is dominated by size of random data set

6.6 which density estimation and clustering algorithms should I use?

- See table 6.1