DDM 2017 - Problem set 3

The first four problems here are focused on the use of kernel density estimation and the choice of optimal band-widths. You will need to do the same task (choosing bandwidth) multiple times, so it is useful to write some code to do this.

To help you get started, the notebook `Histograms and kernel estimators for git-.ipynb` contains example code that should get you up and running fairly quickly (if you understand the code of course!), but also see the lecture notes. You can find the notebook in the problem set folder on the github site.

1. Determine an optimal bandwidth

In this problem set you will use a sample of galactic black hole masses from the literature. You will use Kernel Density Estimation to determine the distribution of their masses and for that you need to determine the best band-width to use.

The file to use is available in the git repository as Datasets/joint-bh-mass-table.csv. A csv file can be easily read in using e.g.

```
from astropy.table import Table
t = Table().read(fname)
```

but if you prefer other approaches, feel free to do so.

a) Select the best band-width for kernel estimation by linear search. Try band-widths between 1 and 7 solar masses. What is the best band-width? Is this a useful estimate?

b) Use cross-validation to find the optimal bandwidth. I recommend using the KFold routine in `sklearn.model_selection`.

2. The likelihood of gravitational wave neutron stars

The file `Datasets/pulsar_masses.vot` contains masses of pulsars from Özel & Freire 2016, Annual Reviews of Astronomy and Astrophysics, taken from the associated web page.

On October 15, 2017, the LIGO/Virgo consortium and host of ground- and space-based observatories announced the discovery of an optical counter-part to the gravitational wave source GW170817. The two neutron stars that are thought have merged to produce the gravitational wave signal, have estimated masses of $M_1 = 1.81\,\mathrm{M}_\odot$ and $M_2 = 1.11\,\mathrm{M}_\odot$.

a) Use the pulsar mass catalogue to estimate the likelihood of finding a neutron star with $M > 1.8\,\mathrm{M}_\odot$.

b) Actually the mass estimates are ranges, and the masses were $M_1 \in [1.36,2.26]$ and $M_2 \in [0.86,1.36]$. What are the likelihoods of those mass ranges and the likelihood of the binary?

c) Simulate the next 5 detections of merging pulsars with LIGO+Virgo. What is your prediction for the average mass neutron star they would detect? [hint: to draw N from a `KernelDensity` object, you can use the `.sample(N)` function]

d) There are many assumptions made that I did not spell out - make a list of some that you worry about.

3. How many peaks?

The file `Datasets/mysterious-peaks.pkl` contains data drawn from a distribution with multiple peaks. How many peaks are there?

4 - The great wall of the SDSS

The Sloan Digital Sky Survey has found a large structure in the distribution of galaxies generally known as the SDSS Great Wall. Here you will use kernel density estimates to estimate the density field on the basis of the galaxy positions.

To get the data you will need `astroML`:

```
from astroML.datasets import fetch_great_wall
X = fetch_great_wall()
```

It is also useful to use the following to create a grid to evaluate the data on:

```
# Create  the grid on which to evaluate the results
Nx = 50
Ny = 125
xmin, xmax = (-375, -175)
ymin, ymax = (-300, 200)

xgrid = np.linspace(xmin, xmax, Nx)
ygrid = np.linspace(ymin, ymax, Ny)
mesh = np.meshgrid(xgrid, ygrid)

tmp = map(np.ravel, mesh)
Xgrid = np.vstack(tmp).T
```

where Xgrid can be used as input to score_samples to get a predicted log likelihood which can then be converted to a 2D image using:

```
# Evaluate the KDE on the grid
log_dens = kde.score_samples(Xgrid)
dens1 = X.shape[0] * np.exp(log_dens).reshape((Ny, Nx))
```

a) Compare the 'gaussian', 'tophat', 'exponential' and 'epanechnikov' kernels for the density inference - use a fixed smoothing parameter (for instance 5) - how does the appearance differ between these different kernels.

b) Use 10-fold cross-validation and determine the best smoothing parameter for the great wall data using the kernel you preferred from a).

c) Would this band-width determined from cross-validation always be the right choice? Why? Why not?

5. Galaxy classification using Bayesian approaches.

This problem is implemented as a IPython notebook. It goes through the classification problem discussed in the lecture and asks you to classify galaxies & calculate mis-classification rates. The problem is found in the `Classification of galaxies.ipynb` file in the `Problem set 3` directory in the git repository.