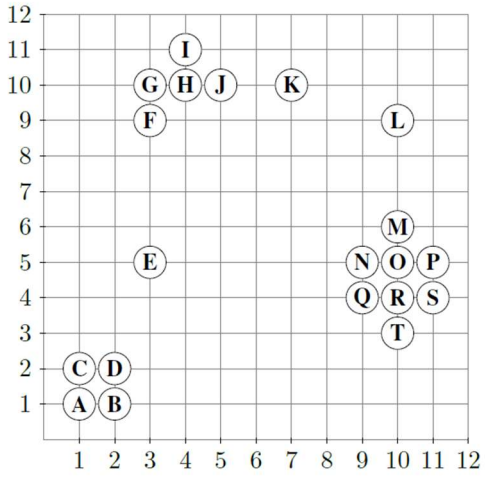
University of Southern Denmark
IMADA
Ricardo Campello

DM583: Data Mining

Exercise 5: Density Estimates and Probabilistic Clustering

Exercise 5-1 Kernel Density Estimates

a) Consider the following dataset in 2 dimensions:



Estimate the density at the locations coinciding with data points H, K, E, as well as at location (3,7), using the **Discrete Kernel** (so-called square/cubic/hypercubic kernel) with window width equal to: (i) $h = 2$; (ii) $h = 4$

Proposed Solution: Recall that, according to the discrete kernel, density estimate at a query location in $\Re^d$ with respect to a data sample of size n consists of first counting the number k of data points within a d-dimensional hypercube (with fixed edge width h) centered at the query, then dividing the corresponding fraction of the dataset, k/n, by the volume of the hypercube. Since $d = 2$ in this exercise, the volume of the hypercube is h^d, i.e., h^2. Thus, density is $(k/n)/h^2$. There are $n = 20$ data points.

We start with $h = 2$. For location H, there are $k = 5$ points within the hypercube with edge size $h = 2$ centered at that location (H, F, G, J, I), so density = $(k/n)/h^2 = (5/20)/2^2 = 1/16$. For location K, there is $k = 1$ point (namely, K itself) within the hypercube with edge size $h = 2$ centered at that location, so density = $(k/n)/h^2 = (1/20)/2^2 = 1/80$. For location E, there is $k = 1$ point (namely, E itself) within the hypercube with edge size $h = 2$ centered at that location, so density = $(k/n)/h^2 = (1/20)/2^2 = 1/80$. For location (3,7), there are no points ($k = 0$) within the hypercube with edge size $h = 2$ centered at that location, so density = $(k/n)/h^2 = (0/20)/2^2 = 0$.

Now for $h = 4$. For location H, there are $k = 5$ points within the hypercube with edge size $h = 4$ centered at that location (H, F, G, J, I), so density = $(k/n)/h^2 = (5/20)/4^2 = 1/64$. For location K, there are $k = 2$ points within the hypercube with edge size $h = 4$ centered at that location (K, J), so density = $(k/n)/h^2 = (2/20)/4^2 = 1/160$. For location E, there is $k = 1$ point (namely, E itself) within the hypercube with edge size $h = 4$ centered at that location, so density = $(k/n)/h^2 = (1/20)/4^2 = 1/320$. For location (3,7), there are $k = 2$ points within the hypercube with edge size $h = 4$ centered at that location (E, F), so density = $(k/n)/h^2 = (2/20)/4^2 = 1/160$.

b) Now, consider the unidimensional dataset corresponding only to the horizontal coordinate (x-axis) in the figure in item (a). Estimate the density at location 6 using this 1D dataset and a **Gaussian Kernel** with width $h = 1$

Proposed Solution: Recall the kernel density estimate equations for the unidimensional case:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

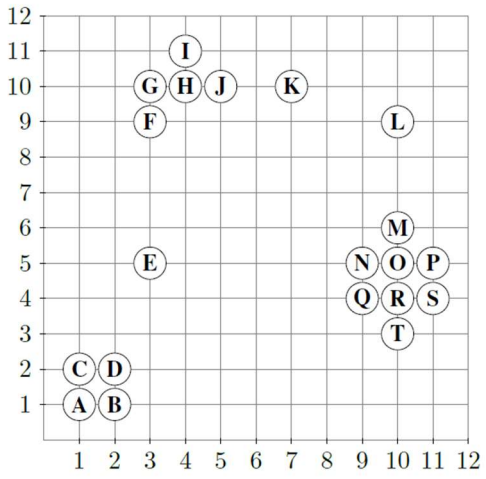
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x - x_i)^2}{2h^2}\right\}$$

We can then use these equations with $n = 20$, $h = 1$, $x = 6$, and x_i ($i = 1, \ldots, 20$) given by the locations of points A, B, C, …, R, S, T, respectively, i.e., 1, 2, 1, 2, 3, 3, 3, 4, 4, 5, 7, 10, 10, 9, 10, 11, 9, 10, 11, 10. Using R as an advanced calculator, we can compute the desired density estimate as follows:

```
> n = 20
> h = 1
> x = 6
> xi = c(1, 2, 1, 2, 3, 3, 3, 4, 4, 5, 7, 10, 10, 9, 10, 11, 9, 10, 11, 10)
> (1/(n*h))*sum((1/sqrt(2*pi))*exp(-(x-xi)^2/(2*h^2)))
[1] 0.03075127
```

Exercise 5-2 KNN Density Estimates

a) Consider the following dataset in 2 dimensions:



Estimate the density at the locations coinciding with data points H, K and E, as well as at location (3,7), using the **K-Nearest Neighbour (KNN) Estimator** with *Euclidean Distance* and $k = 2$. **Note 1** (density at data points): when we are estimating density at a location that coincides with one or more data points, these should be counted as part of the KNN neighbourhood of the location of interest. **Note 2** (ties): when there is more than one data point tied as kth nearest neighbour at the same distance from the query location, these should all be counted as part of the KNN neighbourhood of the location of interest, which in practice corresponds to adjusting the value of k used in the computation of the density estimate for that particular location.

Proposed Solution: Recall that in the KNN density estimator, we divide the fraction of points within the KNN neighborhood of a query location in $\Re^d$ by the volume of the neighborhood, V_{KNN}. In other words, given a fixed value of k and assuming the kth nearest neighbor is unique (no ties), then density is $(k/n) / V_{KNN}$, where n is the sample (dataset) size, which is $n = 20$ in this exercise. For Euclidean distance, the KNN neighborhood around

the query (i.e., the region within a distance from the query that is at most as large as the query's distance to its kth nearest neighbor) is a d-dimensional hypersphere. In this exercise, $d = 2$, so V_{KNN} is the volume of a circle (2D hypersphere), given by $\pi \cdot r^2$, where r is the radius. In our case, the radius is the Euclidean distance from the query location to its kth nearest neighbor, d_{KNN}, so $V_{KNN} = \pi \cdot (d_{KNN})^2$.

We start with $k = 2$:

- Query Location **E**: 1st nearest neighbor (NN) is data point E itself, 2nd NN is clearly D (visually from the figure), which is at a distance $(3^2 + 1^2)^{1/2} = (10)^{1/2}$. So, $d_{KNN} = (10)^{1/2}$. $V_{KNN} = \pi \cdot (d_{KNN})^2 = 10\pi$. Density = $(k/n) / V_{KNN} = (2/20)/10\pi = 1/100\pi \cong 1/314$.
- Query Location **K**: 1st nearest neighbor (NN) is data point K itself, 2nd NN is clearly J, which is at a distance 2. So, $d_{KNN} = 2$. $V_{KNN} = \pi \cdot (2)^2 = 4\pi$. Density = $(k/n) / V_{KNN} = (2/20)/4\pi = 1/40\pi \cong 1/126$.
- Query Location **H**: 1st nearest neighbor (NN) is data point H itself, but in this case, there are 3 data points tied as 2nd NN, namely, G, I, J (visually from the figure), all at a distance 1. So, $d_{KNN} = 1$ and $V_{KNN} = \pi \cdot (1)^2 = \pi$. Notice that, in this case, because of the tie, we actually have an augmented KNN neighborhood size of $k_+ = 4$, rather than $k = 2$. So, density = $(k_+/n) / V_{KNN} = (4/20)/\pi = 1/5\pi \cong 1/16$.
- Query Location **(3,7)**: the two closest neighbors, E and F, are tied at a distance 2. So, $d_{KNN} = 2$ and $V_{KNN} = \pi \cdot (2)^2 = 4\pi$. Despite the tie, since there aren't any other data points within the KNN neighborhood, we don't need to adjust the neighborhood cardinality. So, density = $(k/n) / V_{KNN} = (2/20)/4\pi = 1/40\pi \cong 1/126$. **Note**: *this is the same density estimate previously obtained for location K. Visually comparing both locations, it becomes clear that this estimator is not sensitive to the distances of other points within the KNN neighborhood, except for the kth NN. If this is a desirable property, then the Gaussian Kernel estimator is an alternative that meets this requirement.*

b) Repeat item (a), but now with $k = 3$.

Proposed Solution:

Now with $k = 3$:

- Query Location **E**: 1st nearest neighbor (NN) is data point E itself, 2nd NN is D, 3rd NN is C (visually from the figure), which is at a distance $(3^2 + 2^2)^{1/2} = (13)^{1/2}$. So, $d_{KNN} = (13)^{1/2}$. $V_{KNN} = \pi \cdot (d_{KNN})^2 = 13\pi$. Density = $(k/n) / V_{KNN} = (3/20)/13\pi \cong 1/272$.
- Query Location **K**: 1st nearest neighbor (NN) is data point K itself, 2nd NN is J, 3rd NN is clearly H, which is at a distance 3. So, $d_{KNN} = 3$. $V_{KNN} = \pi \cdot (3)^2 = 9\pi$. Density = $(k/n) / V_{KNN} = (3/20)/9\pi \cong 1/188$.
- Query Location **H**: 1st nearest neighbor (NN) is data point H itself, and there are 3 data points tied as 2nd and 3rd NN, namely, G, I, J, all at a distance 1. So, $d_{KNN} = 1$ and $V_{KNN} = \pi \cdot (1)^2 = \pi$. Notice that, in this case, because of the tie, we actually have an augmented KNN neighborhood size of $k_+ = 4$, rather than $k = 3$. So, density = $(k_+/n) / V_{KNN} = (4/20)/\pi = 1/5\pi \cong 1/16$.
- Query Location **(3,7)**: the two closest neighbors are data points E and F, whereas the 3rd KNN is clearly G, which is at a distance 3. So, $d_{KNN} = 3$ and $V_{KNN} = \pi \cdot (3)^2 = 9\pi$. Density = $(k/n) / V_{KNN} = (3/20)/9\pi \cong 1/188$. **Note**: *this is the same density estimate previously obtained for location K. Visually comparing both locations, it becomes clear that this estimator is not sensitive to the distances of other points within the KNN neighborhood, except for the kth NN. If this is a desirable property, then the Gaussian Kernel estimator is an alternative that meets this requirement.*

Exercise 5-3 EM-GMM Clustering

a) Suppose you are running EM-GMM to perform probabilistic clustering of a 2-dimensional dataset into $k = 2$ clusters and, during execution of the algorithm, at a given iteration, the estimated multivariate normal clusters, C_1 and C_2, are as follows:

$$Cov(C_1) = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix}$$

$$Cov(C_2) = \begin{bmatrix} 0.8 & 0.27 \\ 0.27 & 0.11 \end{bmatrix}$$

$$\mathbf{v}_{C_1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{v}_{C_2} = \begin{bmatrix} 4.5 \\ 2.5 \end{bmatrix}$$

where Cov(.) and $v_{(.)}$ stand for a cluster's estimated covariance matrix and multivariate mean (centroid), respectively. At the iteration in question, the priors $\pi_{(.)}$ of each model component are estimated as:

$$\pi_{C_1} = 0.4$$
$$\pi_{C_2} = 0.6$$

Now consider a hypothetical data point **x** that we have not observed. What is the probability that this point can be produced by each one of the model components (clusters), if we don't know its value?

Proposed Solution:

Since we don't know the value of **x**, the probabilities are the cluster priors, i.e., without further information, there is 40% probability that a given **x** will be produced by C_1 and 60% probability that it will be produced by C_2.

b) Now, suppose we have observed a data point in our dataset as $\mathbf{x} = [2 \;\; 1.5]^T$. Assuming the EM-GMM estimates previously hypothesized in item (a), the Squared Mahalanobis distance from this observation to the clusters can be computed as $d_M(\mathbf{x}, \mathbf{v}_1)^2 = 10$ and $d_M(\mathbf{x}, \mathbf{v}_2)^2 = 9.10596$. What is the probability density of C_1 as evaluated at **x**? Repeat for C_2.

Proposed Solution:

The probability densities of each component in a GMM model are multivariate Gaussians, i.e.:

$$\mathcal{N}(\mathbf{x}_j \,|\, \mathbf{v}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{n/2} \det(\boldsymbol{\Sigma}_i)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_j - \mathbf{v}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_j - \mathbf{v}_i)\right\}$$

Mahalanobis distance (squared) from observation to cluster center

where Σ_i stands for the covariance matrix (of the *i*th component). The dataset is 2-dimensional as stated in item (a), so $n = 2$ in the equation above. The determinant of the covariance matrix of C_1 can be readily computed as $\det(\text{Cov}(C_1)) = 0.125 \times 0.125 - 0 \times 0 = (0.125)^2 = 0.015625$. Similarly, $\det(\text{Cov}(C_2)) = 0.8 \times 0.11 - 0.27 \times 0.27 = 0.0151$. By using the precomputed Squared Mahalanobis distances to readily input their values into the equation above, the probability density of C_1 as evaluated at **x** can be computed as $(1 / 2\pi \cdot 0.015625^{1/2}) \cdot \exp(-0.5 \cdot 10) = 0.008579021$. Similarly, the probability density of C_2 as evaluated at **x** can be computed as $(1 / 2\pi \cdot 0.0151^{1/2}) \cdot \exp(-0.5 \cdot 9.10596) = 0.01364575$.

c) What is the probability that the data point referred to in item (b), $\mathbf{x} = [2 \;\; 1.5]^T$, has been produced by each one of the model components (clusters), now that we have observed its value?

Proposed Solution:

This is asking for the posterior probability of a cluster C_i conditioned to an observation $\mathbf{x}_j$, i.e.:

$$\mu_{ij} = \frac{\pi_i \mathcal{N}(\mathbf{x}_j | \mathbf{v}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \pi_l \mathcal{N}(\mathbf{x}_j | \mathbf{v}_l, \boldsymbol{\Sigma}_l)}$$

Using the results in item (b), we can compute this probability for cluster C_1 as (0.4×0.008579021) / (0.4×0.008579021 + 0.6×0.01364575) = 0.295343 and for C_2 as (0.6×0.01364575) / (0.4×0.008579021 + 0.6×0.01364575) = 0.704657.

d) Now, consider a hypothetical dataset with 15 data points that has been clustered into $k = 3$ clusters using EM-GMM for a certain number of iterations. Suppose the estimated posterior probabilities μ_{ij} are as follows:

0.7	0.9	0.8	0.02	0.08	0.05	0.15	0.25	0.15	0.19	0.05	0.2	0.01	0.01	0.05
0.29	0.05	0.05	0.95	0.9	0.85	0.6	0.65	0.75	0.8	0.4	0.2	0.01	0.06	0.05
0.01	0.05	0.15	0.03	0.02	0.10	0.25	0.1	0.1	0.01	0.55	0.6	0.98	0.93	0.85

Answer the following questions: (i) If we want to obtain a hard (non-overlapping) partition of the dataset in a way that data objects are assigned to clusters according to their probabilities, what would be the result in this case? (ii) Interpret the 11th and the 13th columns of this matrix. (iii) If the algorithm were to be run for an extra iteration, what would be the updated priors associated with each cluster?

Proposed Solution:

Assigning each observation to the most likely cluster means finding the row (cluster) for which each column (observation) has its maximum value (posterior probability), resulting in the following partition: $C_1 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, $C_2 = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{10}\}$, $C_3 = \{\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}, \mathbf{x}_{15}\}$. The 11th column corresponds to an observation with a lot of uncertainty, particularly between clusters C_3 and C_2, whose probabilities are not largely different. This can occur, e.g., if an observation is between two clusters, possibly in a region of prominent overlap. This is in contrast to the 13th observation, where the estimated probabilities suggest that the observation belongs to C_3 almost certainly. The newly estimated priors would be as follows:

π_{C1} = (0.7+0.9+0.8+0.02+0.08+0.05+0.15+0.25+0.15+0.19+0.05+0.2+0.01+0.01+0.05)/15 = 0.2406667
π_{C2} = (0.29+0.05+0.05+0.95+0.9+0.85+0.6+0.65+0.75+0.8+0.4+0.2+0.01+0.06+0.05)/15 = 0.4406667
π_{C3} = (0.01+0.05+0.15+0.03+0.02+0.10+0.25+0.1+0.1+0.01+0.55+0.6+0.98+0.93+0.85)/15 =0.3153333