

1. Considering the following two-dimensional measurements:

	y_1	y_2
x_1	-2	2
x_2	-1	3
x_3	0	1
x_4	-2	1

- (a) What are the maximum likelihood parameters of a multivariate Gaussian distribution for this data set?
- (b) What is the Gaussian's shape? Draw its contour plot.

- (a) Regarding the motivation for finding the "maximum likelihood parameters", [this](#) and other articles are helpful to delineate the MLE method, which aims to find the parameters of a distribution that maximize the probability of observing a given data set. Considering a data set normally distributed (as referenced in the question's statement), the MLE method is equivalent to finding the parameters of the distribution that minimize the sum of squared errors between the data set and the distribution's probability density function.

As we know, a multivariate Gaussian distribution is defined by its mean vector μ and its covariance matrix Σ :

$$\mathcal{N}(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right)$$

The mean vector is simply the average of the data points (considering each coordinate):

$$\mu = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$

The covariance matrix, on the other hand, requires a bit more work, requiring us to calculate the deviations of each data point from the mean:

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$$

Note, of course, that Σ is a symmetric matrix. We can, then, start calculating the maximum likelihood parameters:

$$\mu = \frac{1}{4} \begin{bmatrix} -2 - 1 + 0 - 2 \\ 2 + 3 + 1 + 1 \end{bmatrix} = \begin{bmatrix} -1.25 \\ 1.75 \end{bmatrix}$$

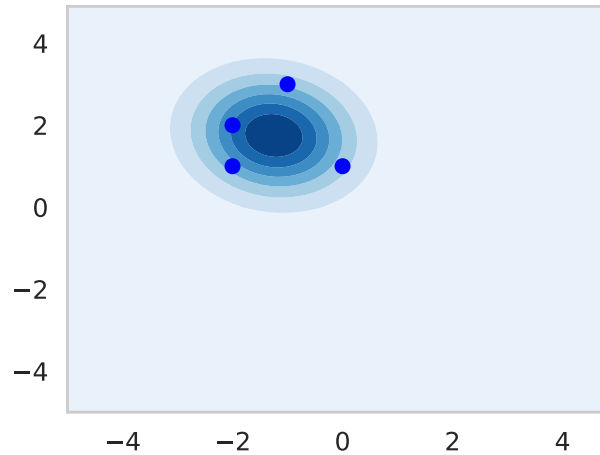
$$\Sigma = \frac{1}{3} \begin{bmatrix} (-2 + 1.25)^2 + \dots + (-2 + 1.25)^2 & (-2 + 1.25)(2 - 1.75) + \dots \\ (-2 + 1.25)(2 - 1.75) + \dots & (2 - 1.75)^2 + \dots + (1 - 1.75)^2 \end{bmatrix} = \begin{bmatrix} 0.916667 & -0.0833333 \\ -0.0833333 & 0.916667 \end{bmatrix}$$

As such, we can calculate both the determinant and the inverse of the covariance matrix, effectively gathering all the needed parameters for the multivariate Gaussian distribution:

$$|\Sigma| = 0.916667^2 - 0.083333^2 = 0.8(3)$$

$$X = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow X^{-1} = \frac{1}{ac - bd} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}; \quad \Sigma^{-1} = \frac{1}{0.8(3)} \begin{bmatrix} 0.916667 & 0.083333 \\ 0.083333 & 0.916667 \end{bmatrix} = \begin{bmatrix} 1.1 & 0.1 \\ 0.1 & 1.1 \end{bmatrix}$$

- (b) The following contour plot was computed utilizing Python (the code snippets are available in this sheet's respective notebook):



Contour plots are a great way to visualize the shape of a multivariate Gaussian distribution in two dimensions. The plot above shows the distribution's probability density function, which is proportional to the probability of observing a data point in a given region - the darker the region, the higher the probability of observing a data point in that region!

Essentially, each line in the plot above represents a constant probability density value. The ellipsoid is always centered at the mean vector, and its shape is determined by the covariance matrix. The ellipsoid's axes' directions are determined by the eigenvectors of the covariance matrix, and its axes' lengths are determined by the square root of the eigenvalues of the covariance matrix. The ellipsoid's axes' lengths are, then, proportional to the standard deviations of the distribution along each axis.

2. Consider the following data-set, paired with a query vector $x_{new} = [1 \ 1 \ 1 \ 1 \ 1]^T$:

	y_1	y_2	y_3	y_4	y_5	z
x_1	1	1	0	1	0	1
x_2	1	1	1	0	0	0
x_3	0	1	1	1	0	0
x_4	0	0	0	1	1	0
x_5	1	0	1	1	1	1
x_6	0	0	1	0	0	1
x_7	0	0	0	0	1	1

- Using Bayes' rule, without making any assumptions, compute the posterior probabilities for the query vector. How is it classified?
- What is the problem of working without assumptions?
- Compute the class for the same query vector under the naive Bayes assumption?
- Consider the presence of missings. Under the same naive Bayes assumption, how would you classify the query vector $x_{new} = [1 \ ? \ 1 \ ? \ 1]^T$?

-
-
-
-

3. Considering the following data set, paired with the query vector $x_{new} = [100 \ 225]^T$:

	y_1	y_2	z
x_1	170	160	0
x_2	80	220	1
x_3	90	200	1
x_4	60	160	0
x_5	50	150	0
x_6	70	190	1

- Compute the most probable class for the query vector assuming that the likelihoods are 2-dimensional Gaussians.
- Compute the most probable class for the query vector, under the Naive Bayes assumption, using 1-dimensional Gaussians to model the likelihoods.

-
-

4. Assuming training examples with m features and a binary class:

- (a) How many parameters are needed to estimate, considering boolean features and:
 - i. No assumptions regarding the data's distribution.
 - ii. Naive Bayes assumption.
- (b) How many parameters are needed to estimate, considering numeric-valued features and:
 - i. Multivariate Gaussian assumption.
 - ii. Naive Bayes assumption.

- (a) i.
- ii.
- (b) i.
- ii.