Statistical Methods of Machine Learning
Assignment 1

Martin Grunbaum
Martin Jorgensen
Thomas Barnholdt

February 18, 2014

I.1.1.1

Given

$$a = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} b = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

Then $a^T b = 1 * 3 + 2 * 2 + 2 * 1 = 3 + 4 + 2 = 9$

I.1.1.2

The *l2-norm* or *Euclidean norm* $||a|| = \sqrt{1^2 + 2^2 + 2^2} = 3$

I.1.1.3

The outer product

$$ab^T = \begin{bmatrix} 1*3 & 1*2 & 1*1 \\ 2*3 & 2*2 & 2*1 \\ 2*3 & 2*2 & 2*1 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 4 & 2 \\ 6 & 4 & 2 \end{bmatrix}$$

I.1.1.4

As M is a diagonal matrix the inverse matrix of M is

$$M^{-1} = \begin{bmatrix} 1/1 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

I.1.1.5

The matrix-vector product $Ma = \begin{pmatrix} 1*1+0*2+0*2 \\ 0*1+4*2+0*2 \\ 0*1+0*2+2*2 \end{pmatrix} = \begin{pmatrix} 1 \\ 8 \\ 2 \end{pmatrix}$

I.1.1.6

$$A^T = (ab^T)^T = \begin{bmatrix} 3 & 6 & 6 \\ 2 & 4 & 4 \\ 1 & 2 & 2 \end{bmatrix}$$

I.1.1.7

The rank of $A = 1$, because the rows are linearly dependent. We can verify this by observing that the first row can produce the second and third rows with a multiple, e.g. the second row (6 4 2) is the same as the first row (3 2 1) x 2.

I.1.1.8

As A is not full rank, it is not invertible.

I.1.2.1

The derivative of $f(w) = (wx+b)^2$ with respect to w is

$$\begin{aligned} ((wx+b)^2)' &= (w^2x^2 + 2wxb + b^2)' \\ &= 2x^2w + 2xb \\ &= 2x(xw+b) \end{aligned}$$

I.1.2.2

In general

$$\left(\frac{f}{g}\right)'(x) = \frac{f'(x)\cdot g(x) - f(x)\cdot g'(x)}{(g(x))^2}$$

Therefore, differentiating for w we get:

$$\begin{aligned}
f(x) &= 1 \\
f'(x) &= 0 \\
g(x) &= (wx+b)^2 \\
g'(x) &= 2x(wx+b) \\
\left(\frac{f}{g}\right)'(w) &= \frac{0\cdot(wx+b)^2 - 1\cdot 2x(wx+b)}{((wx+b)^2)^2} \\
&= \frac{-1\cdot 2x(wx+b)}{(wx+b)^4} \\
&= \frac{-2x}{(wx+b)^3}
\end{aligned}$$

I.1.2.3

In general

$$(f\cdot g)'(x) = f'(x)\cdot g(x) + f(x)\cdot g'(x)$$

Therefore, differentiating for x we get:

$$\begin{aligned}
f(x) &= x \\
f(x)' &= 1 \\
g(x) &= e^x \\
g(x)' &= e^x \\
(f\cdot g)'(x) &= 1e^x + xe^x
\end{aligned}$$

I.2.1

The plots with gaussian distributions for (μ, σ) pairs $(-1, 1)$, $(0, 2)$ and $(2, 3)$ can be seen in Figure 1. The code for generating the plots can be found in `unigauss_run.m`, and the code for our gaussian distribution function can be found in `unigauss.m`.

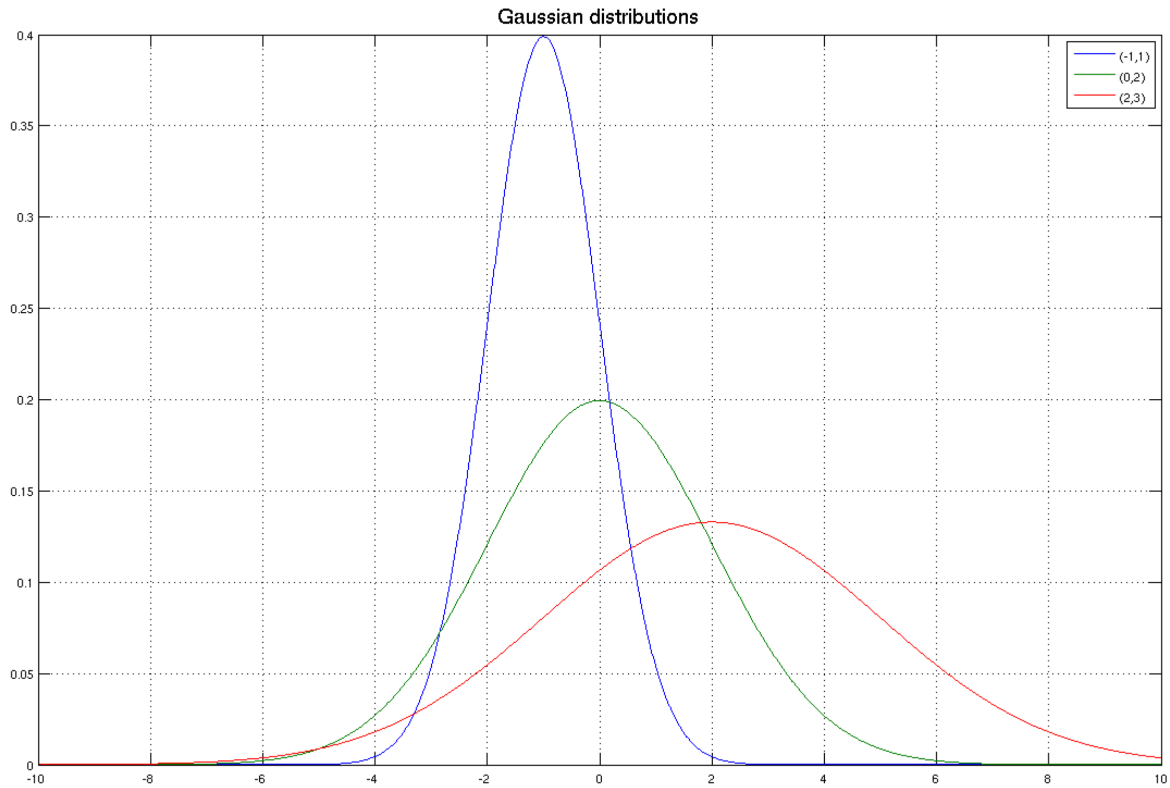

Figure 1: Gaussian distributions plotted with different values for (μ, σ).

I.2.2

Source code is available in `multigauss.m` and `multigauss_run.m`. Plot can be seen in Figure 2.

I.2.3

The $l2$ norm of x is

$$mean = \begin{pmatrix} 1 & 2 \end{pmatrix}^T$$
$$\mu = \begin{pmatrix} 1.0006 & 1.9834 \end{pmatrix}^T$$
$$||x|| = l2(mean - \mu) = 0.0366$$

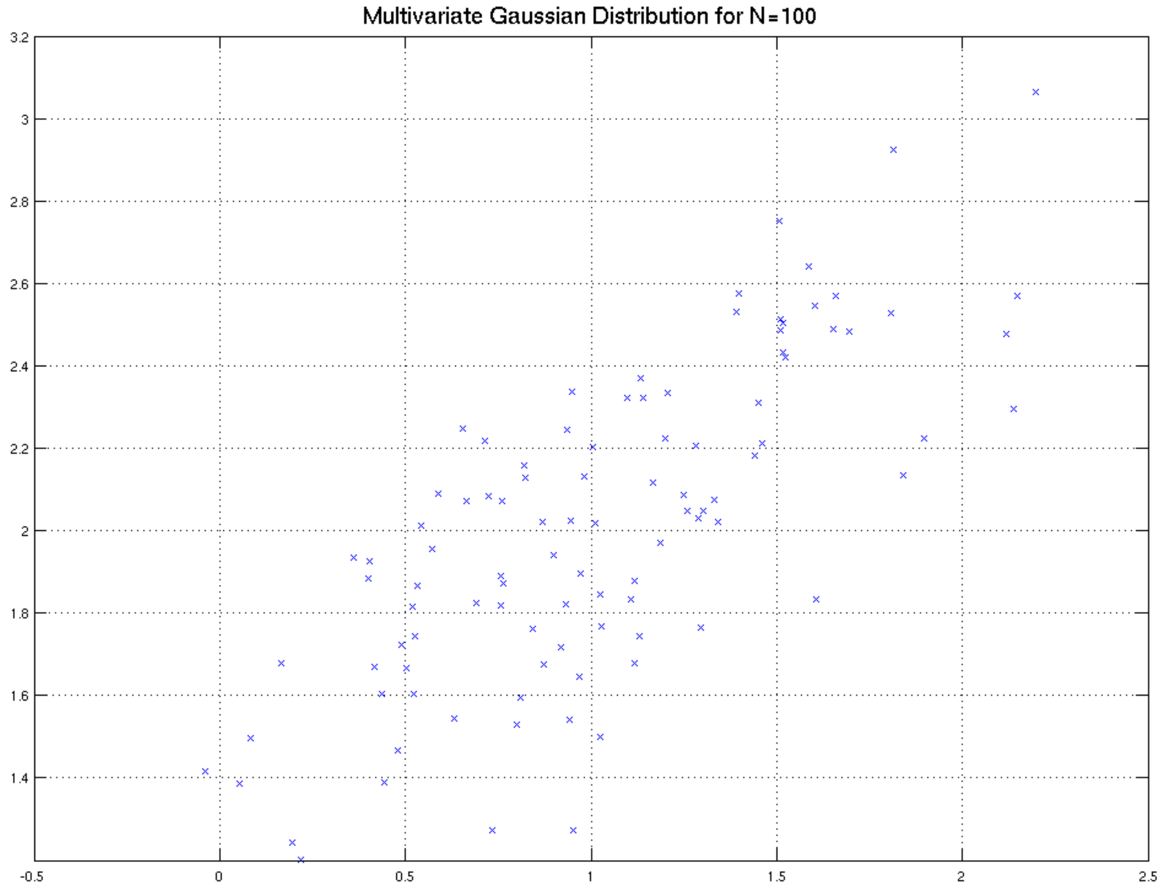

Figure 2: 100 points drawn from a 2-dimensional Multivariate gaussian distribution.

where $l2()$ is a function that calculates the *Euclidean norm* or $l2$ norm of the vector $mean - \mu$.

Figure 3 plots the points drawn along with a red circle for the calculated mean and a green circle for μ. There is a difference between the two because the mean is calculated based on the generated data drawn from the multivariate gaussian distribution at random. If we had a number of points approaching infinite, the difference would approach $\overline{0}$.

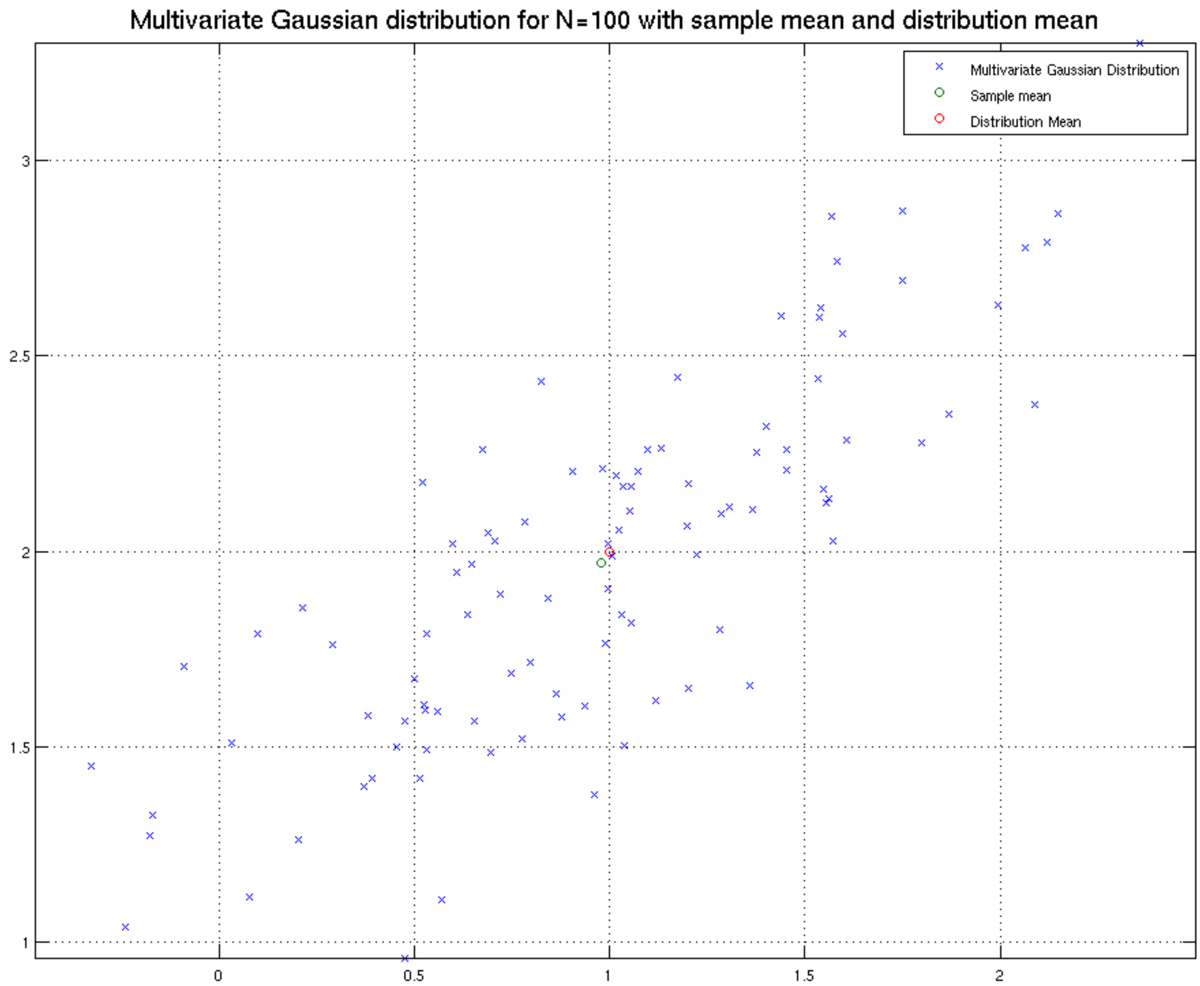

Figure 3: 100 points drawn from a 2-dimensional Multivariate gaussian distribution, plotted with the mean of the points and of μ.

I.2.4

The covariance matrix is full rank 2 and thus has two eigenvectors and eigenvalues. Each eigenvector represents a principal component (or linearly uncorrelated variable), and each eigenvalue a scalar representing the variance. Intuitively, the eigenvectors form a scaled and translated coordinate system centered at the mean of the multivariate Gaussian distribution (μ). If an eigenvalue is 0, the dimensionality is reduced by one. The larger of the two eigenvector/value pairs represents the direction where the ellipsis is widest. The other represents where the ellipsis is narrowest.

The covariance matrix we calculated can be found in Eq 1. Figure 4 shows a plot of the Multivariate gaussian distribution, plotted with the mean, μ and the two eigenvectors centered in the distribution μ. Figure 5 shows a plot of the 3 rotated distributions along with the distribution rotated to match the largest eigenvector along the x-axis. The angle needed for this was -37.2564^o in our case. Source code is available in `multigauss.m` and `multigauss_run.m`.

$$\Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})(x_n - \mu_{ML})^T$$
$$= \begin{pmatrix} 0.3239 & 0.2093 \\ 0.2093 & 0.2080 \end{pmatrix} \tag{1}$$

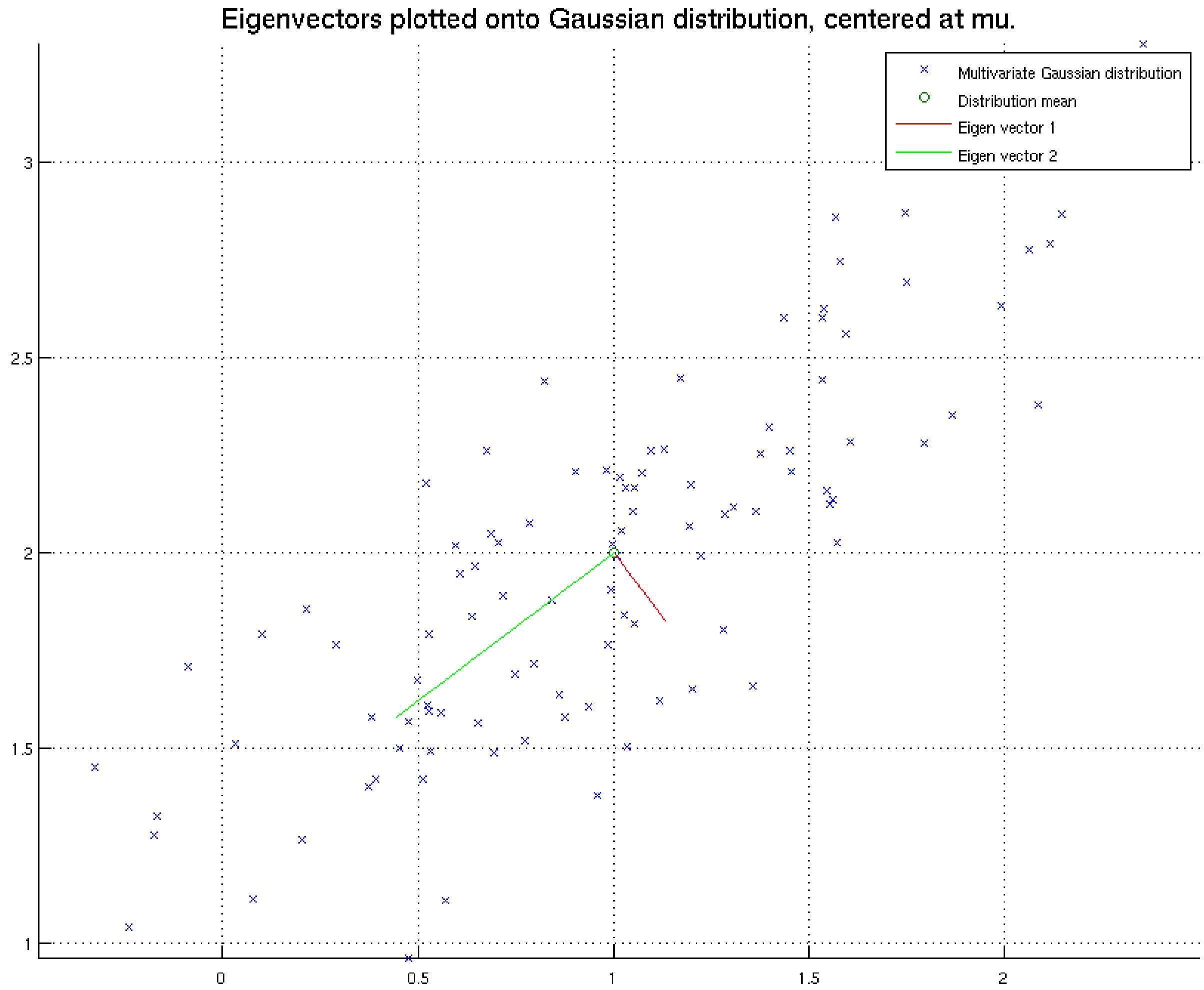

Figure 4: 100 points drawn from a 2-dimensional Multivariate gaussian distribution, plotted with the mean of the distribution, the value of μ and the two eigenvectors centered in the distribution μ.

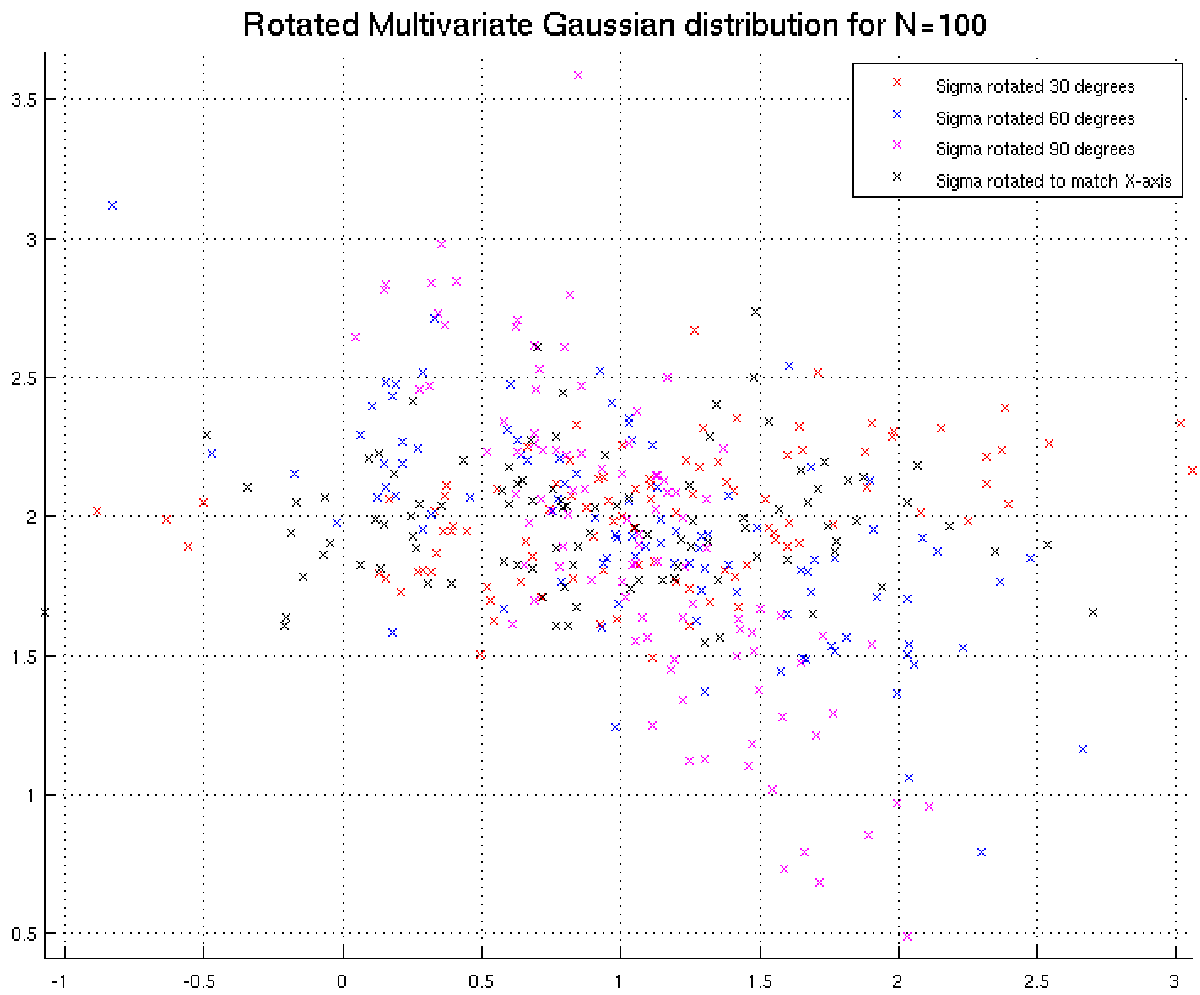

Figure 5: 100 points drawn from a 2-dimensional Multivariate gaussian distribution, rotated at 30, 60 and 90 degrees and lastly also aligned along the x-axis, all distributions in their own color.

I.3

Given:

$$\mu = \begin{pmatrix} \mu_a \\ \mu_b \\ \mu_c \end{pmatrix} \quad x = \begin{pmatrix} x_a \\ x_b \\ x_c \end{pmatrix} \quad \Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} & \Sigma_{ac} \\ \Sigma_{ba} & \Sigma_{bb} & \Sigma_{bc} \\ \Sigma_{ca} & \Sigma_{cb} & \Sigma_{cc} \end{bmatrix}$$

We wish to discover an expression for the conditional distribution $p(x_a|x_b)$ in which x_c has been marginalized out. We first find an expression for $p(x_a|x_b)$ and then marginalize out x_c.

We partition μ, x and Σ as follows:

$$\mu = \begin{pmatrix} \mu_d \\ \mu_c \end{pmatrix}, \text{where } \mu_d = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}$$
$$\overline{x} = \begin{pmatrix} x_d \\ x_c \end{pmatrix}, \text{where } x_d = \begin{pmatrix} x_a \\ x_b \end{pmatrix}$$
$$\Sigma = \begin{bmatrix} \Sigma_{aad} & \Sigma_{abd} \\ \Sigma_{bad} & \Sigma_{bbd} \end{bmatrix}$$
$$\text{where } \Sigma_{aad} = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}$$
$$\text{and } \Sigma_{abd} = \begin{bmatrix} \Sigma_{ac} \\ \Sigma_{bc} \end{bmatrix}$$
$$\text{and } \Sigma_{bad} = \begin{bmatrix} \Sigma_{ca} & \Sigma_{cb} \end{bmatrix}$$
$$\text{and } \Sigma_{bbd} = \begin{bmatrix} \Sigma_{cc} \end{bmatrix}$$

We also partition the precision matrix (inverse of the covariance matrix), as:

$$\Lambda \equiv \Sigma^{-1} = \begin{bmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{bmatrix} \tag{2}$$

Note: This recasts our problem to be to find a conditional distribution $p(x_d|x_c)$ and then to marginalize x_c out

A conditional distribution can be evaluated from the joint distribution $p(x) = p(x_d, x_c)$ by fixing x_c and normalizing. We know from [1] that if a joint distribution $p(x_d, x_c)$ is Gaussian, then the conditional distribution $p(x_d|x_c)$ is also Gaussian.

We wish to find $p(x_d|x_c)$, by considering the quadratic form in the exponent of the Gaussian distribution:

$$\begin{aligned} &-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) = \\ &-\frac{1}{2}(x_d-\mu_d)^T\Lambda_{aa}(x_d-\mu_d) \\ &-\frac{1}{2}(x_d-\mu_d)^T\Lambda_{ab}(x_c-\mu_c) \\ &-\frac{1}{2}(x_c-\mu_c)^T\Lambda_{ba}(x_d-\mu_d) \\ &-\frac{1}{2}(x_c-\mu_c)^T\Lambda_{bb}(x_c-\mu_c) \end{aligned} \tag{3}$$

A Gaussian distribution is completely characterized by its mean and covariance, so we must find expressions for the mean and covariance of $p(x_d|x_c)$, by completing the square. We denote the mean and covariance of this distribution by $\mu_{d|c}$ and $\Sigma_{d|c}$ respectively. By considering the functional dependence of Eq 3 on x_d in which x_c is regarded as a constant, we pick out all terms that are second order in x_d:

$$-\frac{1}{2}x_d^T\Lambda_{aa}x_d \tag{4}$$

From which we have that $\Sigma_{d|c} = \Lambda_{aa}^{-1}$. The terms in Eq 3 which are linear in x_d are $x_d^T\left\{\Lambda_{aa}\mu_d - \Lambda_a(x_c - \mu_c)\right\}$, making use of the fact that $\Lambda_{ba}^T = \Lambda_{ab}$ (See [1, p.85]). The coefficient of x_d in this expression is equal to $\Sigma_{d|c}^{-1}\mu_{d|c}$ and so:

$$\begin{aligned}\mu_{d|c} &= \Sigma_{d|c}\left\{\Lambda_{aa}\mu_d - \Lambda_{ab}(x_c - \mu_c)\right\}\\ &= \mu_d - \Lambda_{aa}^{-1}\Lambda_{ab}(x_c - \mu_c)\end{aligned} \tag{5}$$

Making use of the *Schur complement* of a matrix[1, p.87], we have that:

$$\begin{aligned}\Lambda_{aa} &= (\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}\\ \Lambda_{ab} &= -(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}\Sigma_{ab}\Sigma_{bb}^{-1}\end{aligned}$$

From these we now have the following for the mean and covariance of $p(x_d|x_c)$:

$$\mu_{d|c} = \mu_d + \Sigma_{ab}\Sigma_{bb}^{-1}(x_c - \mu_c) \tag{6}$$

$$\Sigma_{d|c} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba} \tag{7}$$

We must now find the margina distribution given by

$$p(x_d) = \int p(x_d, x_c)\; dx_c$$

We complete the square to integrate out x_c. The terms involving x_c are:

$$\begin{aligned}-\frac{1}{2}x_c^T\Lambda_{bb}x_c + x_c^T m &=\\ -\frac{1}{2}(x_c - \Lambda_{bb}^{-1}m)^T\Lambda_{bb}(x_c - \Lambda_{bb}^{-1}m) + \frac{1}{2}m^T\Lambda_{bb}^{-1}m\end{aligned}$$

Where $m = \Lambda_{bb}\mu_c - \Lambda_{ba}(x_d - \mu_d)$. This is again a standard quadratic form, and we note that the integration becomes an integral over an unnormalized Gaussian, whereby the result is the reciprocal of the normalization coefficient[1, p. 88]. The normalization coefficient depends only on the determinant of the covariance matrix, so we can integrate out x_c and the only term remaining that depends on x_d is m. Combined with the remaining terms from Eq 3 that depend on x_d we have:

$$\begin{aligned}&\frac{1}{2}[\Lambda_{bb}\mu_c - \Lambda_{ba}(x_d - \mu_d)]^T\Lambda_{bb}^{-1}[\Lambda_{bb}\mu_c - \Lambda_{ba}(x_d - \mu_d)]\\ &-\frac{1}{2}x_d^T\Lambda_{aa}x_d + x_d^T(\Lambda_{aa}\mu_d + \Lambda_{ab}\mu_c) + \text{const}\\ = &-\frac{1}{2}x_d^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})x_d\\ &+x_d^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1}\mu_d + \text{const}\end{aligned}$$

Where 'const' is quantities independent of x_d. The covariance and mean of the marginal distribution $p(x_d)$ are thus

$$\Sigma_d = (\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1}$$
$$\mu_d = \Sigma_d(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\mu_d = \mu_d \quad \text{(See [1, p.89,Eq 2.89])}$$

Making use of the *Schur complement* again we have that $(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1} = \Sigma_{aa}$. Finally we have that $\mathbb{E}[x_d] = \mu_d$ and $\text{cov}[x_d] = \Sigma_{aa}$. Intuitively, to obtain the marginal distribution over a subset of multivariate normal random variables, we drop the variables we want to marginalize out from the mean and covariance.

Description	K-value	Accuracy in %
Run on training data	1	100%
Run on test data	1	81.5%
Run on training data	3	86.0%
Run on test data	3	81.5%
Run on training data	5	83.0%
Run on test data	5	68.4%

Table 1: The results from I.4

1 I.4

1.1 I.4.1

The result of our KNN implementation for different k-values and datasets is shown in Table 1. The code to run this particular experiment is in `I_4_1.m`.

With $K = 1$ and running against the training set, the accuracy is 100% since any entry will be matched against itself, and only itself. We also see a general loss of accuracy as K increases. This may be because the point gets matched up against a larger and larger set of the total points, and if there is inherent density clusters in the data then we risk going further and further out as K increases.

1.2 I.4.2

The code for this experiment is in `I_4_2.m`. It uses several auxiliary files found in the same directory. Given a set of possible k values 'PossibleKValues' the following pseudocode runs cross-validation to find the average loss experienced amongst each of the five folds of a cross-validation for each given k value, and then selects the best k as the one with the lowest average loss:

for $k = 1$ to len(PossibleKValues) **do**
  subsets $\leftarrow$ shuffleSplit(dataset,5)
  for $cv = 1$ *to* 5 **do**
    test $\leftarrow$ subsets[cv] ▷ bucketJoiner function
    train $\leftarrow$ dataset $-$ test ▷ bucketJoiner function
    pred $\leftarrow$ kNN(k, train.X, train.y, test.X)
    loss[cv] $\leftarrow 1 - \frac{pred - test.y}{\text{length(pred)}}$
  end for
  avgLoss[k] $\leftarrow$ mean(loss)
end for

The k-value with the lowest average loss for our data was 5, with an accuracy of approximately 80%. The accuracy on the test data is $68,4\%$.

1.3 I.4.3

The code for this experiment can be found in `I_4_3.m`. The experiment is very similar to that of `I.4.4`. Before cross-validating, we normalize the data using the method found in `scale.m`. The $(\text{mean}, \text{var})$ of the training data is: $(3.0288, 7.8218)$, the test data after the normalization have the values $(0.1545, 1.0000)$. The most optimal K value is still 5, but the accuracy has increased to $71,05\%$ when using the normalized test set.

References

[1] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.