Solution of the third exercise assignment

Lucas Nogueira Ribeiro

The Python source code written to solve this assignment is available at `https://github.com/lnribeiro/patternrecognition`.

Exercise #01

First covariance matrix The first covariance matrix `C_I` and its partitions `Caa, Cbb, Cab, Cba` are given by:

```
C_I:
[[ 4.  0.  0.]
 [ 0.  4.  0.]
 [ 0.  0.  4.]]

Caa:
[[ 4.  0.]
 [ 0.  4.]]

Cbb:
4.0

Cab:
[[ 0.]
 [ 0.]]

Cba:
[[ 0.  0.]]
```

The precision matrix `P_I` and its partitions `Paa, Pbb, Pab, Pba` are given by:

```
P_I:
[[ 0.25  0.    0.  ]
 [ 0.    0.25  0.  ]
```

```
 [ 0.    0.    0.25]]

Paa:
[[ 0.25  0.  ]
 [ 0.    0.25]]

Pbb:
0.25

Pab:
[[ 0.]
 [ 0.]]

Pba:
[[ 0.  0.]]
```

The covariance matrix `condC` of the conditional distribution function is given by

```
 condC:
[[ 4.  0.]
 [ 0.  4.]]
```

The mean vector `mu` and its partitions `mu_a, mu_b` are given by

```
mu:
[[ 3.]
 [ 1.]
 [ 2.]]

mu_a:
[[ 3.]
 [ 1.]]

mu_b:
[ 2.]
```

The mean vector `condMean` of the conditional distribution function is given by

```
condMean:
[[ 3.]
 [ 1.]]
```

and $p(\mathbf{x}_a|\mathbf{x}_b)$ is equal to 0.00552447.

Second covariance matrix The second covariance matrix `C_II` and its partitions `Caa, Cbb, Cab, Cba` are given by:

```
C_II:
[[ 0.3      0.376    0.19   ]
 [ 0.376    0.5524   0.248  ]
 [ 0.19     0.248    0.41   ]]

Caa:
[[ 0.3      0.376 ]
 [ 0.376    0.5524]]

Cbb:
0.41

Cab:
[[ 0.19 ]
 [ 0.248]]

Cba:
[[ 0.19   0.248]]
```

The precision matrix `P_II` and its partitions `Paa, Pbb, Pab, Pba` are given by:

```
P_II:
[[ 23.49325875 -15.24256526  -1.66722678]
 [-15.24256526  12.37461623  -0.42150592]
 [ -1.66722678  -0.42150592   3.46660135]]

Paa:
[[ 23.49325875 -15.24256526]
 [-15.24256526  12.37461623]]

Pbb:
3.46660135224

Pab:
[[-1.66722678]
 [-0.42150592]]

Pba:
```

```
[[-1.66722678 -0.42150592]]
```

The covariance matrix `condC` of the conditional distribution function is given by

```
condC:
[[ 0.21195122  0.26107317]
 [ 0.26107317  0.40239024]]
```

The mean vector `mu` and its partitions `mu_a, mu_b` are given by

```
mu:
[[ 3.]
 [ 1.]
 [ 2.]]
mu_a:
[[ 3.]
 [ 1.]]
mu_b:
[ 2.]
```

The mean vector `condMean` of the conditional distribution function is given by

```
condMean:
[[ 2.53658537]
 [ 0.39512195]]
```

and $p(\mathbf{x}_a|\mathbf{x}_b)$ is equal to $1.36126245 \times 10^{-35}$.

Third covariance matrix The second covariance matrix `C_III` and its partitions `Caa, Cbb, Cab, Cba` are given by:

```
C_III:
[[ 0.3     0.      0.    ]
 [ 0.      0.5524  0.    ]
 [ 0.      0.      0.41  ]]
Caa:
[[ 0.3     0.    ]
 [ 0.      0.5524]]
Cbb:
0.41
Cab:
```

```
[[ 0.]
 [ 0.]]
Cba:
[[ 0.  0.]]
```

The precision matrix `P_II` and its partitions `Paa, Pbb, Pab, Pba` are given by:

```
P_III:
[[ 3.33333333  0.          0.        ]
 [ 0.          1.8102824   0.        ]
 [ 0.          0.          2.43902439]]
Paa:
[[ 3.33333333  0.        ]
 [ 0.          1.8102824 ]]
Pbb:
2.43902439024
Pab:
[[ 0.]
 [ 0.]]
Pba:
[[ 0.  0.]]
```

The covariance matrix `condC` of the conditional distribution function is given by

```
condC:
[[ 0.3      0.    ]
 [ 0.       0.5524]]
```

The mean vector `mu` and its partitions `mu_a, mu_b` are given by

```
mu:
[[ 3.]
 [ 1.]
 [ 2.]]
mu_a:
[[ 3.]
 [ 1.]]
mu_b:
[ 2.]
```

The mean vector `condMean` of the conditional distribution function is given by

```
condMean:
[[ 3.]
 [ 1.]]
```

and $p(\mathbf{x}_a|\mathbf{x}_b)$ is equal to $4.39968555 \times 10^{-8}$.

Exercise #02

The mean value of the random variable $\mathbf{x}_b$ is $\mu_b = \mathbb{E}[\mathbf{x}_b] = 2$ for all covariance matrices. In this exercise, we consider $\mathbf{x}_b = 1$.

First covariance matrix $p(\mathbf{x}_b) = 0.09666703$. In this case, the variance of $\mathbf{x}_b$ is equal to 4.

Second covariance matrix $p(\mathbf{x}_b) = 0.0496995$. In this case, the variance of $\mathbf{x}_b$ is equal to 0.41.

Third covariance matrix $p(\mathbf{x}_b) = 0.0496995$. In this case, the variance of $\mathbf{x}_b$ is equal to 0.41.

Exercise #03

The Student's t distribution is depicted in Figure 1 with the suggested parameters. Nothing could be concluded from this figure. Curves generated by different parameters were analyzed. This distribution converges to the standard normal distribution as $\upsilon \to \infty$ for $\mu = 0$ and $\lambda = 1$, as illustrated in Figure 2.

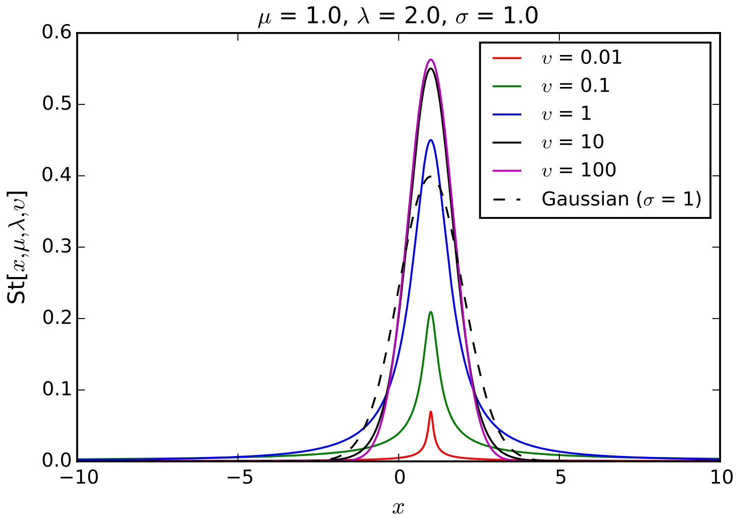

Figure 1: Student's t-distribution.

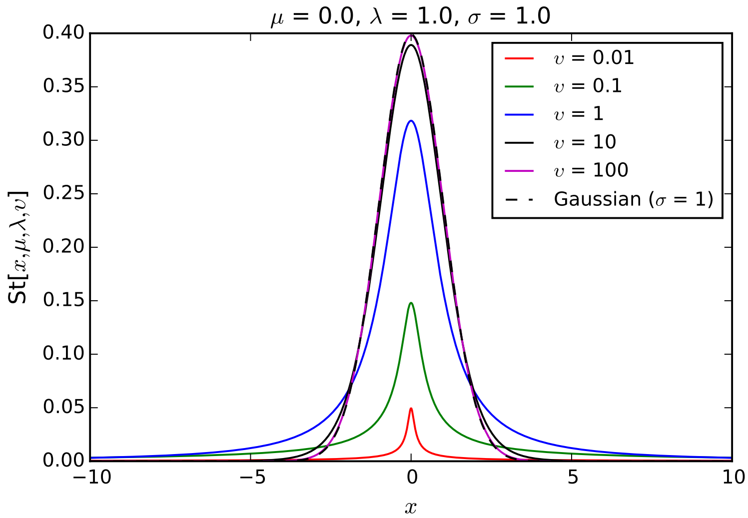

Figure 2: Student's t-distribution for $\mu = 0$ and $\lambda = 1$.

Exercise #04

The provided data was modeled as a mixture of Gaussians as depicted in Figure 3. The red curve is the distribution probability generated by mixing 6 normal distributions with $\sigma = 1$ and $\mu \in \{-10, -6, 12, 15, 19, 50\}$. Note, however, that this model is overfit, since the provided data might not represent the whole data population.

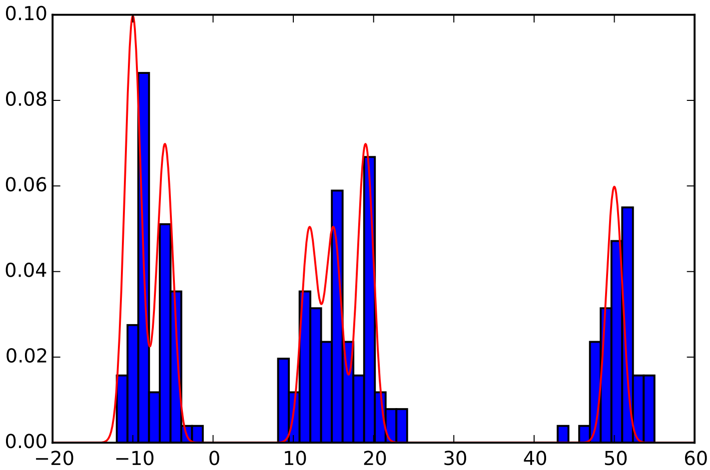

Figure 3: Estimated probability density function provided by the mixture of Gaussians model.

Exercise #05

Kernel density estimators were used in this exercise to estimate the density function of the provided data. The estimates provided by a Gaussian and a rectangular kernel for different window length h are depicted in Figures 4 and 5, respectively. As h increases, the estimated curve becomes smoother.

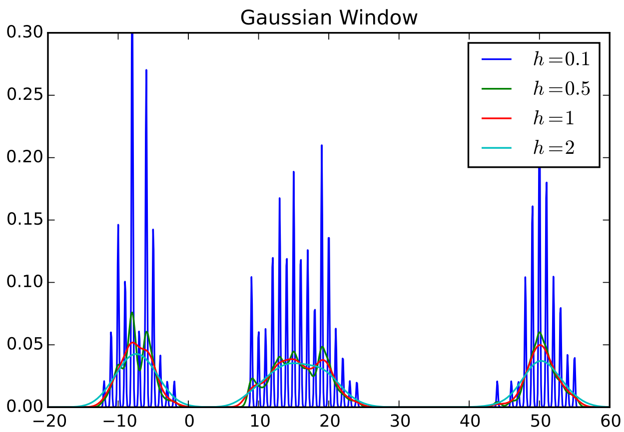

Figure 4: Probability density curve estimated using Gaussian kernel.

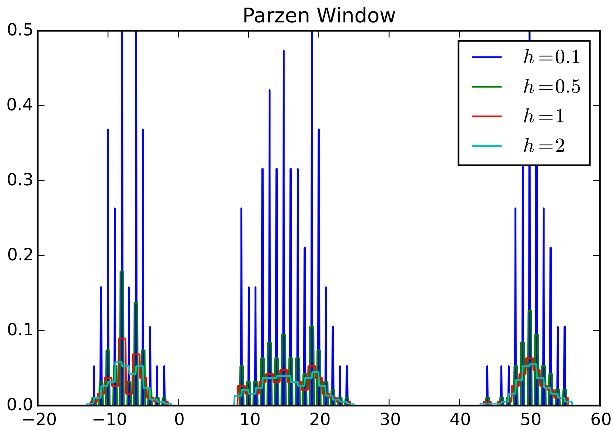

Figure 5: Probability density curve estimated using Parzen kernel.

Exercise #06

Identity basis function The mean square error (MSE) obtained in the training and test stages are

```
MSE [learning]:  1.03728146352
MSE [test]:  1.10867871973
```

The predicted output is shown in Figure 6. In this case, the predictions were not correct in general, resulting in a large MSE value.

Gaussian basis function In the evaluation stage, the MSE and the norm of the regression coefficents vector $\mathbf{w}$ were plotted in function of the number of basis function M in Figures 7 and 8, respectively. It was observed that $\|\mathbf{w}\|_2$ grows greatly with M, indicating that the model is overfit for large M. Additionally, it could be seen that the MSE did not decrease for values larger than 40. The number of basis function was set to $M = 45$.

The MSE obtained in the training and test stages were

```
MSE [learning]:  0.064172108973
MSE [test]:  0.05952864372
```

The predicted values are compared to the actual values in Figure 9. It can be seen that the predictions were accurate in general, as indicated by the low MSE values.

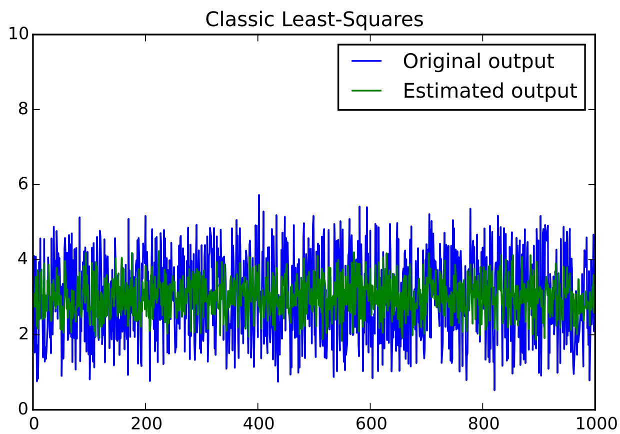

Figure 6: Predicted output using identity basis function.

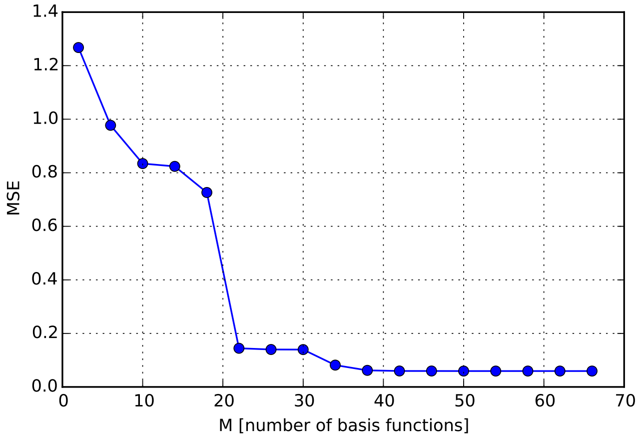

Figure 7: MSE vs. number of basis functions (M).

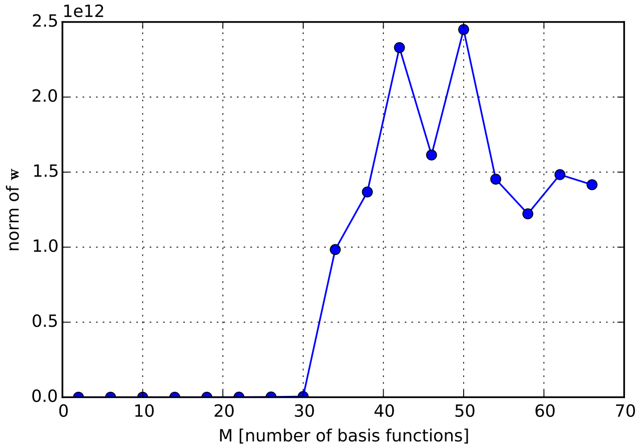

Figure 8: norm of $\mathbf{w}$ vs. number of basis functions (M).

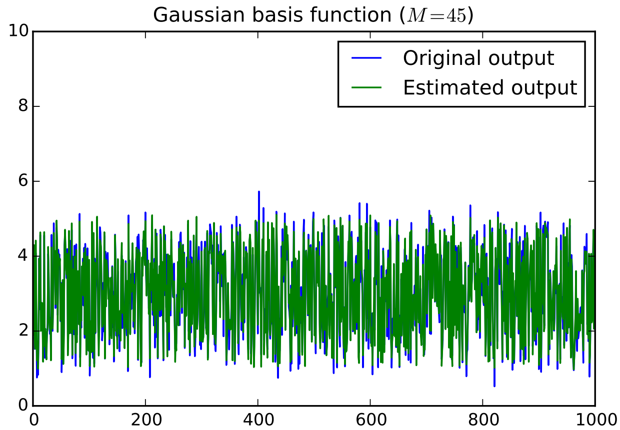

Figure 9: Predicted output using Gaussian basis functions.

Exercise #07

In the previous exercise, the norm of $\mathbf{w}$ grew greatly with the number of basis functions, indicating that the model may be overfit. By considering a regularization component that limits $\|\mathbf{w}\|_2$ in the objective function, the overfitting is reduced. In this exercise, the effect of regularization was analyzed by changing the values of the regularization factor λ. The MSE did not improve with the different values of the regularization parameter, as depicted in Figures 10. However, in Figure 11 it could be seen that the norm of $\mathbf{w}$ decreased as λ grew. In Figure 12, the predictions made with $\lambda = 1$ are compared to the actual values. The MSE in the training and test stages for this λ were:

```
MSE [learning]:  1.04197164446
MSE [test]:  1.11608842949
```

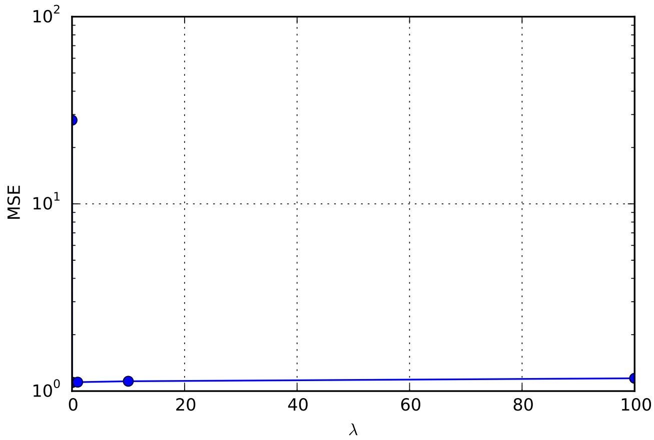

Figure 10: Probability density curve estimated using Parzen kernel.

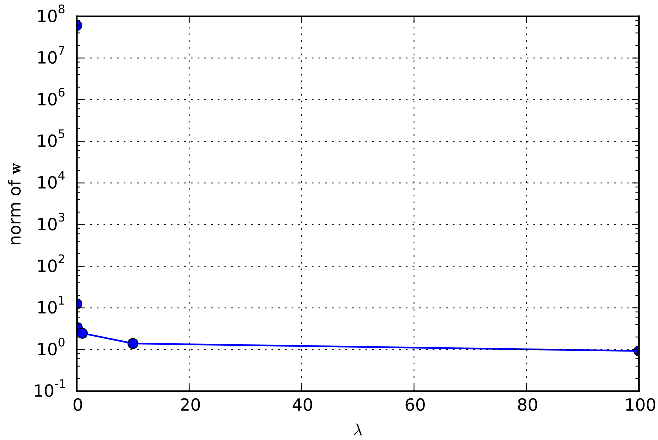

Figure 11: Probability density curve estimated using Parzen kernel.

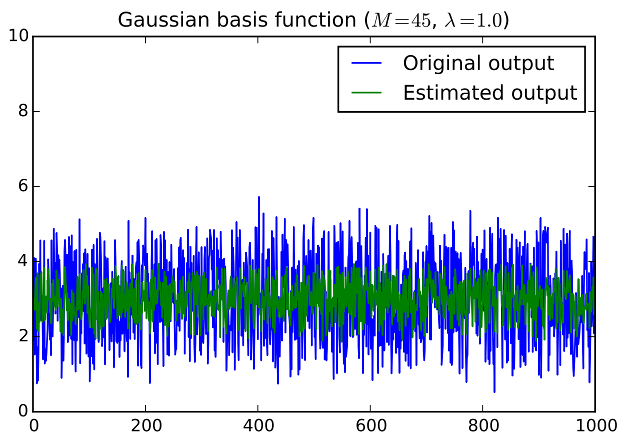

Figure 12: Probability density curve estimated using Parzen kernel.

Exercise #08

In this exercise a linear model was proposed to predict the provided data. The parameters of this model were estimated using a Bayesian linear regression method in which the posterior function was a zero-mean isotropic Gaussian distribution. In this case, the maximum posterior weight vector is equal to its mean vector. The estimated mean vector from the provided data is

```
mn:
[[-0.06587564]
 [ 0.4999401 ]]
```

and the predicted curve is depicted in Figure 13.

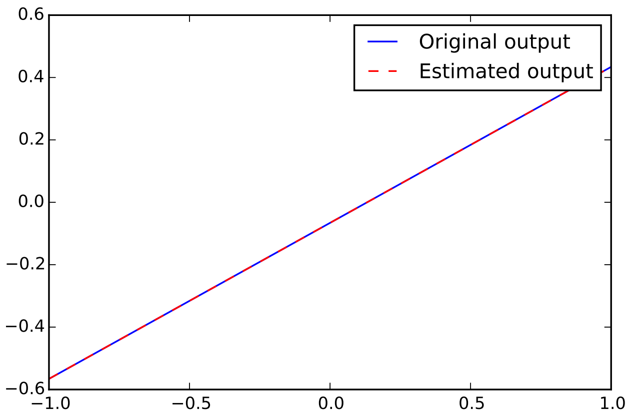

Figure 13: Bayesian linear regression.