Assignment 3

Juraj Šušnjara (s4846559)
Sebastian Tiesmeyer (s4373162)

November 2016

1 Exercise 1 – Bayesian linear regression

1.1

Data set consists of two data points:

$$\{x_1, t_1\} = (0.4, 0.05) \tag{1}$$

$$\{x_2, t_2\} = (0.6, -0.35) \tag{2}$$

Furthermore, we assume that $\alpha = 2$ and $\beta = 10$.

A target variable t is linearly dependent on input variable x subject to a random Gaussian noise with variance β^{-1}. Based on linear relationship we use regression model with polynomial basis functions of the form $f(x, \mathbf{w}) = w_0 + w_1 x$. For the weights, zero-mean Gaussian with precision α is assumed.

$$t = a_0 + a_1 x + \mathcal{N}(0, \beta^{-1}) \tag{3}$$

$$y(x, \mathbf{w}) = \Phi(\mathbf{x})^T \mathbf{w} = w_0 + w_1 x \tag{4}$$

$$p(\mathbf{w}|\alpha) = \mathcal{N}(0, \alpha^{-1}\mathbf{I}) \tag{5}$$

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}(\Phi(\mathbf{x})^T \mathbf{w}, \beta^{-1}) \tag{6}$$

Relationship between prior, likelihood and posterior are shown in following equations:

$$p(\mathbf{w}|\alpha) = \mathcal{N}(0, \alpha^{-1}\mathbf{I}) \tag{7}$$

$$p(\mathbf{t}|\mathbf{w}, \mathbf{x}) = \mathcal{N}(\mathbf{t}|\Phi\mathbf{w}, \beta^{-1}I) \tag{8}$$

$$\Rightarrow$$

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, S_N) \tag{9}$$

with

$$\mathbf{m}_N = \beta S_N \Phi^T \mathbf{t} \tag{10}$$

$$S_N^{-1} = \alpha I + \beta \Phi^T \Phi \tag{11}$$

Based on expression 4 we can see that vector of basis functions is $\Phi(x) = \begin{pmatrix} 1 & x \end{pmatrix}^T$. Now we need to define and calculate $\Phi^T t$ and $\Phi^T \Phi$ for our data set.

$$\Phi^T \mathbf{t} = \sum_n \Phi(x_n) t_n = N \begin{pmatrix} \mu_t \\ \mu_{xt} \end{pmatrix} \tag{12}$$

$$\Phi^T \Phi = \sum_n \Phi(x_n) \Phi(x_n)^T = N \begin{pmatrix} 1 & \mu_x \\ \mu_x & \mu_{xx} \end{pmatrix} \tag{13}$$

where

$$\mu_t = \frac{1}{N} \sum_n t_n \tag{14}$$

$$\mu_x = \frac{1}{N} \sum_n x_n \tag{15}$$

$$\mu_{xt} = \frac{1}{N} \sum_n x_n t_n \tag{16}$$

$$\mu_{xx} = \frac{1}{N} \sum_n x_n^2 \tag{17}$$

We can use previous expressions in order to compute posterior:

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, S_N) \tag{18}$$

$$\mathbf{m}_N = \beta S_N \Phi^T \mathbf{t} = N \beta S_N \begin{pmatrix} \mu_t \\ \mu_{xt} \end{pmatrix} \tag{19}$$

$$S_N^{-1} = \alpha I + \beta \Phi^T \Phi = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha \end{pmatrix} + N\beta \begin{pmatrix} 1 & \mu_x \\ \mu_x & \mu_{xx} \end{pmatrix} \tag{20}$$

Predictive distribution is given by expression 21 and it can be derived using the previously obtained posterior $p(\mathbf{w}|\mathbf{t}, \mathbf{x})$ as the prior for new observation and integrating out $\mathbf{w}$.

$$p(t|x, \mathbf{t}, \mathbf{x}, \alpha, \beta) = \mathcal{N}(t|m(x), s^2(x)) \tag{21}$$

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, S_N) \tag{22}$$

$$p(t|\mathbf{w}, \mathbf{t}, \mathbf{x}) = \mathcal{N}(t|\Phi(x)^T \mathbf{w}, \beta^{-1}) \tag{23}$$

$$\Rightarrow$$

$$p(t|x, \mathbf{t}, \mathbf{x}) = \mathcal{N}(t|m_N^T \Phi(x), \sigma_N^2(x)) \tag{24}$$

where

$$\sigma_N^2(x) = \frac{1}{\beta} + \Phi(x)^T S_N \Phi(x) \tag{25}$$

Finally we can derive expressions for $m(x)$ and $s^2(x)$:

$$m(x) = \Phi(x)^T m_N = N\beta \begin{pmatrix}1 & x\end{pmatrix} S_N \begin{pmatrix}\mu_t \\ \mu_{xt}\end{pmatrix} = \begin{pmatrix}1 & x\end{pmatrix} \begin{pmatrix}-0.0445 \\ -0.2021\end{pmatrix} \tag{26}$$

$$\begin{aligned} s^2(x) = \beta^{-1}\Phi(x)^T S_N \Phi(x) &= \beta^{-1} + \begin{pmatrix}1 & x\end{pmatrix} S_N \begin{pmatrix}1 \\ x\end{pmatrix} \\ &= 0.1 + \begin{pmatrix}1 & x\end{pmatrix} \begin{pmatrix}0.1233 & -0.1712 \\ -0.1712 & 0.3767\end{pmatrix} \begin{pmatrix}1 \\ x\end{pmatrix} \end{aligned} \tag{27}$$

$$S_N^{-1} = \begin{pmatrix}\alpha & 0 \\ 0 & \alpha\end{pmatrix} + N\beta \begin{pmatrix}1 & \mu_x \\ \mu_x & \mu_{xx}\end{pmatrix} \tag{28}$$

1.2

The plot is shown in figure 1. The figure clearly shows that currently, the initial prior still has a large effect. In a maximum likelihood solution, the line would now go through both of the data points. This can be shown by taking a lower value of α (or taking higher β), which will decrease the prior's certainty. The difference with the figure 3.8b in Bishop is explained by the fact that we use only a two-dimensional Gaussian basis function ϕ_j, whereas Bishop uses 9.

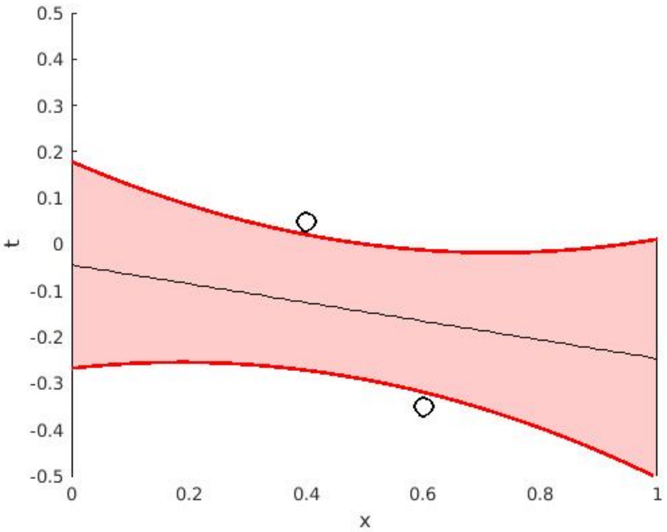


Figure 1: Mean and deviation of predictive Gaussian distribution with 2 points that are our data set.

1.3

In figure 2 we have drawn 5 different functions sampled from posterior distribution.

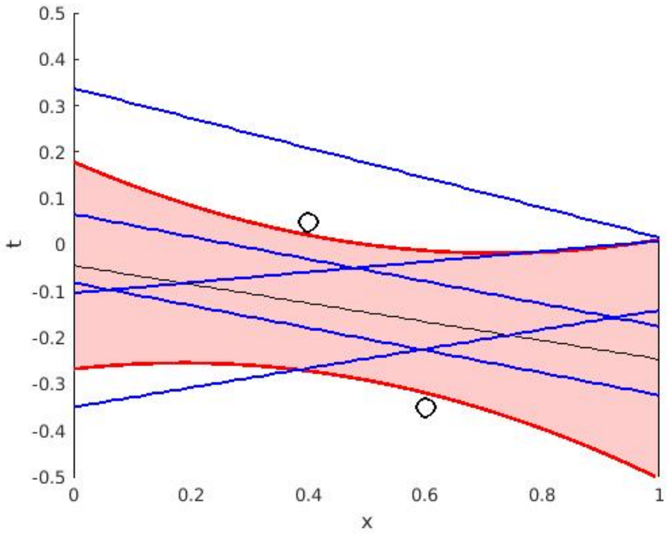


Figure 2: Blue lines represent 5 different functions $y(x, \mathbf{w})$ where weights are sampled from posterior distribution with mean m_N and variance S_N

2 Exercise 2 - Logistic regression

2.1 Part 1 - The IRLS algorithm

Alternative for gradient descent is Newton-Raphson iterative method which is represented in expression 29 where $\mathbf{H}$ represents the Hessian matrix of second derivatives of $f(x)$.

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \mathbf{H}^{-1}\nabla f(\mathbf{x}^{(n)}) \tag{29}$$

2.1.1

Expression for minimizing the function $f(x) = sin(x)$ using the Newton-Raphson iterative method is shown in 30.

$$x^{(n+1)} = x^{(n)} - \frac{f'(x^{(n)})}{f''(x^{(n)})} = x^{(n)} + \frac{cos(x^{(n)})}{sin(x^{(n)})} \tag{30}$$

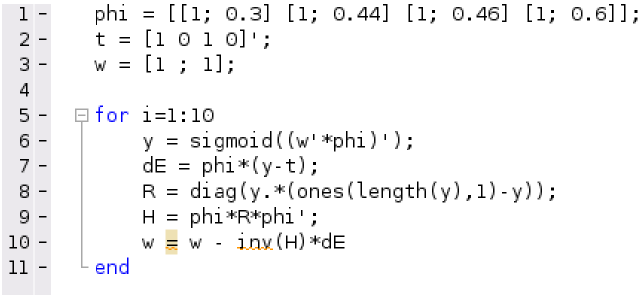
Starting from point $x^{(0)} = 1$ algorithm converges in 3 steps. Values in each step are: $x^{(0)} = 1, x^{(1)} = 1.6421, x^{(2)} = 1.5707, x^{(3)} = 1.5708$. If the starting point $x^{(0)} = -1$ algorithm will converge to point -1.5708 which is another local optima for $sin(x)$.

2.1.2

Based on given data set IRLS algorithm will converge in 5 steps to value

$$\mathbf{w} = \begin{pmatrix} 9.7823 \\ -21.7384 \end{pmatrix}$$

Algorithm is implemented in MATLAB and code is given in figure 3.



```
phi = [[1; 0.3] [1; 0.44] [1; 0.46] [1; 0.6]];
t = [1 0 1 0]';
w = [1 ; 1];

for i=1:10
    y = sigmoid((w'*phi)');
    dE = phi*(y-t);
    R = diag(y.*(ones(length(y),1)-y));
    H = phi*R*phi';
    w = w - inv(H)*dE
end
```

Figure 3: IRLS algorithm for given data set implemented in MATLAB

Now we need to show that this solution corresponds to decision boundary $\Phi = 0.45$ in the logistic regression model. For decision boundary we have next equality:

$$p(\mathcal{C}_1|\Phi) = p(\mathcal{C}_2|\Phi) = 0.5 \tag{31}$$

Now we need to solve that equation in order to calculate Φ.

$$p(\mathcal{C}_1|\Phi) = 0.5 \tag{32}$$

$$\frac{1}{1 + exp(w^T\Phi)} = 0.5 \tag{33}$$

$$w^T\Phi = 0 \tag{34}$$

$$w_0 + w_1\Phi = 0 \tag{35}$$

$$\Phi = -\frac{w_0}{w_1} \tag{36}$$

$$\Phi = 0.45 \tag{37}$$

2.2 Part 2 - Two-class classification using logistic regression

2.2.1

Scatter plot of given data is presented in figure 4. I think logistic regression is a good approach for this type of data because there is a clear separation between two classes. However, data is not linearly separable and thus standard feature space with dummy basis function will not work.

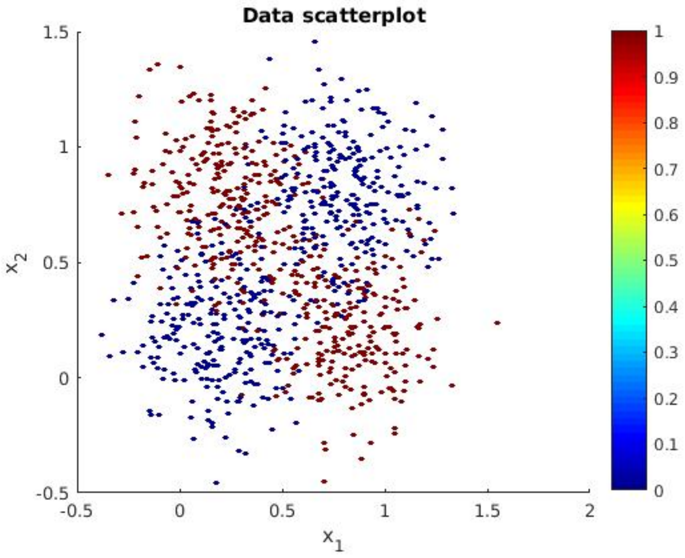


Figure 4: Scatter plot of given two-class data

2.2.2

With initial weights $\mathbf{w} = \begin{pmatrix}0 & 0 & 0\end{pmatrix}^T$ all class probabilities are 0.5 no matter what basis function is used. This is shown in expression 38.

$$p(\mathcal{C}_1|\Phi) = \sigma(\mathbf{w}^T\Phi) = \sigma(\begin{pmatrix}0\\0\\0\end{pmatrix}^T \Phi) = \sigma(0) = 0.5 \tag{38}$$

2.2.3

After running the IRLS algorithm to calculate weights for new data set with

$$\mathbf{w}_0 = \begin{pmatrix}0 & 0 & 0\end{pmatrix}^T$$

we get the result

$$\mathbf{w} = \begin{pmatrix}0.0044 & -0.0214 & -0.0493\end{pmatrix}^T$$

Scatter plot of the data is presented in figure 5. That plot is colored based on data point probabilities $p(C = 1|X_n)$. Colors are scaled from 0.47 to 0.51 because those are close to minimum and maximum values of calculated probabilities.

Cross entropy error is negative logarithm of likelihood and is shown in expression 39. The initial cross entropy error with initial weights $\mathbf{w}_0$ is 693.1472. After optimization cross entropy error is 692.9694. It improved only slightly because class probabilities are all around 0.5 and thus dummy basis function doesn't help to differentiate between two classes.

$$E(\mathbf{w}) = -ln(p(\mathbf{t}|\mathbf{w})) = -\sum_{n=1}^{N}\{t_n ln(y_n) + (1-t_n)ln(1-y_n)\} \tag{39}$$

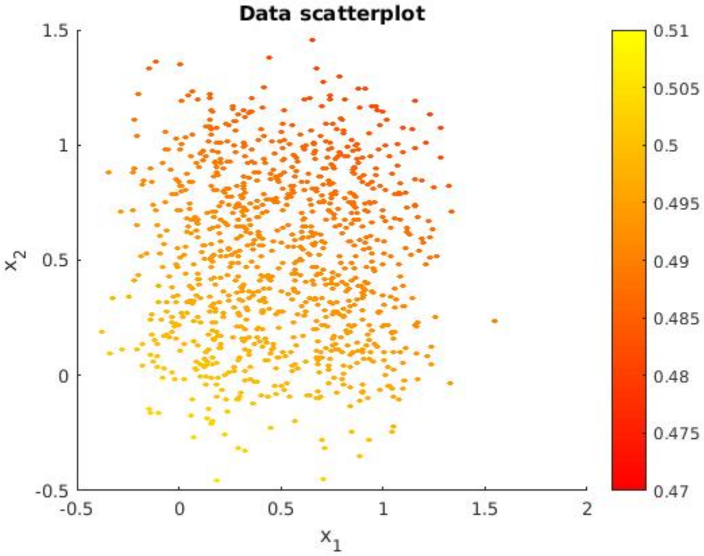


Figure 5: Scatter plot of data colored by their probabilities $p(C = 1|X_n)$

2.2.4

Φ_1 and Φ_2 are calculated using MATLAB's *mvnpdf* function. The resulting scatter plot of data in feature domain is shown in figure 6. With Gaussian basis functions as features it can be seen that logistic regression is good approach to classify the data because different classes can be linearly separated.

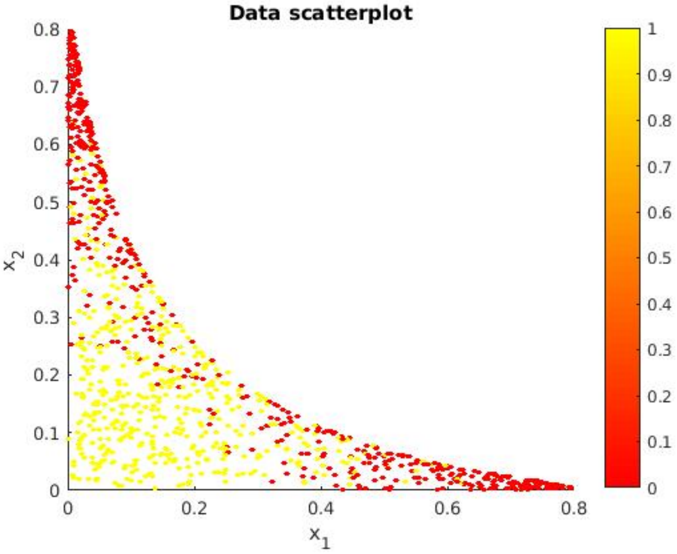


Figure 6: Data shown in feature domain

2.2.5

Result of IRLS algorithm that uses previously defined Gaussian basis functions as features (Φ_1 and Φ_2) is

$$\mathbf{w} = \begin{pmatrix} 7.1083 & -15.4214 & -15.5383 \end{pmatrix}^T$$

Cross error entropy is now 346.5041 which is much smaller than the one calculated before: 692.9694.

By mapping the data to different feature space that data becomes much more easier to linearly separate, although some values near the boundaries still have fairly uncertain probabilities of belonging to either class.

In figure 7 we can see scatter plot of data where colors represent probability of data point belonging to certain class.

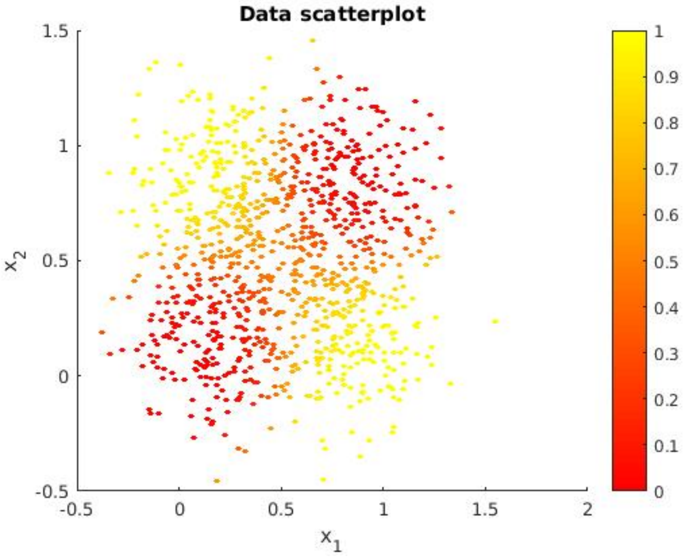


Figure 7: Scatter plot of data colored by their probabilities $p(C = 1|X_n)$