# Statistical Machine Learning 2016

Assignment 3
Deadline: Tuesday 29 November 2016

Instructions:

- **IMPORTANT: Write the full name of all team members on the first page of the report.**
- Weights of the exercises in this assignment:
  - Exercise 1: 8
  - Exercise 2, part 1: 5
  - Exercise 2, part 2: 7
- Working together in **pairs** (that is, at most two persons) and handing in a single set of solutions per couple is recommended.
- Write a **self-contained report** with the answers to each question, **including** comments, derivations, explanations, graphs, etc.
- Note: Answers like 'No', or 'x=27.2' by themselves are not sufficient; this hold also for results that are only available by running your code.
- Note: All figures should have axis labels and a caption or title that states to which exercise (and part) they belong.
- If an exercise requires coding, put **relevant code snippets** in your answer to the question in the report, and describe what it does. E.g. for a plot show how you compute the function.
- Upload reports to **Blackboard** as a **single pdf** file: '`SML_A3_<Namestudent(s)>.pdf`', in combination with **one zip-file** with the executable source/data files (e.g. matlab m-files).
- The grading will solely be based on the report pdf file. The source files are considered as supplementary material.
- Email addresses: `tomc@cs.ru.nl` and `b.kappen@science.ru.nl`
- For problems or questions: use the BB discussion board, email, or just ask.

# Exercise 1 – Bayesian linear regression

This exercise builds on exercise 2, week 8, "Fitting a straight line to data". For a detailed description (and explanation) see file `SML16_ex08+an.pdf` in Blackboard.
The final part of that exercise computed the predictive distribution after a single data point was observed. Here we consider a new data set, consisting of no less than *two* points: $\{x_1, t_1\} = (0.4, 0.05)$ and $\{x_2, t_2\} = (0.6, -0.35)$.

1. Assume $\alpha = 2$ and $\beta = 10$. Compute the predictive distribution $p(t|x, \mathbf{t}, \mathbf{x}, \alpha, \beta)$ after these two points are observed.

2. Plot the mean of the predictive Gaussian distribution and one standard deviation on both sides as a function of $x$ over the interval $[0, 1]$. Plot the data in the same figure. See `a009plotideas.m` in Blackboard for some plotting hints. Compare your plot with fig.3.8b (Bishop, p.157) and explain the difference.

3. Sample five functions $y(x, \mathbf{w})$ from the posterior distribution over $\mathbf{w}$ for this data set and plot them in the same graph (i.e. with the predictive distribution). You may use the Matlab function `mvnrnd`. See again `a009plotideas.m` for some plotting hints.

# Exercise 2 – Logistic regression

### Part 1 – The IRLS algorithm

Many machine learning problems require minimizing some function $f(\mathbf{x})$. For this, an alternative to the familiar gradient descent technique, is the so called Newton-Raphson iterative method:

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

where $\mathbf{H}$ represents the Hessian matrix of second derivatives of $f(\mathbf{x})$, see Bishop, §4.3.3.

1. Derive an expression for the minimization of the function $f(x) = \sin(x)$, using the Newton-Raphson iterative optimization scheme (1), and verify (using Matlab, just up to, e.g., five iterations) how quickly it converges when starting from $x^{(0)} = 1$. What happens when you start from $x^{(0)} = -1$?
Hint: The Hessian of a 1-dimensional function $f(x)$ is just the second derivative $f''$. So, the Newton-Raphson iterative method reduces in 1-d to

$$x^{(n+1)} = x^{(n)} - \frac{f'(x^{(n)})}{f''(x^{(n)})} \tag{2}$$

We want to apply this method to the logistic regression model for classification (see Bishop, §4.3.2):

$$p(\mathcal{C}_1|\boldsymbol{\phi}) = y(\boldsymbol{\phi}) = \sigma(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}) \tag{3}$$

For a data set $\{\phi_n, t_n\}_{n=1}^N$, with $t_n \in \{0, 1\}$, using $y_n = p(\mathcal{C}_1|\phi_n)$ the corresponding cross entropy error function to minimize is

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \tag{4}$$

With one basis function $\phi$ and the dummy basis function 1, the feature vector in (3) becomes $\boldsymbol{\phi} = [1, \phi]^{\mathrm{T}}$. The weight vector including the bias term is then also two dimensional, $\mathbf{w} = [w_0, w_1]^{\mathrm{T}}$. Expressions for the gradient $\nabla E(\mathbf{w})$ and Hessian $\mathbf{H}$ in terms of the data set are given in Bishop, eq.4.96-98. As both are implicitly dependent on the weights $\mathbf{w}$, they have to be recalculated after each step: hence this is known as the 'Iterative Reweighted Least Squares' algorithm.

Consider the following data set: $\{\phi_1, t_1\} = \{0.3, 1\}$, $\{\phi_2, t_2\} = \{0.44, 0\}$, $\{\phi_3, t_3\} = \{0.46, 1\}$ and $\{\phi_4, t_4\} = \{0.6, 0\}$, and initial weight vector $\mathbf{w}^{(0)} = [1.0, 1.0]^{\mathrm{T}}$.

2. Show using e.g. a Matlab implementation that for this situation the IRLS algorithm converges in a few iterations to the optimal solution $\widehat{\mathbf{w}}^{\mathrm{T}} \approx [9.8, -21.7]$, and show that this solution corresponding to a decision boundary $\phi = 0.45$ in the logistic regression model. (The IRLS algorithm should take about five lines of Matlab code inside a loop + initialization).

**Part 2 – Two-class classification using logistic regression**

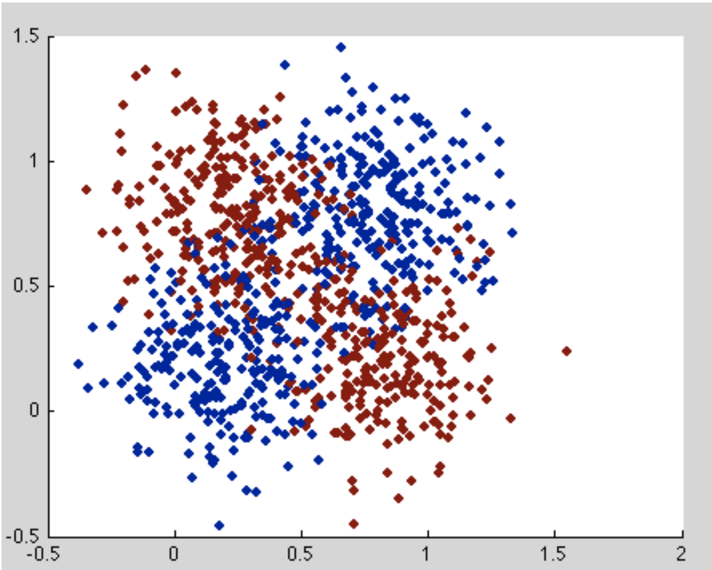


Figure 2.1 - Two class data for logistic regression.

Two-class classification using logistic regression in the IRLS algorithm. The data consists of 1000 pairs $\{x_1, x_2\}$ with corresponding class labels $\mathcal{C}_1 = 0$ or $\mathcal{C}_2 = 1$. Load it into Matlab using

```
data = load('a010_irlsdata.txt','-ASCII');
X = data(:,1:2); C = data(:,3);
```

1. Make a scatter plot of the data, similar to Figure 2.1. (Have a look at Matlab file `a010plotideas.m` in Blackboard for some ideas to make such a scatter plot and the plots later on.) Do you think logistic regression can be a good approach to classification for this type of data? Explain why.

2. Modify the Iterative Reweighted Least Squares algorithm from part 1 to calculate the optimal weights for this data set. Use again a dummy basis function. Initialize with the weight vector $\mathbf{w}^{\mathrm{T}} = [0, 0, 0]$. With these initial weights, what are the class probabilities according to the logistic regression model (i.e., before optimization)?

3. Run the algorithm. Make a scatter plot of the data, similar to Figure 2.1, but now with colors that represent the data point probabilities $P(C = 1|X_n)$ according to the model after optimization. Compare the cross entropy error with the initial value. Did it improve? Much? Explain your findings.

4. Introduce two Gaussian basis functions as features $\phi_1$, $\phi_2$, similar to Bishop, fig.4.12. Use identical, isotropic covariance matrices $\Sigma = \sigma^2 I$ with $\sigma^2 = 0.2$, and center the basis functions around $\boldsymbol{\mu}_1 = (0, 0)$ and $\boldsymbol{\mu}_2 = (1, 1)$. Make a scatter plot of the data in the feature domain. Do you think logistic regression can be a good approach to classification with these features? Explain why.

5. Modify the IRLS algorithm to use the features $\{\phi_1, \phi_2\}$ and the dummy basis function. Initialize with the weight vector $\mathbf{w}^{\mathrm{T}} = [0, 0, 0]$.

   Run the algorithm. Make a scatter plot of the data, similar to Figure 2.1, but now with colors that represent the data point probabilities $P(C = 1|X_n)$ according to this second model (after optimization). Compare the cross entropy error with the initial value. Did it improve? Much? Explain your findings.