COMP4670/8600 - 2014, Semester 1

Introduction to Statistical Machine Learning

Assignment 2

Maximum marks	20
Weight	20% of final grade
Submission deadline	Monday, May 19, 2014, 23:59
Document format	Portable Document Format (PDF); ASCII file for Pyton code
Submission mode	e-mail to Christfried.Webers (@nicta.com.au)
Formula Explanations	All formulas which you derive need to be explained unless you use very common mathematical facts. Picture yourself as explaining your arguments to somebody who is just learning about your assignment. With other words, do not assume that the person marking your assignment knows all the background and therefore you can just write down the formulas without any explanation. It is your task to convince the reader that you know what you are doing when you derive an argument.
Code quality	Python code should be well structured, use meaningful identifiers for variables and subroutines, and provide sufficient comments. Please refer to the examples given in the tutorials.
Code efficiency	An efficient implementation of an algorithm uses fast subroutines provided by the language or additional libraries. For the purpose of implementing Machine Learning algorithms in this course, that means using the appropriate data structures provided by Python and in numpy/scipy (e.g. Linear Algebra and random generators).
Late Penalty	20% per day overdue (a day starts at midnight!)
Cooperation	All assignments must be done individually. Cheating and plagiarism will be dealt with in accordance with University procedures (please see the ANU policies on “Academic Honesty and Plagiarism” `http://academichonesty.anu.edu.au`). Hence, for example, code for programming assignments must not be developed in groups, nor should code be shared. You are encouraged to broadly discuss ideas, approaches and techniques with a few other students, but not at a level of detail where specific solutions or implementation issues are described by anyone. If you choose to consult with other students, you will include the names of your discussion partners for each solution. If you have any questions about this, please ask the lecturer before you act.
Solutions	To be presented in the tutorials.

1 (2/20) Parametric Model

Assume a parametric model depending on all input data $\mathbf{x}_1, \dots, \mathbf{x}_N$ using a nonlinear feature mapping $\boldsymbol{\phi}(\mathbf{x})$ with an error function

$$E(\mathbf{w}) = f\left(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_1), \dots, \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_N)\right) + g(\mathbf{w}^T \mathbf{w})$$

where $g(z)$ is a monotonically increasing function of its scalar argument z.

Assume further that $\mathbf{w} = \mathbf{w}_{||} + \mathbf{w}_{\perp}$ splits into a component $\mathbf{w}_{||}$ which lies in the subspace $\mathbb{S}$ spanned by $\{\boldsymbol{\phi}(\mathbf{x}_1), \dots, \boldsymbol{\phi}(\mathbf{x}_N)\}$ and a component $\mathbf{w}_{\perp}$ in the orthogonal subspace of $\mathbb{S}$.

Prove that the $\mathbf{w}$ which minimises the error $E(\mathbf{w})$ takes the form of a linear combination of the basis functions $\boldsymbol{\phi}(\mathbf{x}_n)$, $n = 1, \dots, N$.

2 (2/20) Conditional Probability and Variance via Parzen Estimator

Given N data points $x_n \in \mathbb{R}$ and targets $t_n \in \mathbb{R}$, $n = 1, \dots, N$, consider a new input x and target t.

Using a *Parzen density estimator*, the joint probability for the new input and target, $p(x, t)$, can be estimated as

$$p(x, t) = \frac{1}{N} \sum_{n=1}^{N} f(x - x_n, t - t_n)$$

where $f(x, t)$ is called a *component density function.* Assume in the following, that the component density function is an isotropic Gaussian with mean $(0, 0)^T$ and covariance $\sigma^2 \mathbf{I}$ where $\mathbf{I}$ is the two-dimensional identity matrix.

1. Write down the conditional density $p(t \,|\, x)$, the conditional mean $\mathbb{E}\left[t \,|\, x\right]$, and the conditional variance $\operatorname{var}[t \,|\, x] = \int (t - \mathbb{E}\left[t \,|\, x\right])^2 \, p(t \,|\, x) \; \mathrm{d}t$, with the help of a function $k(x, x_n)$.

2. Show that for the function $k(x, x_n)$ the following holds

$$\sum_{n=1}^{N} k(x, x_n) = 1.$$

3 (2/20) Maximum Margin Hyperplane

Given two classes and a data set of only two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^D$, one from each class.

(a) Show that, irrespective of the dimensionality D of the data space, these two points are sufficient to determine the maximum margin hyperplane.

(b) What is the dimension of this maximum margin hyperplane?

(c) Given a new data point $\mathbf{x}$, provide a discriminant function $f(\mathbf{x})$ which classifies this new data point.

(d) Discuss the values of the Lagrange parameters.

4 (2/20) Constructing New Kernels

Prove that the following four operations create valid kernels $k(\mathbf{x}, \mathbf{x}')$ given that $k_3(\cdot, \cdot)$, $k_a(\cdot, \cdot)$, and $k_b(\cdot, \cdot)$ are valid kernels of their respective spaces, $\mathbf{A}$ is a symmetric positive semidefinite matrix, $\boldsymbol{\phi}$ is a function from $\mathbf{x}$ to $\mathbb{R}^M$, and $\mathbf{x_a}$ and $\mathbf{x_b}$ are variables (not necessarily disjoint) with $\mathbf{x} = (\mathbf{x_a}, \mathbf{x_b})$. For each of the kernels, define the feature mapping $\boldsymbol{\phi}(\mathbf{x})$ which allows a representation

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}'). \tag{1}$$

(a)
$$k(\mathbf{x}, \mathbf{x}') = k_3(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}'))$$

(b)
$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{A} \mathbf{x}'$$

(c)
$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x_a}, \mathbf{x}'_\mathbf{a}) + k_b(\mathbf{x_b}, \mathbf{x}'_\mathbf{b})$$

(d)
$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x_a}, \mathbf{x}'_\mathbf{a})\, k_b(\mathbf{x_b}, \mathbf{x}'_\mathbf{b})$$

5 (2/20) Kernel for XOR

The 2-ary XOR function is completely described by the following inputs $(x_1, x_2)^T \in \{-1, 1\}^2$ and output $y \in \{-1, 1\}$ given in Table 1.

x_1	x_2	y
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Table 1: 2-ary XOR

In the following, we extend the domain of x_1, x_2, y to the real numbers.

(a) Formally prove the following claim: The XOR problem is not linearly separable in the space of the input vectors $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$.

(b) Consider functions of the form $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + a)^b$, where a and b are integers. For the data of the XOR problem, choose the smallest values a and b in such a way that $k(\mathbf{x}, \mathbf{y})$ is a kernel for which the feature map $\boldsymbol{\phi}(\mathbf{x})$ maps into a space where the XOR problem becomes linearly separable. Justify your choice.

(c) For this choice of a and b, construct the explicit form of the mapping $\boldsymbol{\phi}(\mathbf{x})$.

6 (5/20) (Semi-/Un-)Supervised Learning and EM

In this section we will try various density estimation techniques on the Fisher Iris data set (from the course website). Remember, for unsupervised learning you are not allowed to use the class labels (5th column)!

(a) **Supervised Learning:** Fit three 4D Gaussians to the given data (i.e., one 4D Gaussian per class). Show the means and covariance matrices for each class.

Use 10-fold cross-validation to evaluate the classification error of this approach.

(b) **Unsupervised Learning:** Fit a mixture of 3 multivariate Gaussians to the data, using the EM algorithm to fit the means and covariance matrices. Provide a listing of your code.

Produce a table of both the expected log-likelihood and the log-likelihood of the data versus the EM iteration number for 5 restarts from different random initial conditions. Do you find the same solution each time?

(c) **Semisupervised Learning:** If 95% of the data in the Fisher data set was unlabeled, you could use a supervised classifier from part (a) trained on just the labeled data. Can you think of a way to incorporate the unlabeled data in using the EM algorithm (b) to produce a semi-supervised classifier? Describe such an algorithm. Which do you think would perform better? No implementation needed, but defend your answer.

7 (5/20) Rejection Sampling

In the lecture we discussed the 'Wallaby' distribution

$$p(x) = \frac{3}{10}\mathcal{N}(x \,|\, 5, 0.5) + \frac{3}{10}\mathcal{N}(x \,|\, 9, 2) + \frac{4}{10}\mathcal{N}(x \,|\, 2, 20)$$

1. Implement the rejection sampling algorithm and run it to create 10^6 samples drawn from this distribution. Report your choice of proposal distribution and the number of rejected samples.

2. Use the created samples to approximate the distribution $p(x)$ via a histogram method with bin width 0.1 over the interval $x = [-50, 50]$.

3. Report the sum of squared errors between $p(x)$ and the histogram approximation evaluated for all x at the centre of each bin.

4. Design a faster method to sample from $p(x)$ which does not use rejection sampling. Report the runtimes for both the rejection sampling version and your accelerated version.

5. Provide Python code which when executed outputs all the above report items.