CM50246: Machine Learning and AI Coursework

Neill D. F. Campbell
`n.campbell@bath.ac.uk`

Semester I, 2016-17

1 Aims and Objective

The aim of this course is to provide an understanding of probabilistic modelling techniques and their application to real world problems, specifically in the areas of computer vision and computer graphics.

A standard approach to applied computer science is to design a (mathematical) model that we believe reflects an aspect of the real world; for example, we may believe that the distribution of the height of all the trees on the campus is well modelled by a Gaussian (or Normal) distribution. Once we have such a model, we go out an collect observations of the phenomenon; in our example we could measure the heights of twenty trees. Given our model and this observed data we then wish to perform inference - the act of learning the parameters of the model to fit it to the real observations. With this learnt model, we can now make predictions about the future, e.g. the expected height of the next tree we see. As a final stage, we might wish to evaluate how well our model has performed compared to other possible models - so called model selection. This process forms the core of supervised learning and the course will provide the framework and algorithms to perform tasks of this nature. We will also go further to the more recent realm of unsupervised learning where we might have large quantities of unlabelled data (e.g. all the images on Google image search but we do not know which image contains what).

This course is presented as an introduction into research in this field (and the applied fields of vision and graphics). As such, the lectures will provide you with an in-depth introduction into the language of probability and then a wide overview of the algorithms and techniques that use this language to solve specific tasks. In the remaining section of the semester, the following research based coursework tasks (in part self-taught) will allow you to demonstrate that you can apply standard machine learning techniques to typical problems in machine learning, vision and graphics.

Please do not leave the coursework until the last minute! We are aware that students take a differing selection of modules with varying deadlines but please take the time at the start of the semester to plan your time with the deadlines at the start of the semester. Any issues with deadline clashes must be raised during the first week of term to ensure all students have a fair amount of time to complete their coursework.

Please come along to the lectures! The course is designed to be a research level course that helps students learn to teach themselves. In the lectures, we will try and cover the more difficult conceptual challenges of the course as well as providing an overview of the methods. This will need to be supported by following the suggested reading as well. The lectures will take as interactive a role as possible and I will endeavour to ensure that as many questions as possible are answered. There is also a discussion forum on the moodle site to enable students to share questions and answers. I strongly encourage you to make use of this. Please note that, with respect to questions asked outside of the lectures, priority will be given to students who have attended the lectures.

The following section describes the standard coursework tasks; there will always be scope to (optionally) extend the tasks into more advanced areas for interested students.

2 The Standard Coursework Tasks

What will I need to submit?

For the following tasks you will have to write and submit your code (in `matlab` or `python`), demonstrate it running on the test dataset and submit a *short* written report (as a PDF file). The report should be written in your own words and describe the machine learning models/algorithms/methods you have used for the task. Marks will be awarded for demonstrating conceptual understanding of the processes you are using. It is important to show that you have gained some insight into the problem you are addressing. This should performed by critical thinking:

- Why does the method work / not work?
- When does it work / not work? *(e.g. what assumptions must be valid)*
- How efficient is the method? *(e.g. some methods are more accurate but more computationally expensive)*
- How does it compare with other methods?
- How could you improve it?

The following provides a breakdown of the individual coursework tasks. The marks available for each task are indicated in square brackets (`[Marks]`). Please look at the ratio of the different marks available for different tasks when allocating your time for your coursework to ensure you allocate your effort appropriately. The extension tasks also have a mark allocation, shown as (*`[Extension Marks]`*). Please note, even if you complete multiple extensions, you will not be able to attain more than 100%.

2.1 Task I: Unsupervised Learning

In this task you will be provided with a simple dataset and a script for visualisation.

1. Write a function to perform a K-Means clustering of the dataset. `[10%]`
2. Write a function to fit a Gaussian Mixture Model (GMM) to the dataset. `[10%]`
3. Compare the results of the K-Means and GMM and evaluate the performance with differing numbers of clusters/components and different starting conditions. For the Gaussian Mixture Model start from both a random initialisation and use the output of the K-Means clustering as a starting point. `[10%]`
4. *Extension:* Perform Gaussian Kernel Density Estimation on the datasets and compare the predictive likelihood of some unseen (hold-out) data under the density estimator. Compare the predictive likelihood to that of the GMM and discuss the circumstances under which each might be more appropriate. *`[10%]`*

Some points to consider in your report:

- What can go wrong with the methods?
- When are they suitable?
- What are the advantages / disadvantages?
- How should the models be initialised and why?

Further details on unsupervised clustering and density estimation, as well as their evaluation, will be provided in the lecture course.

2.2 Task II: Supervised Learning (Regression)

In this task you will be provided with a simple dataset and a script for visualisation.

1. Write a function to perform maximum likelihood (ML) linear regression on the dataset. [5%]
2. Add a conjugate prior to perform maximum a priori (MAP) linear regression on the dataset. [5%]
3. Calculate the posterior (a Bayesian approach) for the linear regression with conjugate prior on the dataset. [10%]
4. Convert your linear regression model to a non-linear one using a polynomial feature space. Compare the results using a second order (quadratic), third order (cubic) and fourth order (quartic) feature space. [10%]
5. *Extension:* Implement Gaussian Process (GP) regression using a non-linear kernel (for example a radial basis or squared exponential kernel) or a Relevance Vector Machine (RVM). Use non-linear optimisation to determine the hyper-parameters of the kernel. *[10%]*

Some points to consider in your report:

- What are the differences between the ML, MAP and Bayesian approaches?
- When are they suitable?
- What are the advantages / disadvantages?
- What is meant by over-fitting (e.g. in the context of the non-linear regressor)?
- How should we pick the model to use for a dataset?
- How should the models be initialised and when is this important?

Further details on regressors, as well as their evaluation, will be provided in the lecture course.

2.3 Task III: Supervised Learning (Classification)

In this task you will be provided with a simple dataset and a script for visualisation.

1. Produce a *generative* Naive Bayes classifier for 1D data. Use Gaussians as the generating distribution. [5%]
2. Create a *discriminative* linear (Logistic regression) classifier using the sigmoid function as the activation function and train the parameters using maximum likelihood (ML). [10%]
3. Add a Gaussian prior to the parameters of your discriminative classifier and train using maximum a priori (MAP). [5%]
4. Use a polynomial feature space to convert the classifier to a non-linear discriminative classifier. Compare the results using a second order (quadratic), third order (cubic) and fourth order (quartic) feature space. [10%]
5. *Extension:* Implement a Bayesian *discriminative* linear Logistic classifier using Monte Carlo integration to perform inference. *[10%]*

Some points to consider in your report:

- What are the differences between generative and discriminative classifiers?

- When are they suitable?
- What are the advantages / disadvantages?
- Contrast the ML, MAP and Bayesian approaches?
- How should we pick the model to use for a dataset?

Further details on classifiers, as well as their evaluation, will be provided in the lecture course.

3 Assessment and Deadlines

3.1 What to Hand In for Assessment

For each task prepare a *short* (2-3 sides of A4 plus figures) report providing a brief overview of the design and implementation of the task with a results section that evaluates the performance of the task with suitable graphs. The discussion section of the report, the most important section, should address the topics presented at the start of Section 2; hints for topics to consider for each specific task are provided above.

In addition, for each task, you must submit in your source code with a single master script that, when run, will perform the training and testing for the task and generate all the output figures used in your report. Put all the source code, and a PDF copy of your report in a single ZIP file to be submitted on moodle.

Please ensure that the code you provide will run on the standard Department installations of either `matlab` or `python`. If you do need to use additional libraries, please provide them in the ZIP file. Please take the time to double check that the contents of your ZIP file is self-contained and will run successfully. If we are unable to run your code will we be unable to mark the coursework.

3.2 Division of Marks

Please note the marks available for each part of the task and allocate your time accordingly. It is possible to complete the course to a high standard (distinction) without attempting the extension tasks (provided the code and report is of good quality) therefore you are strongly encouraged to complete the compulsory tasks well before attempting any extensions. Please note - if you complete all the extensions you will not be able to attain more than 100%!

3.3 Programming Languages and Libraries

All the programming tasks in the coursework may be completed in either `matlab` or `python` (using the numerical `numPy` and `sciPy` toolkits); if any student wishes to use an alternative language please contact `n.campbell@bath.ac.uk` to receive permission first.

Both `matlab` or `python` have libraries (or existing source code available) that implement a number of the models and algorithms that are assessed in this coursework. Whilst these may be used to check or compare to your own implementations (remembering that algorithms with random initialisations may return different results on different runs), obviously, you may not just submit code that uses the high level libraries (e.g. the `kmeans` or `gmm` functions in `matlab`). As a guide you should not make use of the “machine learning” toolboxes in either language (with the possible exception for basic utility functions). The goal of the coursework is to demonstrate that you understand the models and algorithms at a level to implement them so please use your common sense in this respect; if you have any concerns about which library functions may be used then please get in contact.

3.4 Deadlines

The following deadlines will be observed.Please do not leave the coursework to the last minute and get in contact early if you believe you will have difficulties in meeting any of the deadlines. We will endeavour to mark timely submissions and provide feedback indicating whether the submission meets, exceeds or fails expectations within two semester weeks. In the case of below standard work an individual meeting will be arranged to discuss the shortcomings and how future work must be improved. Irrespective of the mark achieved, all students will be able to receive such an individual meeting, if they would like it, where more detailed feedback can be provided and any questions answered.

Deadline	**Date**
Materials for Task I	Friday 4th November
Materials for Task II	Friday 25th November
Materials for Task III	Friday 16th December

4 Individual Customisation of Coursework

Individual students may have their own projects or problems in mind that they would like to use machine learning techniques to address. To this end, there is scope to allow individuals to customise their coursework tasks. This may be by providing their own data or alternative input data to similar learning algorithms, or by proposing a different project that requires the implementation of a number of the machine learning techniques discussed in class. At the end of the intensive week of lectures, any student wishing to customise their coursework can arrange a meeting (email `n.campbell@bath.ac.uk`) to present an alternative plan for their coursework and we will review the proposal and grant the request as long as the proposal exceeds the minimum standard for the course (set as demonstrating the skills required to complete tasks 1 and 2 as defined in this document).

5 Plagiarism

Plagiarism is the attempt to pass-off someone else's work as your own, with the intention or expectation of receiving credit for it. It diminishes the degree you are on, the university you are in, and yourself; it is wholly unprofessional. Plagiarism can be intentional or un-intentional, but it is still plagiarism. You are free to use any resource you like as part of this course work. You can even reference and download material from the web if you wish to. But if you do use anything at all from anyone else, or anywhere else, then you must give full credit inside your document. The university web site explaining its position and how to cite others so as to avoid plagiarism can be found on the University's website[1].

[1] `http://www.bath.ac.uk/library/infoskills/referencing-plagiarism/`