Machine Learning and Pattern Recognition, Assignment Sheet 2012

School of Informatics, University of Edinburgh

Instructor: Amos Storkey

Handed out: Sun 19 Feb 2012
Submission Deadline: 4pm, Tues 20 March 2012

This assignment is worth 20% of the total course marks.

CAREFULLY FOLLOW THESE INSTRUCTIONS - Failure to do so will result in a mark of zero. To submit, convert your answers to `pdf` format. Check the file is readable in adobe reader on DICE, rename the file to `answers.pdf` and put it in *directory* called `answers`, along with any other files you used (including all code). From the directory containing the subdirectory `answers`, type

```
submit mlpr 1 answers
```

on DICE to submit your work. Please ensure the `answers.pdf` file is self contained (don't put answers in your code). Please ensure that the answers you give are written in your own words, and explain *your* understanding.

The marking will follow the standard University marking scheme. In the context of this assignment that means:

A Well explained description of points above plus extra achievement at understanding or analysis of results. Clear explanations, evidence of creative or deeper thought will contribute to a higher grade.

B Well explained answers to the questions.

C Fairly accurate answers to many questions, but significant deficiencies.

D Evidence that the student has gained some understanding, but not addressed that specified task properly.

E/F/G serious error or slack work.

The problem in this assignment has been set up as a private competition on Kaggle. If you wish to take part, please register for Kaggle at `http://inclass.kaggle.com/` and take part in the "MLPR Challenge". Taking part will give you an independent metric for your methods. Extra information on the problem is given on the Kaggle site.

In this assignment, you may reuse code from the course notes, or use other public code, so long as you know what it does. You may not use code written by other students.

Questions 2ff relate to the problem of predicting a pixel value given preceding pixels in an image. Images have a very complicated structure, with many short and long range dependencies, and heavy tailed pixel statistics. The problem of predicting the next pixel in an image is intimately tied to image compression: the coding cost is directly related to the predictive log likelihood of the data.

You are given the intensities of the preceding (above and to left) pixels in an image, and need to predict the intensity at the target pixel. A prediction consists of the (log) probabilities associated with each possible intensity value (0 to 63).

This is a useful problem from the point of view of the MLPR course, in that it can be viewed as either a discrete or real valued problem. Pixels intensities take particular discrete values, yet those values are related to one another as they are discretizations of a real valued quantity (light intensity).

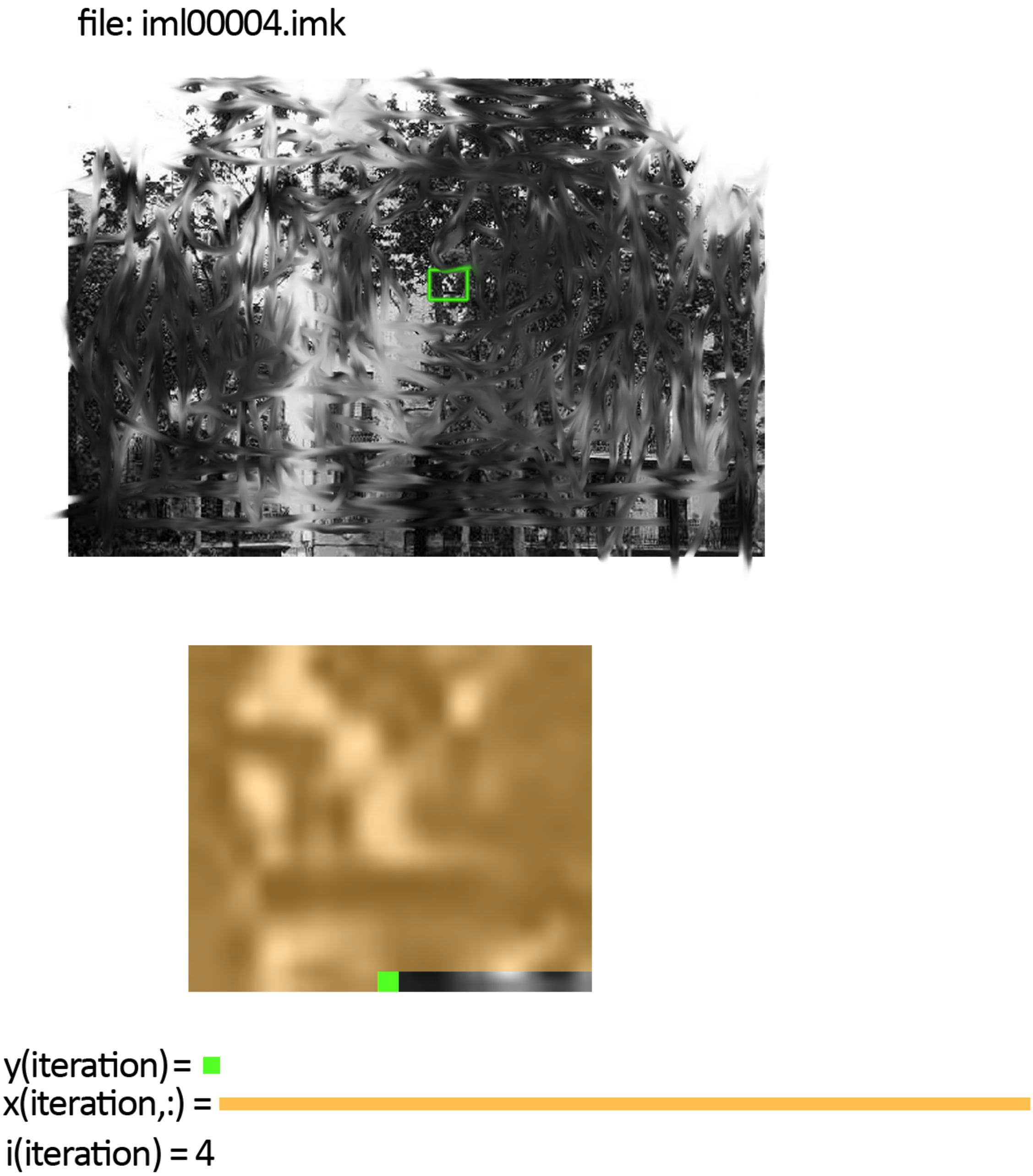
The data is also available on the informatics file system at

`/afs/inf.ed.ac.uk/group/teaching/mlprdata/challengedata/`.

There are three files

- `imdata.mat` is a matlab file that contains `x y i`. i is the image number. x is the data from the image above and to the left of the pixel to be predicted. y is the pixel intensity at the pixel to be predicted. The problem is find $P(y|x,i)$. You are advised to work with log probabilities $\log P(y|x,i)$.
- `imtestdata.mat` contains x and i which are the test points in the same format. The problem is to predict $P(y|x,i)$ for these test cases.
- `predprob.csv` is an example submission file for the Kaggle competition.

The illustration in the figure below clarifies the form of the data (the actual image is obfuscated). The image patch below is 35 pixels (horiz) x 30 pixels (vert). To convert the ith row of the x data into an image patch (with missing values at the end) you could use `reshape([x(i,:) zeros(1,18)],35,30)'`.



For Questions 3ff, spilt the training data into four equal consecutive parts (do not reorder the data: the data order is already randomized). In all your tests below evaluate your methods using 4-fold cross validation. To do this you will want to do some reading on cross-validation (search for '10-fold cross-validation' for example) so that you know what you are doing. Feel free to discuss this with one another. The evaluation metric you should use is the form of perplexity given below. Perplexity is directly related to test probability (and to compression rate). The equation you should use is:

$$\text{Perplexity} = \exp(-\frac{1}{N_t}\sum_{n=1}^{N_t} \log P(x_n|\mathcal{M})) \tag{1}$$

where N_t is the number of test cases.

Questions

1. (2 Marks, 2 Hours max). Consider each of the following examples. Explain the problems of the approach suggested and suggest what you would choose to do instead and why.

   (a) I have two classifiers for a problem, which I have learnt on a small dataset. To get the best predictive classification, I should choose the best of the two classifiers using maximum likelihood, and predict using that classifier.

   (b) I wish to learn a nonlinear regressor for a problem where I have data from a number of more normal cases, but wish to predict in more extreme scenarios (scenarios that it is had to get training measurements for). I choose to use an adaptive radial basis function network (that is a linear parameter model with radial basis features, and where the feature locations are parameters that are optimised).

   (c) I have a problem I have not tackled before, and I know little about the problem. I choose to use a multi-layer perceptron with 2 hidden layers as my model because it is very flexible and can generally approximate any non-linear function with a sufficiently large number of hidden units.

2. (5 Marks, 5 Hours max) Load the data at

   `/afs/inf.ed.ac.uk/group/teaching/mlprdata/challengedata/`

   into Matlab (this matches the data on the Kaggle site). Note the data is in integer format for space reasons. You will probably need to convert this to double format before doing anything: e.g. `x = double(x)`.

   Spend some time visualizing the data in Matlab. Referring to the course notes, and David Barber's book, write code to do PCA on the `x` data.

   (a) Display the mean image patch, and the first three principal vectors (the directions in the original space corresponding to the directions of greatest variation) as image patches (the last pixels will be missing as in the figure shown earlier). The Matlab commands `image` and `imagesc` may be useful. Display these figures in your report.

(b) Which data item in the training set is worst described by the 3 dimensional PCA representation (the mean squared error between the original data and its projection onto the 3D principal subspace)? Give the item number and display the image patch. What is it about this image that is hard to represent?

(c) Display a histogram of the targets. Use a 64 bin histogram. Comment on what you observe (e.g. the from of distribution, the spread, whether it is unimodal etc.).

(d) For each data point, compute the difference between the training target and the last element of the training data (that is, $y(i) - x(i, \mathit{end})$) and display a histogram of these values with 64 bins. Comment on what you observe in relation to the previous histogram. Note the *end* notation used here matches that of MATLAB and in this case refers to the last attribute.

3. (5 Marks, 5 Hours max) Using just the terms $x(:, \mathit{end})$, $x(:, \mathit{end} - 34)$ and $x(:, \mathit{end} - 35)$ as your data, do naive Bayes on this data. Note this involves *multinomial* distributions not Bernoulli ones, as the each attribute takes one of a number of multiple values (again the colon notation is as used in MATLAB).

   (a) First, consider using maximum likelihood. What problems would you face from using maximum likelihood to do parameter estimation for Naive Bayes in this setting?

   (b) Second do a Bayesian version using a Dirichlet distribution for the parameters $\boldsymbol{\theta}$: $P(\boldsymbol{\theta}|\alpha_1, \alpha_2, \ldots, \alpha_{64})$ with $\alpha_i = 1$. The Dirichlet distribution is a conjugate prior for the multinomial distribution. To compute the posterior distribution look up the conjugacy rules for the Dirichlet multinomial (if not vandalised, *Wikipedia: conjugate prior* provides a good table for the parameter updates).

      i. Give the 4-fold cross-validation perplexity (as defined earlier) for this application of Naive Bayes.

      ii. Visualise the predictions you get for a few cases (use histograms). Discuss the disadvantage of using Naive Bayes here: what does it fail to capture, and what effect does that have on the predictive distributions?

      iii. What is the strong assumption made by Naive Bayes? How valid is that assumption in this setting?

      iv. Optional: submit your result to Kaggle.

4. (5 Marks, 5 Hours max)

   (a) Write or find code to do linear regression. Using the same $x(:, \mathit{end})$, $x(:, \mathit{end} - 34)$ and $x(:, \mathit{end} - 35)$, do linear regression for this problem, and report the 4-fold cross-validation perplexity.

   (b) Now using the first 10 components of your PCA representation for the data, do linear regression, and report the 4-fold cross-validation perplexity.

   (c) Discuss the difference between these methods and the results you achieve. Is one particularly better than the other, and if so why?

5. (3 Marks) Experiment with other methods that you think will work well on this problem. Write two or three paragraphs summarising what you do, why you thought it would be a good idea, the results you get and your assessment of why you get the performance you get. If you wish, submit your results to Kaggle, and report the performance.