School of Electronics and Computer Science
University of Southampton

COMP6212 (2016/2017): Computational Finance Lab 0

Issue	Monday, 13 February 2017
Complete by	Wednesday 22 February 2017(12:00)

This is a preparatory exercise for students who have not taken the autumn semster module in Machine Learning (COMP3206/COMP6229).

If you are unfamiliar with MATLAB, you should spend some energy on becoming skilled at it. We assume that at this level (Part III / MSc) if you are competent in one high level language, picking up the basics of a scripting language should be straightforward.

- Familiarize yourself with MATLAB. Work through the examples in the document `http://users.ecs.soton.ac.uk/mn/MatlabIntroduction.pdf`, which are notes accompanying a textbook on MATLAB. Use of the `help` and `lookfor` commands help you learn a broad range of the features of the language. The documents `http://users.ecs.soton.ac.uk/mn/MatlabProgramming.pdf` and `http://users.ecs.soton.ac.uk/mn/MatlabStyle.pdf` are also worth going through, but not essential to get started.

Multivariate Gaussian

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{C}),\ \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \implies \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{m},\ \boldsymbol{A}\boldsymbol{C}\boldsymbol{A}^T)$$

1. Generate 1000 uniform random numbers and plot a histogram. Here are the useful commands in MATLAB to do this.

```
> x = rand(1000,1);
> hist(x,40);
> help hist
> [nn, xx] = hist(x);
> bar(xx);
```

   Repeat the above with 1000 random numbers drawn from a Gaussian distribution of mean 0 and standard deviation 1 using `x = randn(1000,1);`.
   Now try the following

```
> N = 1000;
> x1 = zeros(N,1);
> for n=1:N
>   x1(n,1) = sum(rand(12,1))-sum(rand(12,1));
> end
> hist(x1,40);
```

   What do you observe? Is there a theorem that explains your observation?
   Note: Do not type the MATLAB statements one at a time into command line; type them into a text file `labone.m` and invoke the script by `> labone` against the prompt. Look up `> help path`.

2. Consider the covariance matrix $\boldsymbol{C} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$.
   Factorize this into $\boldsymbol{A}^t\boldsymbol{A} = \boldsymbol{C}$ using `> A = chol(C)`.
   Confirm the factorization is correct by multiplying. Generate 1000 bivariate Gaussian random numbers by `X=randn(1000,2);`
   Transform each of the two dimensional vectors (rows of `X`) by `> Y=X*A`.
   Now draw a scatter plot of `X` and `Y`.
   `> plot(X(:,1),X(:,2),'c.', Y(:,1),Y(:,2),'mx');`
   What do you observe?

   Construct a vector $\boldsymbol{u} = [\sin\theta \ \cos\theta]$, parameterized by the variable θ and compute the variance of projections of the data in $\boldsymbol{Y}$ along this direction:

```
> theta = 0.25;
> yp = Y*[sin(theta); cos(theta)];
> answer = var(yp)
```

   Plot how this projected variance changes as a function of θ:

```
> N = 50;
> plotArray = zeros(N,1);
> thRange = linspace(0,2*pi,N);
> for n=1:N
> ...
> end
> plot(plotArray)
```

   Explain what you observe by calculating the eigenvectors of the covariance matrix.
   How does what you have done above differ for $\boldsymbol{C} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$.
   Export the figures for inclusion in a report `> print -depsc f1.eps`.

Linear Regression

In this task we will use the convex optimization package `CVX`. Download the appropriate version of the package from `http://cvxr.com/cvx/download/`, store it in a convenient place in your filespace, uncompress it and run the script `cvxsetup.m` that comes with it to set paths correctly.

Linear Least Squares Regression:

Download the `Boston Housing` dataset from the `UCI Machine Learning` repository [1]; this comes in two files: `housing.data`, which contains the data and `housing.names`, which describes the different variables and other uses of the dataset. Load the data into `MATLAB` and normalize the variables as follows:

```
% Load Boston Housing Data from UCI ML Repository
%
load -ascii housing.data;

% Normalize the data, zero mean, unit standard deviation
%
[N, p1] = size(housing);
p = p1-1;
Y = [housing(:,1:p) ones(N,1)];
for j=1:p
    Y(:,j)=Y(:,j)-mean(Y(:,j));
    Y(:,j)=Y(:,j)/std(Y(:,j));
end
f = housing(:,p1);
f = f - mean(f);
f = f/std(f);
```

You can predict the response variable (output variable) f, the house price, from the covariates (input variable) by estimating a linear regression:

```
% Least squares regression as pseudo inverse
%
w = inv(Y'*Y)*Y'*f;
fh = Y*w;
figure(1), clf,
plot(f, fh, 'r.', 'LineWidth', 2), grid on
xlabel('True House Price', 'FontSize', 14)
ylabel('Prediction', 'FontSize', 14)
title(['Linear Regression'], 'FontSize', 14)
```

Split the data into a training set and a test set, estimate the regression model ($\boldsymbol{w}$) on the training set and see how training and test errors differ (lookup the command `randperm`). Implement 10-fold cross validation on the data and quantify an average prediction error and an uncertainty on it.

Regression using the CVX Tool:

The least squares regression you have done in the above section can be implemented as follows in the `cvx` tool:

```
cvx_begin quiet
   variable w1( p+1 );
   minimize norm( Y*w1 - f )
cvx_end
fh1 = Y*w1;
```

Check if the two methods produce the same results.

```
figure(2), clf,
plot(w, w1, 'mx', 'LineWidth', 2);
```

Sparse Regression:

Let us now regularize the regression: $w_2 = \min_{\boldsymbol{w}} |Y\boldsymbol{w} - \boldsymbol{f}| + \gamma |\boldsymbol{w}|_1$. You can implement this as follows:

```
gamma = 8.0;
cvx_begin quiet
      variable w2( p+1 );
      minimize( norm(Y*w2-f) + gamma*norm(w2,1) );
cvx_end
fh2 = Y*w2;
plot(f, fh1, 'co', 'LineWidth', 2),
legend('Regression', 'Sparse Regression');
```

You can find the non-zero coefficients that are not switched off by the regularizer:

```
[iNzero] = find(abs(w2) > 1e-5);
disp('Relevant variables');
disp(iNzero);
```

Find out from `housing.names` which of the variables are selected as relevant to the house price prediction problem. Do they appear more relevant than those that were not selected as relevant?

The amount of regularization is controlled by γ, for which I have selected a convenient value. Write a program to change this parameter over the range $0.01 \rightarrow 40$ in 100 steps and plot a graph of how the number of non-zero coefficients changes with increasing regularization.

- Describe the work you have done as a short report. Upload a *pdf* file **no longer than six pages** using the ECS handin system: `http://handin.ecs.soton.ac.uk`. Please make sure your name and email are included.

References

[1] K. Bache and M. Lichman, "UCI machine learning repository." `http://archive.ics.uci.edu/ml`, 2013.