School of Electronics and Computer Science
University of Southampton

COMP3206/COMP6229 (2015/16): Machine Learning Lab 4

Issue	Thursday, 3 Nov. 2016
Deadline	Thursday 10 Nov. 2016(12:00)

Spend no more than 10 hours on this task. Please work independently.

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
s=getenv('USERNAME');
xlabel('True House Price', 'FontSize', 14)
ylabel('Prediction', 'FontSize', 14)
title(['Linear Regression: ' s], '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.

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 → 40 in 100 steps and plot a graph of how the number of non-zero coefficients changes with increasing regularization.

References

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

Mahesan Niranjan November 2016