Coursework	**Spring 2019**
Statistical Pattern Recognition (COMP136)	**Roni Khardon**
Department of Computer Science, Tufts University	**Rishit Sheth/Xinmeng Li**

Programming Project 2

This assignment is due by Tuesday, March 5, 7:30pm. **Please submit a hardcopy of the report (in class) and both the report and code electronically (via provide).**

Overview: Linear Regression & Model Selection

In this project, we investigate (i) the effect of the regularization parameter in regularized linear regression, and (ii) the quality of two model selection techniques.

On the course web page, you will find 5 datasets for experimentation with linear regression. Each dataset comes in 4 files with the training set in `train-name.csv` the corresponding regression values (outputs) in `trainR-name.csv` and similarly for test set. There will be both artificial data and real data. The files for the artificial data are named as `NumExamples-NumFeatures` for easy identification of dataset characteristics. The artificial data was generated using the regression model and is thus useful to test the algorithms when their assumptions hold. The different artificial datasets have different underlying predictive functions (hidden vector w) so they should not be mixed together.

The performance metric for this assignment will be mean square error

$$\text{MSE} = \frac{1}{N}\sum_i (x_i^\top w - y_i)^2$$

where N is the number of inputs in the corresponding data.

Task 0: Preparation

Start by creating 3 additional training sets from the training dataset 1000-100, using the initial 50, 100, and 150 examples, respectively. Call these 50(1000)-100, 100(1000)-100, and 150(1000)-100. The test set does not need to be modified.

Task 1: Regularized Least Squares

In this task, we use regularized linear regression, i.e., given a dataset, the solution vector w is given by Eq. (3.28) of the textbook.

For each of the 8 datasets (5 original and 3 you created), plot the training set MSE and the test set MSE as a function of the regularization parameter λ (use integer values in the range 0 to 150).

Additionally, for the artificial data, compare these to the true MSEs: 3.78 (for 100-10), 3.78 (for 100-100), and 4.015 (on 1000-100).

In your report, provide the results/plots and discuss them: Why can't the training set be used to select λ? How does the choice of the optimal λ vary with the number of features and number of examples? Consider both the cases where the number of features is fixed and where the number of examples is fixed. How do you explain these variations? You might want to plan your plots so that the answers to these questions are easily visible.

Task 2: Cross Validation

Next, we investigate methods for model selection using just the training set.

In this task, we use cross validation to pick the regularization parameter λ in regularized linear regression. This works as follows:

- Use 10 fold cross validation *on the training set* to pick the value of λ in the same range as above. Cross validation is explained below for the benefit of those who have not seen it before.
- Once the value is chosen, train on the entire training set using this value of λ.
- Calculate the MSE on the test set.

Implement this scheme and apply it to the 8 datasets. In your report, provide the results and discuss them: How do the results compare to the best test set results from Task 1 both in terms of the choice of λ and test set MSE? What is the run time cost of this scheme? How does the quality depend on the number of examples and features?

Task 3: Bayesian Model Selection

In this task, we consider the formulation of Bayesian linear regression with the simple prior $p(w|\alpha) = \mathcal{N}(0, \frac{1}{\alpha}I)$. Recall that the evidence function gives a method to pick the parameters α and β. Referring to textbook, the solution is given in Eqs. (3.91), (3.92), (3.95), where m_N and S_N are given in (3.53) and (3.54). These yield an iterative algorithm for selecting α and β using the training set. We can then calculate the MSE on the test set using the MAP (m_N) for prediction.

Implement this scheme, apply it to the 8 datasets, report results, and discuss as in the previous task.

Task 4: Summary

How do the two model selection methods compare in terms of test set MSE and in terms of run time? What are the important factors affecting performance for each method?

Submission

- The implementation must be written in Matlab or Python.
- All your source code for the assignment should be submitted. Please write clear code with documentation as needed. The various parts of the implementation such as training, prediction, testing, and model selection should be *easily* identifiable. The source code should

  1. Run on *homework.eecs.tufts.edu*.
  2. Run without editing.
  3. Run with a single command (if there is more than one execution command required, include those commands in a single Bash script).
  4. Output the requested results.

  You can assume the data files will be available in the same directory as where the code is executed. Please include a short README file with the code execution command.
- For electronic submission, put all the files into a zip or tar archive, for example `myfile.zip` (you do not need to submit the data we give you). Please do not use another compression format such as RAR. Then submit using `provide comp136 pp2 myfile.zip`.
- Your assignment will be graded based on the clarity and correctness of the code, and presentation and discussion of the results.

Addendum: 10-Fold Cross Validation for Parameter Selection

Cross validation (CV) is the standard method for evaluation in empirical machine learning. It can also be used for parameter selection if we make sure to use the training data only.

To select a value for parameter λ of algorithm A(λ) from a finite set $\{\lambda_1, \ldots, \lambda_L\}$ using training dataset (X, y), we do the following:

1. Split the training data (X, y) into 10 disjoint portions (folds) that are approximately equal in size.
2. For each value of λ in $\{\lambda_1, \ldots, \lambda_L\}$:
   (a) For each i in $1 \ldots 10$
      i. Train $A(\lambda)$ on all portions but i and test on i recording the error on portion i.
   (b) Record the average performance of A(λ) on the 10 folds.
3. Pick the value $\lambda^\star$ from $\{\lambda_1, \ldots, \lambda_L\}$ that resulted in the best average performance.

Note: Using this procedure, the parameter is chosen without knowledge of the test data.