CS 534: Machine Learning
Homework 1

(Due Sep 22th at 11:59 PM on Canvas)

Submission Instructions: The homework should be submitted electronically on Canvas. The code can be done in any language that `Euler` supports.

1. **(10 pts) (Faking) Ridge Regression**

   Show that the ridge regression estimates can be obtained by ordinary least squares regression on an augmented data set. We augment the centered matrix $\mathbf{X}$ with k additional rows $\sqrt{\lambda}\mathbf{I}$ and augment $\mathbf{y}$ with k zeros. The idea is that by introducing artificial data having response value zero, the fitting procedure is forced to shrink the coefficients towards zero.

2. **(7 + 30 + 10 + 3 = 50 pts) Predicting Blog Feedback with SGD**

   Consider the BlogFeedback dataset (`BlogFeedback.zip`), which contains information from the blog post and tries to predict the comments the post received in the next 24 hours relative to a basetime. The dataset is split into three subsets: training data based on blog posts from 2010 and 2011, validation data with blog posts in February 2012, and test data with blog posts from March 2012. There are 280 features for each blog post, which are described in detail on the UCL ML repository, BlogFeedback dataset.

   (a) Run ridge regression (you can use an existing toolbox / implementation) to find an optimal regularization parameter λ that gives you the lowest RMSE on the validation set. You may want to consider trying a range of parameter values (λ) on a logspace (e.g., `numpy.logspace` is a good function to use for this particular purpose). What is the RMSE on the test set?

   (b) Implement stochastic gradient descent for ridge regression using the regularization parameter (λ) from the previous part using a fixed learning rate.

   (c) What is a good learning rate for this dataset? Justify the selection by trying various learning rates and illustrating the objective value ($f_o(x)$) on a graph for a range of epochs (one epoch = one pass through the training data). For your chosen learning rate, what is the RMSE on the test set when trained on the entire dataset?

   (d) Compare and contrast the learned coefficients between the model in part (a) with your model from part (c).

3. **(35 + 5 = 40 pts) Spam classification using Naive Bayes**

   Consider the Ling-Spam email spam dataset (`Spam.zip`) which is based on 960 real email messages from a linguistics mailing list that has been preprocessed. The original dataset can be found here. The dataset is split into two subsets: 700-emails for training and 260 for testing, with each subset containing 50% spam and 50% non-spam messages. The email has been preprocessed in the following ways:

   (a) Stop word removal: Certain common English words like “and”, ”the”, and ”of” that are not meaningful in deciding the status have been removed.

   (b) Lemmatization: Words have been converted to lower case and words that have the same meaning but different endings have been adjusted to have the same form. For example, “include”, “includes”, and “included” will be represented as “include”.

(c) Removal of non-words: Numbers and punctuation have been removed. All white spaces have all been trimmed to a single space character.

The features of the emails (`train-features.txt`) are encoded in a sparse matrix format, where only the non-zero entries are stored. Each row in the feature file encodes a non-zero entry, with the first number denoting the message number, the second number the ID of the word, and the third number the number of occurrences in the message. For example, the line `2 977 2` says that email message 2 has 2 occurrences of the word 977.

(a) Implement and train a multinomial Naive Bayes classifier on the training data. You will want to use the `logsumexp` trick to avoid numerical underflow when calculating the class probability. What is your AUC and classification error on the training set?

(b) Using the model parameters you obtained from training, classify each test document as spam or non-spam. What is your AUC and classification error? How does this compare against the AUC and the classification error from your training set?