Machine Learning & Data Mining
Homework 2

Caltech CS/CNS/EE 155
January 12^{th}, 2016

This set is due 2pm, January 19^{th}, via Moodle. You are free to collaborate on all of the problems, subject to the collaboration policy stated in the syllabus. Please include any code with your submission.

1 Effects of Regularization

For this problem, you are required to implement everything by yourself and submit code.

Question A: In order to prevent over-fitting in the least-squares linear regression problem, we add a regularization penalty term. Can adding the penalty term decrease the training (in-sample) error? Will adding a penalty term always decrease the out-of-sample errors? Please justify your answers.

Question B: ℓ_1 regularization is sometimes favored over ℓ_2 regularization due to its ability to generate a sparse w (more zero weights). In fact, ℓ_0 regularization (using ℓ_0 norm instead of ℓ_1 and ℓ_2 norm) can generate a sparser w, which seems favorable in high-dimensional problems. However, it is rarely used. Could you please explain why?

Implementation of L-2 regularization:

We are going to experiment with linear regression for the Red Wine Quality Rating data set. The data set is uploaded onto Moodle, and you can read more about it here: https://archive.ics.uci.edu/ml/datasets/Wine. Note that the original data set has three classes, but one was removed to make this a binary classification problem. Download the data for training and testing. There are two training data sets wine_training1.txt and wine_training2.txt (100 and 40 data points where the second data set is a complete subset of the first) and one testing data set wine_testing.txt (30 data points). You will use the wine.testing.txt dataset to validate your models.

The data in each data set represents how 12 different factors (last 12 columns) affect the wine type (the first column). Each column of data represents a different factor, and they are all continuous. You can read about their significance on the website provided above. Now, we will use *logistic regression* instead of linear regression, so our error function will be different. When evaluating training error (E_{in}) and testing error (E_{out}) use the logistic error:

$$E = -\sum_{n=1}^{N} \log(p(y_i|\mathbf{x}_i))$$

where $p(y_i = -1|\mathbf{x}_i)$ is defined as:

$$\frac{1}{1 + e^{\mathbf{w}^T\mathbf{x}_i}}$$

and $p(y_i = 1|\mathbf{x}_i)$ is defined as:

$$\frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}_i}}$$

Implement the ℓ_2 regularized logistic regression that minimizes:

$$E = -\sum_{n=1}^{N} \log(p(y_i|\mathbf{x}_i)) + \frac{\lambda}{N} w^T w = -\sum_{n=1}^{N} \log\left(\frac{1}{1 + e^{-y_i\mathbf{w}^T\mathbf{x}_i}}\right) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$$

Train the model with 10 different choices of λ starting with $\lambda_0 = 0.0001$ and increasing by a factor of 5, i.e.

$$\lambda_0 = 0.0001, \lambda_1 = 0.0005, \lambda_2 = 0.0025, ..., \lambda_{15} = 610351.5625$$

You may run into numerical instability issues (overflow or underflow). One way to deal with these issues is by normalizing the input data X. Given a column X_i for a single feature in the training set, you can normalize it by setting $X_{ij} = \frac{X_{ij} - \overline{X_i}}{\sigma(X_i)}$ where $\sigma(X_i)$ is the standard deviation of the column's entries, and $\overline{X_i}$ is the mean of the column's entries. Normalization may change the optimal choice of λ. If you normalize the input data, simply plot the enough choices of λ to see any trends.

Question C: Do the following for both training data sets and attach your plots in the homework submission (Hint: use semi-log plot):

i. Plot the in sample error (E_{in}) versus different λs.

ii. Plot the out of sample error (E_{out}) versus different λs.

iii. Plot the L-2 norm of w versus different λs.

Question D: Considering that the data in wine_training2.txt is a subset of the data in wine_training1.txt, compare errors (training and validation) resulting from training with wine_training1.txt (100 data points) versus wine_training2.txt (40 data points). Please briefly explain the difference.

Question E: Briefly explain the qualitative behavior (i.e., over-fitting and under-fitting) of the training and validation errors with different λs while training with data in wine_training1.txt.

Question F: Briefly explain the qualitative behavior of norm of w with different λs while training with the data in wine_train1.txt.

Question G: If the model were trained with wine_training2.txt, which λ would you choose to train your final model? Why?

2 Lasso (ℓ_1) vs. Ridge (ℓ_2) Regularization

For this problem, you are allowed to use packages in Python, MATLAB, or any other language instead of implementing lasso and ridge regularization.

Many datasets we now encounter in regression problems are very high dimensional. One way to handle this is to encourage the weights to be sparse, and only allow a small number of features to have non-zero weight. The direct way of encouraging a sparse weight vector is use ℓ_0 regularization and penalize the ℓ_0 norm, which is the number of non-zero elements in the vector. However, the ℓ_0 norm is far from smooth, and as a result, it is hard to optimize.

Two related methods are Lasso (ℓ_1) regression and Ridge (ℓ_2) regression. Although both result in shrinkage estimators, only Lasso regression results in sparse weight vectors. This problem compares the behavior of the two methods.

Question A: Let $\mathcal{D}$ be a set of data points, and let $\mathbf{w}$ be a D-dimensional weight vector. Both Lasso and Ridge regression can be formulated as a maximum a posteriori (MAP) estimate of $p(\mathbf{w}|\mathcal{D})$ with different priors $p(\mathbf{w}|\lambda)$, where λ is a parameter controlling the shape of the prior.

i. In the case of Lasso regression, the prior is that the weights are independent and identically distributed (i.i.d.) zero-mean Laplacian random variables

$$p(\mathbf{w}|\lambda) = \prod_{j=1}^{D} \text{Lap}(w_j|0, 1/\lambda) = \prod_{j=1}^{D} \frac{\lambda}{2} e^{-\lambda|w_j|}.$$

Show that under this prior,

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathbf{w}|\mathcal{D})$$

is equivalent to

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left(-\log\, p(\mathcal{D}|\mathbf{w}) + \lambda\|\mathbf{w}\|_1\right).$$

ii. In the case of Ridge regression, the prior is that the weights are i.i.d. zero-mean Normal random variables

$$p(\mathbf{w}|\lambda) = \prod_{j=1}^{D} \mathcal{N}(w_j|0, 1/2\lambda) = \prod_{j=1}^{D} \sqrt{\frac{\lambda}{\pi}} e^{-\lambda w_j^2}.$$

Show that under this prior,

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathbf{w}|\mathcal{D})$$

is equivalent to

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left(-\log\, p(\mathcal{D}|\mathbf{w}) + \lambda\|\mathbf{w}\|_2^2\right).$$

iii. Suppose that $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I})$ and $\mathcal{D}$ contains $\mathbf{X}$ and $\mathbf{y}$. Show that

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathcal{D}|\mathbf{w})$$

is equivalent to

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$$

This means that least-squares linear regression is a maximum likelihood estimator when there is white Gaussian noise.

Question B: This section compares the behavior of Lasso and Ridge regression on a synthetic dataset. The dataset consists 1000 independent samples of a 9-dimensional feature vector $\mathbf{x} = [x_1, \ldots, x_9]$ drawn from a uniform distribution on the interval $[-1, 1]$, along with the response

$$y = -4x_1 - 3x_2 - 2x_3 - 1x_4 + 0x_5 + 1x_6 + 2x_7 + 3x_8 + 4x_9 + n = \mathbf{w}^T\mathbf{x} + n,$$

where n is a standard Normal random variable. The file question2data.txt consists of 1000 lines of 10 tab-delimited values. The first 9 columns represent $x_1, \ldots, x_9$, and the last column represents y. Each row consists of one sample. Using Python and numpy, you may load the data as such

```
>>> data = numpy.loadtxt('./data/question2data.txt', delimiter=',')
>>> X = data[:, 0:9]
>>> Y = data[:, 9]
```

Using MATLAB, you may load the data as such

```
>>> data = dlmread('./data/question2data.txt', ',');
>>> X = data(:, 1:9);
>>> Y = data(:, 10);
```

You may also generate the dataset using the process specified above. This problem explores the behavior of the estimated weights as the strength of the regularization (λ) varies.

i. Estimate the weights $\mathbf{w}$ using linear regression with Lasso regularization for various choices of λ. For each of the weights, plot the weight as a function of λ (start with $\lambda = 0$ and increase λ until all weights are small). Using a linear scale for λ will allow the plot to be easily interpreted. Note, in MATLAB you should *not* standardize the variables:

```
>>> lasso(X, Y, 'Lambda', 0:0.01:3, 'standardize',false);
```

In Python, you may use sklearn:

```
from sklearn.linear_model import Lasso
clf = Lasso(alpha=Lambda)
```

Then, Python's sklearn and MATLAB's lasso plots should be identical.

ii. Estimate the weights $\mathbf{w}$ using linear regression with Ridge regularization for various choices of λ. For each of the weights, plot the weight as a function of λ (start with $\lambda = 0$ and increase λ until all weights are small). Note, in MATLAB you should use:

```
>>> Xn = [ones(length(Y), 1), X];
>>> coef = (Xn'*Xn +  diag([0; Lambda(i) * ones(9, 1)]))\ (Xn'*Y);
```

You may also use

```
>>> coef = ridge(Y, X, Lambda(i), 0);
```

but this will produce a scaled version of the plot generated by python's sklearn, due to differences in normalization between MATLAB's "ridge" and sklearn's "Lasso".

In Python, you may use:

```
>>> from sklearn.linear_model import Ridge
>>> clf = Ridge(alpha=Lambda)
```

iii. As regularization parameter increases, what happens to the number of estimated weights that are exactly zero with Lasso regression? What happens to the number of estimated weights that are exactly zero with Ridge regression?

Question C: For general choices of $p(D|\mathbf{w})$, an analytic solution for regularized linear regression may not exist. However, when $p(D|\mathbf{w})$ has a standard normal distribution (corresponding to linear regression), an analytic solution exists for 1-dimensional Lasso regression and for Ridge regression in all dimensions.

i. Solve for $\arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{w}^T\mathbf{x}\|^2 + \lambda\|\mathbf{w}\|_1$ in the case of a 1-dimensional feature space. This is linear regression with Lasso regression.

ii. Suppose that when $\lambda = 0$, $w_1 \neq 0$. Does there exist a value for λ such that $w_1 = 0$? If so, what is the smallest such value?

iii. Solve for $\arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{w}^T\mathbf{x}\|^2 + \lambda\|\mathbf{w}\|_2^2$ for an arbitrary number of dimensions. This is linear regression with Ridge regression.

iv. Suppose that when $\lambda = 0$, $w_i \neq 0$. Does there exist a value for $\lambda > 0$ such that $w_i = 0$? If so, what is the smallest such value?