HOMEWORK 1 - V2

STA414/STA2104 WINTER 2019

University of Toronto

1. Probability and Calculus.

1.1. *Variance and covariance - 15 pts.* Let X, Y be two independent random vectors in $\mathbb{R}^m$.

(a) Show that their covariance is zero.
(b) For a constant matrix $A \in \mathbb{R}^{m \times m}$, show the following two properties:

$$\mathbb{E}(X + AY) = \mathbb{E}(X) + A\mathbb{E}(Y)$$
$$\mathrm{Var}(X + AY) = \mathrm{Var}(X) + A\mathrm{Var}(Y)A^T$$

(c) Using part (b), show that if $X \sim \mathcal{N}(\mu, \Sigma)$, then $AX \sim \mathcal{N}(A\mu, A\Sigma A^T)$. Here, you may use the fact that linear transformation of a Gaussian random vector is again Gaussian.

1.2. *Densities - 10 pts.* Answer the following questions:

(a) Can a probability density function (pdf) ever take values greater than 1?
(b) Let X be a univariate normally distributed random variable with mean 0 and variance 1/100. What is the pdf of X?
(c) What is the value of this pdf at 0?
(d) What is the probability that $X = 0$?

1.3. *Calculus - 10 pts.* Let $x, y \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times m}$. In vector notation, what is

(a) the gradient with respect to x of $x^T y$?
(b) the gradient with respect to x of $x^T x$?
(c) the gradient with respect to x of $x^T Ax$?
(d) the gradient with respect to x of Ax?

2. Regression.

2.1. *Linear regression - 15 pts.* Suppose that $X \in \mathbb{R}^{n \times m}$ with $n \geq m$ and $Y \in \mathbb{R}^n$, and that $Y|X, \beta \sim \mathcal{N}(X\beta, \sigma^2 I)$. We know that the maximum likelihood estimate $\hat{\beta}$ of β is given by

$$\hat{\beta} = (X^T X)^{-1} X^T Y.$$

(a) Find the distribution of $\hat{\beta}$, its expectation and covariance matrix.
(b) Write the log-likelihood implied by the model above, and compute its gradient w.r.t. β.
(c) Assuming that σ^2 is known, what is the probability that an individual parameter $\hat{\beta}_i$ is in the ϵ-neighborhood of the corresponding entry of the true parameter β_i, i.e. $\mathbb{P}(|\hat{\beta}_i - \beta_i| \leq \epsilon)$? (Hint: Use Gaussian CDF $\Phi(t)$.)

2.2. *Ridge regression and MAP - 20 pts.* Suppose that we have $Y|X, \beta \sim \mathcal{N}(X\beta, \sigma^2 I)$ and we place a normal prior on β, i.e., $\beta \sim \mathcal{N}(0, \tau^2 I)$.

(a) Show that the MAP estimate of β given Y in this context is

$$\hat{\beta}_{MAP} = (X^T X + \lambda I)^{-1} X^T Y$$

where $\lambda = \sigma^2/\tau^2$.

(b) Show that ridge regression is equivalent to adding m additional rows to X where the j-th additional row has its j-th entry equal to $\sqrt{\lambda}$ and all other entries equal to zero, adding m corresponding additional entries to Y that are all 0, and and then computing the maximum likelihood estimate of β using the modified X and Y.

2.3. *Cross validation - 30 pts.* In this problem, you will write a function that performs K-fold cross validation procedure to tune the penalty parameter λ in Ridge regression. Your `cross_validation` function will rely on 6 short functions which are defined below along with their variables.

- `data` is a variable and refers to a (y, X) pair (can be test, training, or validation) where y is the target (response) vector, and X is the feature matrix.
- `model` is a variable and refers to the coefficients of the trained model, i.e. $\hat{\beta}_\lambda$.
- `data_shf = shuffle_data(data)` is a function and takes `data` as an argument and returns its randomly permuted version along the samples. Here, we are considering a uniformly random permutation of the training data. Note that y and X need to be permuted the same way preserving the target-feature pairs.
- `data_fold, data_rest = split_data(data, num_folds, fold)` is a function that takes `data`, number of partitions as `num_folds` and the selected partition `fold` as its arguments and returns the selected partition (block) `fold` as `data_fold`, and the remaining data as `data_rest`. If we consider 5-fold cross validation, `num_folds`=5, and your function splits the data into 5 blocks and returns the block `fold` ($\in \{1, 2, 3, 4, 5\}$) as the validation fold and the remaining 4 blocks as `data_rest`. Note that `data_rest` $\cup$ `data_fold` $=$ `data`, and `data_rest` $\cap$ `data_fold` $= \emptyset$.
- `model = train_model(data, lambd)` is a function that takes `data` and `lambd` as its arguments, and returns the coefficients of ridge regression with penalty level λ. For simplicity, you may ignore the intercept and use the expression in question 2.2.
- `predictions = predict(data, model)` is a function that takes `data` and `model` as its arguments, and returns the `predictions` based on `data` and `model`.
- `error = loss(data, model)` is a function which takes `data` and `model` as its arguments and returns the average squared `error` loss based on `model`. This means if `data` is composed of $y \in \mathbb{R}^n$ and $X \in \mathbb{R}^{n \times p}$, and model is $\hat{\beta}$, then the return value is $\|y - X\hat{\beta}\|^2/n$.
- `cv_error = cross_validation(data, num_folds, lambd_seq)` is a function that takes the training `data`, number of folds `num_folds`, and a sequence of λ's as `lambd_seq` as its arguments and returns the cross validation error across all λ's. Take `lambd_seq` as evenly spaced 50 numbers over the interval (0.02, 1.5). This means `cv_error` will be a vector of 50 errors corresponding to the values of `lambd_seq`. Your function will look like:

```
data = shuffle_data(data)
for i = 1,2,...,length(lambd_seq)
```

```
        lambd = lambd_seq(i)
        cv_loss_lmd = 0.
        for fold = 1,2, ...,num_folds
             val_cv, train_cv = split_data(data, num_folds, fold)
             model = train_model(train_cv, lambd)
             cv_loss_lmd += loss(val_cv, model)
        cv_error(i) = cv_loss_lmd / num_folds
return cv_error
```

(a) Download the dataset from the course webpage `dataset.mat` and place it in your working directory, or note its location `file_path`. For example, file path could be `/Users/yourname/Desktop/`

- In `R`:

```
library(R.matlab)
dataset = readMat('file_path/dataset.mat')
data.train.X = dataset$data.train.X
data.train.y = dataset$data.train.y[1,]
data.test.X = dataset$data.test.X
data.test.y = dataset$data.test.y[1,]
```

- In `Python`:

```
import scipy.io as sio
dataset = sio.loadmat('file_path/dataset.mat')
data_train_X = dataset['data_train_X']
data_train_y = dataset['data_train_y'][0]
data_test_X = dataset['data_test_X']
data_test_y = dataset['data_test_y'][0]
```

(b) Write the above 6 functions, and identify the correct order and arguments to do cross validation.

(c) Find the training and test errors corresponding to each λ in `lambd_seq`. This part does not use the `cross_validation` function but you may find the other functions helpful.

(d) Plot training error, test error, and 5-fold and 10-fold cross validation errors on the same plot for each value in `lambd_seq`. What is the value of λ proposed by your cross validation procedure? Comment on the shapes of the error curves.