Homework 3
Quantile Regression, Gaussian Processes Kernels

CMU 10-715: Machine Learning (Fall 2015)
http://www.cs.cmu.edu/~bapoczos/Classes/ML10715_2015Fall/
OUT: Oct 19, 2015
DUE: Nov 2, 2015, 10:20 AM

Guidelines

- The homework is due at 10:20 am on Monday November 2, 2015. Each student will given two late days that can be spent on any homeworks, but at most one late day per homework. Once you have used up your late days for the term, late homework submissions will receive no credit.
- Submit both a paper copy and an electronic copy through through the submission website: https://autolab.cs.cmu.edu/courses/10715-f15. You can sign in using your Andrew credentials. You should make sure to edit your account information and choose a nickname/handle. This handle will be used to display your results for any competition style questions on the class leaderboard.
- Some questions will be *autograded*. Please make sure to carefully follow the submission instructions for these questions.
- We recommend that you typeset your solutions using software such as LaTeX.If you write, ensure your handwriting is clear and legible. The TAs will not invest undue effort to decrypt bad handwriting.
- Programming guidelines:
  - **Octave:** You must write submitted code in Octave. Octave is a free scientific programming language, with syntax similar to that of MATLAB. Installation instructions can be found on the Octave website. (You can develop your code in MATLAB if you prefer, but you *must* test it in Octave before submitting, or it may fail in the autograder.)
  - **Autograding:** This problem is autograded using the CMU Autolab system. The code which you write will be executed remotely against a suite of tests, and the results used to automatically assign you a grade. To make sure your code executes correctly on our servers, you should avoid using libraries which are not present in the *basic* Octave install.
  - **Submission Instructions:** For each programming question you will be given a function signature. You will be asked to write a single Octave function which satisfies the signature. In the code handout linked above, we have provided you with a single folder containing stubs for each of the functions you need to complete. *Do not modify the structure of this directory or rename these files.* Complete each of these functions, then compress this directory *as a tar file* and submit to Autolab online. You may submit code as many times as you like.

    When you download the files, you should confirm that the autograder is functioning correctly by compressing and submitting the directory of stubs provided. This should result in a grade of zero for all questions.
  - **SUBMISSION CHECKLIST**
    - Submission executes on our machines in less than 3 minutes.
    - Submission is smaller than 5000K.
    - Submission is a `.tar` file.
    - Submission returns matrices of the *exact* dimension specified.

As usual, for any programming problems we will use the following conventions:

- N is the number of datapoints, D is the dimension of each input.

- XTrain is an $N \times D$ matrix of the input data, where row i is the features for example i.
- yTrain is an $N \times 1$ vector of the input data, where the ith component is the ith output.
- XTest is an $M \times D$ matrix of the input data, where row i is the features for example i.
- yTrain is an $M \times 1$ vector of the input data, where the ith component is the ith output.

1 Quantile Regression [Eric; 35 pts]

In this section, you will derive the dual of the quantile regression problem and implement a solver.

1.1 Quantile Regression

1. (6pts) By now you may be used to minimizing problems with respect to squared error loss. Let's instead define the following loss:

$$\rho_\tau(z) = z(\tau - I(z < 0)) = \begin{cases} z(\tau - 1) & \text{if } z < 0 \\ z\tau & \text{if } z \geq 0 \end{cases}$$

where $\tau \in (0, 1)$ is called the τth quantile, and $I(z < 0)$ is the indicator function that is 1 if $z < 0$ and 0 otherwise. Show that

$$\text{argmin}_w \sum_i \rho_\tau(y_i - w) = y_\tau$$

where y_τ is an observation sitting at the τth top percentile of the observations (specifically, this means that y_τ is at least exactly τ percent of the observations).

2. (2pts) When $\tau = 0.5$, this loss function has a well known name in statistics. What is it?

3. (6pts) Let $\{x_i\}_{i=1,\ldots,N}$ be points in $\mathbb{R}^K$ with outputs $\{y_i\}_{i=1\ldots n}$ in $\mathbb{R}$. Let $X = (x_1, \ldots, x_N)$. We define the regression quantile as

$$\hat{\beta}(\tau) = \underset{\beta \in \mathbb{R}^K}{\text{argmin}} \sum_{i=1}^{N} \rho_\tau(y_i - x_i^T \beta)$$

Prove that the solution of this problem is equivalent to the solution of the following linear program. Hint: split the problem into positive and negative parts.

$$\underset{\beta \in \mathbb{R}^K, u, v \in \mathbb{R}^N}{\text{argmin}} \; u^T 1\tau + v^T 1(1 - \tau), \quad \text{subject to } X^T\beta - y + u - v = 0, u, v \geq 0$$

4. (6pts) Show that the dual of the above linear program is

$$\max_z y^T z, \quad \text{subject to } Xz = (1 - \tau)X1, z \in [0, 1]^n$$

5. (4pts) What does the value of z_i in the dual problem tell us about $y_i - x_i^T\beta$ in the primal? Specifically, using the KKT conditions, if $z_i = 0$ then what can you say about $y_i - x_i^T\beta$? If $z = 1$? If $z \in (0, 1)$?

6. We have generated a synthetic dataset in quantile.mat. For this problem you will use quantile regression to get the quantile estimates for this dataset.

   You should implement quantile regression by solving the primal LP. You may use any linear programming solver to do so. For example, CVXOPT (http://cvxopt.org/) is a powerful solver for general convex problems. Alternatively, you can use the glpk function in Octave (https://www.gnu.org/software/octave/doc/interpreter/Linear-Programming.html) or linprog in Matlab (http://www.mathworks.com/help/optim/ug/linprog.html).

   You may need to reformulate your problem into a canonical form accepted by the solver. Be sure to account for a non-zero intercept term. Submit the following items in your writeup:

- (6pts) First, plot a scatterplot of the data in XTrain,yTrain. Then, plot three quantile regression lines on top of the scatterplot at the following quantiles: $\tau = 0.25, 0.50, 0.75$.
- (3pts) Report the β values for each value of τ.
- (2pts) Attach your code for this problem.

2 Gaussian Processes and Hyperparameter Tuning [Eric; 25pts]

2.1 Lemma from Class

1. (5pts) First, let's verify a lemma from class. Let X, y be n examples of training data and labels and let X^*, y^* be m examples of test data and labels. Let $0_n, 0_m$ denote zero vectors of length n, m respectively, and let k be some kernel function. Suppose that

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \mathcal{N}\begin{bmatrix} y \\ y^* \end{bmatrix}\left(\begin{bmatrix} 0_n \\ 0_m \end{bmatrix}, \begin{bmatrix} k(X,X) & k(X,X^*) \\ k(X^*,X) & k(X^*,X^*) \end{bmatrix}\right)$$

Show that the posterior distribution is

$$P(y^*|X^*, X, y) = \mathcal{N}_{y^*}(\mu, \Sigma)$$

where $\mu = k(X^*,X)k(X,X)^{-1}y$ and $\Sigma = k(X^*,X^*) - k(X^*,X)k(X,X)^{-1}k(X,X^*)$. Note: For this question, you may assume that the conditional distribution is of a Normal form, however you must derive the mean and variance. Calculating the pdf of the posterior is a long and painful process, and is not recommended.

2.2 GP Regression

For this problem, you will implement a basic Gaussian Process Regression. We will be using the standard radial basis kernel:

$$K(x_i, x_j) = \sigma \exp\left(\frac{-||x_i - x_j||_2^2}{2h^2}\right)$$

where σ, h are known as the scale and bandwith parameters.

For additional help, better performance, and numerical stability, we refer you to chapter 2 of Rasmussen and Williams (http://www.gaussianprocess.org/gpml/chapters/RW2.pdf).

We will test your implementation on the Concrete Compressive Strength dataset from the UCI repository. The strength of concrete is predicted from 8 features consisting of the ingredients that make up the concrete composition and its age. We have given you this dataset as an octave mat file.

We will use the following conventions for this problem:

- X1, X2 are $n_1 \times D$ and $n_2 \times D$ matrices of the input data. Note that n_1 is not necesarily equal to n_2. Each row consists of the features of a particular example.
- K is a $n_1 \times n_2$ kernel matrix for X1,X2.
- GPMean is a $N \times 1$ vector containing the predicted mean values of the GP at XTest.
- GPVariance is a $N \times N$ matrix containing the predicted covariance matrix of the GP at XTest.
- logml is a scalar value containing the log marginal likelihood of the data given the parameters.
- sigma is a the scale parameter described above, and sigmas is a $P_1 \times 1$ vector of potential parameters.
- h is a the bandwith parameter described above, and hs is a $P_2 \times 1$ vector of potential parameters.
- gamma is the noise parameter for the Gaussian Process. Specifically,

$$\text{cov}(y) = K(X,X) + \gamma I$$

1. (3pts) Implement [K] = RBFKernel(X1, X2, sigma, h), which takes as input two matrices of examples with hyperparameters sigma, h, and outputs the kernel matrix where $K_{i,j} = k(\mathsf{X1}_i, \mathsf{X2}_i)$, where k is the RBF function described above. Bonus: do this without any for loops.

2. (7pts) Implement [GPMean, GPVariance] = GPRegression(XTrain, yTrain, XTest, gamma, sigma, h), which carries out the Gaussian Process regression and returns the estimated mean and variances for the variables in XTest. See page 19 of chapter 2 in Rasmussen and Williams for help on making this computationally efficient and numerically stable.

3. (3pts) Now, we need to find hyperparameters for the Gaussian Process. One reasonable method for Gaussian processes is to choose parameters that maximizes the log marginal likelihood. First implement [logml] = LogMarginalLikelihood (XTrain, yTrain, gamma, sigma,h) which computes the log marginal likelihood of the training data given the parameters.

4. (3pts) Implement [gamma, h,sigma] = HyperParameters (XTrain, yTrain, hs, sigmas), which does a grid search across the parameters in hs,sigmas and returns the combination that minimizes the log marginal likelihood. Also set gamma to be $0.01 \cdot \sigma_y$ where σ_y is the standard deviation of the training example outputs.

5. (4pts) Run your Gaussian process regression method on the dataset provided in concrete.mat. Compare and report your results with a naive mean prediction. Get your hyperparameters by using your implemented HyperParameters functions and searching over the space of hs = logspace(-1,1,10)' * norm(std(XTrain)) and sigmas = logspace(-1,1,10)' * std(yTrain).

3 Kernel two sample-test [Fish; 40 pts]

Suppose you are collecting data on the expression level of gene No. 10715 after inserting a secret drug into mice liver. There are two labs, Lab A and Lab B, that run the experiments for you and send you the results. Of course you would hope that the environment and quality of each lab would not cause a difference in the data between the two locations. To make it simpler, assume the data from Lab A is i.i.d drawn from a disribution p, and the data from lab B are i.i.d. drawn from a distribution q. The question you would like to answer is: given data $X = \{x_1, x_2, \cdots, x_m\}$ collected from lab A and $Y = \{y_1, y_2, \cdots, y_m\}$ collected from lab B, is $p = q$?

1. (10pts) Let $\mathcal{X}$ be a sample space, and consider two distributions p and q. $p = q$ if and only if $\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{y\sim q}[f(y)]$ for all $f \in \mathcal{F}(\mathcal{X})$ where $\mathcal{F}(\mathcal{X})$ is the space of bounded continuous functions from $\mathcal{X} \to \mathbb{R}$. Using this theorem, we define the maximum mean discrepency as

$$\mathrm{MMD}[\mathcal{F}, p, q] := \sup_{f\in\mathcal{F}} \left(\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)]\right). \tag{1}$$

   To answer the question of whether $p = q$, if $\mathrm{MMD}[\mathcal{F}, p, q] = 0$, then we have $p = q$. Write the empirical version of this MMD statement that we can estimate with a dataset X, Y from the two distributions and all the functions in some $\mathcal{F}$.

2. (10pts) The issue with the estimate from question 2 is that we need to find a sufficiently large function class to identify p and q, which is not practical. One way to solve this problem is to kernelize the function to implicitly project the data into a potentially infinite space. More importantly, using a kernel allows us to use the special properties for functions in a Reproducing Kernel Hilbert Space (RKHS): $\mathcal{H}$ is a $RKHS$ if there exists a feature mapping ϕ from space $\mathcal{X}$ to $\mathbb{R}$ such that, for all $x \in \mathcal{X}$,

$$f(x) = \langle f, \phi(x)\rangle_{\mathcal{H}} \tag{2}$$

   for every $f \in \mathcal{H}$. The subscript for the inner product indicates that the inner product is done in the RKHS instead of our sample space. Note that here f refers to the function as an object (you can imagine it as an vector in the RKHS), and $f(x) \in \mathbb{R}^d \to \mathbb{R}$ is defined over $\mathcal{X}$.

Replace $f(x)$ in (1) with the inner product in (2), and set $\mathcal{F}$ to be a unit ball in a RKHS:

$$\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \leq 1, \text{ where } \|f\|_{\mathcal{H}} := \sqrt{\langle f, f \rangle_{\mathcal{H}}}\}$$

Derive an upper bound for $\text{MMD}^2[\mathcal{F}, p, q]$ using $\mathbb{E}_{x\sim p}[\phi(x)]$ and $\mathbb{E}_{y\sim q}[\phi(y)]$.

3. (10pts) Replace $\mathbb{E}_{x\sim p}[\phi(x)]$ and $\mathbb{E}_{y\sim q}[\phi(y)]$ with its empirical estimates to get the kernel method of estimating MMD.

4. (10pts) We have provided a dataset containing two vectors drawn from some mystery distributions p and q in twosample.mat. Use the the RBF kernel to test whether the two vectors of variables have the same distribution. You can use your RBFKernel function that you wrote in question 2.1.1 with parameters $h = 10, 1, 0.1$ and $\sigma = 1$ to calculate the MMD. Use the following threshold: if MMD is less than 0.01, we say they are the same distribution, otherwise they are different.

   Report in your writeup the calculated empirical MMD and your corresponding conclusion.

4 Saddle Points in optimization[Fish; 14 pts] (Bonus)

Often we solve constrained optimization problem by first transforming it into a non-constrained optimization problem. The most common way to conduct such transformation is to introduce Lagrange multipliers and construct a dual problem for the primal problem. Before solving the dual problem, one question we would like to answer is: *Is the optimal value for the dual problem equal to the primal problem?*
Consider the convex optimization problem:

$$\min f(x) \tag{3}$$
$$\text{s.t. } g_i(x) \leq 0, \quad \forall i \in [m], \tag{4}$$
$$f_i(x) = 0 \quad \forall i \in [k], \tag{5}$$

where $f_1, f_2, \cdots, f_k$ are affine functions and $f, g_1, \cdots, g_m$ are convex functions. In this question, we are going to prove that for $x^* \in \mathbb{R}$, if there exists Lagrange Multipliers $\lambda_i^* \geq 0$ for $i \in [m]$ such that (x^*, λ^*) is a saddle point of Lagrange function $L(x, \lambda)$, then x^* is the optimal solution for the primal problem. A point (x^*, λ^*) is said to be a saddle point of function $L(x, \lambda)$ if

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*) \quad \forall x \in \mathbb{R}^d, \lambda \in R_+^m \times \mathrm{R}^k. \tag{6}$$

1. (2pts) Introduce Lagrange Multipliers, $\lambda_1, \lambda_2, \cdots, \lambda_{k+m}$, and write out the Lagrange function for the primal problem.

2. (2pts) Show that the infimum

$$\tilde{L}(\lambda) = \inf_x L(x, \lambda) \tag{7}$$

   of the Lagrange function in $x \in X$ is a lower bound for the optimal primal value $f(x^*)$. Also prove that

$$\sup_{\lambda_1, \lambda_2, \lambda_m \geq 0} L(\lambda) \tag{8}$$

   is also a lower bound for the optimal primal value $f(x^*)$.

3. (2pts) If (x^*, λ^*) is a saddle point of the function $L(x, \lambda)$. Prove that the left half of the saddle point conditions implies $f_i(x^*) = 0$ for $i \in [k]$ and $\sum_{i=1}^{m} \lambda_i^* g_i(x^*) = 0$, so we can conclude that $f(x^*) = L(x^*, \lambda^*)$.

4. (2pts) Complete the proof by saying the right half of the saddle point condition implies x^* is the optimum solution to the primal.

5. (2pts) The other direction of the saddle point theory says that if x^* is a solution for the primal problem and the primal problem satisfies Slater C.Q., then there is a $\lambda^* \in R_+^m \times \mathrm{R}^k$ such that (x^*, λ^*) is a saddle point of $L(x, \lambda)$. We say if a problem statisfies Slater C.Q., then there is a λ^* such that (x^*, λ^*) satisfies KKT conditions. Write out the KKT conditions for the optimization problem.

6. (2pts) Use Primal feasibility, dual feasibility and complementary slackness to show the left half of the saddle point conditions.

7. (2pts) Use dual feasibility to show the right half of the saddle point condition is a convex function in x, so the stationary condition in KKT implies that the right half of the saddle point condition should be satisfied. (Hint: Use the convexity properties we had proved in HW1.)