COMS W4721: Machine Learning for Data Science

Columbia University, Spring 2017

Homework 3: Due March 26, 2017 by 11:59pm

Please read these instructions to ensure you receive full credit on your homework. Submit the written portion of your homework as a *single* PDF file through Courseworks (less than 5MB). In addition to your PDF write-up, submit all code written by you in their original extensions through Courseworks (e.g., .m, .r, .py, .c). Any coding language is acceptable, but do not submit notebooks, do not wrap your files in .rar, .zip, .tar and do not submit your write-up in .doc or other file type. Your grade will be based on the contents of one PDF file and the original source code. Additional files will be ignored. We will not run your code, so everything you are asked to show should be put in the PDF file. Show all work for full credit.

Late submission policy: Late homeworks will have 0.1% deducted from the final grade for each minute late. *Your homework submission time will be based on the time of your **last** submission to Courseworks. I will not revert to an earlier submission!* Therefore, do not re-submit after midnight on the due date unless you are confident the new submission is significantly better to overcompensate for the points lost. Submission time is non-negotiable and will be based on the time you submitted your last file to Courseworks. The number of points deducted will be rounded to the nearest integer.

Problem 1 (Gaussian process coding) – 30 points

In this problem you will implement the Gaussian process model for regression. You will use the same data used for homework 1 to do this, which is again provided in the data zip file for this homework. Recall that the Gaussian process treats a set of N observations $(x_1, y_1), \ldots, (x_N, y_N)$, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, as being generated from a multivariate Gaussian distribution as follows,

$$y \sim Normal(0, \sigma^2 I + K), \quad K_{ij} = K(x_i, x_j) \quad \Big(\text{use: } \exp\Big\{-\frac{1}{b}\|x_i - x_j\|^2\Big\}\Big).$$

Here, y is an N-dimensional vector of outputs and K is an $N \times N$ kernel matrix. For this problem use the Gaussian kernel indicated above. In the lecture slides, we discuss making predictions for a new y' given x', which was Gaussian with mean $\mu(x')$ and variance $\Sigma(x')$. The equations are shown in the slides.

There are two parameters that need to be set for this model as given above, σ^2 and b.

a) Write code to implement the Gaussian process and to make predictions on test data.

b) For $b \in \{5, 7, 9, 11, 13, 15\}$ and $\sigma^2 \in \{.1, .2, .3, .4, .5, .6, .7, .8, .9, 1\}$—so 60 total pairs (b, σ^2)—calculate the RMSE on the 42 test points as you did in the first homework. Use the mean of the Gaussian process at the test point as your prediction. Show your results in a table.

c) Which value was the best and how does this compare with the first homework? What might be a drawback of the approach in this homework (as given) compared with homework 1?

d) To better understand what the Gaussian process is doing through visualization, re-run the algorithm by using *only* the 4th dimension of x_i (car weight). Set $b = 5$ and $\sigma^2 = 2$. Show a scatter plot of the data ($x[4]$ versus y for each point). Also, plot as a solid line the predictive mean of the Gaussian process at each point *in the training set*. You can think of this problem as asking you to create a test set by duplicating $x_i[4]$ for each i in the training set and then to predict that test set.

Problem 2 (Boosting coding) – 30 points

In this problem you will implement boosting for the "least squares" (LS) classifier that we briefly discussed in Lecture 8. Recall that this "classifier" performed least squares linear regression treating the ± 1 labels as real-valued responses. Also recall that we criticized this classifier as being "weak," without using that word, and so boosting this classifier can be a good illustration of the method (even though it performs well on the data set you will be using).

Using the training data provided, implement boosting for the LS classifier. You should use the bootstrap method as discussed in the slides to do this, where each bootstrap set $\mathcal{B}_t$ is the size of the training set. Recall that if your error $\epsilon_t > 0.5$, you can simply change the sign of the regression vector w (including the intercept) and recalculate the error.

Information about the data used for this problem can be found here:

`https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+`

but you must use the data provided on Courseworks. Note that the intercept dimension hasn't been included in the features provided, so you should add a dimension equal to 1.

a) Run your boosted LS classifier for $T = 1500$ rounds. In the same plot, show the training and testing error of $f_{boost}^{(t)}(\cdot)$ for $t = 1, \ldots, T$.

b) In a separate plot, show the upper bound on the training error as a function of t. You will need to use ϵ_t to do this. This upper bound is given in the slides for Lecture 13.

c) Plot a histogram of the total number of times each training data point was selected by the bootstrap method across all rounds. In other words, sum the histograms of all $\mathcal{B}_t$.

d) In two separate plots, show ϵ_t and α_t as a function of t.