STA 414/2104, Spring 2014 — Assignment #1

Due at the start of class on March 4. Please hand it in on 8 1/2 by 11 inch paper, stapled in the upper left, with no other packaging.

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. In particular, you should not leave any discussion with someone else with any written notes (either paper or electronic).

In this assignment, you will apply Gaussian process regression models to a dataset on median housing prices in various small survey regions of the United States, and compare how well the models predict test cases for different forms of the covariance function, when hyperparameters of the covariance function are set by maximizing marginal likelihood, by cross validation, or by averaging over their posterior distribution.

You will need to write some R functions to do Gaussian process modeling, for which you can use the week 6 demo functions on the course web page as a starting point. You will also need to write a script to apply these functions to the dataset in the ways described below. Some hints on using R for this assignment will be on the course web page soon.

You will hand in the R functions and script you used, the results of applying them to the data, and a discussion of these results.

The dataset. The data is from the 1990 U.S. census. It was originally processed by Rafal Kustra, and can be found at http://www.cs.utoronto.ca/ delve/data/census-house/desc.html

Each training or test case is for a small survey region, in which the following covariates (input variables) are known:

1. logarithm of the total number of households
2. fraction Asian or Pacific Islander
3. fraction 25 to 64 years old
4. fraction of households with 2 or more persons that are family households
5. fraction of households with black householder
6. fraction of vacant housing units for sale only
7. average number of rooms in an owner-occupied housing unit
8. fraction of vacant-for-sale housing units vacant more then six months

These covariates are selected (somewhat arbitrarily) from a much larger set that are available.

The objective is to predict the logarithm of the median price of a housing unit for the survey region (the response variable). In the original data, median prices less than 15,000 were all set to 14,999, and median prices over 500,000 were all set to 500,001. For this assignment, I have instead filled in these extreme values randomly so that the distribution looks reasonable (but the relationship to the covariates will not be entirely correct). The proper way to treat this issue is to consider these "censored" values as unknown, and sample from their posterior distribution, but we haven't covered how to do that in this course.

From the larger set of cases available, I have randomly selected two training sets of 250 cases and one test set of 2500 cases, which are available from the course web page (separate files for the covariates, x, and the response variable, y, with each case being on a different line, with no header line). The two training sets are to be processed independently (not using any information from one set when processing the other). The y values for the test cases are to be looked at only after predictions have been made for these test cases, to see how good the predictions were.

The ordering of cases within each training set is random (so you needn't randomly reorder them when selecting subsets for cross validation).

Fitting linear models with lm. You should start by fitting a simple linear model to the data in each training set using the standard R "lm" function, modeling the response in terms of the eight covariates above. You should then see how good, in terms of squared error, the fitted regression coefficients are at predicting the response variable (log median price) in the test cases. You should make predictions and compute squared errors with R code you write yourself (not using R's "predict" function).

You should also fit linear models with "lm" using the eight covariates and the squares of these covariates (16 predictor variables in all), and using the eight covariates plus their squares and their cubes (24 predictor variables in all). You should see what the average squared error on test cases is for these models as well, for both training sets.

Fitting a Gaussian process model that mimics a linear model. You should next apply a Gaussian process model that should behave almost exactly the same as the model fit with "lm". For this Gaussian process model, the noise-free covariance function should be

$$K(x, x') \;=\; 100^2 \;+\; 100^2 \sum_{i=1}^{8} x_i x'_i$$

This corresponds to giving independent $N(0, 100^2)$ priors to the regression coefficients in a simple linear model. These priors are wide enough that they should have almost no effect, giving results almost the same as fitting a linear model by least squares (ie, maximum likelihood). You should fix the noise variance to 1 (the value chosen should have no effect on the mean predictions). Note that there are no unknown hyperparameters in this covariance function, so you can just directly apply the function for making predictions with a Gaussian process.

Fitting Gaussian process models by maximizing marginal likelihood. You should then fit a Gaussian process model with the following covariance function (similar to one discussed in the lecture notes):

$$K(x, x') \;=\; 100^2 \;+\; \gamma^2 \exp\Big(-\rho^2 \sum_{i=1}^{8} (x_i - x'_i)^2\Big)$$

Here, γ and ρ, along with the noise standard deviation, σ, are unknown hyperparameters. You should estimate values for these parameters by maximizing the marginal likelihood, as in the gp_find_hypers function in the week 6 demo. You will need to provide starting values for the optimization; you should try several, to see whether that affects the results. As for

all the models fitted, you should evaluate how good the results are by the average squared error when predicting the y values in the test set. You should repeat the whole procedure for both training sets, to see how much difference the random choice of training set makes to the results.

The range of covariates 1 and 7 is greater than the range of the other covariates (which are all fractions between 0 and 1). You should fit the model described in the preceding paragraph again after dividing covariates 1 and 7 by ten (in both training sets and in the test set). This may make using the same scale factor, ρ, for all covariates more reasonable. You should compare the results with those done without this re-scaling of these two covariates.

You should use the re-scaled covariates for all the models described in the rest of this assignment handout.

You should try two more covariance functions by this method. One is like the covariance function above, but using the absolute values of differences in covariates rather than squares:

$$K(x, x') \;=\; 100^2 \;+\; \gamma^2 \exp\Big(-\rho \sum_{i=1}^{8} |x_i - x'_i|\Big)$$

Note that ρ is not squared above.

If we're not sure whether using the absolute value or the square is better, we can use a covariance function that has both, as follows:

$$K(x, x') \;=\; 100^2 \;+\; \gamma_1^2 \exp\Big(-\rho \sum_{i=1}^{8} |x_i - x'_i|\Big) \;+\; \gamma_2^2 \exp\Big(-\rho^2 \sum_{i=1}^{8} (x_i - x'_i)^2\Big)$$

Note that γ_1 and γ_2 allow for the two terms to have different importance, but that ρ is common to both terms.

Fitting Gaussian process models using cross validation. Next, you should try fitting models using the same three covariance functions as above, but with the hyperparameters found by cross validation, minimizing average squared error on cases left out.

You should use R's "nlm" function for this, as was done in gp_find_hypers, but instead of minimizing minus the log of the marginal likelihood as in gp_find_hypers, you should minimize a cross validation estimate of squared prediction error. To get this estimate, you should divide the training set into ten equal parts (taking the first 50 cases, the next 50, etc.), and predict each part from the other nine parts, with whatever the current hyperparameter values are.

You should do this for both training sets, and evaluate how good the results are by the squared error when predicting the test cases, which you can compare to the squared error when hyperparameters are estimated using marginal likelihood.

Fitting Gaussian process models using importance sampling. Finally, for each of the three covariances functions used above, you should use importance sampling from the prior distribution to make predictions for test cases that average over the posterior distribution for the hyperparameters. (Doing this for both training sets, with covariates 1 and 7 re-scaled as described above.)

The prior distribution used should be that $\log \sigma$ has the $N(-1, 1)$ distribution, and the logs of all the other hyperparameters have the $N(0, 1)$ distribution, independently. You should

sample at least 1000 values for the hyperparameters (though you may use less if really necessary because your computer is slow), and then give each value a weight proportional to the probability density of the responses in the training set. Your predictions for test cases should be the weighted average of the predictions with all these sets of hyperparameters.

What to hand in. You should hand in a paper printout of your functions, the R scripts used to apply them to the datasets, and the output you obtained, along with a discussion of what it means. In the discussion, you should compare the average squared prediction errors of the methods, how much computation time they took, and what values for the hyperparameters they chose. You should also discuss how much the results varied between the two training sets. Your aim in the discussion should be to understand what happened, not just report it. You might wish to do additional experiments, or examine or plot additional information, in order to reach such an understanding.