Homework 4 - SVMs & Return of the Bayes

Harvard CS109B, Spring 2017

Feb 2017

Problem 1: Celestial Object Classification

SVMs are computationally intensive, much more so than other methods we've used in the course. Expect run times for your analyses to be much larger than before. Several SVM packages are available, we recommend using the `e1071` library, though you're free to use whatever package you feel comfortable with – we'll provide extra hints for the `svm` function from this package.

In this problem, the task is to classify a celestial object into one of 4 categories using photometric measurements recorded about the object. The training and testing datasets are provided in the `dataset_1_train.txt` and `dataset_1_test.txt` respectively. Overall, there are a total of 1,379 celestial objects described by 61 attributes. The last column contains the object category we wish to predict, `Class`.

We'll be working with Support Vector Machines, trying out different kernels, tuning, and other fun things. *Hint*: Use the `kernel`, `degree`, `cost`, `gamma` arguments of the `svm` function appropriately.

First, ensure that the that `Class` is a factor (quantitative values). These should be object categories and not integer values – use `as.factor` if needed.

1. Fit an RBF kernel to the training set with parameters `gamma` and `cost` both set to 1. Use the model to predict on the test set.
2. Look at the confusion matricies for both the training and testing predictions from the above model. What do you notice about the predictions from this model? *Hint*: The `confusionMatrix` function in the `caret` package is quite useful.
3. For the RBF kernel, make a figure showing the effect of the kernel parameter γ on the training and test errors? Consider some values of `gamma` between 0.001 and 0.3. Explain what you are seeing.
4. For the RBF kernel, make a figure showing the effect of the `cost` parameter on the training and test errors? Consider some values of `cost` in the range of 0.1 to 20. Explain what you are seeing.
5. Now the fun part: fit SVM models with the linear, polynomial (degree 2) and RBF kernels to the training set, and report the misclassification error on the test set for each model. Do not forget to tune all relevant parameters using 5-fold cross-validation on the training set (tuning may take a while!). *Hint*: Use the `tune` function from the `e1071` library. You can plot the error surface using `plot` on the output of a `tune` function.
6. What is the best model in terms of testing accuracy? How does your final model compare with a naive classifier that predicts the most common class (3) on all points?

Hint: This is a moderate-sized dataset, but keep in mind that computation will always be a limiting factor when tuning machine learning algorithms. For timing reference, attempting 40 combinations of `cost` and `gamma` using an RBF kernel on the training dataset took about 15 minutes to tune on a recent Macbook. The other kernels were much faster, e.g. linear should be done in only a few minutes.

Problem 2: Return of the Bayesian Hierarchical Model

We're going to continue working with the dataset introduced in Homework 3 about contraceptive usage by 1934 Bangladeshi women. The data are in `dataset_2.txt` which is now a merge of the training and test data that appeared in Homework 2.

In order to focus on the benefits of Hierarchical Modeling we're going to consider a model with only one covariate (and intercept term).

1. Fit the following three models
   (a) Pooled Model: a single logistic regression for `contraceptive_use` as a function of `living.children`. Do not include `district` information. You should use the `glm` function to fit this model. Interpret the estimated model.
   (b) Unpooled Model: a model that instead fits a separate logistic regression for each `district`. Use the `glm` function to this model. *Hint* The separate logistic regression models can be fit using one application of `glm` by having the model formula be `contraceptive_use ~ -1 + living.children * as.factor(district)`. Explain why this model formula is accomplishing the task of fitting separate models per district. Examine the summary output of the fitted model. Briefly explain the reason for many of the `NA` estimates of the coefficients.
   (c) Bayesian Hierarchical Logistic Model: a Bayesian hierarchical logistic regression model with `district` as the grouping variable. Use the `MCMChlogit` function in the `MCMCpack` library using arguments similar to the reaction time model in the lecture notes. Make sure that both coefficients of the linear predictor are assumed to vary by `district` in the model specification. Describe briefly in words how the results of this model are different from the pooled and unpooled models of parts (a) and (b).

2. In class we discussed that one of the benefits of using Bayesian hierarchical models is that it naturally shares information across the groupings. In this case, information is shared across districts. This is generally known as shrinkage. To explore the degree of shrinkage, we are going to compare coefficients across models and districts based on your results from part 1 above.
   (a) Create a single figure that shows the estimated coefficient to `living.children` as a function of district in each of the three models above. The horizontal axis should be the districts, and the vertical axis should be the estimated coefficient value (generally three estimated coefficients at each district corresponding to the three models). Make sure that the points plotted for each model are distinct (different colors and/or plotting characters), and that you create a legend identifying the model-specific points on the figure. You may want to consider adjusting the vertical axis if some estimated coefficients are so large (positively or negatively) that they obscure the general pattern of the bulk of points. Be sure to explain your decision.
   (b) Write a short summary (300 words or less) that interprets the graph from part (a). Pay particular attention to the relationship between the coefficients within each district, and how or whether the number of observations within each district plays a role in the relationship. You may speculate on the reasons for what you are seeing.

3. Another benefit of shrinkage is how it affects probability estimates (recall the lucky, drunk friend from lecture whose classical estimate for the probability of guessing correctly was 100%). Extract the estimated probabilities from each model applied to the training data. That is, for the pooled and unpooled analyses, use the `predict` function applied to the fitted object, using the argument `type="response"`. For the hierarchical model, the `$theta.pred` component of the fitted model contains the estimated probabilities.
   (a) Plot histograms of the vectors of probability estimates for each model separately. Make sure you standardize the horizontal axis so that the scales are the same. How does the distribution of estimated probabilities compare across the three models?
   (b) Create a scatter plot comparing predicted values from Unpooled and Hierarchical Models, making sure that the scale of the horizontal and vertical axes are the same, and that the plotting region is square rather than rectangular. Include on the plot the line $y = x$ (why do you think this is a useful line to superimpose?). Briefly interpret the relationship between the probability estimates for these two models. Are there particular features of the plot that highlight the intended benefits of using a hierarchical model over the unpooled analysis? Briefly explain.

Problem 3: AWS Preparation

In prepartion for the upcoming Spark and Deep Learning modules, submit your AWS account information. This should have been created in Homework 0. We need specifically:

- The email address associated with your AWS account
- The email address associated with your Harvard ID, if different from above
- Your AWS ID. This should be a 10 digit number. (Instructions)

We need this information to enable GPU capable compute instances.