Homework 1: Smoothers and Generalized Additive Models

Harvard CS 109B, Spring 2017

Jan 2017

Problem 1: Predicting Taxi Pickups

In this problem, the task is to build a regression model that can predict the number of taxi pickups in New York city at any given time of the day. The data set is provided in the files `dataset_1_train.txt` and `dataset_1_test.txt`, and contains details of taxi pickups in the month of Jan 2015. The first column contains the time of the day in minutes, the second column contains information about the day of the week (1 through 7, with 1 denoting Monday) and the last column contains the number of pickups observed at that time.

Visualize the data and check if the pattern of taxi pickups makes intuitive sense.

Part 1a: Explore different regression models

You are required to fit a regression model that uses the time of the day (in minutes) as a predictor and predicts the average number of taxi pick ups at that time. For this part of the question, you can ignore the `DayOfWeek` predictor. Fit the following models on the training set and compare the R^2 of the fitted models on the test set. Include plots of the fitted models for each method.

1. Regression models with different basis functions:

- Simple polynomials with degrees 5, 10, and 25
- Cubic B-splines with the knots chosen by visual inspection of the data.
- Natural cubic splines with the degree of freedom chosen by cross-validation on the training set

2. Smoothing spline model with the smoothness parameter chosen by cross-validation on the training set
3. Locally-weighted regression model with the span parameter chosen by cross-validation on the training set

In each case, analyze the effect of the relevant tuning parameters on the training and test R^2, and give explanations for what you observe.

Is there a reason you would prefer one of these methods over the other?

Hints: - You may use the function `poly` to generate polynomial basis functions (use the attribute `degree` to set the degree of the polynomial), the function `bs` for B-spline basis functions (use the attribute `knots` to specify the knots), and the function `ns` for natural cubic spline basis functions (use the attribute `df` to specify the degree of freedom). You may use the `lm` function to fit a linear regression model on the generated basis functions. You may use the function `smooth.spline` to fit a smoothing spline and the attribute `spar` to specify the smoothness parameter. You may use the function `loess` to fit a locally-weighted regression model and the attribute `span` to specify the smoothness parameter that determines the fraction of the data to be used to compute a local fit. Functions `ns` and `bs` can be found in the `splines` library.

- For smoothing splines, `R` provides an internal cross-validation feature: this can be used by leaving the `spar` attribute in `smooth.spline` unspecified; you may set the `cv` attribute to choose between leave-one-out cross-validation and generalized cross-validation. For the other models, you will have to

write your own code for cross-validation. Below, we provide a sample code for k-fold cross-validation to tune the `span` parameter in `loess`:

```
# Function to compute R^2 for observed and predicted responses
rsq = function(y, predict) {
  tss = sum((y - mean(y))^2)
  rss = sum((y-predict)^2)
  r_squared = 1 - rss/tss

  return(r_squared)
}

# Function for k-fold cross-validation to tune span parameter in loess
crossval_loess = function(train, param_val, k) {
  # Input:
  #   Training data frame: 'train',
  #   Vector of span parameter values: 'param_val',
  #   Number of CV folds: 'k'
  # Output:
  #   Vector of R^2 values for the provided parameters: 'cv_rsq'

  num_param = length(param_val) # Number of parameters
  set.seed(109) # Set seed for random number generator

  # Divide training set into k folds by sampling uniformly at random
  # folds[s] has the fold index for train instance 's'
  folds = sample(1:k, nrow(train), replace = TRUE)

  cv_rsq = rep(0., num_param) # Store cross-validated R^2 for different parameter values

  # Iterate over parameter values
  for(i in 1:num_param){
    # Iterate over folds to compute R^2 for parameter
    for(j in 1:k){
      # Fit model on all folds other than 'j' with parameter value param_val[i]
      model.loess = loess(PickupCount ~ TimeMin, span = param_val[i],
                          data = train[folds!=j, ],
                          control = loess.control(surface="direct"))

      # Make prediction on fold 'j'
      pred = predict(model.loess, train$TimeMin[folds == j])

      # Compute R^2 for predicted values
      cv_rsq[i] = cv_rsq[i] + rsq(train$PickupCount[folds == j], pred)
    }

    # Average R^2 across k folds
    cv_rsq[i] = cv_rsq[i] / k
  }

  # Return cross-validated R^2 values
  return(cv_rsq)
}
```

Part 1b: Adapting to weekends

Does the pattern of taxi pickups differ over the days of the week? Are the patterns on weekends different from those on weekdays? If so, we might benefit from using a different regression model for weekdays and weekends. Use the `DayOfWeek` predictor to split the training and test sets into two parts, one for weekdays and one for weekends, and fit a separate model for each training subset using locally-weighted regression. Do the models yield a higher R^2 on the corresponding test subsets compared to the (loess) model fitted previously? (You may use the loess model fitted in 1A (with the span parameter chosen by CV) to make predictions on both the weekday and weekend test sets, and compute its R^2 on each set separately, you may also use the same best_span calculated in 1A)

Problem 2: Predicting Crime in the City

In this problem, the task is to build a model that can predict the per-capita crime rate in a given region of the US. The data set is provided in the files `dataset_2_train.txt` and `dataset_2_test.txt`. Each row corresponds to a region in the US: the first column contains the number of violent crimes per 100K population, and the remaining columns contain 8 attributes about the region. All numeric predictors are normalized into the range 0.00-1.00, and retain their distribution and skew (e.g. the population predictor has a mean value of 0.06 because most communities are small)

Examine the relationship between the crime rate and the individual predictors visually. Do some of the predictors have a non-linear relationship with the response variable, warranting the use of a non-linear regression model?

Part 2a: Polynomial regression

Fit the following models on the training set and compare the R^2 score of the fitted models on the test set:

- Linear regression
- Regression with polynomial basis functions of degree 2 (i.e. basis functions x, x^2 for each predictor x)
- Regression with polynomial basis functions of degree 3 (i.e. basis functions x, x^2, x^3 for each predictor x)
- Regression with B-splines basis function on each predictor with three degrees of freedom

Part 2b: Generalized Additive Model (GAM)

Do you see any advantage in fitting an additive regression model to this data compared to the above models?

1. Fit a GAM to the training set, and compare the test R^2 of the fitted model to the above models. You may use a smoothing spline basis function on each predictor, with the same smoothing parameter for each basis function, tuned using cross-validation on the training set.
2. Plot and examine the smooth of each predictor for the fitted GAM, along with plots of upper and lower standard errors on the predictions. What are some useful insights conveyed by these plots, and by the coefficients assigned to each local model?
3. Use a likelihood ratio test to compare GAM with the linear regression model fitted previously. Re-fit a GAM leaving out the predictors 'PrecentageAsian' and 'PercentageUrban'. Using a likelihood ratio test, comment if the new model is preferred to a GAM with all predictors.

Hint: You may use the `gam` function for fitting a GAM, and the function `s` for smoothing spline basis functions. These functions are available in the `gam` library. For k-fold cross-validation, you may adapt the sample code

provided in the previous question. The `plot` function can be used to visualize the smooth of each predictor for the fitted GAM (set the attribute `se` to `TRUE` to obtain standard error curves). You may use the `anova` function to compare two models using a likelihood ratio test (with attribute `test='Chi'`).

Part 2c: Including interaction terms

Re-fit the GAM with the following interaction terms included:

- A local regression basis function involving attributes 'Population', 'PercentageUrban' and 'MedIncome'
- A local regression basis function involving a race-related attribute and 'MedIncome'

You can retain the smoothing parameter chosen earlier for the smoothing spline basis functions, and only tune the smoothing parameter for the interaction terms. Do the interaction terms yield an improvement in the test R^2? Use a likelihood ratio test to check if the fit of these models is better than the previous GAM without interaction terms.

Hint: You may use the function `lo` for local regression basis functions.