CPH 636 — Spring 2009 — Dr. Charnigo

Lecture 10

Nearest neighbors: regression problems

Introduction. We have a continuous response variable Y and several continuous explanatory variables $X_1, \ldots, X_j$. We are already familiar with linear regression models (Lectures 4 and 5), regression trees (Lecture 8), and neural networks (Lecture 9) as possible data mining methods for this scenario. Today we will introduce one more data mining method for this scenario, namely "nearest neighbors analysis".

To motivate nearest neighbors analysis, imagine that you want to predict the glucose level for a 33-year-old mother of three with body mass index 36.1 and diastolic blood pressure 90; let us call her "Person A". Suppose that in your training data set there is a 32-year-old mother of three with body mass index 36.4 and diastolic blood pressure 92; let us call her "Person B". Suppose, moreover, that Person B has a glucose level of 129. Even if you have little or no insight about how glucose level should be related to age, number of pregnancies, body mass index, or diastolic blood pressure, you may be willing to guess that Person A should have a glucose level close to 129.

In general, nearest neighbors analysis entails identifying one or more subjects in the training data set who are "similar" on the explanatory variables to the new person for whom you want to make a prediction. These subjects in the training data set are called "nearest neighbors". When there is one nearest neighbor, the prediction for the new person is the value of the response variable for the one nearest neighbor. When there are multiple nearest neighbors, the prediction for the new person is the average value of the response variable among the multiple nearest neighbors.

All of this sounds fairly simple, and it is, except that we have to determine which subjects in the training data set are "similar" to the new person as well as how many nearest neighbors to consult.

A first attempt at defining similarity. Suppose that Person A, for whom we wish to make a prediction, has $X_1 = a_1, X_2 = a_2, \ldots, X_j = a_j$. For each person in the training data set, we can define a "similarity score" indicating how similar he or she is to Person A on $X_1, \ldots, X_j$. Explicitly, suppose that subject i in the training data set has $X_1 = x_{1,i}, X_2 = x_{2,i}, \ldots, X_j = x_{j,i}$. Consider the similarity score defined by

$$\frac{1}{1 + (x_{1,i} - a_1)^2 + (x_{2,i} - a_2)^2 + \cdots + (x_{j,i} - a_j)^2}.$$

The similarity score is a positive number that cannot exceed 1 and that equals 1 only when subject i is identical to Person A on $X_1, \ldots, X_j$. A higher similarity score indicates greater similarity between Person A and subject i. [[We could also define a "dissimilarity score" by $(x_{1,i} - a_1)^2 + (x_{2,i} - a_2)^2 + \cdots + (x_{j,i} - a_j)^2$, in which case a lower dissimilarity score would be indicative of greater similarity between Person A and subject i.]]

Difficulties in defining similarity. There are some problems with using similarity scores as defined above.

1. *Scaling.* The units in which $X_1, \ldots, X_j$ are expressed matter. Suppose that Person A is a 33-year-old mother of three with body mass index 36.1 and diastolic blood pressure 90, Person B is a 55-year-old mother of three with body mass index 35.1 and diastolic blood pressure 90, and Person C is a 33-year-old mother of three with body mass index 35.1 and diastolic blood pressure 93. With all explanatory variables

expressed in their usual units, the similarity score between Person A and Person B is

$$\frac{1}{1 + 22^2 + 0^2 + 1^2 + 0^2} = 0.002,$$

while the similarity score between Person A and Person C is

$$\frac{1}{1 + 0^2 + 0^2 + 1^2 + 3^2} = 0.091.$$

On this basis, we would conclude that Person C is more similar to Person A than is Person B, which seems intuitively reasonable. Yet, if we express age in decades rather than in years, the similarity score between Person A and Person B is

$$\frac{1}{1 + 2.2^2 + 0^2 + 1^2 + 0^2} = 0.146,$$

while the similarity score between Person A and Person C remains 0.091 because they have the same age. Now we would conclude that Person B is more similar to Person A than is Person C, which does not seem intuitively reasonable. In any event, the possibility that we can get different results based on how we scale our explanatory variables seems unsatisfactory.

2. *Relevance*. Suppose that X_1 has no association with Y but that X_2 has a strong association with Y. Suppose, moreover, that Person B is very different from Person A on X_1 and slightly different on X_2, while Person C is identical to Person A on X_1 and moderately different on X_2. Person C will have a higher similarity score than Person B, but we would rather consult Person B because Person B is closer on the explanatory variable that matters. Thus, whether similarity scores should be based on all of the explanatory variables is unclear.

3. *Dimensionality.* If j is large, then we face the curse of dimensionality. Even the subjects in the training data set with the highest similarity scores may be quite different from the new person for whom we want to make a prediction.

Discussion questions. How might we remedy the first difficulty described above? What about the second and third difficulties?

Nearest neighbors prediction. Once we adjust the similarity score as indicated in our dialogue for the first discussion question, we make predictions as suggested in the Introduction. Letting k denote the number of nearest neighbors to be consulted, we find the k subjects in the training data set who are most similar to Person A. Then the prediction for Person A is the average value of Y among the k nearest neighbors.

How many nearest neighbors? The number of nearest neighbors may be specified ahead of time or determined with the aid of a validation data set. To make the determination using a validation data set, perform nearest neighbors analysis several times with different values of k, then choose the value of k for which average squared error on the validation data set is minimized.

Enterprise Miner takes $k = 16$ by default, but you can change this. A large k implies that you are taking an average over many Y values in the training data set (lower variance, higher bias), while a small k implies that you are taking an average over few Y values in the training data set (higher variance, lower bias).

Let n denote the size of the training data set. Very roughly, a nearest neighbors analysis with k nearest neighbors is comparable in its complexity to a linear regression model with p parameters, where p is the positive integer closest to n/k. In particular, note that $k = n$ and $p = 1$ correspond to making the same prediction for everyone, namely the sample mean of Y in the training data set.

Illustrative example. Refer to {em_report.html} in the {NearNeighbor} folder on my home page. As in Lecture 9, we consider the diabetes data set introduced in Written Assignment 1. The response variable is GLU. The explanatory variables are NPREG, BP, BMI, and AGE. I standardized the explanatory variables before using Enterprise Miner. For this illustration, I used the Enterprise Miner default of $k = 16$ nearest neighbors.

Scrolling down to the part labeled "Memory-Based Reasoning" (another name for nearest neighbors analysis), we click on "Output". The average squared error on the training data set is 793.55, the average squared error on the validation data set is 494.69, and the average squared error on the test data set is 1508.37.

We already know, from considering a related example in Lecture 9, that the strange pattern of average squared errors is an artifact of the random split for this data set, not a deficiency of the data mining method. Even so, nearest neighbors analysis appears to fare less well than either the regression tree or the neural network. However, we must keep in mind that changing k would alter the results, and there is no reason to believe that $k = 16$ is optimal for this data set. In particular, the average squared error on the test data set might have been considerably less than 1508.37 with a different k. That said, we would not choose k to minimize average squared error on the test data set; the role of the test data set is final evaluation, not model selection. Yet, choosing k to minimize average squared error on the validation data set might also yield a smaller average squared error on the test data set.

Predictions for specific individuals can be obtained by clicking on the "Datastep Score Code" link and appropriately modifying the contents as indicated in {SASEnterInstr3.txt}.

Nearest neighbors: classification problems

Introduction. We have a categorical response variable Y and several continuous explanatory variables $X_1, \ldots, X_j$. We are already familiar with logistic regression models (Lecture 6), discriminant analysis (Lecture 7), classification trees (Lecture 8), and neural networks (Lecture 9) as possible data mining methods for this scenario. Now we will show that nearest neighbors analysis can be adapted to this scenario as well.

Nearest neighbors prediction. Letting k denote the number of nearest neighbors to be consulted, we find the k subjects in the training data set who are most similar to Person A. The prediction for Person A is whichever category of Y appears most frequently among the k nearest neighbors. For instance, if $k = 16$ and 11 of the nearest neighbors have $Y = 1$ while 5 have $Y = 0$, then we predict that Person A will have $Y = 1$.

Discussion question. How should ties be handled? For instance, what should be done if $k = 16$ and 8 of the nearest neighbors have $Y = 1$ while 8 have $Y = 0$?

Illustrative example. Refer to {em_report.html} in the {NearNeighbor2} folder on my home page. Once more we consider the diabetes data set. Now the response variable is DIAB, while the explanatory variables are GLU, NPREG, BP, BMI, and AGE. As before, I standardized the explanatory variables before using Enterprise Miner, and I used the Enterprise Miner default of $k = 16$ nearest neighbors.

The misclassification rate on the training data set is 0.240, the misclassification rate on the validation data set is 0.240, and the misclassification rate on the test data set is 0.240. The misclassification rate on the test data set is identical to what we obtained with the neural network in Lecture 9. However, there is no reason to believe that $k = 16$ is optimal for this data set. In particular, the misclassification rate on the test data set might be less than 0.240 if k were changed. Again, we would not choose k to minimize the misclassification rate on the test data set; the role of the test data set is final evaluation, not model selection. Yet, choosing k to minimize the misclassification rate on the validation data set might also yield a smaller misclassification rate on the test data set.

Strengths and weaknesses of supervised learning techniques

Introduction. We are now in a position to examine the strengths and weaknesses of competing data mining methods. Among several aspects of performance to consider are predictive ability, interpretability of results, handling of irrelevant inputs, sensitivity to outliers in the explanatory variables, and accommodation of missing values. In what follows, I refer to linear regression, logistic regression, and discriminant analysis collectively as "linear methods". This part of Lecture 10 closely follows Section 10.7 of Hastie *et al.*

Predictive ability. Neural networks and nearest neighbors analysis are regarded by the experts as strong in predictive ability. Trees are viewed as comparatively weak, while the linear methods are considered intermediate.

These characterizations should not be taken to imply, for instance, that a neural network will make better predictions than a tree on every data set that one may encounter. Rather, these characterizations summarize the experts' overall impressions based on their experiences with many data sets.

Interpretability of results. The linear methods and trees are most amenable to interpretation. Neural networks and nearest neighbors analysis are least amenable to interpretation, neural networks because of the complicated structure and nearest neighbors analysis because of the lack of structure.

Handling of irrelevant inputs. Trees are strong in this regard since, at any given branch, only the single most relevant input is consulted. The linear methods are also strong since there exist convenient variable selection algorithms. Neural networks and nearest neighbors analysis are weak; one sometimes relies on variable selection algorithms from the linear methods.

Sensitivity to outliers in the explanatory variables. Trees and nearest neighbors analysis are less sensitive to outliers in the explanatory variables. The linear methods and neural networks are more sensitive.

Accommodation of missing values. Trees are considered strongest, as surrogate splits naturally bypass missing values. Neural networks, linear methods, and nearest neighbors analysis are viewed as comparatively weak.