Mertricks

İsmim Mert, metric'lerle yaşarım, trick'lere bayılırım. The Matrix serisini de severim.

From Least Squares to KNN, Statistical Models, Supervised Learning and Function Approximation

April 24, 2016May 1, 2016
Infinity returns!

As you know, I follow the book **The Elements of Statistical Learning** and my main motivation is to help people like me who are suffering from not understanding anything on their first attemp to read the book since the book is not quite appropriate for beginners. This is a little bit tough subject but I will try to simplfy it as much as I can or at least to give some intuition that lays the foundation of the theories we will encounter.

Starting with Two Simple Methods: Least Squares and Nearest Neighbors

These are the good starting points in this journey because they are the easiest, most basic, and most intuitive approaches that have their origins in the pre-computer era, and the large portions of today's much fancier methods are the intelligent derivations of these two approaches.

The linear regression with **least squares** makes a huge assumptions: the model is linear. On the other hand, **k-nearest neighbors** makes very mild assumption about the structure of the model. The assumption made in linear regression makes it relatively stable but its predictions tend to be inaccurate because the real world is almost never linear. The KNN, on the other hand, tends to give accurate predictions but it can be easily unstable due to its mild assumptions. Let's dive into them.

Linear Models and Least Squares

From this point on, the book starts to express every calculation and computation in matrix and vector forms so the understanding of linear algebra would be a huge time and life saver in the remaining of the book. The residual sum of squares is given by:

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2.$$

Here we are trying to find such betas(coefficients) to minimize the RSS. In this example, there is only one input variable X and one output variable Y. So the alternate representation of this is in vector form:

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta),$$

Consider Y is a vector of outputs and X is a matrix of inputs (1 for every row of the first column, and the value of the random variable X for the second column). Taking the derivative of it and equating it to zero, we obtain betas as:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},$$

This is a basic linear algebra and differentiation job. But why did we include ones for the first columns? Let's leave it for further subjects.

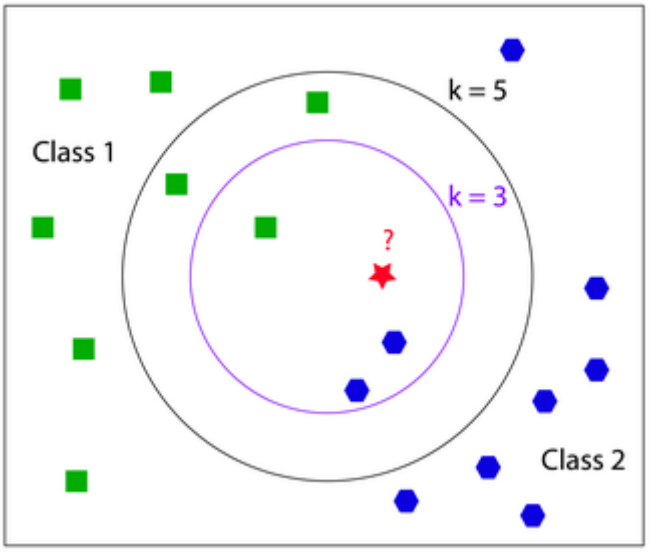
The result is a two-element vector, the first element is intercept and the other is the slope. Why? Because we include the intercept in X and made it two columns matrix.

This formula requires X transpose * X to be nonsingular so that it has an inverse. What does it mean? In linear algebra, a matrix has an inverse if it is ***nonsingular*** or in other words has ***full-rank***. This means that the columns of the matrix have to be **linearly independent** of each other so that its determinant will be non-zero quantity. Since we use determinant in the denominator to find the inverse of a matrix, a number / 0 is undefined. That is why it has to be full rank. Fortunately, there are more intelligent ways to calculate betas utilizing matrix decomposition methods such as **QR decomposition** that does not suffer from ill-conditioned zero determinant case.

Nearest-Neighbor Methods

This is a more intuitive one than least squares linear regression is. In order to form a prediction for an observation, we find the nearest neighbors of it in terms of some distance measures such as euclidean distance.Then we average the outcomes of the neighbors to predict the outcome of the observation.

This simple method has only one parameter which is k: the number of neighbors. As k goes to infinity the prediction will be less accurate since we average over more observations but this decreases the variance of the predictions. On the other hand, as k goes to 1, the prediction tends to be more accurate because an observation is likely to be very similar to its nearest neighbor but the variance of prediction will be higher. **In this case, we cannot use least squares methods to find optimal k because it would always pick 1 for k.**



One big disadvantage of this method is that in order to find an observation's nearest neighbors it needs **n*n distance matrix** to store each observation's distance to all other observations. Here, n is the number of observations in the data and as n increases, n*n increases even faster which makes the computation and storage very costly.

From Least Squares to Neares Neighbors

Least squares linear regression in which linear decision boundary is assumed is the most rigid one and stable to fit. Nearest neighbors, on the other hand, does not assume any stringent assumptions about the decision boundary and can adapt any situation. However, this makes it unstable relative to least squares (hight variance, low bias).

As it is always said, there is no free lunch in this science, so each method has its own advantages and disadvantages depending on data at hand. In some ways, these two methods can be enhanced to develop more advanced methods:

1- In KNN, all k-nearest neighbors of an observation has equal weights in averaging but it doesn't have to be in this way. We might use some weights that decrease smoothly to zero with distance from the target point. We do this by a function called **Kernel**.

2- If the space is high dimensional then distance kernels can be modified to emphasize some variables more than others.

3- Rather than fitting a linear regression to entire data globally, we can fit linear models locally by locally weighted least squares.

4- We can expand the original inputs to a basis in which we can develop any complex models. Truncated power basis is an important methods for this.

5- We can non-linearly transform linear models by projection pursuit or neural network.

Statistical Decision Theory

This section of the book is little bit difficult if you do not know basic probability. The book says than the statistical decision theory requires a **loss function** L(Y, f(X)) for penalizing errors in prediction. **This is intuitive because we can quantify or measure how well our model is only when we have some penalty or loss function and all what we do is to minimize this loss.** Squared error loss by far is the most convenient one:

$$\begin{aligned} \mathrm{EPE}(f) &= \mathrm{E}(Y - f(X))^2 \\ &= \int [y - f(x)]^2 \Pr(dx, dy), \end{aligned}$$

E means **expected value** of its argument which is also known as mean. The first line might be obvious to you, but in the second line the things become messy. Well, it is the basic properties of probability theory actually. The quantity that we are trying to find its expected value is a continuous jointly distributed random variable. It consists of Y as a one random variable and f(X) as the other random variable and our quantity is the function of the two random variables: Y-f(X). In order to find the expected value of it, we need to integrate it with its joint probability density function. For more information, you can visit ***my notes of probability course from MIT. (https://mertricks.com/2016/04/24/mitx-6-041x-introduction-to-probability-the-science-of-uncertainty/)***

If you revisited the probability then you may remember that a joint pdf of two random variable can be splitted into two parts as:

$$p_{xy}(x, y) = p_{y|x}(y)p_x(x),$$

So when you replace Pr(dx, dy) in the original EPE formula above with this split and rearrange the terms, you can obtain:

$$\int_D g(x, y)p_{xy}(x, y)dxdy = \int_{D_x} p_x(x) \left(\int_{D_y} g(x, y)p_{y|x}(y)dy \right) dx$$

here g(x,y) denotes the functions of two random variables which is **[y-f(x)]^2**. Then it is simply conditioned on X, and EPE can be written as:

$$\mathrm{EPE}(f) = \mathrm{E}_X \mathrm{E}_{Y|X} \left([Y - f(X)]^2 | X \right)$$

This is again from probability theory. This is specifically called **total expectation theorem** (or more specifically **law of iterated expectation**, which is an abstract version of total expectation theorem). I cannot go over details of these theorems but you can learn by your self and use my notes of probability. My aim is just to give you some guidance to make you understand what is going on here. Going back to our discussion, EPE can be minimized poinwise. The solution is the conditional expectation which is known as regression function. Then when least squares is used in fitting, the best prediction of Y at any point X=x is the conditional mean of Y.

The nearest-neighbor methods attempt to directly implement this recipe using the training data. **The difference is that expectation is approximated by averaging over sample data, and conditioning at a point is relaxed to conditioning on some region close to the target point.**

The two methods end up approximating conditional expectations by averages. But where do they differ? **The least squares assumes that f(x) is well approximated by a globally linear function whereas knn assumes f(x) is well approximated by a locally constant function.** KNN fails in high dimensional space because the neares neighbors need not be close to the target point and can result in large errors. That is why KNN is avoided when the input space is high dimensional.

Function Approximation

All we try to do is to approximate to the true function (relationship) between inputs and output. What ever you call it machine learning, statistical learning, mathematical modelling, and so on, they all have this aim. The aim is to find a useful approximation to f(x). We can treat supervised learning as a function estimation problem in this sense.

The majority of approximations have associated a set of parameters θ so that we can modify them to suit the data at hand. In linear regression those parameters are betas, β, (the coefficients). Linear basis expansions could be another class of useful approximation and sigmoid transformation common to neural networks could be an example of nonlinear transformation.

$$h_k(x) = \frac{1}{1 + \exp(-x^T \beta_k)}.$$

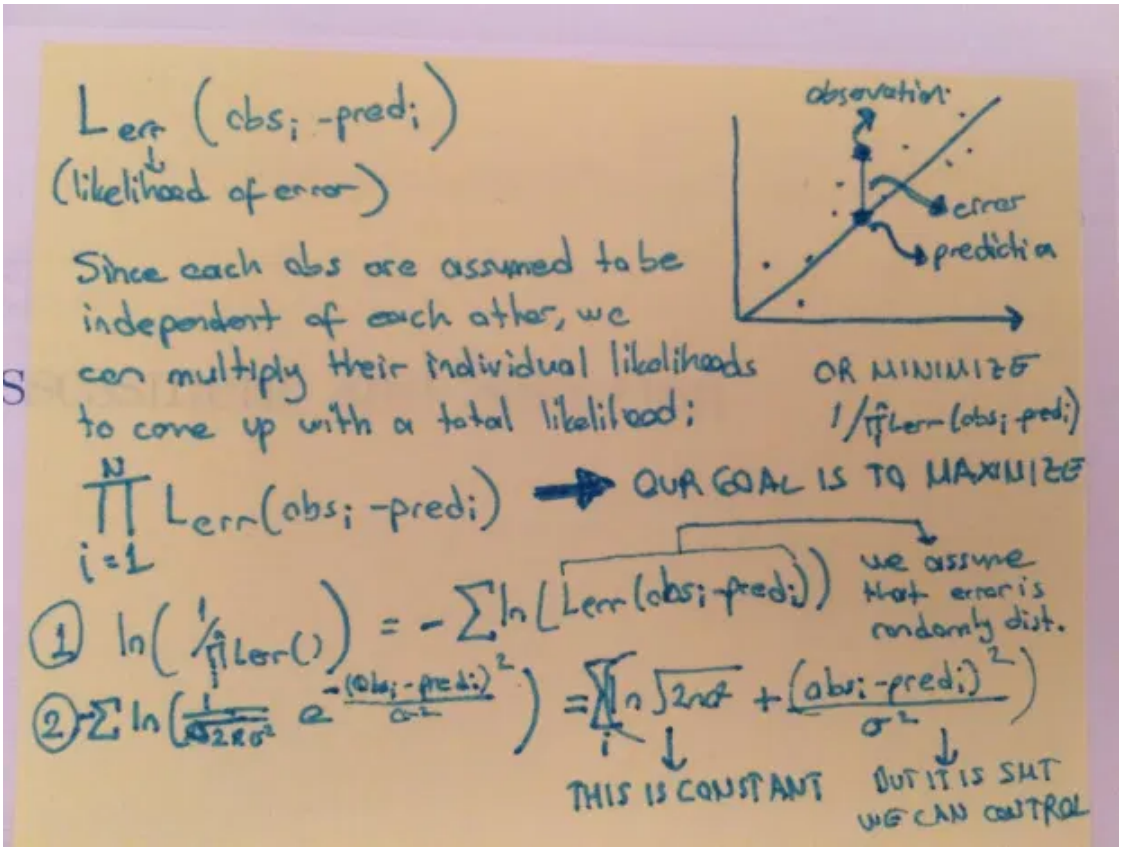
Least squares methods can be used to estimates such parameters by minimizing the residual sum of squares. Linear models has a closed form solution to the minimization problems but in some models the solution requires either iterative methods or numerical optimization.

In some situations least squares does not make much sense. A more general principle for estimation is **maximum likelihood estimation**. If we have observations from a density $\Pr\theta(y)$ indexed by some parameters θ, the log probability of the observed sample is

$$L(\theta) = \sum_{i=1}^{N} \log \Pr_\theta(y_i).$$

Maximum likelihood estimation says that the probability of observed sample is maximum. The intuition behind it is that we are likely to observe high probability observations more often than we observe low probability observations. So we assume that what we observe must be the most likely one.

Why is maximum likelihood estimation is more general methods than least squares is? How do they related to each other? Well actually under the assumption of normally distributed error terms, least squares methods are derived from maximum likelihood estimation method and hence they are identical:



As you can see the picture above, the least squares and maximum likelihood are closely related.

Posted in: machine learning, statistical learning | *Tagged: k-nearest neighbors, KNN, least squares, machine learning, maximum likelihood, maximum likelihood estimation, statistical learning*