5 Background Review

I’m an exchange student, and I’ve never taken any lesson in Berkeley before. But I’m quite good at Linear Algebra and Calculus. I’ve studied it in my college, and get a good grade. I have studied about Linear models and least-squares problems, determinants and their properties, Cramer’s rule. Eigenvalues and eigenvectors are also taught. In our school, the vector calculus are included in Multivariable Calculus. We studied vector calculus, partial derivatives, and double/triple intergrals and line integrals in the plane/3-space.

I have studied about concept of probability and statistics, but I’m not so familiar to stochastic processes. I’ll learn something about that.

Optimization models are only introduced in our algorithm class. I applied for MATH 170 this semester, so I think I will be familiar with it soon.

I’ve studied several programming languages including C++ and Java, but I’ve never had a lesson about python. I’ve studied it on my own, and I don’t think it’s difficult. I’ve solved several problems with python, but to be honest I use C++ more frequently, and our course about data structure also uses C++ rather than python.

6 Your own Question

If the fitting function is nonlinear, we can’t use the least-square regression. At this time, we can use Gradient Descent algorithm to solve the problem. But I find these two ways are similar. Both of these two algorithms are trying to estimate a function and estimate the new data according to the dependent variables. The goals are always calculating the minimum of |y hat - y|. So what’s the difference about these two ways of regression?