Simulation of the Activation Detection in fMRI

---Final Project Report

Xiang Hao

Goal

The goal of this project is to simulate the activation detection in fMRI, which is, in some sense, how to compute the response magnitude of certain area of the brain.

Part I: background

If we want to compute the response magnitude, there is one question here. That is how to measure the level of similarity? This is equal to how to estimate the response magnitude of certain area in the brain.

One method is the general linear model.

$$Y = \beta_1 + X\beta_2 + \varepsilon$$

In this model, X is the response function. Y is the data. beta2 is the response magnitude. Epsilon is the noise. For beta1, if there is no stimulus. The beta2 is zero. Then y is just beta1 plus the noise. So the beta1 is the value while the brain is at rest.

Ok, so we have the general linear model. We need to fit it to the data. If there is a good fit between the model and the data, it means that the data is probably caused by the stimulation.

To get beta, we need to minimize the sum of squared errors, and in R we can use lm command to do it.

After we obtain the beta2, the next step is to test whether or not there is an effect in certain voxel. That is whether the beta2 is big enough to say the voxel is activated. Our method is to use the ratio of effect to its standard error and this is a t statistic.

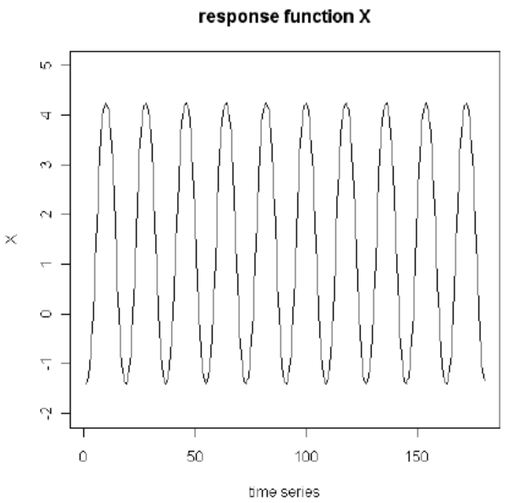
The null hypothesis is that there is no effect and the alternative hypothesis is that there is an effect. Under the null hypothesis, we compute the T. If the computed p-value is below the threshold chosen for statistical significance, usually the 0.05, then the null hypothesis is rejected, which mean there is an effect.

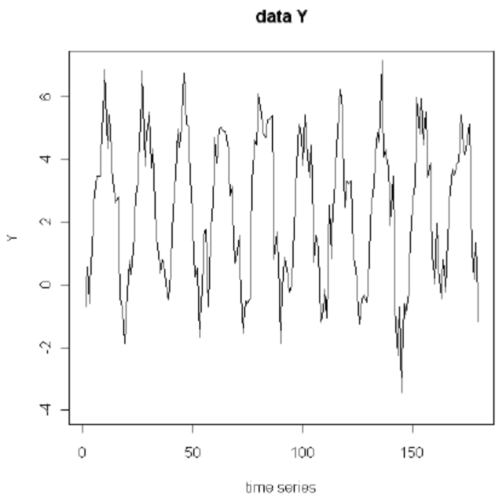
Part II: Simulation

1 Simulate the data.

In this project, we did not use real data, so the first step is to simulate the response function X and the data Y, which should be gotten form certain area of the brain. In addition, while we generate the data Y, the noise should be correlated, but here we first do a simulation while the noise is independent and then we do another simulation while noise is correlated.

Here is what we got:





Where $Y = 1 + 1 * X + e$

2 Compute the response magnitude.

While the noise is independent, we can simply use the R routine lm to compute the response magnitude.

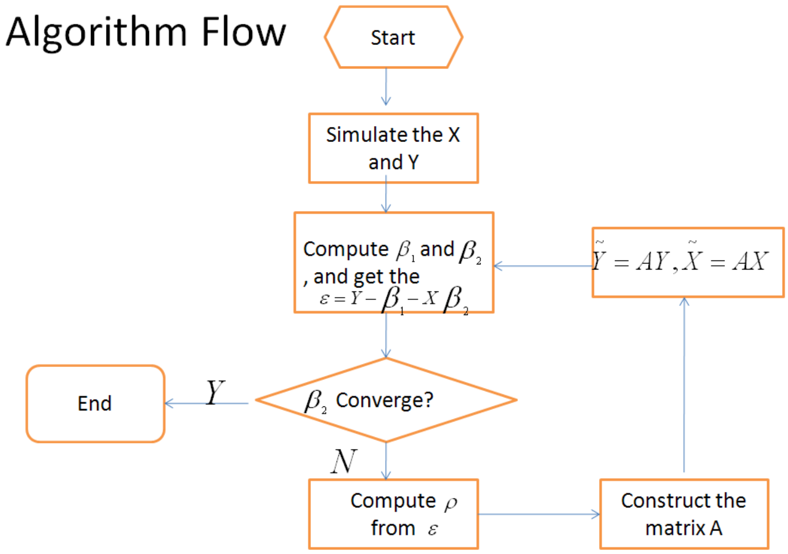
Result: $Y = 0.96714 + 0.97833 * X + e$

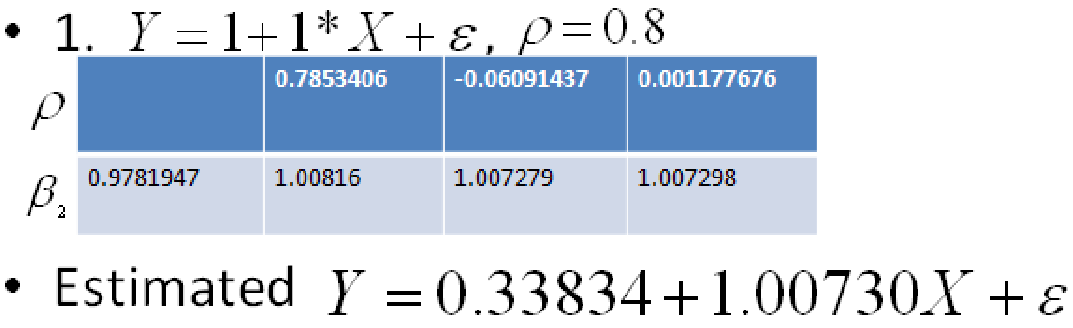
On the other hand, while the noise is correlated, we need a matrix A to whiten the noise, which is to make the noise become uncorrelated. We can find that in order to get the matrix A, we only need to know the correlation rho, and the rho can be computed from the noise.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -\rho R & R & 0 & \dots & 0 \\ 0 & -\rho R & R & \ddots & \vdots \\ \vdots & \ddots & & \ddots & 0 \\ 0 & \dots & 0 & -\rho R & R \end{pmatrix}$$

Where $R = \frac{1}{\sqrt{1-\rho^2}}$, so we just need to know ρ in order to get matrix A.

Here is the algorithm flow of computing the response magnitude while the noise is correlated.


Algorithm Flow



- 1. $Y = 1 + 1 * X + \varepsilon,\ \rho = 0.8$

ρ		0.7853406	-0.06091437	0.001177676
β_2	0.9781947	1.00816	1.007279	1.007298

- Estimated $Y = 0.33834 + 1.00730X + \varepsilon$

Here is an example.

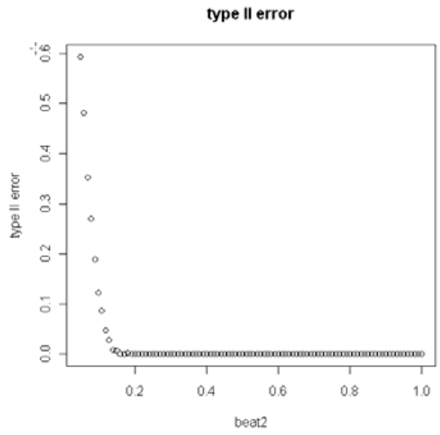
While simulating the Y, the beta1 and beta2 are both 1 and rho is 0.8. We got beta1 is 0.33834 and beta2 is 1.00730 at last.

In the table, the first column is the rho and second is the beta2. The first beta2 is 0.9781947 and this is gotten while the noise is still correlated. The first rho is also quite close to the real value, and the later rho is quite small, since the when you get the first rho, you construct the matrix A, and this matrix A seems to effectively whiten the correlated noise. This maybe the reason why in practice just one iteration seem to be enough.

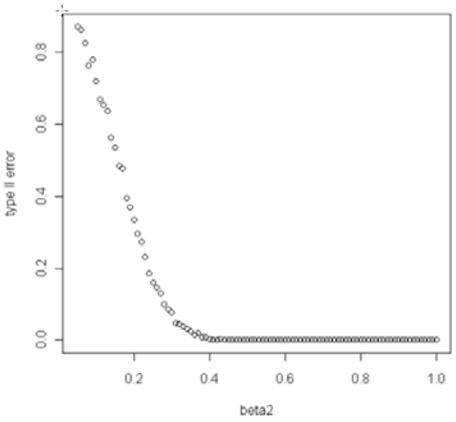
3 Compute the type I and type II error.

For, Type I error, when generating Y, I use beta2 is 0, that mean that should no effect at all. In the t test the null hypothesis is there is no effect, and this is true. Then I count the number of times that it rejects the null hypothesis and got 0.053. (alpha is 0.05)

About the type II error, if we want to compute the type II error, we need to be sure about that that is an effect in certain areas of the brain, but we do know how big does the beta2 have to be to make sure there is an effect. So in some sense, here is not type II error. We can see this in this way that try different beta2 and see the number of times that the null hypothesis is accept.



Noise is independent.



Noise is correlated.

Here is tow plots of the beta2 and the type II error. We can see that as the beta2 increases, type II error goes to 0, but in the left picture, the type II error becomes 0 around 0.2, but in the right one, it is 0.4, which implies that we need a higher beta2 in order to see there is an effect in the area of the brain while the noise is correlated.