Gaussian Naive Bayes with Smooth Basis Functions

Sunsern Cheamanunkul
scheaman@andrew.cmu.edu

Advisors: Tom Mitchell and Rebecca Hutchinson

Carnegie Mellon University

1 Introduction

We examine advantages of using smooth basis functions in classifying fMRI (functional MR Imaging) data. fMRI data is a measurement of neural activity in the brain. It allows us to see how each part of the brain responses to stimuli. The task in which we are interested is to identify mental states from some given fMRI data. Specifically, we want to classify between two different states using labeled data. There are two main challenges for this task. First, fMRI data is very noisy and high-dimensional. Second, we only have limited amount of training data.

One of the traditional ways is to use Gaussian Naive Bayes (GNB) classifier. The Naive part means we assume that each data point, i.e. activation at each voxel at each time point, is independent. The Gaussian part means we also assume that each data point is drawn from a normal distribution with parameters, μ and σ.

Technically, fMRI measures changes in the blood oxygenation level, also known as hemodynamic response (HR) which is an indirect correlate of neural activity in the brain. The hemodynamic response to a stimulus can last some period of time. Therefore, it makes sense to believe that fMRI data is composed of many HRs fired at different times. Since the HR is simply blood flow, we should be able to model each HR with a smooth function over time. Consequently, this idea suggests us that we should be able model fMRI data using composition of smooth functions.

In this thesis, we modify the traditional GNB so that, instead of learning μ at each time point independently, we learn weights of smooth basis functions and calculate μ accordingly. Furthermore, by doing so, it allows us to capture the time-dependence feature of the data. In addition, we also have the benefit of having fewer parameters to estimate. In our model with basis functions, for each voxel, we only need to estimate $B \times C$ parameters where B is the number of basis functions we use and C is number of classes, while the traditional GNB needs to estimates $T \times C$ parameters where T is the number of time points. Usually, T is greater than B.

2 Approach

Our goal is to incorporate the idea of basis functions into the traditional GNB calculation. We design a new classifier based on the traditional GNB, called GNB with basis functions. Most part of the calculation remains the same. That is, at each time point, we want to find μ that maximizes the likelihood of a class given the data. However, instead of estimating μ using the sample mean, we estimate weights of basis functions and calculate μ based on them. We assume that data from any two different voxels is independent, and data at different time points is also independent. Also, we assume that data at each time points is drawn from a normal distribution. Given labeled training examples, we can learn the weights of basis functions for each voxel by the following calculation.

Let X be a matrix containing only training examples of class c.
For each class c,

$$\begin{aligned} P(c|X) &= \frac{P(X|c)P(c)}{P(X)} \\ &\propto P(X|c)P(c) \end{aligned}$$

Assume any two examples are independent. Suppose there are N examples in X and each example contains T data points per voxel.

$$\begin{aligned} P(X|c) &= \prod_{i=1}^{N} P(X_i|c) \\ &= \prod_{i=1}^{N} \prod_{t=1}^{T} P(X_{it}|c) \end{aligned}$$

Let $h_i(t)$ is the value of the i-th basis function at time point t.

Let B be the number of basis functions.

$$P(X_{it}|c) \quad \tilde{} \quad N(\mu_c(t), \sigma_c(t))$$

where

$$\begin{aligned} \sigma_c(t) &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_{it} - \bar{X}_t)^2} \\ \mu_c(t) &= \sum_{b=1}^{B} w_b^{(c)} h_b(t) \end{aligned}$$

We want to find $\vec{w^{(c)}}$ that maximizes $P(c|X)$, or equivalently $P(X|c)$.

$$\begin{aligned} l &= \ln P(X|c) \\ &= \sum_{i=1}^{N}\sum_{t=1}^{T} \ln P(X_{it}|c) \\ &= \sum_{i=1}^{N}\sum_{t=1}^{T} \ln\left(\frac{1}{\sigma_c(t)\sqrt{2\pi}} \exp\left(\frac{-1}{2\sigma_c(t)^2}\left(X_{it} - \mu_c(t)\right)^2\right)\right) \end{aligned}$$

$$\frac{\partial l}{\partial w_k^{(c)}} = \sum_{i=1}^{N}\sum_{t=1}^{T}\left[\left(\frac{X_{it} - \mu_c(t)}{\sigma_c(t)^2}\right) h_k(t)\right]$$

Solve $\frac{\partial l}{\partial w_k^{(c)}} = 0 \; \forall k$

$$\begin{aligned} \frac{\partial l}{\partial w_k^{(c)}} &= 0 \\ \sum_{i=1}^{N}\sum_{t=1}^{T}\left[\left(\frac{X_{it} - \mu_c(t)}{\sigma_c(t)^2}\right) h_k(t)\right] &= 0 \end{aligned}$$

Assume that each $\sigma_c(t)^2$ is roughly equal. Thus,

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{t=1}^{T}\left(X_{it}h_k(t) - h_k(t)\sum_{b=1}^{B} w_b^{(c)}h_b(t)\right) &= 0 \\
\sum_{i=1}^{N}\sum_{t=1}^{T} X_{it}h_k(t) &= \sum_{i=1}^{N}\sum_{t=1}^{T}\left(h_k(t)\sum_{b=1}^{B} w_b^{(c)}h_b(t)\right) \\
&= N\sum_{t=1}^{T}\left(h_k(t)\sum_{b=1}^{B} w_b^{(c)}h_b(t)\right) \\
&= N\sum_{t=1}^{T}\sum_{b=1}^{B} h_k(t)w_b^{(c)}h_b(t) \\
&= N\sum_{b=1}^{B}\sum_{t=1}^{T} h_k(t)w_b^{(c)}h_b(t)
\end{aligned}
$$

So $\sum_{b=1}^{B}\left(w_b^{(c)}\sum_{t=1}^{T} h_k(t)h_b(t)\right) = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T} X_{it}h_k(t)}{N}$.

We can represent this equation in matrix form,

$$
\begin{pmatrix} & \vdots & \\ \sum_t^T h_1(t)\cdot h_k(t) & \cdots & \sum_t^T h_b(t)\cdot h_k(t) \\ & \vdots & \end{pmatrix}\begin{pmatrix} \vdots \\ w_k^{(c)} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \frac{\sum_i^N \sum_t^T X_{it}\cdot h_k t}{N} \\ \vdots \end{pmatrix}
$$

Next, we represent the preceding equation by $A \cdot \vec{w^{(c)}} = Z$.

So $\vec{w^{(c)}} = A^{-1}Z$

However, we can write $A = R{\cdot}R^T$ where $R = \begin{pmatrix} h_1(1) & h_1(2) & \cdots & h_1(T) \\ h_2(1) & h_2(2) & \cdots & h_2(T) \\ & \vdots & \vdots & \\ h_B(1) & h_B(2) & \cdots & h_B(T) \end{pmatrix}$

Thus $\vec{w^{(c)}} = (R \cdot R^T)^{-1}Z$

Once we know $\vec{w^{(c)}}$, we can obtain $\mu_c(t)$ by using this equation $\mu_c(t) = \sum_{b=1}^{B} w_b^{(c)}h_b(t)$.

Next, we need to consider about choices of basis functions, $h_i(t)$. There are two important questions we need to answer: what our basis functions should be and how many basis functions we should use. Basis functions can be any continuous functions. However, Gamma functions and Gaussian functions are commonly used to model the hemodynamic response. Therefore, we will focus on those two in this study. From this point on, we will look into answering the questions about basis functions and examine how well the GNB with basis functions does. Figure 1 and 2 show examples of possible basis functions. Figure 3 shows the output μ's of the two classifiers.

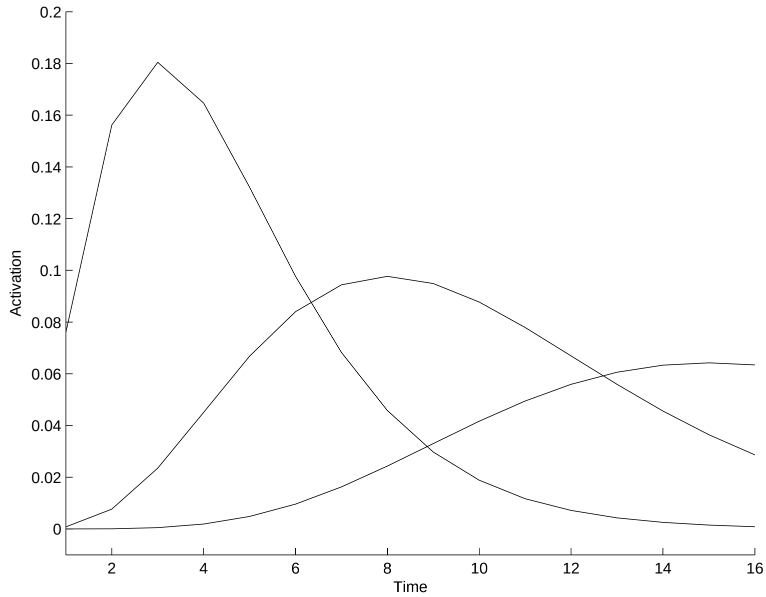

Figure 1: 3 Gamma functions as basis functions

3 Experiments

Basically, our experiment framework is consist of two parts: training and testing. Both training examples and testing examples are labeled. For training, we give training examples and their labels to the classifier. After being trained, the classifer takes in testing examples and outputs their predicted labels. To measure the performance of the classifer, we compare the predicted labels with the true labels.

In this study, we conduct two types of experiments: experiments using synthetic data and experiments using real data. By experimenting on the synthetic data, we can compare the GNB with basis functions with the traditional GNB, given that we have control of the data. In addition, by experimenting on the real data, we will have a better idea of how the GNB

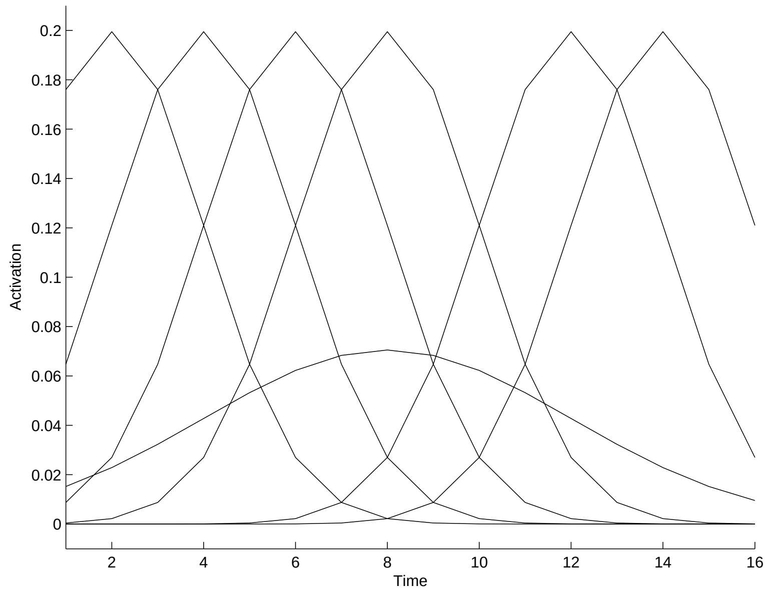

Figure 2: 7 Gaussian functions as basis functions

with basis functions does with the real fMRI data.

3.1 Experiments with synthetic data

Working with synthetic data gives us many advantages since we can control the true nature of the data. Moreover, we can have as many examples as we want. The synthetic data is generated by using a set of basis functions, $g_i(t)$. These $g_i(t)$'s are known to the generator. Then, for each voxel and each class, the generator randomly draws a weight for each $g_i(t)$, say z_i, from a uniform distribution from 0 to 1. Next, the generator adds some noise to the weighted sum of $g_i(t)$. Specfically, the generator outputs $f(t)$ which is drawn from a normal distribution with $(\mu = \sum_i z_i g_i(t), \bar{\sigma})$ where $\bar{\sigma}$ is the noise parameter.

3.1.1 Correct basis functions

For our first experiment, we generate examples using 3 Gamma functions as basis funtions. That is $g_i(t) = \frac{1}{\tau(n-1)!}\left((t/\tau)^{(n-1)}e^{-(t/\tau)}\right)$ and the parameters (τ, n) are the following: (1.5,3), (2,5), (2.5,7). There are 2 classes. We generate 100 available training examples and 100 available testing examples

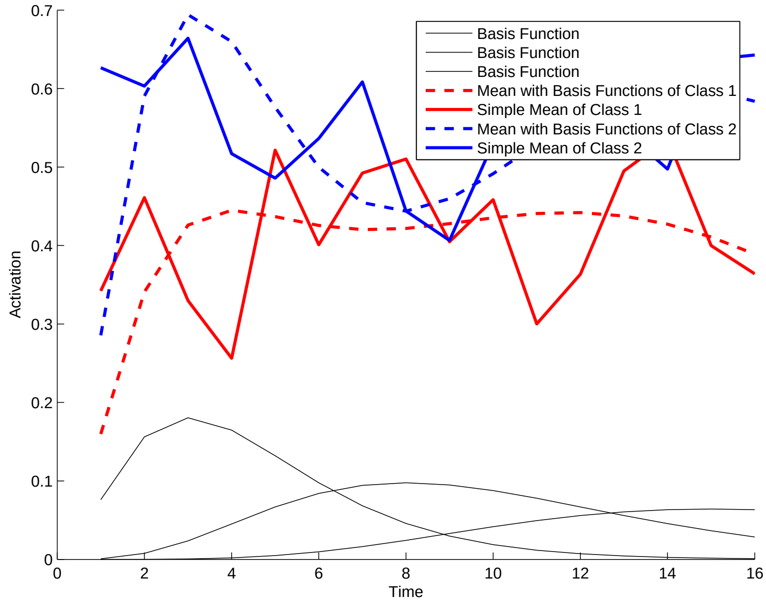

Figure 3: μ's of trained classifiers of a voxel

where exactly half of them are from class 1 and the other half are from class 2. We randomly select only N examples for the actual training, but we use all 100 testing examples for testing. Each example contains the data of 40 voxels and, in each voxel, there are 16 data points representing data at each time point.

For this particular experiment, we set $h_i(t) = g_i(t)$. That is we give the correct information about basis functions to the classifier. However, z_i is still unknown to the classifier, and the classifier needs to estimate them.

We run the experiment in different settings. We vary the noise, $\bar{\sigma}$, and number of training examples used, N. In each setting, we repeat the experiment 30 times. The result is shown in Table 1 and 2.

We can see a significant improvement when there is enough training examples, and it depends on how much noise in the data. Hence we can conclude from this experiment that the GNB with basis functions performs better than the traditional GNB when the classifier knows exactly what $g_i(t)$'s are.

N	$\bar{\sigma}$			
	0.1	0.2	0.3	0.4
10	0.869 ± 0.068	0.633 ± 0.067	0.563 ± 0.055	0.546 ± 0.037
20	0.997 ± 0.006	0.789 ± 0.052	0.640 ± 0.064	0.605 ± 0.044
30	1.000 ± 0	0.879 ± 0.032	0.707 ± 0.046	0.642 ± 0.053
40	1.000 ± 0	0.922 ± 0.027	0.721 ± 0.037	0.669 ± 0.050
50	1.000 ± 0	0.945 ± 0.031	0.741 ± 0.045	0.699 ± 0.043
60	1.000 ± 0	0.961 ± 0.015	0.775 ± 0.032	0.702 ± 0.031
70	1.000 ± 0	0.972 ± 0.012	0.799 ± 0.031	0.721 ± 0.034
80	1.000 ± 0	0.975 ± 0.011	0.814 ± 0.027	0.734 ± 0.029

Table 1: Accuracies of the traditional GNB

N	$\bar{\sigma}$			
	0.1	0.2	0.3	0.4
10	0.903 ± 0.068	0.657 ± 0.062	0.567 ± 0.052	0.546 ± 0.042
20	0.999 ± 0.003	0.838 ± 0.044	0.647 ± 0.052	0.608 ± 0.038
30	1.000 ± 0	0.925 ± 0.027	0.738 ± 0.052	0.659 ± 0.050
40	1.000 ± 0	0.952 ± 0.023	0.761 ± 0.037	0.683 ± 0.046
50	1.000 ± 0	0.969 ± 0.017	0.795 ± 0.046	0.704 ± 0.038
60	1.000 ± 0	0.976 ± 0.012	0.822 ± 0.032	0.726 ± 0.032
70	1.000 ± 0	0.986 ± 0.009	0.847 ± 0.028	0.750 ± 0.033
80	1.000 ± 0	0.982 ± 0.008	0.864 ± 0.022	0.764 ± 0.026

Table 2: Accuracies of the GNB with correct basis functions

3.1.2 Sensitivity tests

Now, let's consider the following questions. What would happen when the classifier does not have prior knowledge about correct basis functions? How sensitive is the accuracy we change the number of basis functions or the shape of basis functions? The setup of this experiment is the same as the previous, except that we are not going to tell the classifier what the $g_i(t)$'s are and how many there are. In addition, we fix $\bar{\sigma} = 0.3$ for this experiment. The result is shown in Table 3, 4 and 5.

Basis functions	N			
	20	40	60	80
traditional GNB	0.635 ± 0.045	0.722 ± 0.043	0.780 ± 0.028	0.815 ± 0.025

Table 3: Accuracy of the traditional GNB

Basis functions	N			
	20	40	60	80
1 Gamma fn (2,5)	0.582 ± 0.042	0.620 ± 0.044	0.658 ± 0.043	0.694 ± 0.027
2 Gamma fns (1.5,3)(2,5)	0.653 ± 0.054	0.754 ± 0.035	0.826 ± 0.032	0.848 ± 0.026
3 Gamma fns (1.5,3)(2,5)(2.5,7)	0.655 ± 0.046	0.761 ± 0.041	0.818 ± 0.029	0.856 ± 0.019
3 Gamma fns (2,2)(3,4)(2.5,6)	0.661 ± 0.047	0.750 ± 0.039	0.809 ± 0.034	0.834 ± 0.023
3 Gamma fns (1,5)(1,7)(1,9)	0.649 ± 0.052	0.750 ± 0.032	0.805 ± 0.029	0.841 ± 0.025
3 Gamma fns (2,1)(3,3)(4,5)	0.640 ± 0.048	0.742 ± 0.039	0.802 ± 0.033	0.835 ± 0.020
5 Gamma fns - correct ones incl	0.656 ± 0.048	0.744 ± 0.034	0.817 ± 0.033	0.855 ± 0.018
5 Gamma fns - no correct ones	0.654 ± 0.058	0.748 ± 0.045	0.812 ± 0.029	0.848 ± 0.026
7 Gamma fns - correct ones incl	0.632 ± 0.051	0.755 ± 0.042	0.791 ± 0.030	0.832 ± 0.029
7 Gamma fns - no correct ones	0.650 ± 0.039	0.744 ± 0.042	0.783 ± 0.030	0.845 ± 0.025

Table 4: Accuracies of the GNB with basis functions using Gamma functions as basis functions

The result shows that when we have enough expressive power, i.e. enough number of basis functions, we can get a comparable accuracy to the traditional GNB by using Gamma functions or Gaussian functions with reasonable parameters. However, if we want get higher accuracy than what the traditional GNB does, we need to be very careful about choosing basis functions. Having too many basis functions results in a drop in accuracy. In one of the cases shown in Table 4, although the set of 7 Gamma basis functions contains the correct basis set, the accuracy in this case is lower than the case where we use 5 Gamma functions with none of the correct basis as its subset. Having too few basis functions also results in a drop in accuracy because of the lack of expressive power. In conclusion, we need to have enough basis functions so that we have enough expressive power, but we cannot have too many of them. This experiment suggests a way to find the correct basis set for the data. First, we try to find out how many basis functions should we

Basis functions	N			
	20	40	60	80
3 Gaussian fns	0.652 ± 0.044	0.734 ± 0.035	0.809 ± 0.032	0.836 ± 0.025
5 Gaussian fns	0.650 ± 0.054	0.731 ± 0.031	0.792 ± 0.032	0.829 ± 0.023
7 Gaussian fns	0.664 ± 0.051	0.748 ± 0.041	0.790 ± 0.035	0.835 ± 0.026
3 Linear fns	0.636 ± 0.050	0.736 ± 0.036	0.794 ± 0.031	0.825 ± 0.022
4 Linear fns	0.638 ± 0.041	0.735 ± 0.044	0.789 ± 0.031	0.823 ± 0.028

Table 5: Accuracies of the GNB with basis functions using other kind of basis funtions

use for the given data. We can use any reasonable functions, Gamma or Gaussian, for the first step. Keep in mind that the functions should be expressive enough to be able to represent the true basis functions. Next, once we know how many basis function are there, we fine-tune the form and the parameters of the basis functions to get the best performance. Although this method does not guarantee the best basis set, it is a fast and deterministic way to obtain a reasonable basis set. Figure 4 shows accuracies of the GNB with correct basis set at when we vary the number of training examples.

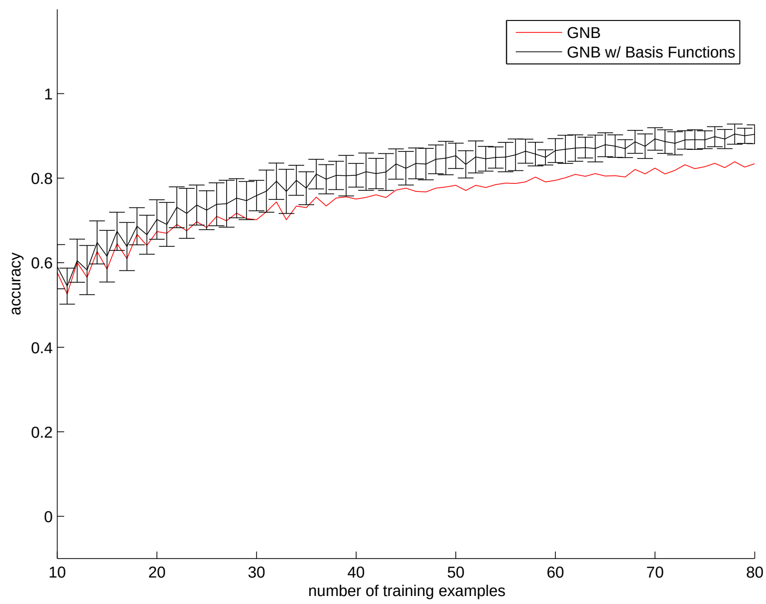

Figure 4: Comparison of accuracy between the two classifiers

3.2 Experiments with real data

Now, we are going to conduct an experiment using real fMRI data. The data is obtained from a study called "'StarPlus"'. On each trial, the subject was presented a sentence and a picture. Our goal is to identify if the subject is looking at a picture or a sentence. There are total of 80 labeled examples we can use. Half of them are from the picture class, and the rest are from the sentence class. We only consider the data from 40 most active voxels. The images were taken every 500 ms for 8 seconds. That means we have 16 data points for every voxel. We only use N examples to train the classifiers, and use the rest to test. We use the method suggested in the previous experiment to find the number and parameters of basis functions. The result is shown in Table 6-9.

	N			
Basis functions	10	15	20	25
normal GNB	0.688 ± 0.069	0.717 ± 0.053	0.754 ± 0.059	0.782 ± 0.054
1 Gamma fn	0.674 ± 0.087	0.730 ± 0.086	0.809 ± 0.055	0.806 ± 0.056
2 Gamma fns	0.682 ± 0.081	0.731 ± 0.069	0.780 ± 0.059	0.793 ± 0.053
3 Gamma fns	0.692 ± 0.086	0.752 ± 0.057	0.797 ± 0.053	0.809 ± 0.053
5 Gamma fns	0.691 ± 0.074	0.761 ± 0.045	0.773 ± 0.059	0.795 ± 0.059
7 Gamma fns	0.674 ± 0.058	0.736 ± 0.072	0.762 ± 0.065	0.793 ± 0.047

Table 6: Accuracies from the classifiers using fMRI data, varying the number of basis functions

	N		
Basis functions	30	35	40
normal GNB	0.759 ± 0.059	0.747 ± 0.061	0.778 ± 0.067
1 Gamma fn	0.814 ± 0.051	0.797 ± 0.057	0.803 ± 0.058
2 Gamma fns	0.817 ± 0.063	0.822 ± 0.062	0.811 ± 0.080
3 Gamma fns	0.801 ± 0.055	0.807 ± 0.052	0.803 ± 0.060
5 Gamma fns	0.801 ± 0.045	0.810 ± 0.055	0.809 ± 0.065
7 Gamma fns	0.773 ± 0.047	0.780 ± 0.050	0.785 ± 0.070

Table 7: Accuracies from the classifiers using real fMRI data, varying the number of basis functions (continued)

In general, we can see improvement over the traditional GNB. Especially, when $N = 35$, we have a clear improvement with 3 Gamma functions over the traditional GNB. As we can see from the result, we have comparable or

Basis functions	N			
	10	15	20	25
3 Gamma fns (2,2)(3,4)(2.5,6)	0.678 ± 0.063	0.750 ± 0.053	0.766 ± 0.058	0.793 ± 0.056
3 Gamma fns (1,5)(1,7)(1,9)	0.689 ± 0.078	0.751 ± 0.062	0.773 ± 0.061	0.773 ± 0.063
3 Gamma fns (2,1)(3,3)(4,5)	0.673 ± 0.080	0.760 ± 0.071	0.783 ± 0.050	0.804 ± 0.068
3 Gaussian fns	0.659 ± 0.076	0.731 ± 0.055	0.783 ± 0.064	0.805 ± 0.060

Table 8: Accuracies from the GNB with basis functions, varying the form of basis functions

Basis functions	N		
	30	35	40
3 Gamma fns (2,2)(3,4)(2.5,6)	0.786 ± 0.077	0.802 ± 0.050	0.806 ± 0.069
3 Gamma fns (1,5)(1,7)(1,9)	0.811 ± 0.048	0.830 ± 0.062	0.815 ± 0.060
3 Gamma fns (2,1)(3,3)(4,5)	0.805 ± 0.047	0.807 ± 0.055	0.804 ± 0.062
3 Gaussian fns	0.791 ± 0.065	0.825 ± 0.058	0.809 ± 0.051

Table 9: Accuracies from the GNB with basis functions, varying the form of basis functions (continued)

better results when we use the GNB with basis functions. An explanation for this phenomenon is that the GNB with basis functions needs fewer parameters to estimate, compared to the traditional GNB. The result suggests that, in the case of limited training examples, the GNB with basis functions is a better choice for this learning task. Nonetheless, from a theoretical point of view, this should apply only when we have a small N. When N is large enough, the traditional GNB should be able to estimate all the parameters it needs, and it should be able to perform at the same or higher level as the GNB with basis functions. Figure 5 compares the accuracies of both classifiers when we vary the number of training examples.

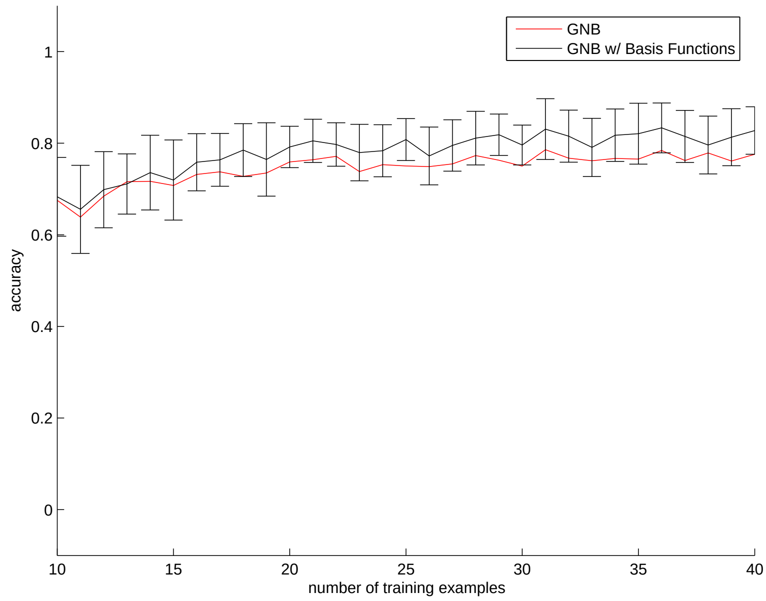

Figure 5: Accuracies of the two classifiers on real data, using 3 Gamma functions as basis functions

4 Conclusion and future work

Motivated by the hemodynamic response, we extend the traditional GNB by adding smooth basis functions. There are two main benefits of using basis functions. First, basis functions can help capture the smoothness feature in the time-series data. Second, we reduce the number of parameters we need to estimate. As the result of the experiments, we see improvement with the GNB with basis functions over the traditional GNB when the number of training examples is small. However, we need to consider about the number and the shapes of basis functions for each data set. We suggest a fast and deterministic method to find out what they are, but it does not guarantee the best set of basis functions. One suggestion is to estimate the number and the parameters of basis functions by using EM algorithm. The idea of using basis functions is not limited only to fMRI data. We can apply the GNB with basis functions with other time-series data as well.