Assignment 1. Bayesian Classifier

Li Yuhui 2013012470

Experiment Procedure

For each point, calculate the discriminant value as scheme 1.

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}ln2\pi - \frac{1}{2}\ln|\Sigma_i| + lnP(w_i)$$

$$(Scheme\ 1)$$

Where,

$$\mu_i = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\Sigma_i = XX^T$$

Where n is the number of samples, m is the number of feature (dimension).

Then the classification of samples becomes

$$\mathrm{argmax}_i g_i(x)$$

$$(Scheme\ 2)$$

To further describe the empirical training error, Bhattacharyya error bound is calculated as scheme 3.

$$\mathrm{error} = \sqrt{P(w_1)P(w_2)}\exp(-k\left(\frac{1}{2}\right))$$

$$(Scheme\ 3)$$

Where,

$$\mathrm{k}\left(\frac{1}{2}\right) = \frac{1}{8}(\mu_2-\mu_1)^t\left[\frac{\Sigma_2-\Sigma_1}{2}\right]^{-1}(\mu_2-\mu_1) + \frac{1}{2}\ln\frac{|\frac{\Sigma_2-\Sigma_1}{2}|}{\sqrt{|\Sigma_2||\Sigma_1|}}$$

Additionally, random samples are required to be generated according to a normal distribution and be visualized, and the "*currency of guessing*" mechanism is required to be explained in Chinese, which are both shown in Appendix.

Experiment Setup

All the codes are written in python 2.7 and experiment on windows 10. Useful packages are used including numpy and matplotlib. Codes are submitted in supplementary file.

Experiment Result

Based on scheme 1, scheme 2, scheme 3, experiments are done using 1, 2 and 3 features (x1, x2, x3) of ten samples to build a dichotomizer for class w1, w2 and w1, w3. The result is shown as Table 1, Table 2, Table 3, Table 4, Table 5, Table 6.

Table 1. Classification of one-feature Samples in class w1, w2.

Samples	w1		w2	
	x1	Predicted	x1	Predicted
1	-5.01	w1	-0.91	w1
2	-5.43	w2	1.30	w1

3	1.08	w1	-7.75	w2
4	0.86	w1	-5.47	w2
5	-2.67	w1	6.14	w2
6	4.94	w2	3.60	w1
7	-2.51	w1	5.37	w2
8	-2.25	w1	7.18	w2
9	5.56	w2	-7.39	w2
10	1.03	w1	-7.50	w2
Empirical Training Error	30%			
Bhattacharyya Error Bound	48.68%			

Table 2. Classification of two-feature Samples in class w1, w2.

Samples	w1			w2		
	x1	x2	Predicted	x1	x2	Predicted
1	-5.01	-8.12	w1	-0.91	-0.18	w1
2	-5.43	-3.48	w2	1.30	-2.06	w1
3	1.08	-5.52	w1	-7.75	-4.54	w2
4	0.86	-3.78	w1	-5.47	0.50	w2
5	-2.67	0.63	w2	6.14	5.72	w2
6	4.94	3.29	w2	3.60	1.26	w1
7	-2.51	2.09	w2	5.37	-4.63	w2
8	-2.25	-2.13	w1	7.18	1.46	w2
9	5.56	2.86	w2	-7.39	1.17	w2
10	1.03	-3.33	w1	-7.50	-6.32	w1
Empirical Training Error	45%					
Bhattacharyya Error Bound	47.98%					

Table 3. Classification of three-feature Samples in class w1, w2.

Samples	w1				w2			
	x1	x2	x3	Predicted	x1	x2	x3	Predicted
1	-5.01	-8.12	-3.68	w1	-0.91	-0.18	-0.05	w2
2	-5.43	-3.48	-3.54	w1	1.30	-2.06	-3.53	w2
3	1.08	-5.52	1.66	w1	-7.75	-4.54	-0.95	w2
4	0.86	-3.78	-4.11	w1	-5.47	0.50	3.92	w2
5	-2.67	0.63	7.39	w2	6.14	5.72	-4.85	w2
6	4.94	3.29	2.08	w1	3.60	1.26	4.36	w1
7	-2.51	2.09	-2.59	w1	5.37	-4.63	-3.65	w2
8	-2.25	-2.13	-6.94	w1	7.18	1.46	-6.66	w2
9	5.56	2.86	-2.26	w2	-7.39	1.17	6.30	w2

10	1.03	-3.33	4.33	w1	-7.50	-6.32	-0.31	w2
Empirical Training Error	15%							
Bhattacharyya Error Bound	45.38%							

Table 4. Classification of one-feature Samples in class w1, w3.

Samples	w1		w3	
	x1	Predicted	x1	Predicted
1	-5.01	w1	5.35	w3
2	-5.43	w1	5.12	w3
3	1.08	w1	-1.34	w1
4	0.86	w1	4.48	w3
5	-2.67	w1	7.11	w3
6	4.94	w3	7.17	w3
7	-2.51	w1	5.75	w3
8	-2.25	w1	0.77	w1
9	5.56	w3	0.90	w1
10	1.03	w1	3.52	w3
Empirical Training Error	25%			
Bhattacharyya Error Bound	44.69%			

Table 5. Classification of two-feature Samples in class w1, w3.

Samples	w1			w3		
	x1	x2	Predicted	x1	x2	Predicted
1	-5.01	-8.12	w1	5.35	2.26	w3
2	-5.43	-3.48	w1	5.12	3.22	w3
3	1.08	-5.52	w1	-1.34	-5.31	w1
4	0.86	-3.78	w1	4.48	3.42	w3
5	-2.67	0.63	w1	7.11	2.39	w3
6	4.94	3.29	w3	7.17	4.33	w3
7	-2.51	2.09	w1	5.75	3.97	w3
8	-2.25	-2.13	w1	0.77	0.27	w1
9	5.56	2.86	w3	0.90	-0.43	w3
10	1.03	-3.33	w1	3.52	-0.36	w3
Empirical Training Error	20%					
Bhattacharyya Error Bound	40.57%					

Table 6. Classification of three-feature Samples in class w1, w3.

Samples	w1				w3			
	x1	x2	x3	Predicted	x1	x2	x3	Predicted
1	-5.01	-8.12	-3.68	w1	5.35	2.26	8.13	w3
2	-5.43	-3.48	-3.54	w1	5.12	3.22	-2.66	w3
3	1.08	-5.52	1.66	w1	-1.34	-5.31	-9.87	w3
4	0.86	-3.78	-4.11	w1	4.48	3.42	5.19	w3
5	-2.67	0.63	7.39	w1	7.11	2.39	9.21	w3
6	4.94	3.29	2.08	w3	7.17	4.33	-0.98	w3
7	-2.51	2.09	-2.59	w1	5.75	3.97	6.65	w3
8	-2.25	-2.13	-6.94	w1	0.77	0.27	2.41	w1
9	5.56	2.86	-2.26	w3	0.90	-0.43	-8.71	w3
10	1.03	-3.33	4.33	w1	3.52	-0.36	6.43	w3
Empirical Training Error	15%							
Bhattacharyya Error Bound	38.69%							

Based on the experiments above, the curve of empirical training error and Bhattacharyya error bound for different feature Numbers are shown in Figure 2, Figure 3.

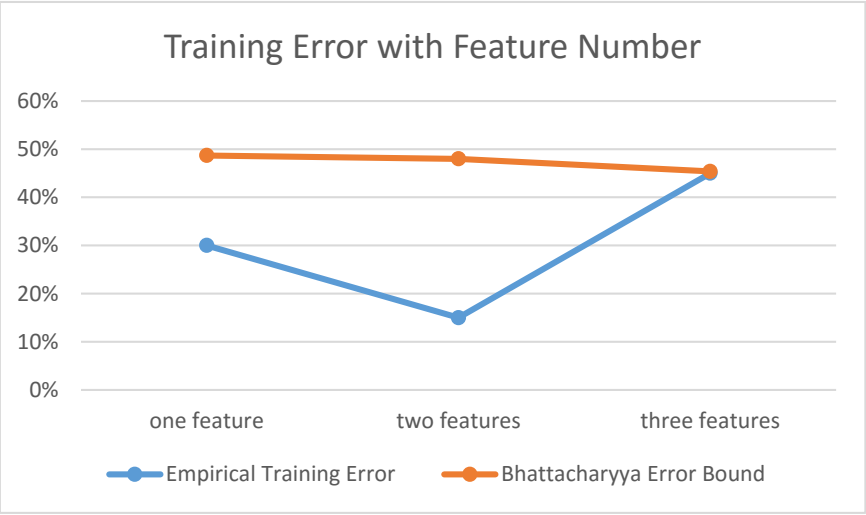

Figure 1. The Empirical Training error and Bhattacharyya Error Bound for Different Feature Numbers of Class w1 and w2.

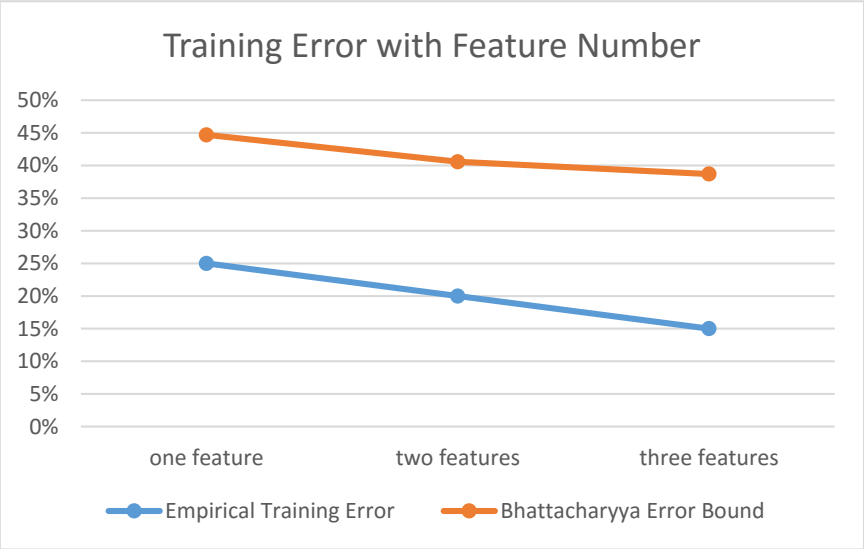

Figure 2. The Empirical Training error and Bhattacharyya Error Bound for Different Feature Numbers of Class w1 and w3.

Also, three categories are considered and the experiment result is shown as Table 7.

Table 7. Classification of three-feature Samples in Three Classes

Sample	x1	x2	x3	Mahalanobis Distance			Classification	
				w1	w2	w3	1	2
1	1.00	2.00	1.00	1.01	0.86	2.67	2	1
2	5.00	3.00	2.00	1.56	1.76	0.65	3	1
3	0	0	0	0.49	0.27	2.24	1	1
4	1.00	0	0	0.49	0.45	1.46	1	1

Actually, from the point of view of Bhattacharyya error bound (see Fig.1, Fig. 2), the error bound should decrease as the data dimensions of data increase. However, in practical, the empirical error might be larger (classification problem of w1 and w2, Fig.1) or smaller (classification problem of w1 and w3, Fig.3) as the data dimension increases. The empirical error defined in Scheme 4.

$$\hat{e}(h) = \frac{1}{m}\sum_{i=1}^{m}\{h(x^{(i)}) \neq y^{(i)}\}$$

(Scheme 4)

Where, m is the number of training data.

Based on the definition, the ambiguous result may attribute to two possible reasons. One is the small amount of data. As there are only 10 samples for training and 20 samples to calculate empirical error, one error may contribute significantly to total empirical error (For example, one error point contributes 5% to empirical error in this problem). Therefore, more amount of data is required to intensify the robustness of the empirical error. The other reason may be the feature extraction problem. When dimension of features becomes large as number of samples is relatively

small, overfitting may occur. Or sometimes there exists redundant dimension of feature, which interfere with the training machine. Therefore, the other solution is to decide which dimension of feature is most important and wipe off other dimensions.

To extend the problem of empirical error, scheme 5 is firstly introduced as follows.

$$\mathrm{R}(w) \leq Remp(w) + \emptyset(n/h)$$

(Scheme 5)

Where, R(w) is the expectation error. Remp(w) is the empirical error. $\emptyset(n/h)$ is the confidence interval, which is related to VC dimension and number of data, and decreases as n/h increases.

For each machine learning model, the aim is to minimize R(w). But in reality, as R(w) is hard to be calculated, we try to minimize $\emptyset(n/h)$ and see Remp(w) as the estimate of R(w). This means that number of data should be increased and VC dimension of model (i.e. complexity of model) should be decreased. Therefore, in this problem, as the VC dimension of model (Bayesian discriminant) seems to be small enough, the number of training samples should be enlarged to ensure better generalization.

Appendix

1. **Write a procedure to generate random samples according to a normal distribution N(μ,Σ) in d dimensions, and try to visualize an example in 3D space (take 3 dimension as example) .**

   As the p.d.f of a normal distribution N(μ,Σ) in d dimensions is shown in Scheme 6, random samples in 3D space is visualized as Fig. 3.

$$\mathrm{N}(\mathrm{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\}$$

(Scheme 6)

   Where, μ is the mean and Σ is the covariance.

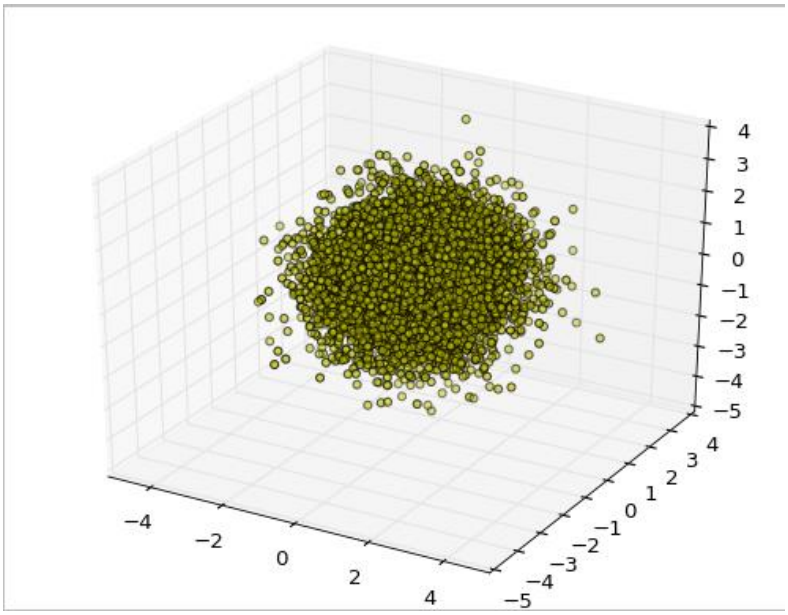

Figure 4. An Example of a Normal Distribution N(0,1) in 3 Dimensions. (10000 points)

2. **Read the following Nature Views "The currency of guessing" and write a paragraph in Chinese to explain the possible mechanisms/circuits of brain's Bayesian decision.**

这篇文章以猴子认知红、蓝形状为实验，试图论证脑神经在认知事物，尤其是对简单事物的决策时，更有可能仅仅取决于好的结果。并且，这一任务决策的方式与顶叶皮质区的神经元（LIP）有关。

当一个决策完全通过运动来表达的情况下，大脑结构中高水平控制的神经元调控条件和行为计划之间的逻辑联系。同样，在简单的决策任务中，当新的事物（符号）出现时，大脑会将事物中潜在的信息提取出来，并且神经活动会由于新的信息而产生改变。而这些累积的神经活动反应了对过去和新的信息的整合。从而基于这些先验知识，对未来类似的事物进行类似于贝叶斯决策（logLR）的后验判断。