1. In this problem, the cheeta and grass are two kind of random variables that has multinomial distribution:

$$P_{C_1,\ldots,C_n}=\frac{n!}{\prod_{k=1}^{N} c_k!}\prod_{j=1}^{N} \pi_j^{\,c_j} \tag{1}$$

So we can use the result of problem(2):

$$\pi_j^* = \frac{c_j}{n} \tag{2}$$

Then we can get the prior:

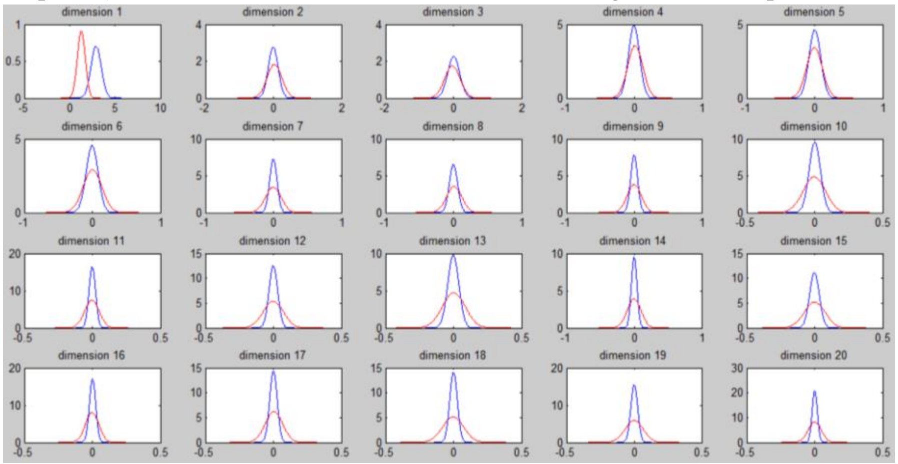
$$P_Y(cheeta) = \frac{250}{1053+250} = 0.1919$$

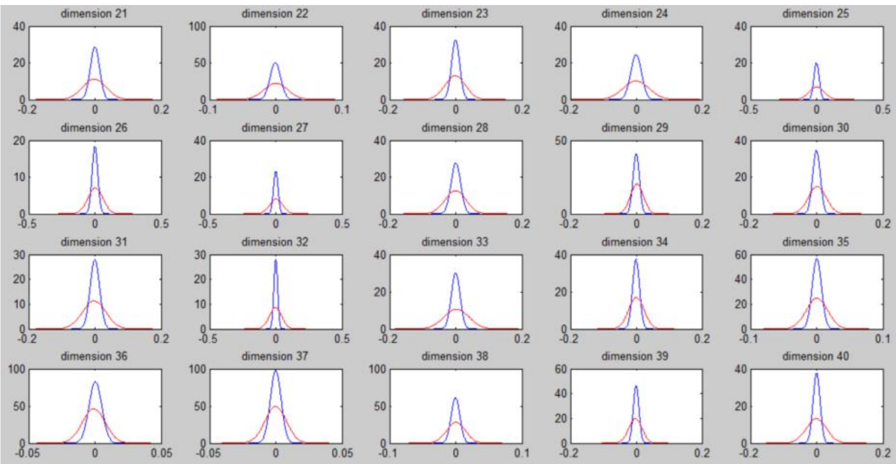
$$P_Y(grass) = \frac{1053}{1053+250} = 0.8081$$

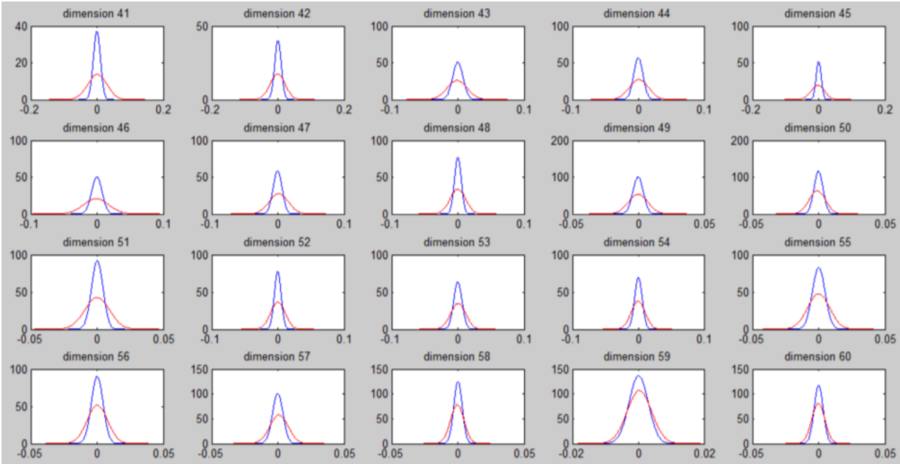
The result is same to the result last week.

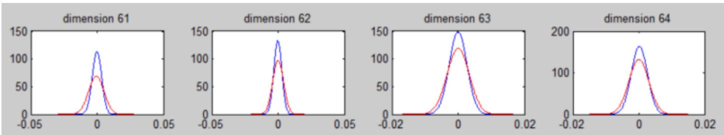
Interpretation: Last week, we just calculate the total number of samples, the ratio of cheeta and grass in the training sample. This is just an intuitive analysis of the training set. This time we have to ML estimate the prior π_{cheeta} and π_{grass}. We use prior as parameter and the frequency number of cheeta and grass as training sample data to maximize the probability(1). Then the result(2) is the ML estimation of the prior. Although it seems same like what we did last week, this time we prove that the frequency is the proper estimation of the prior for this kind of multinomial model.

2. 64 plots (blue line is the Gaussian distribution of background, red represents cheeta):



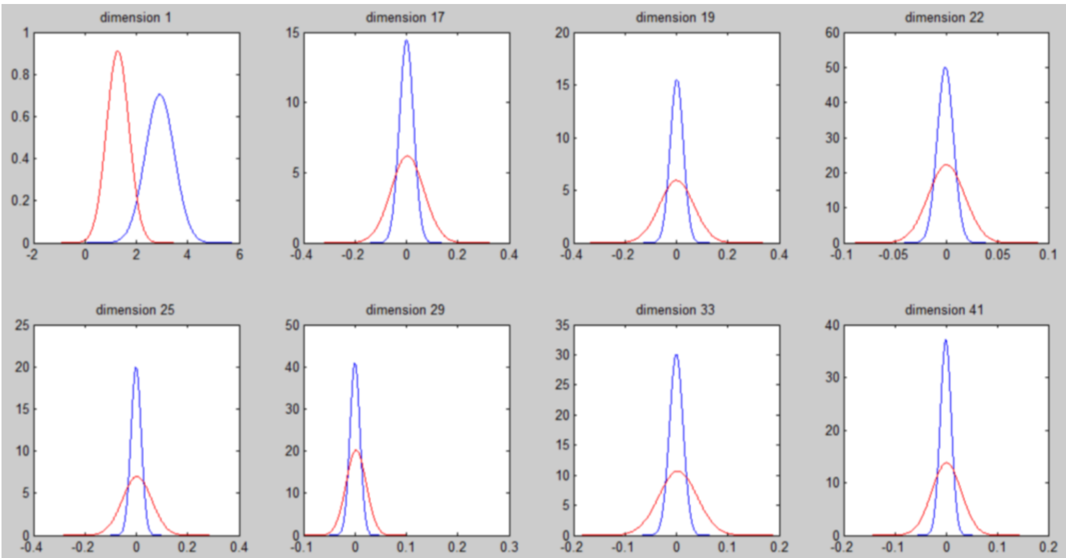




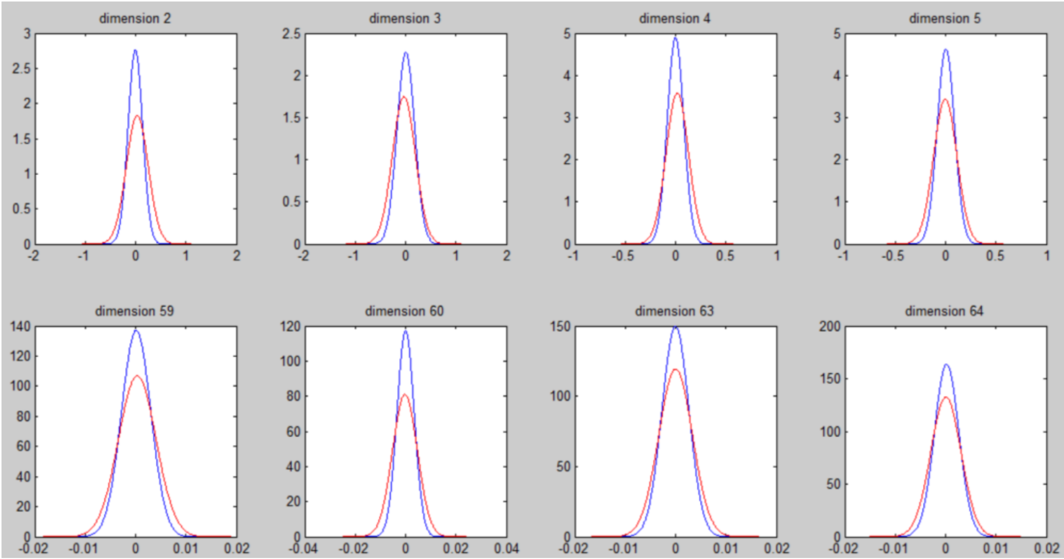


I evaluate the dimension by the area of overlap. Those distribution overlap regions mean that 2 class both have a lot probability, so classifiers cannot classify classes properly in those area. Those data sets have significant amount of overlap do not help or even have bad effect on the classification.

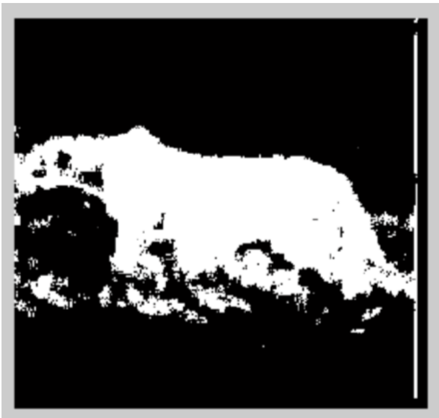
8 best:



8 worst: the 2nd to 8th graph show significant ratio of overlap.



3. (1)



Employ all 64 dimension in the classifier. Calculate mean and covariance for all dimensions, then calculate the posterior probabilities.

$$g_0(x) = \frac{1}{1 + exp\{d_0(x-\mu_0) - d_1(x-\mu_1) + \alpha_0 - \alpha_1\}}$$

Perror= Px|y(g(x) = cheetah|grass) * Py(grass) + Px|y(g(x) = grass|cheeta)

= 0.0855

(2)

We choose the set:

```
Choices = [1,17,19 ,22, 25,29,33,41];
```

Calculate mean and covariance (only represent the procedure of background samples here):

```
mus1 = sum(sampleBG)/1053;
sigs1 = cov(sampleBG);
```

Calculate g(x) to classify (count is the iterator to go through the data):

```
gBG(count) = 1/(1+exp(dxy(a(count, :), mus1, sigs1) -
dxy(a(count, :), mus2, sigs2) + alphaBG - alphaFG));
```

Result mask:

Calculate the probability of error:

Px|y(g(x) = cheetah|grass) = 0.0458

Px|y(g(x) = grass|cheeta) = 0.0942

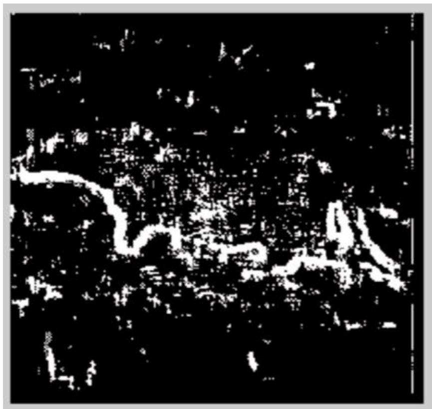
Perror= Px|y(g(x) = cheetah|grass) * Py(grass) + Px|y(g(x) = grass|cheeta) = 0.0942 * Py(cheeta)

= 0.0896 * 0.1919 + 0.0401 * 0.8081

= 0.0496

The selected 8 dimension result is better than the total 64 dimension result. This is because there are a lot of dimension's data set cannot classify two classes clearly and in 64 dimension, the classifier consider the all the 64 dimension (good and bad) together, then the bad dimension sets have negative effect on the whole classification. For 8 dimension, the classifier only consider those "seemed good" dimensions. Without other bad dimension's interfaces, the classifier can perform better.

To represent the negative effect, I exaggerate the problem and use worst 8 to do the classify. The result is this:



We can see that is almost like a random result except some trivial margin can be recognized. A great proportion of cheeta cannot be recognized.

Although the 64 dimension data reserves more information, but without our selection or some method as supervised learning, the procedure can never know which is useful, which has negative effect or which one contains the information we want, so the output of the 64 dimension learning cannot perform better than eight dimension for this problem.