機械学習特論

~理論とアルゴリズム~

(Linear Discriminant Analysis)

講師: 西郷浩人

Contents

- In the previous lecture we learned how to obtain maximum likelihood estimate from Gaussian distribution.
- In this lecture we learn how to classify data points that follow Gaussian distribution by using posterior probability.

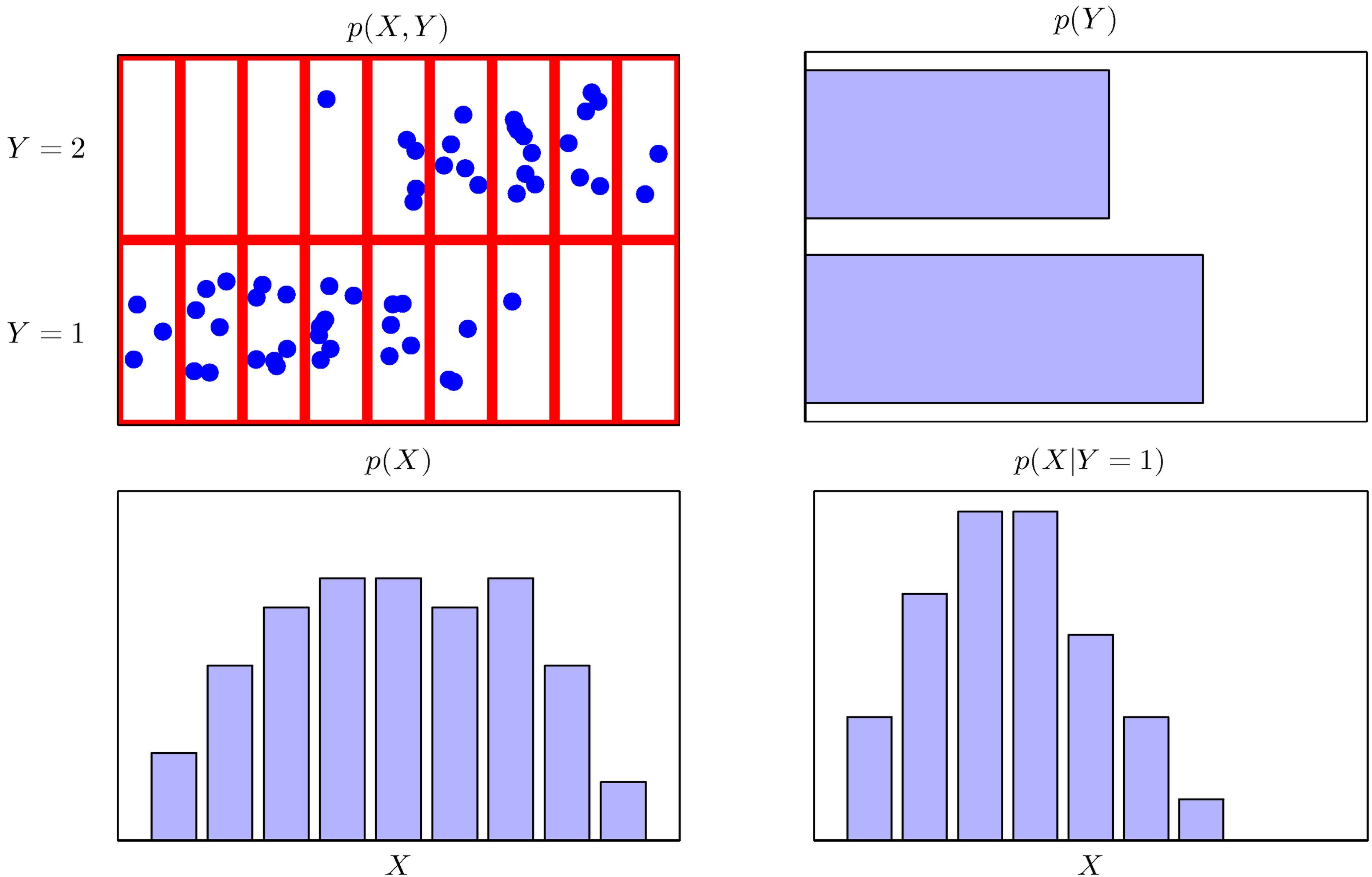
Estimate of posterior probability by generative approach

- First estimate conditional distribution p(X|Y), then estimate posterior probability p(Y|X) by using Bayes' theorem.

$$p(Y|X) \approx p(X|Y)\, p(Y)$$

we assume p(X|Y) follows Gaussian distribution.

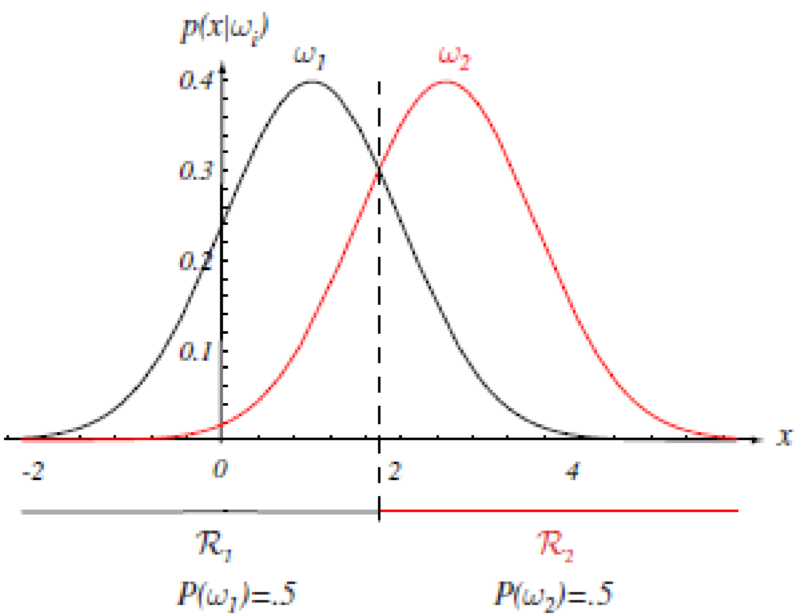
Probabilities

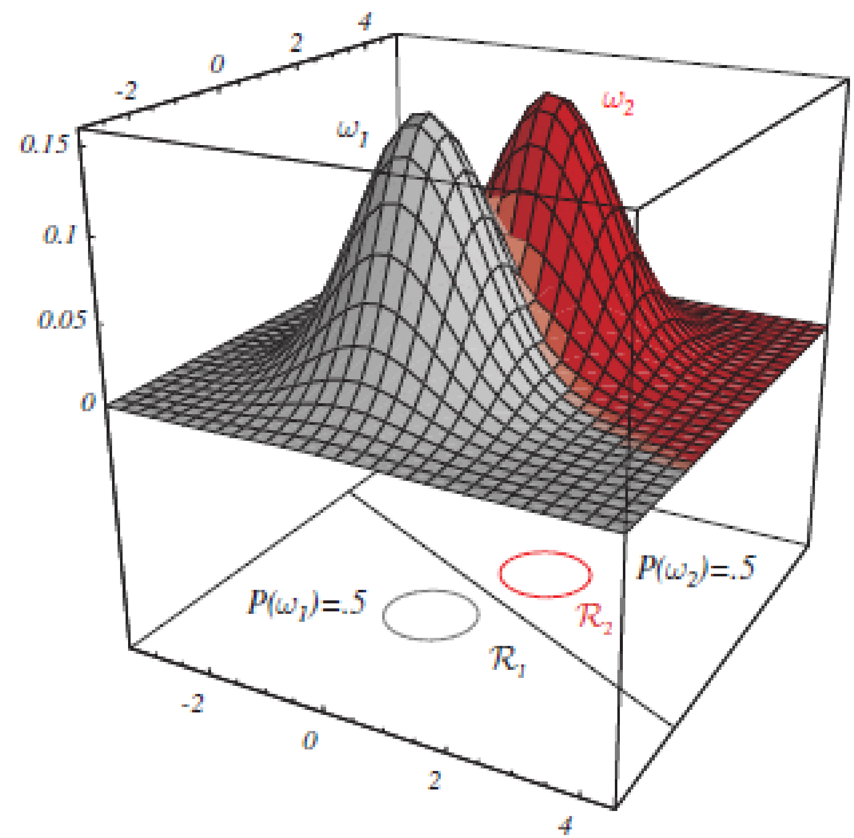


Assuming normality

- If p(y|x) follows Gaussian distribution, then the PDF of x can be written as below. (In the following example, we deal with binary classification problem.)

$$p(\boldsymbol{x}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(\frac{-1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_y)' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_y) \right)$$





Posterior probability

- Taking log of the Bayes' theorem $p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)\,p(y)}{p(\boldsymbol{x})}$

$$\log p(y|\boldsymbol{x}) = \log p(\boldsymbol{x}|y) + \log p(y) - \log p(\boldsymbol{x})$$

- Introducing normality assumption

$$p(\boldsymbol{x}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) = \frac{1}{(2\pi)^{D/2}} \frac{1}{\left|\boldsymbol{\Sigma}_y\right|^{1/2}} \exp\left(\frac{-1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_y)' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_y) \right)$$

$$\begin{aligned}
\log p(y|\boldsymbol{x}) &= \frac{-D}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma_y|) - \frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_y)' \boldsymbol{\Sigma}_y^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_y) + \log\left(\frac{n_y}{n}\right) - \log p(x) \\
&= \frac{-1}{2} \log(|\Sigma_y|) - \frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_y)' \boldsymbol{\Sigma}_y^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_y) + \log(n_y) + C
\end{aligned}$$

  – where $p(y) = \frac{n_y}{n}$ $\quad C = \frac{-D}{2} \log(2\pi) - \log n - \log p(\boldsymbol{x})$

Linear Discriminant Analysis (LDA)

$$\log p(y|\boldsymbol{x}) = \frac{1}{2}\log(|\Sigma_y|) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_y)'\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_y) + \log(n_y) + C$$

- Here we assume that covariance matrices are same for all ys:

$$\Sigma_{y=1} = \Sigma_{y=2} = \ldots = \Sigma_y$$

$$\log p(y|\boldsymbol{x}) = \frac{1}{2}\log(|\Sigma|) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_y)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_y) + \log(n_y) + C$$

- Since $\boldsymbol{x}'\Sigma^{-1}\boldsymbol{x}$ is independent from label y, so we can ignore it
  The resulting discriminant function is

$$\log p(y|\boldsymbol{x}) = \boldsymbol{x}'\Sigma^{-1}\boldsymbol{\mu}_y - \frac{1}{2}\boldsymbol{\mu}_y'\Sigma^{-1}\boldsymbol{\mu}_y + \log n_y + C'$$

where

$$C' = C + \frac{1}{2}\log(|\Sigma|)$$

Binary class classification by LDA

- Consider separating class 1 from class 2. One can calculate posterior probabilities of the both cases, and choose one with the larger probability.

$$\log p(y=1|\boldsymbol{x}) = \boldsymbol{x}'\Sigma^{-1}\boldsymbol{\mu}_{y=1} - \frac{1}{2}\boldsymbol{\mu}_{y=1}'\Sigma^{-1}\boldsymbol{\mu}_{y=1} + \log n_{y=1} + C'$$

$$\log p(y=2|\boldsymbol{x}) = \boldsymbol{x}'\Sigma^{-1}\boldsymbol{\mu}_{y=2} - \frac{1}{2}\boldsymbol{\mu}_{y=2}'\Sigma^{-1}\boldsymbol{\mu}_{y=2} + \log n_{y=2} + C'$$

- For classifying into two classes, it suffices to take a difference.

$$f(\boldsymbol{x}) = \log p(y=1|\boldsymbol{x}) - \log p(y=2|\boldsymbol{x})$$

$$= \boldsymbol{x}'\Sigma^{-1}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2}) - \frac{1}{2}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2})'\Sigma^{-1}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2}) + \log n_{y=1} - \log n_{y=2}$$

Example

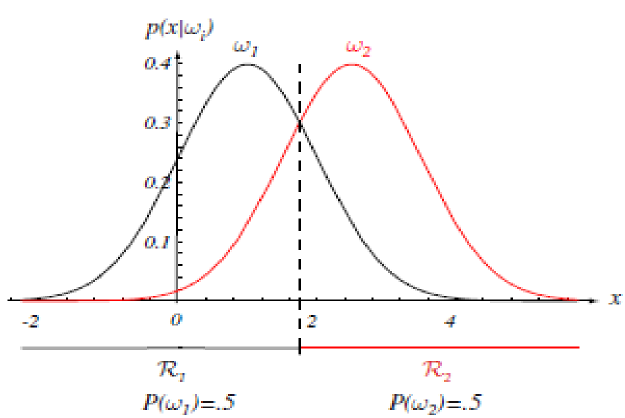
$$f(\boldsymbol{x}) = \boldsymbol{x}'\Sigma^{-1}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2}) - \frac{1}{2}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2})'\Sigma^{-1}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2}) + \log\frac{n_{y=1}}{n_{y=2}}$$

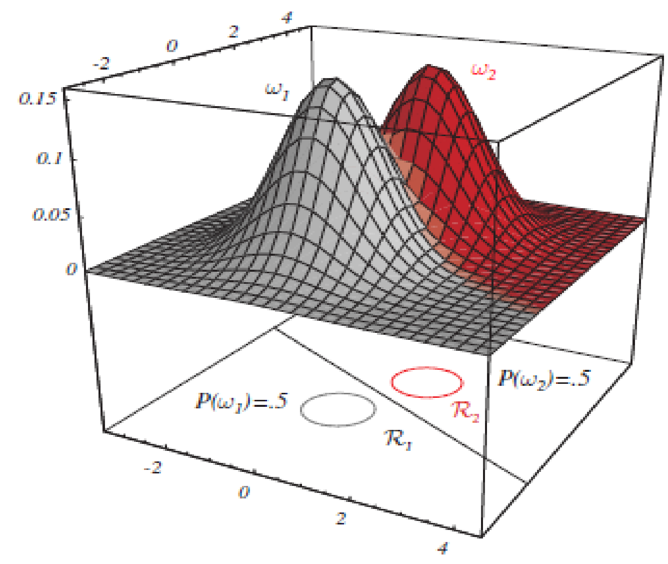
- The above function is a linear function $f(\boldsymbol{x}) = \boldsymbol{x}'\boldsymbol{a} + b$

$$\boldsymbol{a} = \Sigma^{-1}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2})$$

$$b = \frac{-1}{2}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2})'\Sigma^{-1}(\boldsymbol{\mu}_{y=1} - \boldsymbol{\mu}_{y=2}) + \log\frac{n_{y=1}}{n_{y=2}}$$

- The function is a linear function with respect to x, and the resulting separating hyperplane is linear.



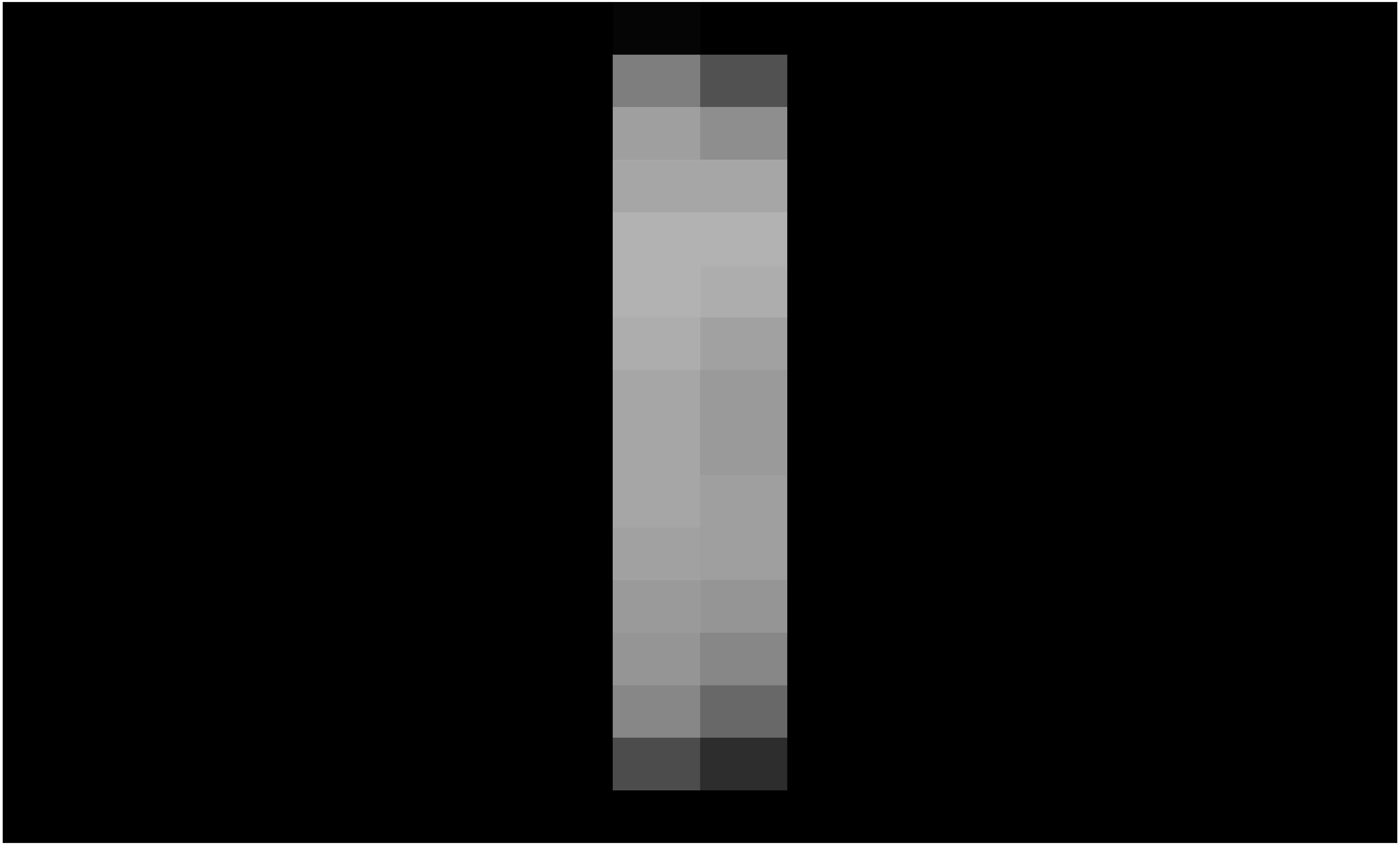


Classifying hand-written digits

- Download (handWrittenData)

```
load digit.mat
who %show variables
size(X) %show size
size(T) % show size
imshow(reshape(X(:,10,1),[16 16])');
```

  - X: 256 x 500 x 10 dimensional data
  - Let our data be X(a,b,c), then ‘a’ has color density on 16x16 pixels, ‘b’ has a data index (1:500). ‘c’ has a digit type. For example, X(:,10,1) correspond to “1” written by the 10th subject.
  - Display by “imshow(reshape(X(:,10,1),[16 16])')”
    - “reshape” converts a length 256 vector X(:,10,1) into a 16 x 16 matrix.



•

```
%mean of the features of a digit “1”
mu1=mean(X(:,:,1),2);
%mean of the features of a digit “2”
mu2=mean(X(:,:,2),2);
%covariance matrix common to both “1” and “2”
S=(cov(X(:,:,1)')+cov(X(:,:,2)'))/2;
%inverse of a matrix
invS=inv(S);
```

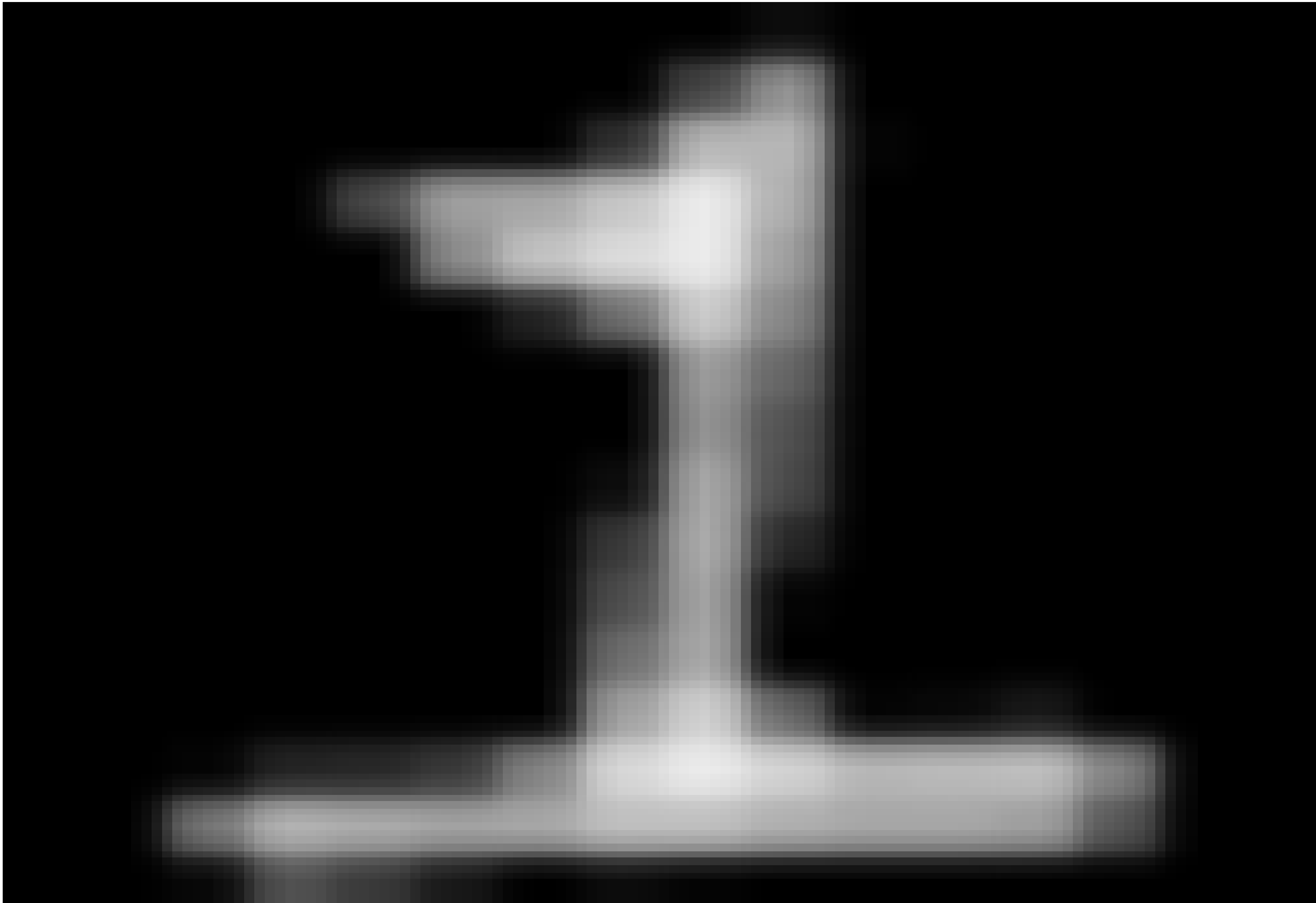
```
%prepare test data
t=T(:,:,1); % 200 “1”s
%posterior probability for “1”.
```

$$\log p(y=1|\boldsymbol{x}) = \boldsymbol{x}'\Sigma^{-1}\boldsymbol{\mu}_{y=1} - \frac{1}{2}\boldsymbol{\mu}_{y=1}'\Sigma^{-1}\boldsymbol{\mu}_{y=1}$$

```
p1=t'*invS*mu1 - mu1'*invS*mu1/2;
%posterior probability for “2”
```

$$\log p(y=2|\boldsymbol{x}) = \boldsymbol{x}'\Sigma^{-1}\boldsymbol{\mu}_{y=2} - \frac{1}{2}\boldsymbol{\mu}_{y=2}'\Sigma^{-1}\boldsymbol{\mu}_{y=2}$$

```
p2=t'*invS*mu2 - mu2'*invS*mu2/2;
result=sign(p1-p2) %”1” if positive, “2” otherwise
%number of correct answers
sum(result==1);
%rate of correct answers
sum(result==1)/length(result)
%displaying the misclassified digit
err=find(result~=1)
imshow(reshape(t(:,err),[16 16])')
```



Ex. 1

- Automatically discriminate the numbers by the 7-th subject. Which digit is misclassified to which one?
  - T(:, 7, :)
- Complete the following table, where the largest number corresponds to the prediction by LDA.

True

prediction

0	1	2	3	4	5	6	7	8	9	0
1	7296									
2	7268									
3	7267									
4	7270									
5	7261									
6	7248									
7	7271									
8	7270									
9	7278									
0	7244									

Hint

```
%Computation of a common covariance matrix
[a b c] = size(X);
S = zeros(a);
mu = zeros(a,c);
for i=1:c
 mu(:,i) = mean(X(:,:,i),2);
 S = S+cov(X(:,:,i)');
end
S=S/c;
invS = inv(S);
```

- X(:,:,i) is a matrix consisting of a digit “i” (256 pixels) from 500 subjects.

Hint 2

- Rather than taking difference between posterior probabilities of two classes, just compute posterior for each c class and store them in a table.

$$\log p(y=c|\boldsymbol{x}) = \boldsymbol{x}'\Sigma^{-1}\boldsymbol{\mu}_{y=c} - \frac{1}{2}\boldsymbol{\mu}_{y=c}'\Sigma^{-1}\boldsymbol{\mu}_{y=c}$$

Hint 3

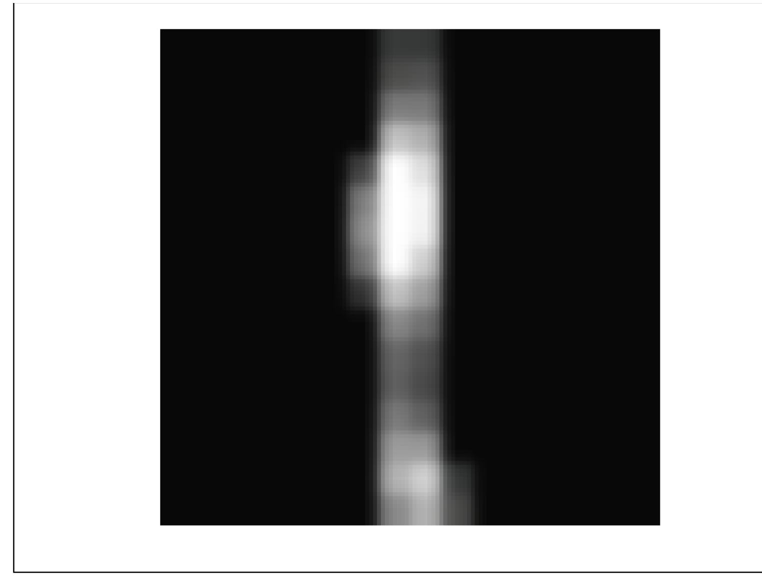
- When a table of posterior probabilities is computed, max function can return the largest probabilities of each column in the table.

```
[M I]=max(p)
```

Appendix

```
%displaying the feature
%that discriminates “1” from “2”

f12=(reshape(mu1-mu2,[16 16]));
imshow(f12')
```



```
%displaying the feature
%that discriminates “2” from “1”

f21=(reshape(mu2-mu1,[16 16]));
imshow(f21')
```