$$P(C|x_1, x_2) = \frac{P(x_1, x_2|C) \cdot P(C)}{P(x_1, x_2)}$$

$$= \frac{P(x_1|C) \cdot P(x_2|C) \cdot P(C)}{P(x_1, x_2)}$$

Lecture 8: Naive Bayesian & LDA

Pilsung Kang

School of Industrial Management Engineering

Korea University

AGENDA

01 Naive Bayesian Classifier

02 Linear Discriminant Analysis

03 R Exercise

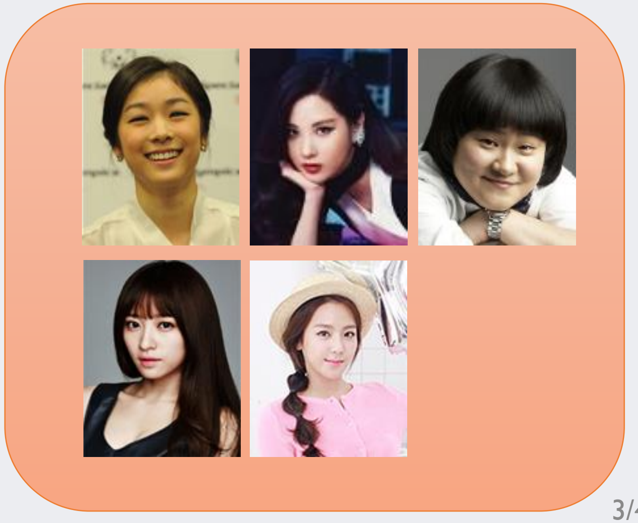
Naive Bayesian Classifier

- Classification revisited

Men

Vs.

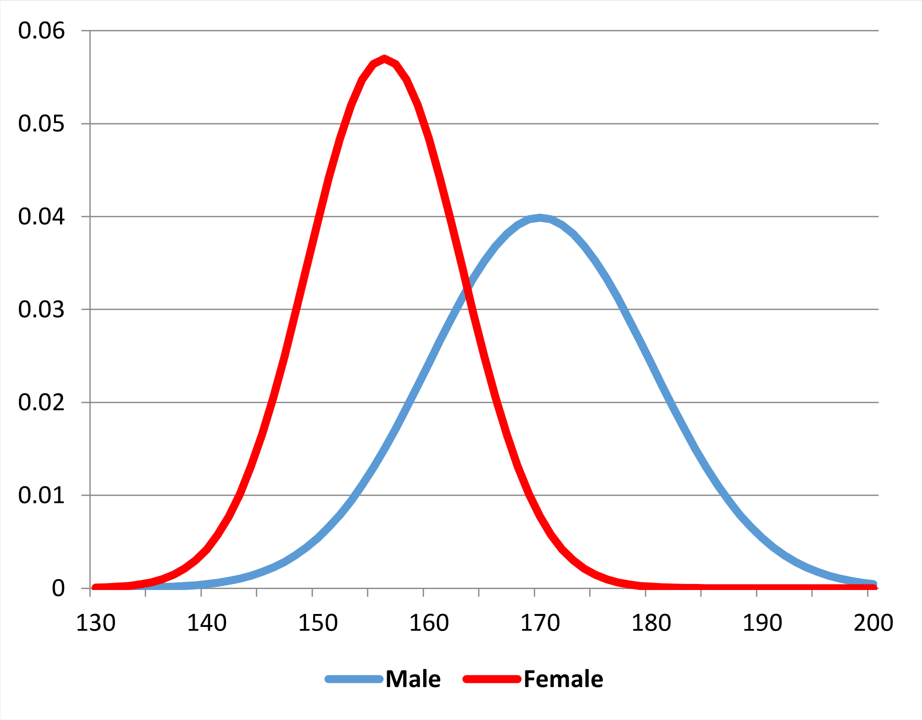
Women



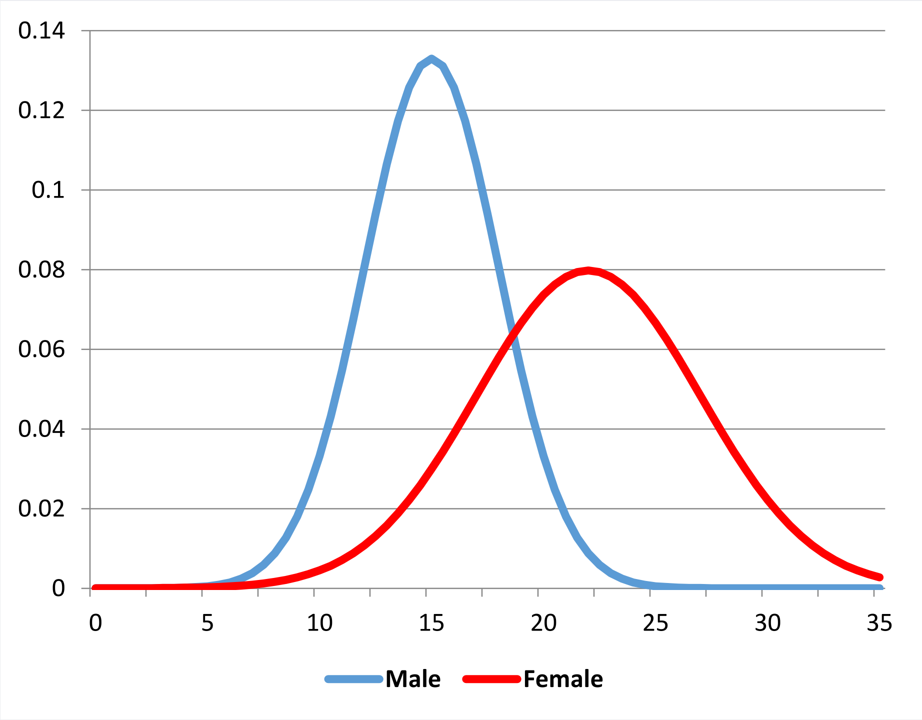
Naïve Bayesian Classification: Concept

- Assumption
  - ✓ There are two attributes: height & body fat percentage (BFP)
  - ✓ There are equal number of male and female
  - ✓ Actual probability distributions for all attribute & gender pairs are known

Height

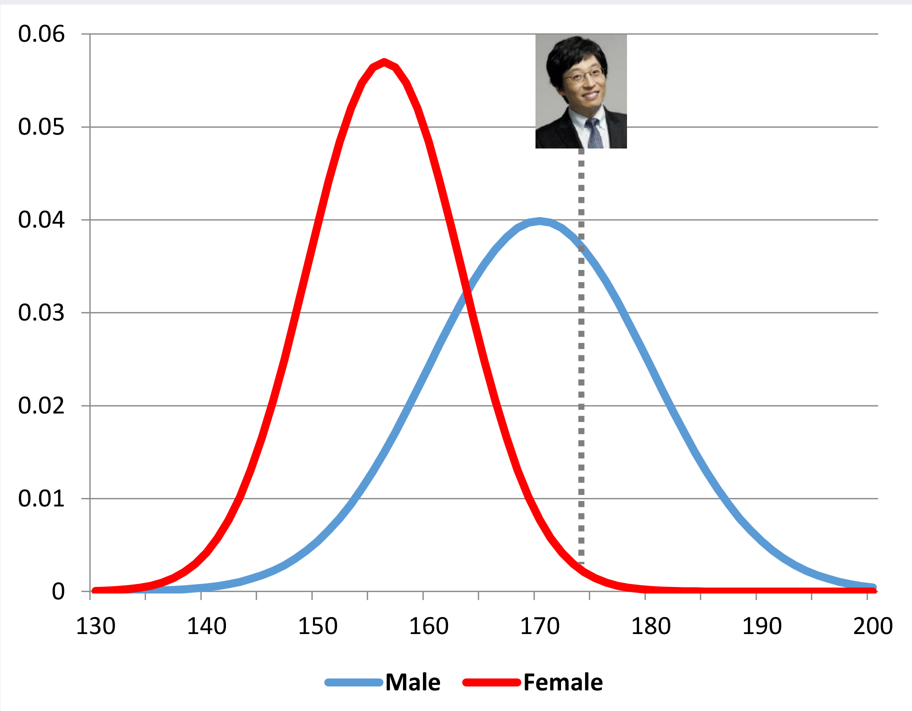


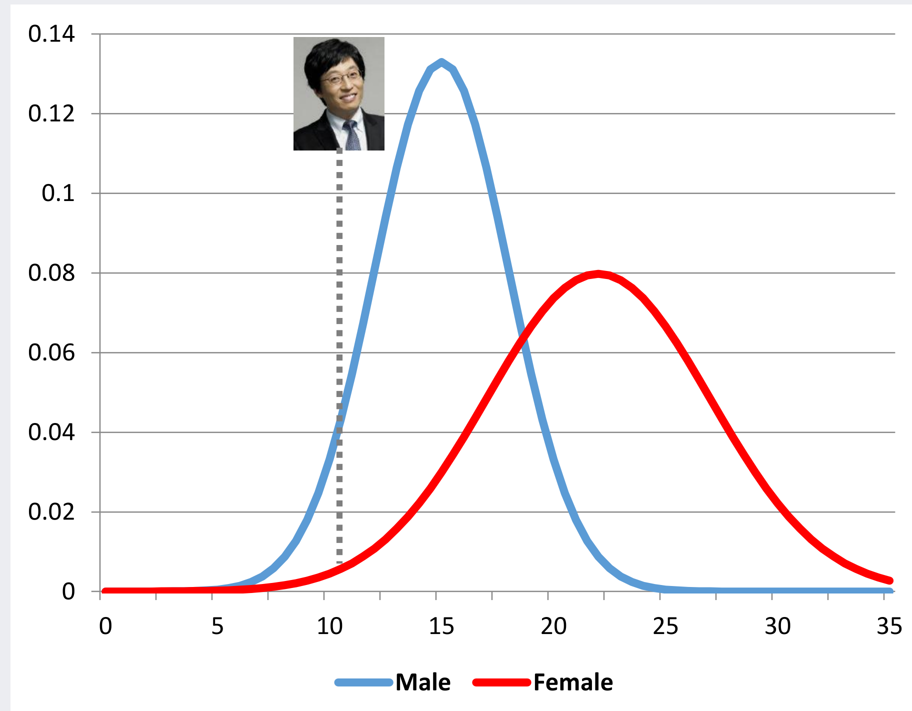
BFP



Naïve Bayesian Classification: Concept

- Let's classify  with the given information

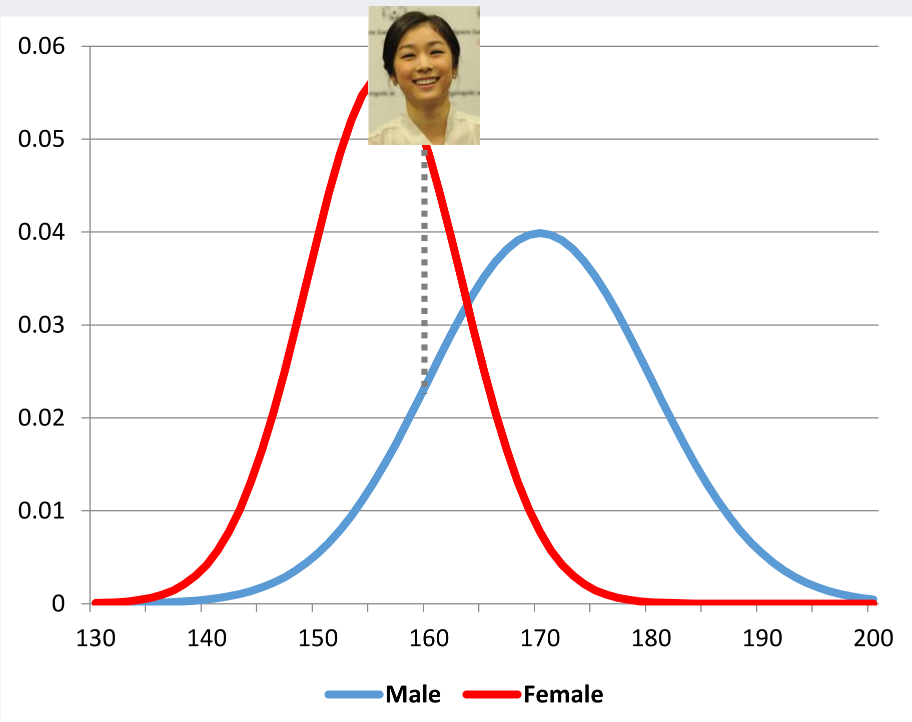


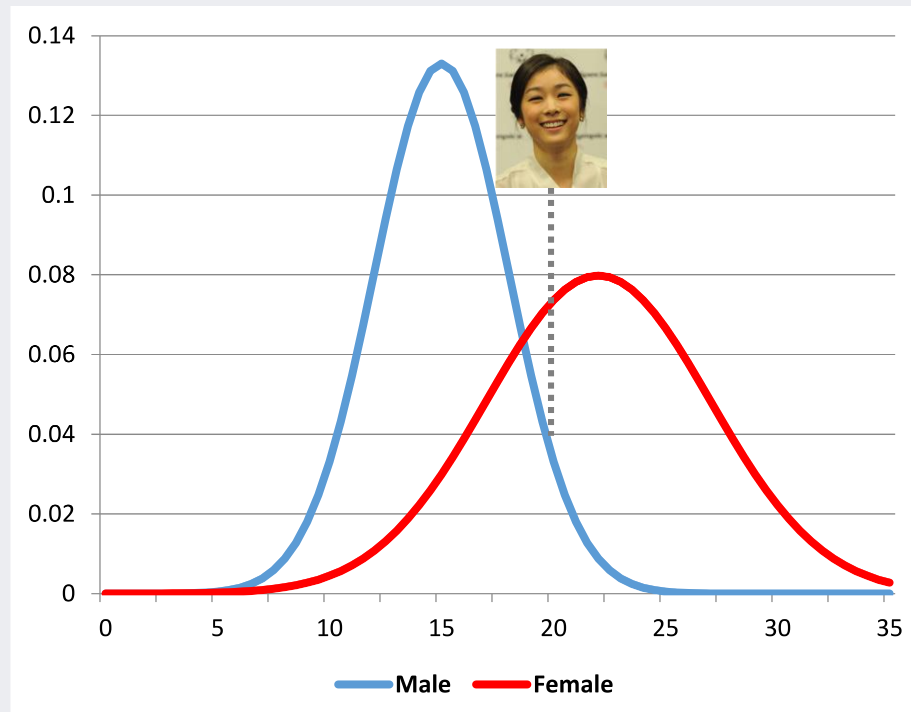


✓ Classify him as Male

Naïve Bayesian Classification: Concept

- Let's classify  with the given information

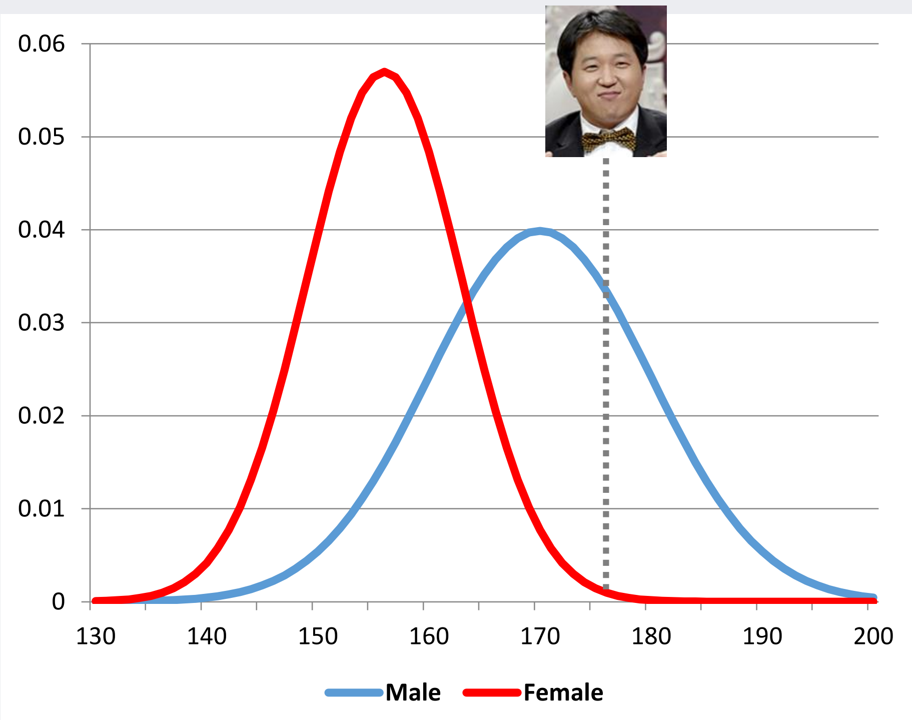


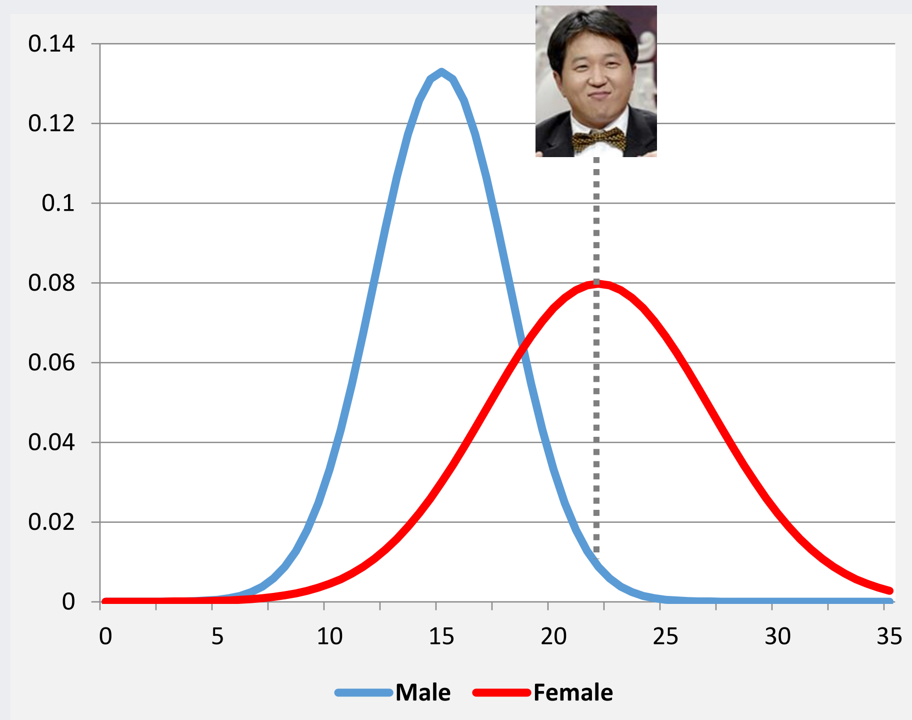


✓ Classify her as Female

Naïve Bayesian Classification: Concept

- What about  ?





✓ Classify this person as Male or Female???

Naïve Bayesian Classification: Theory

- Bay's Rule (one of the most important rules in statistics)

$$P(C_i|x_1, x_2) = \frac{P(x_1, X_2|C_i) \cdot P(C_i)}{P(x_1, x_2)}$$

- Naive: Let's assume that all variables are statistically independent to each other

$$= \frac{P(x_1|C_i) \cdot P(x_2|C_i) \cdot P(C_i)}{P(x_1, x_2)}$$

Naïve Bayesian Classification: Theory

- For the previous example, we should compare the following two probabilities

$$P(M|H,W,BFS) = \frac{P(H|M) \cdot P(W|M) \cdot P(BFS|M) \cdot P(M)}{P(H,W,BFS)}$$

$$P(F|H,W,BFS) = \frac{P(H|F) \cdot P(W|F) \cdot P(BFS|F) \cdot P(F)}{P(H,W,BFS)}$$

✓ Assign to the class with the highest posterior probability

Naïve Bayesian Classification

- Compute the posterior probabilities

$$P(H|M) \cdot P(W|M) \cdot P(BFS|M) = 0.035 \times 0.01 \times 0.5 = 0.000175$$

$$P(H|F) \cdot P(W|F) \cdot P(BFS|F) = 0.001 \times 0.08 \times 0.5 = 0.00004$$

- Classify the person as Male

Exact Bayesian Classifier

1

Find all the other records whose variable values are exactly identical to the test entity

- Find all the other people with the same height and BFS.

Person	Height	BFS	Class
홍길동	178	11	M
김영희	178	11	F
김철수	178	11	M
김가네	178	11	M

Variables are not assumed to be statistically independent

$$P(C_i \mid x_1, x_2, \ldots, x_d) = \frac{P(x_1, x_2, \ldots, x_d \mid C_i) P(C_i)}{P(x)}$$

Exact Bayesian Classifier

2

Find the prevalent class

- Determine what classes they all belong to and which class is more prevalent.

Person	Height	BFS	Class
홍길동	178	11	M
김영희	178	11	F
김철수	178	11	M
김가네	178	11	M

- 3 males and 1 female.

Exact Bayesian Classifier

3

Assign the prevalent to the new record

Person	Height	BFS	Class
홍길동	178	11	M
김영희	178	11	F
김철수	178	11	M
김가네	178	11	M

- 3 males and 1 female.
- He is classified as male.

Difficult to find the exact same records when the there are many attributes(features) with small number of training data.

Naive Bayesian Classification: Procedure

1

Prepare the training data

- Define attributes and collect training data
  - ✓ Total training data: 200 (100 males, 100 females)
  - ✓ Height & BFS

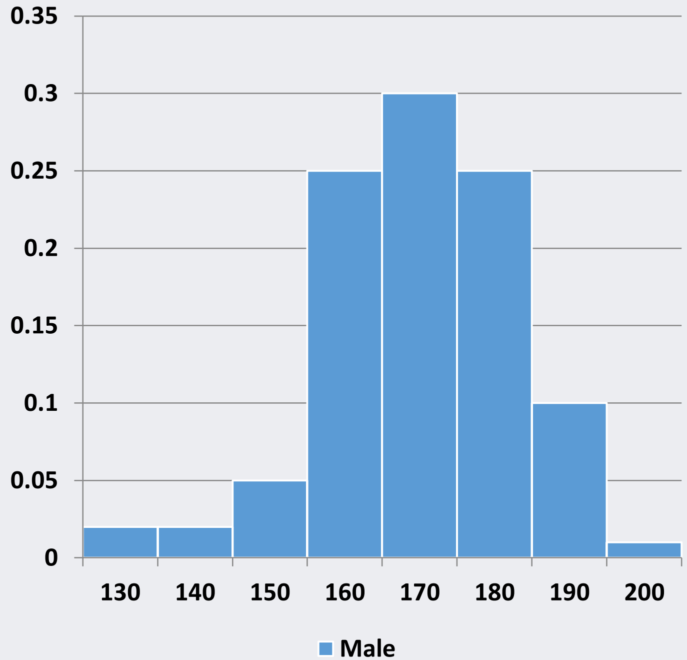
Record	Height	BFS	Class
1	187	15	M
2	165	25	F
3	174	14	M
4	156	29	F
…	…	…	…
N	168	12	M

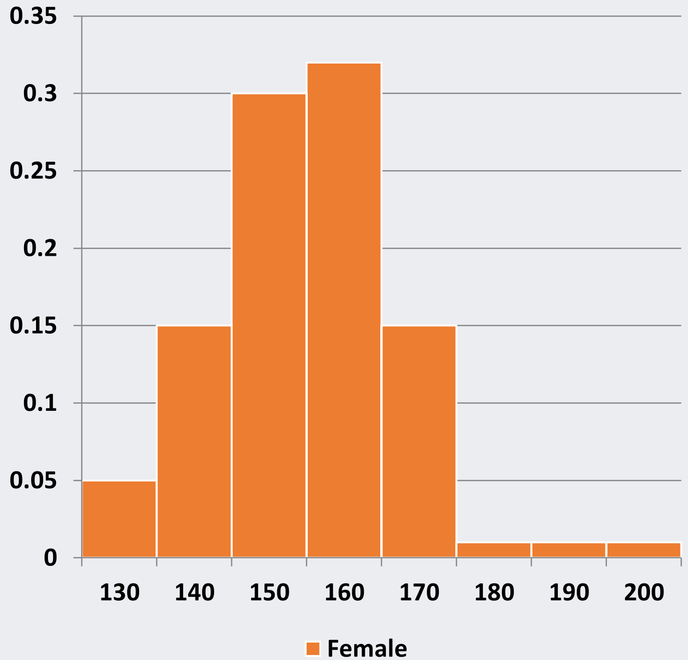
Naive Bayesian Classification: Procedure

2

Estimate the probability distribution

- Estimate the probability distribution of the attributes for each class.
- Height



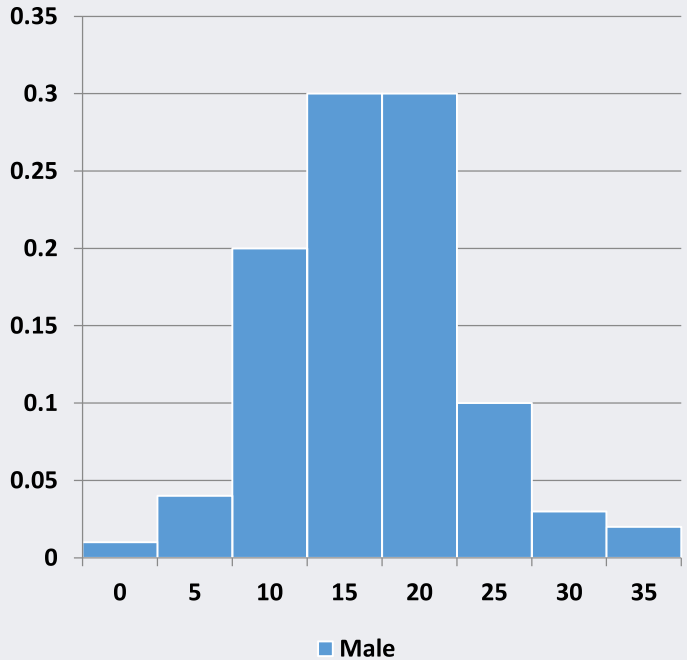


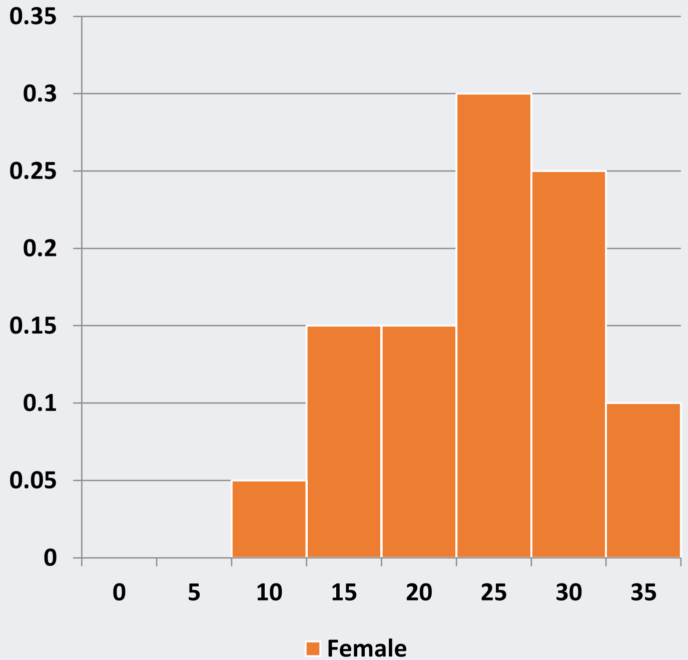
Naive Bayesian Classification: Procedure

2

Estimate the probability distribution

- Estimate the probability distribution of the attributes for each class.
- BFP



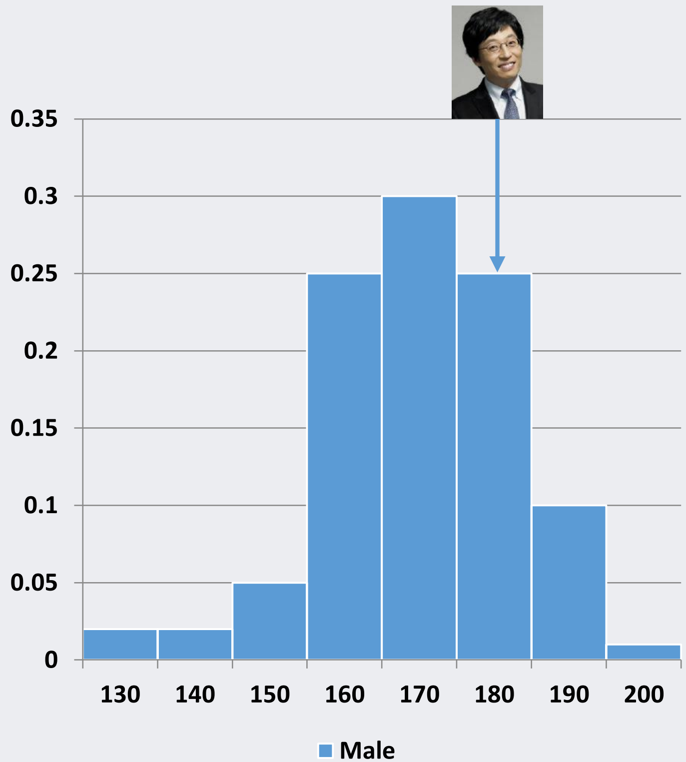


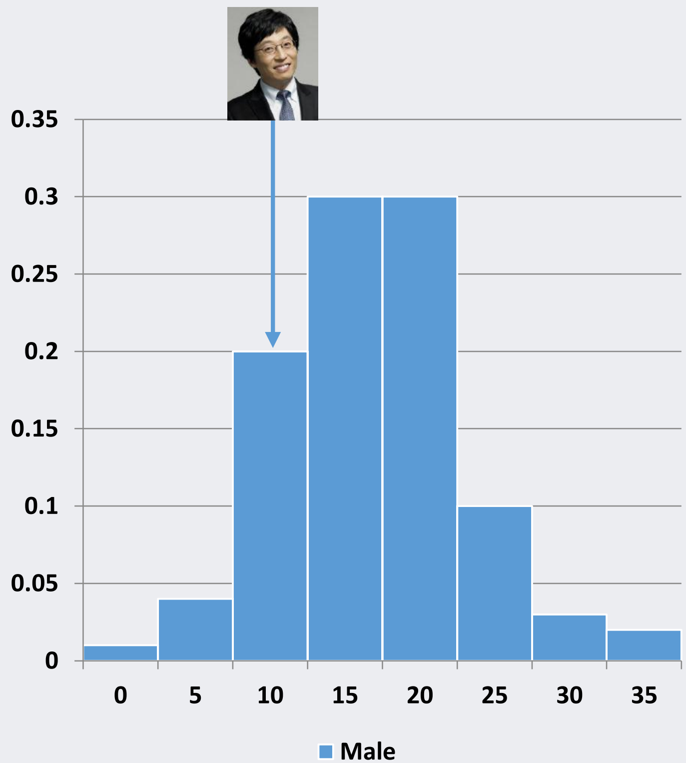
Naive Bayesian Classification: Procedure

3

Compute the conditional probability for each attribute

- P(Height = 178 | Male) = 0.25, P(BFP = 11 | Male) = 0.2



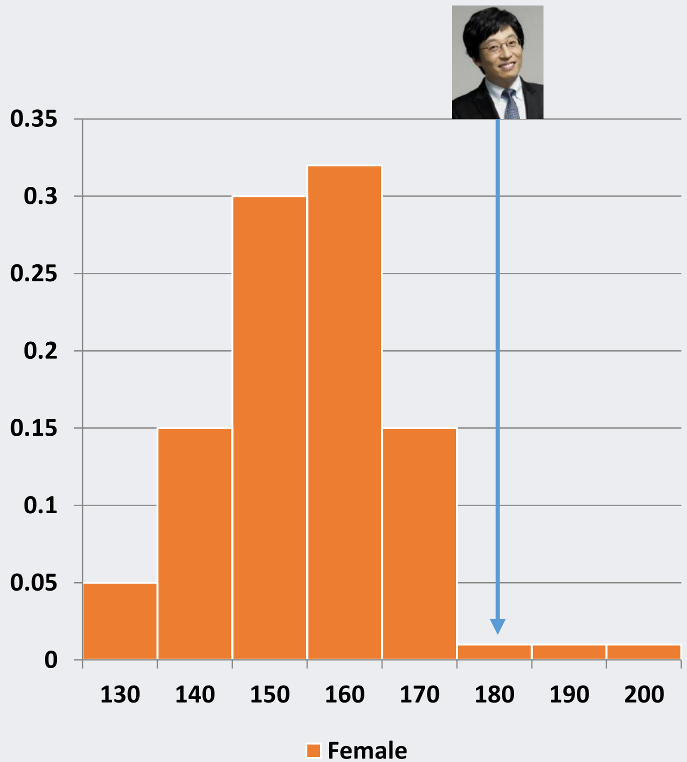


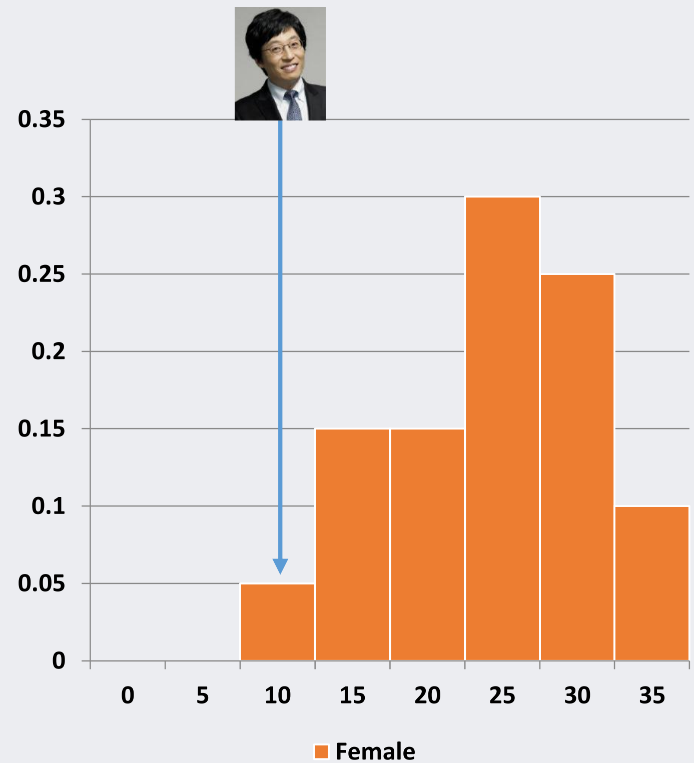
Naive Bayesian Classification: Procedure

3

Compute the conditional probability for each attribute

- P(Height = 178 | Female) = 0.01, P(BFP = 11 | Female) = 0.05





Naive Bayesian Classification: Procedure

4

Compute the posterior probability

- Compute the posterior probability for each class
  - ✓ P(Height = 178, BFP = 11 | Male)*P(Male)

    = P(Height = 178 | Male)* P(BFP = 11 | Male) *P(Male)

    = 0.25*0.2*0.5 = 0.025
  - ✓ P(Height = 178, BFP = 11 | Female) *P(Female)

    = P(Height = 178 | Female)* P(BFS = 11 | Female) *P(Female)

    = 0.01*0.05*0.5 = 0.00025

Naive Bayesian Classification: Procedure

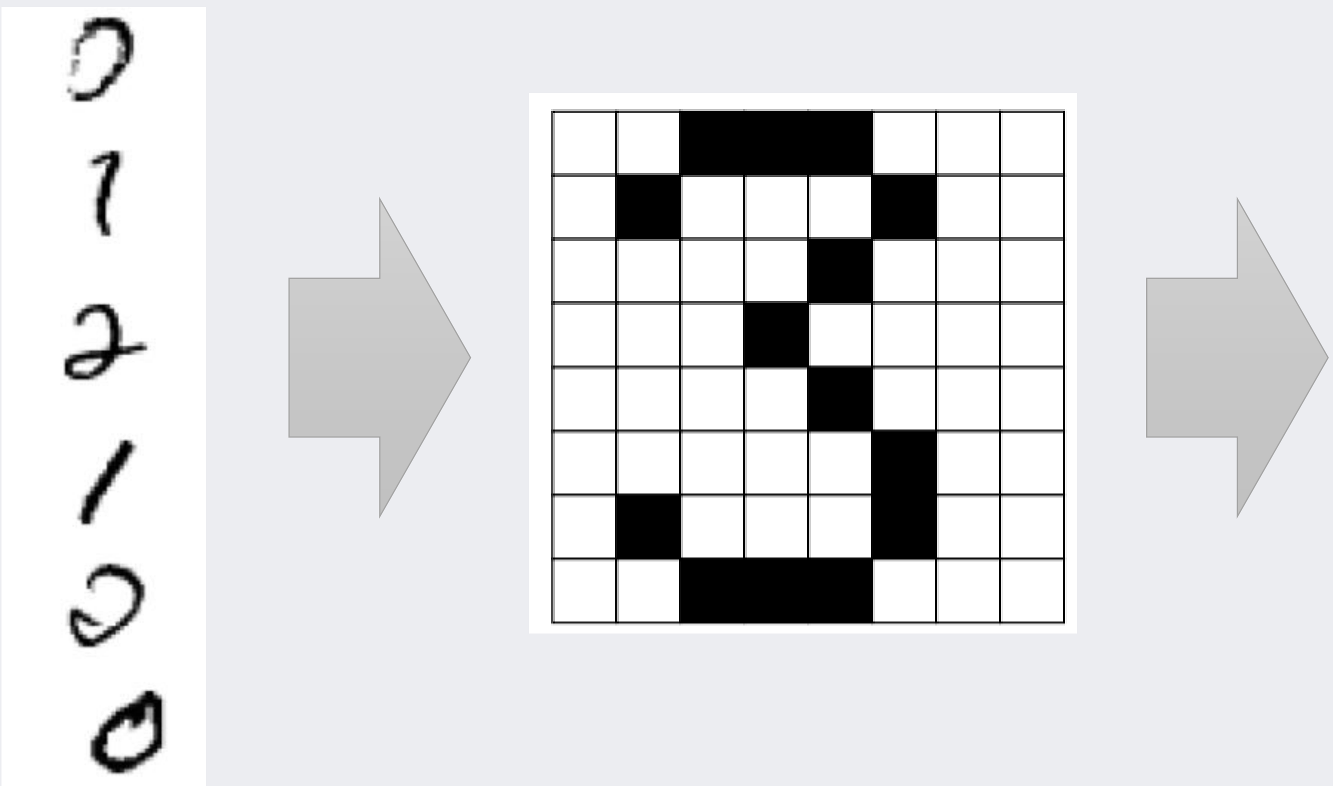
Make a decision

- P(Height=178, BFS=11 | Male)P(Male) > P(Height=178, BFS=11 | Female)P(Female)
  - ✓ Classify him as male
- What if there are 400 males and 100 females in the training data?
  - ✓ Consider the prior probability P(Male) & P(Female)
  - ✓ P(Height=178, BFS=11 | Male)*P(Male) = 0.05*0.8 = 0.04
  - ✓ P(Height=178, BFS=11 | Female)*P(Female) = 0.0005*0.2 = 0.0001
  - ✓ Classify him as male

5

Naive Bayesian Classification: Example

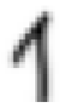
- Hand digit recognition
  - ✓ Input: pixel grids
  - ✓ Classes: a digit 0-9



Which digit?

Naive Bayesian Classification: Example

- Feature definition
  - ✓ One feature f_{ij} for each grid position <i, j>
  - ✓ Possible feature values are on/off, based on whether intensity is more or less than 0.5 in underlying image.
  - ✓ Each input maps to a feature vector, e.g.



$$\rightarrow \langle F_{0,0} = 0 \;\; F_{0,1} = 0 \;\; F_{0,2} = 1 \;\; F_{0,3} = 1 \;\; F_{0,4} = 0 \;\; \ldots F_{15,15} = 0 \rangle$$

- Naïve Bayesian Model

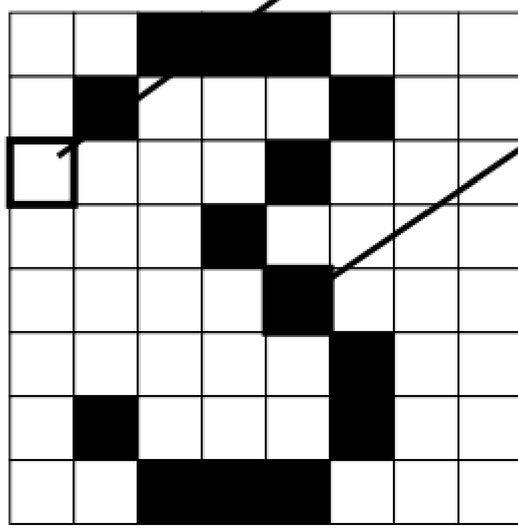
$$P(Y|F_{0,0} \ldots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$$

Naive Bayesian Classification: Example

- What has to be learned?

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

Naive Bayesian Classification: Example

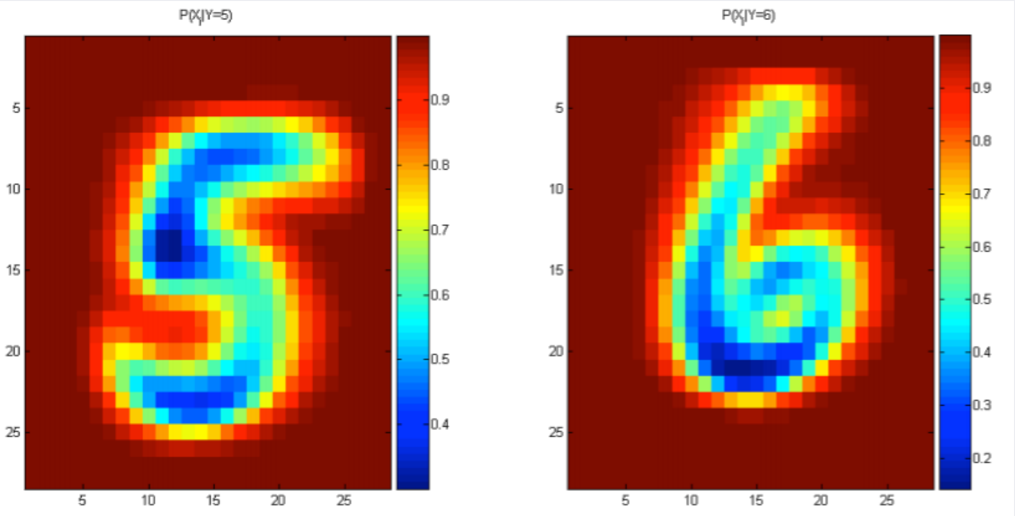
- Training
  - ✓ Count the target class ratio for each grid
    - Prior:

$$P(Y=y) = \frac{Count(Y=y)}{\sum_{y'} Count(Y=y')}$$

    - Observation distribution:

$$P(X_i = x|Y=y) = \frac{Count(X_i = x, Y=y)}{\sum_{x'} Count(X_i = x', Y=y)}$$

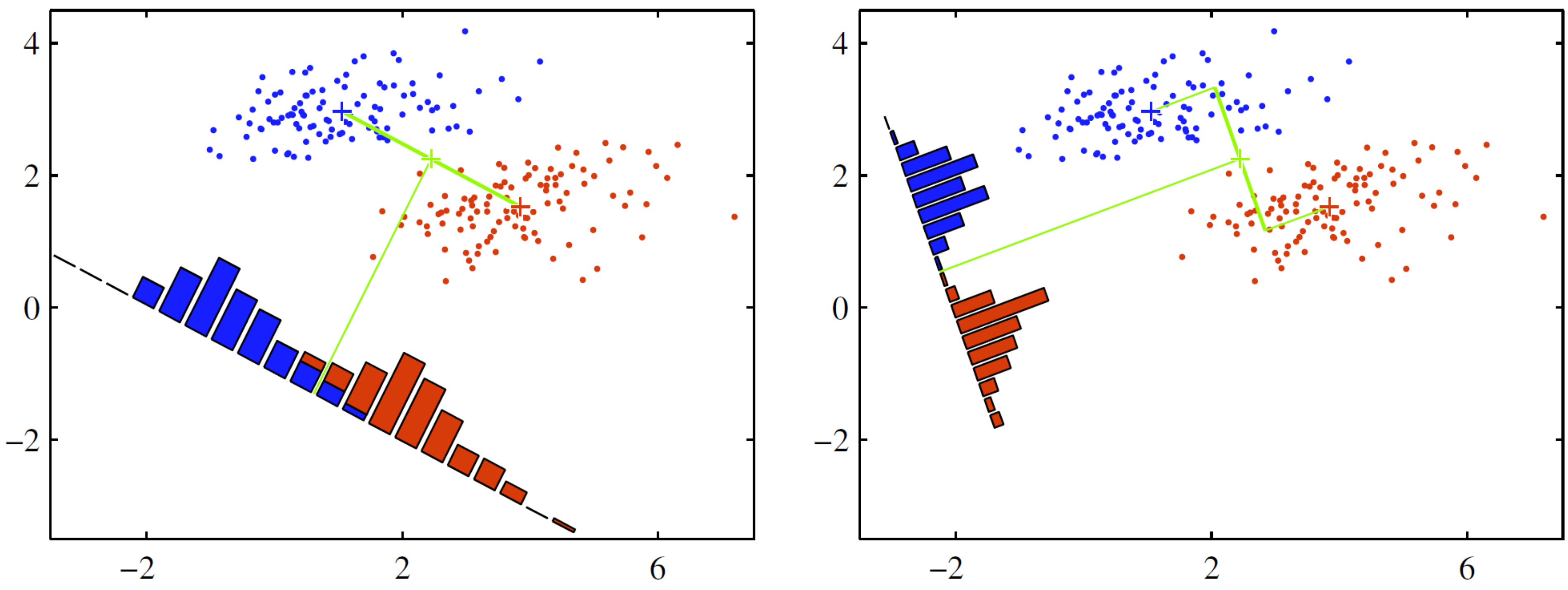
- Trained examples



AGENDA

Linear Discriminant Analysis

- Fisher's LDA
  - ✓ Which line is better to discriminate two classes after projection?

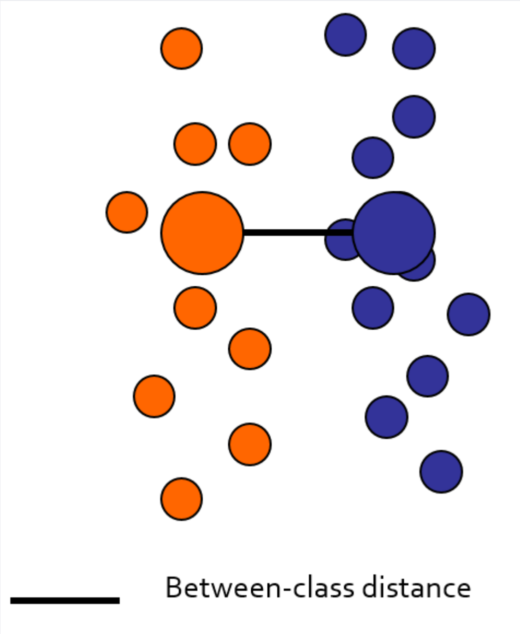


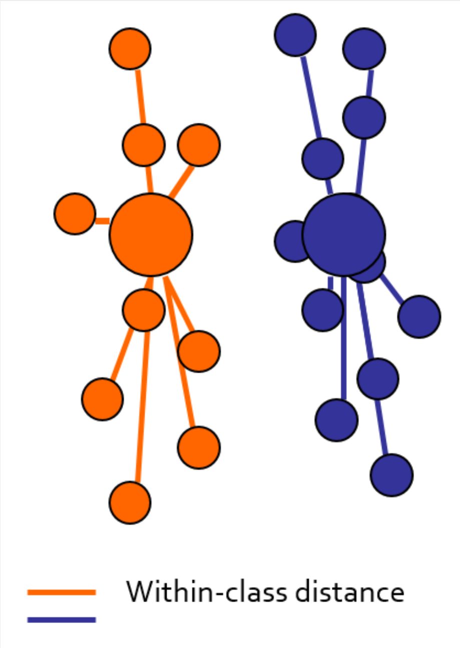
✓ Find the most distinguishable vector!

(Source: Bishop (2006))

Linear Discriminant Analysis

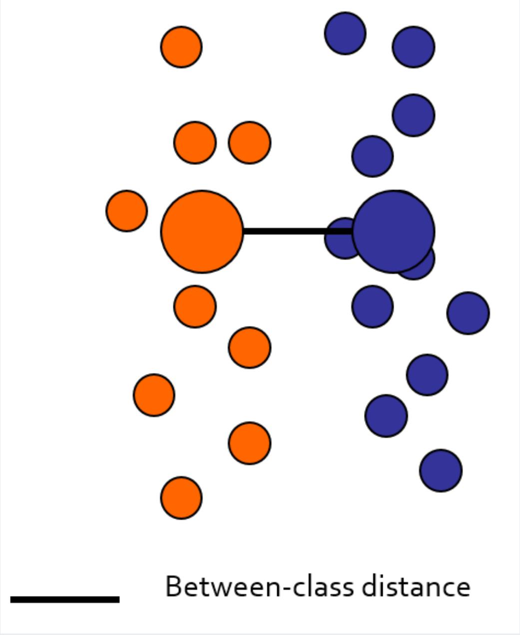
- Two type of class distances
  - ✓ Between-class distance
    - Distance between the centroids of different classes
  - ✓ Within-class distance
    - Accumulated distance of an instance to the centroid of its class

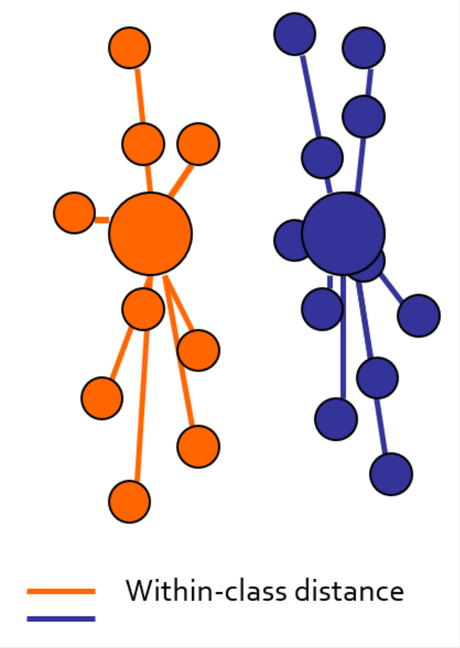


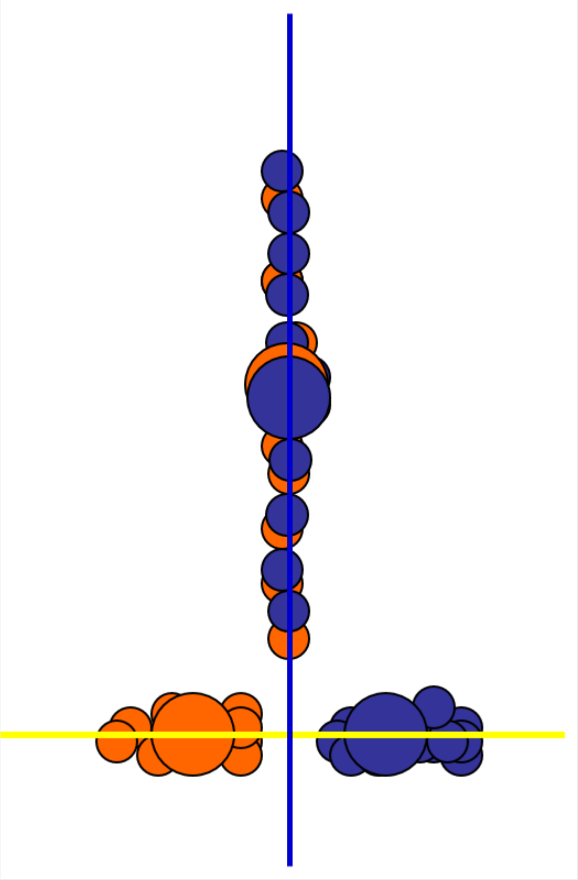


Linear Discriminant Analysis

- (Fisher's) Linear Discriminant Analysis
  - ✓ Find most discriminant projection by maximizing between-class distance (variance) and minimizing within-class distance (variance)







Linear Discriminant Analysis

- Fisher's LDA (cont')
  - ✓ Take the D-dimensional input vector $\mathbf{x}$ and project it down to one dim.

$$y = \mathbf{w}^T \mathbf{x}$$

  - ✓ Consider a two-class problem in which there are N_1 & N_2 observations in C_1 and C_2, respectively.

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in C_1} \mathbf{x}_n \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in C_2} \mathbf{x}_n$$

Linear Discriminant Analysis

- Fisher's LDA (cont')
  - ✓ Objective 1: Choose w to maximize the separation of the projected class means (between class variance)

$$m_2 - m_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1) \qquad m_k = \mathbf{w}^T\mathbf{m}_k$$

  - ✓ Objective 2: Choose w to minimize the variance in each class after projection (within class variance)

$$s_k^2 = \sum_{n \in C_k} (y_k - m_k)^2$$

Linear Discriminant Analysis

- Fisher's LDA (cont')
  - ✓ Fisher's criterion
    - The ratio of the between-class variance to the within-class variance

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

Linear Discriminant Analysis

- Fisher's LDA (cont')

  ✓ Find w

  - Differentiating the Fisher's criterion w.r.t. $\mathbf{w}$, then $J(\mathbf{w})$ is maximized when

$$(\mathbf{w}^T \mathbf{S}_B \mathbf{w})\mathbf{S}_W \mathbf{w} = (\mathbf{w}^T \mathbf{S}_W \mathbf{w})\mathbf{S}_B \mathbf{w}$$

  - $\mathrm{S_B}\mathbf{w}$ is always in the direction of $(\mathbf{m}_2\text{-}\mathbf{m}_1)$
  - Can drop the scalar factor $(\mathbf{w}^T\mathrm{S_B}\mathbf{w})$ and $(\mathbf{w}^T\mathrm{S_W}\mathbf{w})$
  - Then, obtain ***Fisher's linear discriminant***

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

AGENDA

R Exercise

- Target Data: Wisconsin Breast Cancer
  - ✓ Predicting whether a patient is malignant or benign
  - ✓ The real-valued features for the following information are computed for each cell nucleus:
    - a) radius (mean of distances from center to points on the perimeter)
    - b) texture (standard deviation of gray-scale values)
    - c) perimeter
    - d) area
    - e) smoothness (local variation in radius lengths)
    - f) compactness (perimeter^2 / area - 1.0)
    - g) concavity (severity of concave portions of the contour)
    - h) concave points (number of concave portions of the contour)
    - i) symmetry
    - j) fractal dimension ("coastline approximation" - 1)
  - ✓ Mean, standard error, and worst values are used as input variables

R Exercise

- Write a performance evaluation function and initialize the result summary table

```
# Performance Evaluation Function -------------------------------------
perf_eval <- function(cm){

  # True positive rate: TPR (Recall)
  TPR <- cm[2,2]/sum(cm[2,])
  # Precision
  PRE <- cm[2,2]/sum(cm[,2])
  # True negative rate: TNR
  TNR <- cm[1,1]/sum(cm[1,])
  # Simple Accuracy
  ACC <- (cm[1,1]+cm[2,2])/sum(cm)
  # Balanced Correction Rate
  BCR <- sqrt(TPR*TNR)
  # F1-Measure
  F1 <- 2*TPR*PRE/(TPR+PRE)

  return(c(TPR, PRE, TNR, ACC, BCR, F1))
}

# Result summary
Perf.Table <- matrix(0, nrow = 2, ncol = 6)
rownames(Perf.Table) <- c("Naive Bayes", "LDA")
colnames(Perf.Table) <- c("TPR", "Precision", "TNR", "Accuracy", "BCR", "F1-Measure")
```

R Exercise

- Load the data and divide the dataset into training (70%) and test (30%)

```
# Load the wdbc data
Wdbc.Data <- read.csv("wdbc.csv", header = FALSE)

# Divide the dataset into the training (70%) and Test (30%) datasets
trn.idx <- sample(1:nrow(Wdbc.Data), round(0.7*nrow(Wdbc.Data)))
```

R Exercise

- Train the Naive Bayesian Classifier

```
# Classifier 1: Naive Bayesian Classifier ---------------------------------
# e1071 package install
install.packages("e1071")
library(e1071)

# Use the dataset without normalization
NB.Trn.Data <- Wdbc.Data[trn.idx,]
colnames(NB.Trn.Data)[31] <- "Target"

NB.Tst.Data <- Wdbc.Data[-trn.idx,]
colnames(NB.Tst.Data)[31] <- "Target"

# Training the Naive Bayesian Classifier
NB.model <- naiveBayes(Target ~ ., data = NB.Trn.Data)
NB.model$apriori
NB.model$tables
```

R Exercise

- Check the trained parameters

```
> NB.model$apriori
Y
  B   M
150 248
```

```
> NB.model$tables
$V1
   V1
Y        [,1]     [,2]
  B 17.44373 3.259898
  M 12.12940 1.780143

$V2
   V2
Y        [,1]     [,2]
  B 21.67427 3.723244
  M 17.78270 3.960919

$V3
   V3
Y         [,1]     [,2]
  B 115.16073 22.17670
  M  77.91379 11.74798

$V4
   V4
Y        [,1]     [,2]
  B 976.8833 374.8957
  M 461.6625 133.4494

$V5
   V5
Y          [,1]       [,2]
  B 0.10332747 0.01313281
  M 0.09271343 0.01372253

$V6
   V6
Y          [,1]       [,2]
  B 0.14425960 0.05483254
  M 0.07890472 0.03337068
```

R Exercise

- Evaluate the classification performance

```r
# Predict the new input data based on Naive Bayesian Classifier
NB.posterior = predict(NB.model, NB.Tst.Data, type = "raw")
NB.prey = predict(NB.model, NB.Tst.Data, type ="class")

NB.cfm <- table(NB.Tst.Data[,31], NB.prey)
NB.cfm

Perf.Table[1,] <- perf_eval(NB.cfm)
Perf.Table
```

```
> NB.cfm
   NB.prey
      B   M
  B  52  10
  M   8 101
> Perf.Table[1,] <- perf_eval(NB.cfm)
> Perf.Table
                  TPR Precision       TNR  Accuracy       BCR F1-Measure
Naive Bayes 0.9266055 0.9099099 0.8387097 0.8947368 0.8815628  0.9181818
LDA         0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  0.0000000
```

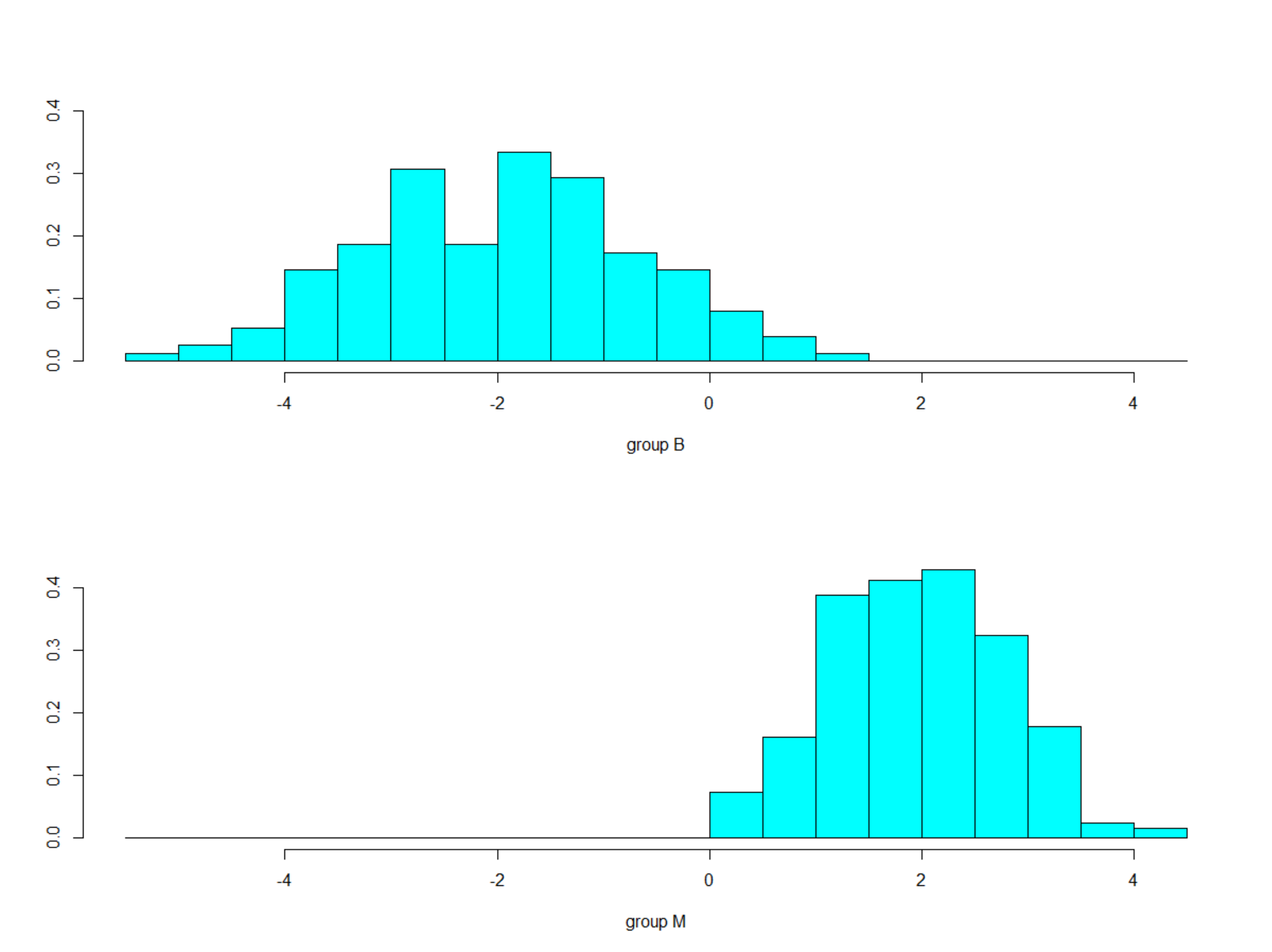
R Exercise

- Train LDA

```r
# Classifier 2: Linear Discriminant Analysis ------------------------------
install.packages("MASS")
library(MASS)

# Use the dataset without normalization
LDA.Trn.Data <- Wdbc.Data[trn.idx,]
colnames(LDA.Trn.Data)[31] <- "Target"

LDA.Tst.Data <- Wdbc.Data[-trn.idx,]
colnames(LDA.Tst.Data)[31] <- "Target"

# Training LDA
LDA.model <- lda(Target ~ ., data = LDA.Trn.Data)

# Training result of LDA
plot(LDA.model)
```

R Exercise

- Best projection and the histogram of the two classes



R Exercise

- Evaluate the performance

```r
# Predict the unknown observations based on the LDA
LDA.Predict <- predict(LDA.model, LDA.Tst.Data)

names(LDA.Predict)
LDA.Predict$class
LDA.Predict$posterior
LDA.Predict$x

LDA.cfm <- table(LDA.Tst.Data$Target, LDA.Predict$class)
LDA.cfm

Perf.Table[2,] <- perf_eval(LDA.cfm)
Perf.Table
```

```
> LDA.cfm

     B   M
  B  55   7
  M   1 108
> Perf.Table[2,] <- perf_eval(LDA.cfm)
> Perf.Table
                  TPR Precision       TNR  Accuracy       BCR F1-Measure
Naive Bayes 0.9266055 0.9099099 0.8387097 0.8947368 0.8815628  0.9181818
LDA         0.9908257 0.9391304 0.8870968 0.9532164 0.9375277  0.9642857
```