Dimensionality Reduction

Linear Discriminant Analysis

Dimensionality reduction algorithms

Linear

- PCA
- MDS
- Factor analysis
- LDA (supervised)

Nonlinear

- t-SNE
- SNE
- QDA (supervised)
- Sammon mapping
- Isomap
- Local Linear Embedding (LLE)
- CCA
- MVU
- Laplacian Eigenmaps

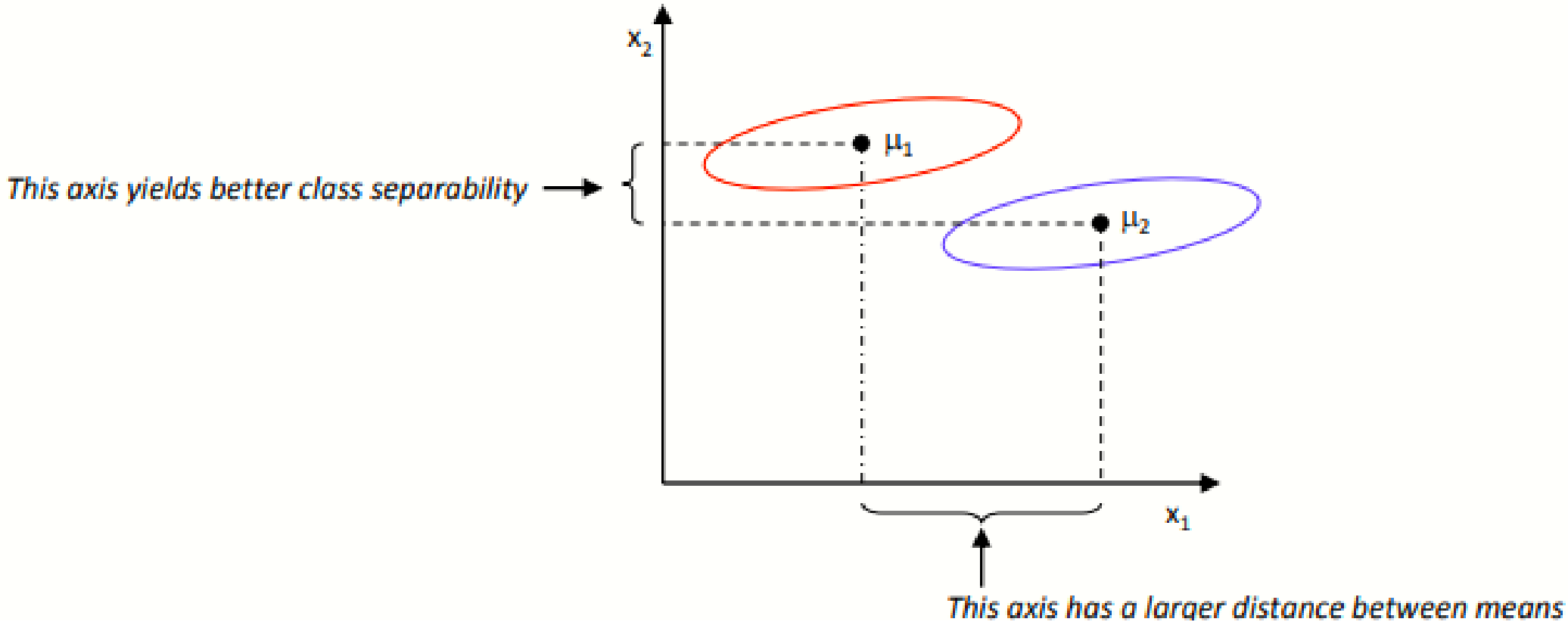
Linear Discriminant Analysis

- Want to reduce dimensionality while preserving ability to discriminate
- Suppose that we have N p-dimensional data points x_i, for $i = 1, \ldots, N$, which belong to C known classes $\omega_1, \ldots, \omega_c$.

How do we utilize the label information to find informative directions (i.e., projection vectors)?

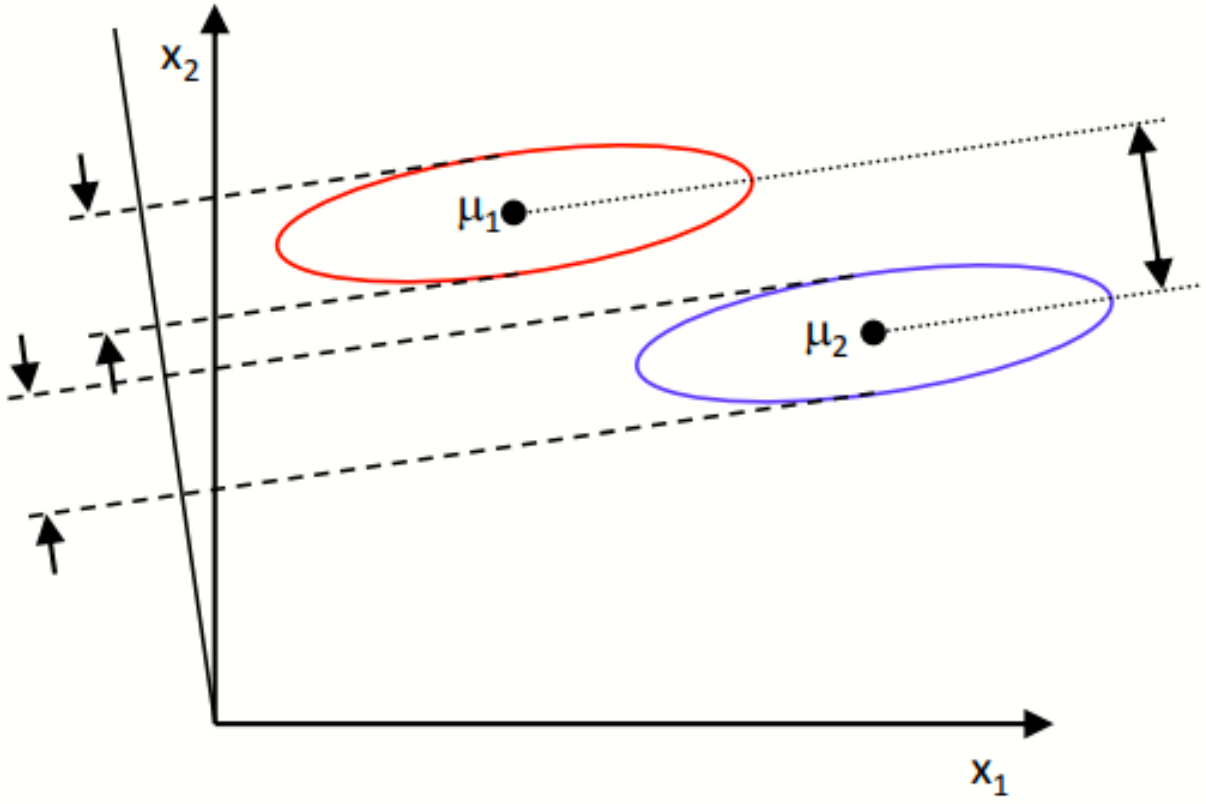
One idea…

We could then choose the distance between the projected means as our objective function…



Fisher's solution

- Find projections maximizing **between-class scatter** while minimizing **within-class scatter**



We want a projections where examples from the same class are projected very close to each other and, at the same time, the projected means are as far apart as possible

Fisher's solution

- Scatter of class ω_i

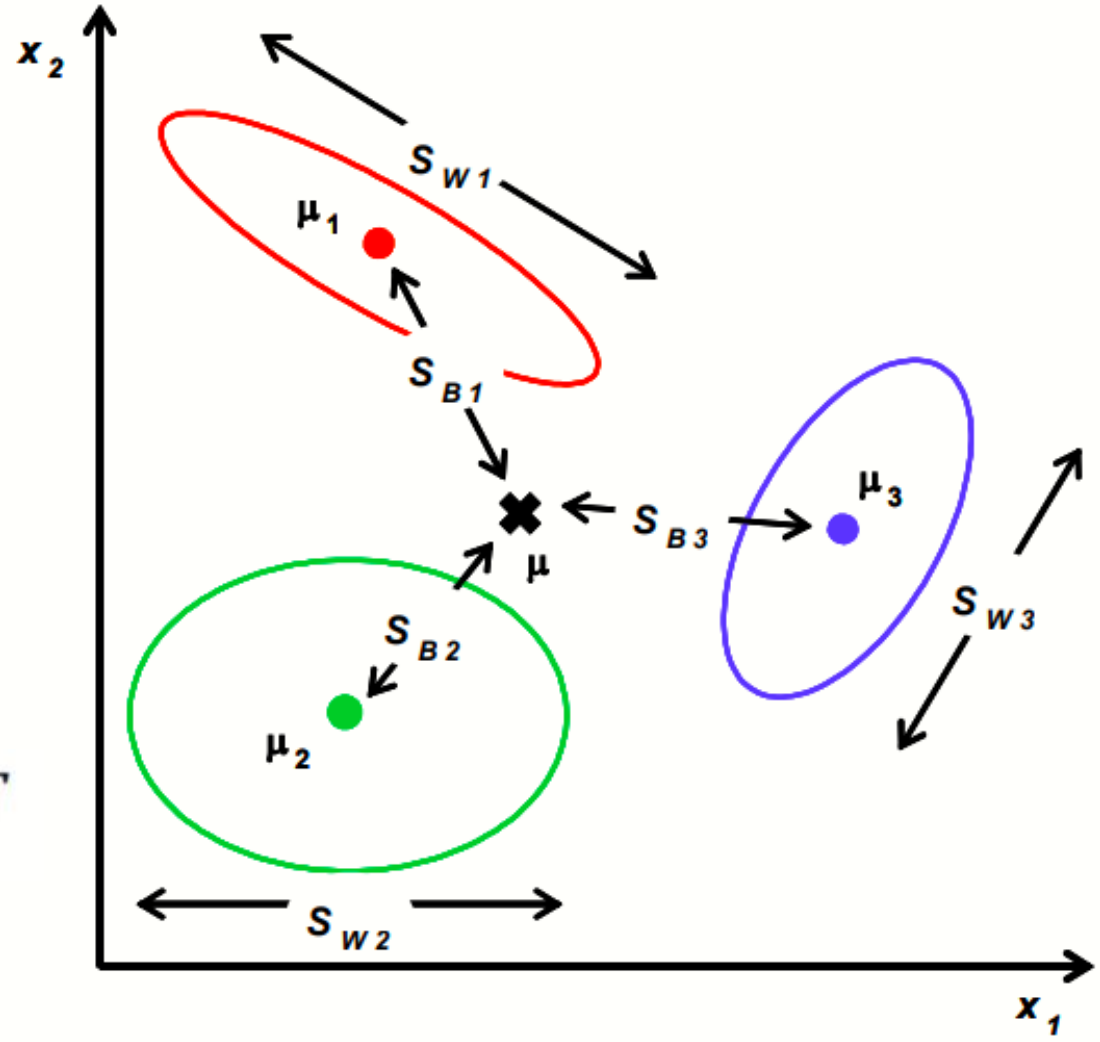
$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

- Within-class scatter

$$S_W = \sum_{i=1}^{C} S_i$$

- Between-class scatter

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$



Fisher's Linear Discriminant Analysis

- Linear discriminant analysis constructs one or more **discriminant equations** D_i (linear combinations of the p predictor variables x_k) such that the different groups differ as much as possible on D.
- Discriminant function:

$$D_i = b_0 + \sum_{k=1}^{p} b_k X_k$$

- Examine the absolute value of the coefficients b_k to determine which predictors play an important role; the larger the value, the more important the predictor.

LDA in R

```
library(tidyverse)
library(caret)
library(MASS)

#use the trusty old “iris” data for an example
data(iris)

preproc.param <- iris %>% preProcess(method = c("center", "scale"))

# Transform the data using the estimated parameters
transformed <- preproc.param %>% predict(iris)

# Fit the model
lda.model <- lda(Species~., data = transformed)
```

```
Call:
lda(Species ~ ., data = transformed)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa       -1.0111914   0.8504137   -1.3006301  -1.2507035
versicolor    0.1119073  -0.6592236    0.2843712   0.1661774
virginica     0.8992841  -0.1911901    1.0162589   1.0845261

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.6867795  0.01995817
Sepal.Width   0.6688251  0.94344183
Petal.Length -3.8857950 -1.64511887
Petal.Width  -2.1422387  2.16413593

Proportion of trace:
   LD1    LD2
0.9912 0.0088
```

The first trace number indicates the percentage of between-group scatter that the first discriminant function is able to explain from the total amount of between-group scatter.

High trace number → discriminant function plays an important role!

Make predictions

predictions <- model %>% predict(transformed)

```
> tail(predictions$x)
            LD1        LD2
145 -6.847359  2.4289507
146 -5.645003  1.6777173
147 -5.179565 -0.3634750
148 -4.967741  0.8211405
149 -5.886145  2.3450905
150 -4.683154  0.3320338
```

```
> tail(predictions$posterior)
              setosa   versicolor virginica
145 4.048249e-46 2.524984e-07 0.9999997
146 4.970070e-39 7.473361e-05 0.9999253
147 4.616611e-36 5.898784e-03 0.9941012
148 5.548962e-35 3.145874e-03 0.9968541
149 1.613687e-40 1.257468e-05 0.9999874
150 2.858012e-33 1.754229e-02 0.9824577
```

```
> tail(predictions$class)
[1] virginica virginica virginica virginica virginica virginica
```

Model accuracy

```
> table(Original=iris$Species,Predicted=predictions$class)
            Predicted
Original     setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49
```

```
> mean(predictions$class==transformed$Species)
[1] 0.98
```

```
lda.data <- cbind(transformed, predict(lda.mdl)$x)

ggplot(lda.data, aes(LD1, LD2)) +   geom_point(aes(color = Species))
```