Master in Innovation and Research in Informatics (MIRI)

Data Science track

Multivariate Analysis (MVA)
Linear Discriminant Analysis and extensions

Prof: Arturo Palomino

Dante Conti

Credits:
Prof. Daniel Fernández
daniel.fernandez.martinez@upc.edu

Outline

1. Setting the scene
2. How does Linear Discriminant Analysis (LDA) work?
3. PCA vs. LDA
4. LDA in R
5. Extensions to LDA
6. Examples

First steps. Discriminant Analysis

- We will be focusing today on the **classification** aspect.

- The response variable is **qualitative** (categorical)

- Predicting a **qualitative response** for an observation can be referred to as **classifying that observation**, since it involves assigning the observations to a category, or class.

- **Linear Discriminant Analysis (LDA)** is a **dimensionality reduction** technique used as a pre-processing step in Machine Learning and pattern **classification** applications.

- Analysis for **pre-determined groups** ⇨ Supervising learning

Machine Learning. Types of learning

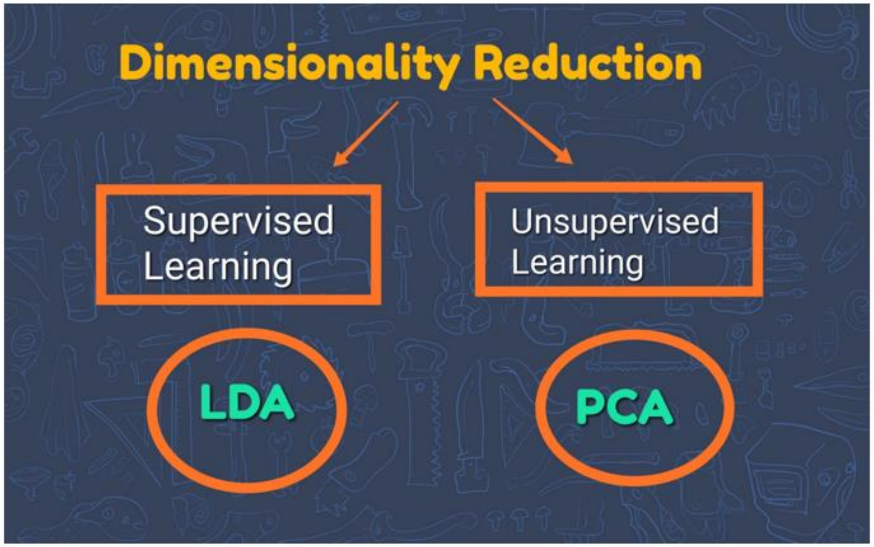
1. **Supervised:** From a learning set (a.k.a. training set) with the correct labels of the observations, the algorithm "learns" to generate the correct label for all possible observations. Learning from examples.
   - Regression
   - Classification
   - Dimensionality Reduction

2. **Unsupervised:** From observations without labels, the algorithm detects similarities between observations in such a way that similar observations are grouped / classified.
   - Clustering
   - Dimensionality Reduction

3. **Others:** semi-supervised, etc

Dimensionality reduction

The main goal of dimensionality reduction techniques is to reduce the dimensions by removing the redundant and dependent features by transforming the features from higher dimensional space to a space with lower dimensions



LDA: Different question compared to to PCA & FA (e.g. maximize variation explained)

LDA: Essentially **don't care** how much variance is explained by groups

Think LDA as:

"How far can I separate known groups given measurements of several variables on individuals within these groups"

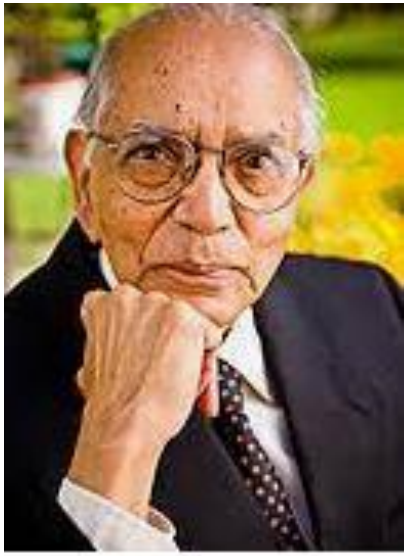
"What distinguishes my groups?"

LDA is a **supervised classification technique** which takes labels into consideration. This category of dimensionality reduction is used in biometrics, bioinformatics and chemistry.

LDA developers

LDA was developed as early as 1936 by Ronald A. Fisher.
The original Linear discriminant applied to only a 2-class problem.
It was only in 1948 that C.R. Rao generalized it to apply to multi-class problems.

Sir Ronald Fisher (1890-1962)



Prof. C.R. Rao (1920-)

LDA (in layman's terms)

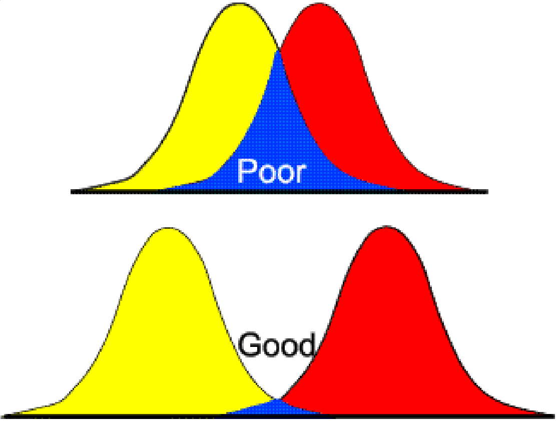
Typically, LDA is used when we already have predefined classes/categories of response and we want to build a model that helps in distinctly predicting the class, if any new observation comes into equation

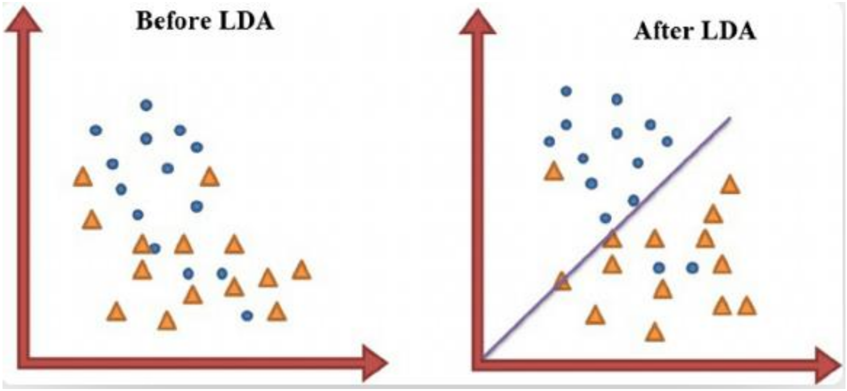
Source: https://www.flickr.com/photos/15609463@N03/14898932531

Problem:separate two or more groups of individuals, given measurements for these individuals on several variables (i.e. quantitative variables)

LDA (in layman's terms)

Goal: differentiate or discriminate the response variable into its distinct classes





How? by constructing discriminant functions that are <u>linear combinations of the variables</u>.

Objectives?

- **Description**: to be able to describe observed cases mathematically in a manner that separates them into groups as well as possible.
- **Prediction**: to be able to classify new observations as belonging to one or another of the groups.

LDA (in layman's terms). Examples

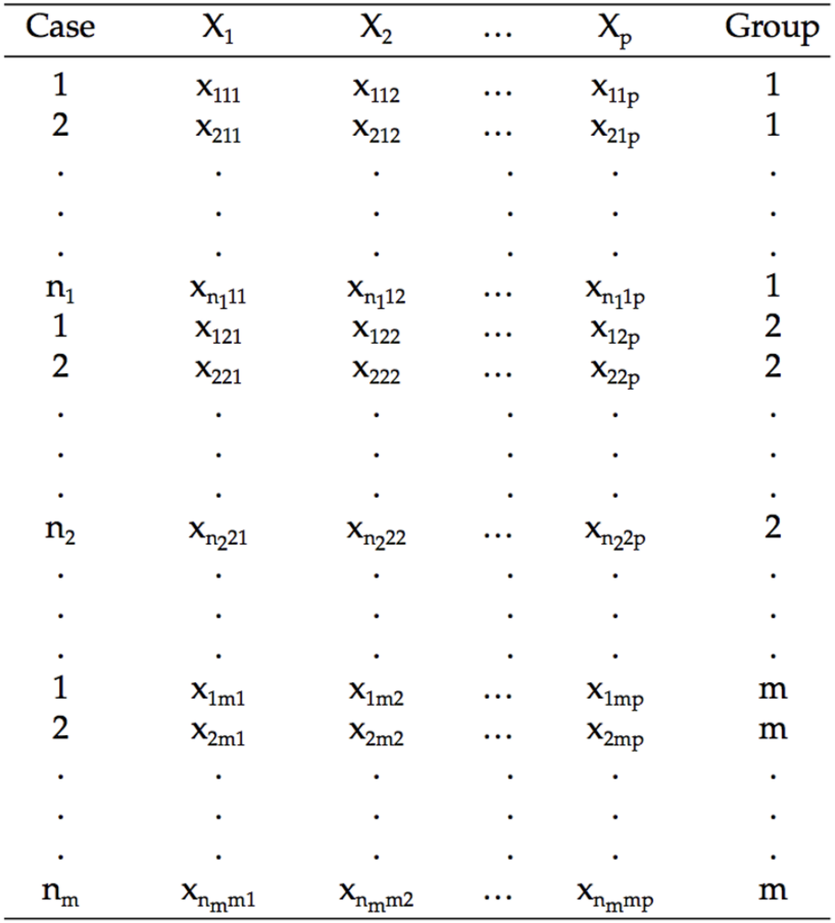
Real-life **examples**:

1. When we want to predict whether an applicant for a bank loan is likely to default or not.

2. Predict likelihood of a heart attack based on various health indicators.

3. Predict stability level — "Good", "Requires Inspection" or "Requires Repair/Replacement"- of an engine/machine based on various performance indicators.

Data set



Case	X_1	X_2	…	X_p	Group
1	x_{111}	x_{112}	…	x_{11p}	1
2	x_{211}	x_{212}	…	x_{21p}	1
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n_1	$x_{n_1 11}$	$x_{n_1 12}$	…	$x_{n_1 1p}$	1
1	x_{121}	x_{122}	…	x_{12p}	2
2	x_{221}	x_{222}	…	x_{22p}	2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n_2	$x_{n_2 21}$	$x_{n_2 22}$	…	$x_{n_2 2p}$	2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
1	x_{1m1}	x_{1m2}	…	x_{1mp}	m
2	x_{2m1}	x_{2m2}	…	x_{2mp}	m
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n_m	$x_{n_m m1}$	$x_{n_m m2}$	…	$x_{n_m mp}$	m

Dimensions of matrix is n x **(p+1)**

n cases: $n = n_1+n_2+\ldots+ n_m$

p variables: $X_1,\ldots, X_p$

1 group indicator (variable p+1)

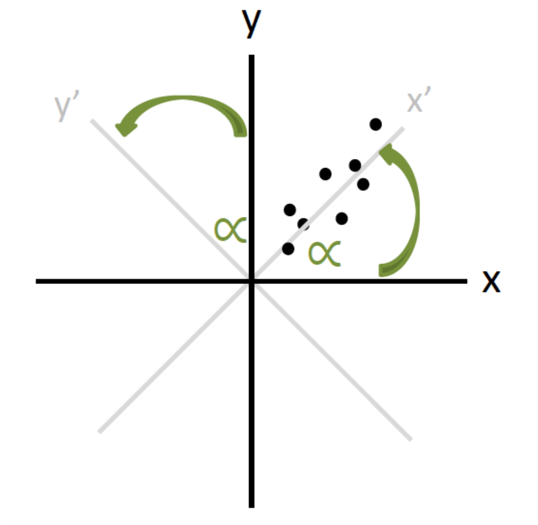
m groups

We know the group membership for each case (row)) – Supervised learning

The data for a DA **do not need to be standardized** to have zero means and unit variance (as was common in PCA).

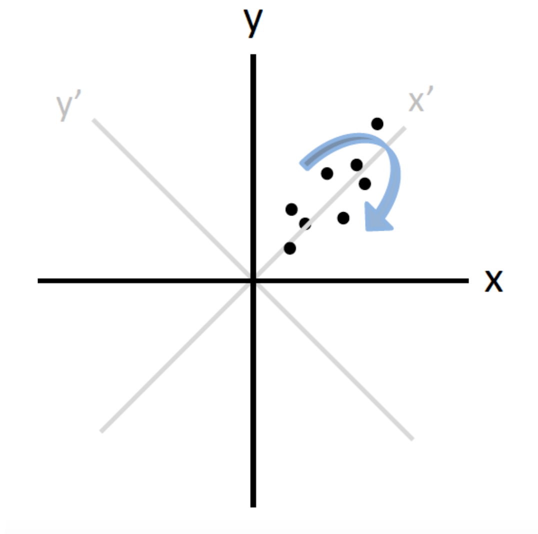
This is because the outcome of a DA is not affected by the scaling of $X_1,\ldots, X_p$

How does LDA work?



1. Find the axis that gives the greatest separation between 2 groups

2. Fix the axis



3. Rotate around the fixed axis to maximize difference between first 2 groups and the 3rd group

4. Repeat steps 2 & 3 for all groups included

How does LDA work?

Let’s explain LDA with **an example:**
(based on Joshua Starmer’s work)

- Let’s imagine we have a new drug for a particular disease and we run a clinical trial test. As a results **we observed** that
  - ➢ The new drug works well for some people
  - ➢ But it does not work well for other people (actually they feel worse)

- The question to solve is:

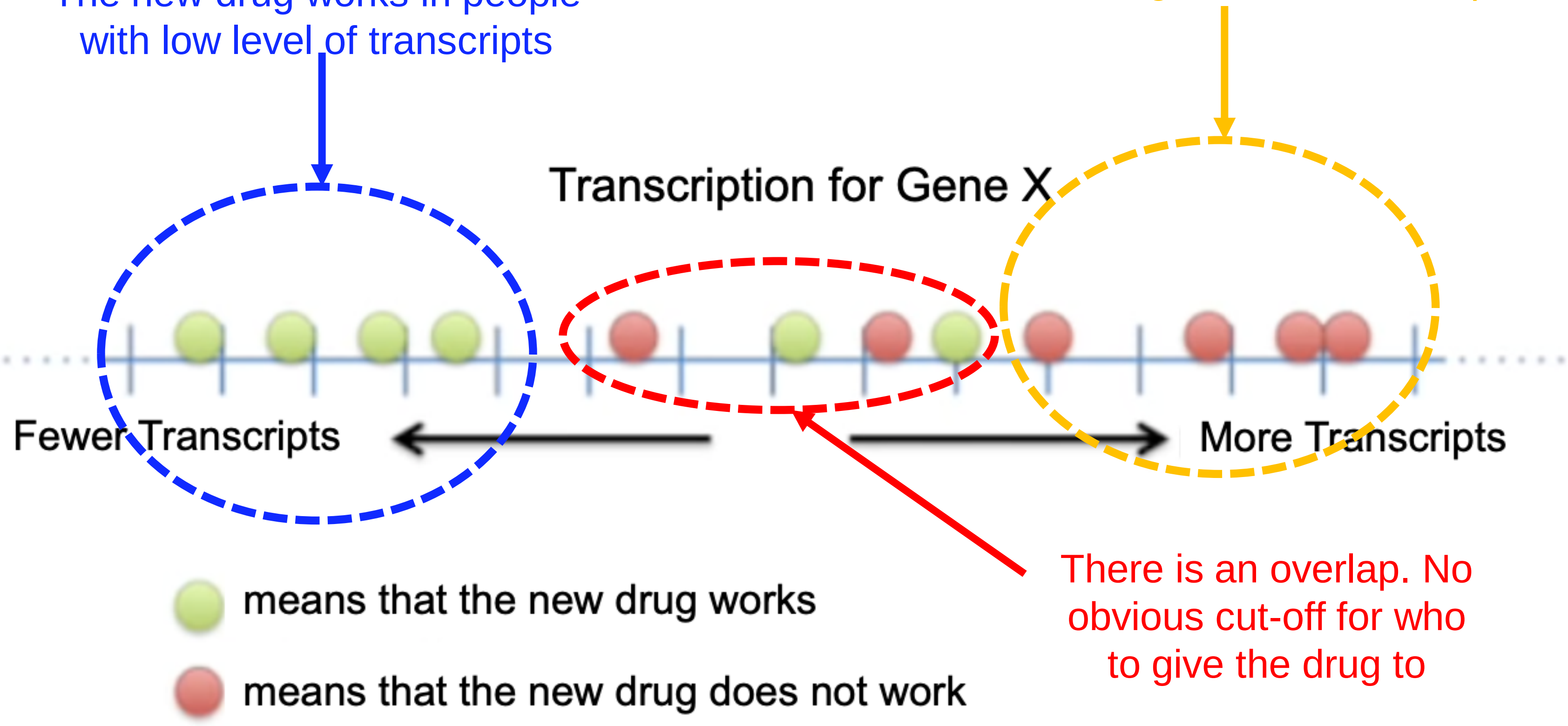
How do we decide who to give the drug to?

How does LDA work?

Let’s use (for example) a variable/feature representing 1 gene expression (Gene X):

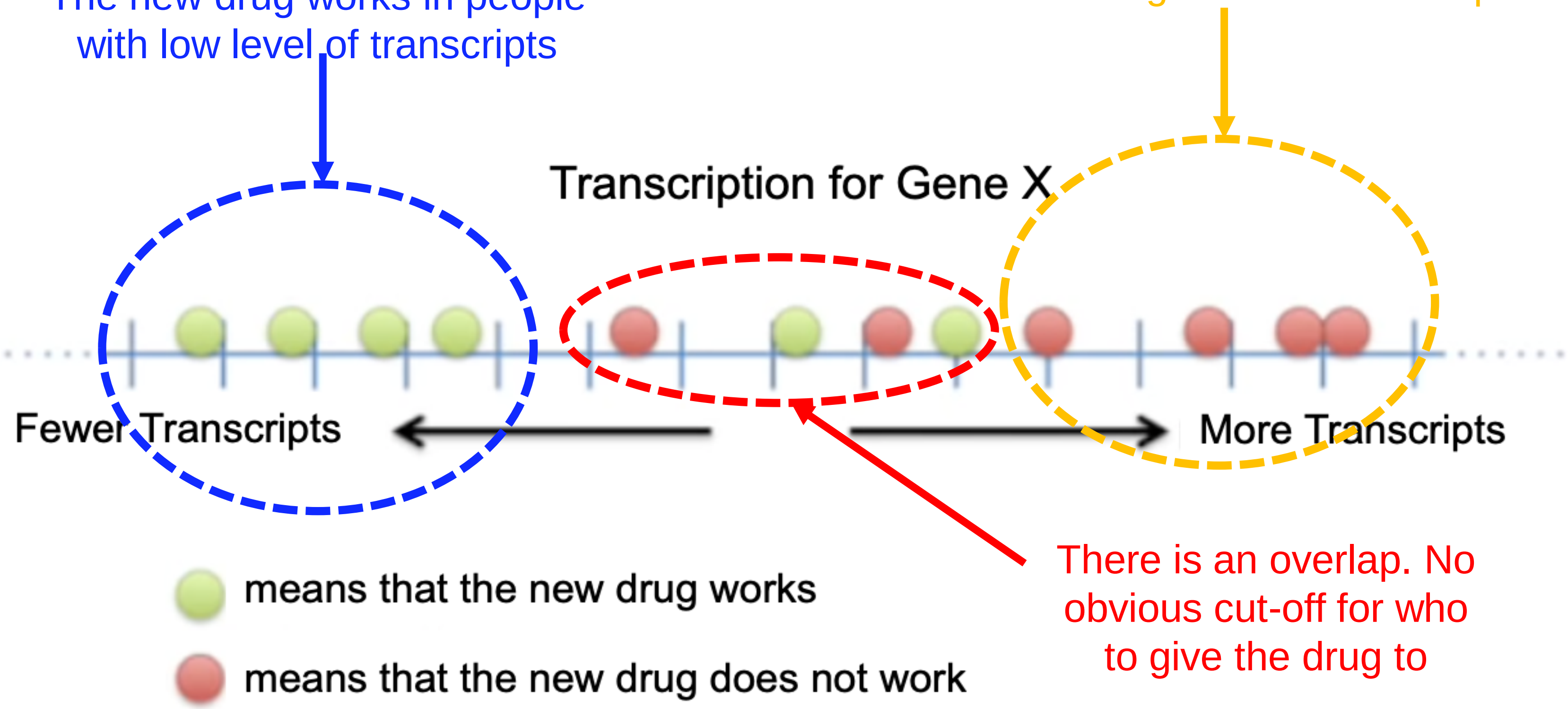
The new drug works in people with low level of transcripts

The new drug does not work in people with a high level of transcripts


Transcription for Gene X
Fewer Transcripts
More Transcripts
means that the new drug works
means that the new drug does not work

There is an overlap. No obvious cut-off for who to give the drug to

How does LDA work?



The new drug works in people with low level of transcripts

The new drug does not work in people with a high level of transcripts

There is an overlap. No obvious cut-off for who to give the drug to

🟢 means that the new drug works

🔴 means that the new drug does not work

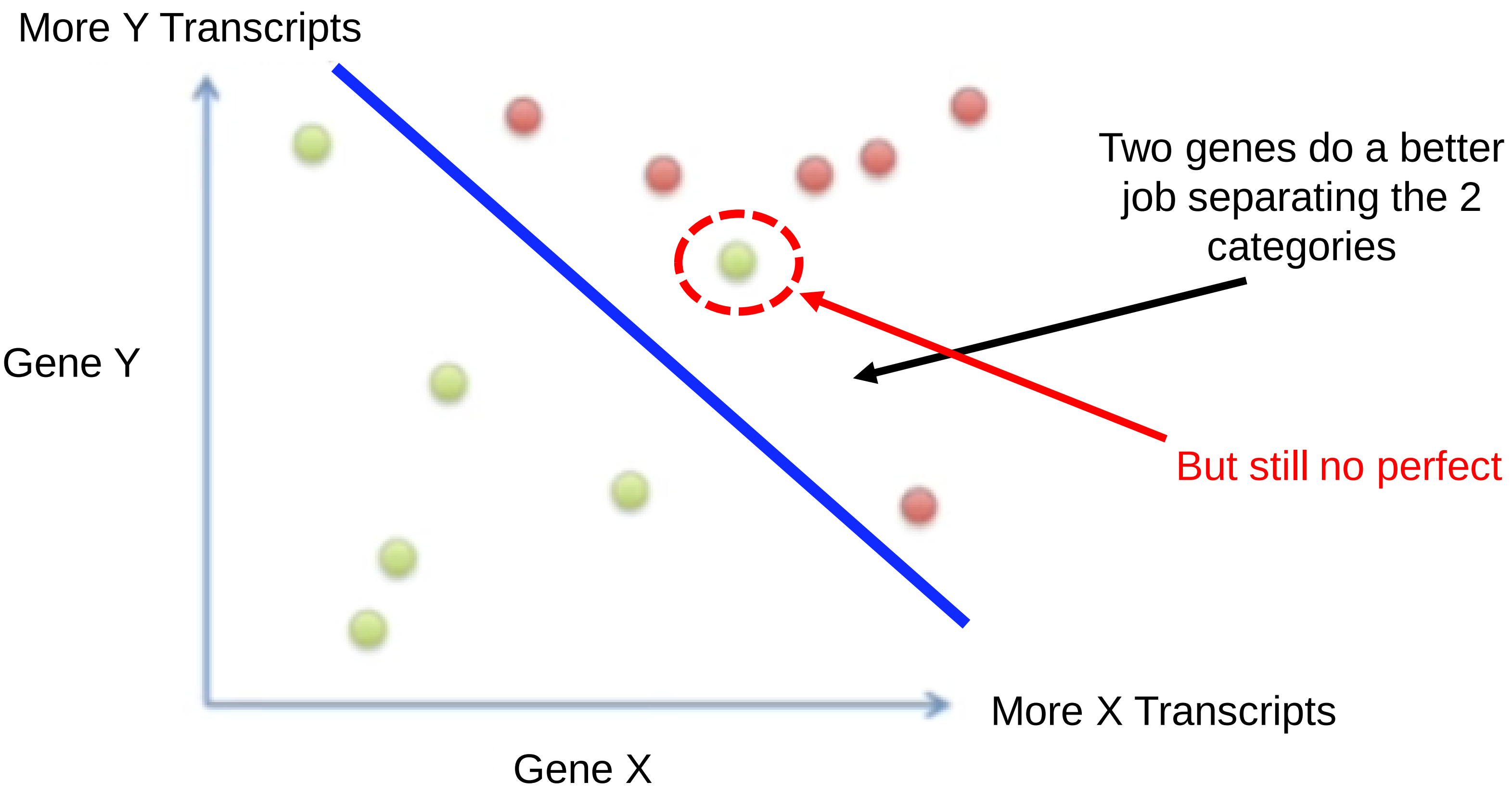
In short, Gene X works relatively well at separating who should take the drug from who shouldn’t.

Can we do better?

What if we use more than one gene to make a decision?

How does LDA work?

Using two genes now: Gene X and Gene Y



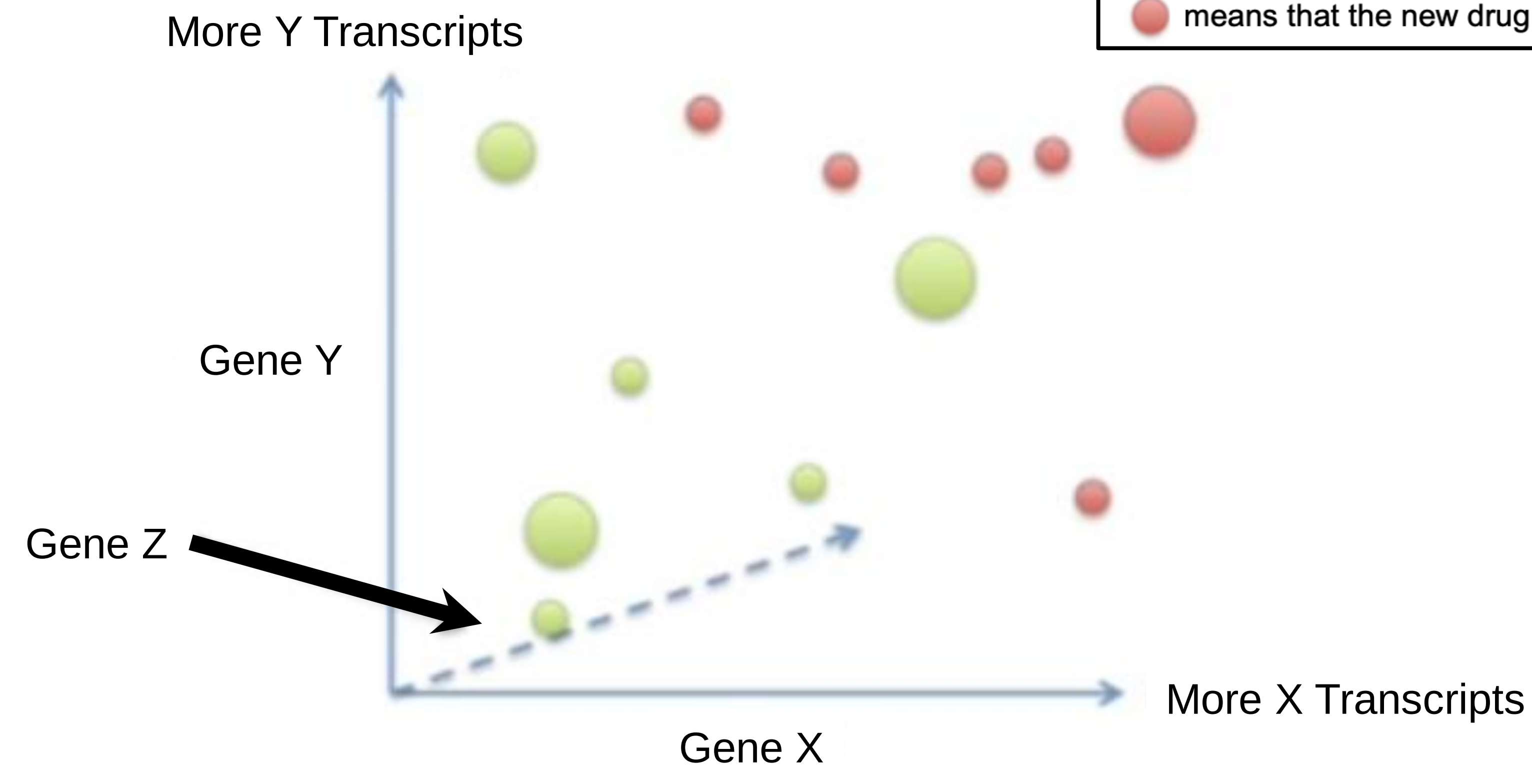
We can draw a line that separates both groups

How does LDA work?

Can we improve with 3 genes: Gene X, Gene Y, and Gene Z

🟢 means that the new drug works

🔴 means that the new drug does not work



More Y Transcripts

Gene Y

Gene Z

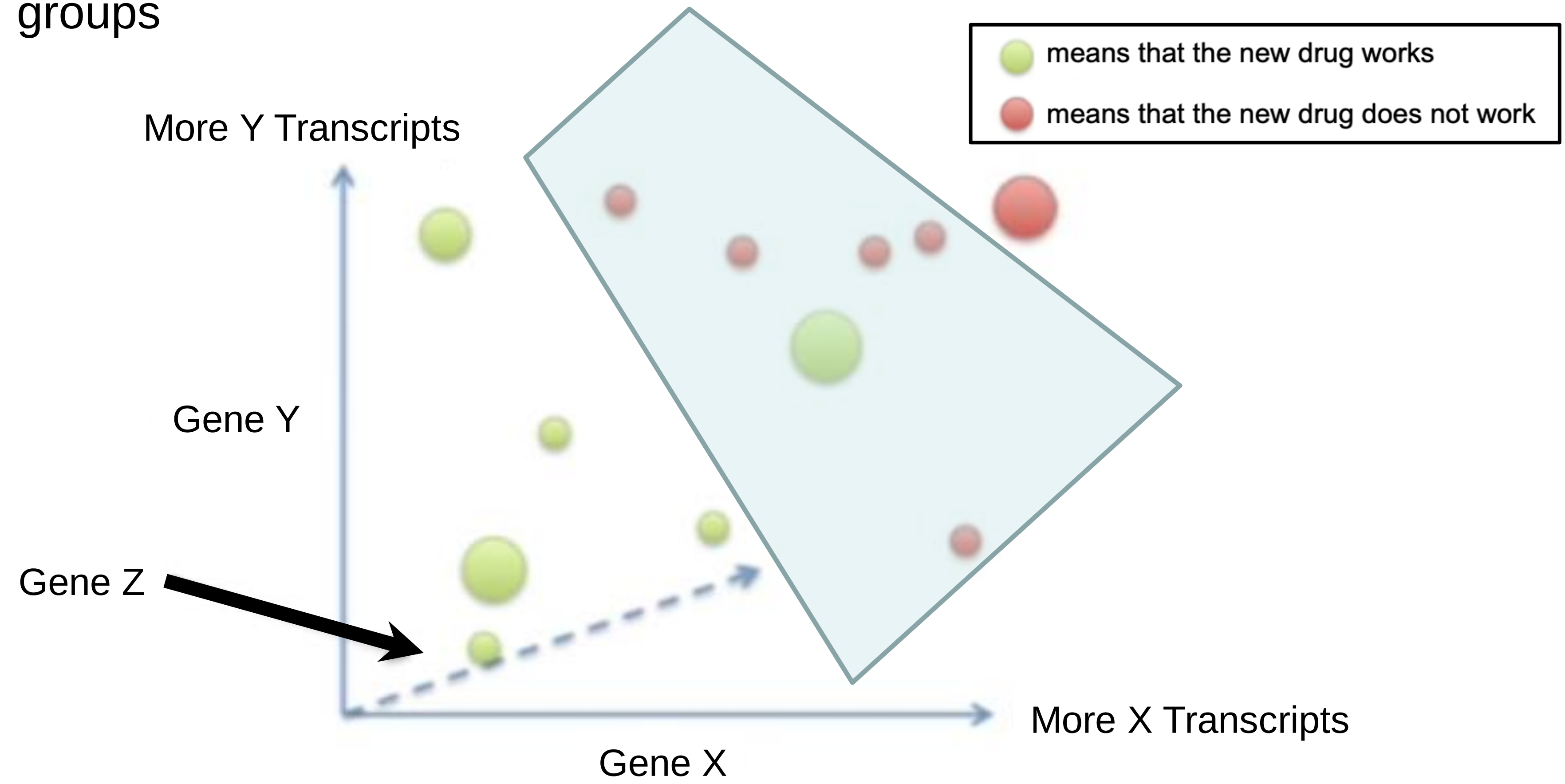
Gene X

More X Transcripts

Gene Z is located on Z-axis (depth). Big (small) circles are people close (further away) along the Z-axis.

How does LDA work?

As we are using 3 variables, we need a plane to try to separate the two groups


● means that the new drug works
● means that the new drug does not work
More Y Transcripts
Gene Y
Gene Z
More X Transcripts
Gene X

It’s **hard to decide** in a flat screen whether this plane separates well the two categories or not.

We need to rotate the plane to check from different angles to really know.

How does LDA work?

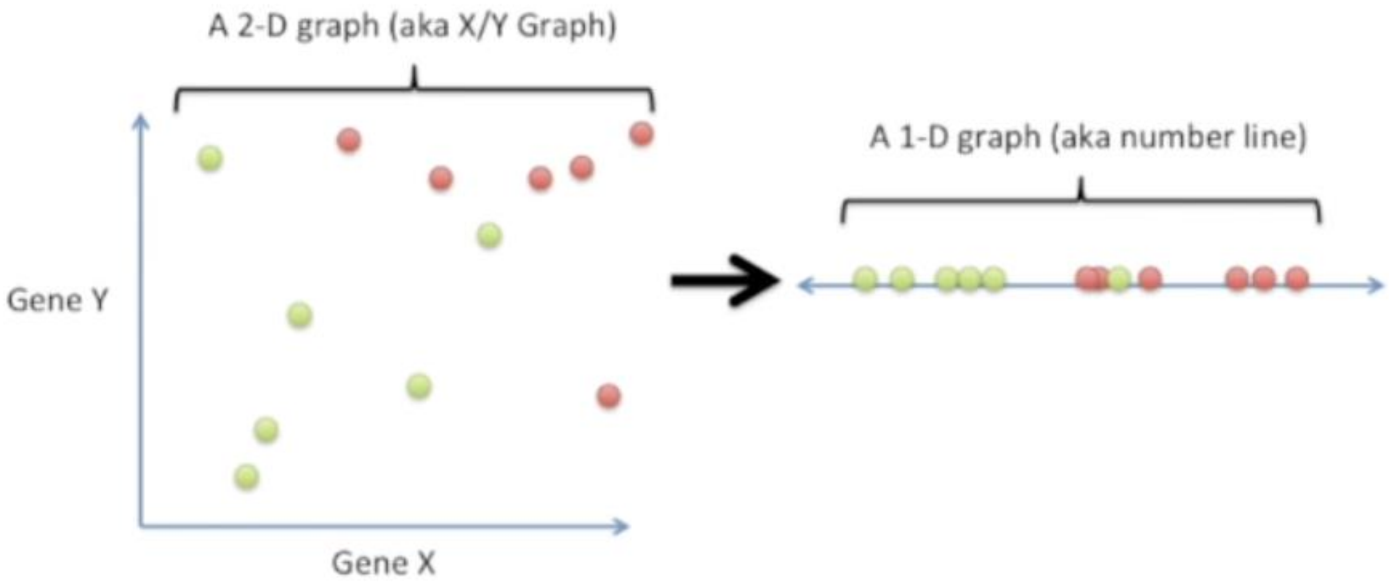
What if we need **4 or more genes** to separate the two categories (green and red balls)?

- We **cannot** draw a 4-D (or more-D) graph.
- That’s the same problem we have when we talk about PCA
- PCA reduces dimensionality focused on linear combination of the features (genes) that can explain the variation better (maximize the variation)
- However, we are not now interested in that.
- We are focused on maximizing the separability among the known categories. That’s what LDA does.

How does LDA work?

Let’s start with a simple example:

Reducing a 2-D graph to a 1-D graph with the goal of maximizing the separability between the two categories



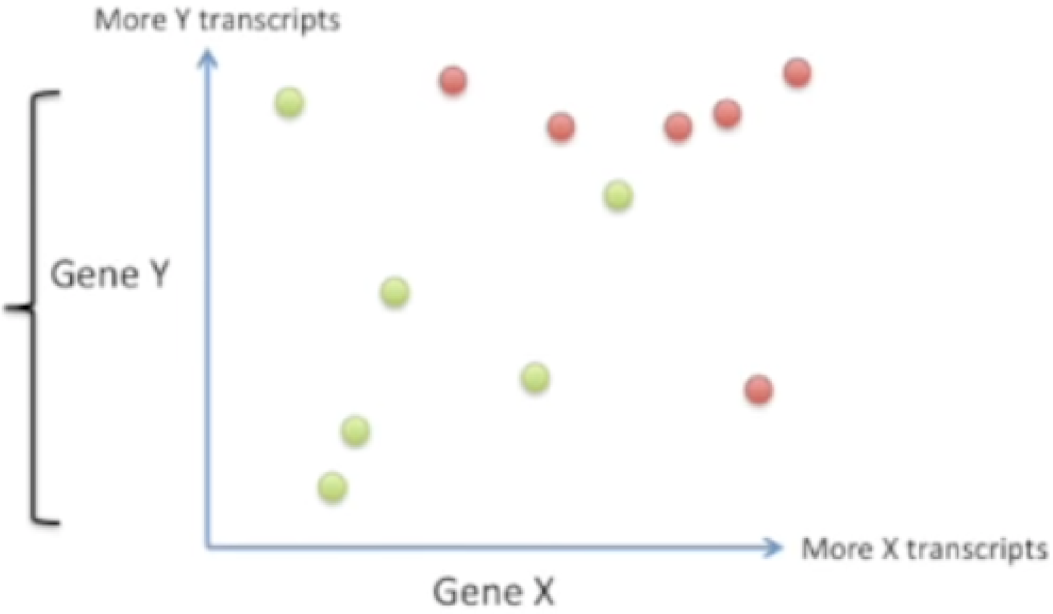
What is the best way to do it?

Let’s start by looking at a bad way

How does LDA work?

One bad option would be to ignore Gene Y

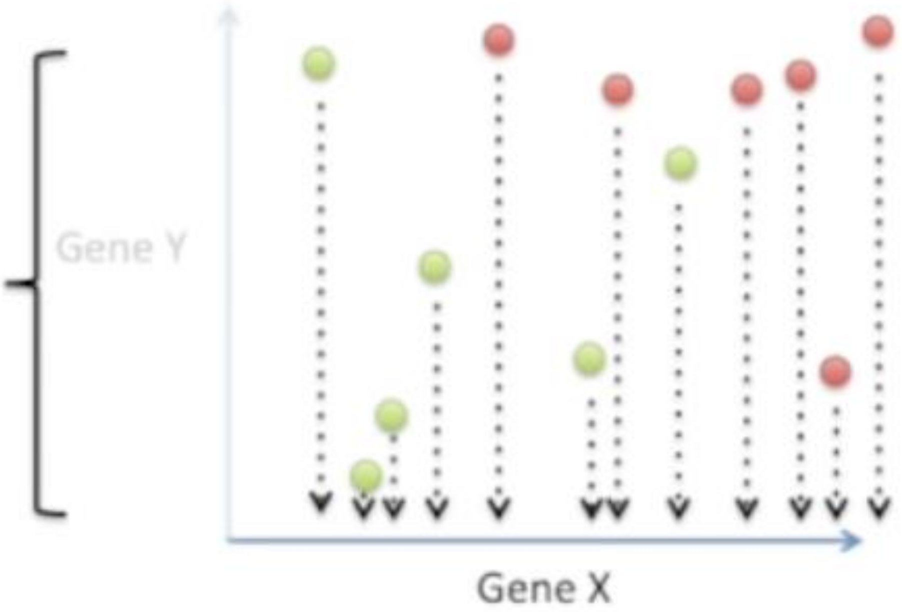
One bad option would be to ignore Gene Y



How does LDA work?

If we do that, we just project the data down onto the x-axis

One bad option would be to ignore Gene Y

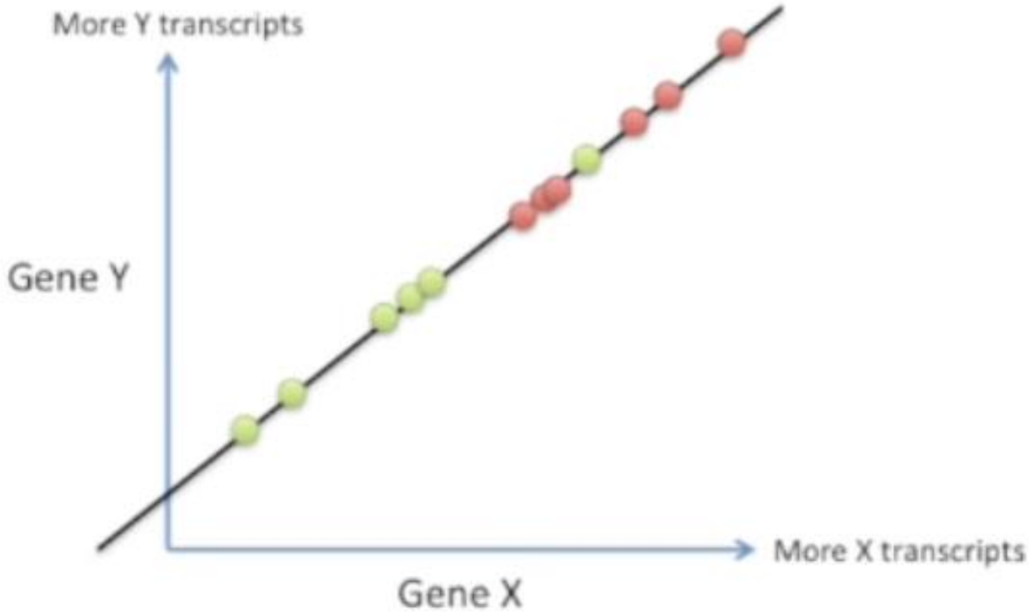


It **ignores** the useful information that Gene Y can provide

LDA **provides** a better way

How does LDA work?

Using LDA to reducing a 2-D graph to a 1-D graph to maximize the separability between the two categories



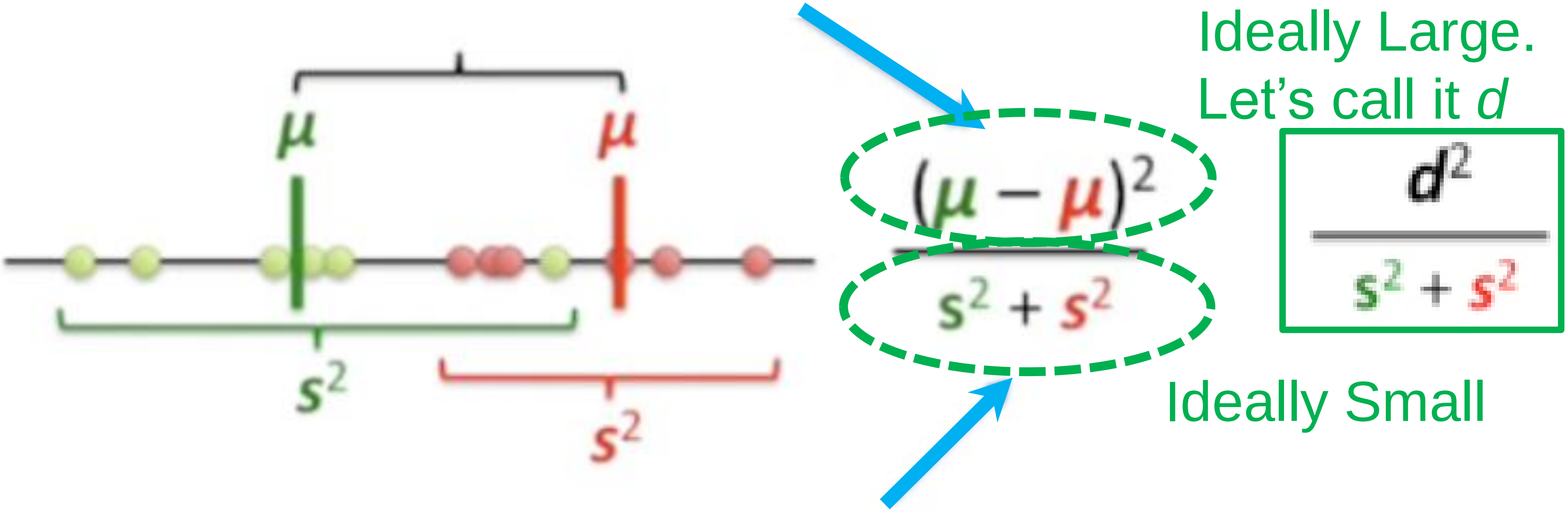
LDA create the information provided by two genes to create a new axis

LDA projects the data onto this new axis in a way to maximize the separation of the two categories. **How does it do that?**

How LDA creates a new axis?

The new axis is created according to two criteria (considered simultaneously):

1. Maximize the distance between the two means



Ideally Large. Let’s call it *d*

$$\frac{(\mu - \mu)^2}{s^2 + s^2}$$

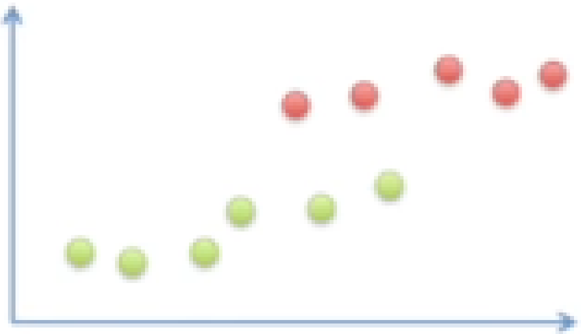
$$\boxed{\frac{d^2}{s^2 + s^2}}$$

Ideally Small

2. Minimize the variation (a.k.a. “scatter”, s^2) within each category

How does LDA work?

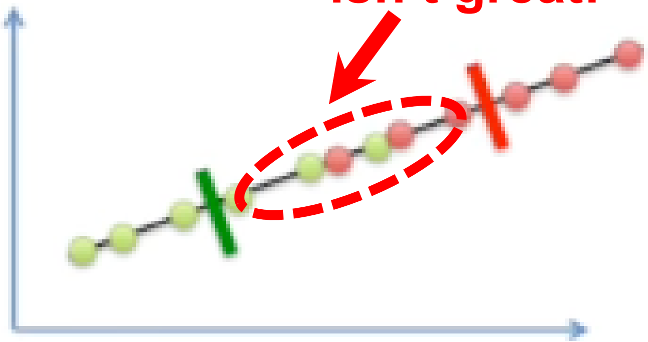
Importance of both: distance and scatter



- Overlap in y-axis
- Lot of spread in x-axis

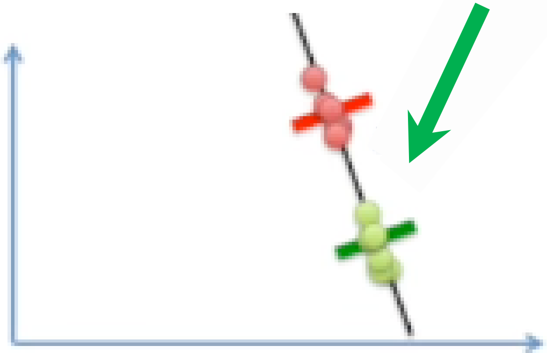
Only maximize **distance**

Overlap is bad. The separation **isn't great!**



Optimizing **distance** and **scatter**

We get very **nice separation**

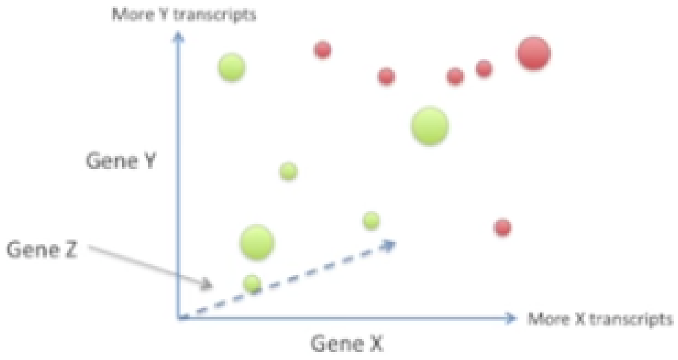


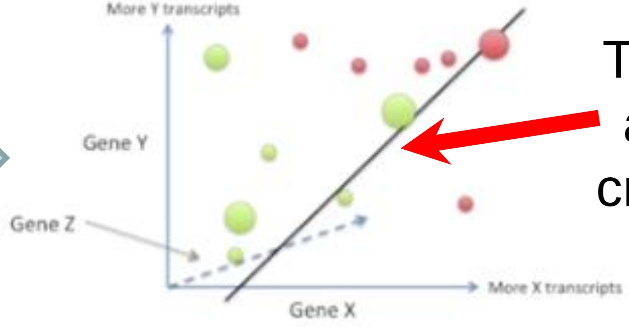
How does LDA work?

What if we have more than 2 dimensions in 2 categories?

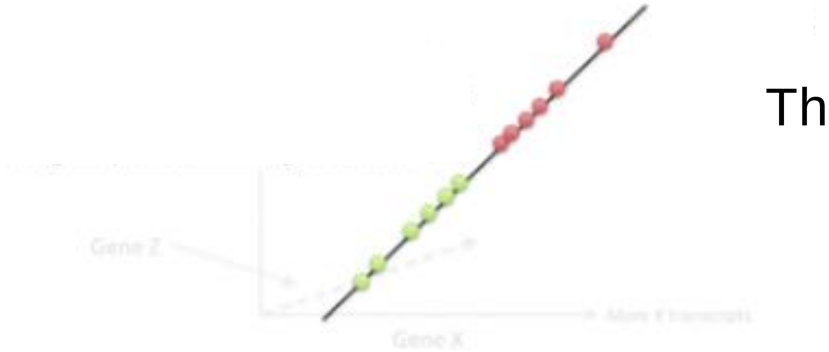
Good news: the process is the same

We create an axis that **maximizes the distance** between the means of the 2 categories while **minimizing the scatter.**





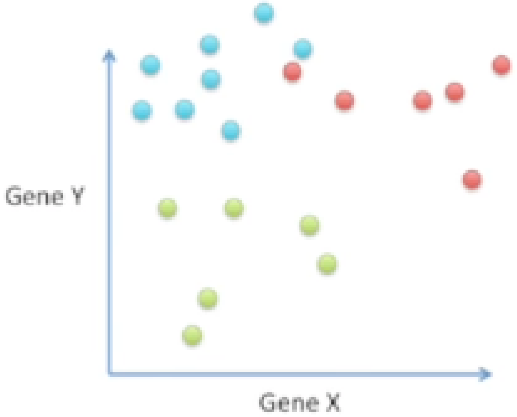
This is the new axis that LDA creates (max d, min scatter)



The data is projected in the new axis

How does LDA work?

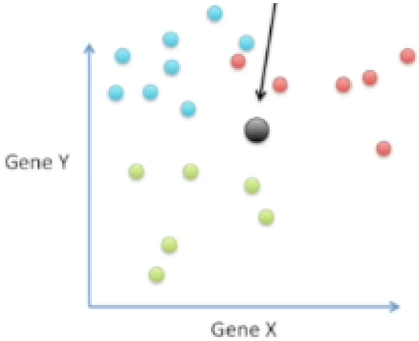
What if we have 3 categories?

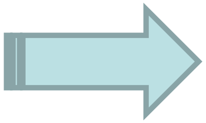


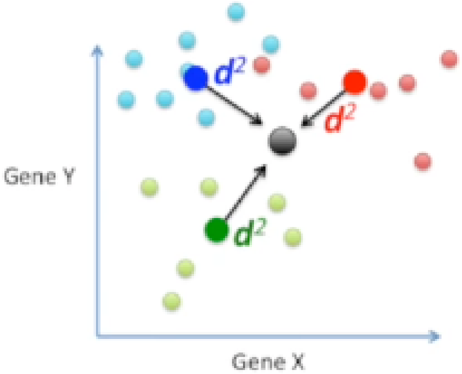
Two differences:

1) The distances among the means are calculated differently

Find the centroid to all the data set (<u>main centroid</u>)



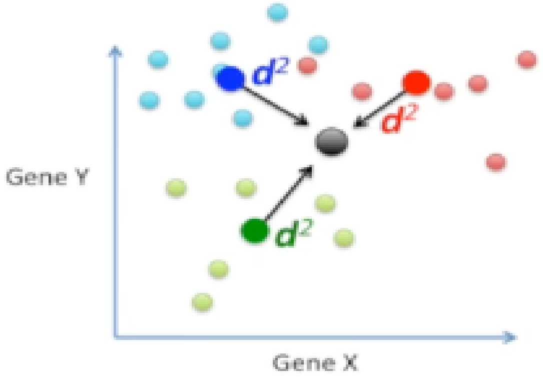




Measure the distances between the centroid in each category and the <u>main centroid</u>

How does LDA work?

What if we have 3 categories?

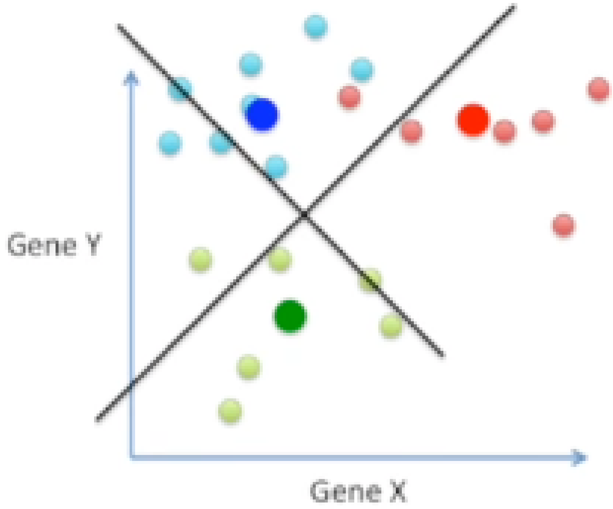


Now maximize the distance between each category and the centroid <u>while minimizing the scatter for each category</u>

So, equation to optimize

$$\frac{d^2 + d^2 + d^2}{s^2 + s^2 + s^2}$$

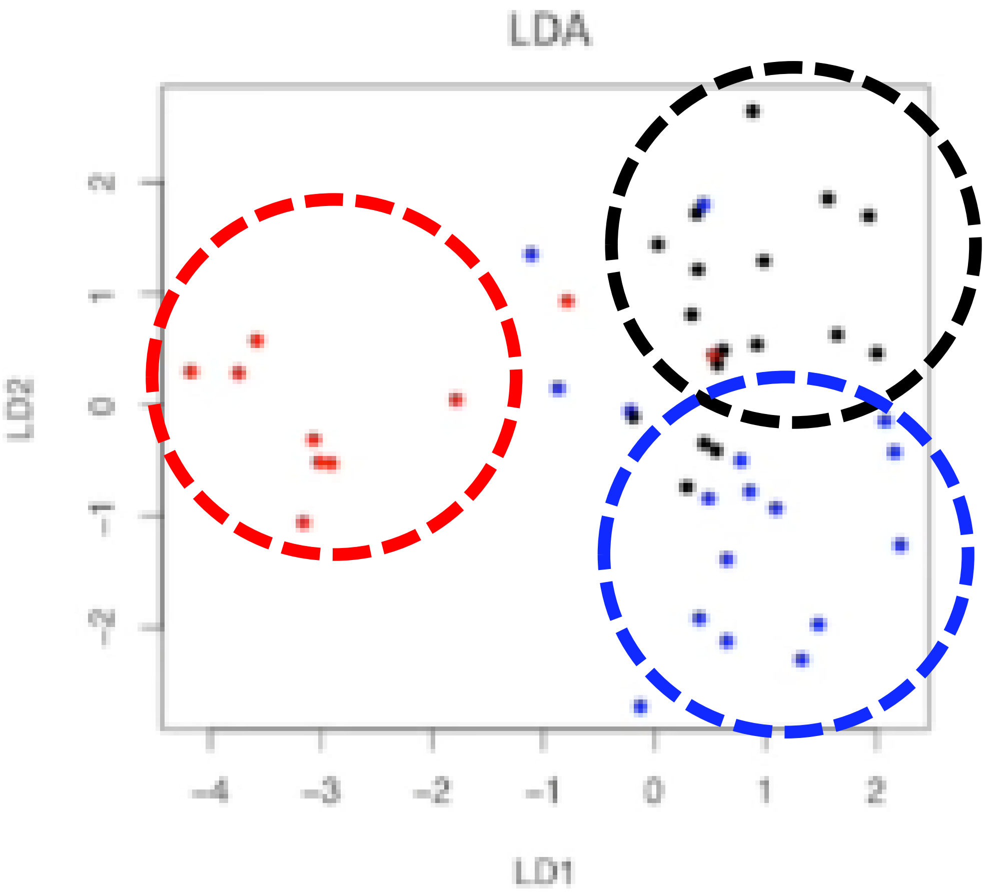
2) LDA creates 2 axes to optimize separation (scatter) of the data with 3 categories (#axes=#centroids -1)



With two genes, it is easy and the X/Y plot does not change much

But what if we used data from more than 2 genes (e.g. 10,000 genes => 10,000 dimensions)? **The same!**

LDA with 3 categories and 10,000 genes



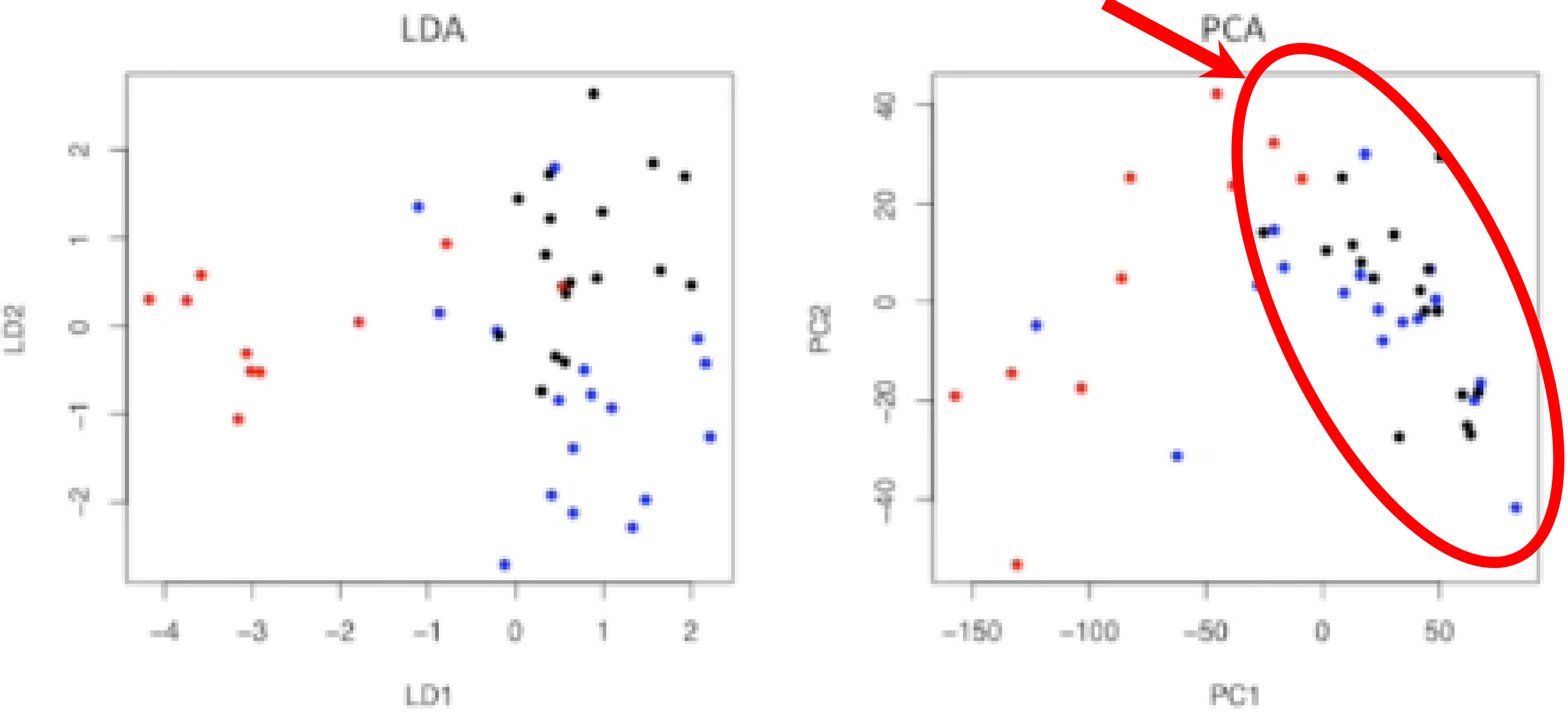
Plotting the raw data set would require 10,000 axes

We used LDA to reduce that number to two dimensions

Although the separation isn’t perfect, it is still easy to detect 3 separate categories

LDA vs. PCA for 10,000 genes

Separation isn't nearly good



Why does PCA work that bad?

Because PCA is not looking for separation. It is only looking for the genes with the most variation.

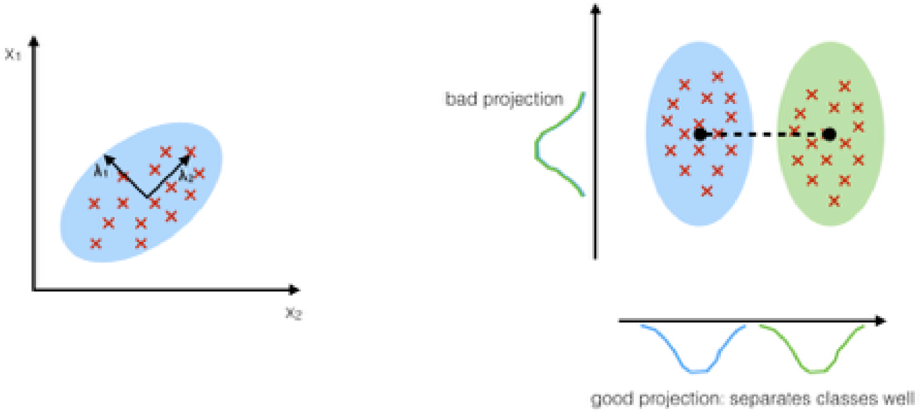
PCA vs. LDA

- PCA is an **unsupervised algorithm**. It ignores class labels.
- LDA is an **supervised algorithm**.
- Both do dimension reduction.
- Both rank the new axes in **order of importance**:
  - PCA looks the variables with the most variation.
    - PC1 accounts for the **most variation in the data,** PC2 does the second best job, etc.
  - LDA tries to maximize the separation of known categories
    - LD1 accounts for the **most separation between categories**, LD2 does the second best job, etc.

PCA vs. LDA

PCA:
component axes that
maximize the variance

LDA:
maximizing the component
axes for class-separation



- Both can let you see which variables are driving the new axes

  - In PCA means to check the loading scores.

  - In LDA means to check which variables correlate with the new axes.

- The 2 techniques can be used together for dimensionality reduction: PCA is used first followed by LDA.

LDA (maths)

- There are three basic steps:

1) Calculate the separability between different classes **(between-class variance)** defined as the distance between class means

$$S_b = \sum_{i=1}^{g} N_i(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

2) Calculate the **within-class variance** defined as the distance between the mean and the sample of every class.

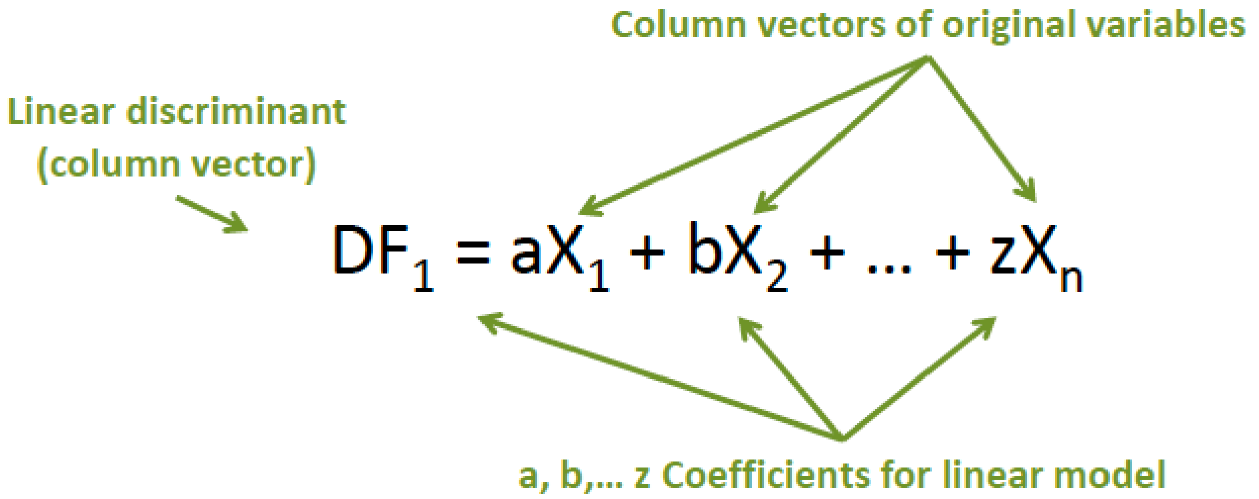
$$S_w = \sum_{i=1}^{g} (N_i - 1)S_i = \sum_{i=1}^{g}\sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

3) **Fisher's criterion**: Construct the lower-dimensional space that **maximizes** the between-class variance and **minimizes** within-class variance. In the equation below P is the lower-dimensional space projection.

$$P_{lda} = \arg\max_{P} \frac{\left|P^T S_b P\right|}{\left|P^T S_w P\right|}$$

LDA (maths)

- Discriminant functions (DF) (as the PCs in PCA) are linear combinations of the original variables



Column vectors of original variables

Linear discriminant (column vector)

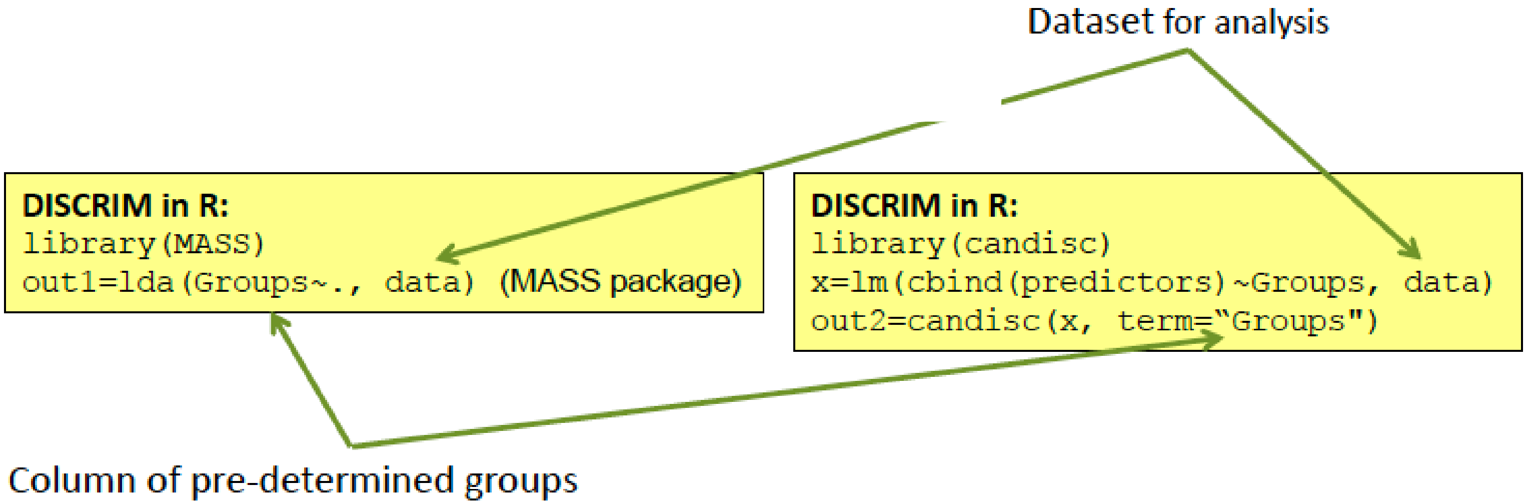
$$DF_1 = aX_1 + bX_2 + \ldots + zX_n$$

a, b,... z Coefficients for linear model

- LDA projects a DF for each observation in the dataset (like **PCA scores**)
- The previous slide set the matrix algebra to find the coefficients.

LDA in R

Two packages to run LDA in R: `MASS` or `candisc` package



Dataset for analysis

DISCRIM in R:

```
library(MASS)
out1=lda(Groups~., data)  (MASS package)
```

DISCRIM in R:

```
library(candisc)
x=lm(cbind(predictors)~Groups, data)
out2=candisc(x, term="Groups")
```

Column of pre-determined groups

The variables to include in the analysis must be specified:

- **If . is specified**: all variables in the dataset are included.
- Alternatively, we can specify an **equation** (e.g. $Y \sim X_1+X_2+X_3$)

If you want to use `candisc`:

1. Fit a linear regression model with group as a response.
2. Run `candisc` for performing a LDA

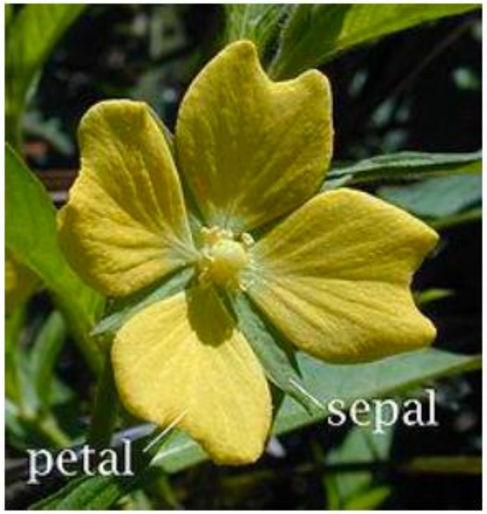
LDA in R (MASS package)

Example: Classification of Iris flowers (R data= iris)

Iris setosa

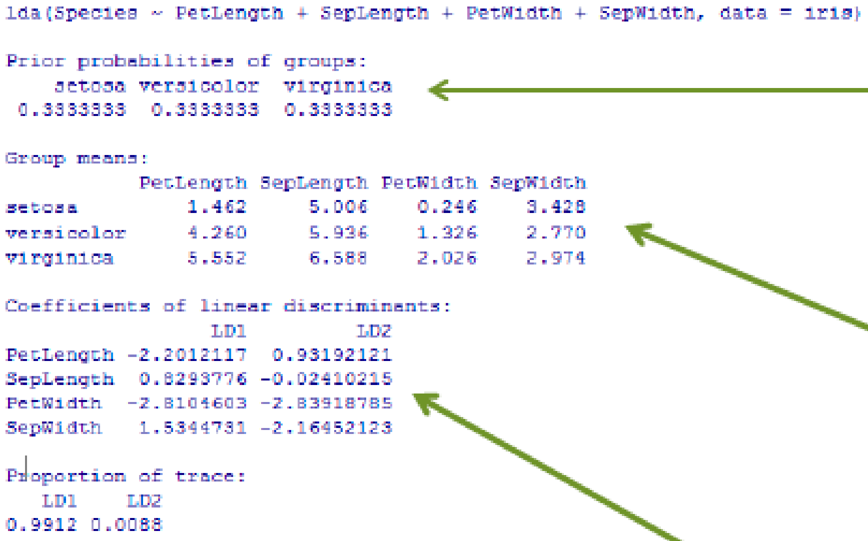
Iris versicolor

Iris virginica



Classify according to sepal/petal length/width

LDA in R (MASS package)



```
lda(Species ~ PetLength + SepLength + PetWidth + SepWidth, data = iris)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           PetLength SepLength PetWidth SepWidth
setosa         1.462     5.006    0.246    3.428
versicolor     4.260     5.936    1.326    2.770
virginica      5.552     6.588    2.026    2.974

Coefficients of linear discriminants:
                 LD1         LD2
PetLength -2.2012117  0.93192121
SepLength  0.8293776 -0.02410215
PetWidth  -2.8104603 -2.83918785
SepWidth   1.5344731 -2.16452123

Proportion of trace:
   LD1    LD2
0.9912 0.0088
```

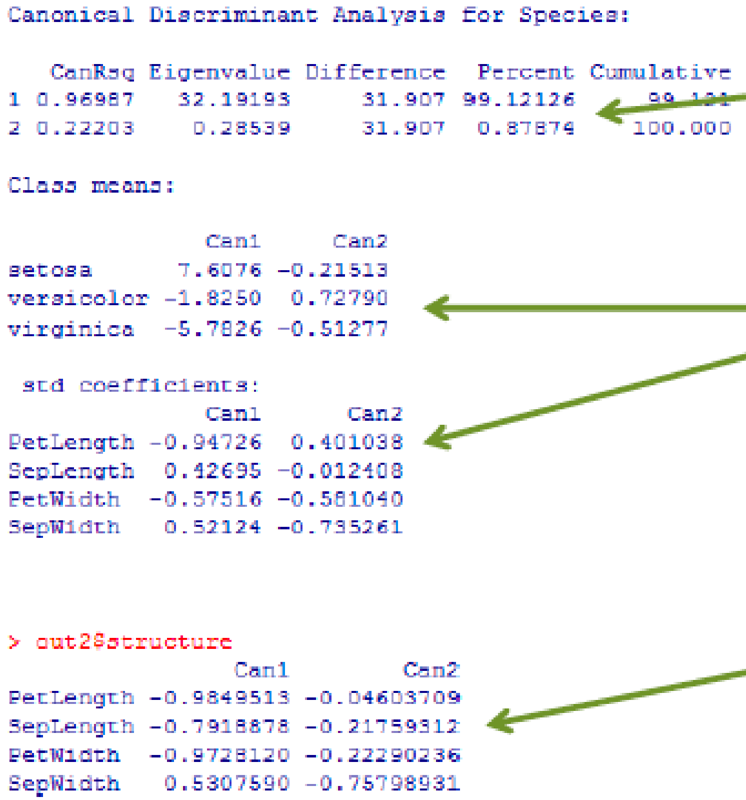
The initial probability of belonging to a group
(more important for predicting class)

Mean observation values for variables in each pre-defined group

Coefficients of linear discriminants are the solutions to our linear functions

Proportion of variance explained by linear discriminants

LDA in R (candisc package)



```
Canonical Discriminant Analysis for Species:

    CanRsq Eigenvalue Difference  Percent Cumulative
1 0.96987   32.19193     31.907 99.12126     99.121
2 0.22203    0.28539     31.907  0.87874    100.000

Class means:

                Can1     Can2
setosa        7.6076 -0.21513
versicolor   -1.8250  0.72790
virginica    -5.7826 -0.51277

 std coefficients:
                Can1      Can2
PetLength -0.94726  0.401038
SepLength  0.42695 -0.012408
PetWidth  -0.57516 -0.581040
SepWidth   0.52124 -0.735261


> out2$structure
               Can1        Can2
PetLength -0.9849513 -0.04603709
SepLength -0.7918878 -0.21759312
PetWidth  -0.9728120 -0.22290236
SepWidth   0.5307590 -0.75798931
```

Proportion of variance explained by linear discriminants

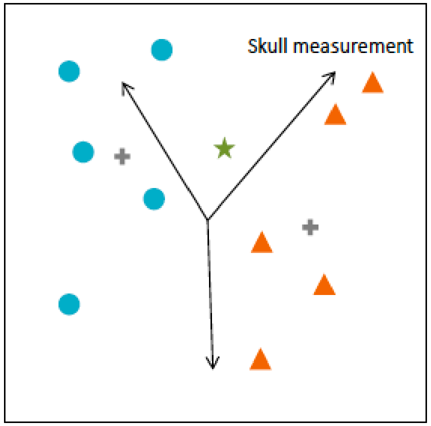
Mean discriminant values for each pre-defined group

Standard error of the means are also given

By querying the analysis structure we can see the discriminant loadings which tell us the relationship between the DF values and the original variables (like PCA)

LDA. Predictions. First way

Problem: A new skull is found but we don’t know whether it belongs to homo erectus or homo habilis or if it’s a new group?



Popular method in taxonomy and anthropology

How can we predict?

1. Calculate group centroid
2. Find out which centroid is the closest position to the unknown data point.
3. New groups are defined when we find a significant difference between new find and predefined groups.

LDA. Predictions. Second way

We define $\hat{\delta}_k(x)$ as the estimated probability that *x* belongs to that particular class (calculate using a Gaussian distribution).

1. We use conditional probability to turn these into estimates for class probabilities:

$$\hat{P}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{c=1}^{K} e^{\hat{\delta}_c(x)}}$$

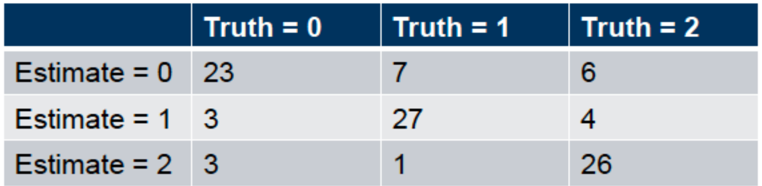
So classifying to the largest $\hat{\delta}_k(x)$ amounts to classifying to the class for which $\hat{P}(Y = k|X = x)$ is largest.

For example, when K=2, we classify to class 2 if

$$\hat{P}(Y = 2|X = x) \geq 0.5$$

LDA. Judging the performance

- Calculate the misclassification rate (confusion matrix): the proportion of the “known” individuals that would be misclassified using the DF to classify them.

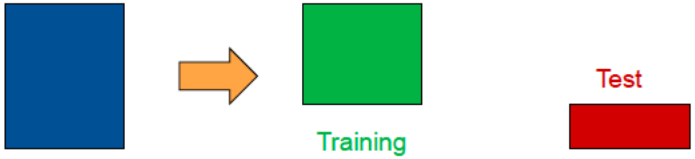


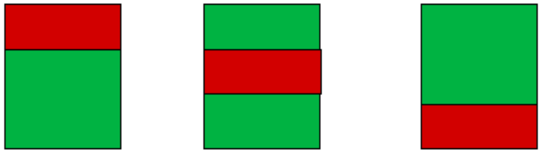
	Truth = 0	Truth = 1	Truth = 2
Estimate = 0	23	7	6
Estimate = 1	3	27	4
Estimate = 2	3	1	26

Misclassification rate:
1- (sum(diagonal entries)/total)=
1-(76/100)=0.24 (24%)

Problem? Since the DF has been derived from the “known” individuals, the results underestimates the misclassified rate (overfit)

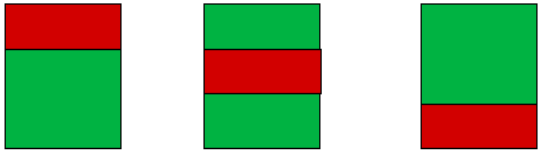
- Two approaches:
  1. Separate Training & Test Data
  2. Cross-validation (CV, or “leave-one-out”) method: every row is the test case once, the rest is training data.





LDA. Predictions. First way

2. Cross-validation (CV, or “leave-one-out”) method: every row is the test case once, the rest is training data.



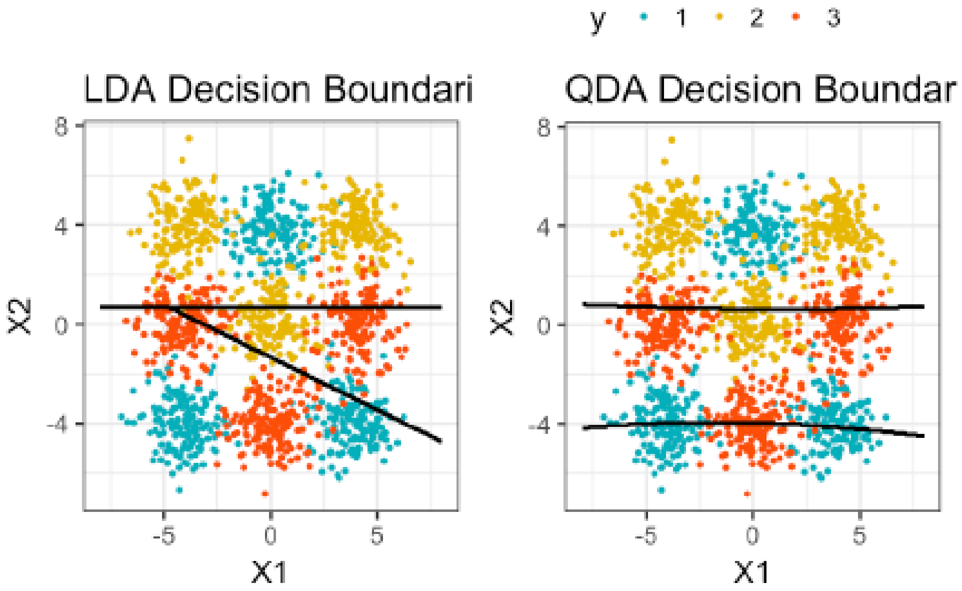
This uses all but one of the “known” individuals to derive a classification rule and then, based on that rule, **classifies the other individual**.

This is done (separately) for all the known individuals, and the misclassification rate is the proportion of those classifications that are incorrect.

The R function `predict` do that for lda objects.

Extensions to LDA

- **Quadratic Discriminant Analysis (QDA)**
  - ○ **More flexible** than LDA: doesn't assume equality if var/cov.
  - ○ Each class uses its **own covariance matrix**.
  - ○ **R function qda** in the `MASS` package
  - ○ Problem: it can **overfit** the data (no generalizable to future observations).
  - ○ **LDA** better for **small** training sets. **QDA** for **large** ones.



Source: https://www.r-bloggers.com/2013/07/a-brief-look-at-mixture-discriminant-analysis/

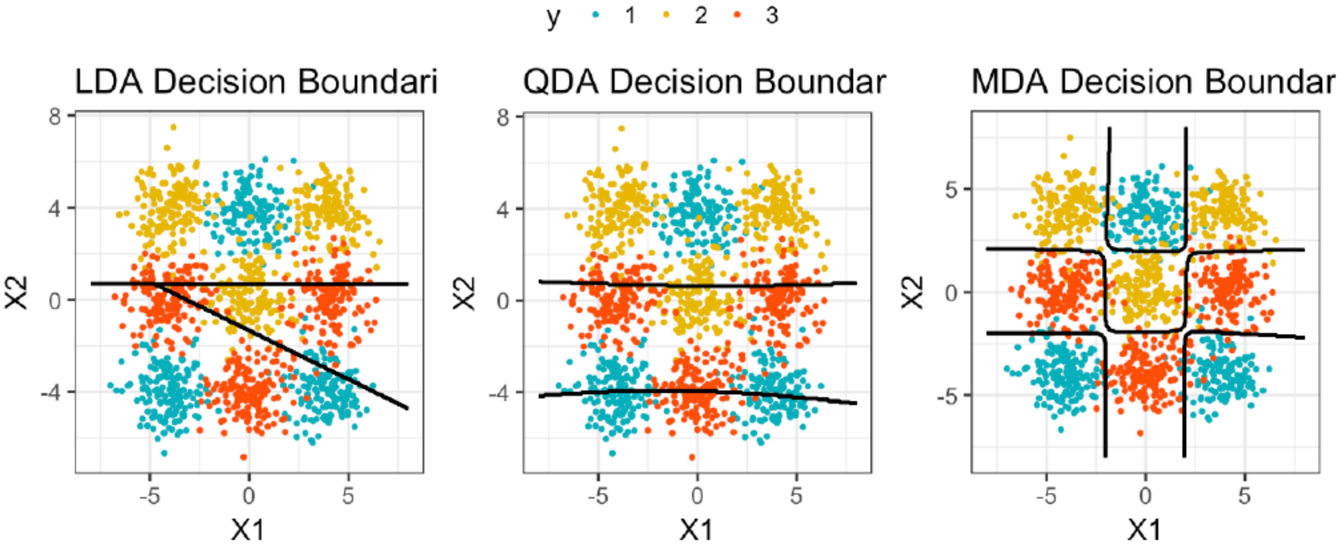
Extensions to LDA

- **Flexible Discriminant Analysis (FDA)**:
  - Where non-linear combinations of predictor such as **splines** is used.
  - Useful to model multivariate **non-normality or non-linear** relationships among variables within each groug
  - **R function `fda`** in the `mda` package (https://rdrr.io/cran/mda/man/fda.html)

- **Regularized Discriminant Analysis (RDA)**:
  - Builds a classification rule by regularizing the group covariance matrices, which allows a **more robust model against multicollinearity**.
  - Very useful for data set containing highly **correlated predictors.**
  - RDA is a compromise between LDA and QDA.
  - **R function `rda`** in the `KlaR` package (https://www.rdocumentation.org/packages/klaR/versions/0.6-15/topics/rda)

Extensions to LDA

- **Mixture Discriminant Analysis (MDA)**:
  - The **LDA** classifier assumes that each class comes from a single **normal (or Gaussian) distribution**. That might be very restrictive!
  - For MDA, there are classes, and each class is assumed to be a **Gaussian mixture of subclasses**, where each data point has a probability of belonging to each class (more detail in Week 11)



- When the populations are not close to multivariate normal, an alternative to LDA is **logistic regression**.