11. Wrap Up

OSCAR GONZALEZ, PHD

MACHINE LEARNING

This Semester

Data science and machine learning

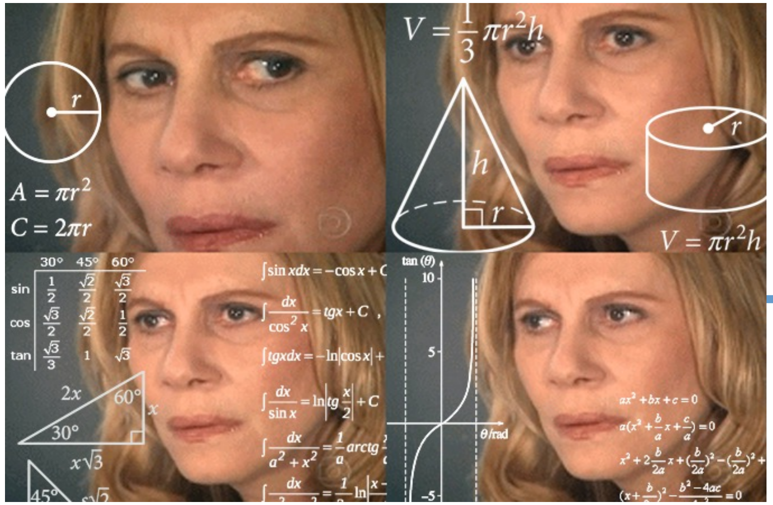
- Find patterns and build models that are generalizable

Supervised Learning

- Classifiers – predict a binary variable
- Regression algorithms – predict a continuous variable

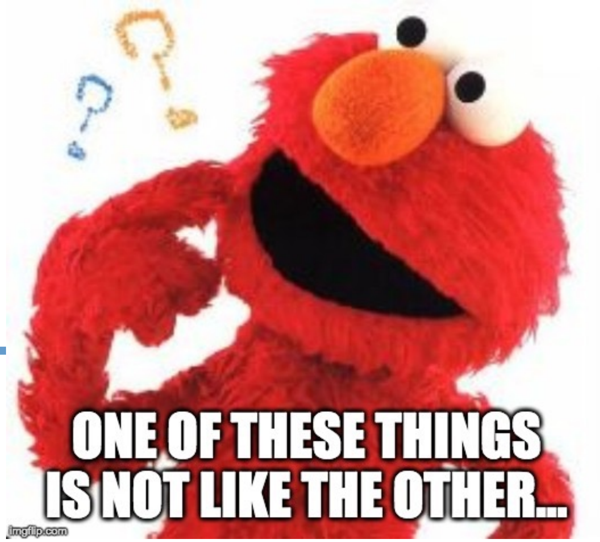
Core machine learning concepts

- Overfitting
- Bias-variance trade-off
- Regularization

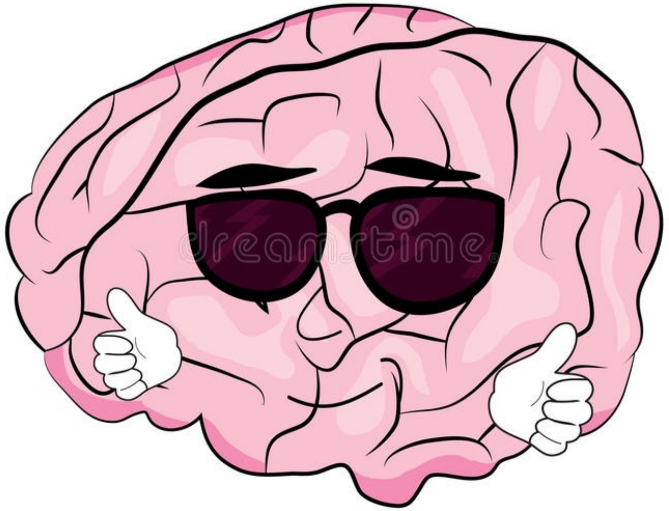


The Symbolists

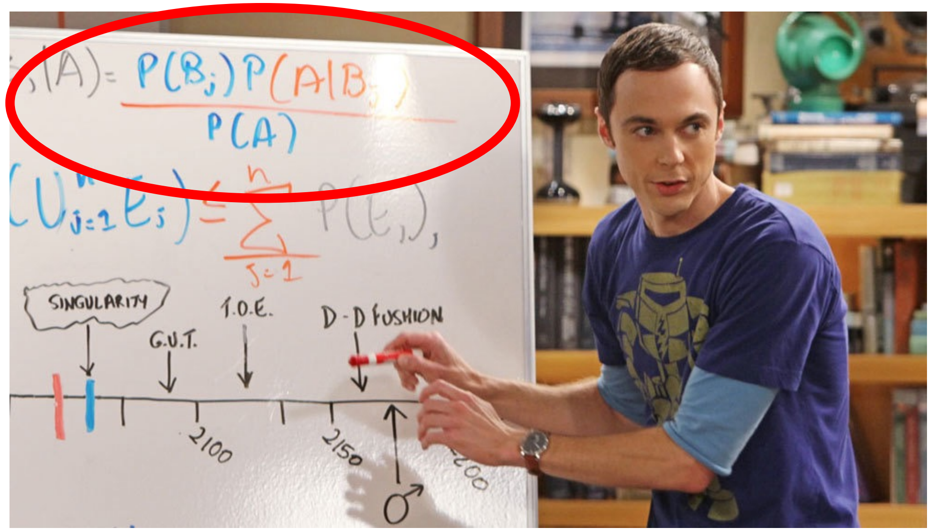
The Evolutionaries



The Analogizers



The Connectionists



The Bayesians

Domingos' Five Groups of Machine Learning

Group	Origins	Master Algorithm	Method
Symbolists	Logic, philosophy	Inverse deduction	Rule-based Method
Connectionists	Neuroscience	Backpropagation	Neural Networks
Evolutionaries	Evolutionary biology	Genetic programming	Genetic Algorithm
Bayesians	Statistics	Probabilistic inference	Bayes Theorem
Analogizers	Psychology	Kernel Machines	Support Vector Machines

Models that generalize

Cross-validation assesses overfitting, but does not modify the model…

Regularization

- Add constraints/penalties during estimation
- In regression – shrink coefficient size

$$Lasso = RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$Ridge = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

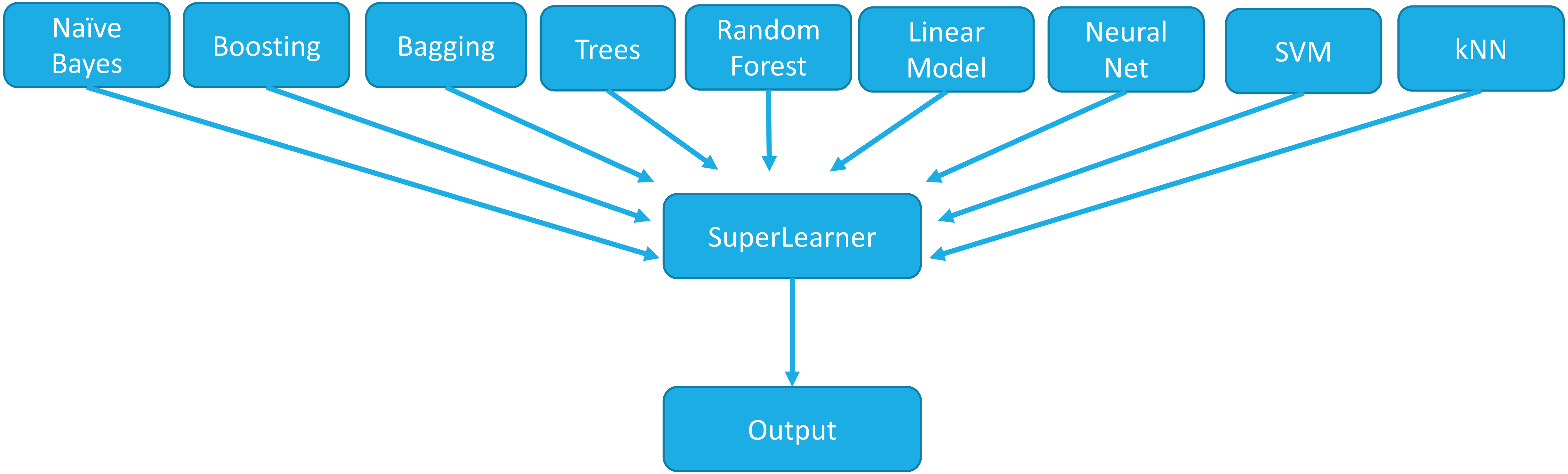
Feature selection

- Simpler models are likely to generalize
- Filter, wrapper, and embedded methods

Perhaps now you’re feeling like…

Stacker (or SuperLearner, *Metalearner*)

Goal: Boost prediction accuracy by combining predicted values



Today’s Agenda

1. Ethics Redux
2. Unsupervised Learning
3. Clustering methods
   - Hierarchical clustering
   - K-means clustering
4. Dimension reduction methods
   - Principal components
   - Factor analysis
5. Wrap Up

Ethics and Algorithm Bias

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Ethical Implications

Aim for responsible/sensitive application of machine learning

- Realize our responsibilities

Algorithms might discriminate against certain minority groups

Algorithm bias in high-stakes applications…

- Self-driving cars
- Advertisements
- Criminal justice
- Child welfare
- Social media pages

Our training in psychology could make a difference

Informed Consent

Any psychology study requires *informed consent*

Any psychology study requires debriefing the participant into what they did and what we will do with the info

How would this look like if we do it in the big data world?

- *“Your click will be used for ad targeting”*
- Annoying… but transparent

Covariate collection

In a good psychology study, we collect demographic information

- Sometimes used to test differential item functioning

Should apps be allowed to ask for demographic information?

- Gender, race, or if we have a health condition/disability
- If not, how do you evaluate if the algorithm is fair?
- Proxies – zip code for race, income, education status

Research Methods

If we want to generalize our findings…

- Collect data on the outcome and stuff we think is related
- Representative participants in controlled environments
- Control for covariates

Problem: most of the machine learning algorithms are carried out in *found* data

- Analyze what has been collected
- Our biases bleed over to the data (e.g., incarcerating minorities at a higher rate)
- … and we use these data to train our models

A Data Science Checklist

❑ Have we listed how this technology can be attacked or abused?

❑ Have we tested our training data to ensure it is fair and representative?

❑ Have we studied and understood possible sources of bias in our data?

❑ Does our team reflect diversity of opinions, thoughts, and backgrounds?

❑ What kind of user consent do we need to collect to use the data?

❑ Do we have a mechanism for gathering consent from users?

❑ Have we explained clearly what users are consenting to?

Loukides, M., Mason, H., & Patil, D. J. (2018). *Ethics and Data Science*. O'Reilly Media, Inc.

A Data Science Checklist

❑ Do we have a mechanism for redress if people are harmed by the results?

❑ Can we shut down this software in production if it is behaving badly?

❑ Have we tested for fairness with respect to different user groups?

❑ Have we tested for disparate error rates among different user groups?

❑ Do we monitor for model drift to ensure our software remains fair over time?

❑ Do we have a plan to protect and secure user data?

Loukides, M., Mason, H., & Patil, D. J. (2018). *Ethics and Data Science*. O’Reilly Media, Inc.

A Golden Rule for Data Use

Treat others' data as you would have others treat your own data

Consent

- Can the data be used? – Typically, a binary decision

Clarity

- Simple explanation for data use – Twitter: public tweets… but for sale(?)

Consistency

- Keeping the trust over time – Facebook and data leaks

A Golden Rule for Data Use

Treat others' data as you would have others treat your own data

Control

- Who controls the data? Can I change my mind?

Consequences

- Be aware of the harm we could cause
- Be aware of the unknown unknowns

Design with equity in mind

Machines do not make decisions. Humans make decisions.

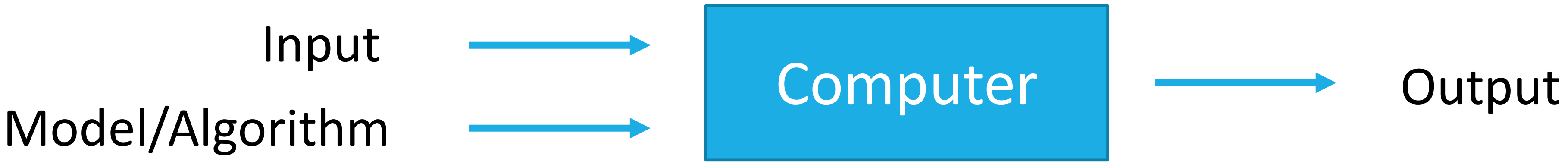
Unsupervised Learning

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Rough Idea

Traditional Analyses



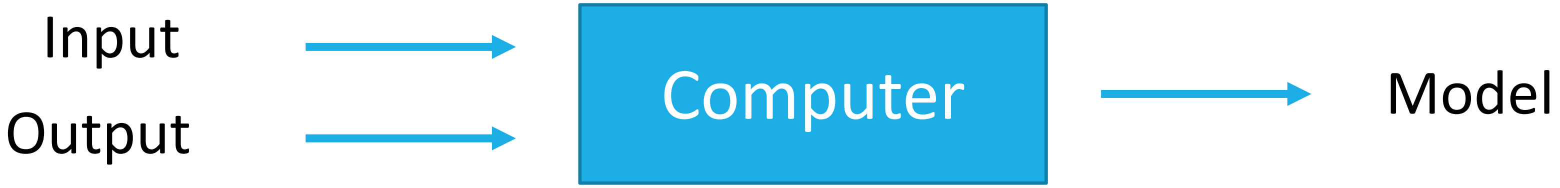
Input

Model/Algorithm

Computer

Output

Machine Learning



Input

Output

Computer

Model

*Learn by Example

Two Types of Methods

Prediction methods

- Y: outcome
- X: a matrix of predictors
- Learn by example: *here are some respondents with high CESD scores and their characteristics, and respondents with low CESD scores and their characteristics*

Descriptive methods

- Just a set of variables (no outcome)
- Fuzzy goals – how is the data organized?
- Make sense of the examples: *let’s find subgroups of CESD respondents*

Two Types of Methods

~~Prediction methods~~ Supervised Learning

- Y: outcome
- X: a matrix of predictors
- Learn by example: *here are some respondents with high CESD scores and their characteristics, and respondents with low CESD scores and their characteristics*

~~Descriptive methods~~ Unsupervised Learning

- Just a set of variables (no outcome)
- Fuzzy goals – how is the data organized?
- Make sense of the examples: *let’s find subgroups of CESD respondents*

Supervised Learning

We know more about supervised learning than unsupervised learning

If you’re asked to predict a variable, we know how to…

- Use algorithms to estimate p(y|X)
- Assess the model via cross-validation
- Evaluate the model (classification rate, mean squared error…)
- Describe our results

Unsupervised Learning

Goal: discover meaningful patterns in data without an outcome

- We <u>are not</u> interested in prediction – there's no outcome

Inherently exploratory, descriptive methods

More challenging than supervised methods

- No straight-forward way to check our answer

Questions

- Can we extract similar groups/clusters from the data?
- Are there disease subtypes?
- Visualizing high-dimensional data

Unsupervised Learning Methods

Cluster Analysis: finding groups of respondents who are similar

- Based on distance measures (remember KNN?)
- Several flavors: hierarchical, *k*-means …

Dimension Reduction: from many dimensions to less dimensions

- One dimension = one variable/predictor
- Linear combinations of predictors – principal components

The Challenges Ahead

1. No direct way to do cross-validation
   - With supervised learning, we see how well we predicted the outcome
   - In unsupervised learning… ?
2. Difficult to validate the results
   - How do we know that the groups we find are really there?
3. Difficult to compare models
   - How do we choose which clustering strategy to use?

Clustering

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Clustering

Split the dataset into meaningful groups where observations are *similar* to each other

Used to find structure in the dataset

- Depression patients – are there subtypes?
- Grocery buyers – market segmentation?

Two methods

Hierarchical clustering

- We do not know in advance how many clusters we have
- Bottom-up approach
- Sequential approach

K-means clustering

- Assigning observations to a pre-determined number of clusters
- Top-downish approach
- Iterative approach

Two methods

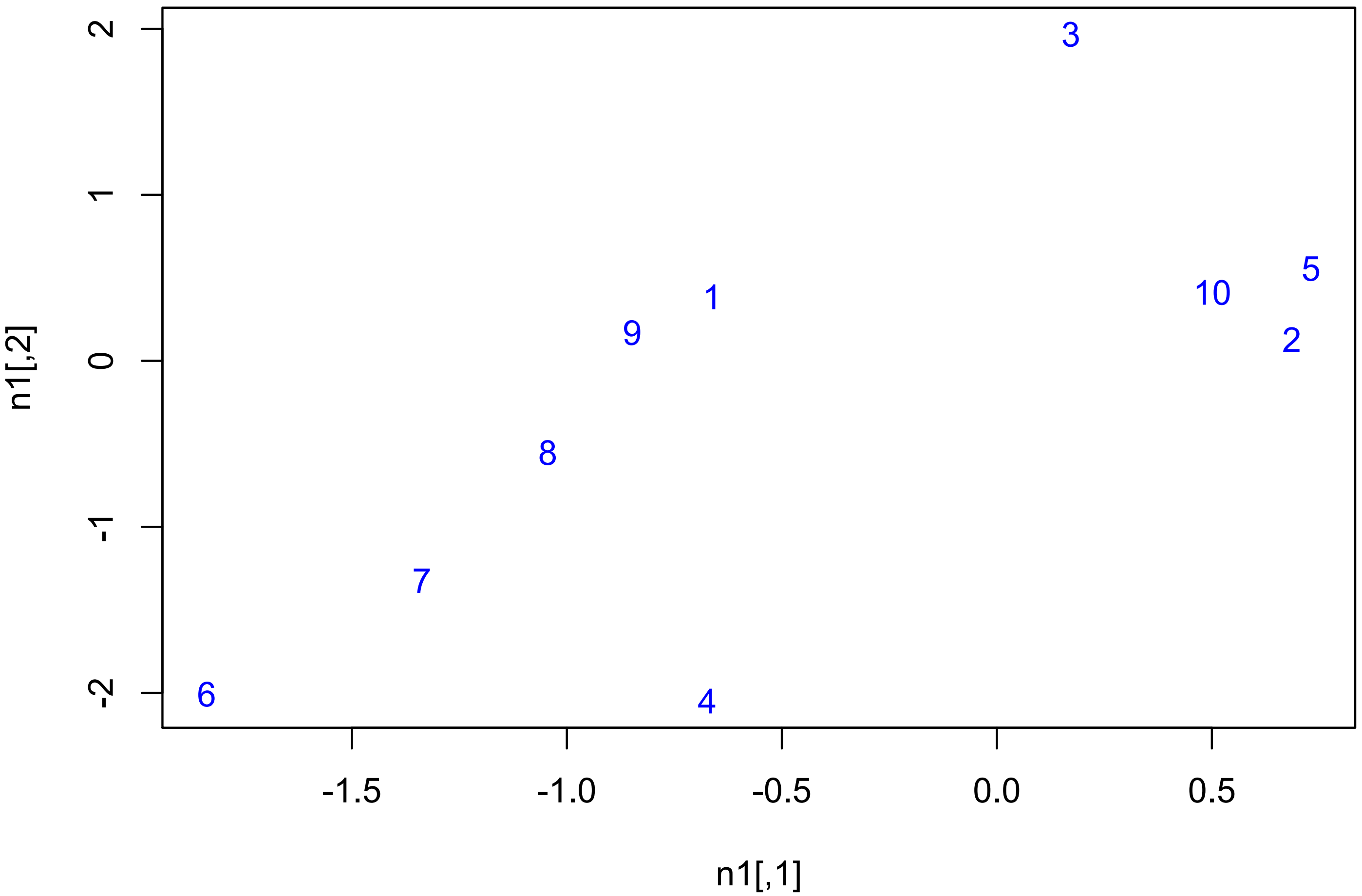
Hierarchical clustering

- We do not know in advance how many clusters we have
- Bottom-up approach
- Sequential approach

K-means clustering

- Assigning observations to a pre-determined number of clusters
- Top-downish approach
- Iterative approach

Example



Hierarchical Clustering Algorithm

Estimate a similarity measure between every pair of observations (i.e., Euclidean distance)

$$d(x1, x2) = \sqrt{(x_{i1} - x_{i'1})^2 + (x_{i2} - x_{i'2})^2}$$

Algorithm

- 1. Every observation starts with its own cluster (*n* clusters)
- 2. Fuse the observations that are most similar to each other (*n-1* clusters)
- 3. Continue until all observations go into one cluster

Other dissimilarities

- Manhattan, correlation-based, binary

Not even *close*!

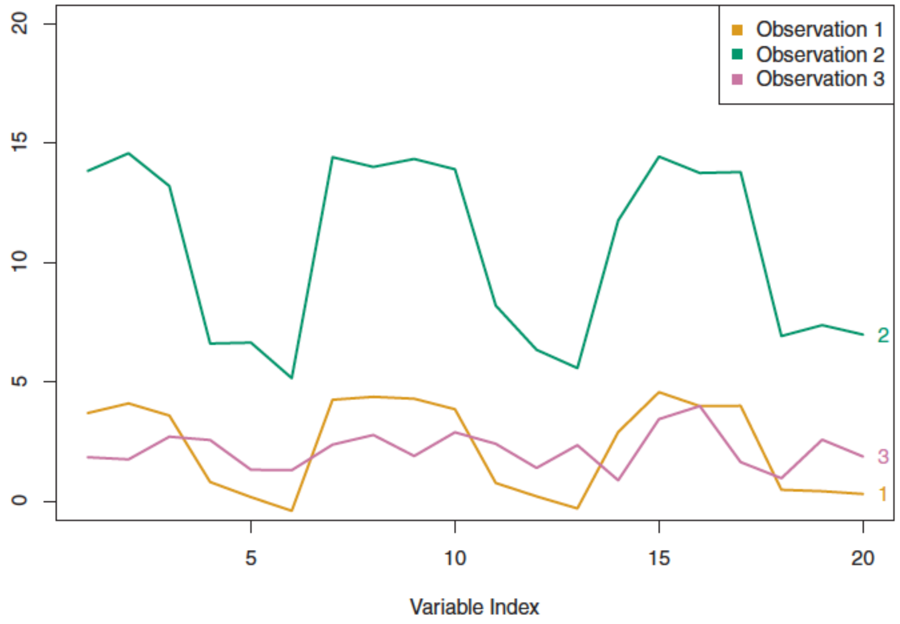
Plot: variable index vs. value

Consider observations 1 and 3

- Similar values on variables, but different patterns
- Close based on Euclidean distance, but far based on correlation-based distance

Consider observations 1 and 2

- Different values on variables, but similar patterns
- Close based on correlation-based distance, but far based on Euclidean distance



Linkage – dissimilarity across clusters

Compute pairwise dissimilarities between cluster A and cluster B

Complete

- Dissimilarity is the largest intercluster dissimilarity
- Yields balance solutions to cases per cluster
- Not as sensitive to outliers

Single

- Dissimilarity is the minimum intercluster dissimilarity
- Most flexible in terms of shapes
- Sensitive to outliers/noise and tends to fuse observations one-at-a-time

Linkage – dissimilarity across clusters

Compute pairwise dissimilarities between cluster A and cluster B

Average

- Dissimilarity is the average intercluster dissimilarity
- Trade off between single and complete linkage

Centroid

- Dissimilarity from the centroids of each cluster (mean of all of the vars in cluster)
- Note that the centroid might not a point in the dataset….

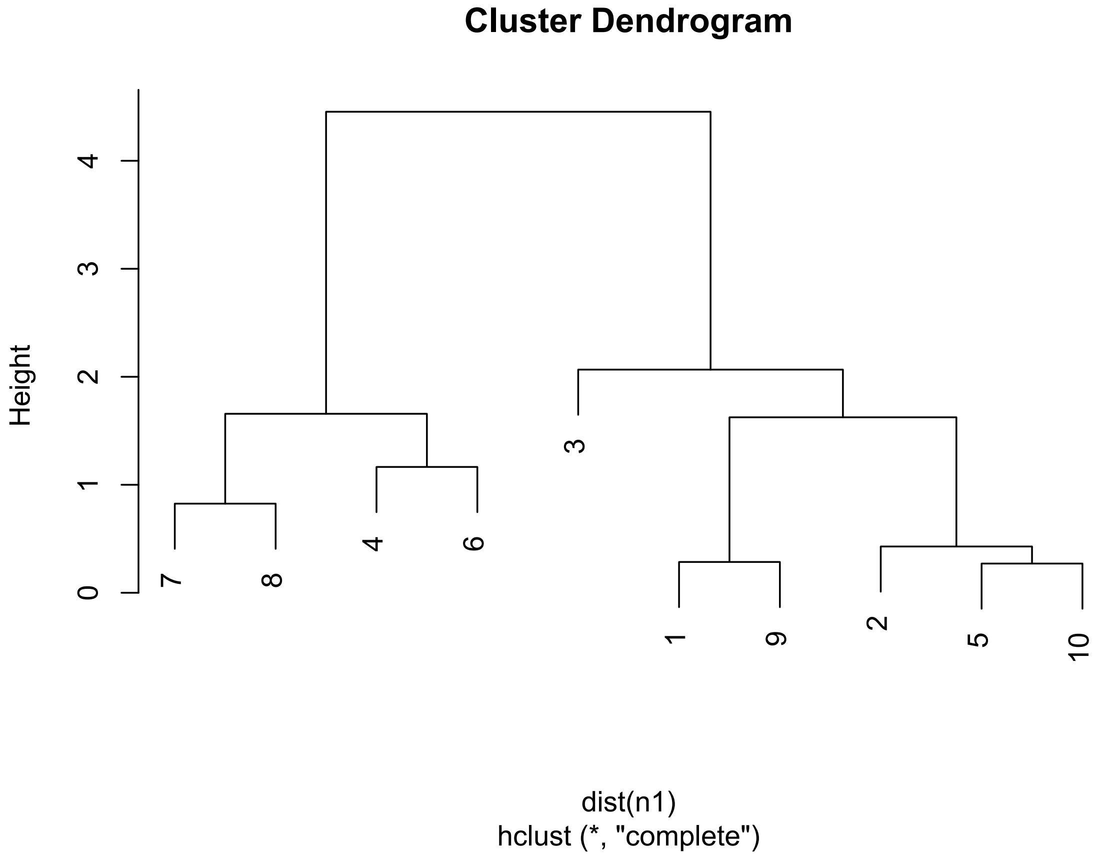
Dendogram

Tree-like structure to represent clusters

- Read from the bottom, up
- Each leaf represents an observation
- Observations start to fuse into clusters
- Higher up leaves fuse with branches

Earlier fusing – more similar

Later fusing – more different



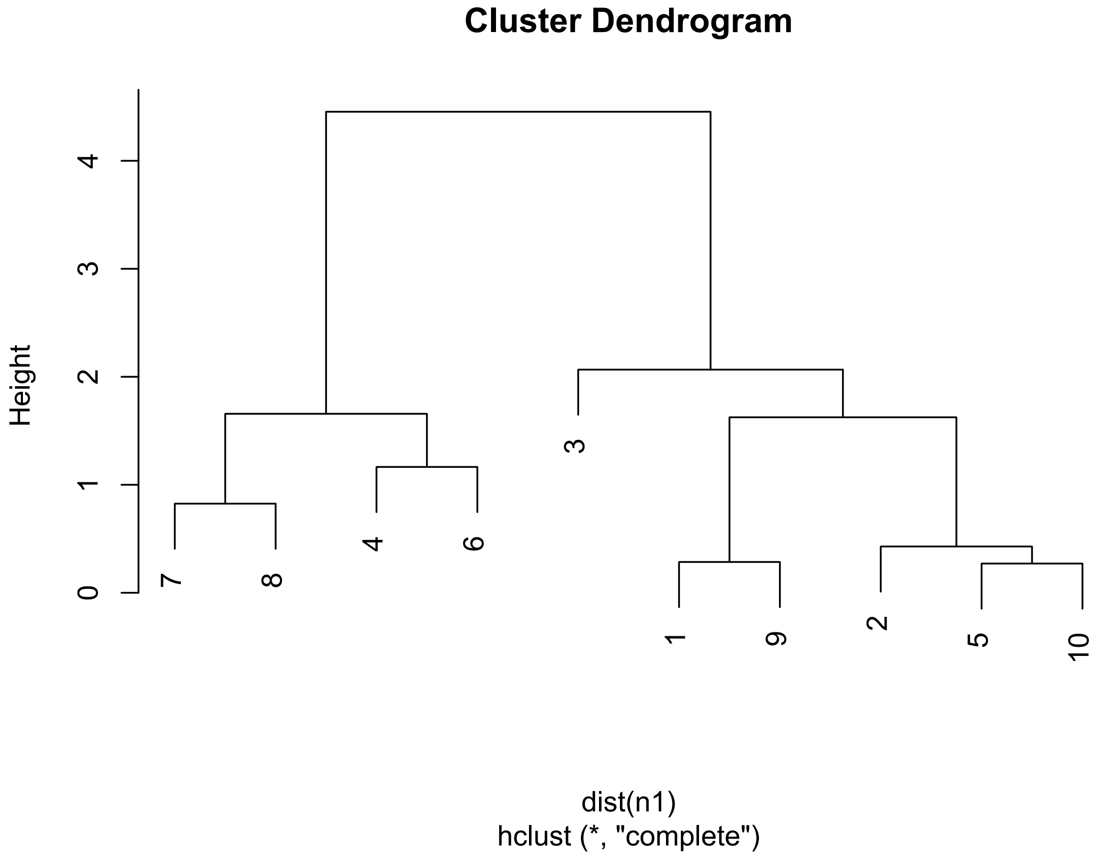
Dendogram

Vertical distance is proportional to cluster differences

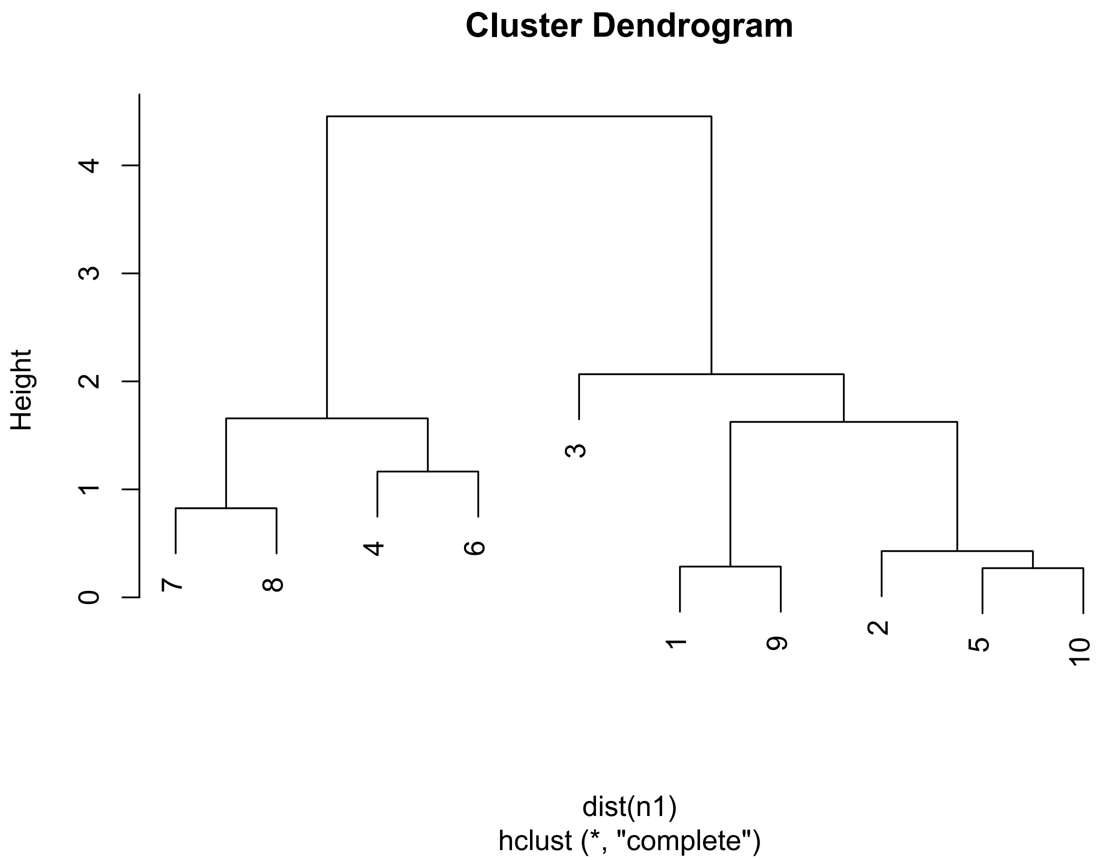
- Horizontal axis does not provide information about similarity

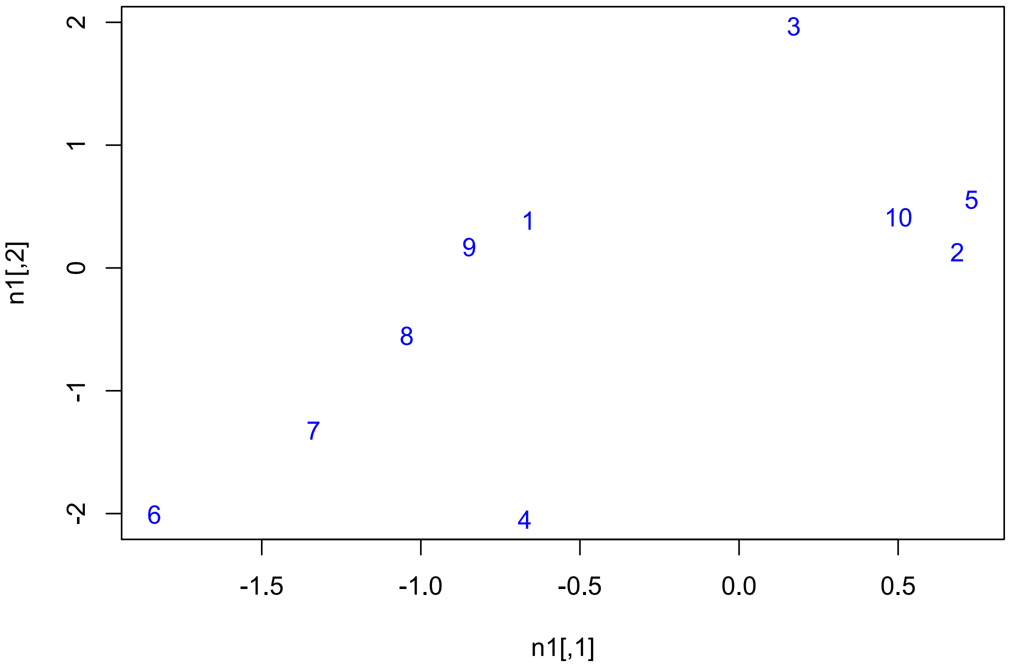
Draw a horizontal line to determine the number of clusters

- Eye-ball it
- No clear criterion

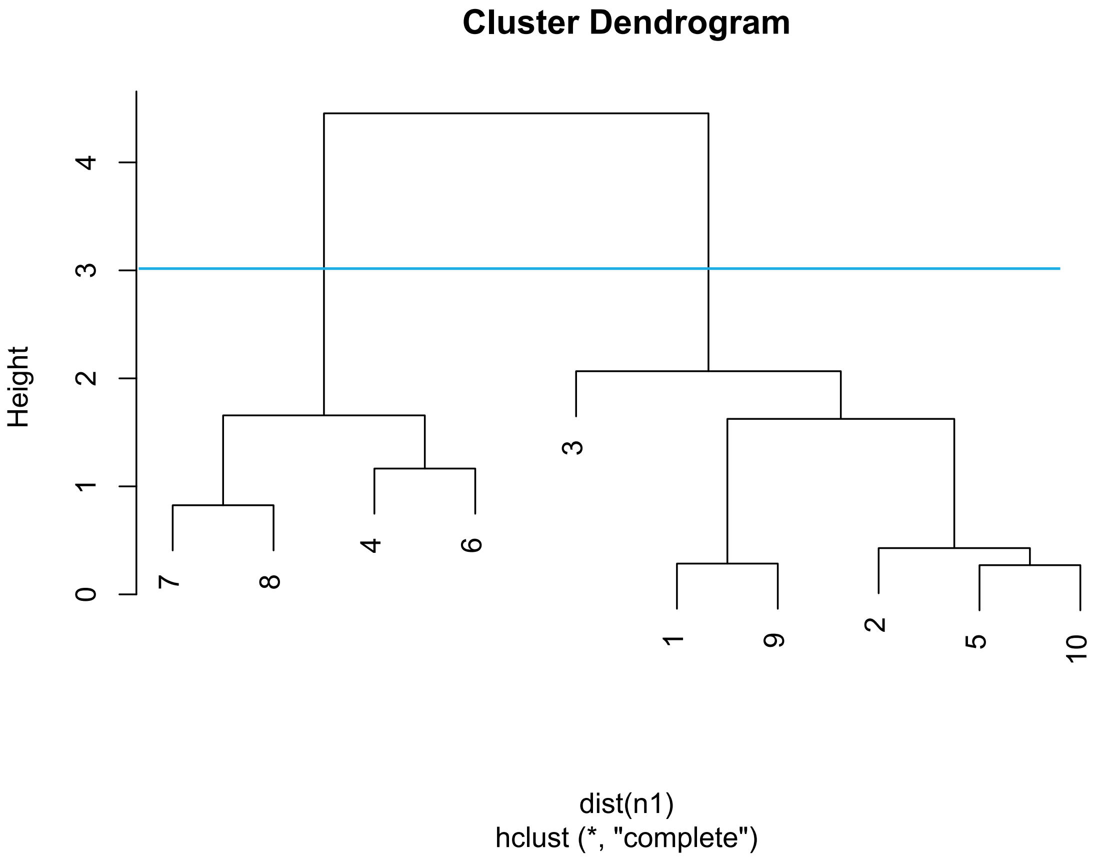


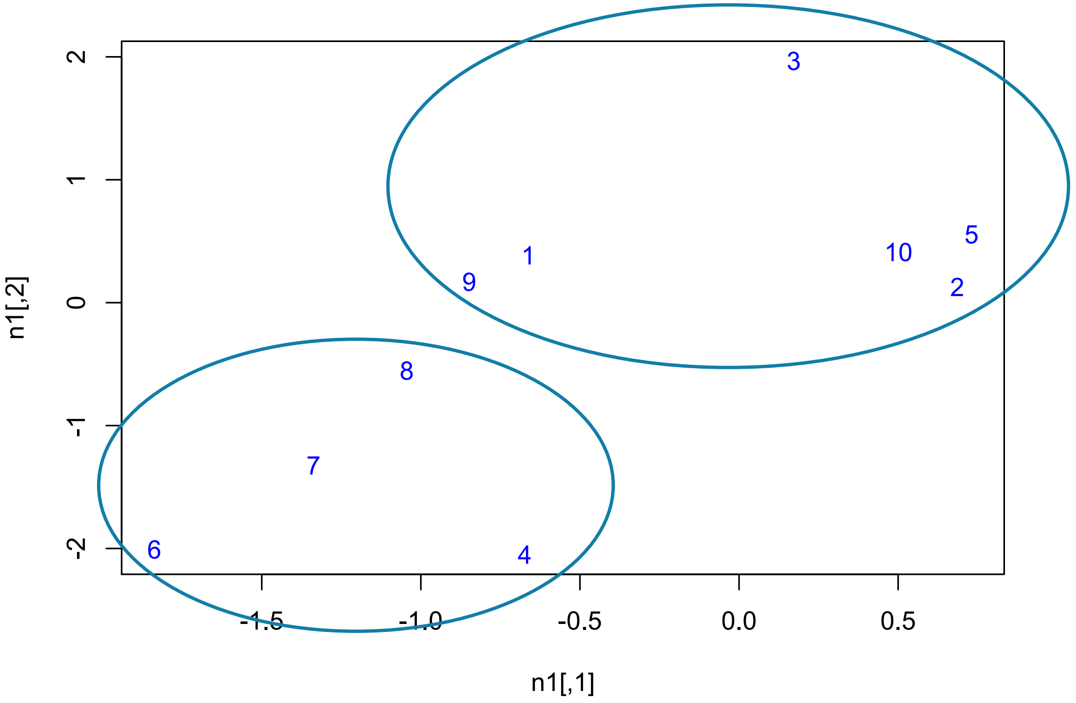
Dendogram and Bivariate Plot



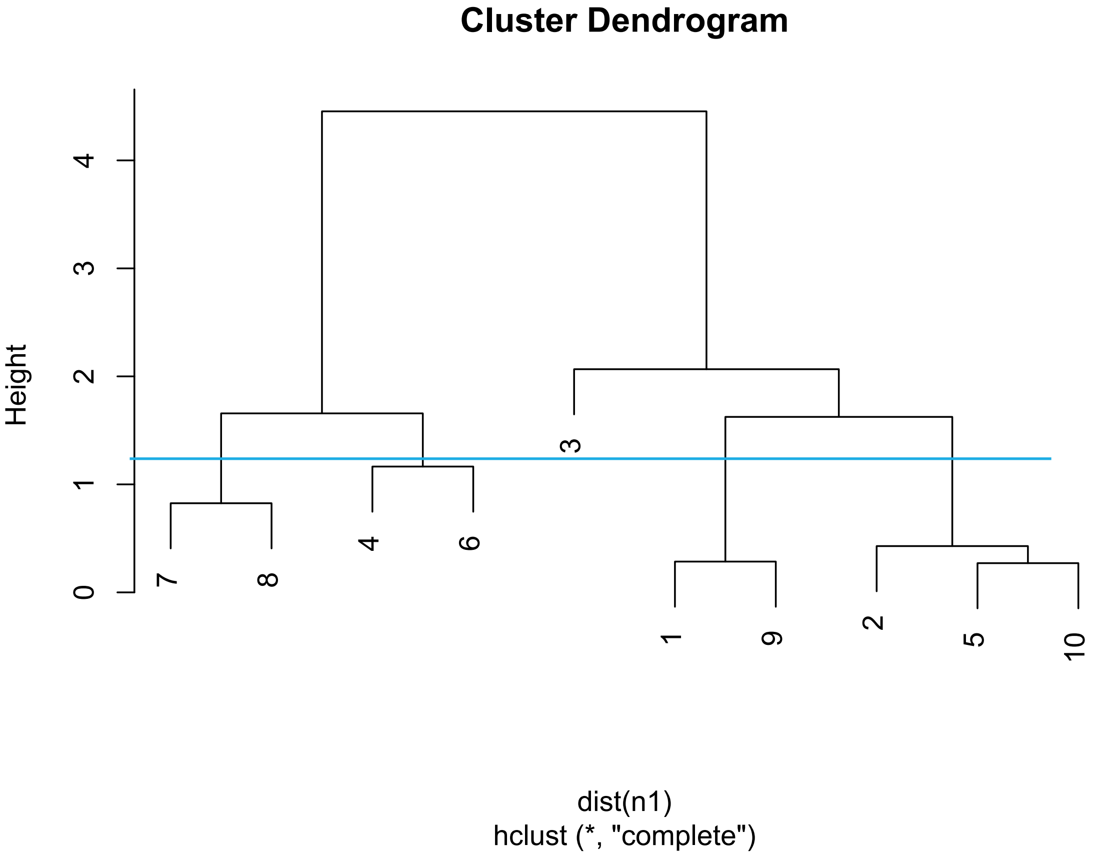


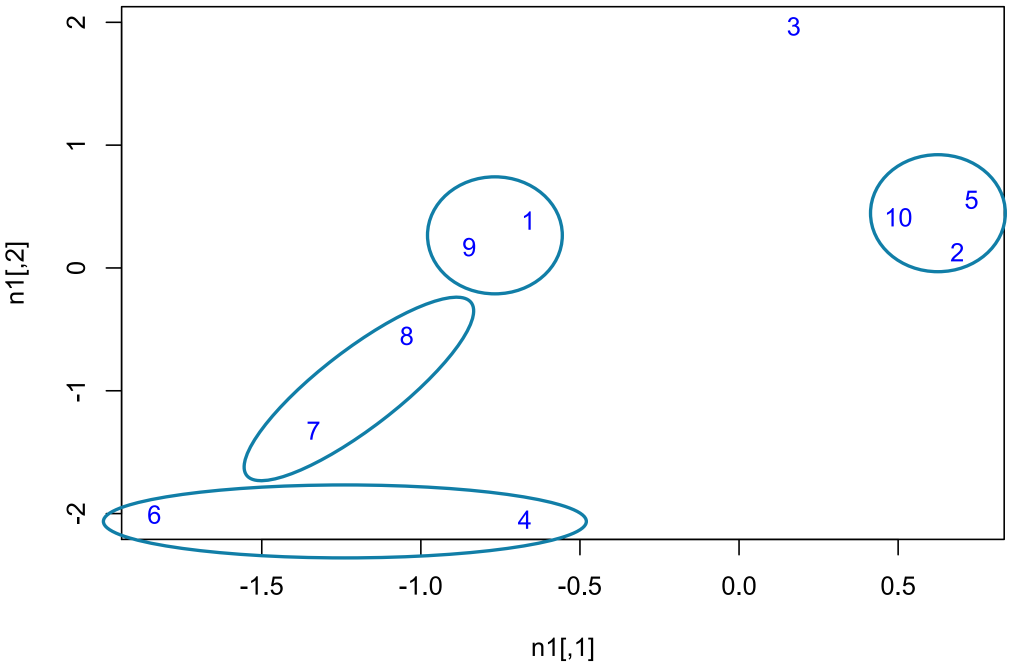
Dendogram and Two Clusters





Dendogram and Four Clusters (kinda)





Two methods

Hierarchical clustering

- We do not know in advance how many clusters we have
- Bottom-up approach
- Sequential approach

K-means clustering

- Assigning observations to a pre-determined number of clusters
- Top-down approach
- Iterative approach

K-mean Clustering

Goal: make within-cluster variation as small as possible

- How much members from the cluster differ from each other

Minimize the sum of the variation across clusters:

$$\sum_{k=1}^{K} W(C_k)$$

where W estimates the variability of each of the k clusters C_k

Estimating Cluster Variability

A common metric for the variability of the cluster $W(C_k)$ is the sum of the pairwise squared Euclidean distances in the cluster

- Not to be confused with RSS

In this case:

- x = observation
- i = individual
- j = predictor
- k = cluster
- $\#C_k$ = number of individuals in cluster

$$W(C_k) = \frac{1}{\#C_k} \sum_{i,i'} \sum_{j=1}^{P} \left(x_{ij} - x_{i'j}\right)^2$$

Algorithm

1. Randomly assign a number, from 1 to K, to each observation
    - Initial cluster assignment

Iterate until cluster assignments stop changing

- For each of the K clusters, compute cluster centroid (vector of means for each of the variables)
- Assign each observation to the cluster whose centroid is the closest (via Euclidian distance)
- Wash-rise-repeat

Run multiple times to prevent a local solution

Code Bits

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Let’s simulate data

Easier: we know the truth

```
 data.1 = data.frame(grp = 1,
x = rnorm(100, mean = -2, sd = 1),
y = rnorm(100, mean =  2, sd = 1))

 data.2 = data.frame(grp = 2,
x = rnorm(200, mean =  2, sd = 1),
y = rnorm(200, mean = -2, sd = 1))

 data.3 = data.frame(grp = 3,
x = rnorm(125, mean =  2, sd = 1),
y = rnorm(125, mean =  2, sd = 1))

 data = rbind(data.1, data.2, data.3)

 mydata = data[ ,c('x','y')]
```

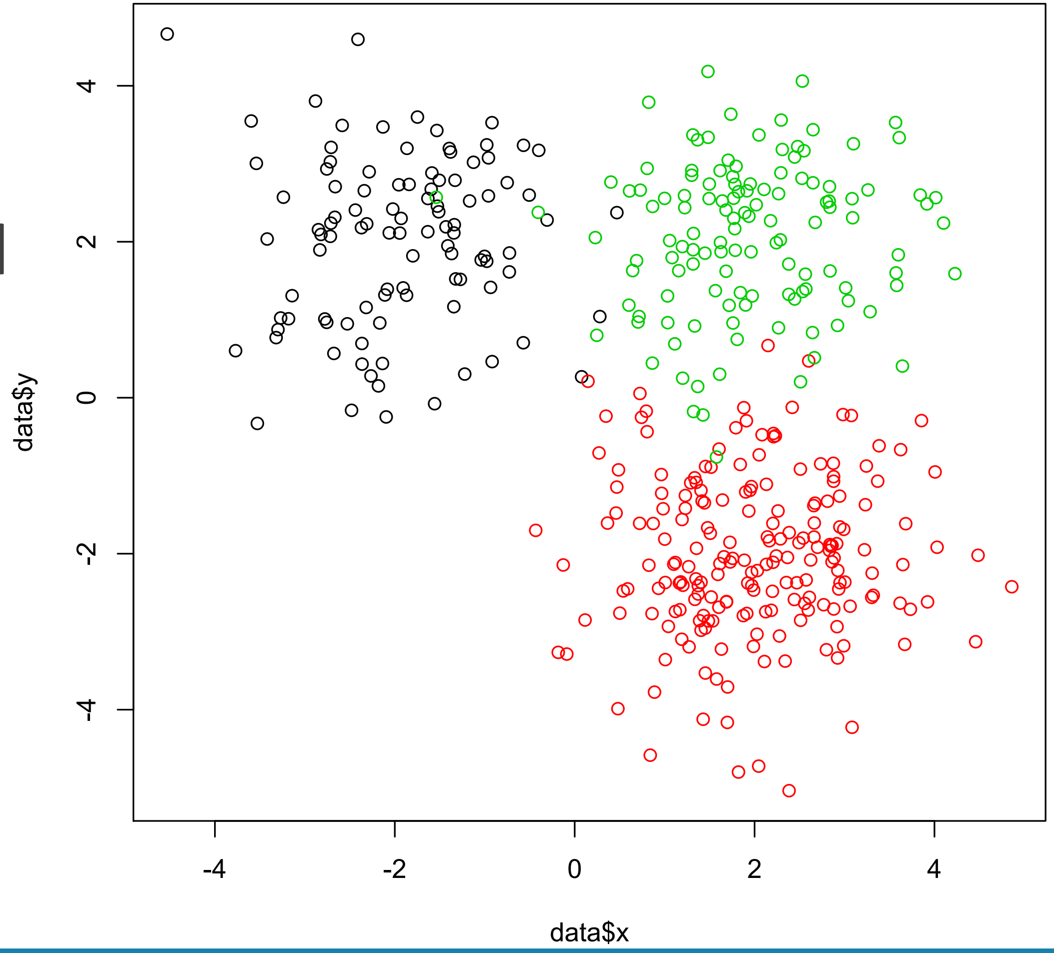
Simulated clusters

```
plot(data$x,data$y,
  col=data$grp)
```

Clusters seem generally well defined

Strategy

- Estimate clustering approach
- Cross-tabulate to estimate recovery



K-means

Estimate the within-cluster variability for solutions from 1 to 15 with `kmeans` from R `stat`

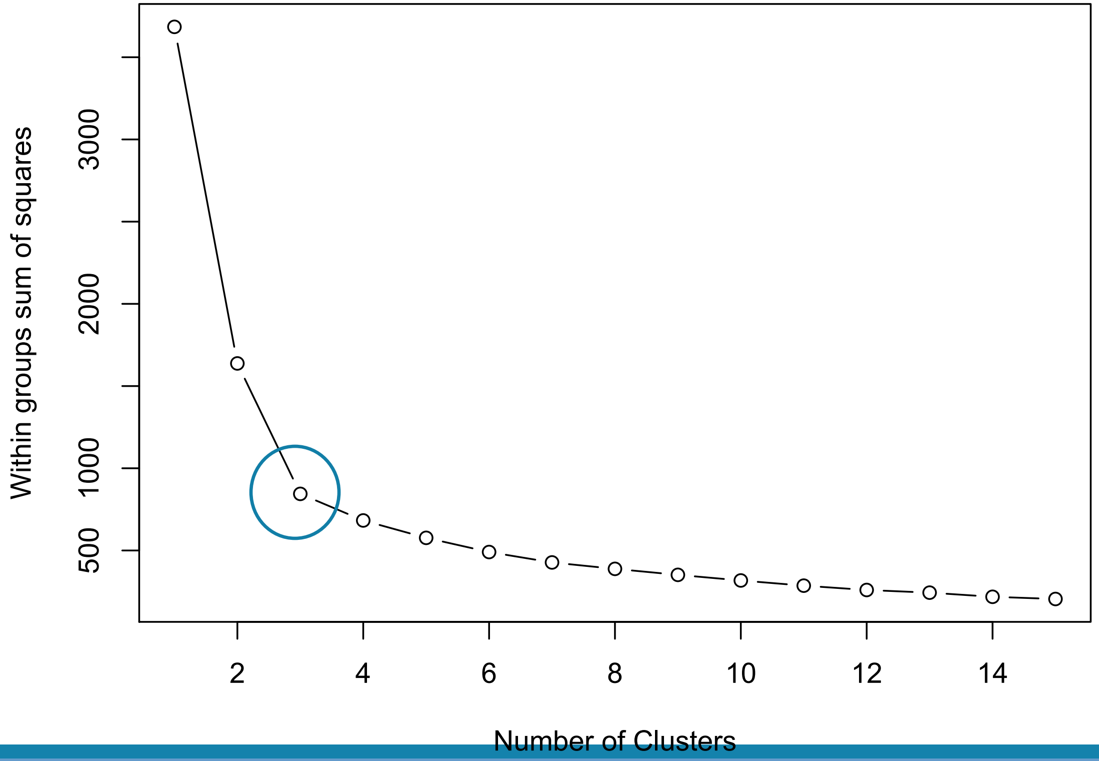
```
#one cluster (aka, no clusters)
wss <- sum(apply(mydata,2,var))/(nrow(mydata)-1)

#clusters 2 to 15
for (i in 2:15) wss[i] <- sum(
kmeans(mydata,centers=i, nstart=20)$withinss)
```

# clusters vs. within-cluster variability

```
 plot(1:15, wss, type="b", xlab="Number of
Clusters”, ylab="Within groups sum of
squares")
```

Recommended solution:

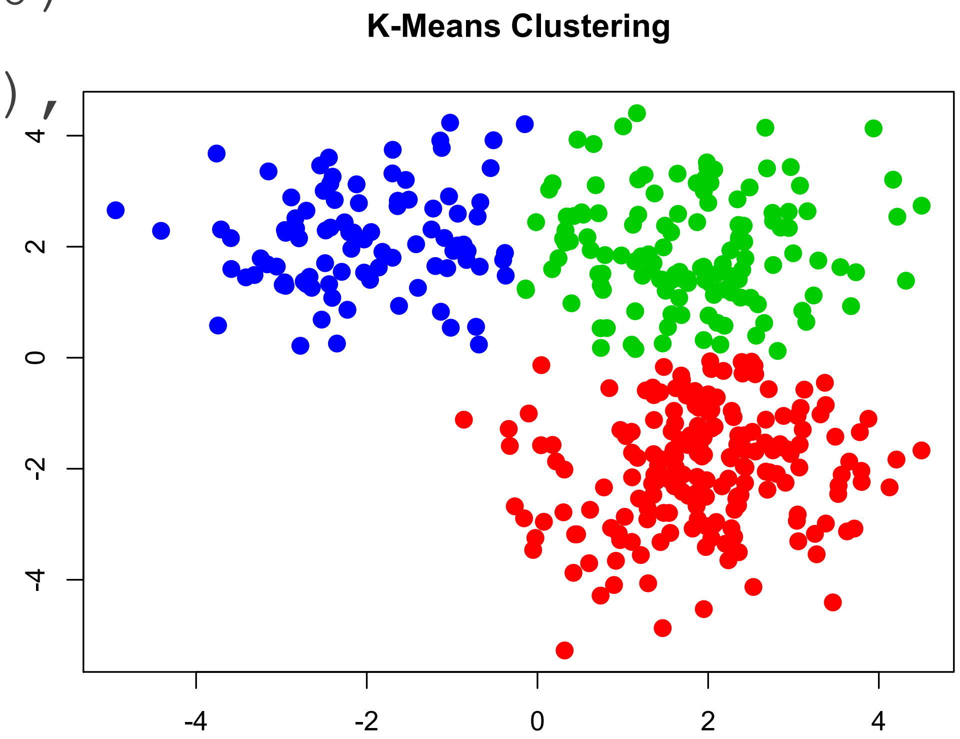
- The #clusters before the reduction in within-cluster variability starts to even out
- In this case, choose 3



K-means Solution

```
km.1 = kmeans(mydata, 3, nstart=20)

plot(mydata, col =(km.1$cluster+1),
main="K-Means Clustering",
xlab ="", ylab="", pch=20, cex=2)


table(data$grp, km.1$cluster)

     1   2   3

1    0   6  94

2  196   4   0

3    2 122   1
```

# Cluster number is just a label!



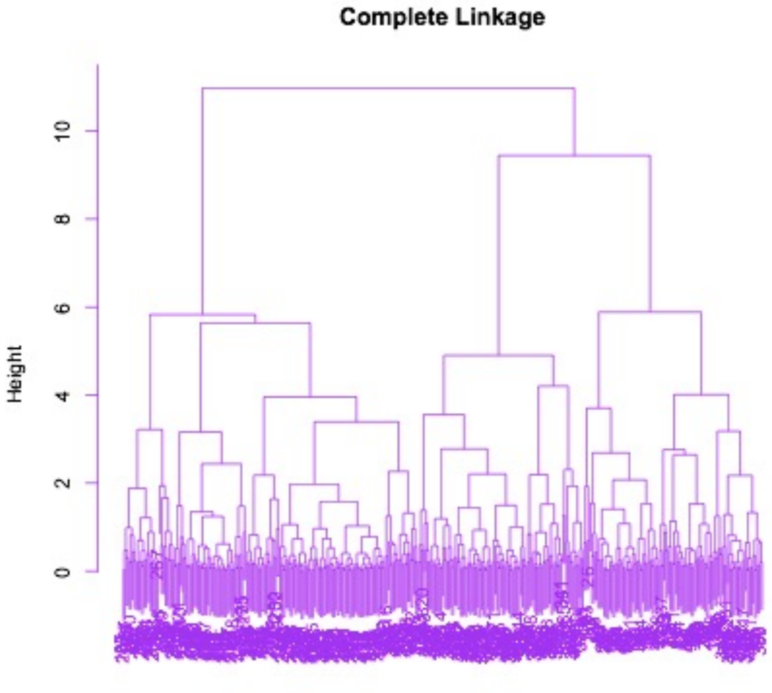
Hierarchical Clustering

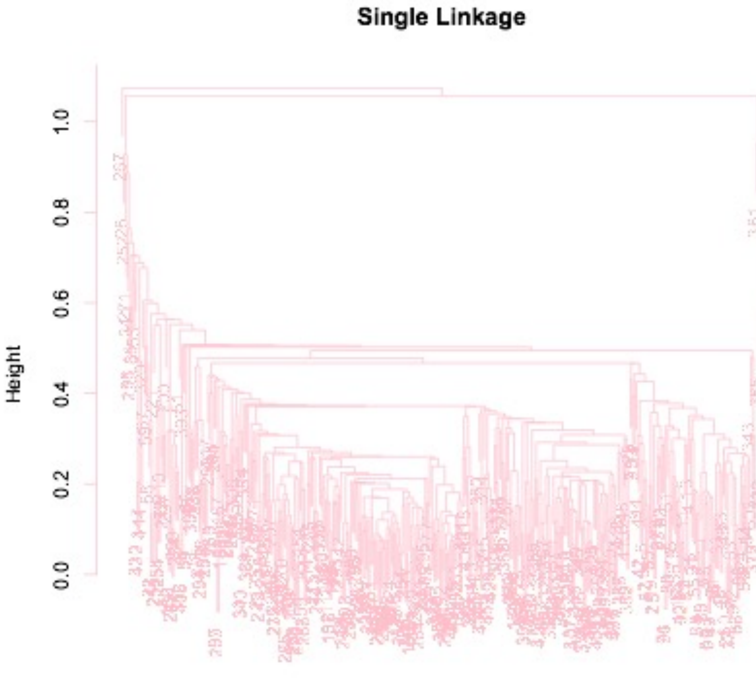
Estimate using the `hclust` function from R `stat`

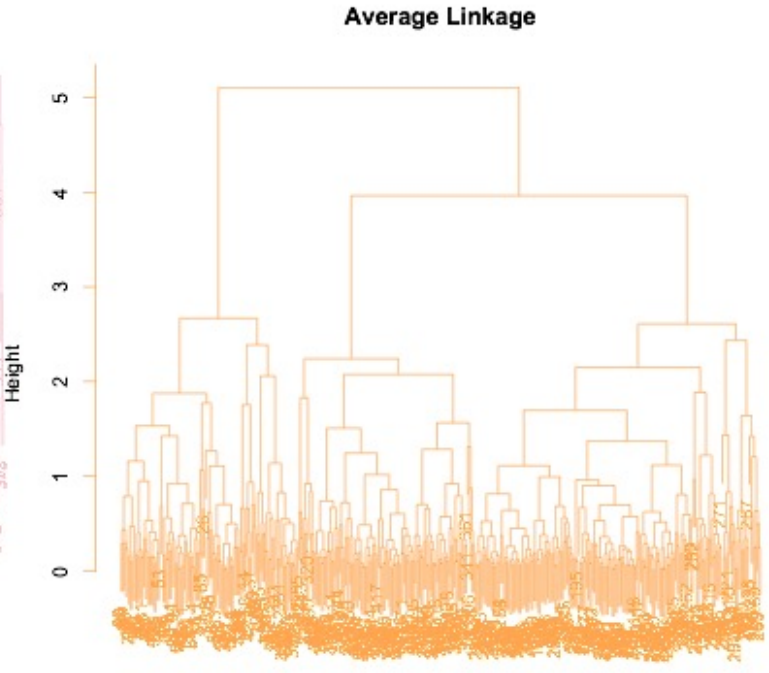
- `dist` function estimates the similarity – by default, Euclidean

```
hc.clust.1 = hclust(dist(mydata), method='complete')

hc.clust.2 = hclust(dist(mydata), method='single')

hc.clust.3 = hclust(dist(mydata), method='average')
```

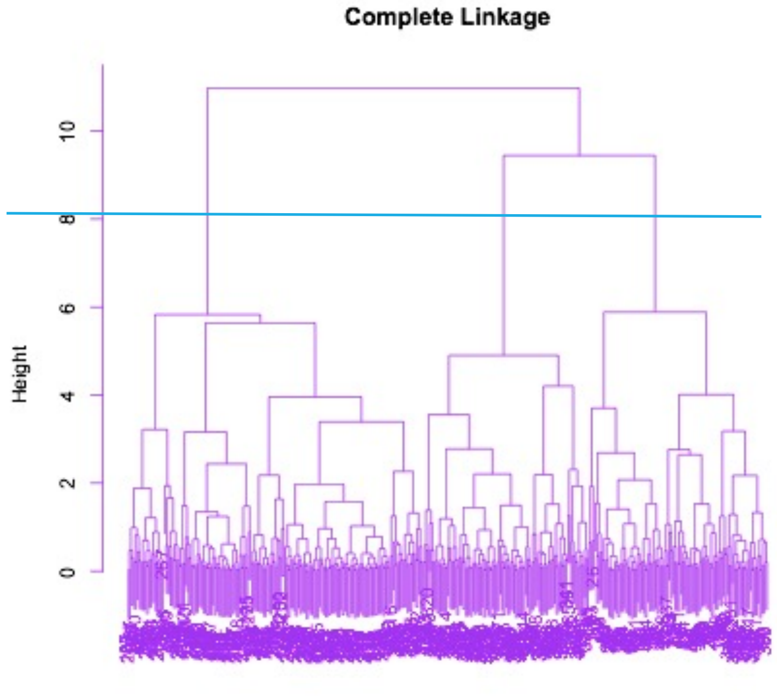
Plots

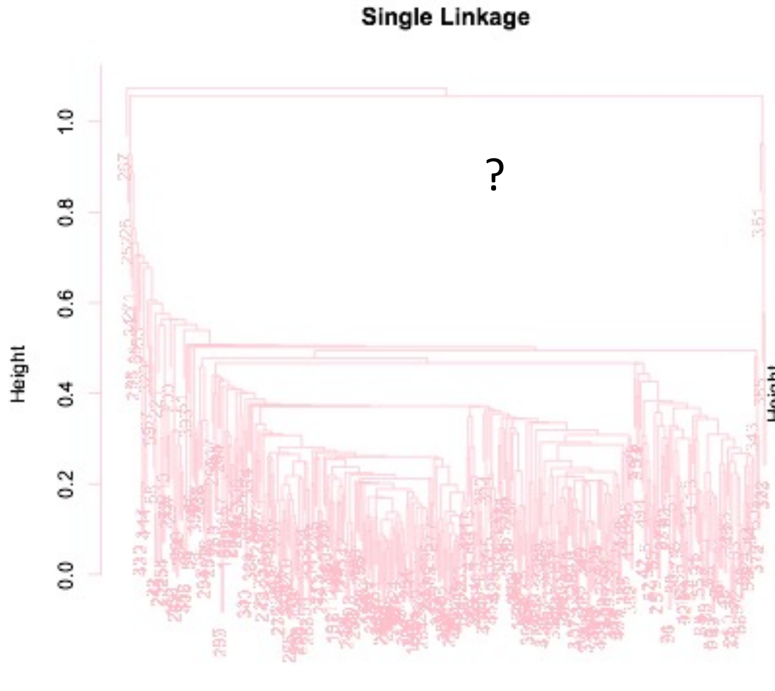

Complete Linkage

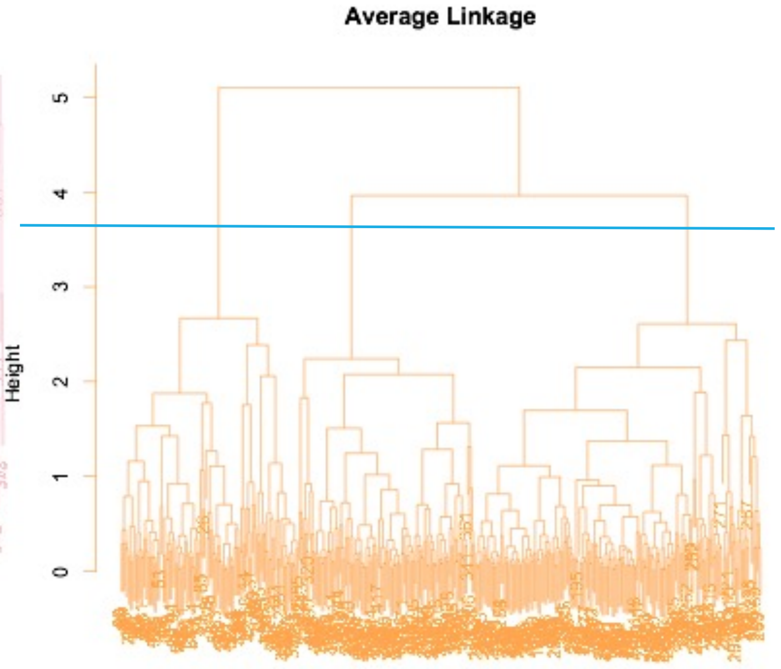

Single Linkage


Average Linkage

Plots


Complete Linkage
Height
0
2
4
6
8
10


Single Linkage
?
Height
0.0
0.2
0.4
0.6
0.8
1.0


Average Linkage
Height
0
1
2
3
4
5

Evaluated Solutions

#Complete linkage

```
pred.1 =
cutree(hc.clust.1,3)
table(pred.1, data$grp)
pred.1   1   2   3
     1  99   0  20
     2   1   8 101
     3   0 192   4
```

#Single linkage

```
pred.2 =
cutree(hc.clust.2,3)
table(pred.2, data$grp)
pred.2   1   2   3
     1 100 199 121
     2   0   1   0
     3   0   0   4
```

#Average linkage

```
pred.3 =
cutree(hc.clust.3,3)
table(pred.3, data$grp)
pred.3   1   2   3
     1  99   0  19
     2   1  10 106
     3   0 190   0
```

Complete and average linkage appear very similar and recover the true clusters

Single linkage assigned everybody to one cluster

Clustering Notes

OSCAR GONZALEZ, PHD

MACHINE LEARNING

K-means vs Hierarchical

Hierarchical

- (dis)advantage: provides multiple solutions
- Hierarchy to the clusters
- Flexible in terms of shapes

K-means

- Single solution per pre-specified k
- Clusters are more even/spherical

Both: fast algorithms

Cousins of K-means

K-median clustering (R package `Gmedian`)

- Use the median to cluster around, instead of the mean
- Minimize the absolute deviation from the median

K-medoids clustering (R package `pam`)

- Use the most centrally located point to cluster around, instead of the mean
- Medoid: point that has the minimum distance to the other points
- Motivation: the mean of all the variables (centroid) might not be a point in the dataset
- Minimize the sum of pairwise dissimilarities

Clustering Solutions

Subjective and deterministic

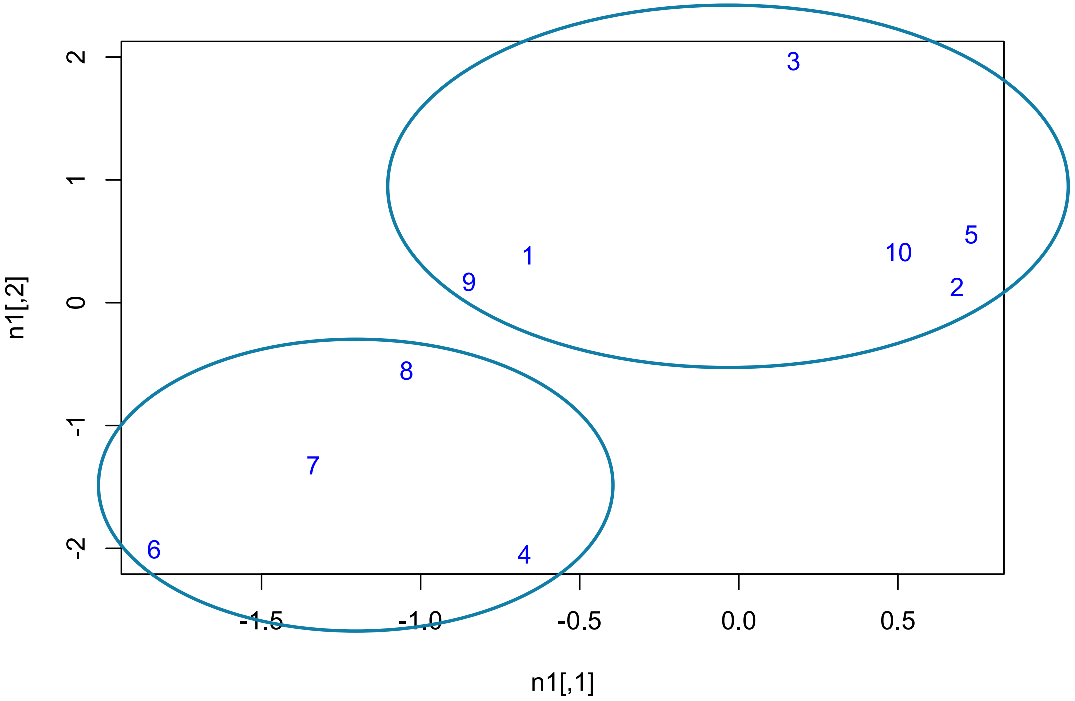
- What if one case does not belong to a cluster?
- Algorithms might force it into one

# of clusters depend on our decisions

- Linkage, scale of predictors, metric…

Interpret with caution!

- For the hierarchical toy example, I plotted randomly-generated data
- Dendogram suggests two clusters…illusion of clusters



Dimension Reduction

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Motivations

Visualizing many predictors is difficult

- We could do one-way or two-way associations, but difficult in 3D or higher

Some predictors may be redundant or not all might be important

There might be estimation problems when we have too many predictors

- Select some predictors via feature selection
- Combine predictors via dimension reduction

Dimension Reduction

Map the higher-dimensional data to a lower-dimension while not losing much information

- In this case, we would like to keep the variance of the variables
- Use these lower dimensions to predict/plot etc.

Principal components analysis (PCA)

- Best linear combination of variables
- Not all variables are needed
- Orthogonal components drawn sequentially

Example

Suppose that you have 10 predictors and we need to find a way to describe them…

- We cannot really plot them all – we need a plot in 10 dimensions
- Summary information?
- Bivariate plots?

Examining predictors one or two at the time only represent a fraction of the total information in the data

Example

Perhaps not all of the dimensions are *important* or *interesting*

- If we know where you stand on 9 of the predictors, we'd have a pretty good guess of where you stand on the 10th
- So, 10th dimension is not *needed*

If we could map the dimensions to a different space, perhaps we could keep the dimensions that contain most of the information

Principal Components

Let’s make the problem simpler – just two predictors, x1 and x2

- we can plot this, but this approach generalizes to more dimensions

We can make a linear combination of the predictors

$$z_1 = \phi_1 x_1 + \phi_2 x_2$$

z1 is the principal component and ϕ_j are the weights that control the contribution of each predictor

- … awfully similar to a regression equation…
- NOT regression! We do not have an outcome or an error term

Best component

Objective: find the weights to the predictors that will give the components the most variance

Conceptually:

If you are measured in 2 variables, you can differ in two things

- Variability in x1 and variability in x2

If we standardize the variables, each the predictor will have a variance of 1

- Total variance in the dataset = 2, because of 2 predictors

By combining (summing) predictors, you combine variance

Best component

Objective: find the weights to the predictors that will give the components the most variance

For the following component

$$z_1 = \phi_{11}x_1 + \phi_{12}x_2 + \cdots + \phi_{110}x_{10}$$

$$subject\ to: \sum_{j=1}^{p} \phi_{1p}^2 = 1$$

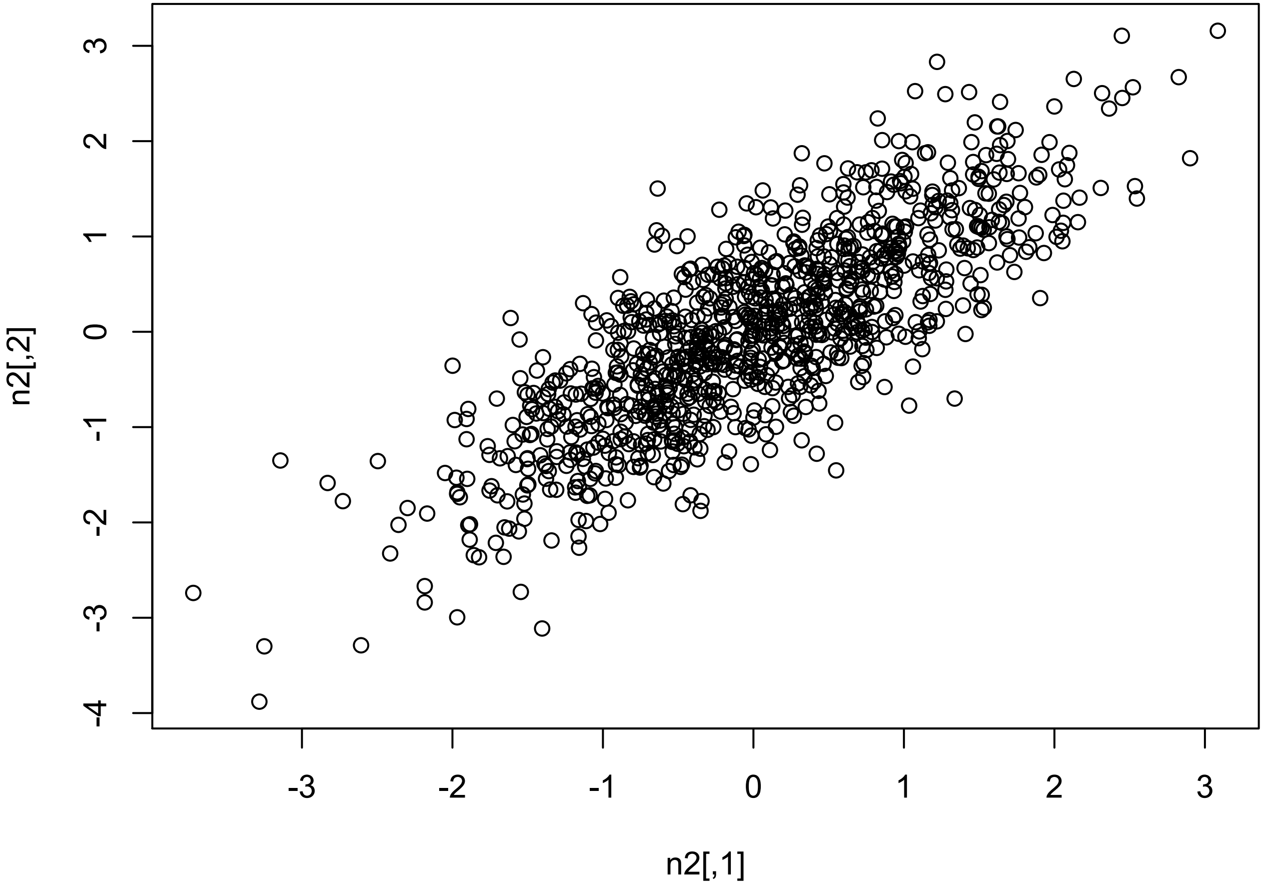
Estimated via eigendecomposition

- Eigenvector: axes of the projected space
- Eigenvalue: variance of the component

Bivariate Plot

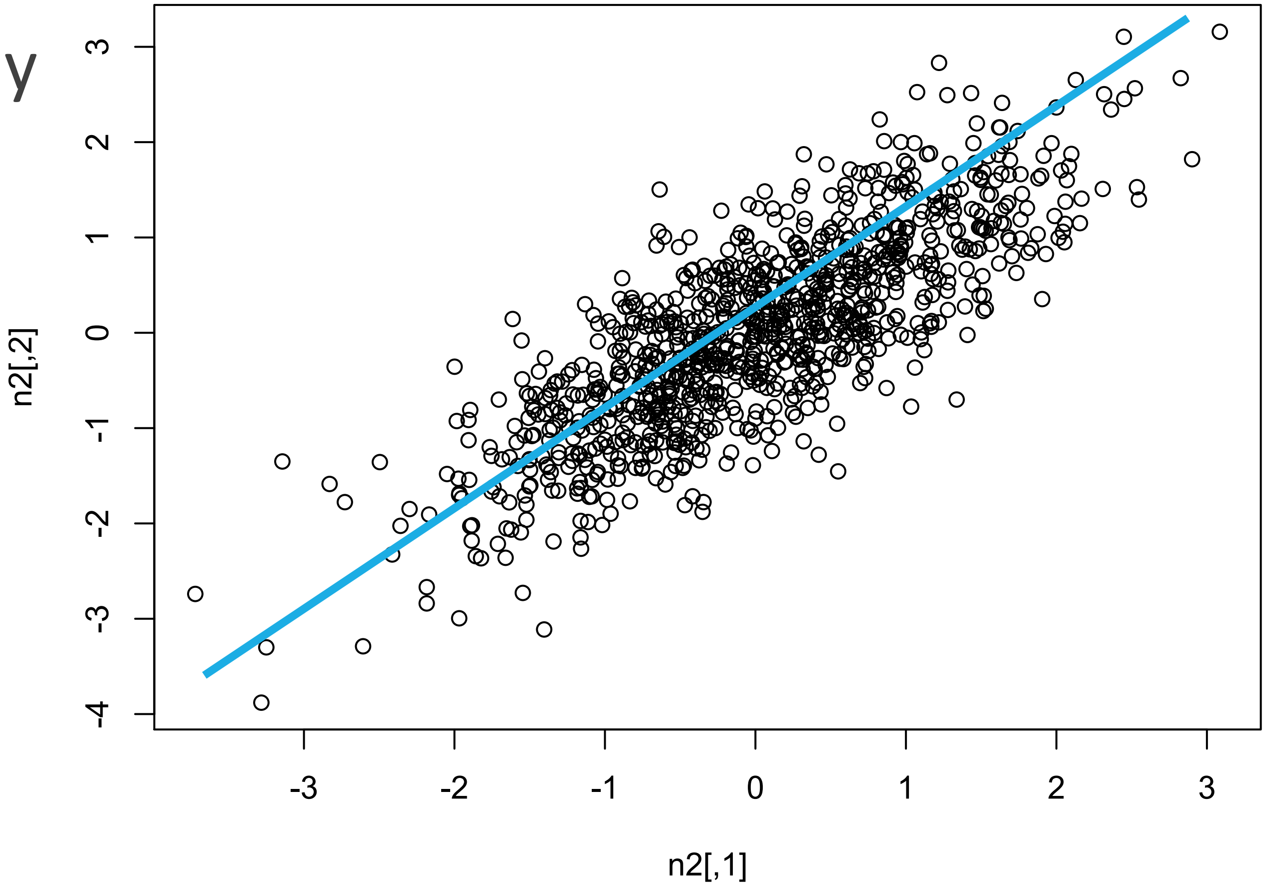
Let’s graphically represent the principal components

- Introduction to a new set of axes
- PCA – axes closest to the data points



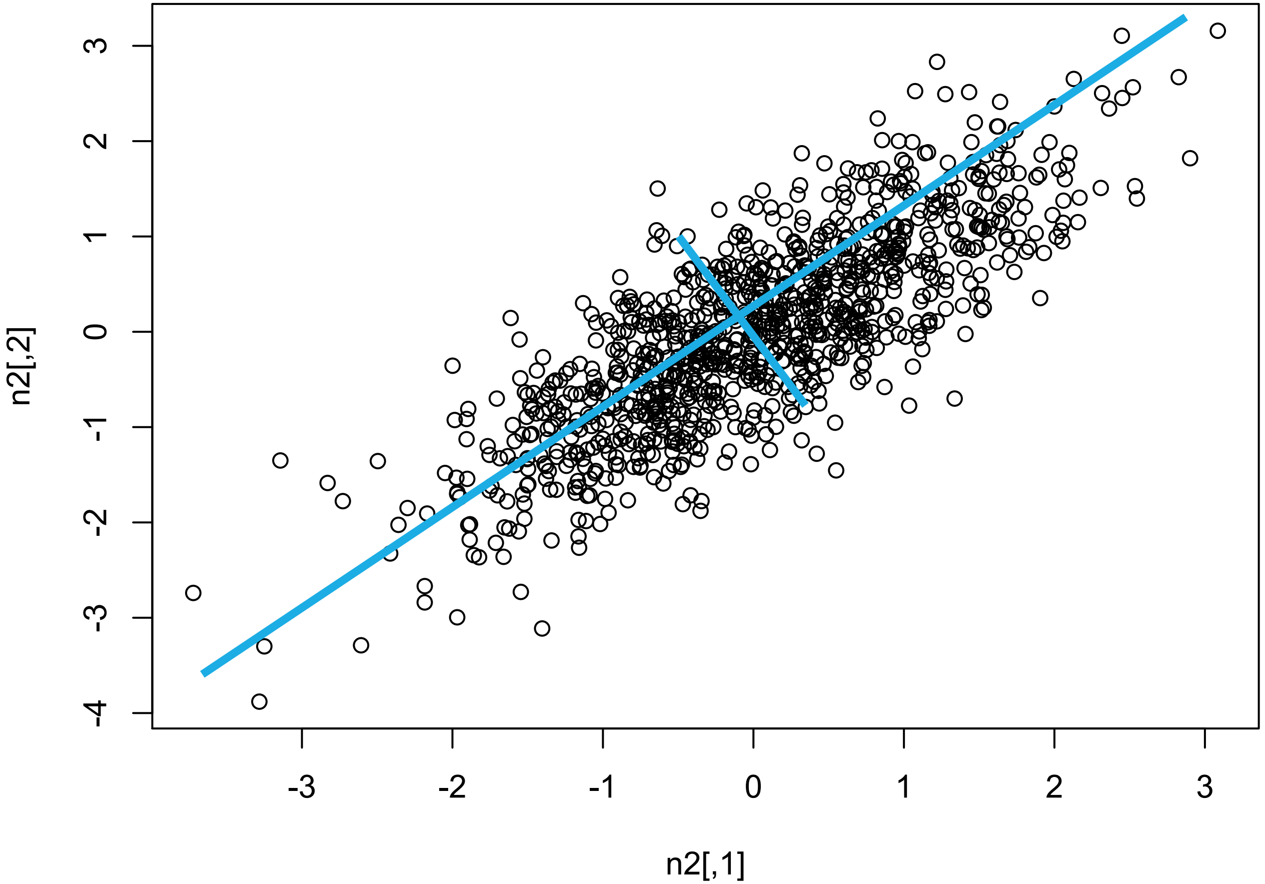
First Principal Component

Observations vary roughly along this line



A Second Principal Component

Observations vary roughly along this other line…

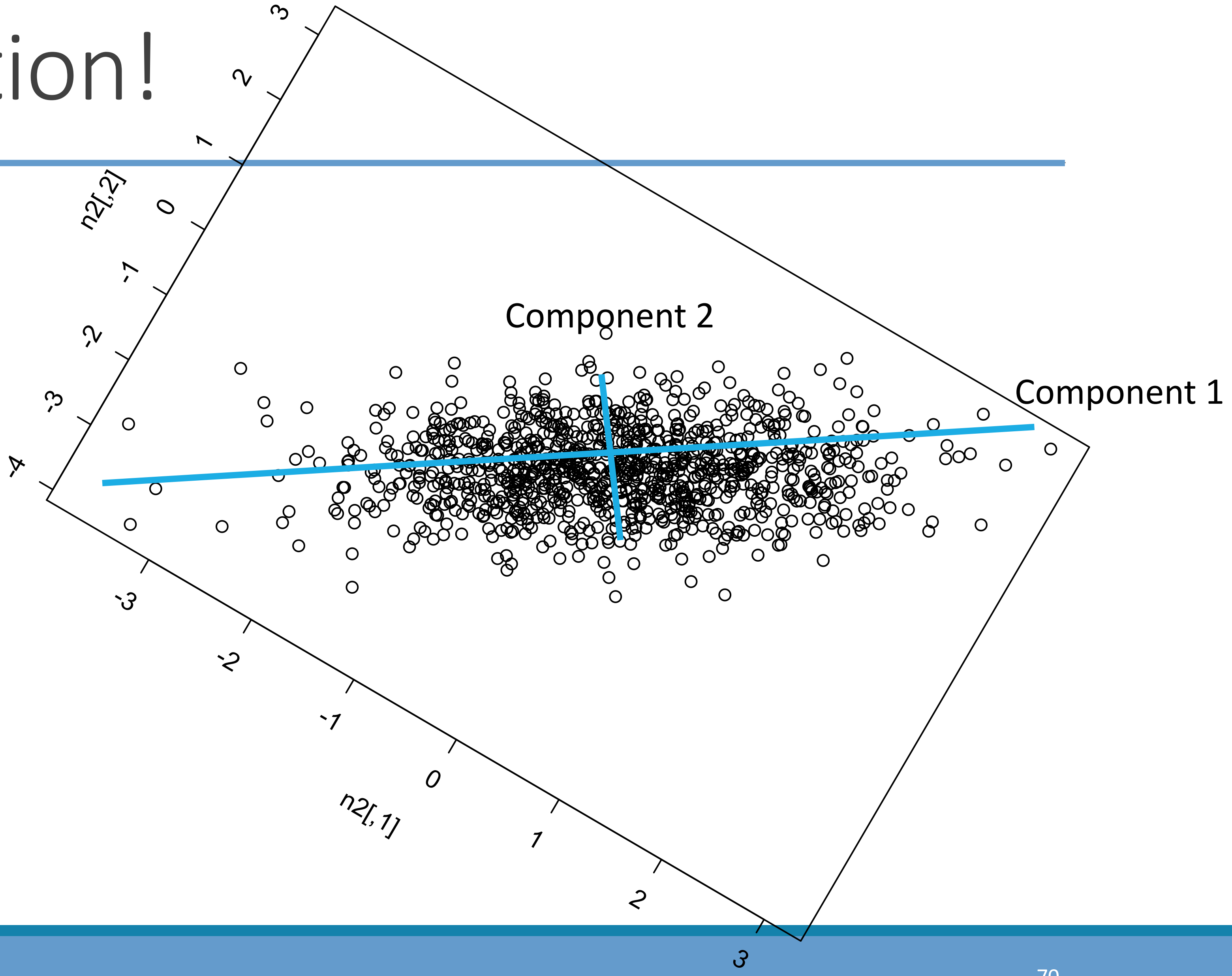


A Mental Rotation!

I can describe your data point if I know your value on component 1 and component 2

I'd argue that component 1 is most important because it covers a greater span

- You vary more in component 1
- Gives me a good idea where you're located



Principal Components

Goal: help describe high-dimensional data with less dimensions

There are as many components as there are dimensions

- 10 predictors = 10 dimensions = 10 units of variance = 10 components
- Well… components go up to total *p* or *n - 1*

Components redistribute the variance, but they still must add to the total variance of the predictors

- Components with more variance are more important
- We might only need a handful of components to describe you well, instead of 10 dimensions – so get rid of components that are not important

How much information did we lose?

Proportion of variance explained

- Again, the predictors have a total amount of variance
- Sum variance of the remaining components and take proportion from total

Similar information is used to decide which components to keep

- Keep as many components to account for a sizable amount of the variance
- Scree plots are helpful – at what point the cumulative variance explained levels-off
- Largely subjective …
- … but what about prediction accuracy? – principal components regression

Code Bits

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Arrest Dataset

The `USArrests` dataset is from R base

Four variables across the 50 US states

- Murder – arrests per 100K
- Assault – arrests per 100K
- UrbanPop - % of urban population
- Rape – arrests per 100K

Note, predictors are on different scales, so we will need to standardize them

- `scale` function

PCA

We will carry out the analysis using the `prcomp` function from R `stat`

```
out=prcomp(USArrests,scale=TRUE) #four components
out$rotation #get weights
```

```
             PC1    PC2    PC3    PC4
Murder   -0.536  0.418 -0.341  0.649
Assault  -0.583  0.188 -0.268 -0.743
UrbanPop -0.278 -0.873 -0.378  0.134
Rape     -0.543 -0.167  0.818  0.089
```

Component scores

We can easily obtain component scores

```
round(head(out$x),3)
```

```
              PC1    PC2    PC3    PC4
Alabama    -0.976  1.122 -0.440  0.155
Alaska     -1.931  1.062  2.020 -0.434
Arizona    -1.745 -0.738  0.054 -0.826
Arkansas    0.140  1.109  0.113 -0.181
California -2.499 -1.527  0.593 -0.339
Colorado   -1.499 -0.978  1.084  0.001
```

As expected, the mean is zero, and the variance is max'ed at PC1

```
round(apply(out$x,2,var),3)
  PC1   PC2   PC3   PC4
2.480 0.990 0.357 0.173
```

```
round(colMeans(out$x),3)
PC1 PC2 PC3 PC4
  0   0   0   0
```

Percentage of Variance Explained

```
var1=apply(out$x,2,var)
```

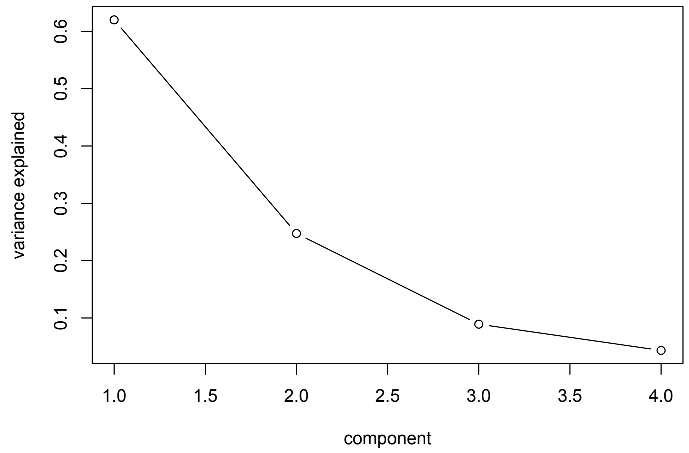
```
a1=var1/sum(var1)
```

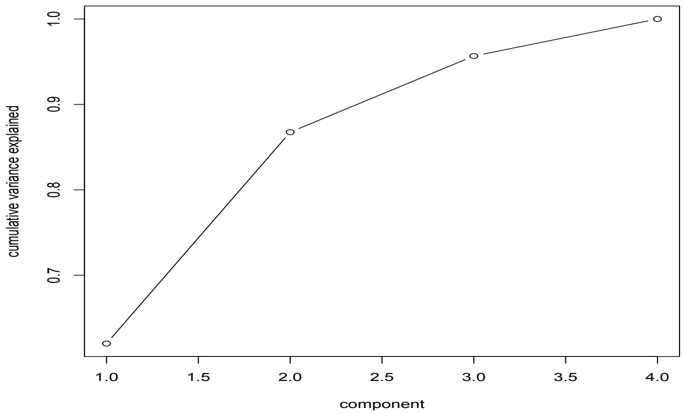
```
a1
```

```
  PC1   PC2   PC3   PC4
0.620 0.247 0.089 0.043
```

```
plot(a1,type='b',
xlab='component',
ylab='variance explained')
```

```
plot(cumsum(a1),type='b',
xlab='component',
ylab='cumulative variance explained')
```





The eigen-thingy

The goal of PCA is to break the matrix into its natural components

$A = v\lambda v^{-1}$, where A is the correlation matrix of the data, v is the eigenvector, and λ is the eigenvalues

- In practice - v is the matrix of loadings of the variables on the components
- λ is the variance estimate of the component on a diagonal matrix

So, if we pre- and post-multiply the diagonal matrix of variance estimates by the eigenvectors, we get back the correlation matrix

- *Eigendecomposition*

Break-down

$$\lambda = \begin{bmatrix} 2.480 & 0 & 0 & 0 \\ 0 & .990 & 0 & 0 \\ 0 & 0 & .357 & 0 \\ 0 & 0 & 0 & .173 \end{bmatrix}$$

$$v = \begin{bmatrix} -.535 & .418 & -.341 & .649 \\ -.583 & .188 & -.268 & -.743 \\ -.278 & -.872 & -.378 & .134 \\ -.543 & -.167 & .818 & .089 \end{bmatrix}$$

$$v^{-1} = inverse\ of\ v = \begin{bmatrix} -.535 & -.583 & -.278 & -.543 \\ .418 & .188 & -.872 & -.167 \\ -.341 & -.268 & -.378 & .816 \\ .649 & -.743 & .134 & .089 \end{bmatrix}$$

```
round(cor(USArrests),3)
           Murder Assault UrbanPop  Rape
Murder      1.000   0.802    0.070 0.564
Assault     0.802   1.000    0.259 0.665
UrbanPop    0.070   0.259    1.000 0.411
Rape        0.564   0.665    0.411 1.000
```

Matrix is symmetric, which helps a lot

λ would have real numbers and v is orthogonal

$$A = v\lambda v^{-1}$$

General Notes

OSCAR GONZALEZ, PHD

MACHINE LEARNING

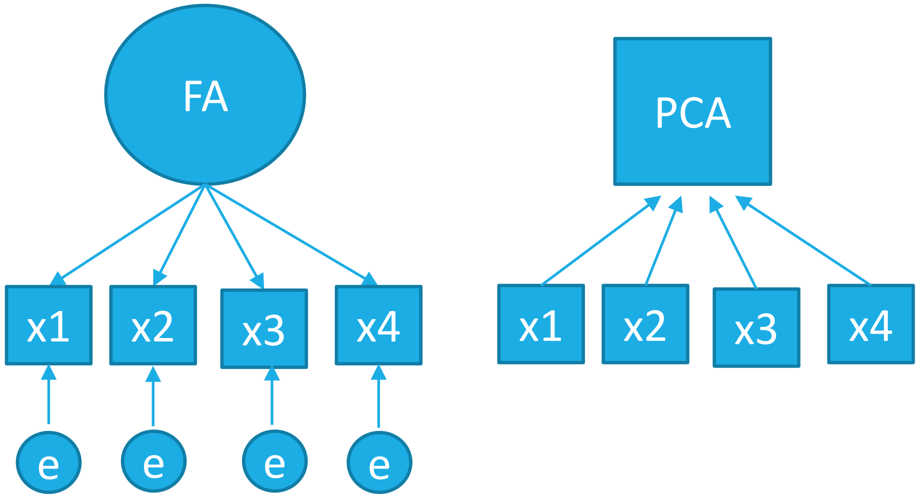
Differences with Factor Analysis

In psychology, we might be more familiar with factor analysis

- Underlies a lot of our assessments

The biggest difference is theoretical

- In PCA, the variables combine to determine a composite – deterministic relation
- In FA, the factor is what is driving the value of the variables, along with an error term



Consider…

- self-control; x1: set goals, x2: work hard

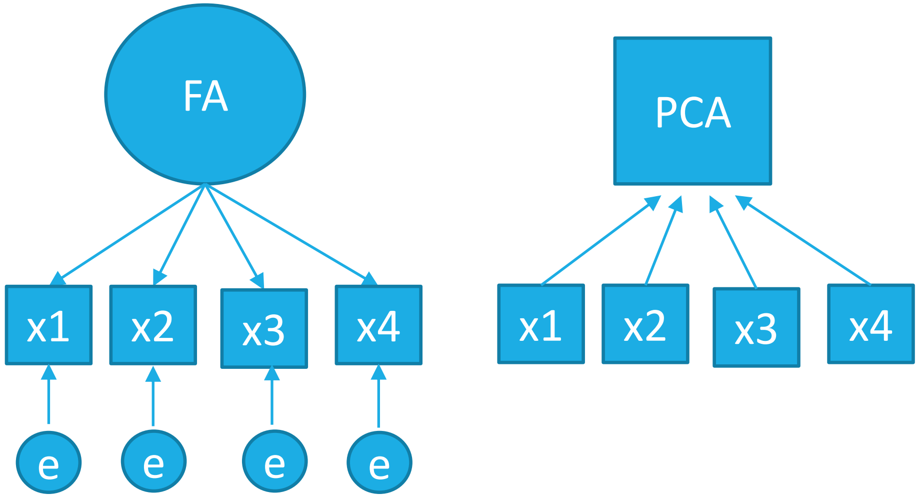
Differences with Factor Analysis

Goal

- PCA – combine variables to maximize their variance
- FA – reproduce the variance-covariance matrix of the variables

Estimation differences

- PCA decomposes the full variance-covariance matrix of the variables
- FA decomposes a "reduced covariance matrix" where the 1's in the diagonal are replaced by the squared multiple correlation (SMC)



*SMC – variance explained from a variable predicted from the rest
Ex. x1 ~ b1*x2+b2*x3+b3*x4+e1

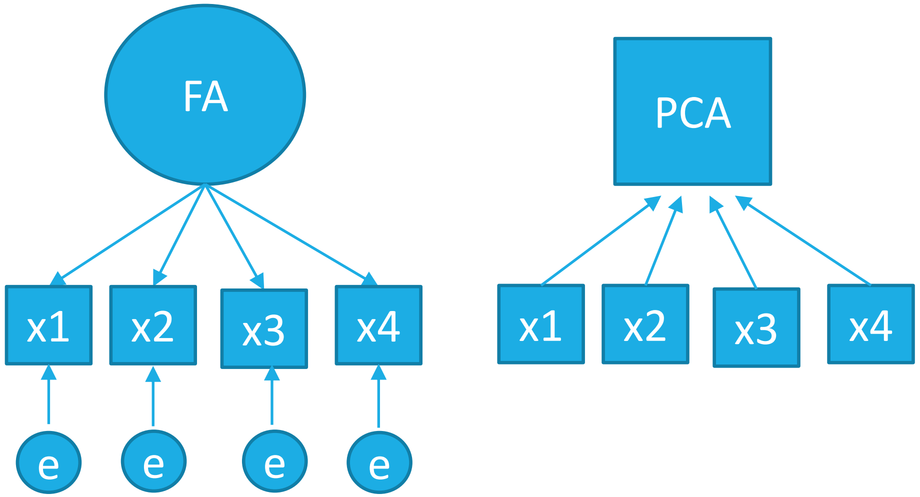
Differences with Factor Analysis

Interpretability

- In FA, factors are rotated (similar to what I showed with the axes) to interpret the solution – if variables are not related, then why bother?
- In PCA, the interpretation is a by-product of the estimates

When are FA and PCA similar?

- *All variance is common variance*
- As the # of variables increases…

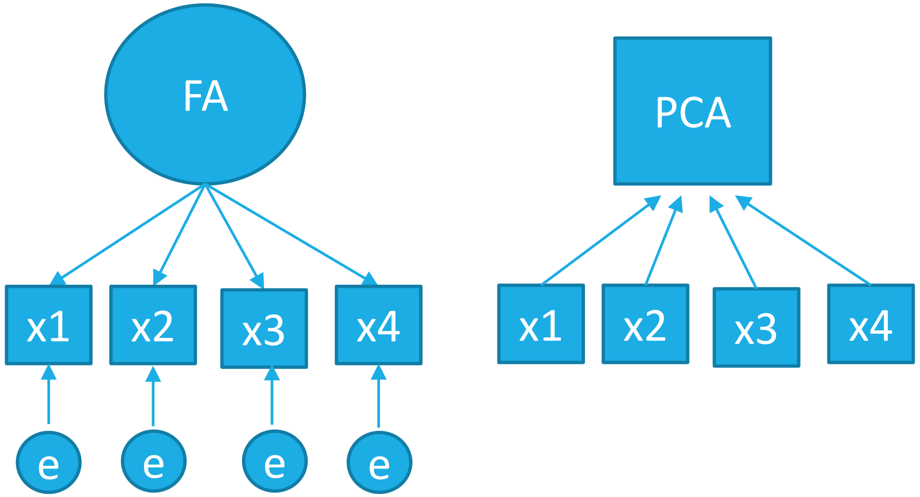


Murky distinctions

The distinction between FA and PCA has been murky

A generation of us grew up in a time when…

- The SPSS routine will tell you that it was doing factor analysis…
- In reality, it was doing PCA…
- So, what people would claim were factors were not really factors…
- Perhaps that why some quants think SPSS is deceiving



Caveats

Variables have to be standardized/transformed

- PCA maximizes variance
- If predictors have different variances, then PCA will extract predictors with most variance first…
- We’d like the solution to be independent of the metric of the predictors
- Once we settle on a metric, solution will be unique

PCA does not reference any outcome

- If we want to use them for prediction, throw into regression equation (PCR)
- If we would like extract components in reference to a criterion, perhaps we’d have to do PLS (Partial Least Squares)

Unsupervised Learning

We know a lot about supervised learning

- Models: lasso, logistic regression, trees…
- Evaluation: rmse, variance explained, sensitivity, cross-validation…

We know less about unsupervised learning

- Difficult to find the right model
- Difficult to evaluate

Still an important approach in the exploratory phase of data analysis

- This field continues to grow

Unsupervised Learning

Clustering

- Finding groups in the dataset that are similar to each other
- K-means and hierarchical clustering
- Interpret with caution

Dimension reduction

- Project to a lower-order dimension to plot/predict/store data
- Use principal components analysis to estimate components that maximize variance
- Different from factor analysis, which is most common in psych (…or is it?)

Wrap-up

OSCAR GONZALEZ, PHD

MACHINE LEARNING

PSYC834 – Machine Learning

Introduction to Machine Learning

- Learn patterns from datasets using data-driven algorithms

Classification problems (predicting binary outcomes)

- Algorithms: Trees, random forest, logistic regression, knn, naïve Bayes…

Regression problems (predicting continuous outcomes)

- Algorithms: Regression, trees, random forest, lasso, forward selection…

Unsupervised learning

Ethics

Revisiting the First Lecture

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Main Take-Aways

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Definitions

Machine Learning: *a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.*

(SAS website)

A thing (machine) *learns* when it changes a previous behavior in a way that it performs better in the future.

- Performance-based definition

Machine Learning Methods

Supervised Learning: Learn the mapping between the inputs and the output

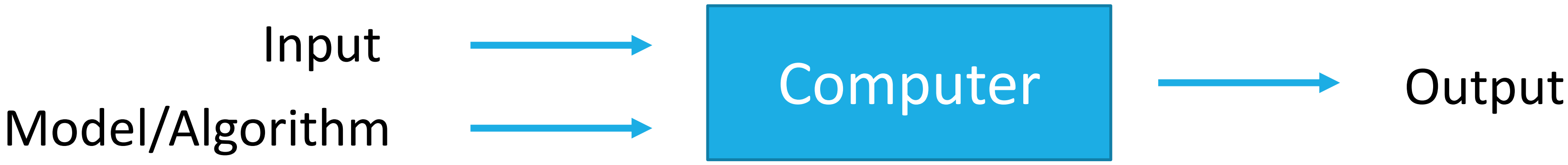
- Focus on prediction; learning by example
- Develop model in the training dataset
- Evaluate model in the testing dataset

Unsupervised Learning: Discover patterns in the dataset

- Finding clusters or reducing dimensions of the data
- Largely descriptive and difficult to evaluate

Rough Idea

Traditional Analyses



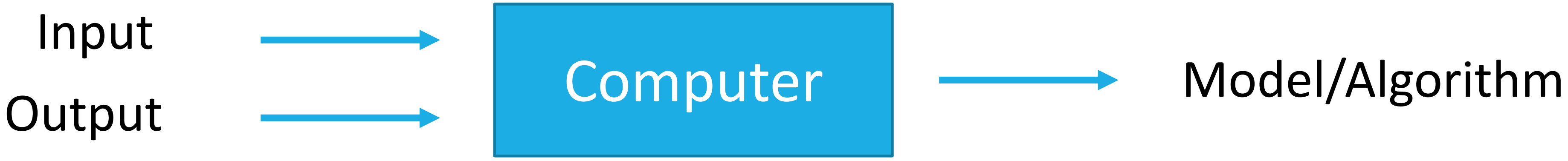
Input

Model/Algorithm

Computer

Output

Machine Learning



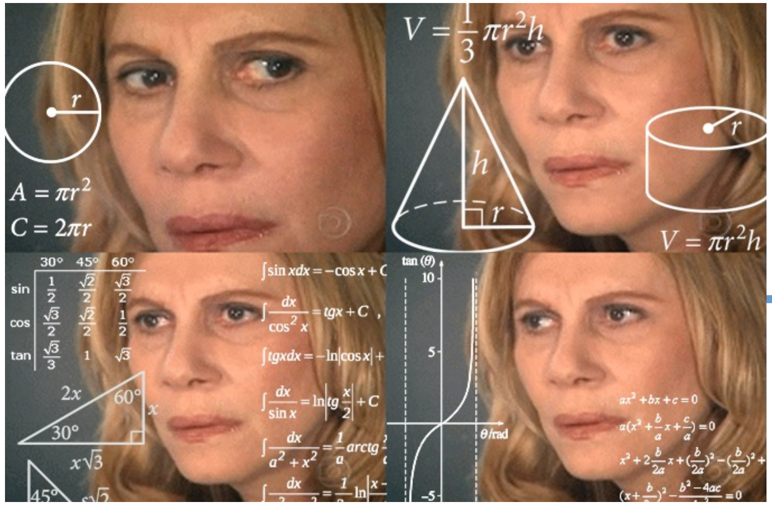
Input

Output

Computer

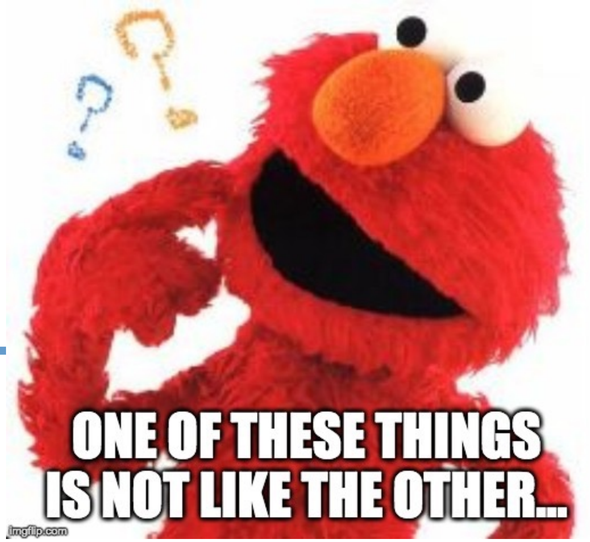
Model/Algorithm

*Learn by Example

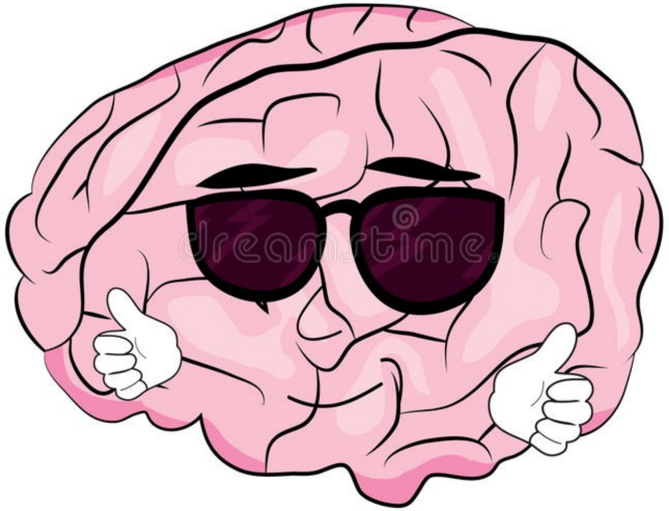


The Symbolists

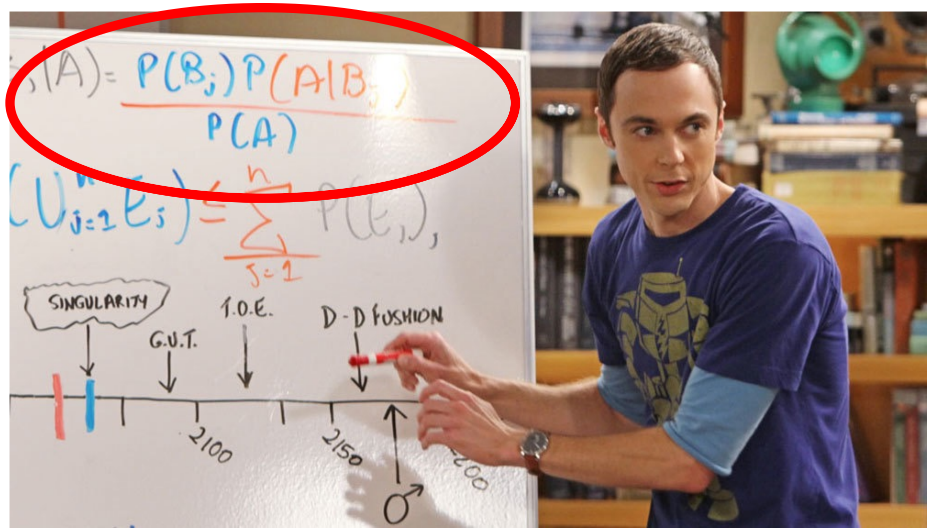
The Evolutionaries



The Analogizers



The Connectionists



The Bayesians

Why prediction?

In the future I might not have the outcome!

When I encounter a new case without the outcome… I can use the predictive model to get an idea of the outcome given inputs

Build a predictive model that does not make many mistakes

- Low misclassification rate (classification) or high variance explained (regression)
- I might be making decisions based on what the predictive model suggests…

Cross-validation

Allows us to check the robustness/overfitting of our model

Split your dataset into training and testing datasets

- Develop the model in the training dataset
- Evaluate the model in the testing dataset

No testing data? *k*-fold cross-validation

Ultimate goals

- Generalizability
- Prevent overfitting (beyond learning the quirks of our training data)
- Balance bias-variance trade-off

No Free Lunch

It is difficult to know a priori how well an algorithm will perform before we try it

- Assumptions in one problem might not hold for another problem (Wolpert, 1996)

There are too many algorithms… which one do you try first?

Perhaps *stacking* can provide a good solution

Data Trumps Algorithm

Sometimes running the machine learning algorithm is the easy part

Difficulties arise in …

- Collecting data
- Cleaning datasets
- Make meaningful variables
- Run and re-run the model
- Interpretability
- Decision-making

Data science is complex

Data trumps algorithm

- Even the simplest algorithm will do well by obtaining more data

Your Presentations

OSCAR GONZALEZ, PHD

MACHINE LEARNING

Final Presentations

Present your current findings on your project

- 15min or so
- Similar to the flash talks (due midnight 04/22/2024) – view on 04/23/2024
- Treat it as a conference presentation… and instead of feedback you'll get questions

Provide relevant background, describe the algorithm(s) you used, findings, and general discussion

Questions?

Final paper is due on 04/30/2024 at 11:30am

WHO CAN DO IT?
YOU CAN DO IT!
zipmeme

Last words

OSCAR GONZALEZ, PHD

MACHINE LEARNING

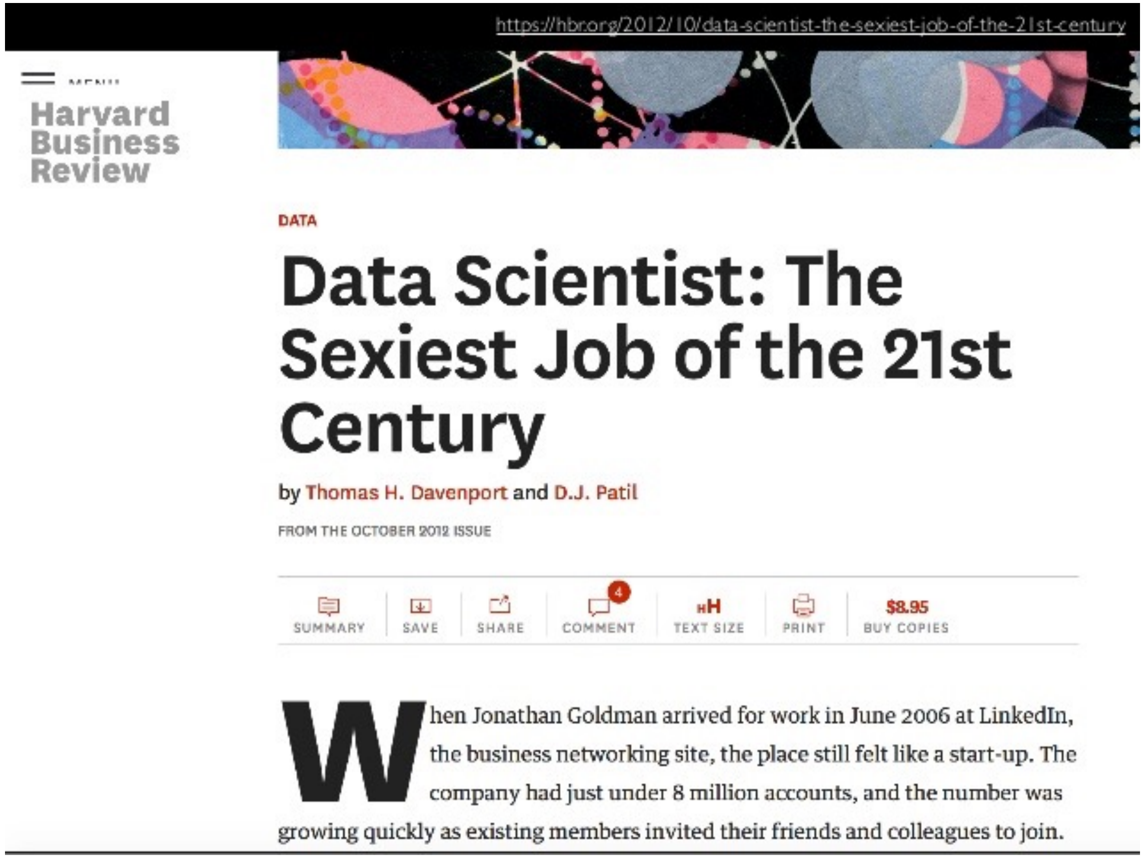
Machine Learning

It appears that it is here to stay, and it is currently making its way into psychology

Very marketable skill

- Say that you have been exposed to this

We learn patterns from a dataset, and we evaluate what we found



https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

DATA

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join.

It’s been a pleasure!

Thank you for signing up to this class…

Thank you for staying in this class…

Thank you for giving this class a chance!

Liked the course? Feedback? Let me know in your evals.

Today’s Class

1. Ethics Redux
2. Unsupervised Learning
3. Clustering methods
   - Hierarchical clustering
   - K-means clustering
4. Dimension reduction methods
   - Principal components
   - Factor analysis
5. Wrap Up

Next Class

Offline Presentations on 04/23/2024

Papers due on 04/30/2024 @ 11:30am