AI and ML Bootcamp – Day 1
Basic Machine Learning

Machine Learning Bootcamp

Santiago Segarra

Electrical and Computer Engineering
Computer Science
Rice University
segarra@rice.edu

May 7, 2025

It's great to meet you

- Associate Professor in ECE at Rice
  - $\Rightarrow$ Started 7 years ago
  - $\Rightarrow$ Courtesy appointments at CS and STATS
- Postdoc at MIT for 2 years (2016-2018)
  - $\Rightarrow$ Institute for Data, Systems, and Society
- PhD at University of Pennsylvania (2011-2016)
  - $\Rightarrow$ Electrical and Systems Engineering

- My research interests include
  - $\Rightarrow$ Network theory
  - $\Rightarrow$ AI and Machine learning (on graphs)
  - $\Rightarrow$ Graph signal processing

Some things that I work on

SP and ML on Graphs

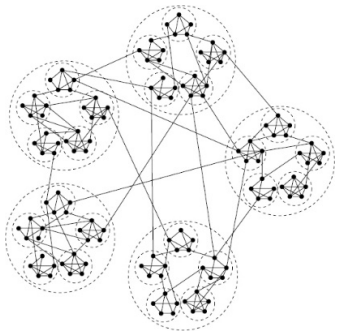
- Sampling and interpolation of network data
- Optimal design of graph processes
- Inference of network topology
- Deep learning on graphs

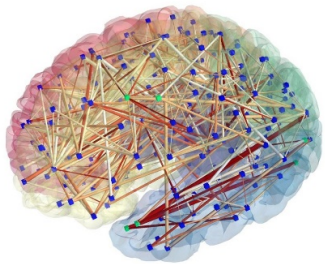
Foundations of Network Theory

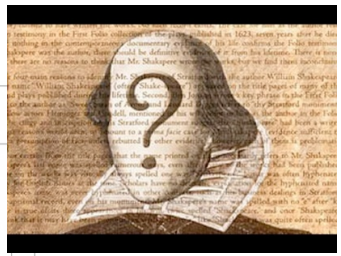
- Hierarchical clustering of networks
- Stability of centrality measures
- Metric structures of networks

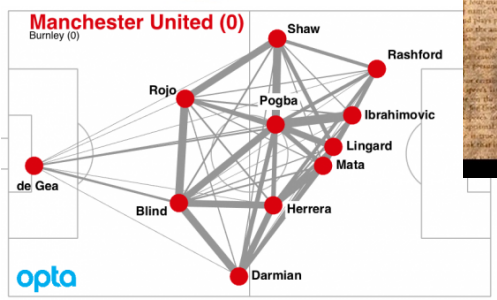
Interdisciplinary Applications

- Text authorship attribution
- Diagnosis of neural pathologies
- MetaGenomics









This afternoon in a nutshell

- We have 4 exciting modules ahead of us
  - $\Rightarrow$ Linear regression
  - $\Rightarrow$ Logistic regression
  - $\Rightarrow$ Decision trees and random forests
  - $\Rightarrow$ Clustering and dimensionality reduction

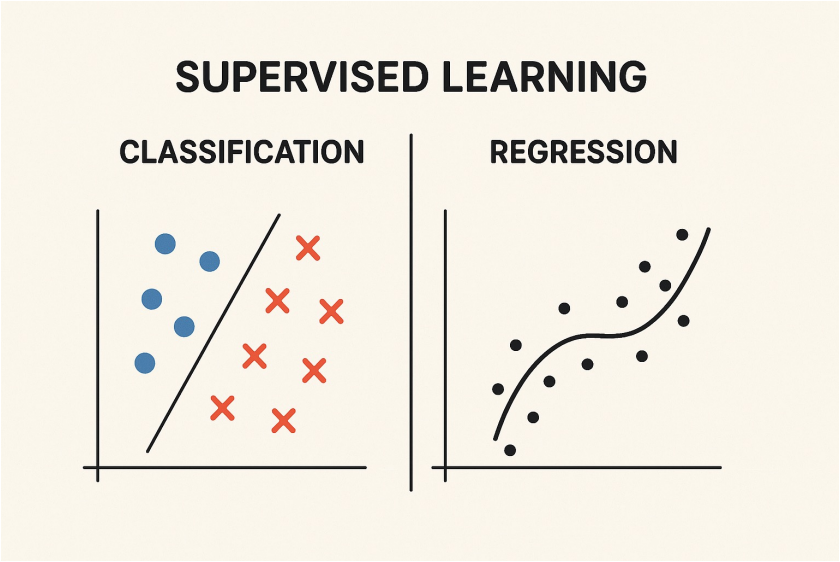
This afternoon in a nutshell

- We have 4 exciting modules ahead of us
  - $\Rightarrow$ Linear regression
  - $\Rightarrow$ Logistic regression
  - $\Rightarrow$ Decision trees and random forests
  - $\Rightarrow$ Clustering and dimensionality reduction
- Time for questions, let's make this interactive

This afternoon in a nutshell

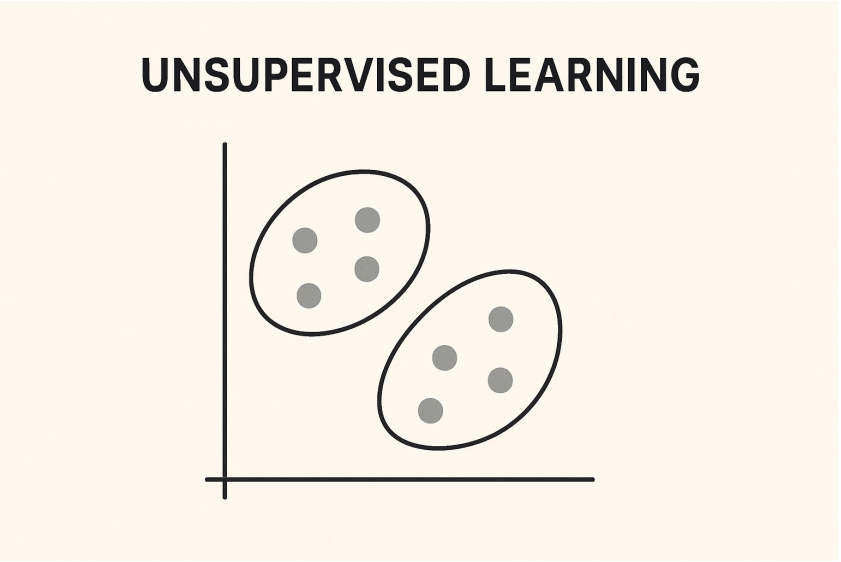
- We have 4 exciting modules ahead of us
  - $\Rightarrow$ Linear regression
  - $\Rightarrow$ Logistic regression
  - $\Rightarrow$ Decision trees and random forests
  - $\Rightarrow$ Clustering and dimensionality reduction
- Time for questions, let's make this interactive
- 'The Elements of Statistical Learning', Hastie, Tibshirani, and Friedman

Agenda at a high level



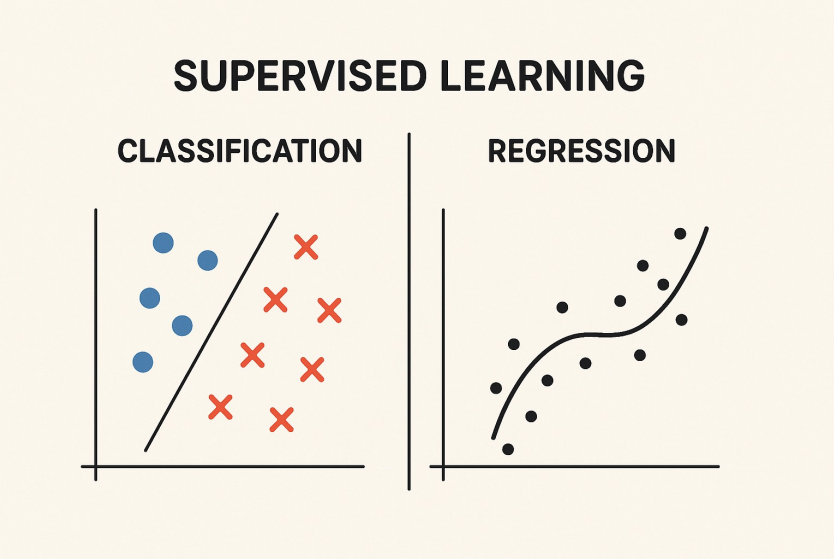
Logistic regression Linear regression

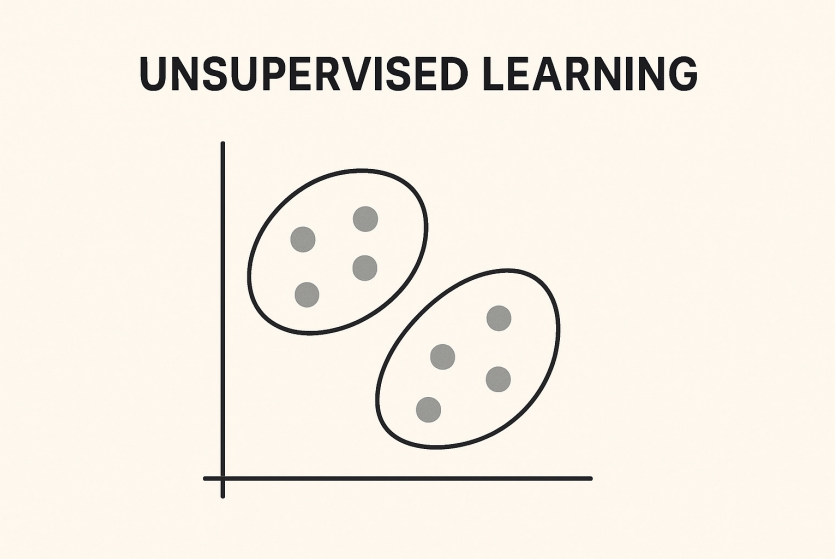
Decision trees and random forests



Clustering and dimensionality reduction

Agenda at a high level





Logistic regression Linear regression

Decision trees and random forests

Clustering and dimensionality reduction

- PS: Figures generated using ChatGPT 4o

Linear regression

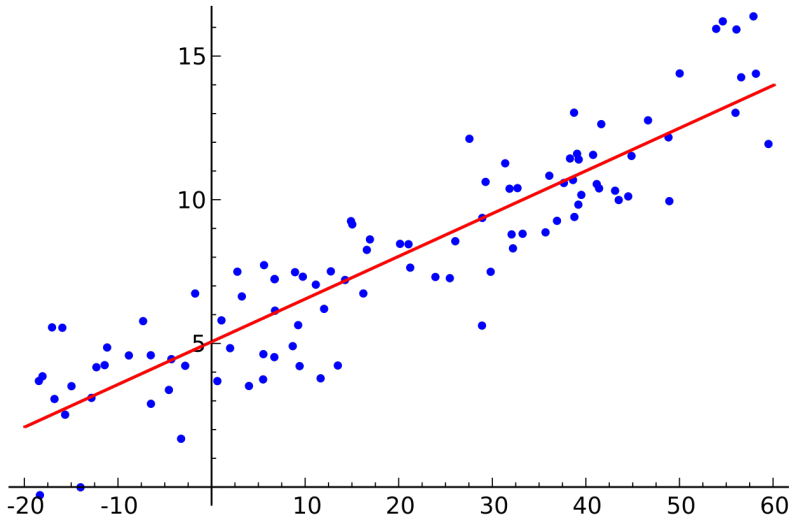
Linear regression

Logistic regression

Decision trees and random forests

Clustering and dimensionality reduction

Linear methods for regression



- Linear regression models assume that the output y is a linear function of the inputs $x_1, x_2, \ldots, x_p$
- Largely developed in the pre-computer era of statistics
- Still constitute a good model choice for plenty of problems

  $\Rightarrow$ Simple and easily interpretable
- Especially when we have small number of training samples
- Intuition serves as basis for more complex (non-linear) methods

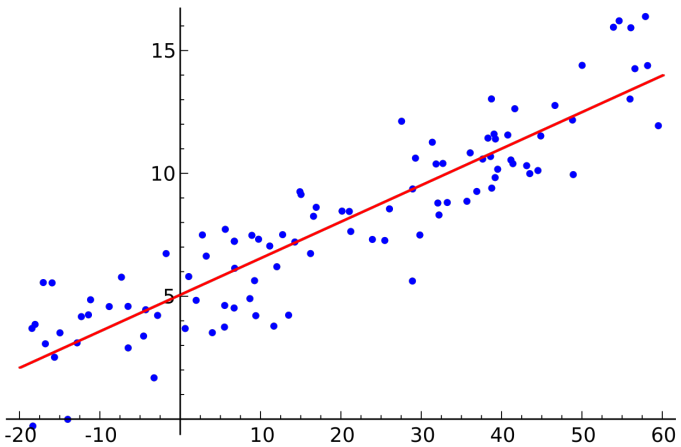
Linear regression formula

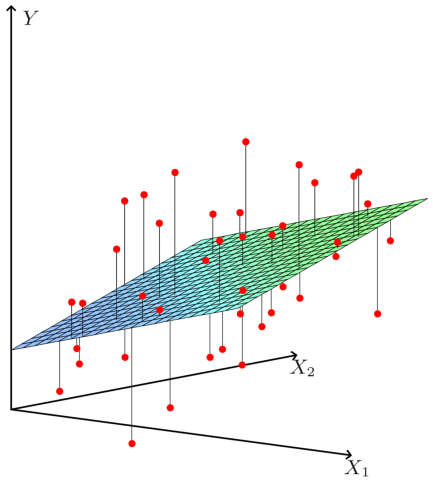
- We have an input vector $\mathbf{x} = [x_1, x_2, \ldots, x_p]^\top$
- We want to predict a real-valued output $y = f(\mathbf{x})$
  - $\Rightarrow$ Salary as a function of age and level of education
  - $\Rightarrow$ House price as a function of size, location, and construction date
  - $\Rightarrow$ Anxiety as a function of days since vacation and daily exercise

Linear regression formula

- We have an input vector $\mathbf{x} = [x_1, x_2, \ldots, x_p]^\top$
- We want to predict a real-valued output $y = f(\mathbf{x})$
  - $\Rightarrow$ Salary as a function of age and level of education
  - $\Rightarrow$ House price as a function of size, location, and construction date
  - $\Rightarrow$ Anxiety as a function of days since vacation and daily exercise

$$y = f(\mathbf{x}) = \beta_0 + x_1\beta_1 + x_2\beta_2 + \ldots + x_p\beta_p = \beta_0 + \sum_{j=1}^{p} x_j\beta_j = \beta_0 + \mathbf{x}^\top\boldsymbol{\beta}$$





Where's the 'linear' in linear regression?

- No matter where the $\mathbf{x}$ is coming from, the model is a linear combination given by the parameters β_j

Where's the 'linear' in linear regression?

- No matter where the **x** is coming from, the model is a linear combination given by the parameters β_j
- The variables **x** can come from many different sources
  - $\Rightarrow$ Measured quantitative inputs
  - $\Rightarrow$ Transformations of these inputs (log, powers, roots)
  - $\Rightarrow$ Indicator variables
  - $\Rightarrow$ Interaction variables $x_3 = x_1 x_2$

Where's the 'linear' in linear regression?

- No matter where the **x** is coming from, the model is a linear combination given by the parameters β_j
- The variables **x** can come from many different sources
  - $\Rightarrow$ Measured quantitative inputs
  - $\Rightarrow$ Transformations of these inputs (log, powers, roots)
  - $\Rightarrow$ Indicator variables
  - $\Rightarrow$ Interaction variables $x_3 = x_1 x_2$
- Notice that we pre-compute these 'non-linear' features and then we combine them linearly
  - $\Rightarrow$ This is still a linear regression

A simple motivating example

- We usually have a set of training data $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})$
  - $\Rightarrow$ From which we estimate the parameters β_j

A simple motivating example

- We usually have a set of training data $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})$
  - $\Rightarrow$ From which we estimate the parameters β_j
- What are $\mathbf{x}$, y, p and N in the following example?

State	GDP (millions)	Population	Unemployment Rate
North Dakota	52 089	757 952	2.4
Alabama	204 861	4 863 300	3.8
Mississippi	107 680	2 988 726	5.2
Arkansas	120 689	2 988 248	3.5
Kansas	153 258	2 907 289	3.8
Georgia	525 360	10 310 371	4.5
Iowa	178 766	3 134 693	3.2
West Virginia	73 374	1 831 102	5.1
Kentucky	197 043	4 436 974	5.2
Tennessee	???	6 651 194	3.0

Credit: NYU DSGA 1002

A simple motivating example

- We usually have a set of training data $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})$
  - $\Rightarrow$ From which we estimate the parameters β_j
- What are $\mathbf{x}$, y, p and N in the following example?

State	GDP (millions)	Population	Unemployment Rate
North Dakota	52 089	757 952	2.4
Alabama	204 861	4 863 300	3.8
Mississippi	107 680	2 988 726	5.2
Arkansas	120 689	2 988 248	3.5
Kansas	153 258	2 907 289	3.8
Georgia	525 360	10 310 371	4.5
Iowa	178 766	3 134 693	3.2
West Virginia	73 374	1 831 102	5.1
Kentucky	197 043	4 436 974	5.2
Tennessee	???	6 651 194	3.0

Credit: NYU DSGA 1002

- What do we expect the signs of β_1 and β_2 be?

A simple motivating example

- If we solve for β_j (we'll see how soon) the we can predict the GDP of Tennessee
  - $\Rightarrow$ The fit is quite good given the simplicity of our model

State	GDP	Estimate
North Dakota	52 089	46 241
Alabama	204 861	239 165
Mississippi	107 680	119 005
Arkansas	120 689	145 712
Kansas	153 258	136 756
Georgia	525 360	513 343
Iowa	178 766	158 097
West Virginia	73 374	59 969
Kentucky	197 043	194 829
Tennessee	328 770	345 352

Credit: NYU DSGA 1002

How do we obtain the parameters β?

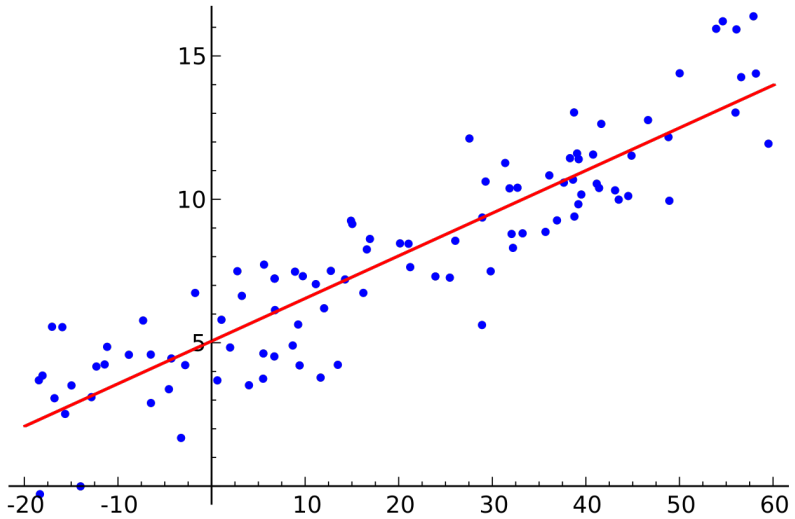
- Of course, we just use a library in python or R

How do we obtain the parameters β?

- Of course, we just use a library in python or R
- But, what is really going on? (at a high level)
- We select the β_j through a least squares method

  $\Rightarrow$ Minimize the residual sum of squares (empirical loss)

$$\text{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{N} (y^{(i)} - f(\mathbf{x}^{(i)}))^2$$



Linearizing prostate cancer

- Data from a study by Stanley et al. (1989)
- Relation between level of prostate-specific antigen and several clinical measures
  - $\Rightarrow$ log cancer volume
  - $\Rightarrow$ log prostate weight
  - $\Rightarrow$ age
  - $\Rightarrow$ seminal vesicle invasion
  - $\Rightarrow \ldots$

Linearizing prostate cancer

- Data from a study by Stanley et al. (1989)
- Relation between level of prostate-specific antigen and several clinical measures
  - $\Rightarrow$ log cancer volume
  - $\Rightarrow$ log prostate weight
  - $\Rightarrow$ age
  - $\Rightarrow$ seminal vesicle invasion
  - $\Rightarrow \ldots$
- Randomly split dataset into training ($N = 67$) and test set ($N = 30$)
- Compute the $\boldsymbol{\beta}$ via least squares as described before

Linearizing prostate cancer

- Let's interpret these results

Term	Coefficient	Std. Error	Z Score
`Intercept`	2.46	0.09	27.60
`lcavol`	0.68	0.13	5.37
`lweight`	0.26	0.10	2.75
`age`	−0.14	0.10	−1.40
`lbph`	0.21	0.10	2.06
`svi`	0.31	0.12	2.47
`lcp`	−0.29	0.15	−1.87
`gleason`	−0.02	0.15	−0.15
`pgg45`	0.27	0.15	1.74

Improving linear models via regularization

- Linear models with many variables can defeat their own purpose
- Overparametrization can lead to low prediction accuracy

  $\Rightarrow$ Why can overparametrization be a problem in linear regression?

Improving linear models via regularization

- Linear models with many variables can defeat their own purpose
- Overparametrization can lead to low prediction accuracy

  $\Rightarrow$ Why can overparametrization be a problem in linear regression?
- Interpretability might be harmed by having too many variables

  $\Rightarrow$ Sacrifice a bit of prediction power for interpretability

Improving linear models via regularization

- Linear models with many variables can defeat their own purpose
- Overparametrization can lead to low prediction accuracy
  $\Rightarrow$ Why can overparametrization be a problem in linear regression?
- Interpretability might be harmed by having too many variables
  $\Rightarrow$ Sacrifice a bit of prediction power for interpretability
- Both these problems can be tackled via regularization
  $\Rightarrow$ Ridge regression
  $\Rightarrow$ Lasso regression

Ridge regression

- Impose a penalty on the size of the coefficients $\boldsymbol{\beta}$

  $\Rightarrow$ A large β_j must mean that x_j has true explanatory power

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{N} (y^{(i)} - \beta_0 - \sum_{j=1}^{p} x_j^{(i)} \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Ridge regression

- Impose a penalty on the size of the coefficients $\boldsymbol{\beta}$

  $\Rightarrow$ A large β_j must mean that x_j has true explanatory power

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{N} (y^{(i)} - \beta_0 - \sum_{j=1}^{p} x_j^{(i)} \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- What is the parameter λ controlling?

Ridge regression

- Impose a penalty on the size of the coefficients $\boldsymbol{\beta}$

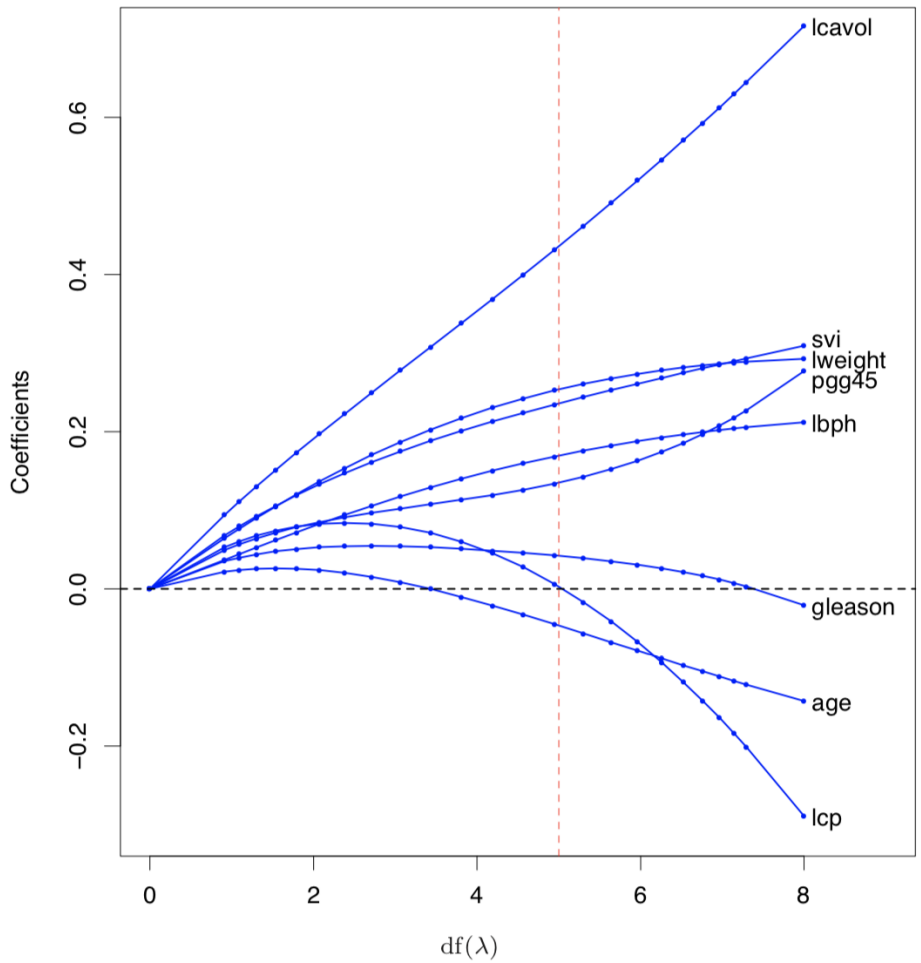
  $\Rightarrow$ A large β_j must mean that x_j has true explanatory power

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{N} (y^{(i)} - \beta_0 - \sum_{j=1}^{p} x_j^{(i)} \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- What is the parameter λ controlling?
- How to fix λ in practice?

Ridge regression

- We can compute $\hat{\beta}^{\text{ridge}}$ for different values of λ



Lasso regression

- Simply change the ℓ_2 norm penalization by an ℓ_1 norm

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{N} (y^{(i)} - \beta_0 - \sum_{j=1}^{p} x_j^{(i)} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

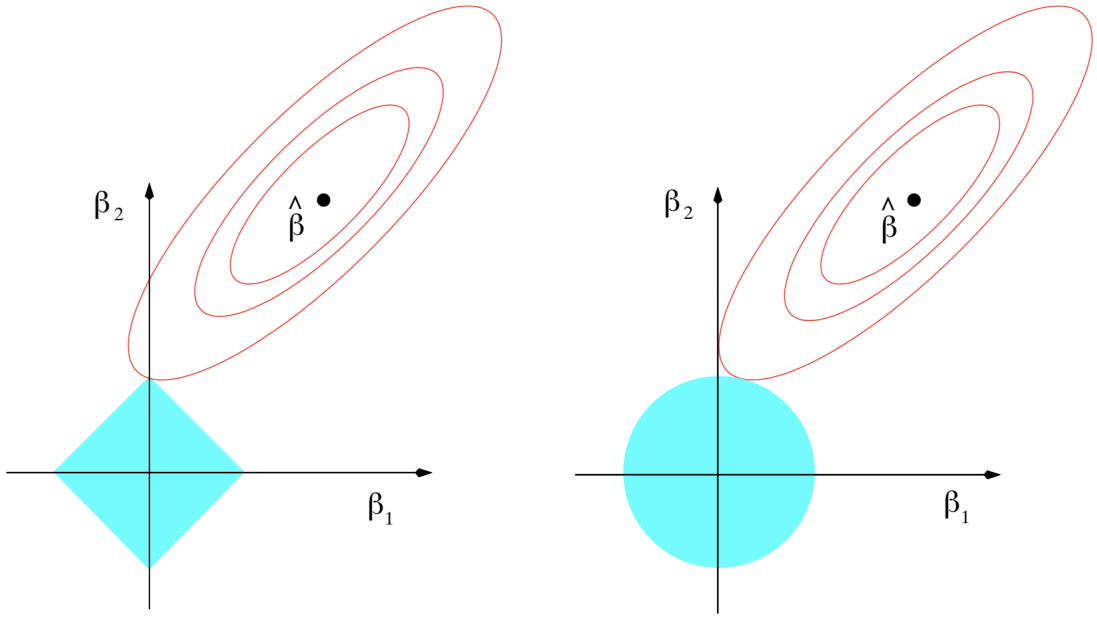
- What is this difference doing?

Lasso regression

- Simply change the ℓ_2 norm penalization by an ℓ_1 norm

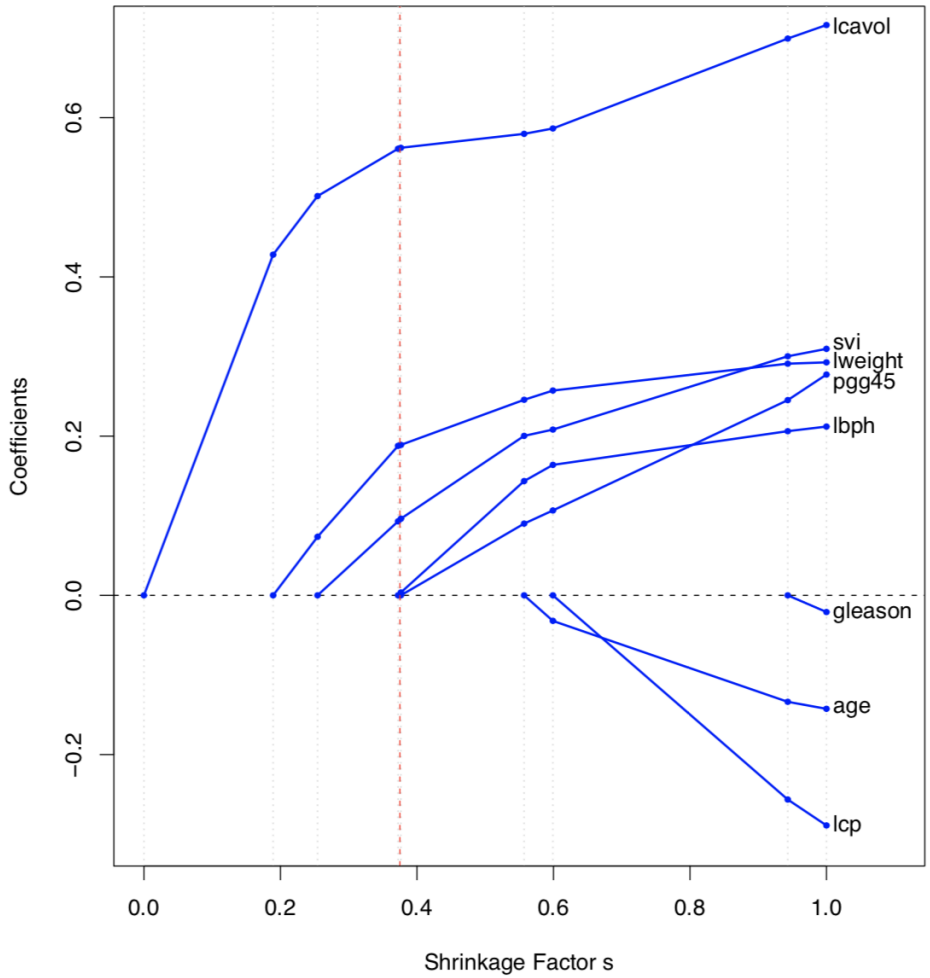
$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{N} (y^{(i)} - \beta_0 - \sum_{j=1}^{p} x_j^{(i)} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- What is this difference doing?



Lasso regression

- We can compute $\hat{\beta}^{\text{lasso}}$ for different values of λ



Improving cancer prediction via regularization

- Lasso works the best and selects the right variables

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.465	2.477	2.452	2.468
lcavol	0.680	0.740	0.420	0.533
lweight	0.263	0.316	0.238	0.169
age	−0.141		−0.046	
lbph	0.210		0.162	0.002
svi	0.305		0.227	0.094
lcp	−0.288		0.000	
gleason	−0.021		0.040	
pgg45	0.267		0.133	
Test Error	0.521	0.492	0.492	0.479

Pros and cons of linear regression

Pros

- Coefficients are easy to interpret as feature influence
- Supports regularization (e.g., Ridge, Lasso) to reduce overfitting
- Computationally efficient and scales well to large datasets
- Provides a good baseline model and is easy to implement

Cons

- Assumes a linear relationship between inputs and output
- Struggles with complex, non-linear patterns unless features are engineered
- Sensitive to outliers, which can distort the fit significantly

Logistic regression

Linear regression

Logistic regression

Decision trees and random forests

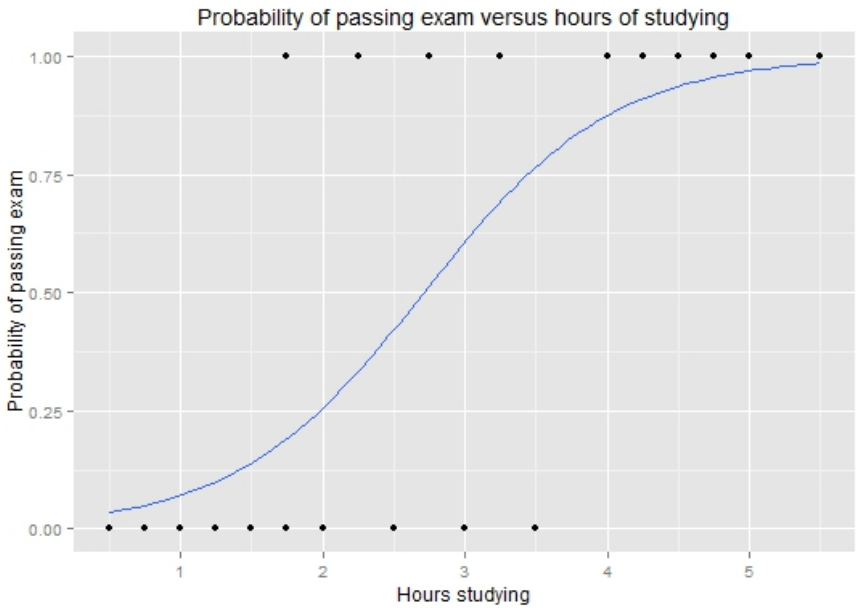
Clustering and dimensionality reduction

A baseline method for classification

- Linear regression is the simplest regression method
- Logistic regression is the simplest classification method

  $\Rightarrow$ Don't get fooled by the name
- Is an email spam based on the sender, the number of typos, the occurrence of certain words?
- Is a transaction fraudulent based on the date, the place, the amount, and the type?
- Would a person click in this ad based on his/her age, gender, location, and preferences?

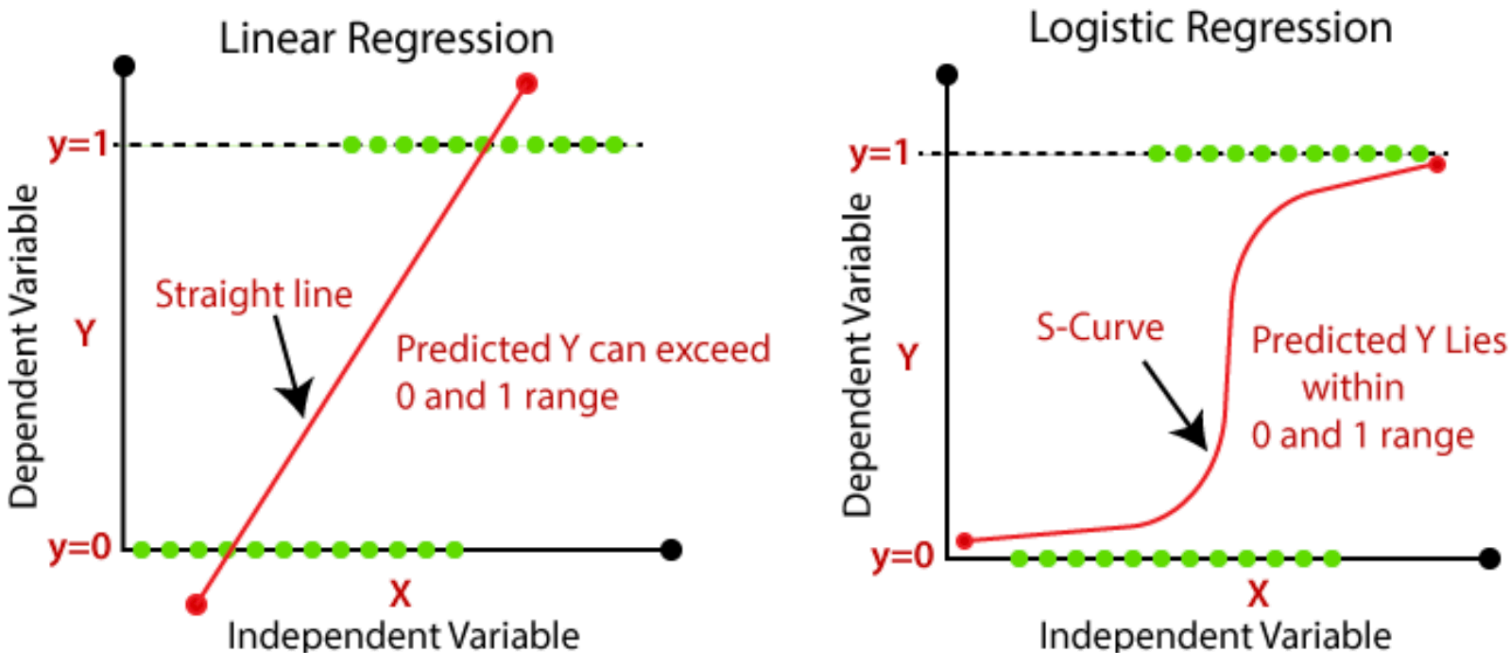
A simple toy example



- Typical example with a single feature
- Find a sigmoid function that best fits the observations
- The output is actually a probability

  $\Rightarrow$ Threshold at 0.5 to obtain a classifier

What if we fit a linear regression?



Credit: JavaTPoint

- One could fit a linear regression and then threshold the output
- Usually a bad choice since the least squares criterion does not really apply

Where is the linearity here?

- Well, that *S*-shaped curve does not seem very linear

Where is the linearity here?

- Well, that S-shaped curve does not seem very linear
- A logistic regression models the log odds ratio as linear

$$\log\left(\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... = \mathbf{x}^\top \boldsymbol{\beta}$$

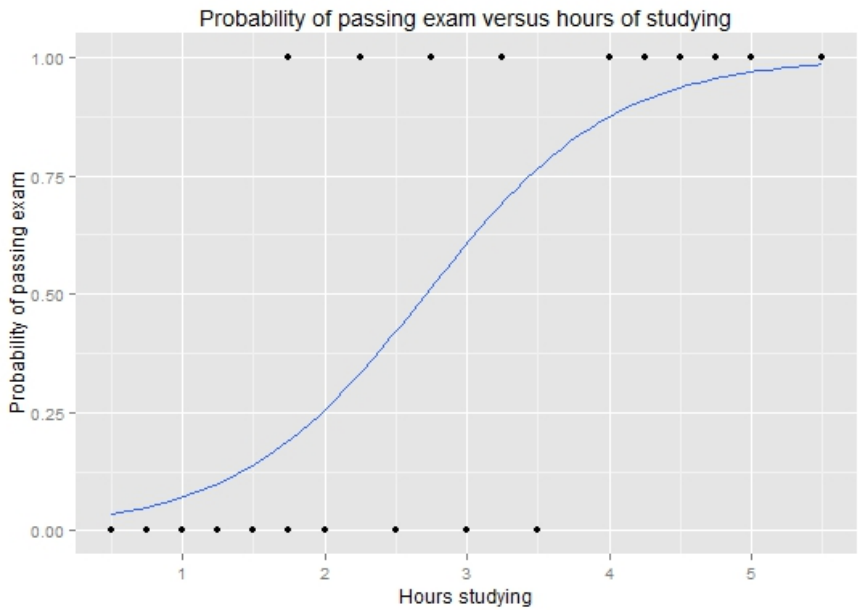
- Can be extended to $K > 2$ classes by fitting $K - 1$ log odds ratios
- E.g., will this customer rate the service as Good, Fair, or Bad?

Fitting a logistic regression

- Logistic regression models are fit via maximum likelihood
- Compute the probability of observing the samples given a set of β
- Change β to maximize such probability
- No closed-form solution, but simple iterative procedures

Fitting a logistic regression

- Logistic regression models are fit via maximum likelihood
- Compute the probability of observing the samples given a set of β
- Change β to maximize such probability
- No closed-form solution, but simple iterative procedures



Pros and cons of logistic regression

Pros

- Outputs class probabilities, enabling calibrated decisions
- Coefficients are interpretable and useful for feature analysis
- Simple, fast, and easy to implement

Cons

- Assumes a linear decision boundary in feature space
- Can perform poorly on complex or highly non-linear datasets
- Sensitive to outliers

Decision trees and random forests

Linear regression

Logistic regression

Decision trees and random forests

Clustering and dimensionality reduction

Decision trees

- Tree-based methods partition the feature space into a set of rectangles
- Fit a simple model (like a constant) in each one
- Conceptually simple yet powerful

Decision trees

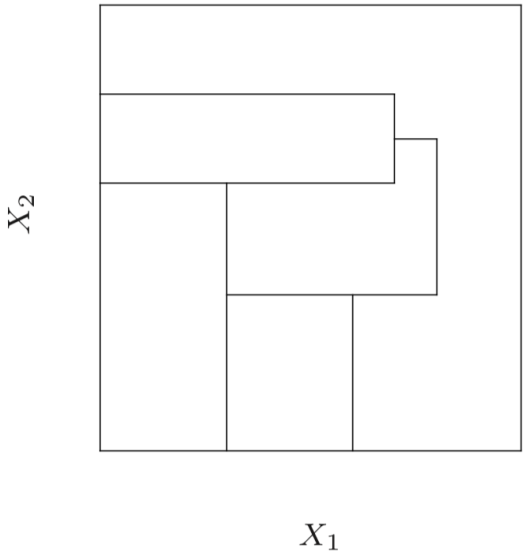
- Tree-based methods partition the feature space into a set of rectangles
- Fit a simple model (like a constant) in each one
- Conceptually simple yet powerful
- Can be used either for regression or classification
- Mimic human-decision making in some contexts

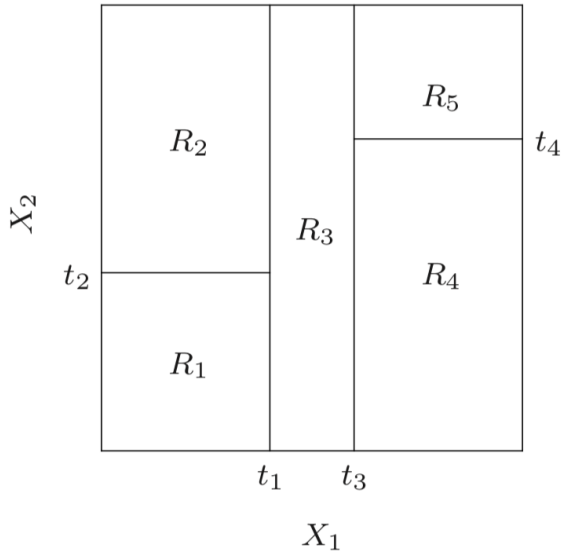
Recursive binary partitions

- Consider a regression problem with output y and two features x_1, x_2
- Partition the feature space with straight lines and assign constants to y

Recursive binary partitions

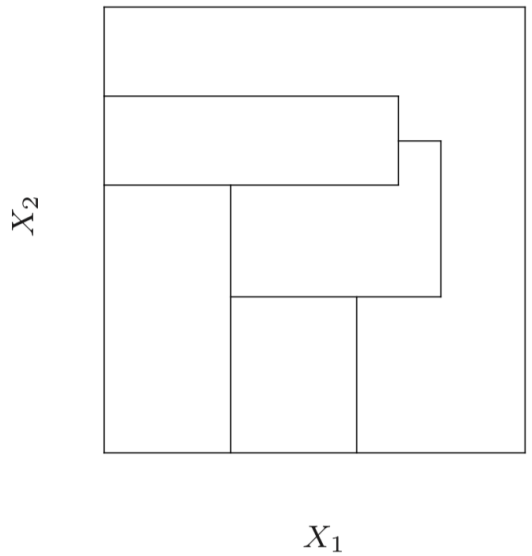
- Consider a regression problem with output y and two features x_1, x_2
- Partition the feature space with straight lines and assign constants to y

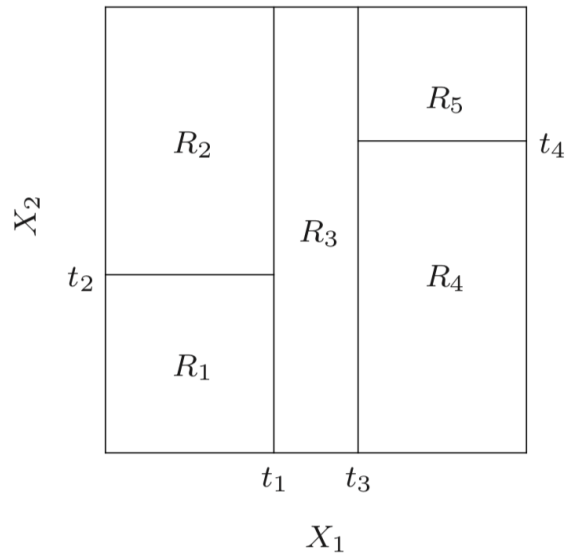




Recursive binary partitions

- Consider a regression problem with output y and two features x_1, x_2
- Partition the feature space with straight lines and assign constants to y





- The corresponding regression model predicts

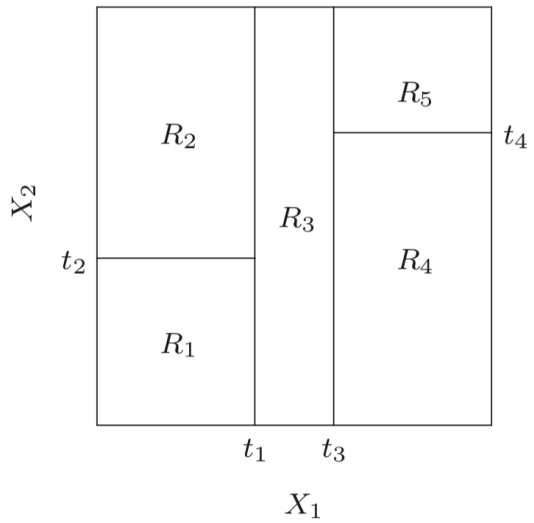
$$\hat{y} = \sum_{m=1}^{5} c_m \mathbb{I}(\mathbf{x} \in R_m)$$

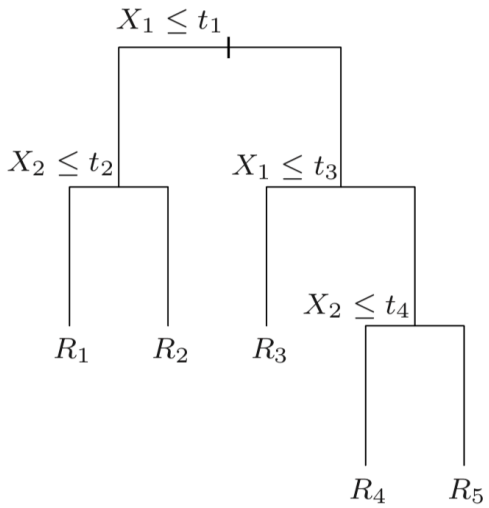
Where is the tree?

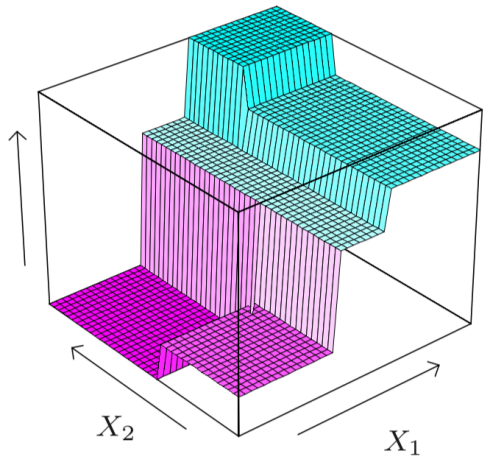
- First split the space into two regions
  - $\Rightarrow$ Model the response by the mean of y in each region
- Choose variable and split-point to achieve best fit and repeat

Where is the tree?

- First split the space into two regions
  - $\Rightarrow$ Model the response by the mean of y in each region
- Choose variable and split-point to achieve best fit and repeat







Interpretability

- A key advantage of the recursive binary tree is its interpretability
- Feature space partition is fully described by a single tree
- With more than two inputs the rectangular visualization gets messy
  $\Rightarrow$ But the tree is still valid
- Mimic decision making of, e.g., medical doctors in some situations
  $\Rightarrow$ Stratify the population into classes of high and low features

Regression trees

- Scalar output y and p features $\mathbf{x} \in \mathbb{R}^p$
- Train from several observations $(\mathbf{x}^{(i)}, y^{(i)})$
- The algorithm needs to automatically decide (at each point):
  - $\Rightarrow$ The splitting variable
  - $\Rightarrow$ The splitting point

Regression trees

- Scalar output y and p features $\mathbf{x} \in \mathbb{R}^p$
- Train from several observations $(\mathbf{x}^{(i)}, y^{(i)})$
- The algorithm needs to automatically decide (at each point):
  - $\Rightarrow$ The splitting variable
  - $\Rightarrow$ The splitting point
- Given a partition of feature space into M regions $R_1, \ldots R_M$
- The constant prediction that minimizes the sum of squares is the average

$$\hat{c}_m = \text{mean}(y^{(i)} | \mathbf{x}^{(i)} \in R_m)$$

Regression trees

- Scalar output y and p features $\mathbf{x} \in \mathbb{R}^p$
- Train from several observations $(\mathbf{x}^{(i)}, y^{(i)})$
- The algorithm needs to automatically decide (at each point):
  - $\Rightarrow$ The splitting variable
  - $\Rightarrow$ The splitting point
- Given a partition of feature space into M regions $R_1, \ldots R_M$
- The constant prediction that minimizes the sum of squares is the average

$$\hat{c}_m = \text{mean}(y^{(i)} | \mathbf{x}^{(i)} \in R_m)$$

- Finding the best binary partition is computationally infeasible

Greedy partitions

- Starting with all of the data, consider splitting variable j and split point s
  $\Rightarrow$ Let's look at the two following half-planes

$$R_1(j,s) = \{\mathbf{x}|x_j \leq s\}, \quad R_2(j,s) = \{\mathbf{x}|x_j > s\}$$

- We seek to optimize j and s by minimizing

$$\min_{j,s}\left[\min_{c_1}\sum_{\mathbf{x}^{(i)}\in R_1(j,s)}(y^{(i)}-c_1)^2 + \min_{c_2}\sum_{\mathbf{x}^{(i)}\in R_2(j,s)}(y^{(i)}-c_2)^2\right]$$

- Notice that this is not globally optimal but it is relatively easy to implement

How large should the tree be?

- Very large tree might overfit the data and perform poorly
- Small tree might not capture the important structure in the data
- Tree size is an important hyperparameter of the model
- It is better to grow a large tree and then prune it

  $\Rightarrow$ Cost-complexity pruning

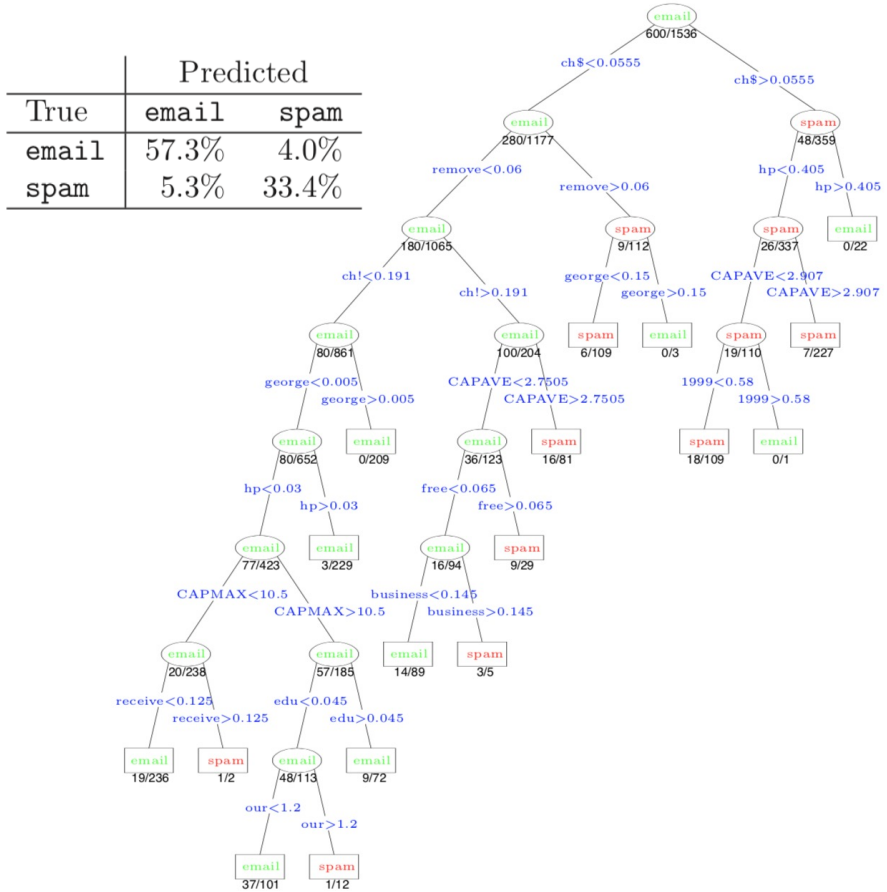
Spam example

- Classification tree to detect spam based on email features
- Features: number of times that specific words and characters are used

Spam example

- Classification tree to detect spam based on email features
- Features: number of times that specific words and characters are used

	Predicted	
True	email	spam
email	57.3%	4.0%
spam	5.3%	33.4%



Pros and cons of decision trees

Pros

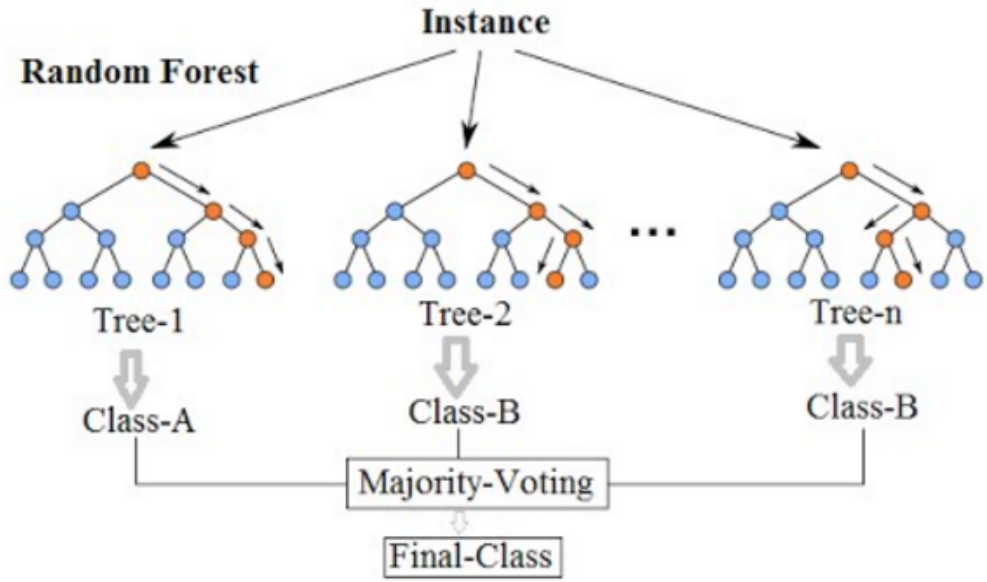
- Easy to visualize, explain, and interpret
- Handles missing data and both numerical/categorical features well
- Performs implicit feature selection through splitting criteria

Cons

- Highly prone to overfitting without pruning
- Sensitive to small variations in the data (high variance)
- Typically outperformed by ensemble methods like random forests or boosting

Ensemble methods

- A single decision tree is rarely used if we want to maximize performance
- Rely on the idea of ensemble methods to generate random forests



- Each tree is based on only a subset of the features

Pros and cons of random forests

Pros

- Reduces overfitting by averaging many decorrelated trees
- Strong performance on many datasets, including imbalanced ones

Cons

- Less interpretable than a single decision tree
- Can be computationally expensive with many trees or high-dimensional data

Clustering and dimensionality reduction

Linear regression

Logistic regression

Decision trees and random forests

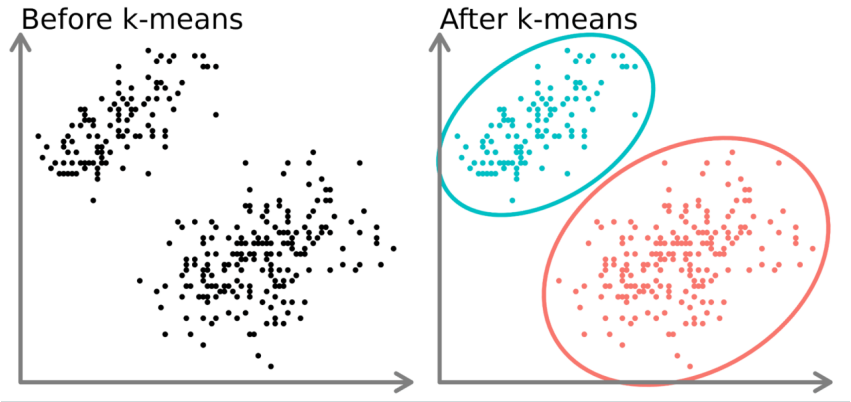
Clustering and dimensionality reduction

Unsupervised vs. supervised learning

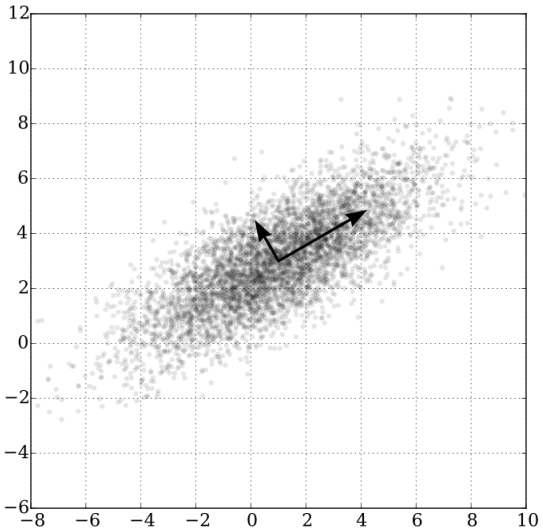
- This whole time we have been talking about supervised learning methods
- Given a set of samples $(\mathbf{x}^{(i)}, y^{(i)})$, we want to learn a map from $\mathbf{x}$ to y so that when presented with a new $\mathbf{x}^{(j)}$ we can predict $y^{(j)}$
- Although arguably the most common use of learning, it is not the only one
- In this last module, we will focus on the most common unsupervised learning methods
- Draw inferences from datasets $\mathbf{x}^{(i)}$ without labels

Notions that we will cover in this module

- K-means Clustering



- Principal Component Analysis (PCA)



Clustering

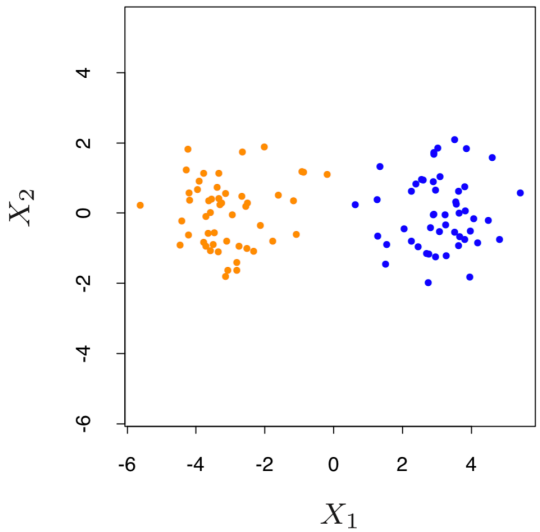
- Grouping or segmenting a collection of objects into subsets or clusters such that those within each cluster are more closely related to one another than objects assigned to different clusters
- Very intuitive definition but somewhat ambiguous

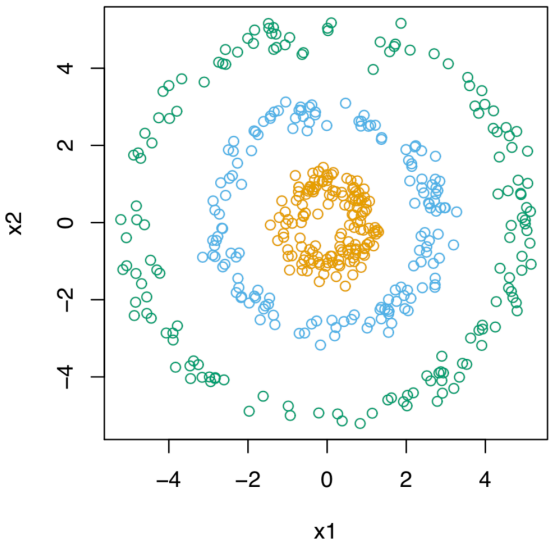
  $\Rightarrow$ What do we mean by 'more closely related'?

Clustering

- Grouping or segmenting a collection of objects into subsets or clusters such that those within each cluster are more closely related to one another than objects assigned to different clusters
- Very intuitive definition but somewhat ambiguous

  $\Rightarrow$ What do we mean by 'more closely related'?

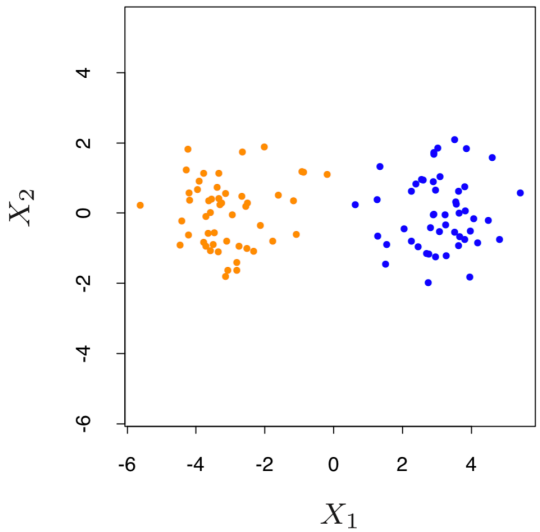


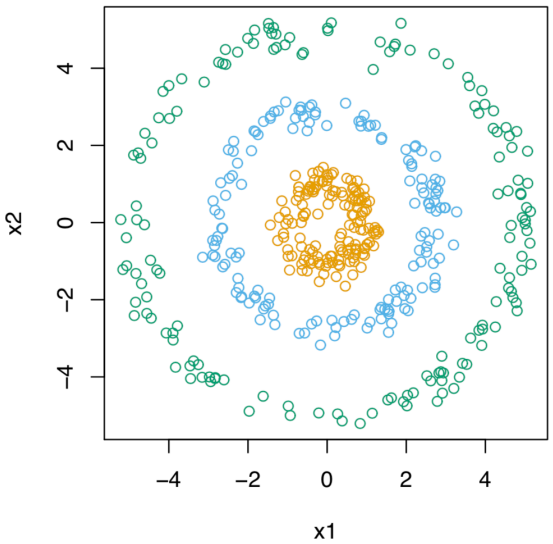


Clustering

- Grouping or segmenting a collection of objects into subsets or clusters such that those within each cluster are more closely related to one another than objects assigned to different clusters
- Very intuitive definition but somewhat ambiguous

  $\Rightarrow$ What do we mean by 'more closely related'?





- Need to define notion of similarity (dissimilarity) between points

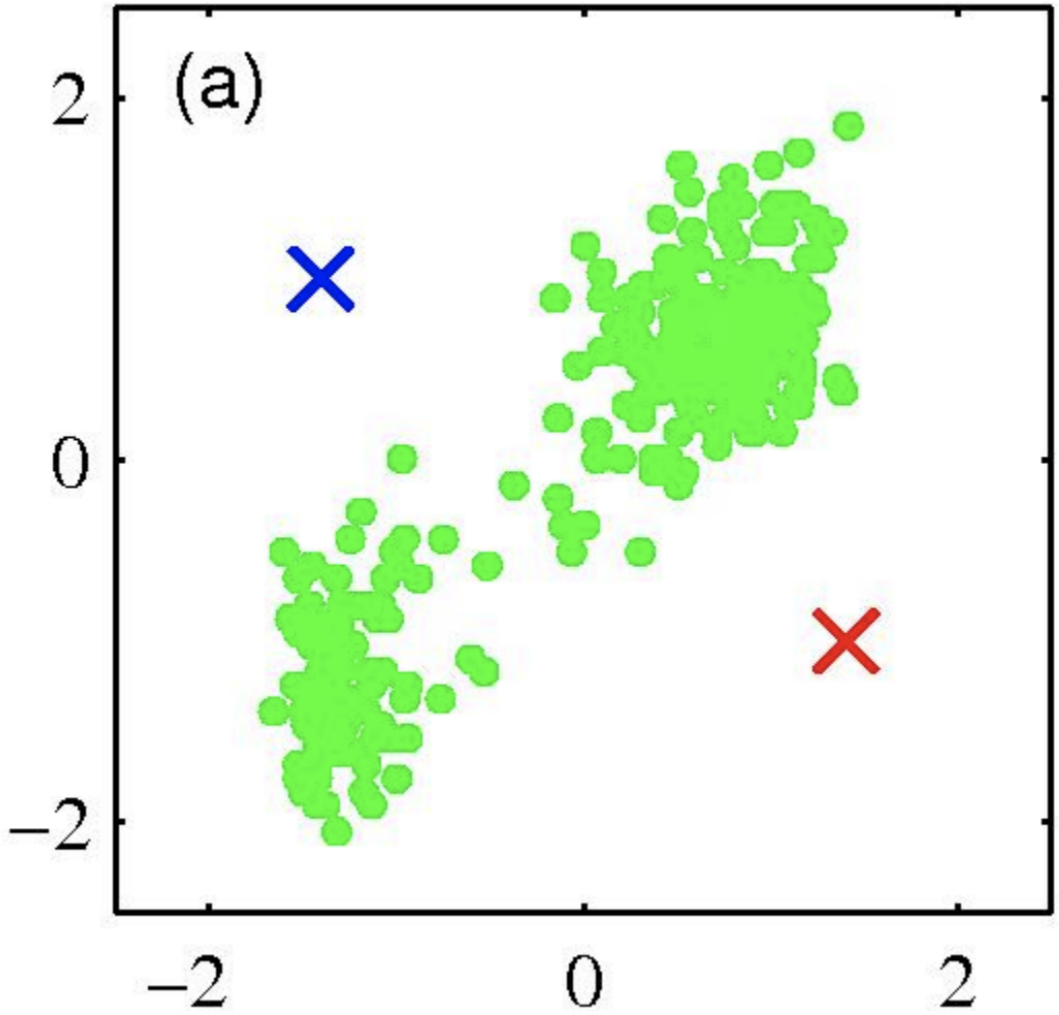
k-means clustering

- An iterative clustering algorithm for a point cloud
- Initialize: Pick k random points as cluster centers
- Alternate:
  - $\Rightarrow$ 1. Assign data points to closest cluster center
  - $\Rightarrow$ 2. Change the cluster center to the average of its assigned points
- Finalize: When no assignments change
- Guaranteed to converge in a finite number of iterations

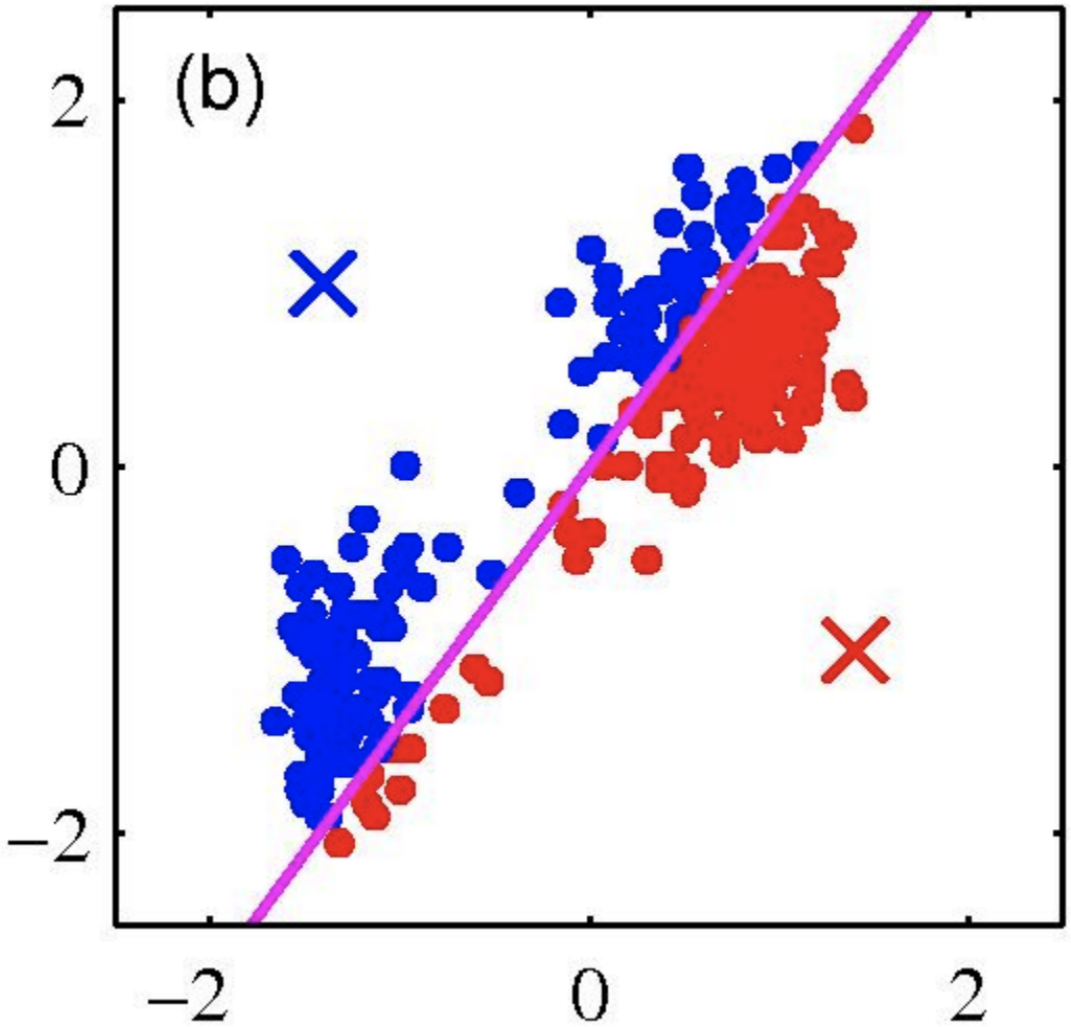
$$\min_{\{\mathbf{u}_i\}_{i=1}^k} \min_{\{C_i\}_{i=1}^k} \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{u}_i\|_2^2$$

- Block coordinate descent in the above expression

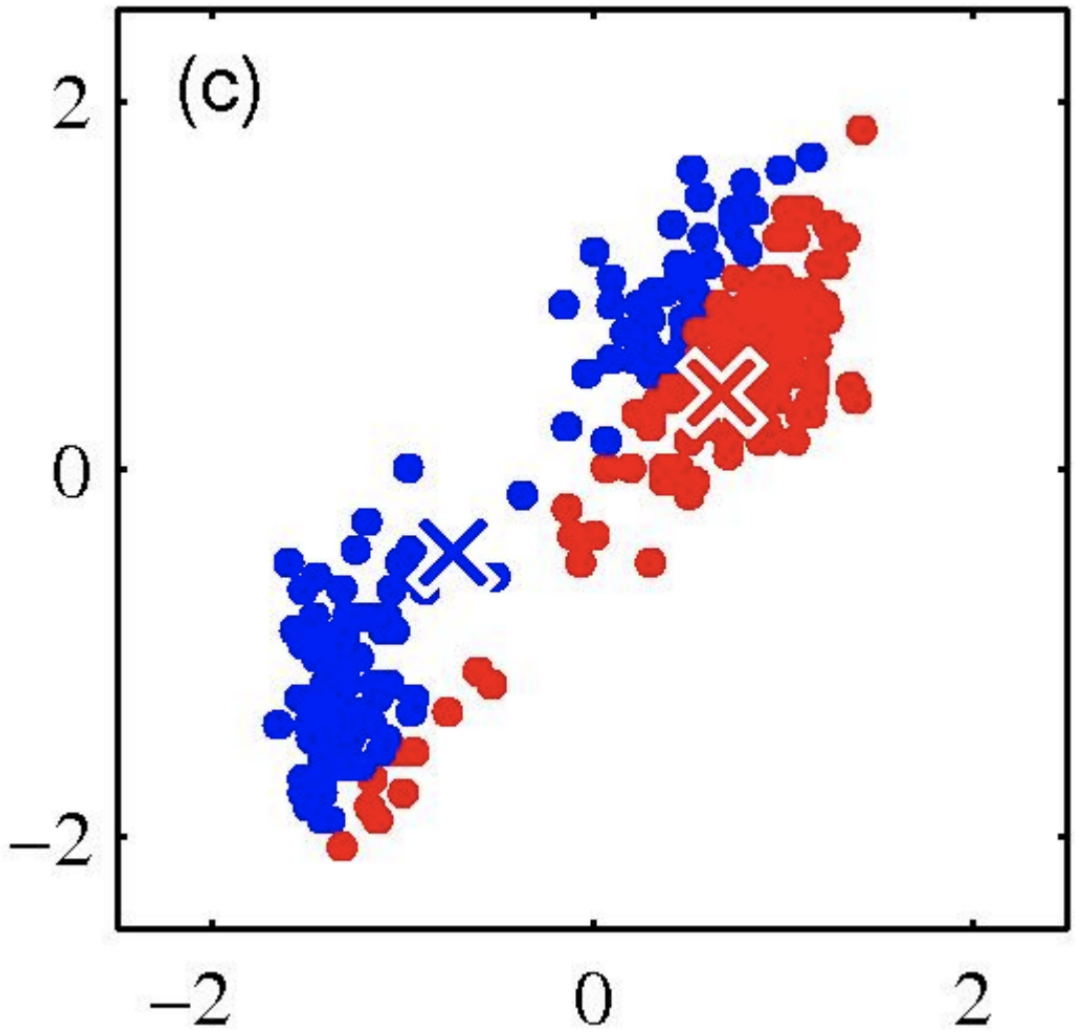
k-means clustering: example



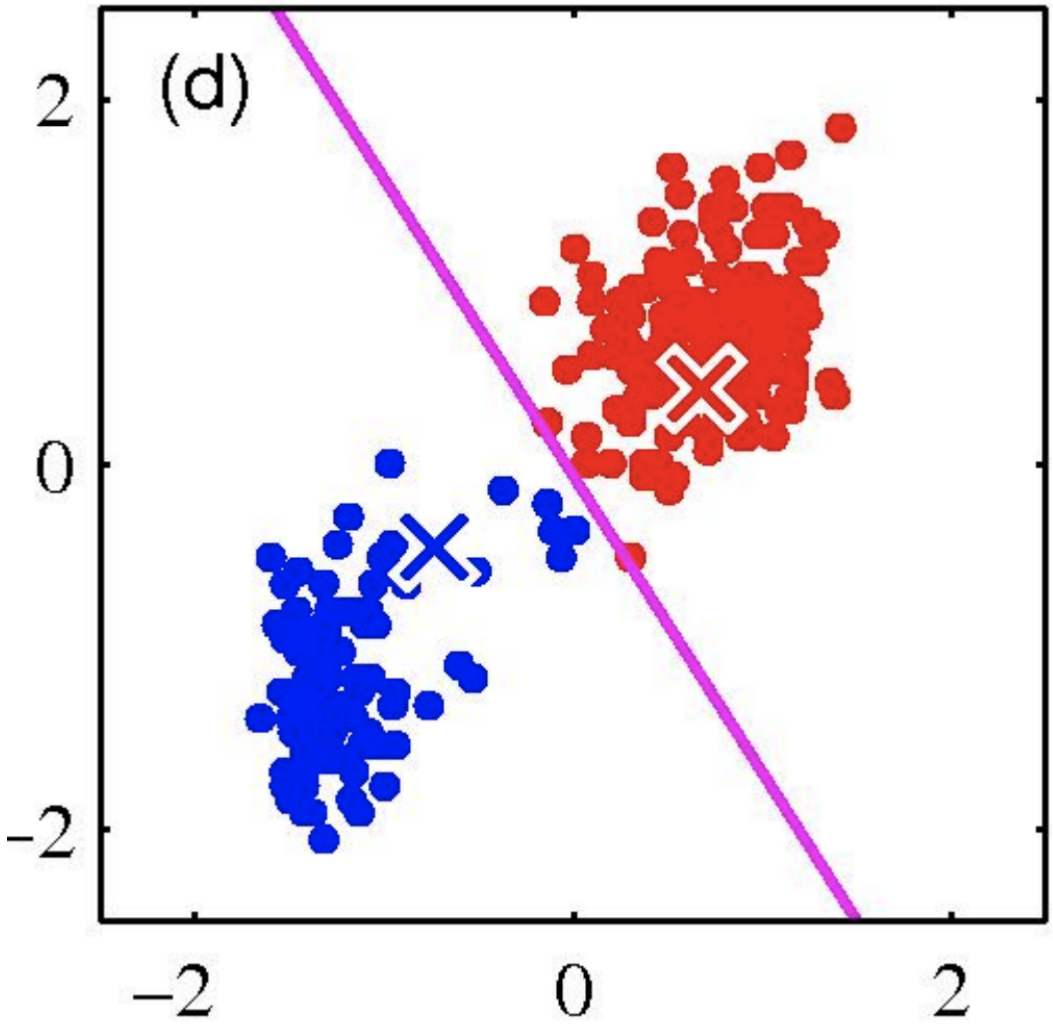
k-means clustering: example



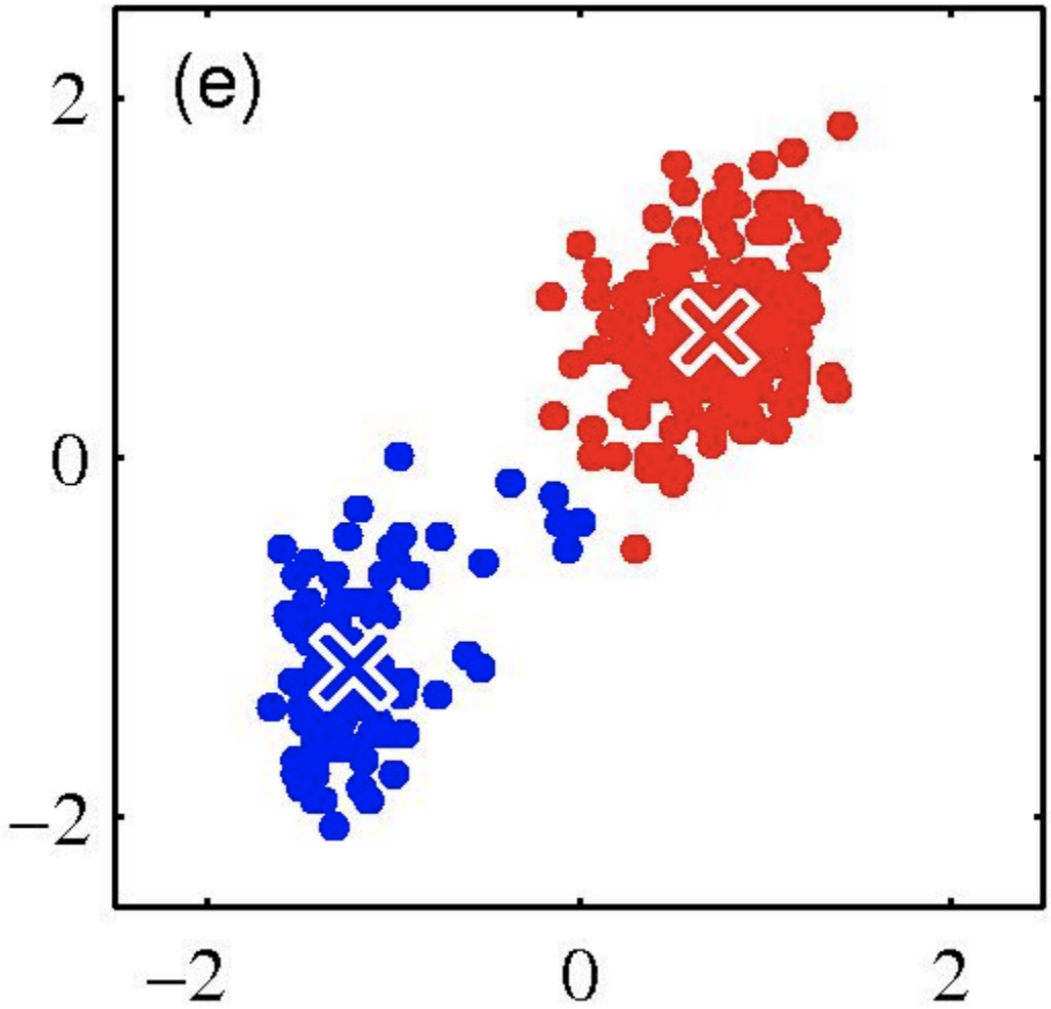
k-means clustering: example



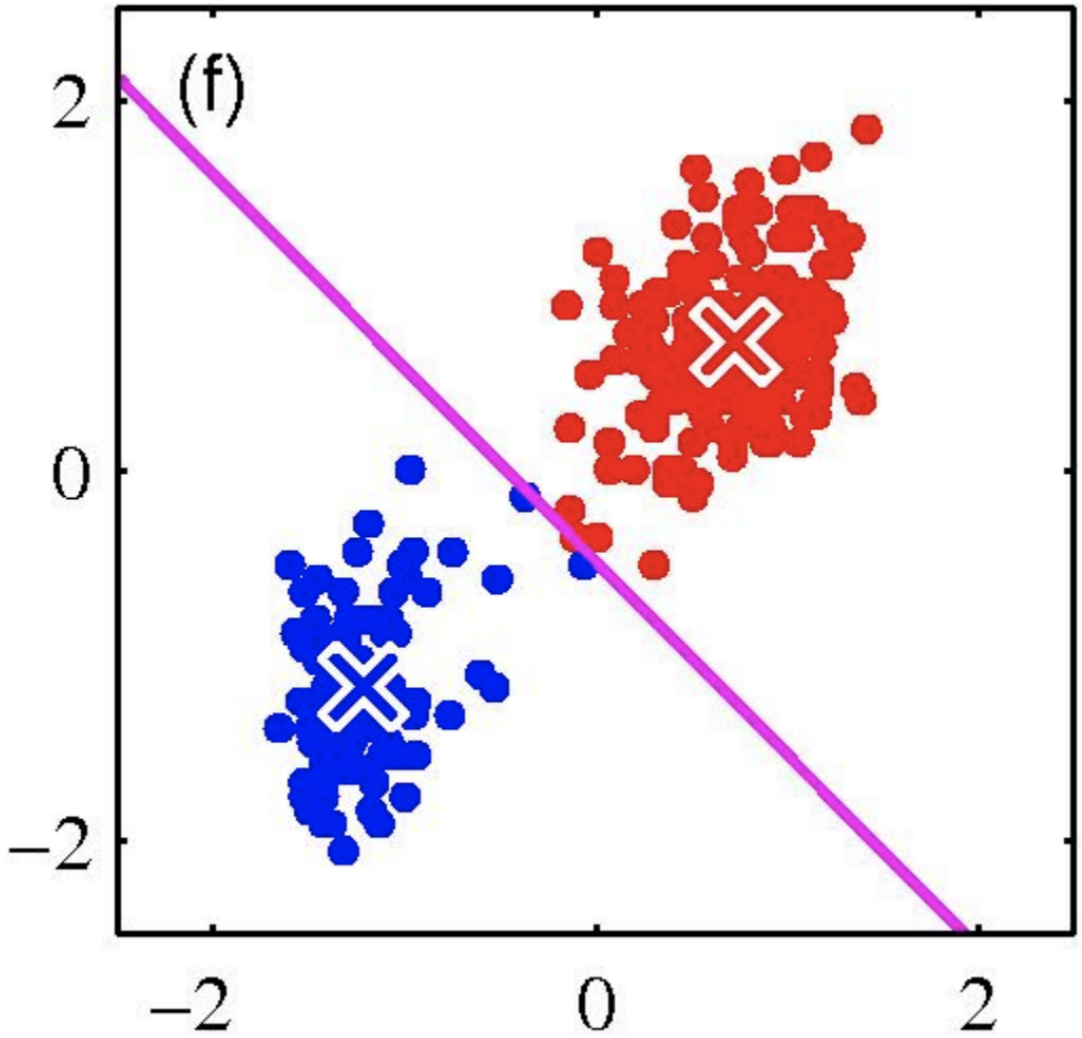
k-means clustering: example



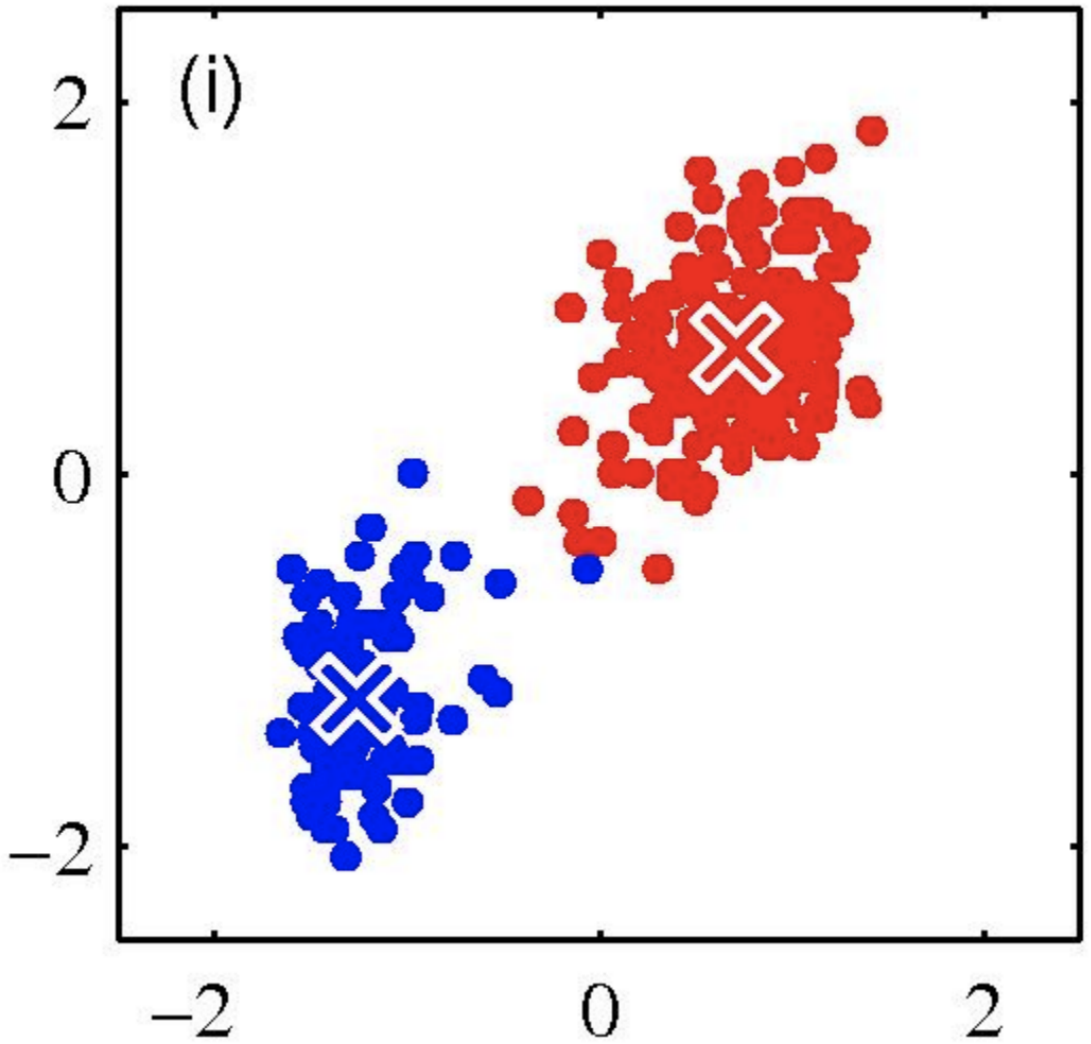
k-means clustering: example



k-means clustering: example



k-means clustering: example



Pros and cons of k-means clustering

Pros

- Simple and fast to implement, even for large datasets
- Efficient and scalable with low memory footprint

Cons

- Sensitive to initialization and outliers
- Assumes spherical clusters of similar size and density

Principal component analysis

- Sequence of projections of the data, mutually uncorrelated, and ordered in variance

Principal component analysis

- Sequence of projections of the data, mutually uncorrelated, and ordered in variance

$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \mathbf{V}_q \mathbf{V}_q^\top \mathbf{x}^{(i)}\|$$

- How can we interpret this equation, where $\mathbf{V}_q$ is orthogonal of size $p \times q$?

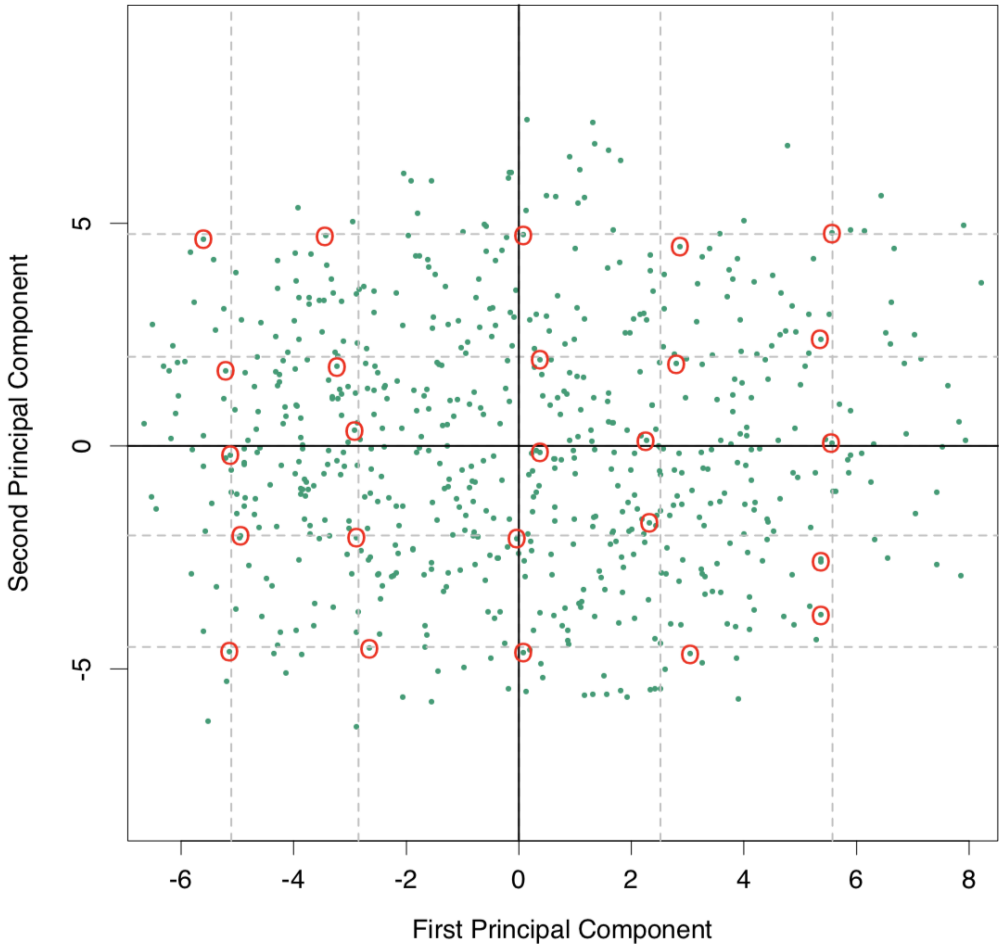
Principal component analysis

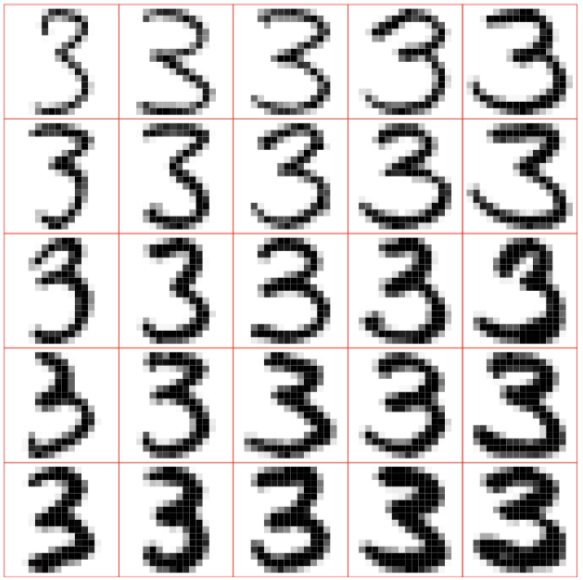
- Sequence of projections of the data, mutually uncorrelated, and ordered in variance

$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \mathbf{V}_q \mathbf{V}_q^\top \mathbf{x}^{(i)}\|$$

- How can we interpret this equation, where $\mathbf{V}_q$ is orthogonal of size $p \times q$?
- PCA is very useful for pre-processing before supervised learning
- Not truly feature selection but feature engineering
- Can be turned into feature selection via sparse PCA

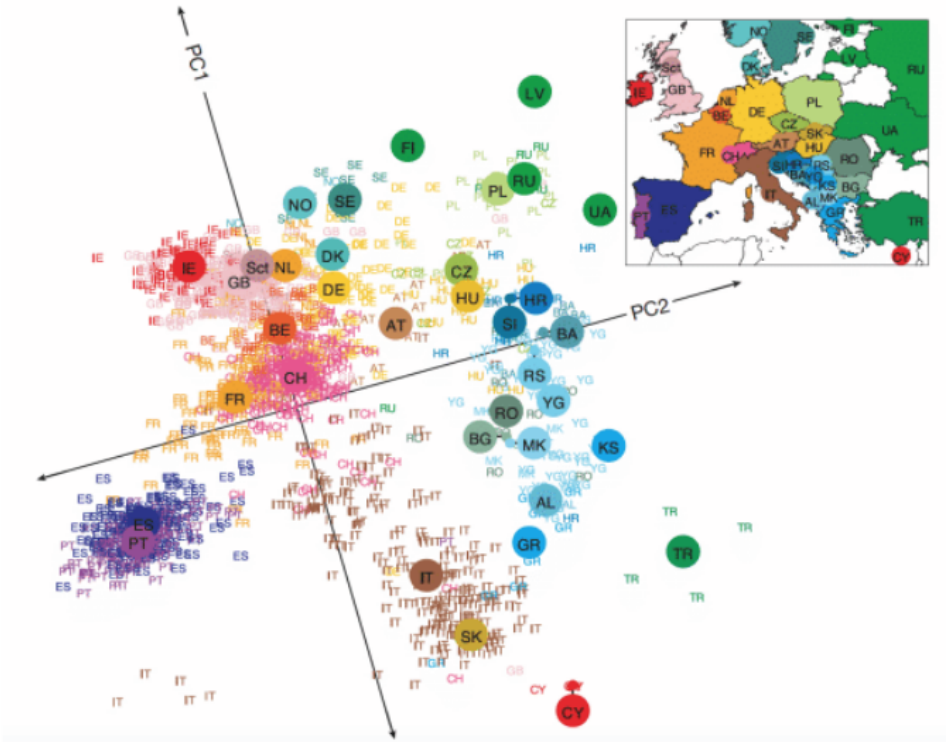
Principal components of digits





Cute application

- Genes mirror geography within Europe, Novembre et al., Nature (2008)
- Two-dimensional embedding of 'gene similarity' matrix
  - $\Rightarrow$ Consistent with origins of individuals in European map



Pros and cons of PCA

Pros

- Easy to compute
- Reduces dimensionality and mitigates the curse of dimensionality
- Can improve speed and performance of downstream ML algorithms

Cons

- Principal components are linear combinations and hard to interpret
- May discard important information if too much variance is removed

Some basic methods that we did not cover

- k-nearest-neighbors
- Simple, non-parametric method based on distance to k closest points
- No training required; sensitive to k and distance metric
- Naive Bayes
- Fast probabilistic classifier using Bayes' rule
- Assumes independence between input features
- Support Vector Machines
- Finds maximum-margin hyperplane to separate classes
- Can handle non-linear boundaries using kernel tricks
- Boosting Methods
- Combines many weak learners into a strong model
- Sequentially focuses on correcting previous errors

What comes next?

- Intro to neural networks and deep learning with Tasos
  - $\Rightarrow$ Basis for modern ML
  - $\Rightarrow$ Transformers (basis for LLMs) build on top of these
- Methods behind modern cool AI-stuff
  - $\Rightarrow$ AI generating realistic faces
  - $\Rightarrow$ AI replicating a landscape in the style of Dali
  - $\Rightarrow$ Large language models (ChatGPT)