Machine Learning Concepts

1.4

Quiz

Topics from last week

- Supervised Learning models
  - Decision trees
  - Linear models: ridge, lasso, elasticNet
  - Nearest Neighbour
- Penalty, shrinkage, Complexity control
-

Last week summary

We reviewed a number of **Machine Learning Models** and their **R package**:

- **Linear Model:** assumptions, advantage, feature transformations
- **Nearest Neighbour:** algorithm, advantages, difficulties, complexity
- **Trees**: algorithm, advantages, complexity and complexity control

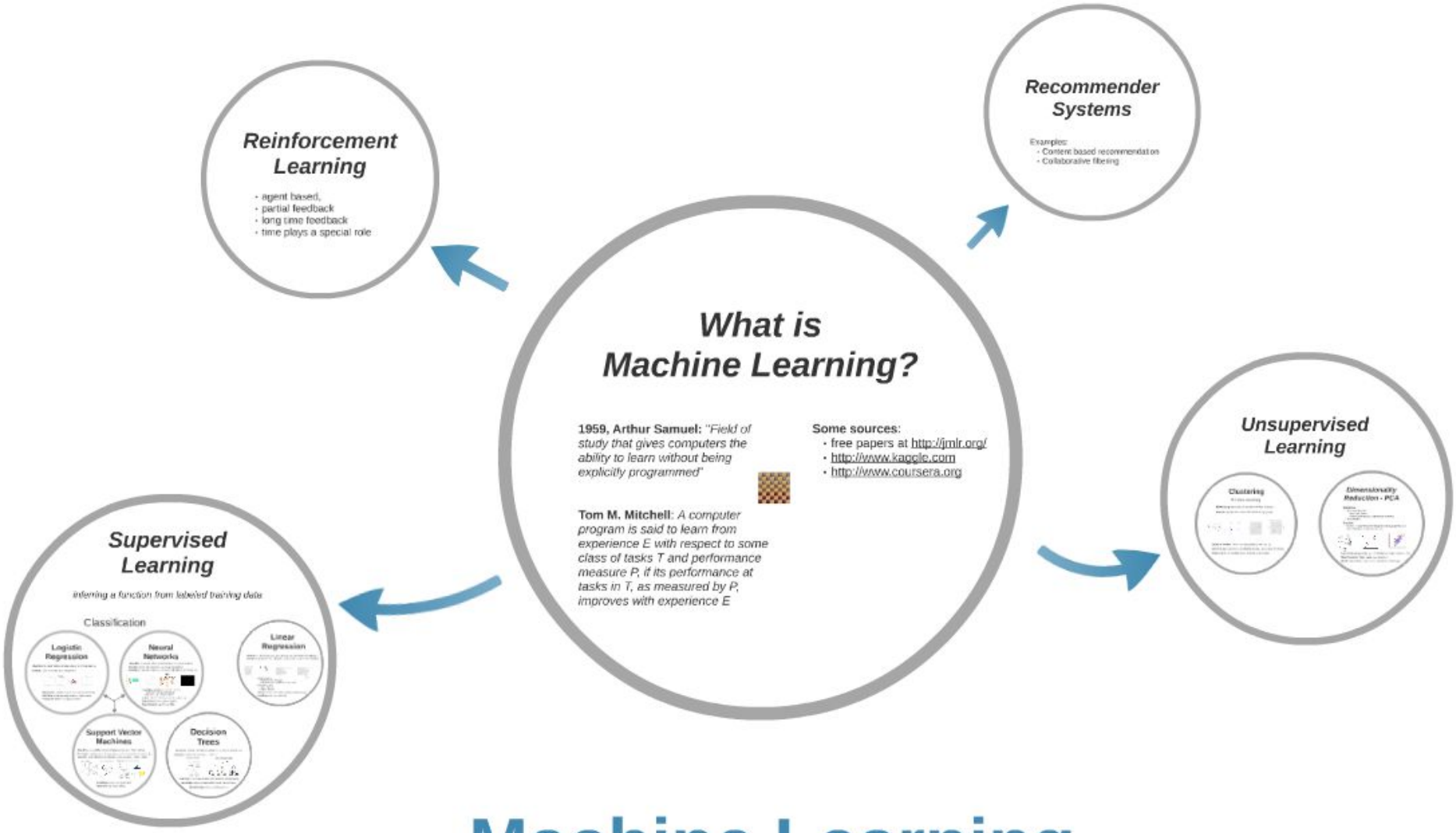
We reviewed a number of **shrinkage** parameters:

- **Ridge**, **Lasso**, **ElasticNet** for Linear Regression
- Number of neighbours in K Nearest Neighbour
- Trees: max_depth, min_split etc.

We studied **hyperparameter** calibration:

- Default parameters
- **Grid search**

https://prezi.com/06swcwazd0ai/machine-learning/



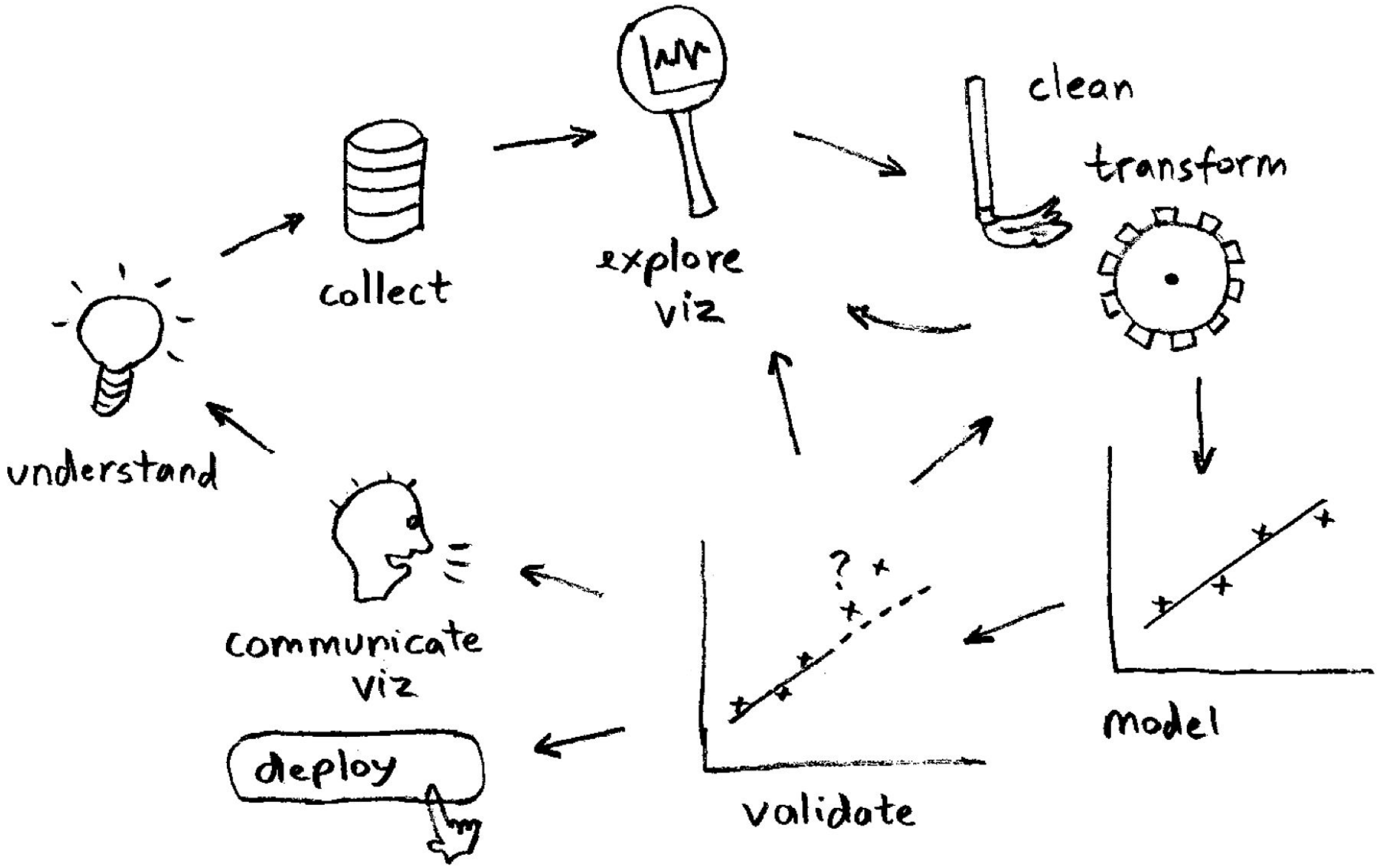
Machine Learning

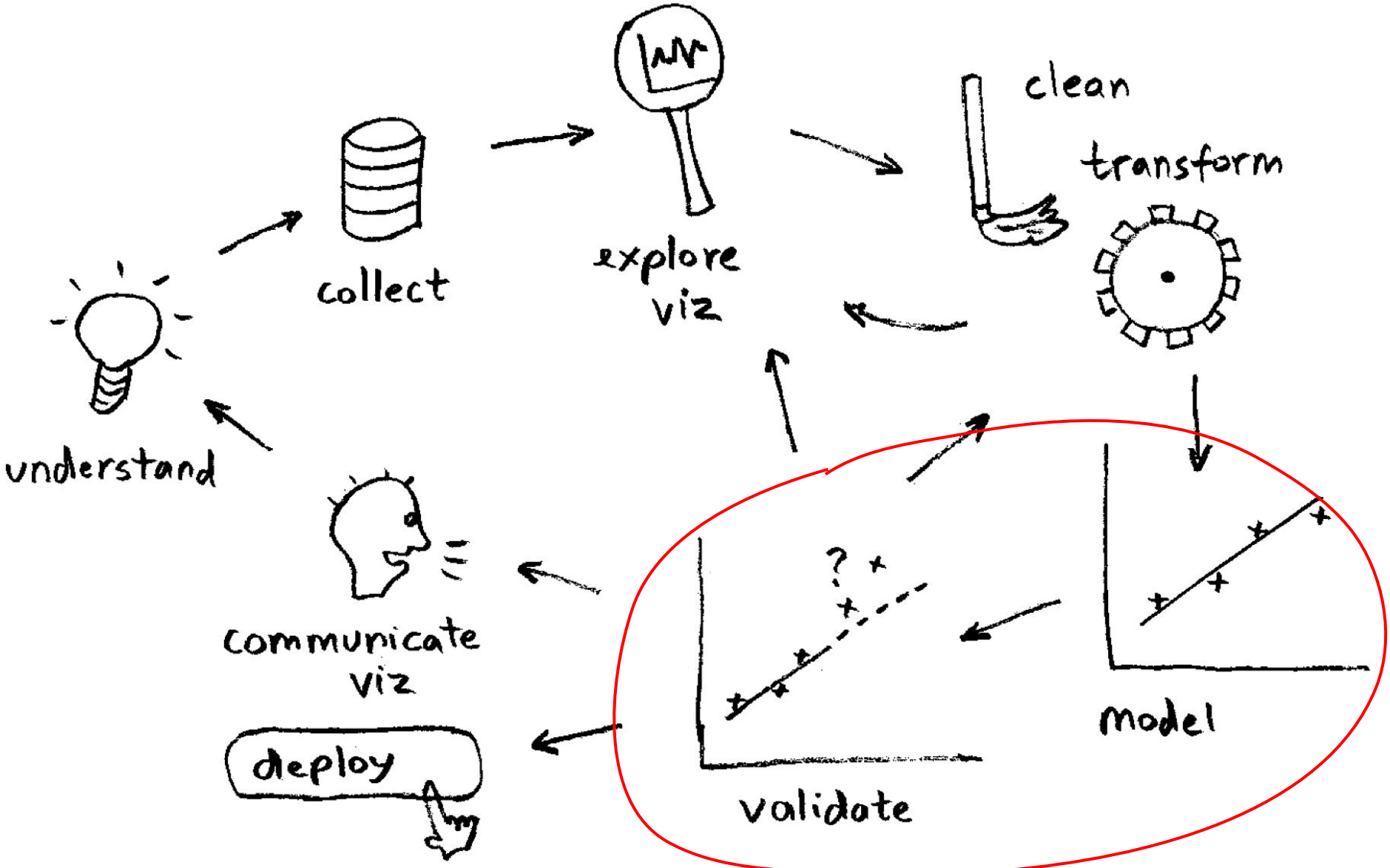
Unsupervised Learning

Supervised Vs unsupervised Learning

Learning by

Learning without


clean
transform
collect
explore
viz
understand
communicate
viz
deploy
validate
model


collect
explore
viz
clean
transform
understand
communicate
viz
deploy
validate
model

K-means Clustering

K-means Clustering

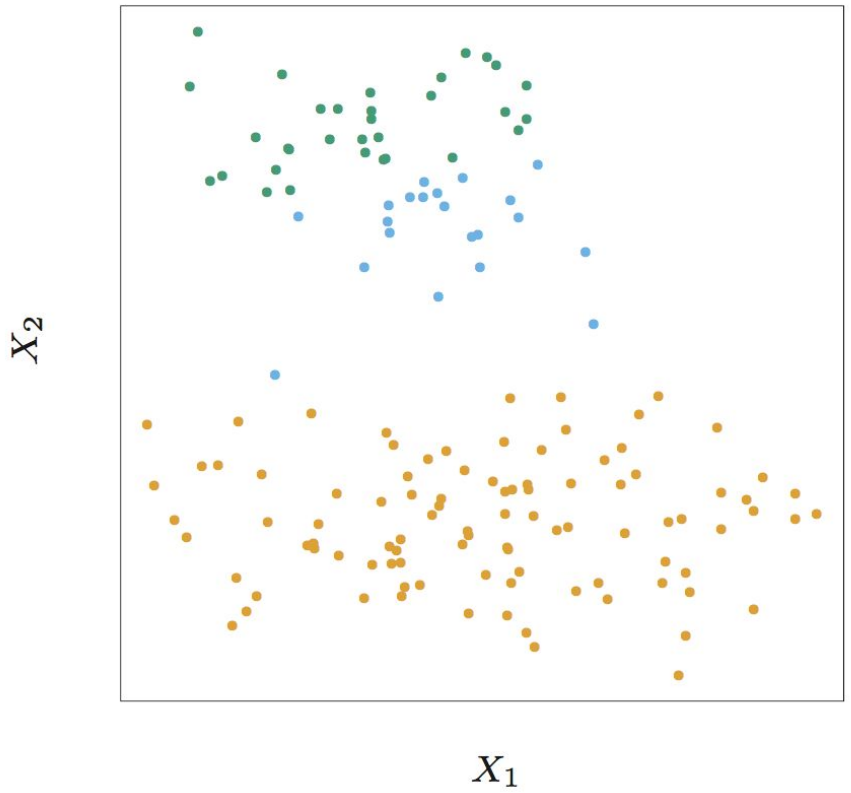
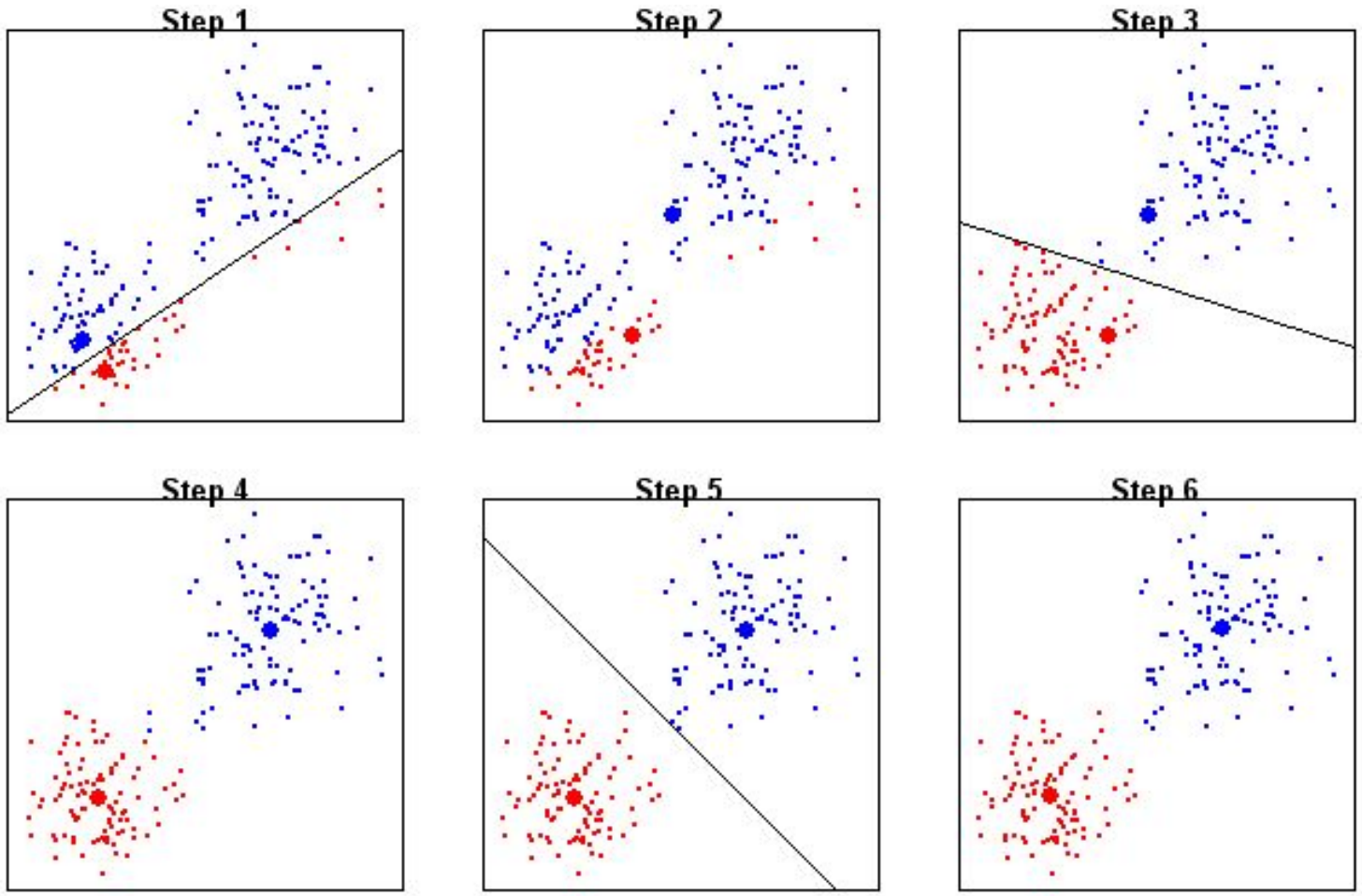


FIGURE 14.4. *Simulated data in the plane, clustered into three classes (represented by orange, blue and green) by the K-means clustering algorithm*

K-means Clustering: How does it work?



Some questions

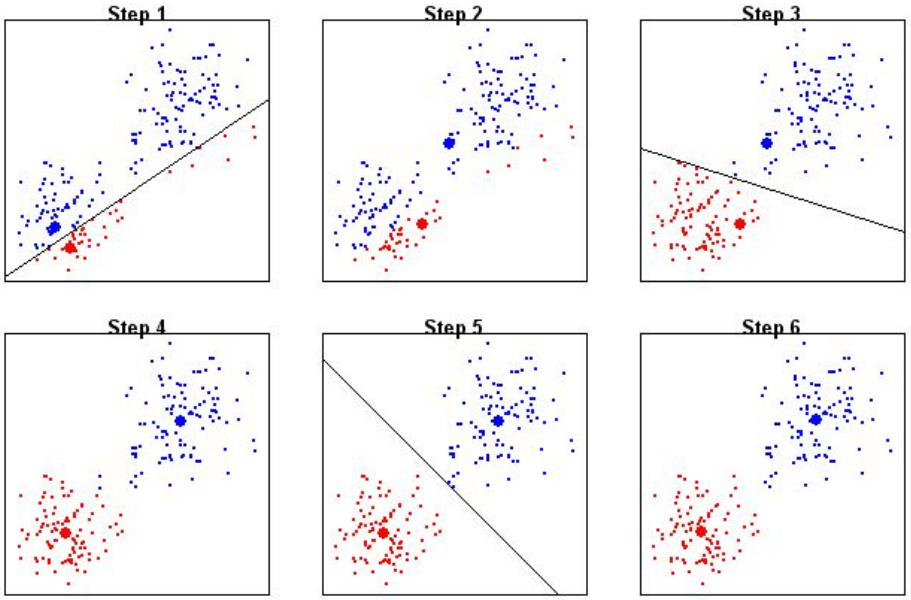
- When do we stop?
- Do clusters depend on initial cluster points?
- What can we do about this exposure?
- Which cluster to choose in the end?

Some answers

- When do we stop?
  When no data point changes cluster
- Do clusters depend on initial cluster points?
  Yes
- What can we do about this exposure?
  Repeat the algorithm multiple times
- Which cluster to choose in the end?
  The one with smallest Within-cluster point jitter

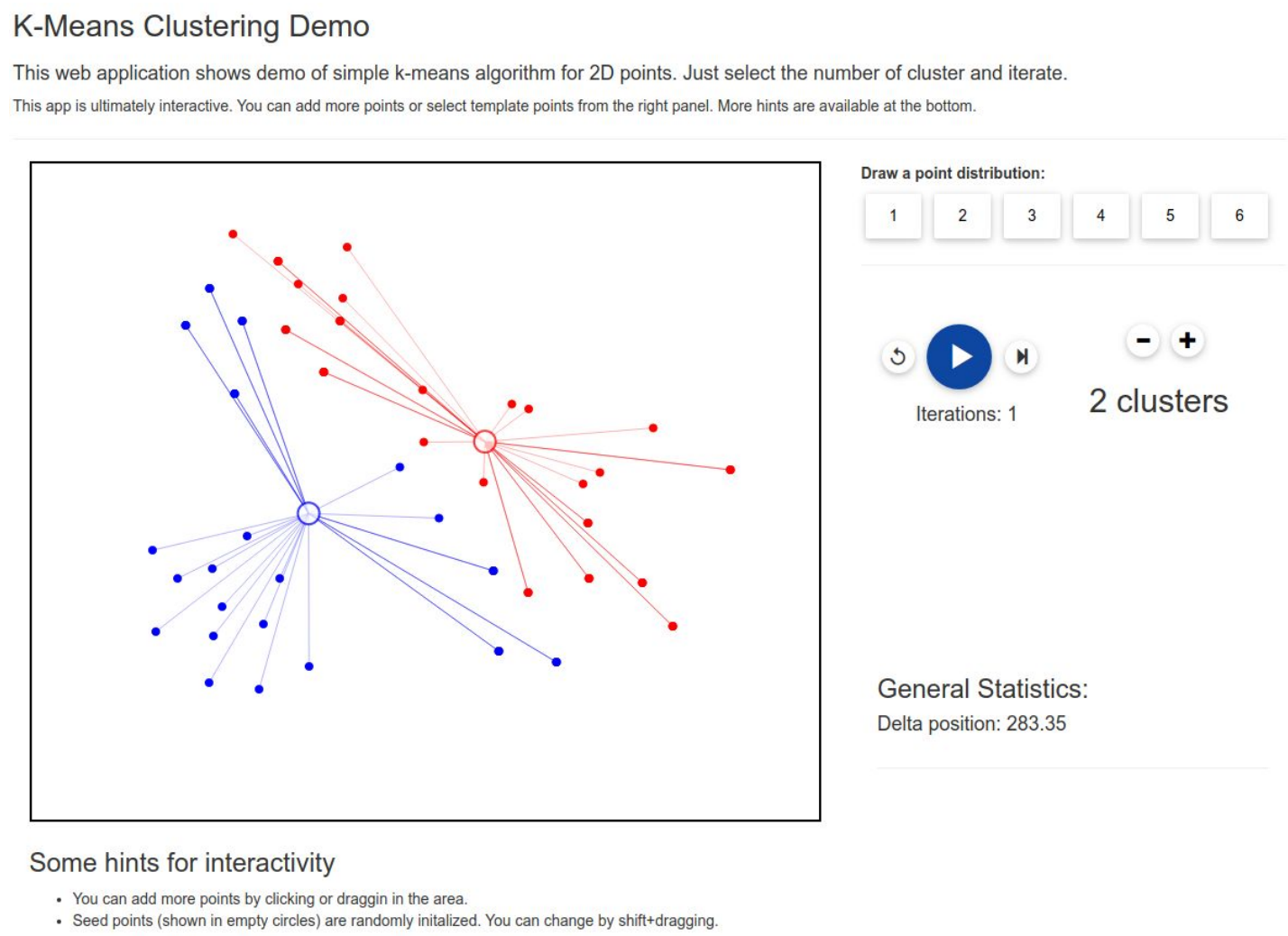
K-means Clustering: How does it work?

1. Take randomly or using a heuristic K data points. Let these be the initial cluster centers
2. Assign each datapoint from the dataset to the closest cluster
3. Update cluster center by calculating the new center of each cluster determined by its new members
4. Repeat from point 2, or stop if no changes were done



K-means Clustering: Online demo

https://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means/

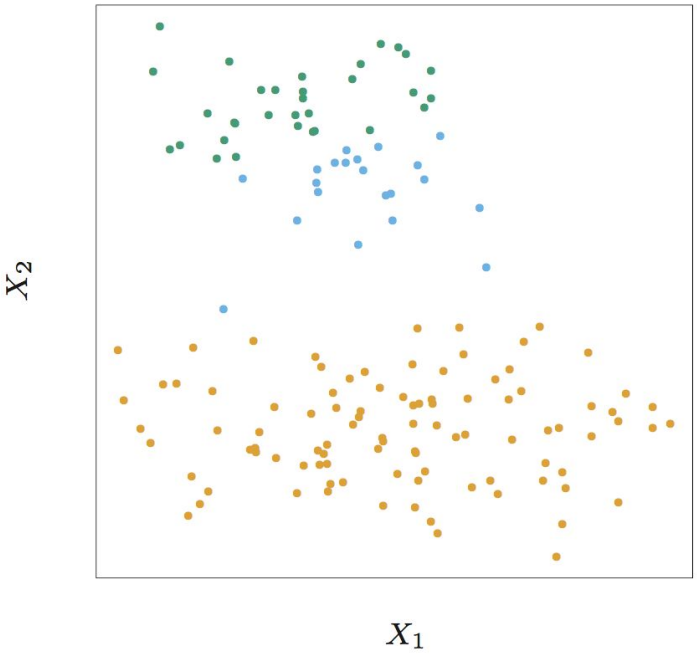


K-means Clustering - Concept of Distance

$$D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot d_j(x_{ij}, x_{i'j})$$

$$\bar{D} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot \bar{d}_j$$

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j})$$



$D(x_i, x_{i'})$: dissimilarity between objects i and i' and average dissimilarity

d_j: dissimilarity between two objects by attribute j and average dissimilarity

w_j: weight of j^{th} attribute

Dissimilarity types

Quantitative

$$D_I(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot (x_{ij} - x_{i'j})^2$$

Ordinal

For example: (e.g. A, B, C, D): take the order of elements for distance definition

Categorical

Example: 1 if (i == j), 0 otherwise

Within, Total, Between cluster point scatter

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k}||x_i - x_{i'}||^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k}||x_i - \bar{x}_k||^2,$$

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} d_{ii'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left(\sum_{C(i')=k} d_{ii'} + \sum_{C(i')\neq k} d_{ii'}\right)$$

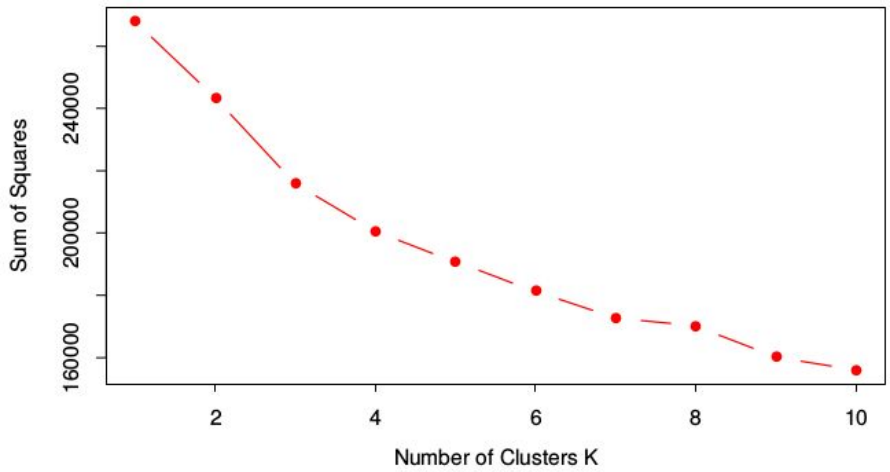
$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d_{ii'}$$

$$W(C) = T - B(C)$$

Minimizing W(C) is equivalent to maximizing B(C)

Human tumor microarray data clustering

Total within cluster sum of squares for K-means clustering applied to the human tumor microarray data (6830x64 matrix of real numbers) and the clusters at K = 3.



Cluster	Breast	CNS	Colon	K562	Leukemia	MCF7
1	3	5	0	0	0	0
2	2	0	0	2	6	2
3	2	0	7	0	0	0
Cluster	Melanoma	NSCLC	Ovarian	Prostate	Renal	Unknown
1	1	7	6	2	9	1
2	7	2	0	0	0	0
3	0	0	0	0	0	0

What is a good K value?
Gap Curve

Read ESLII 14.3.11 Practical Issues (page 518-519)

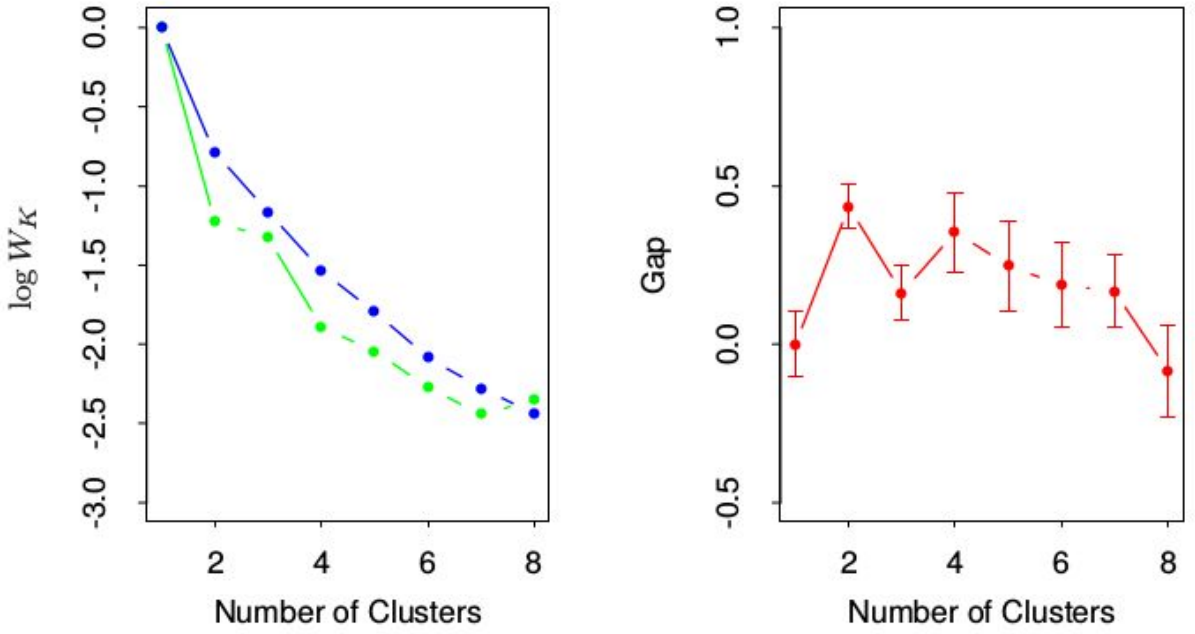


FIGURE 14.11. *(Left panel): observed (green) and expected (blue) values of* $\log W_K$ *for the simulated data of Figure 14.4. Both curves have been translated to equal zero at one cluster. (Right panel): Gap curve, equal to the difference between the observed and expected values of* $\log W_K$*. The Gap estimate* K^* *is the smallest* K *producing a gap within one standard deviation of the gap at* $K+1$*; here* $K^* = 2$.

NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set

	Index
1	CH (Calinski and Harabasz 1974)
2	CCC (Sarle 1983)
3	Pseudot2 (Duda and Hart 1973)
4	KL (Krzanowski and Lai 1988)
5	Gamma (Baker and Hubert 1975)
6	Gap (Tibshirani *et al.* 2001)
7	Silhouette (Rousseeuw 1987)
8	Hartigan (Hartigan 1975)
9	Cindex (Hubert and Levin 1976)
10	DB (Davies and Bouldin 1979)
11	Ratkowsky (Ratkowsky and Lance 1978)
12	Scott (Scott and Symons 1971)
13	Marriot (Marriot 1971)
14	Ball (Ball and Hall 1965)
15	Trcovw (Milligan and Cooper 1985)
16	Tracew (Milligan and Cooper 1985)
17	Friedman (Friedman and Rubin 1967)
18	Rubin (Friedman and Rubin 1967)
19	Dunn (Dunn 1974)

Image and signal compression
Vector Quantization

Read ESLII 14.3.9 Vector Quantization (page 514-515)

FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2 × 2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

K - Medoids

- Replace cluster centers by one data point
- Solve the minimization problem

$$\min_{C,\,\{i_k\}_1^K} \sum_{k=1}^{K} \sum_{C(i)=k} d_{ii_k}$$

Motivation: non-quantitative attributes (e.g. matrix below) or need for robustness

Proximity Matrix:

TABLE 14.3. *Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.*

	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS	YUG
BRA	5.58										
CHI	7.00	6.50									
CUB	7.08	7.00	3.83								
EGY	4.83	5.08	8.17	5.83							
FRA	2.17	5.75	6.67	6.92	4.92						
IND	6.42	5.00	5.58	6.00	4.67	6.42					
ISR	3.42	5.50	6.42	6.42	5.00	3.92	6.17				
USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17		
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92

Hierarchical Clustering

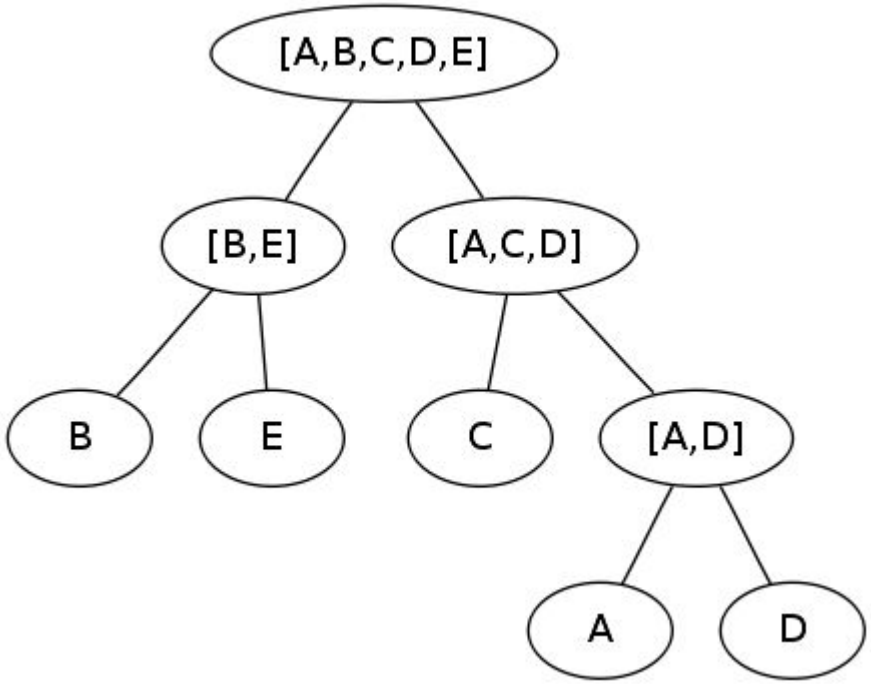
Agglomerative vs Divisive

Agglomerative: bottom to top

- Start with N clusters, with size 1
- Merge those two, that are closest to each other
- Continue this until we end up with 1 cluster

Divisive: top to bottom

- Start with one cluster
- Removing one element that has highest average dissimilarity to the group
- Migrate one by one elements to the new group, until there are no longer any observations in original group that are, on average, closer to those in the new group.
- Continue this with the child groups, choosing the cluster at each level with the largest diameter.



Average, Complete, Single Linkage

- **Average Linkage**

  Distance of two groups is the average

  $$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$

- **Complete Linkage**

  Distance of the two groups is the furthest two elements

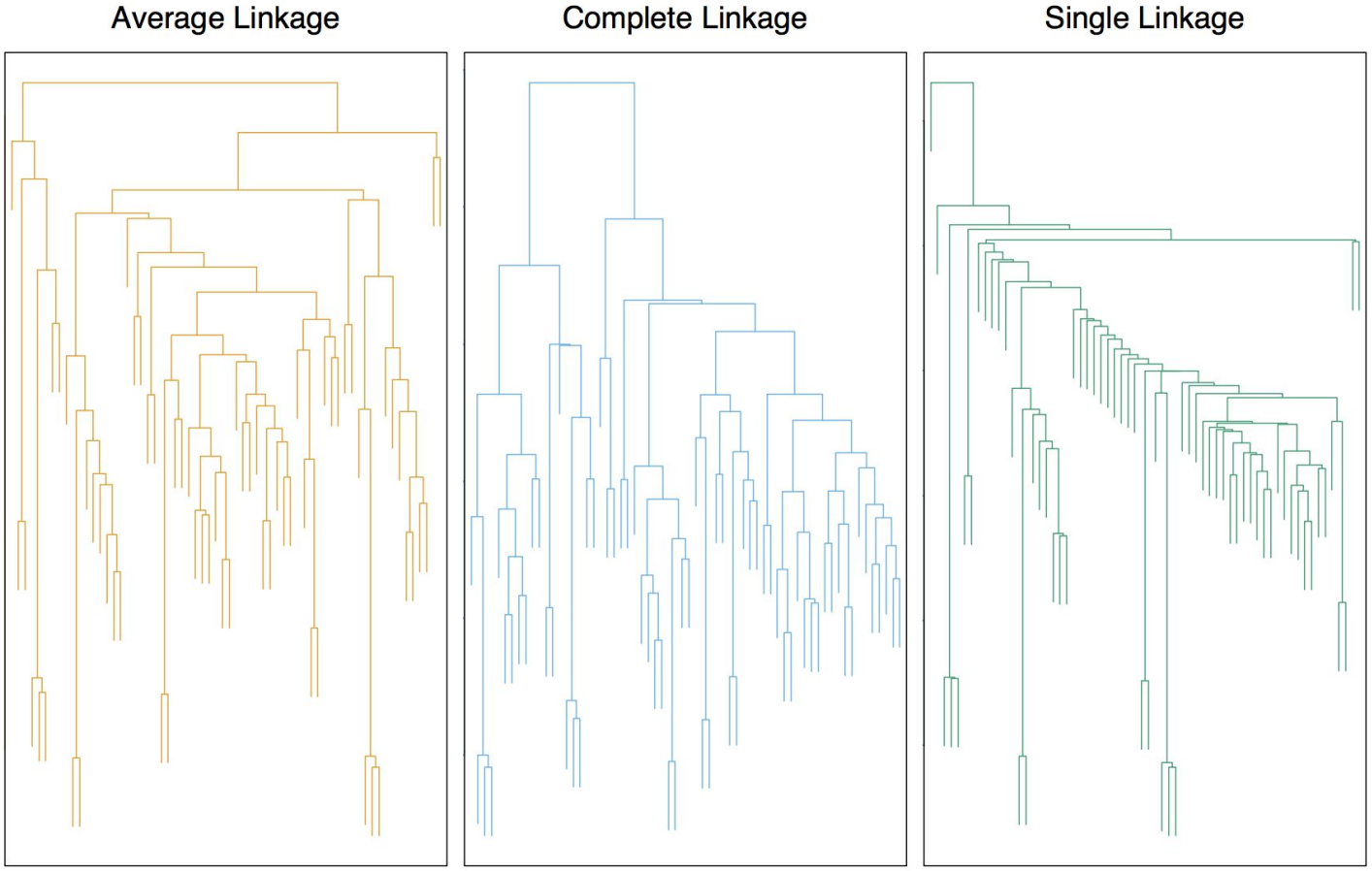
  $$d_{CL}(G,H) = \max_{\substack{i \in G \\ i' \in H}} d_{ii'}$$

- **Single Linkage**

  The distance of two groups is the closest two elements

  $$d_{SL}(G,H) = \min_{\substack{i \in G \\ i' \in H}} d_{ii'}$$

Average, Complete, Single Linkage



$$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$

$$d_{CL}(G,H) = \max_{\substack{i \in G \\ i' \in H}} d_{ii'}$$

$$d_{SL}(G,H) = \min_{\substack{i \in G \\ i' \in H}} d_{ii'}$$

Principal Component Analysis

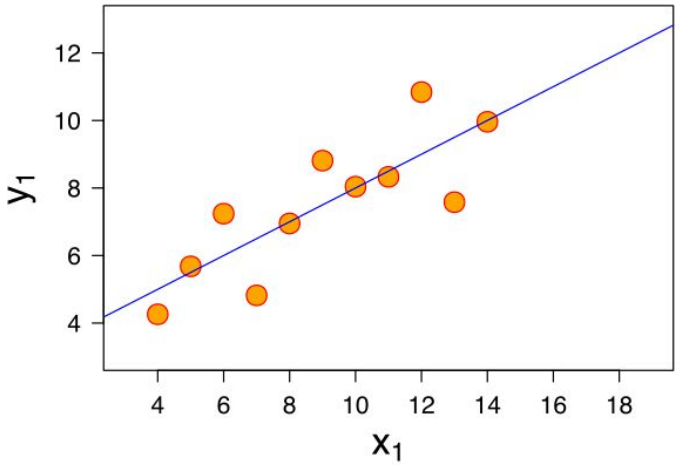
PCA: https://www.youtube.com/watch?v=TJdH6rPA-TI
Eigenvectors and eigenvalues: https://www.youtube.com/watch?v=PFDu9oVAE-g

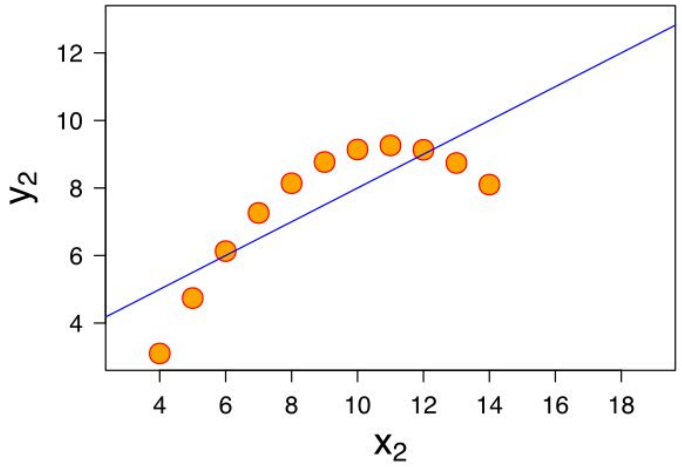
Why PCA?

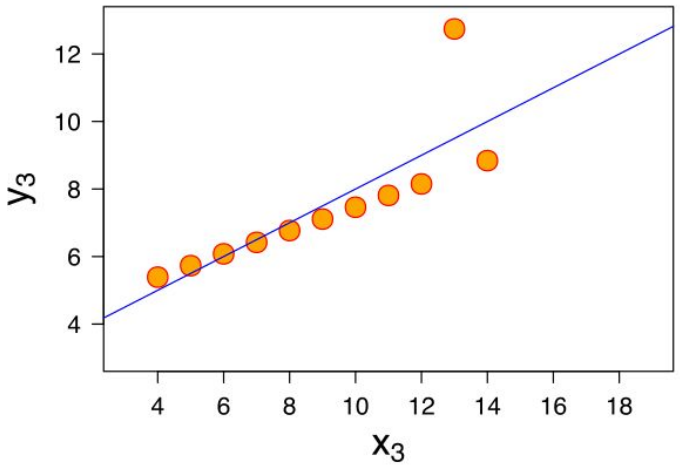
- One of the more-useful methods from applied linear algebra
- Non-parametric way of extracting meaningful information from confusing data sets
- Uncovers hidden, low-dimensional structures that underlie your data
- These structures are more-easily visualized and are often interpretable to content experts

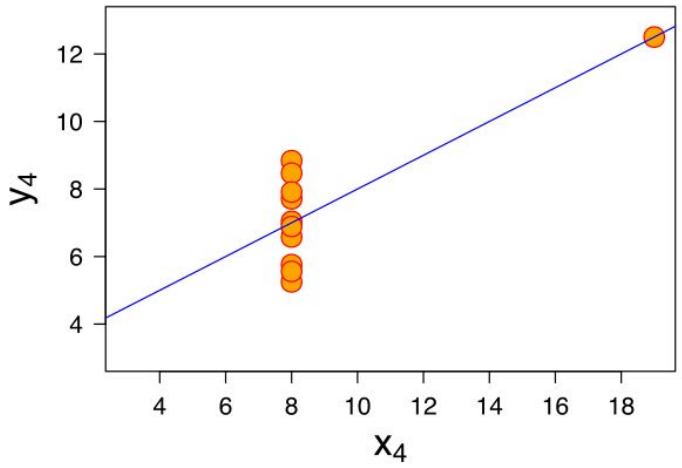
Source: https://towardsdatascience.com/introduction-to-principle-component-analysis-d705d27b88b6

Correlation of 0.816









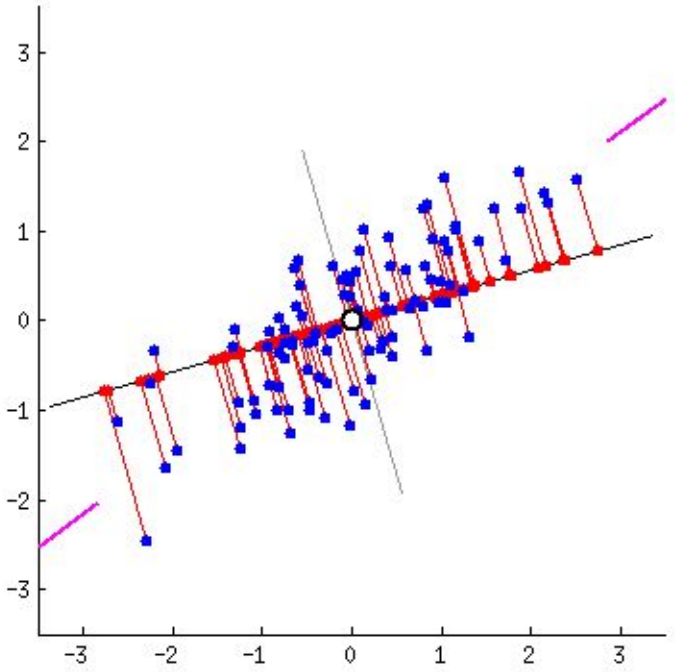
Covariance and Correlation

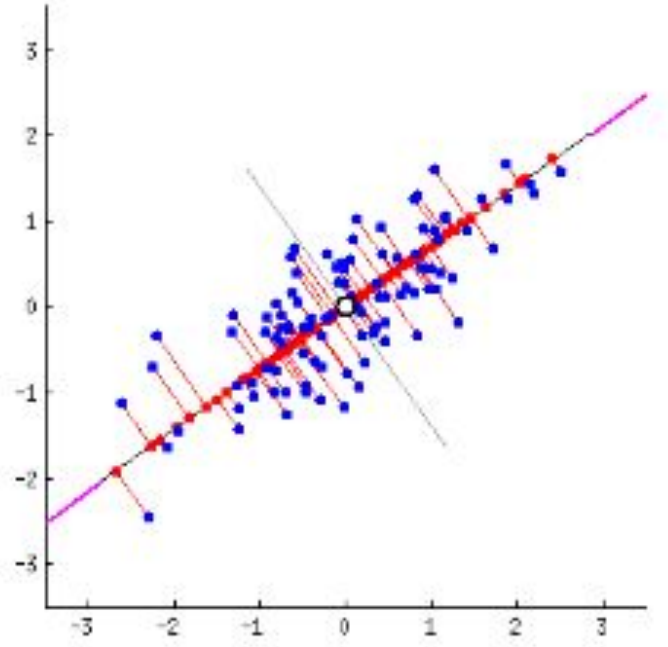
$$\operatorname{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - E(X))(y_i - E(Y))$$

$$\Sigma = \begin{bmatrix} \operatorname{Var}(X) & \operatorname{Cov}(X,Y) & \operatorname{Cov}(X,Z) \\ \operatorname{Cov}(X,Y) & \operatorname{Var}(Y) & \operatorname{Cov}(Y,Z) \\ \operatorname{Cov}(X,Z) & \operatorname{Cov}(Y,Z) & \operatorname{Var}(Z) \end{bmatrix}$$

Minimize distance from a subspace

Which is going to be the first principal component if we minimize distance?





3D to 2 principal components (2D)

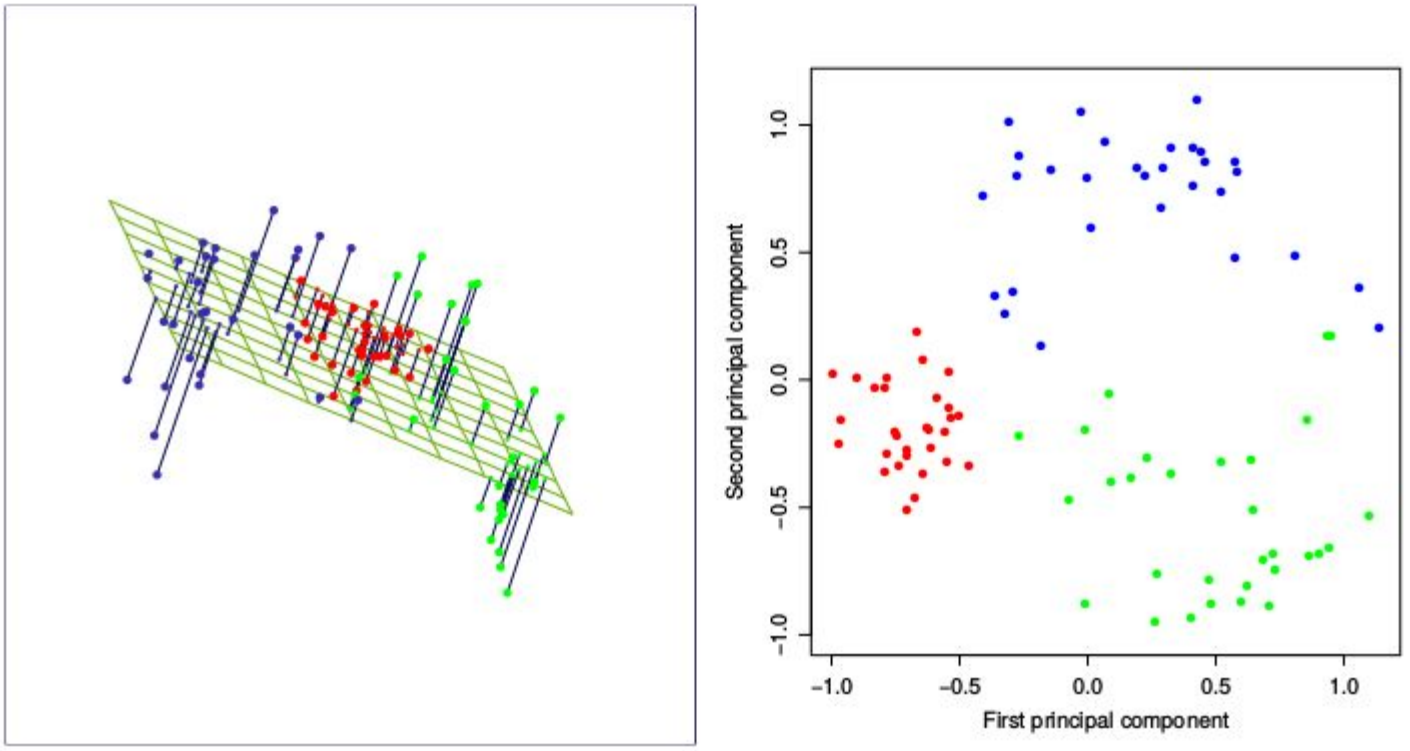
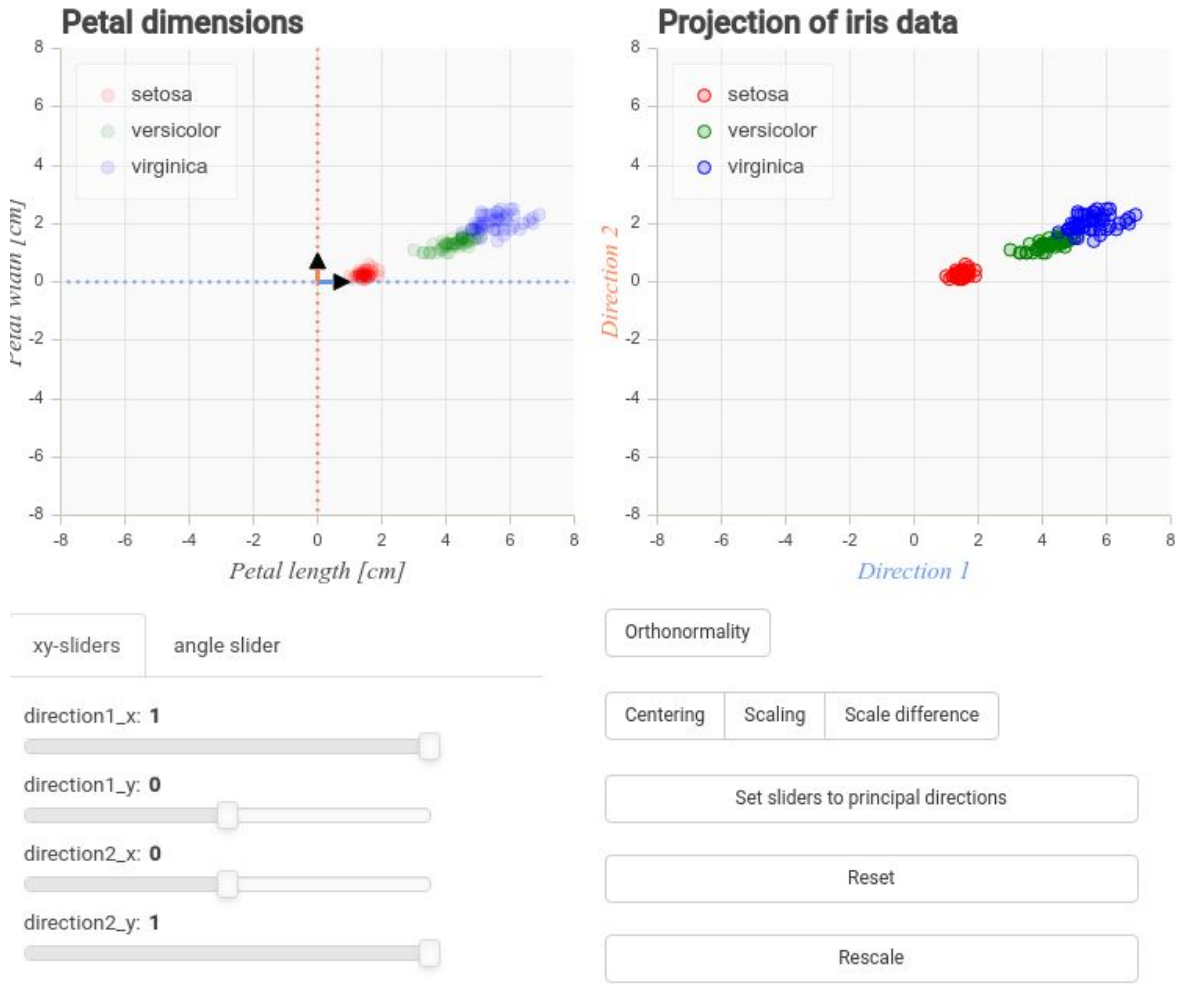


FIGURE 14.21. *The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by* $\mathbf{U}_2\mathbf{D}_2$*, the first two principal components of the data.*

PCA: Online demo

http://www2.imm.dtu.dk/courses/02450/DemoPCA.html



PCA - some theory

Principal components of a set of data in $\mathbf{R}^p$ provide a sequence of best linear approximations to that data, of all ranks $q \leq p$.

$x_1, x_2, \ldots, x_N$ are p dimensional observations from $\mathbf{R}^p$

$f(\lambda) = \mu + \mathbf{V}_q\lambda$ are the rank-q linear model representing X_i

Reconstruction error:

$$\min_{\mu, \{\lambda_i\}, \mathbf{V}_q} \sum_{i=1}^{N} \|x_i - \mu - \mathbf{V}_q\lambda_i\|^2$$

$$\begin{aligned} \hat{\mu} &= \bar{x}, \\ \hat{\lambda}_i &= \mathbf{V}_q^T(x_i - \bar{x}) \end{aligned}$$

Example - Handwritten digits

130 threes on 16x16 images

Although there are a possible of 256 principle components.

First two principal components:

- V1 - lengthening of the lower tail
- V2 - character thickness

Approximately 50 principal components account for 90% of the variations.

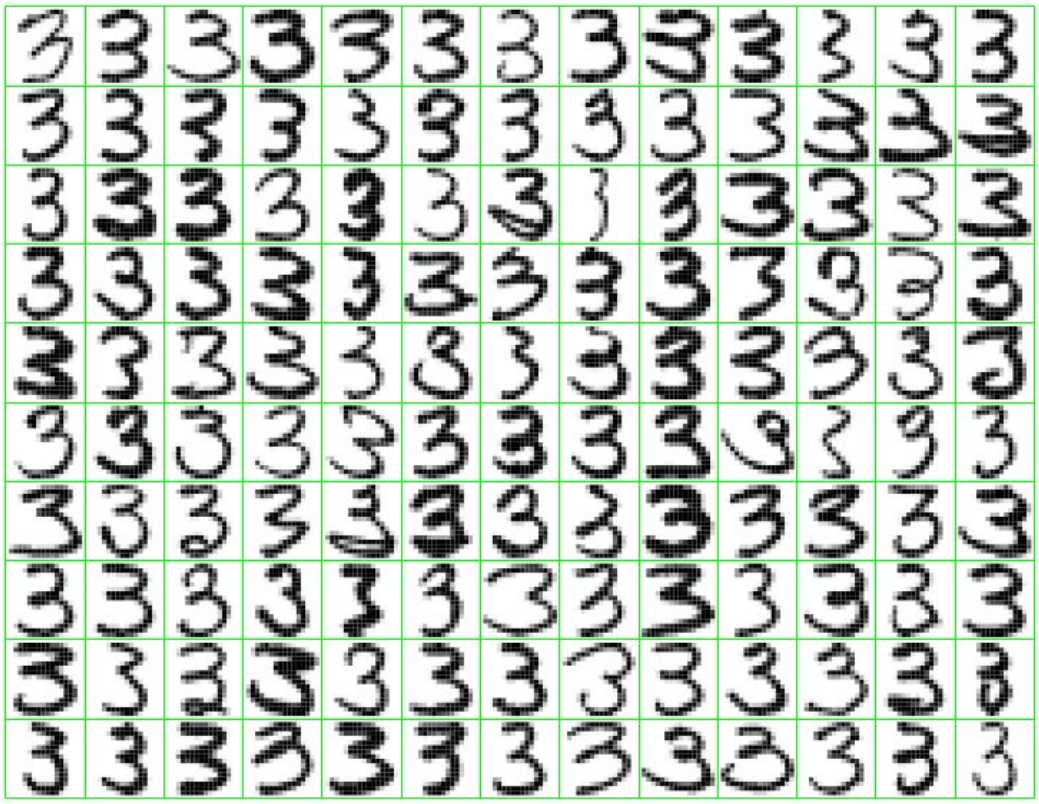


FIGURE 14.22. *A sample of* 130 *handwritten 3's shows a variety of writing styles.*

$$\hat{f}(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2$$

$$= \boxed{3} + \lambda_1 \cdot \boxed{3} + \lambda_2 \cdot \boxed{3}$$

First two principal components visualised

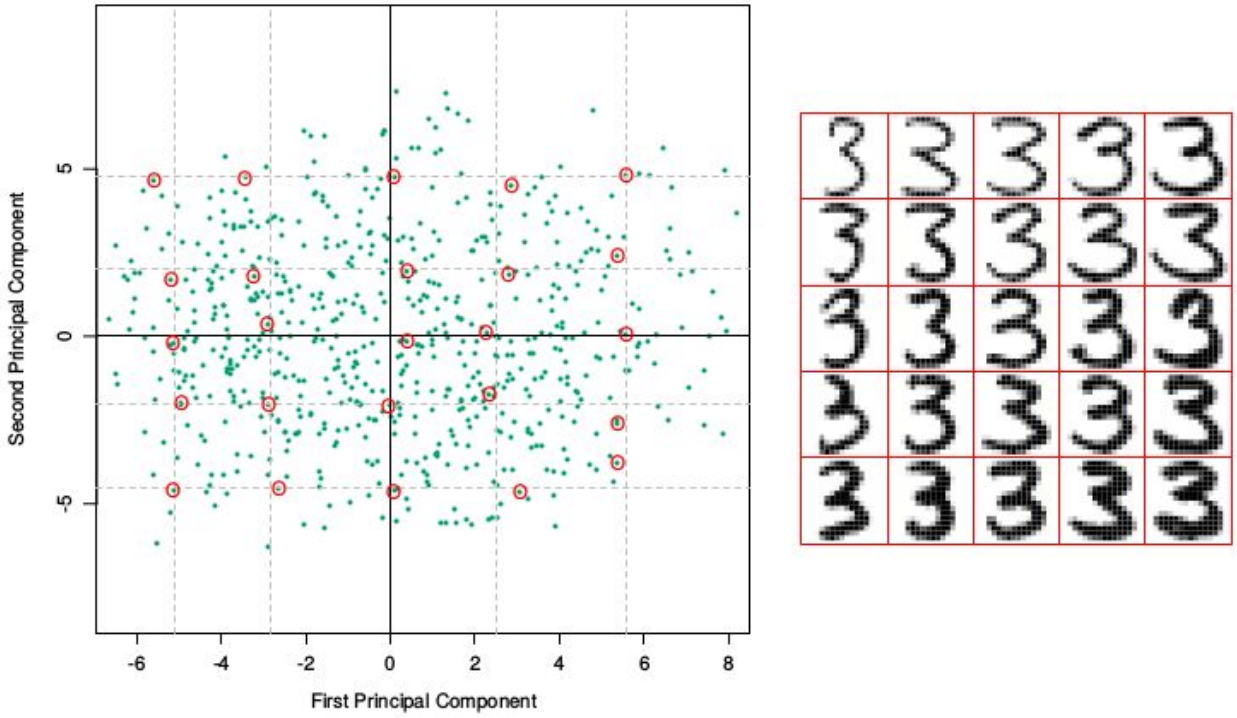


FIGURE 14.23. *(Left panel:) the first two principal components of the hand-written threes. The circled points are the closest projected images to the vertices of a grid, defined by the marginal quantiles of the principal components. (Right panel:) The images corresponding to the circled points. These show the nature of the first two principal components.*

Independent Component Analysis

Read ESLII 14.7.2 Independent Component Analysis(page 557-565)

Instead of the uncorrelated components (used in PCA), ICA is looking for statistical independence.

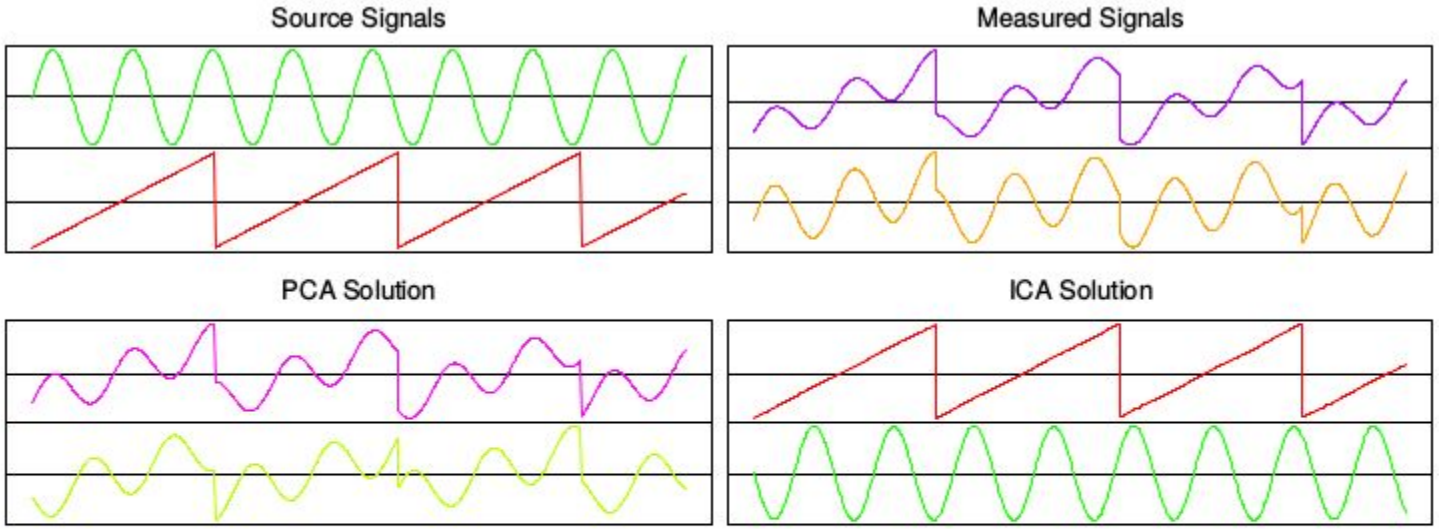


FIGURE 14.37. *Illustration of ICA vs. PCA on artificial time-series data. The upper left panel shows the two source signals, measured at* 1000 *uniformly spaced time points. The upper right panel shows the observed mixed signals. The lower two panels show the principal components and independent component solutions.*

Independent Component Analysis

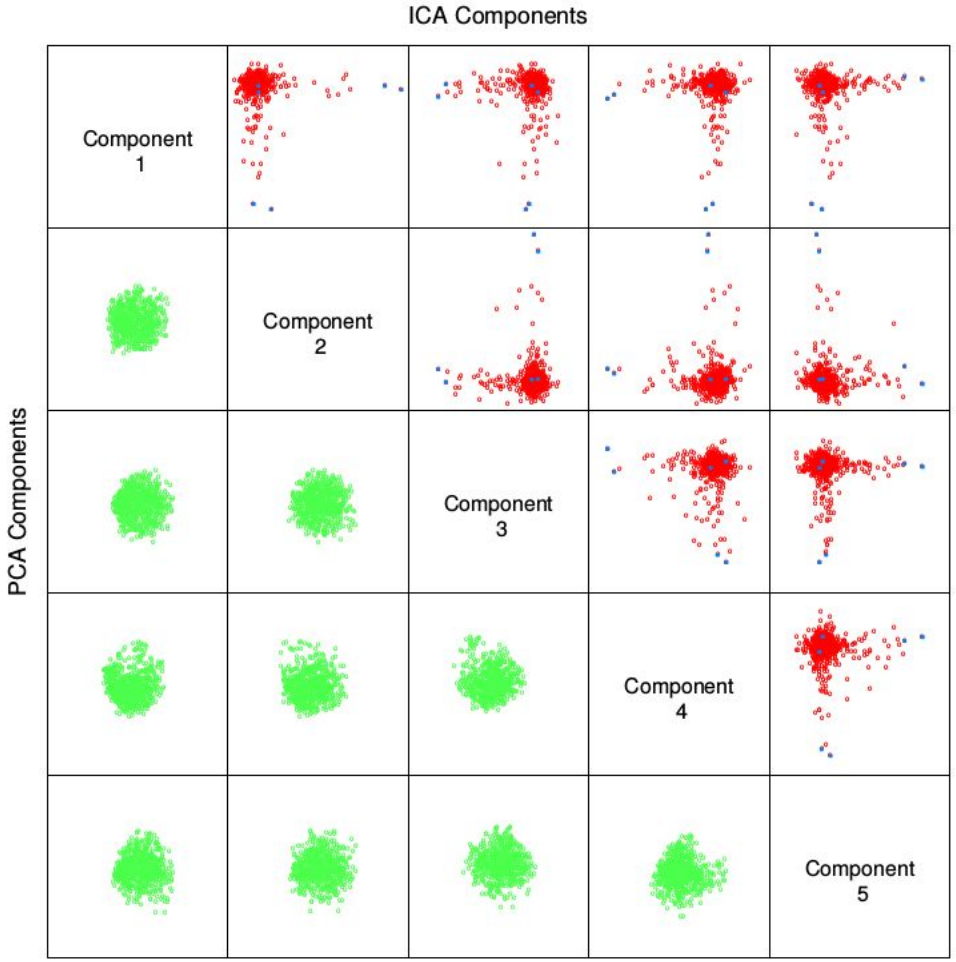


FIGURE 14.39. *A comparison of the first five ICA components computed using* `FastICA` *(above diagonal) with the first five PCA components(below diagonal). Each component is standardized to have unit variance.*

ICA vs PCA

PCA

- provides a reduced-rank representation of data
- helps to compress data

ICA

- provides a representation of the data as independent sub-elements
- helps to separate data

PCA for Supervised Learning

Model Prediction Power

Regression

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Mean Squared Error

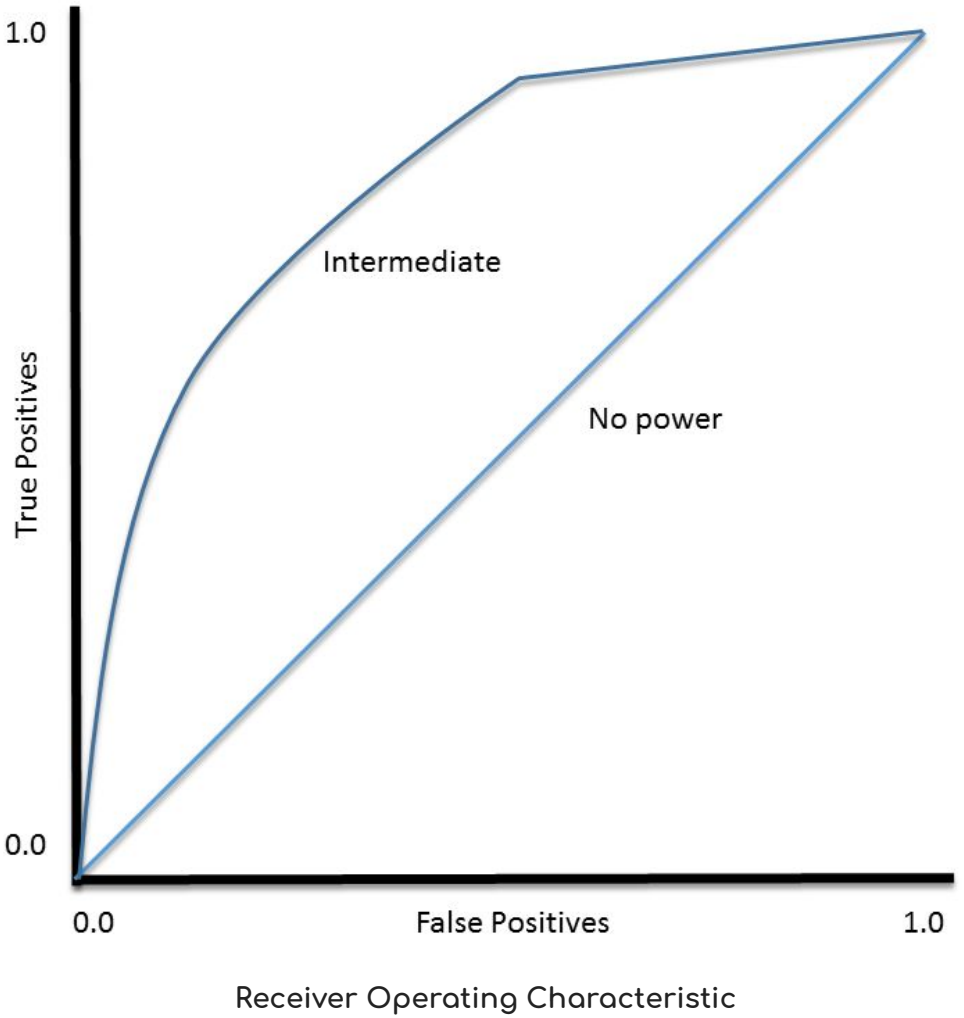
Classification

		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

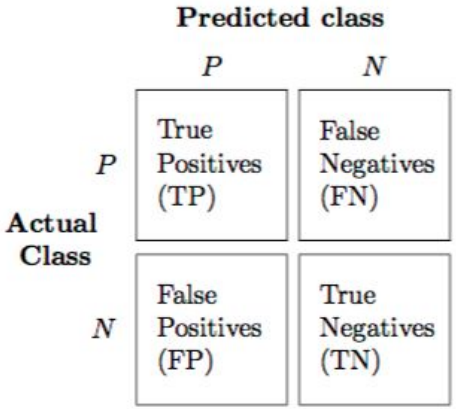
Confusion Matrix

Classification ROC / AUC

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)



Classification ROC / AUC



Sensitivity: is the true positive also called recall

Specificity: is the true negative rate

Accuracy: is the percentage of correctly classified instances out of all instances

Kappa: accuracy normalized at the baseline of random chance on dataset:

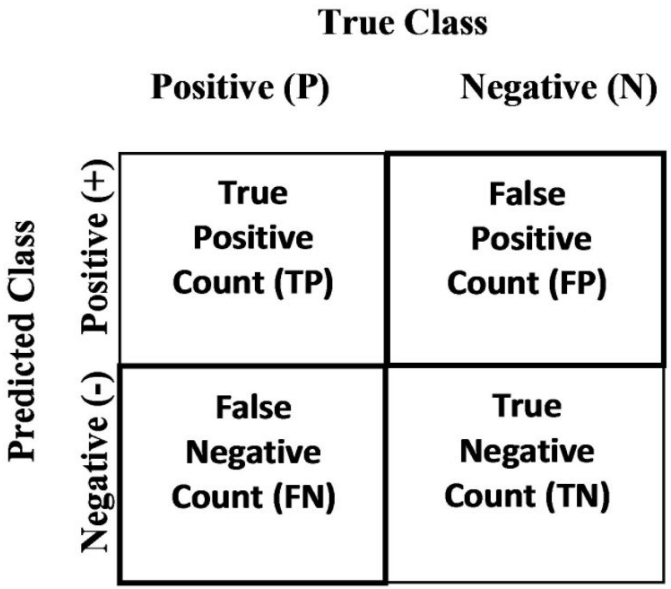
$$(p_0 - p_e) / (1 - p_e)$$

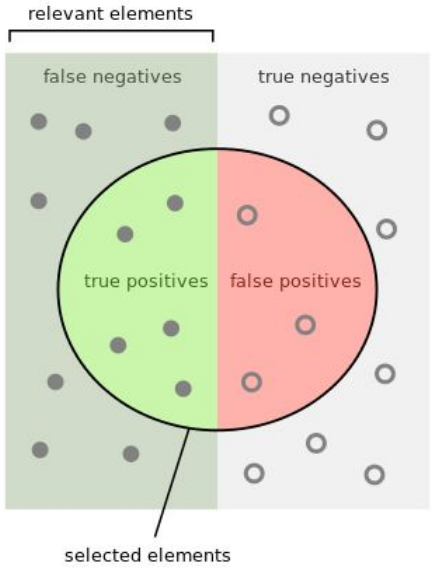
Where p_0 is accuracy and p_e is the hypothetical probability of chance agreement

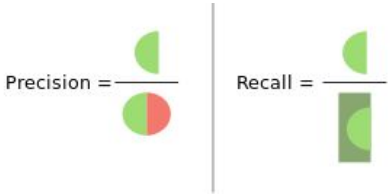
We need a **threshold** for predicted probabilities: if **x > threshold** then Positive else Negative

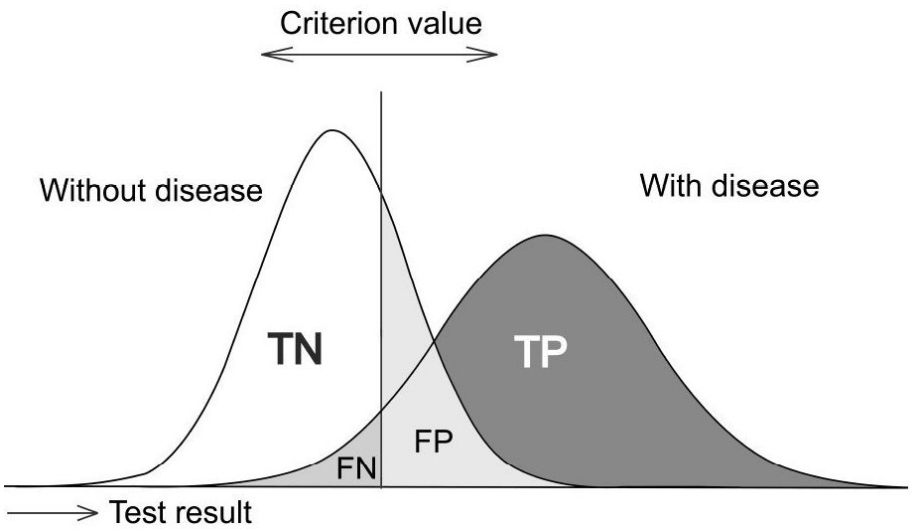
```
 [1] 0.6807279 0.8032791 0.9616134 0.4459540 0.7059463 0.7450944 0.5721412 0.9668678 0.9999920
[10] 0.9999338 0.6248030 0.1692712 0.8579875 0.1692712 0.9197069 0.9998928 0.7389576 0.9969105
[19] 0.9728985 0.5847868 0.9959735 0.9993547 0.9118485 0.9999315 0.7495186 0.8684869 0.9565866
[28] 0.6237776 0.9530365 0.9591537 0.1960257
```

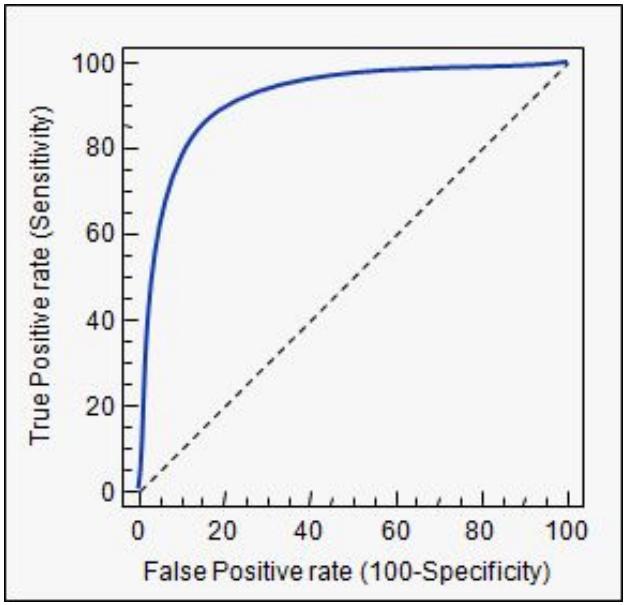
using threshold = 0.9


True Class
Positive (P)
Negative (N)
Predicted Class
Positive (+)
Negative (-)
True Positive Count (TP)
False Positive Count (FP)
False Negative Count (FN)
True Negative Count (TN)


relevant elements
false negatives
true negatives
true positives
false positives
selected elements


Precision =
Recall =


Criterion value
Without disease
With disease
TN
TP
FP
FN
Test result


True Positive rate (Sensitivity)
False Positive rate (100-Specificity)
0
20
40
60
80
100

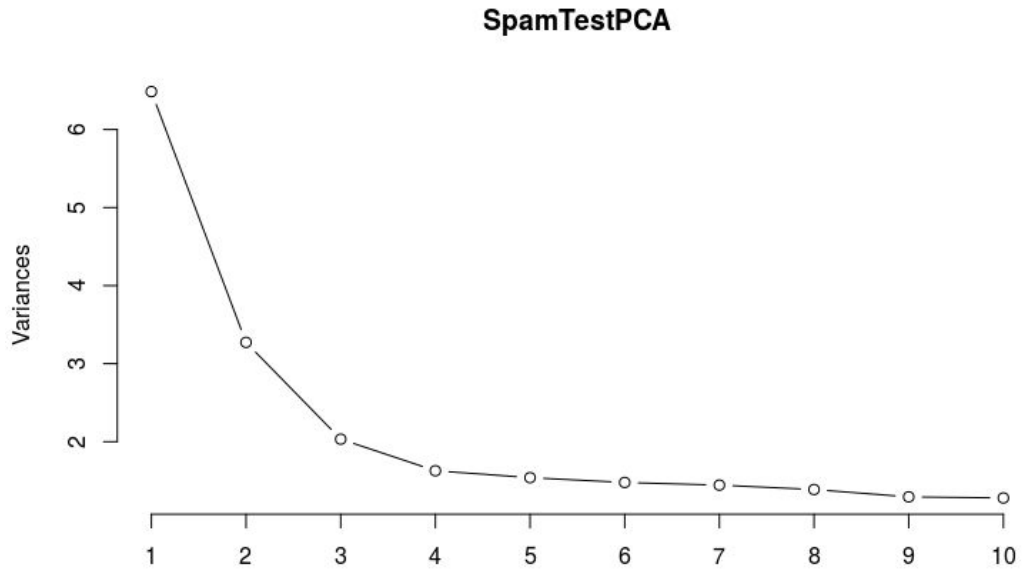
Classification of Spam Data

```
> colnames(SpamTrain, 2)
 [1] "word_freq_make"              "word_freq_address"           "word_freq_all"
 [4] "word_freq_3d"                "word_freq_our"               "word_freq_over"
 [7] "word_freq_remove"            "word_freq_internet"          "word_freq_order"
[10] "word_freq_mail"              "word_freq_receive"           "word_freq_will"
[13] "word_freq_people"            "word_freq_report"            "word_freq_addresses"
[16] "word_freq_free"              "word_freq_business"          "word_freq_email"
[19] "word_freq_you"               "word_freq_credit"            "word_freq_your"
[22] "word_freq_font"              "word_freq_000"               "word_freq_money"
[25] "word_freq_hp"                "word_freq_hpl"               "word_freq_george"
[28] "word_freq_650"               "word_freq_lab"               "word_freq_labs"
[31] "word_freq_telnet"            "word_freq_857"               "word_freq_data"
[34] "word_freq_415"               "word_freq_85"                "word_freq_technology"
[37] "word_freq_1999"              "word_freq_parts"             "word_freq_pm"
[40] "word_freq_direct"            "word_freq_cs"                "word_freq_meeting"
[43] "word_freq_original"          "word_freq_project"           "word_freq_re"
[46] "word_freq_edu"               "word_freq_table"             "word_freq_conference"
[49] "char_freq_semicolon"         "char_freq_parenth"           "char_freq_brack"
[52] "char_freq_excl"              "char_freq_dollar"            "char_freq_hash"
[55] "capital_run_length_average" "capital_run_length_longest" "capital_run_length_total"
[58] "spam"
```

Coding Exercise: classification

Exercise 1: try all known models on spam classification.
Start with: `ml.1.4/lect/classification_and_pca.R`
Do a PCA for spam data. Test spam classification with and without PCA.
Conclusions?

Principal Component Variance for Spam data




```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
PC9
Standard deviation     2.5467 1.80890 1.42543 1.27561 1.24043 1.21531 1.20138 1.17758
1.13682
Proportion of Variance 0.1138 0.05741 0.03565 0.02855 0.02699 0.02591 0.02532 0.02433
0.02267
Cumulative Proportion  0.1138 0.17119 0.20684 0.23538 0.26238 0.28829 0.31361 0.33794
```

Some principles in research and industry

Some principles I found useful

- Have a **goal** and formalize it
- Study **existing solutions**
- Study the **literature**
- Make a **plan** with a timeline
- **Log** your progress and results
- **Share** your thoughts and results, test them with others. The majority of the ideas look good only until you say them out loud or discuss, test them with others
- Be your greatest **critic** when it comes to fitting vs overfitting
- **Cut** unsuccessful attempts and directions, even if a lot of work was put into them
- **Derive conclusions**, summarize results. Research can be left open forever.
- **Analyze mistakes** and make a plan how to avoid them in the future
- **Close** the project
- Make it **reusable** for others
- There is usually a long way to go from a model that works to a model in **production**

Paper Review

Review of ML outside and within finance

Success stories of ML

- Playing 44 Million chess games and learning without hardcoded openings or strategies
- Crowdsourced traffic monitoring
- Kaggle (over half a million data scientists)
- CrowdAnalytics

Hedge fund attempts to do crowdsourcing:

- Quantopian:
  - Stock data analysis and testing platform
  - 100000 - 140000 data scientists
- Quantiacs
  - commodity futures
  - 6000 quants
  - Apparently 15M invested in crowdsourced algorithms
- Numerai - Richard Craib - ethereum based crypto compensation
  - Claim to have 30000 data scientists involved in their company
  - Completely sanitized and encrypted data
  - Global equity trading strategy

Is ML suitable for finance?

How to address non-stationary data in finance?

Is statistical regularity going away after running a strategy on it? (e.g. mean reversions)

How to deal with overfitting?

How to deal with the high number of tested algorithms and how to separate accuracy from luck?

Summary

We reviewed a number of **Unsupervised Machine Learning Models** and their **R package**:

- **K-means clustering**: algorithm, distance metrics, techniques
- **K-medoids clustering**: algorithm
- **Hierarchical clustering:** algorithm, agglomerative, divisive
- **Principal Component Analysis**: the algorithm, objectives
- **ICA**: difference to PCA

We looked at applications of **PCA for Supervised Learning**

- **PCA for Ridge**, **Lasso**, **ElasticNet**
- PCA for K Nearest Neighbour
- PCA for trees

We collected a number of principles for research

Summary of ML#1

We reviewed a number of **Machine Learning Concepts** for Supervised Learning:

- Model **training**
- **Train**, **Validation**, **Test** sets.
- **Train** and **Test Error**
- **Overfitting**, **Bias vs. Variance**
- **Complexity** control

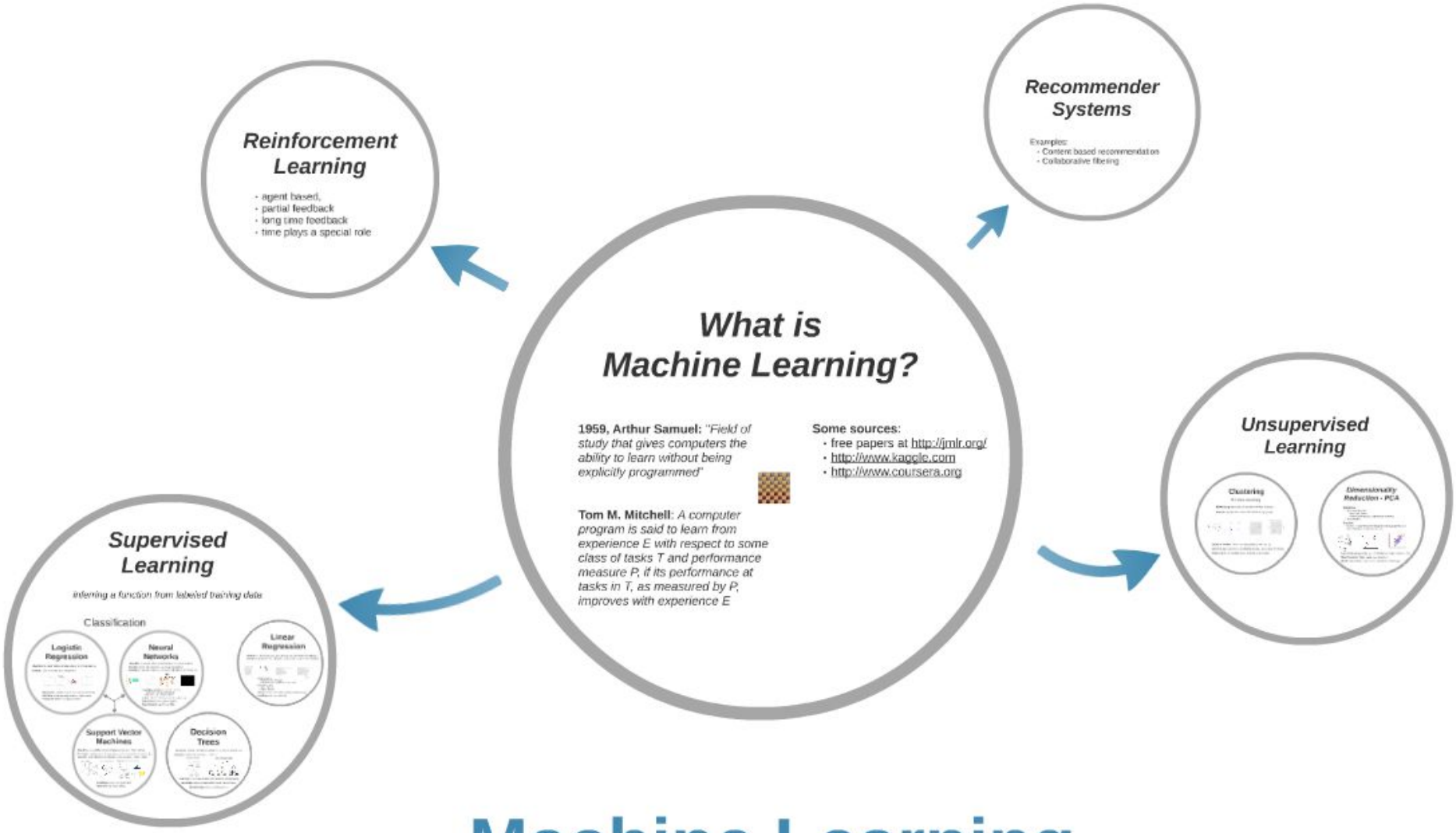
Supervised Learning Models and their complexity controls:

- Manual model
- **Linear Models**: Lasso and Ridge
- **Nearest Neighbour**
- **Decision Trees**

For **Unsupervised Learning** we discussed:

- **Clustering**: K-means clustering, K-medoids clustering, Hierarchical clustering
- **Principal Component Analysis (PCA)** and Independent Component Analysis (ICA)

https://prezi.com/06swcwazd0ai/machine-learning/



Machine Learning