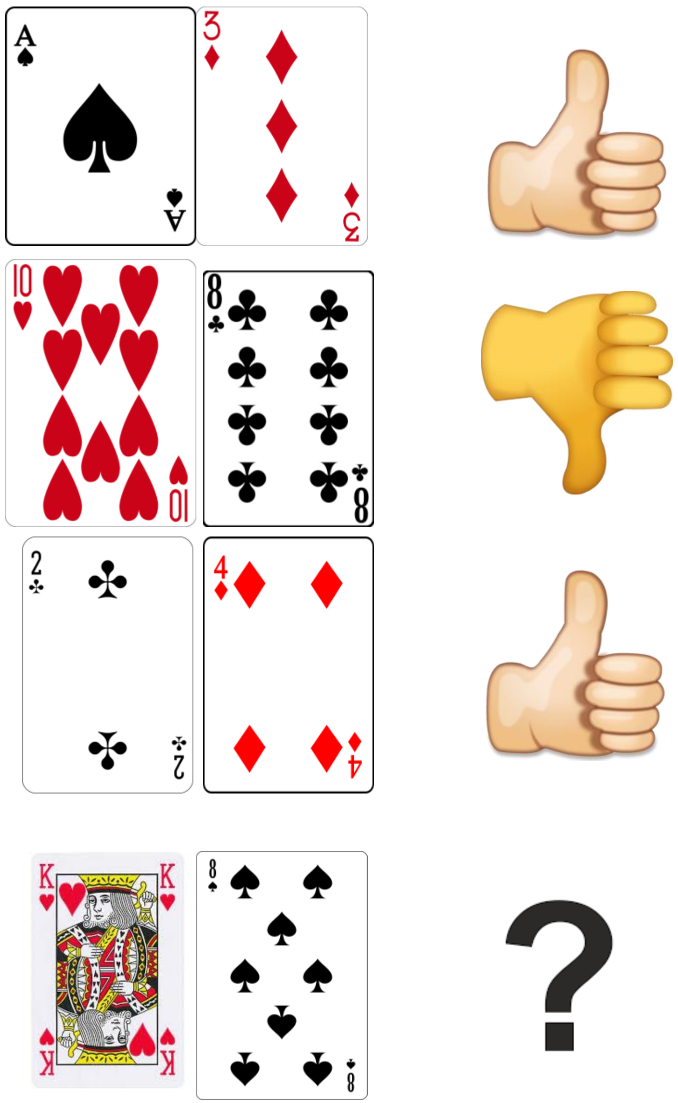
Modul 3

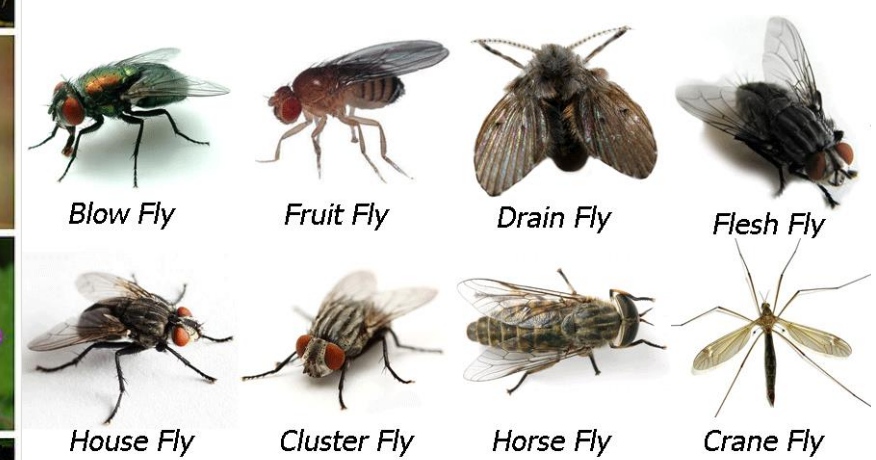
Unsupervised Learning

Data Science Program

Purwadhika
Startup and Coding School

Supervised vs. unsupervised learning





Recap

- Regression vs. classification?
- What algorithms have been learned?
- How do they compare?
- What are key issues with classification and regression?

Clustering intuitively

Purwadhika
Startup and Coding School

Clustering can be personal to me

Purchase PDF   Export

ELSEVIER

Pattern Recognition

Volume 40, Issue 12, December 2007, Pages 3452-3466

Possibilistic fuzzy co-clustering of large document collections

William-Chandra Tjhi, Lihui Chen

Show more

https://doi.org/10.1016/j.patcog.2007.04.017

Get rights and content

Purwadhika
Startup and Coding School

Clustering applications

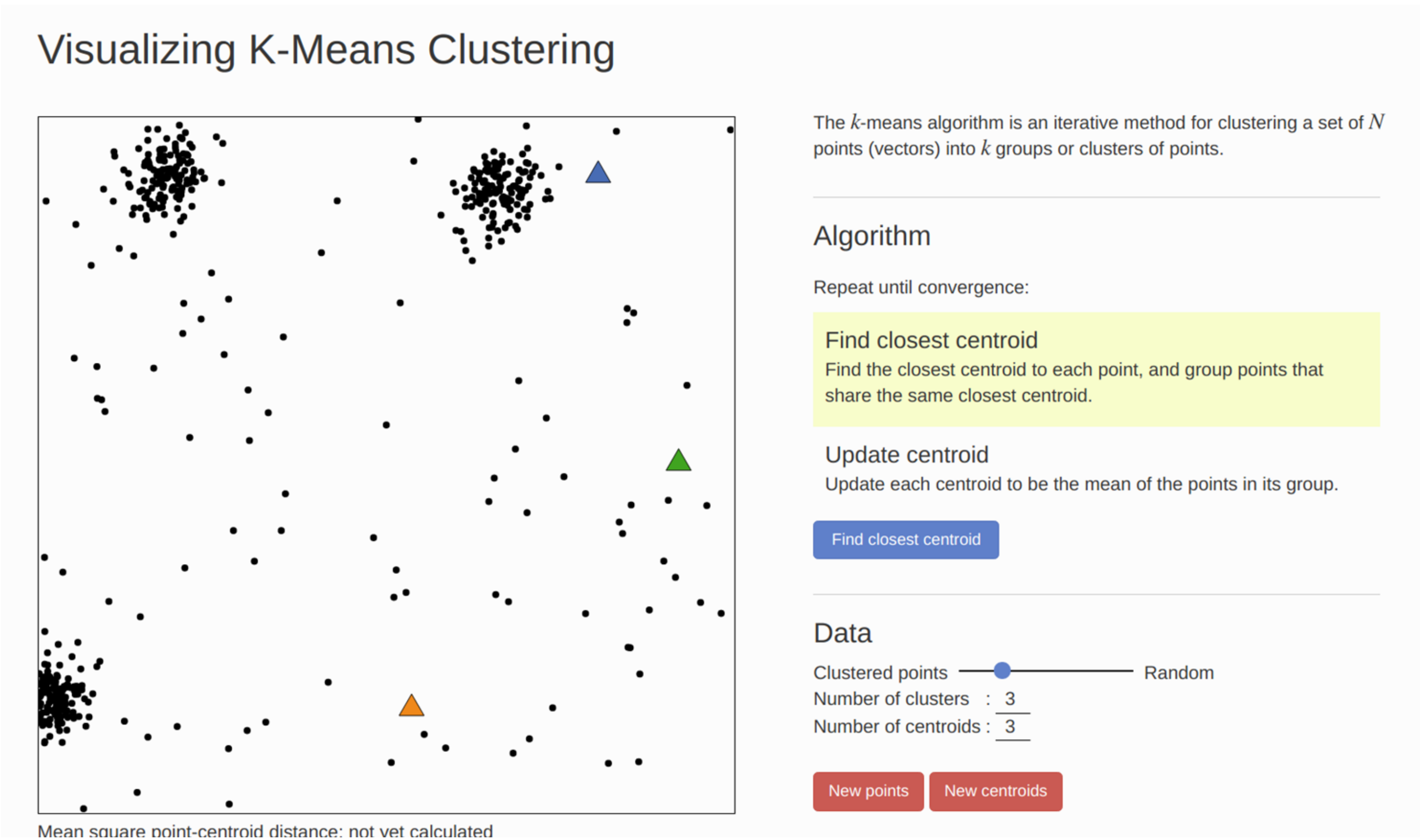
Customer segmentation (e.g. for cost-benefit analysis of new products)

Topic identification (e.g. to speed up manual vetting)

Image or geo-spatial segmentation (e.g. Gojek's supply-demand optimization)

Maybe most importantly, getting a sense of data prior to in-depth modeling!

K-means: the most intuitive clustering



Exercise 1

Code your own k-means and test it on Iris dataset

No sklearn.cluster.Kmeans yet!

Purwadhika
Startup and Coding School

Distance measures

Euclidean

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

Manhattan

$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

Jaccard index

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Cosine similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Numerical features

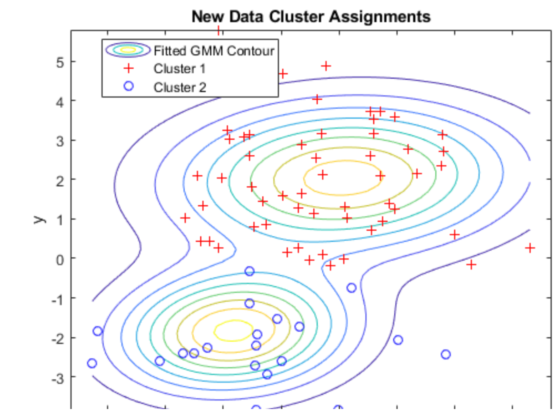
Categorical features

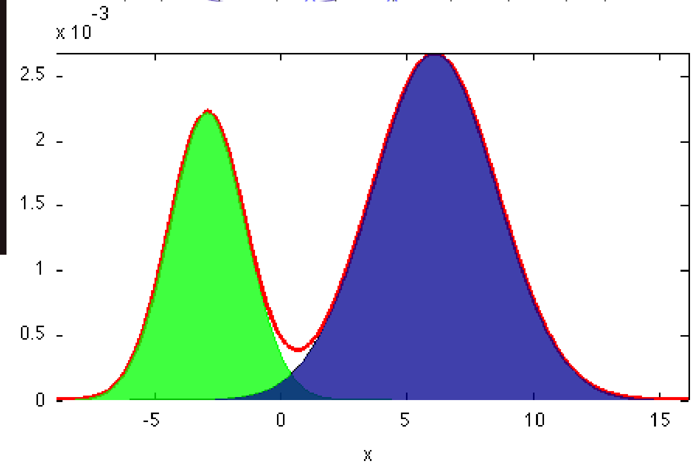
High-dimensional features

Others: correlation, KL divergence, edit distance

Soft, not hard, partition - Gaussian Mixture Model

Real scenario





We can apply the EM algorithm. We assign random values to parameters Θ as the initial values. We then iteratively conduct the E-step and the M-step as follows until the parameters converge or the change is sufficiently small.

In the **E-step**, for each object, $o_i \in \mathbf{O} (1 \le i \le n)$, we calculate the probability that o_i belongs to each distribution, that is,

$$P(\Theta_j|o_i,\Theta) = \frac{P(o_i|\Theta_j)}{\sum_{l=1}^{k} P(o_i|\Theta_l)}. \qquad (11.13)$$

In the **M-step**, we adjust the parameters Θ so that the expected likelihood $P(\mathbf{O}|\Theta)$ in Eq. (11.11) is maximized. This can be achieved by setting

$$\mu_j = \frac{1}{k}\sum_{i=1}^{n} o_i \frac{P(\Theta_j|o_i,\Theta)}{\sum_{l=1}^{n} P(\Theta_j|o_l,\Theta)} = \frac{1}{k}\frac{\sum_{i=1}^{n} o_i P(\Theta_j|o_i,\Theta)}{\sum_{i=1}^{n} P(\Theta_j|o_i,\Theta)} \qquad (11.14)$$

and

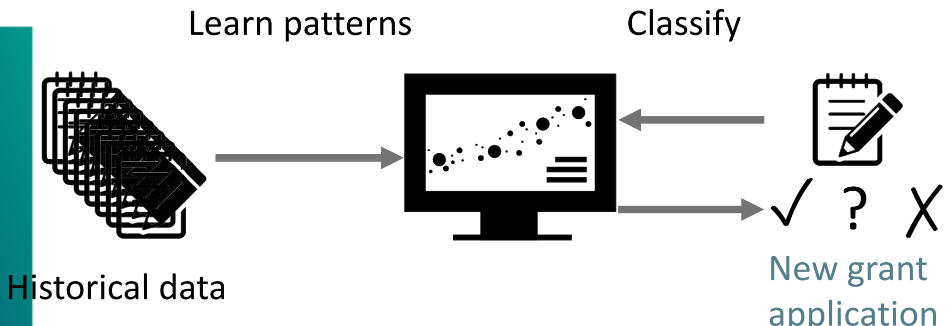
$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} P(\Theta_j|o_i,\Theta)(o_i-u_j)^2}{\sum_{i=1}^{n} P(\Theta_j|o_i,\Theta)}}. \qquad (11.15)$$

Purwadhika
Startup and Coding School

Demo GMM and K-means on Iris

Exercise 2: perform GMM vs. K-means on Digits & 20Newsgroup; check posterior to observe overlapping clusters

My experience in using unsupervised generative modelling



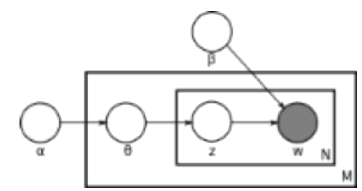
Item Name (Appl)

back end system management software with appointment scheduling system

Item Name (Appl)

corporate website design and development

Latent Dirichlet Allocation



$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta),$$

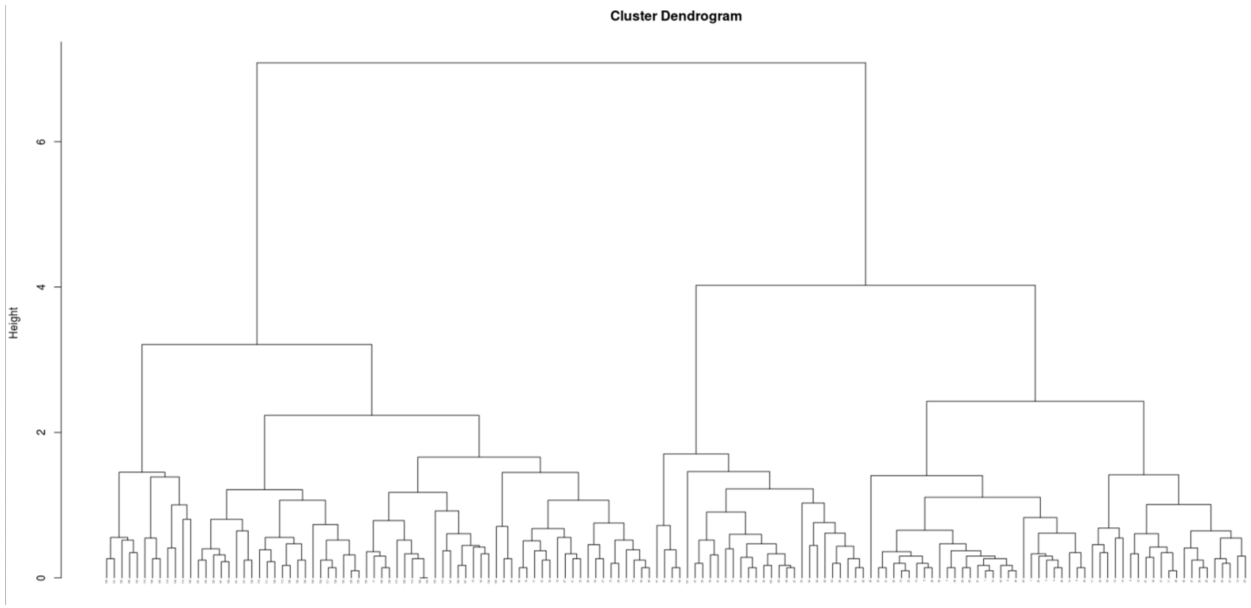
$$p(\theta \mid \alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1},$$

Topic Terms	Weight
system	**0.438**
booking	0.168
appointment	0.127
portal	0.11
scheduling	0.09
website	**0.24**
e-commerce	**0.09**

Topic Terms	Weight
website	**1.38**
development	1.45
design	0.59
e-commerce	**0.35**
system	**0.134**
package	0.07
front-end	0.05

Purwadhika
Startup and Coding School

Sensing the number of clusters - hierarchical clustering



Minimum distance : $dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|\, \boldsymbol{p} - \boldsymbol{p}' \,|\}$

Maximum distance : $dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|\, \boldsymbol{p} - \boldsymbol{p}' \,|\}$

Mean distance : $dist_{mean}(C_i, C_j) = |\, \boldsymbol{m}_i - \boldsymbol{m}_j \,|$

Average distance : $dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i, p' \in C_j} |\, \boldsymbol{p} - \boldsymbol{p}' \,|$

Demo HAC on Iris

Exercise 3: perform HAC on Digits; compare accuracies across linkages

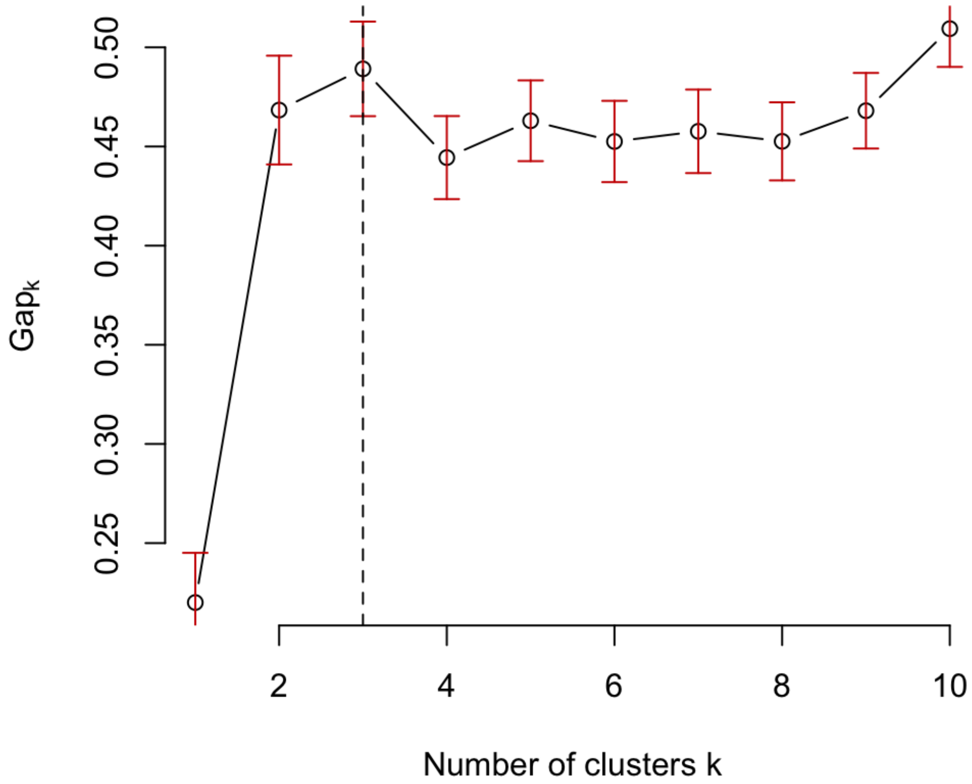
Purwadhika
Startup and Coding School

Finding the number of clusters more systematically

- **Key principle:** minimum intra-cluster distance, maximum intra-cluster distance
- Silhouette index

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

  - b(i): distance from i to the closest object from another cluster
  - a(i): average intra-cluster distance



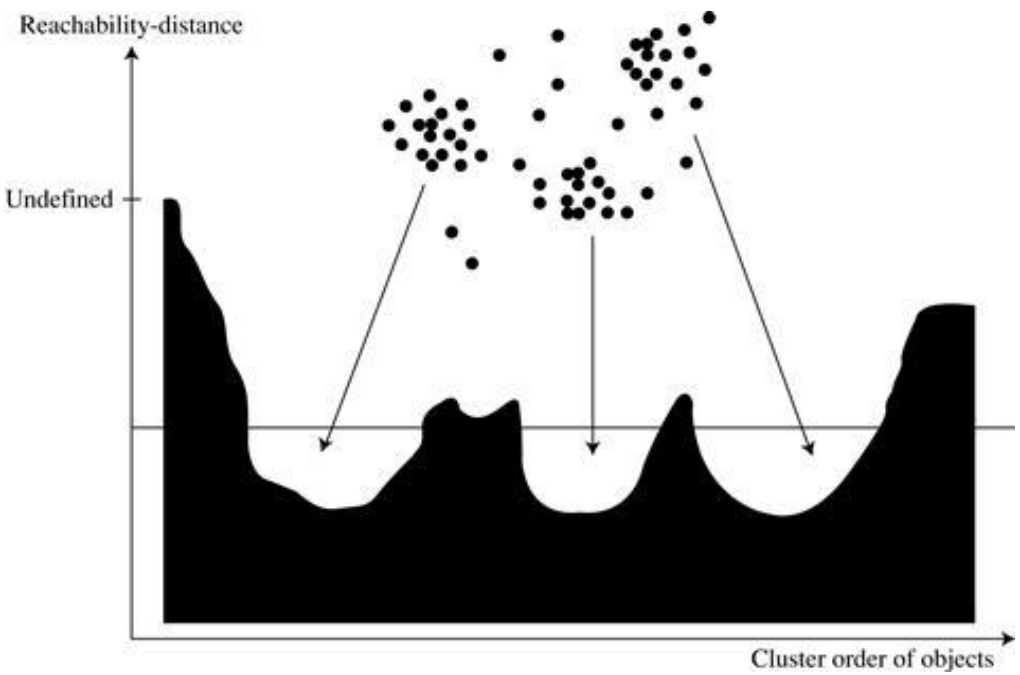
Demo Silhouette plotting on Digits (K-means)

Exercise 4: plot Silhouette indices for k=2,...,30 for Digits, 20Newsgroup for HAC and GMM

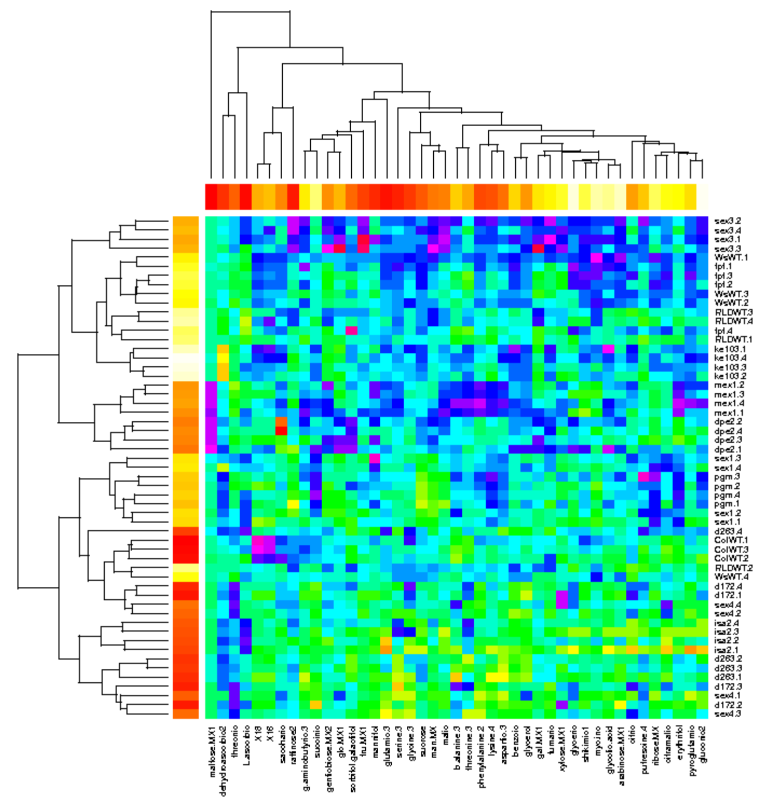
Other clustering variants: density and co-clustering

Density-based clustering (dealing with outliers)

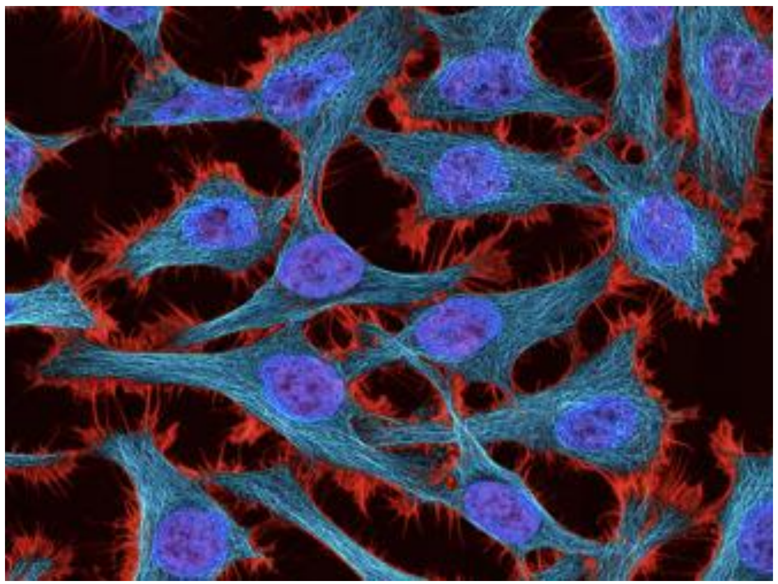
DBScan, OPTICS

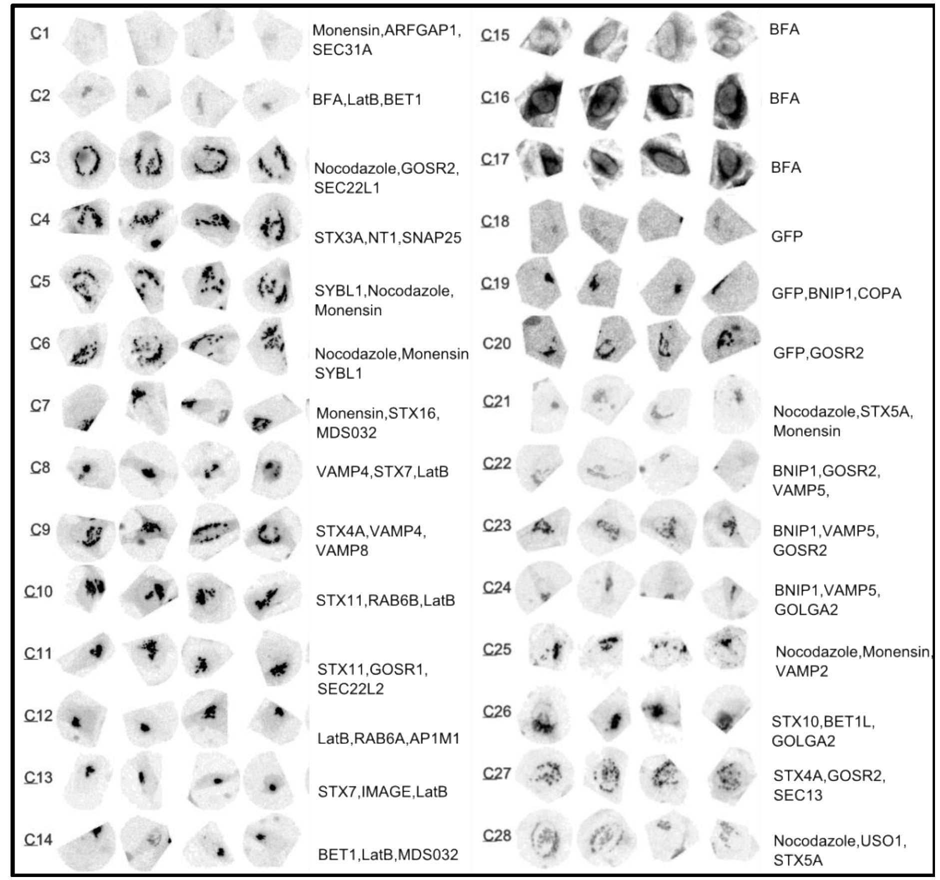


Co-clustering or biclustering (dealing with high-dimensional features)



Real world clustering is often messy

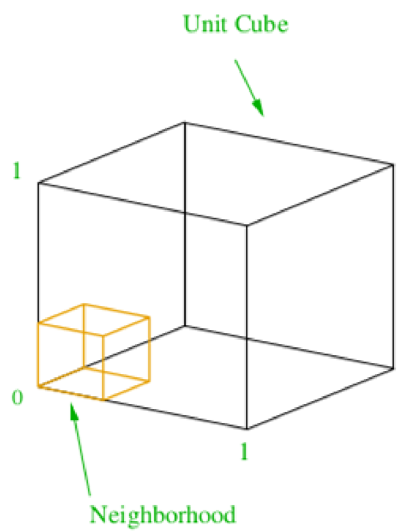


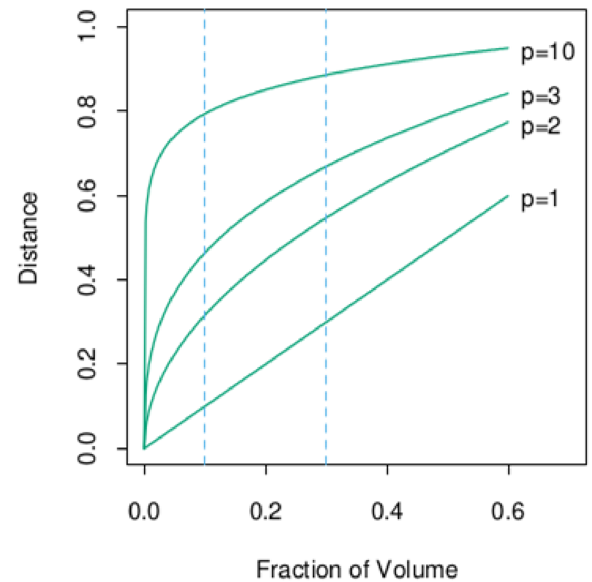


Case study 1: customer segmentation

Purwadhika
Startup and Coding School

The problems with high-dimensionality

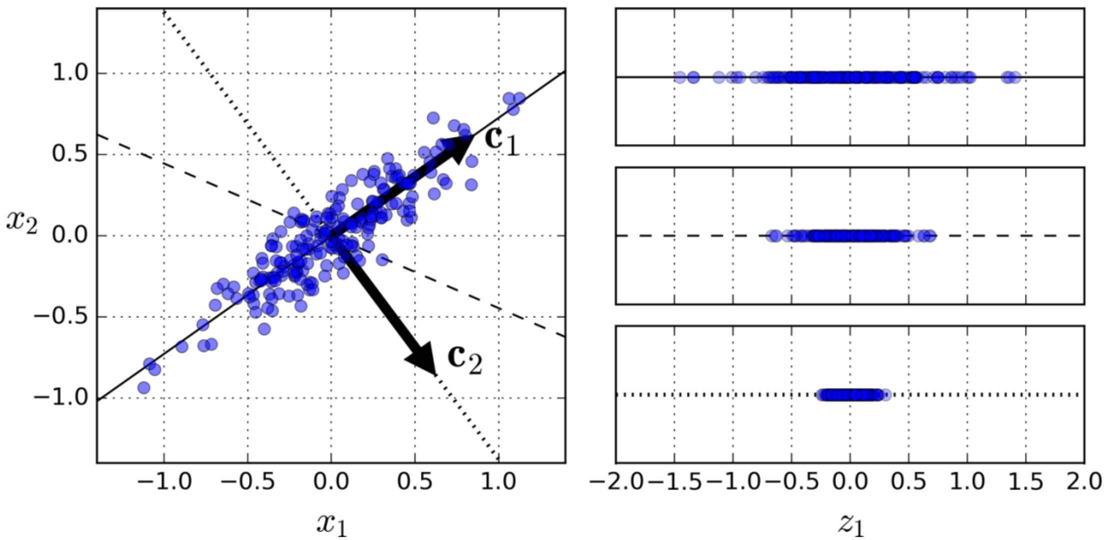


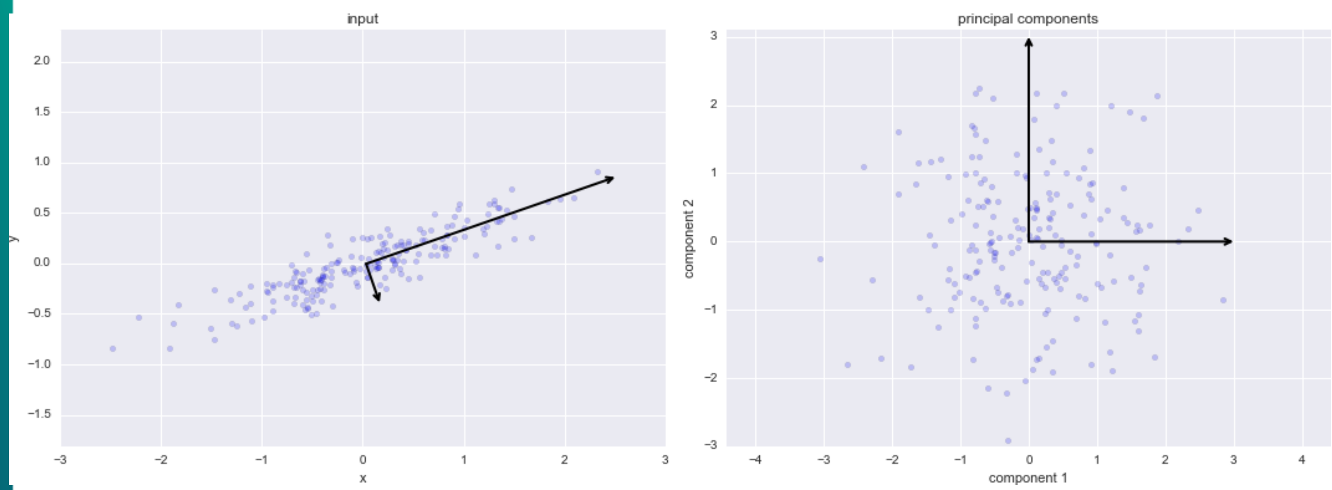


Hard to visualise, some dimensions are noisy, some dimensions are correlated, sparsity requiring impractical volume of training data and strange distance measure behavior (http://mlwiki.org/index.php/Euclidean_Distance)

Purwadhika
Startup and Coding School

Principal component analysis





Optimization problem: project x but try to maximize the variance

Setting the gradient to 0 gives eigenvalue equation:

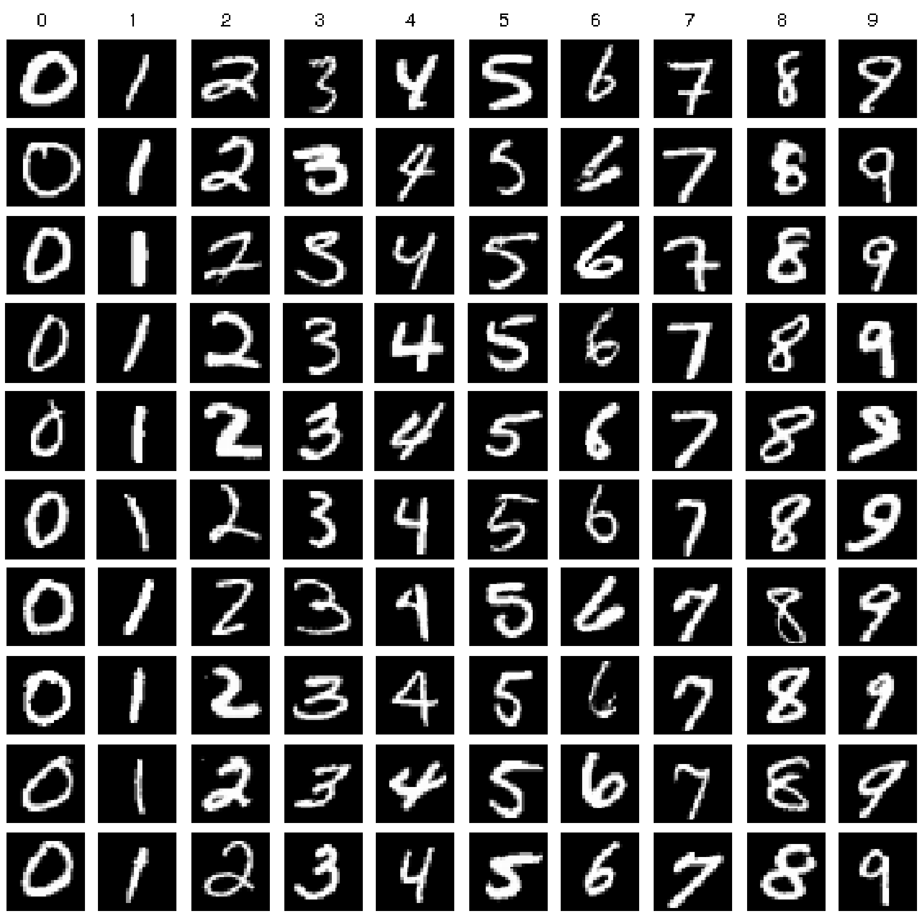
Σa1 = λ1a1

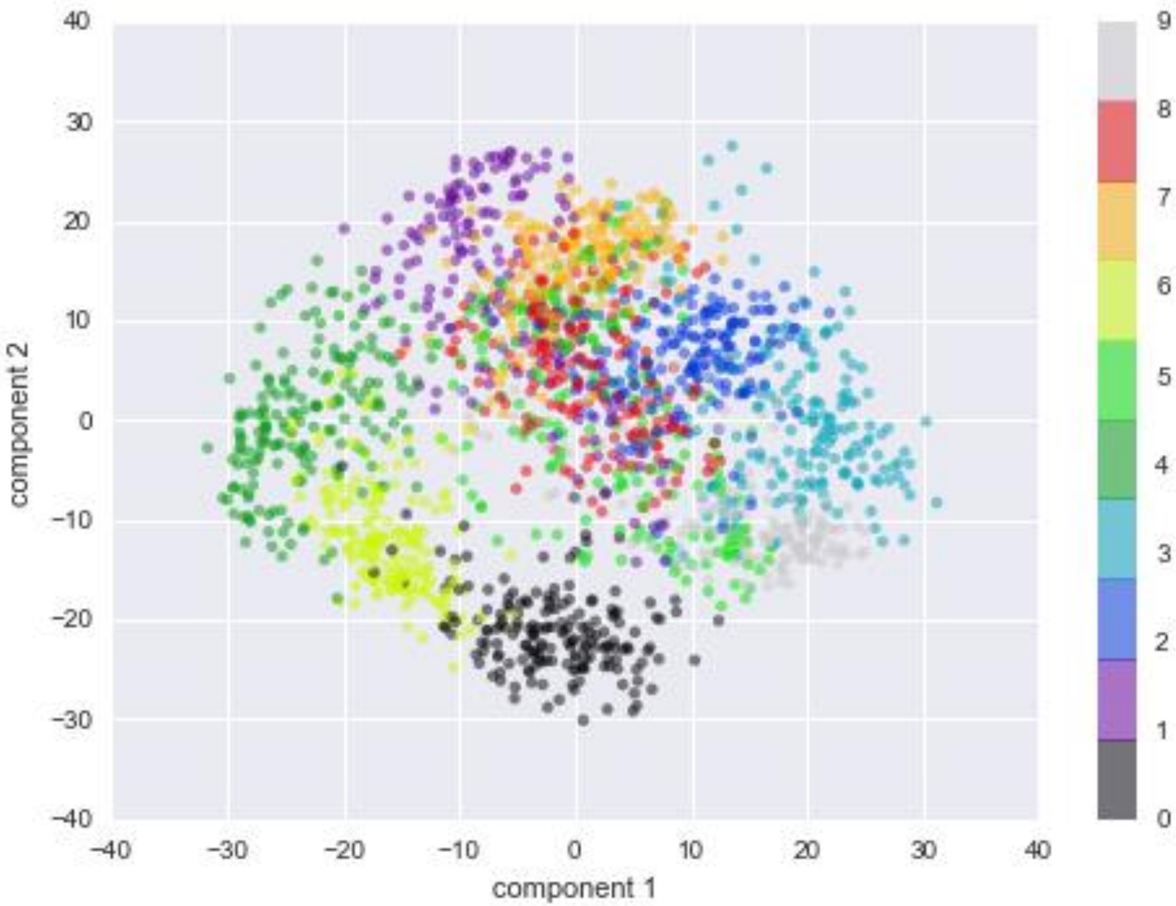
i.e. projecting x in the direction of the eigenvector with the highest eigenvalue gives the highest variance

Repeat the process for the second axis to project results in eigenvector corresponding to the second highest eigenvalue, and so on …

These eigenvectors are effectively the principal components

Demo PCA on Digits





Purwadhika
Startup and Coding School

PCA hands-on

Exercise 5: perform PCA on Digits, 20Newsgroup and analyse if it improves clustering accuracy

Exercise 6 (optional): implement your own naive PCA

- Generate the covariance matrix of a dataset
- Perform eigen decomposition of the covariance matrix
- Sort eigenvalues
- Project dataset into the sorted eigenvectors

Case study 2: repeat customer segmentation, this time using PCA for dimensionality reduction

Purwadhika
Startup and Coding School

What is association rules mining?

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.3		

Association rules mined

	antecedants	consequents	support	confidence	lift
8	(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)	0.137755	0.888889	6.968889
9	(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.127551	0.960000	6.968889
10	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.815789	8.642959
11	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.837838	8.642959
16	(SET/6 RED SPOTTY PAPER CUPS, SET/6 RED SPOTTY...	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.122449	0.812500	6.125000
17	(SET/6 RED SPOTTY PAPER CUPS, SET/20 RED RETRO...	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.975000	7.644000
18	(SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.975000	7.077778
22	(SET/6 RED SPOTTY PAPER PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.127551	0.800000	6.030769

Association rules mining steps:

Transaction table ->
Frequent itemsets (support) ->
Association rules (confidence)

Support, confidence, lift

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Support (A -> B): **P(A ∪ B)**

Is it rare?

Confidence (A -> B): **P(B | A)** *How strong is the rule?*

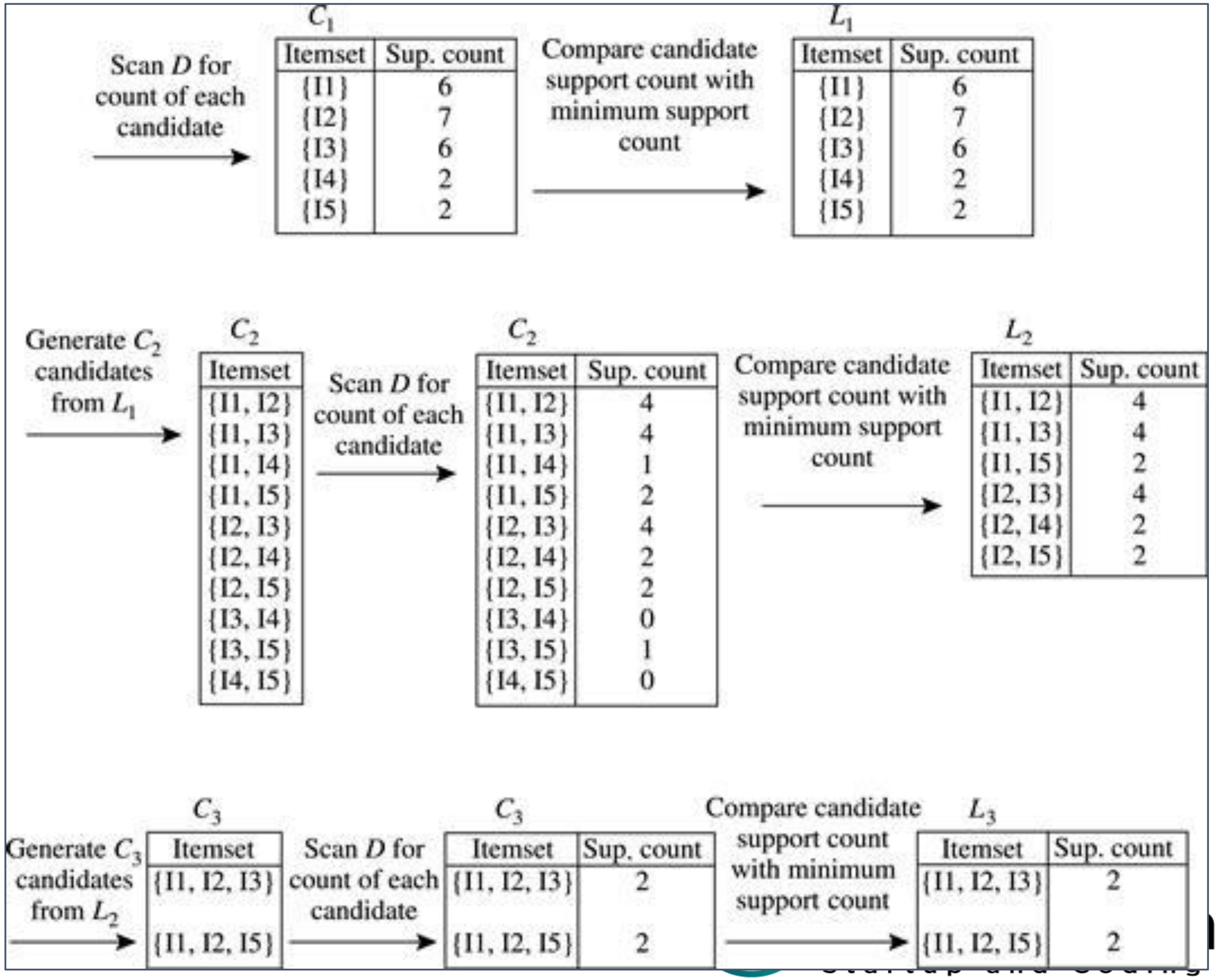
Lift (A -> B): **P(A ∪ B) / {P(A) P(B)}** *Complementary or cannibalism?*

Scale is the key issue

- **Apriori principle:** the subsets of a frequent itemset must be frequent
- When items are consistently ordered across sets, **to avoid redundancy**, generate k-itemset only from two (k-1)-itemsets that share the same (k-2) items.

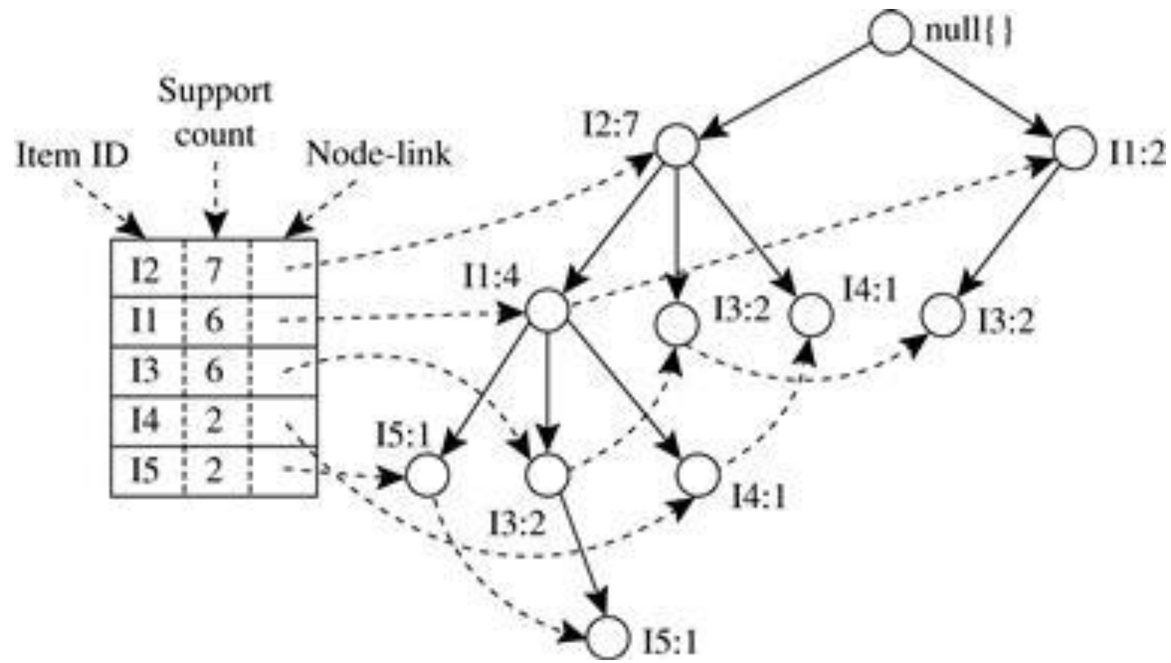
Finding frequent itemsets by Apriori

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



FP-Growth – a faster algorithm

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



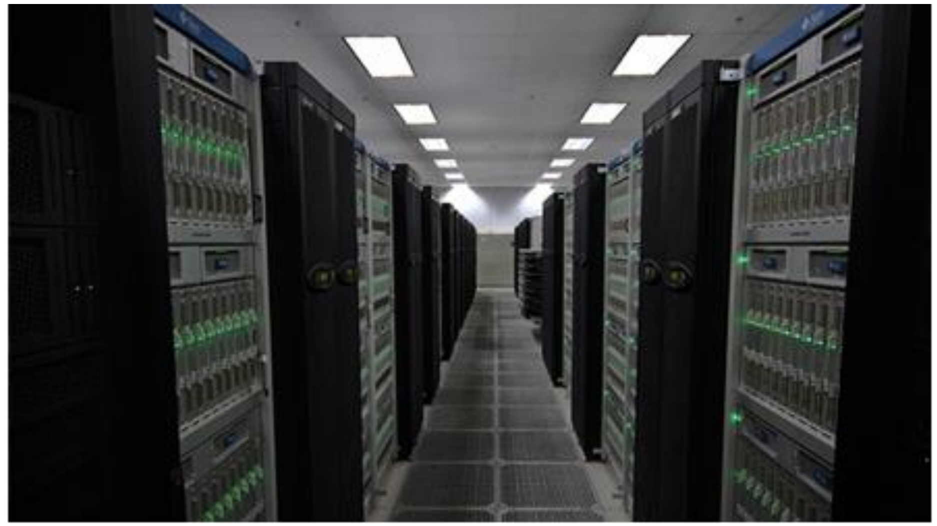
Item	*Conditional Pattern Base*	*Conditional FP-tree*	*Frequent Patterns Generated*
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

Startup and Coding School

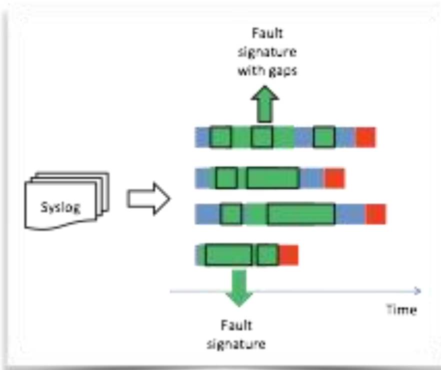
Demo association rules mining on AllElectronics dataset

Exercise 7 (optional): perform frequent itemsets on 20Newsgroup to find associated words/phrases

My association rules story



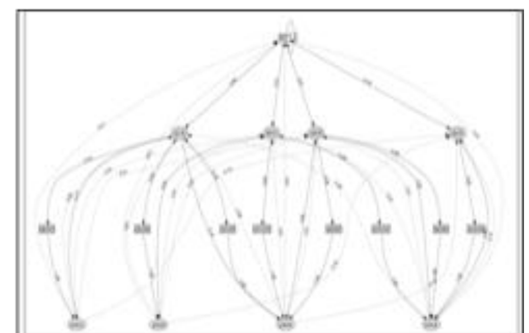
- (.*)LustreError: (.*)Skipped (.*)previous similar message (.*) ---> (.*)LustreError: (.*)Skipped (.*)previous similar message (.*)
- (.*)LustreError: (.*)Skipped (.*)previous similar messages (.*) ---> (.*)LustreError: (.*)processing error (.*)lens (.*)ref (.*)
- (.*)LustreError: (.*)Skipped (.*)previous similar messages (.*) ---> (.*)LustreError: (.*)type (.*)status (.*)lens (.*)ref (.*)
- (.*)LustreError: (.*)page (.*)map (.*)index (.*)priv (.*) ---> (.*)LustreError: (.*)page (.*)map (.*)index (.*)priv (.*)
- (.*)LustreError: (.*)processing error (.*) lens (.*)ref (.*) ---> (.*)LustreError: (.*) Skipped (.*)previous similar messages (.*)
- (.*)LustreError: (.*)processing error (.*) lens (.*)ref (.*) ---> (.*)LustreError: (.*) processing



$S = ACBDAHCBADFGAIEB$

	A	B	C	D	E	F	G	H	I
frequency	4	3	2	2	1	1	1	1	1

subsequence	*freq*	$g = 0$	$g = 1$	$g = 2$	gap symbols
AB	3	0	1	2	C(2), E, H, I
AC	2	1	1	0	H
AD	2	1	0	1	B, C
BA	2	1	1	0	D
BD	2	1	1	0	A
CA	2	0	1	1	B(2), D
CB	2	2	0	0	-
CD	2	0	1	1	A, B(2)
DA	2	1	0	1	F, G



Purwadhika
Startup and Coding School

Case study 3: perform association rules mining on e-commerce dataset, taking into account customer segments

Ensemble supervised learning

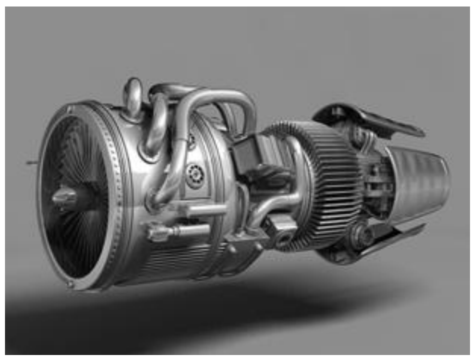
Consolidating predictions from multiple models

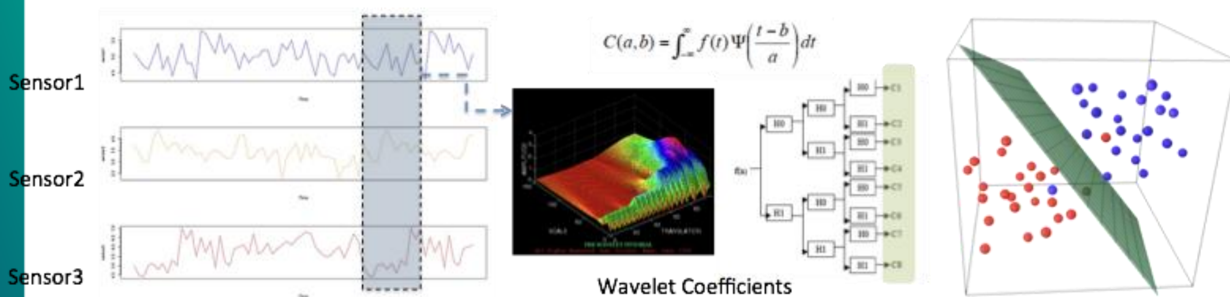
Consistently topping Kaggle competitions

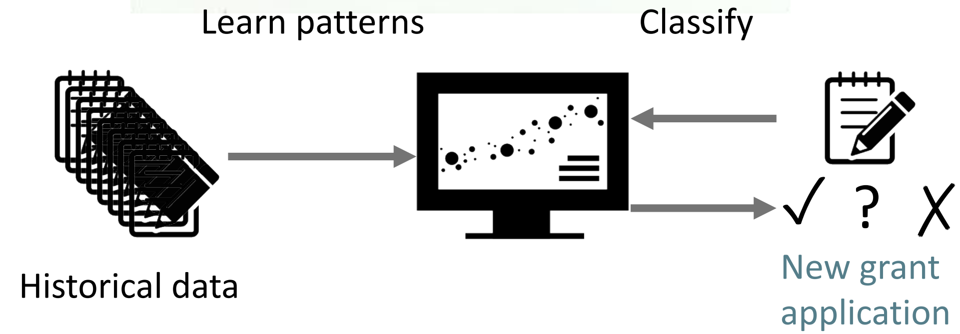
Main methods:

- Voting
- Bagging
  - Random Forest: special case for tree-based bagging
- Boosting

My “ensemble learning story”



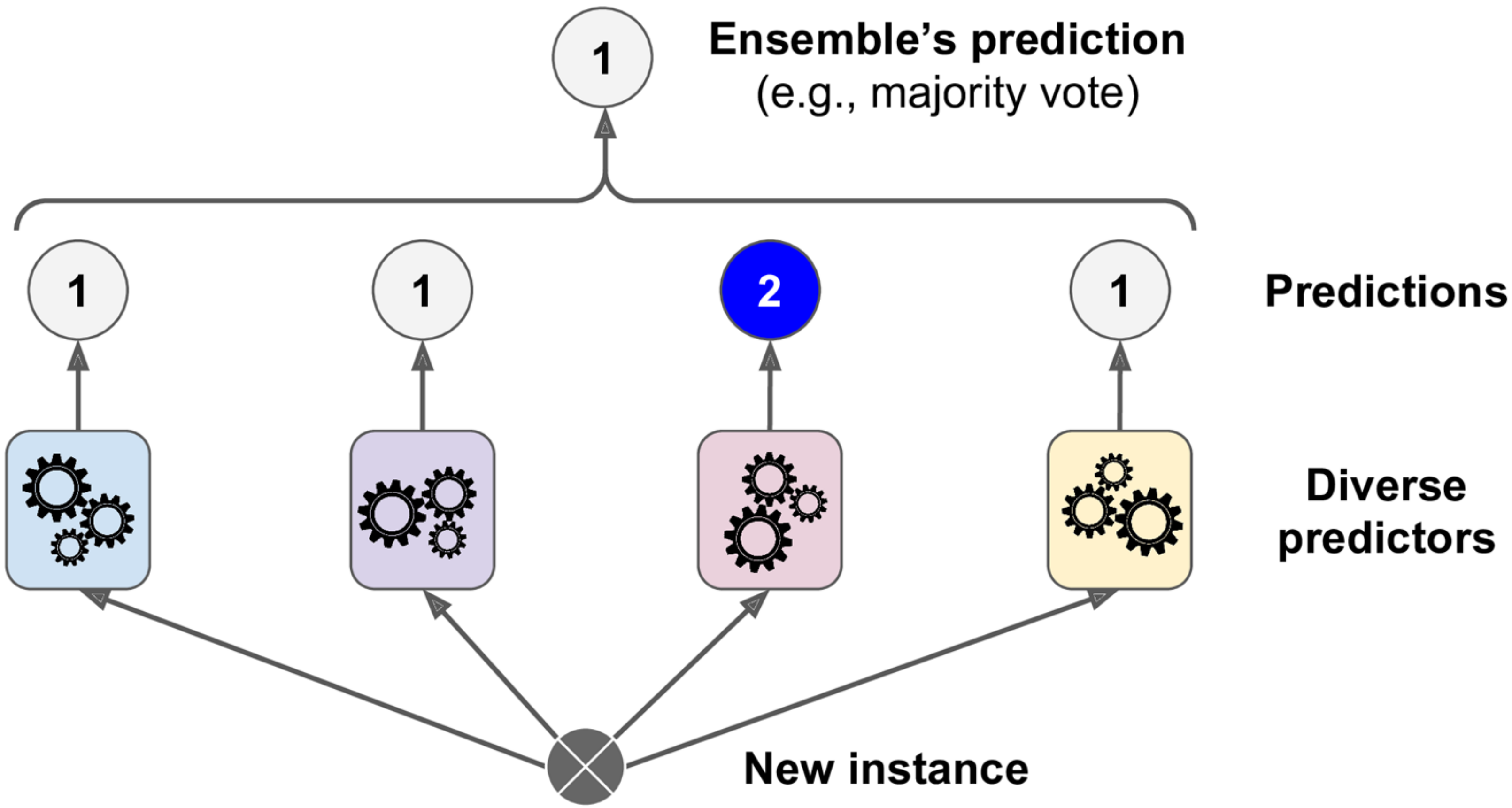




traveloka

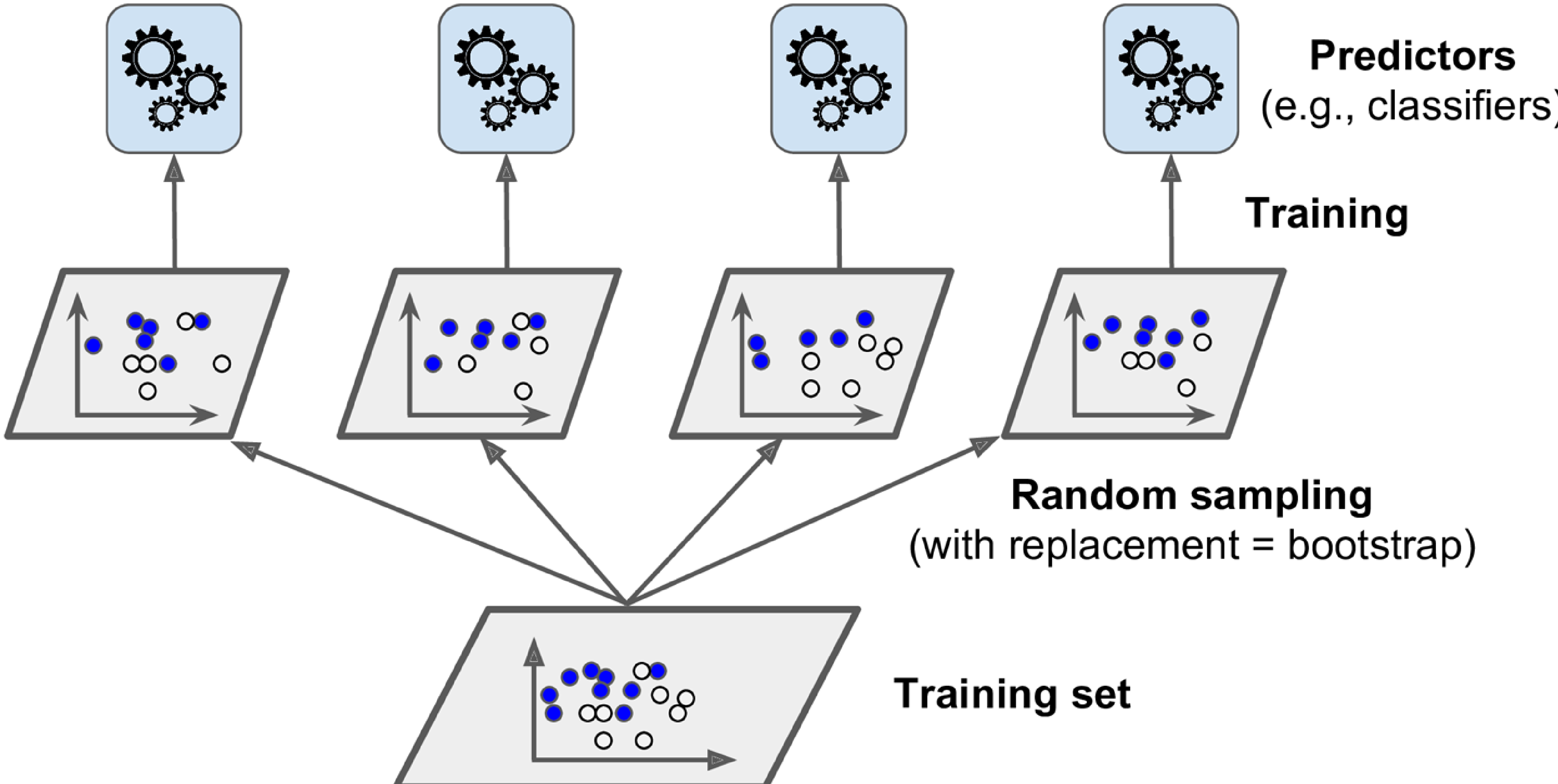
Purwadhika
Startup and Coding School

Voting

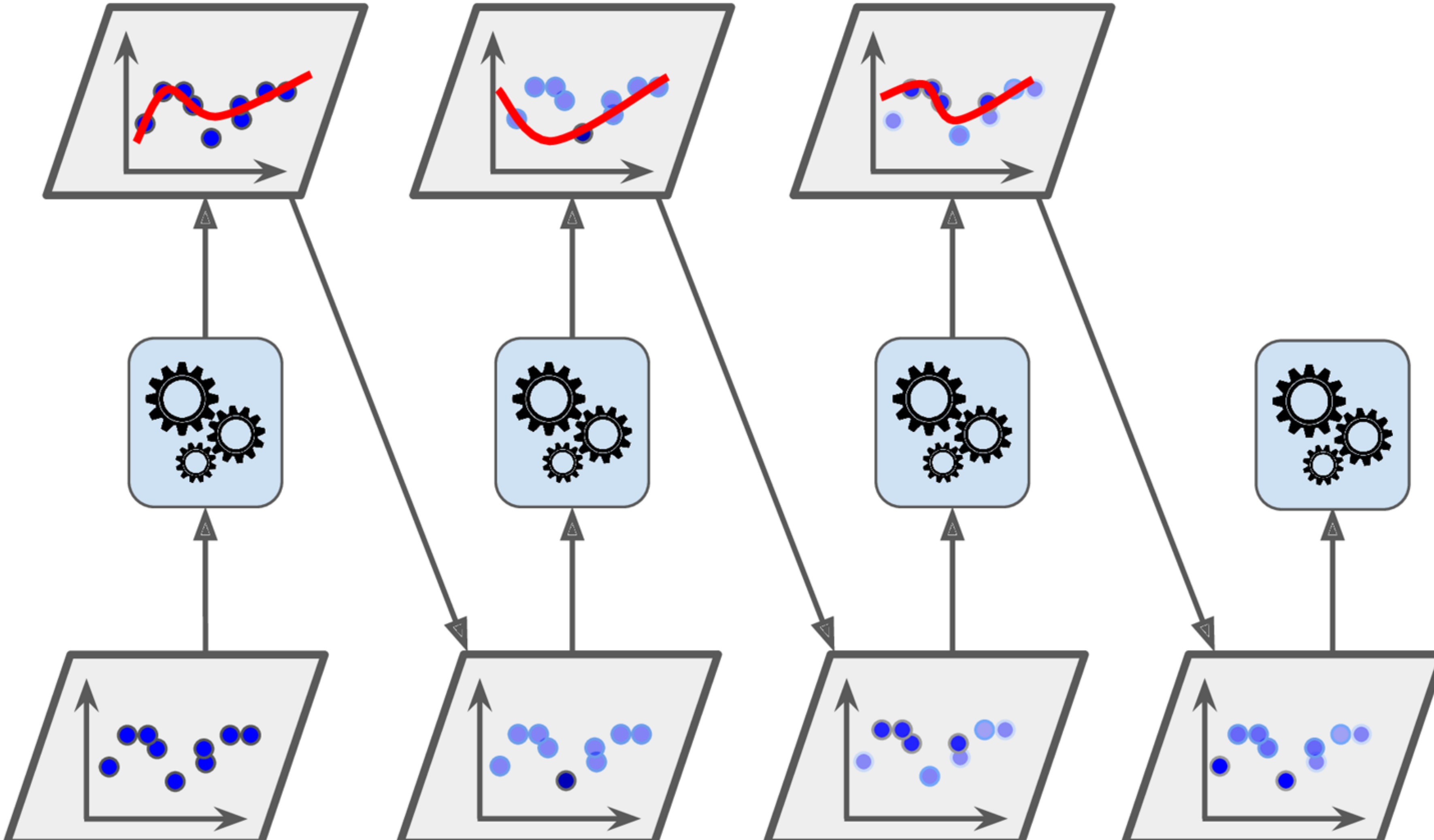


Purwadhika
Startup and Coding School

Bagging


Predictors
(e.g., classifiers)
Training
Random sampling
(with replacement = bootstrap)
Training set

Boosting (AdaBoost)

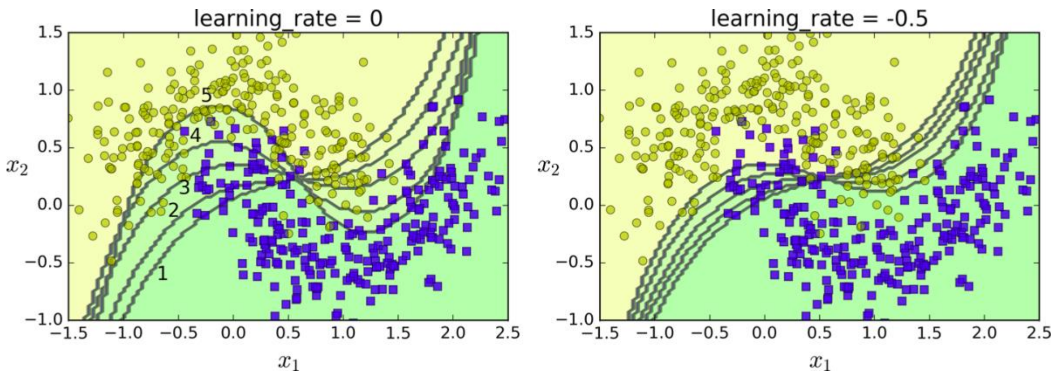


Adaboost

Subsequent iteration has previously misclassified instances up-weighted

$$w_i^{j+1} \leftarrow w_i^{j} e^{\alpha_j}$$

$$\alpha_j = \eta \log \frac{1-r_j}{r_j} \text{ and } r_j = \frac{\sum_{p \in P} w_p}{\sum_{i=1}^{N} w_i}$$

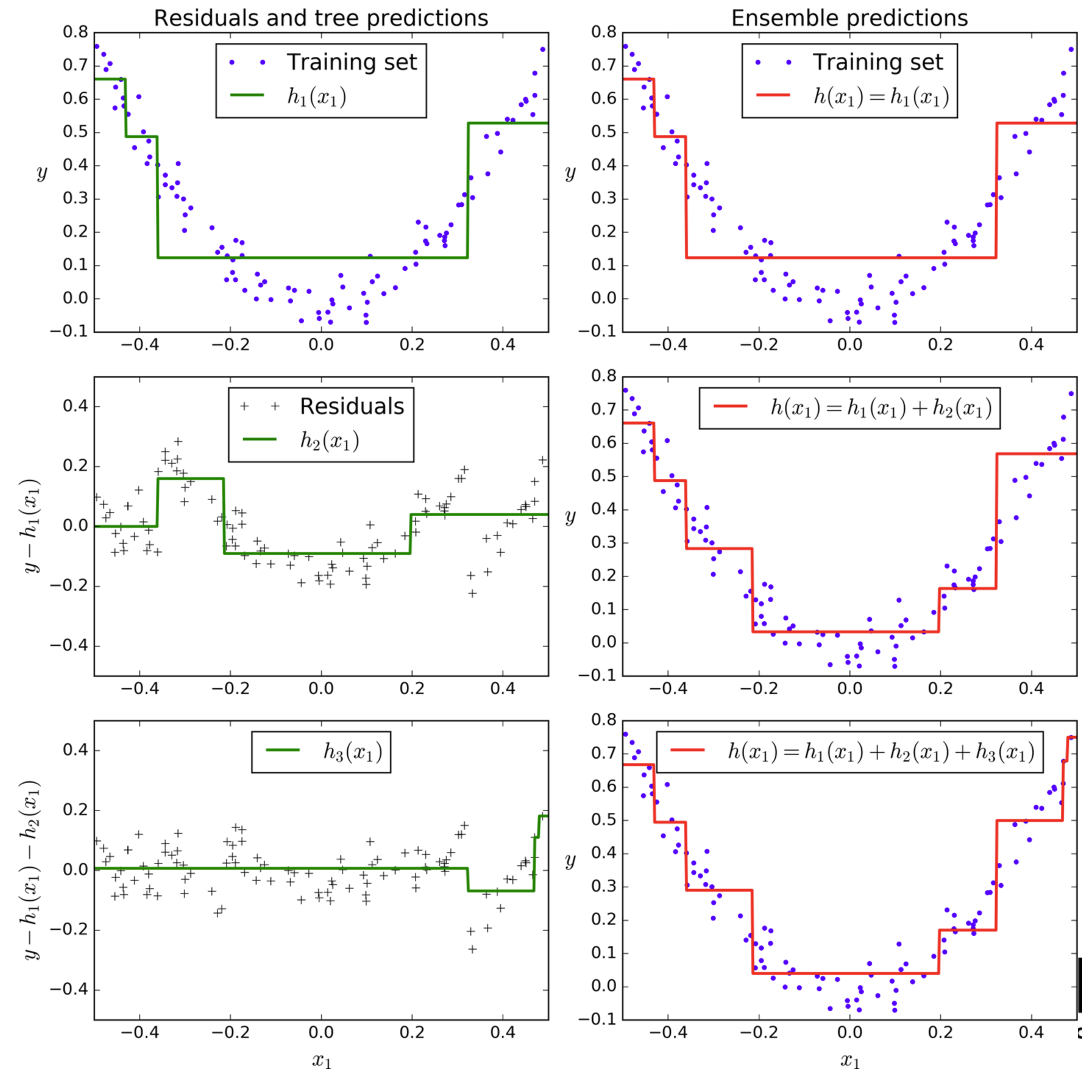


where P is the index of misclassified instances

Gradient Boosting

Subsequent iteration learns to fit residuals from previous iteration's model

If each iteration subsamples the training instances, the algorithm is called Stochastic Gradient Boosting



Demo ensemble learning on Iris

Ensemble learning hands-on

Exercise 8:

Perform voting, bagging, Random Forest and boosting on 20Newsgroup

Compare the accuracies

References

- Data Mining Concepts and Techniques 3rd edition, Han Jiawei, Michelline Kamber and Jian Pei, 2011
- Hands-on Machine Learning with Scikit-Learn and TensorFlow, Aurelie Geron, 2017
- Python Data Science Handbook, Jake VanderPlas, 2016
- Data Mining, Ian Witten and Eibe Frank, 2005
- The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman
- The Data Science Handbook, Field Cady, 2017

Purwadhika
Startup and Coding School