ml for ir

Information Retrieval and Machine Learning

Vasudeva Varma
Search and Information Extraction Lab
IIIT Hyderabad
vv@iiit.ac.in http://www.iiit.ac.in/~vasu

what is learning?

Very loosely:

We have *lots(?)* of data and wish to automatically ***learn*** ***concept definitions*** in order to determine if new examples belong to the concept or not.

how does machine learning work?

outlook	temperature	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunn				yes
rainy				yes
				yes
				yes
overcast			ALSE	yes
rainy			TRUE	no

A slightly … e learner will find t … e most informati … s. What do you th … s case?

Outlook:

Sunny -> No

Overcast -> Yes

Rainy-> Yes

Class

What w

<Feature Name>:

<value> -> <prediction>

<value> -> <prediction>

…

what will be the prediction?

Model

Outlook:
- Sunny -> No
- Overcast -> Yes
- Rainy-> Yes

New Data

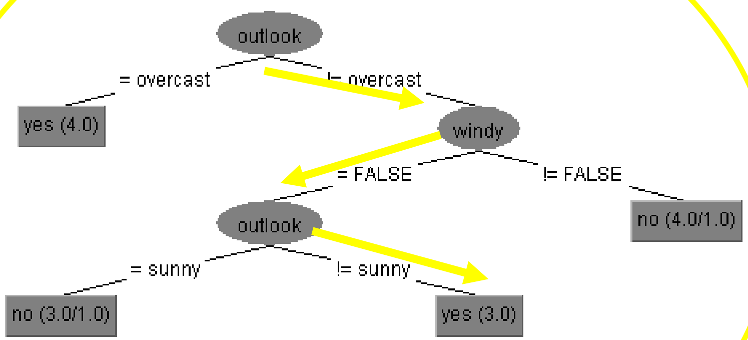
outlook	temperature	humidity	windy	play
rainy	cool	high	FALSE	Yes

two simple algorithms

– 0R – Predict the majority class

– 1R – Use the most predictive single feature

more complex algorithm…



* Only makes 2 mistakes!

- Decision Trees

What will it do with this example?

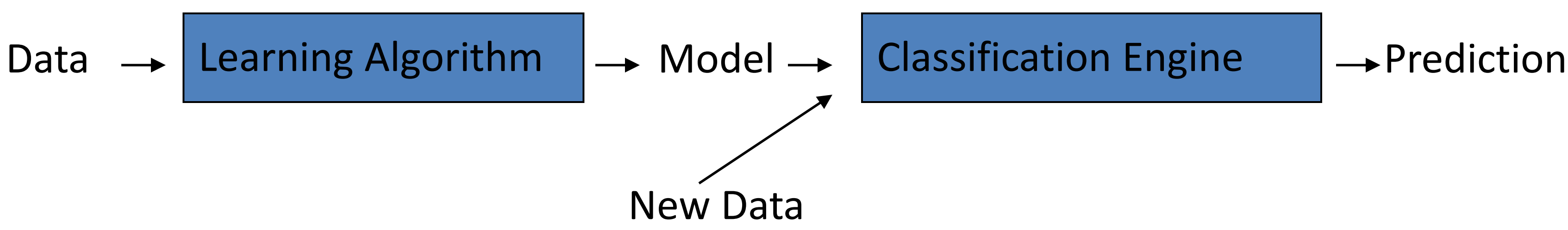
outlook	temperature	humidity	windy	play
rainy	cool	high	FALSE	?

why is it better?

- Not because it is more complex
  - Sometimes more complexity makes performance worse
- What is different in what the three rule representations assume about your data?
  - 0R
  - 1R
  - Trees
- The best algorithm for your data will give you exactly the power you need

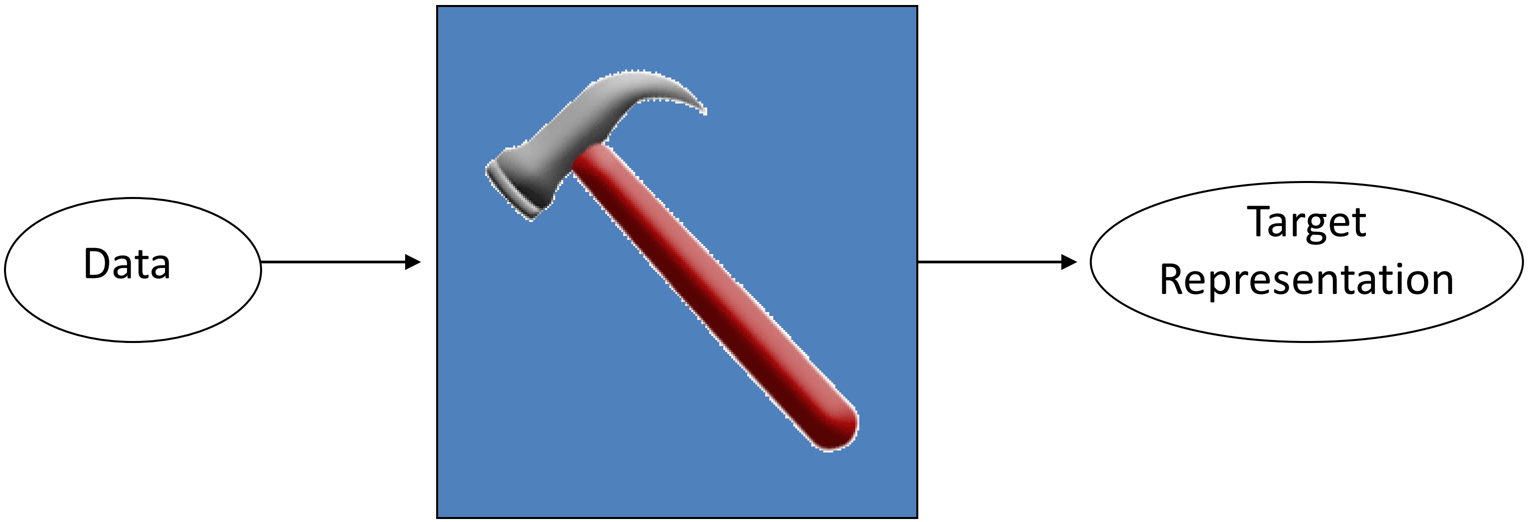
how does machine learning work?

- Automatically or *semi-automatically*
  - Inducing concepts (i.e., rules) from data
  - Finding patterns in data
  - Explaining data
  - Making predictions

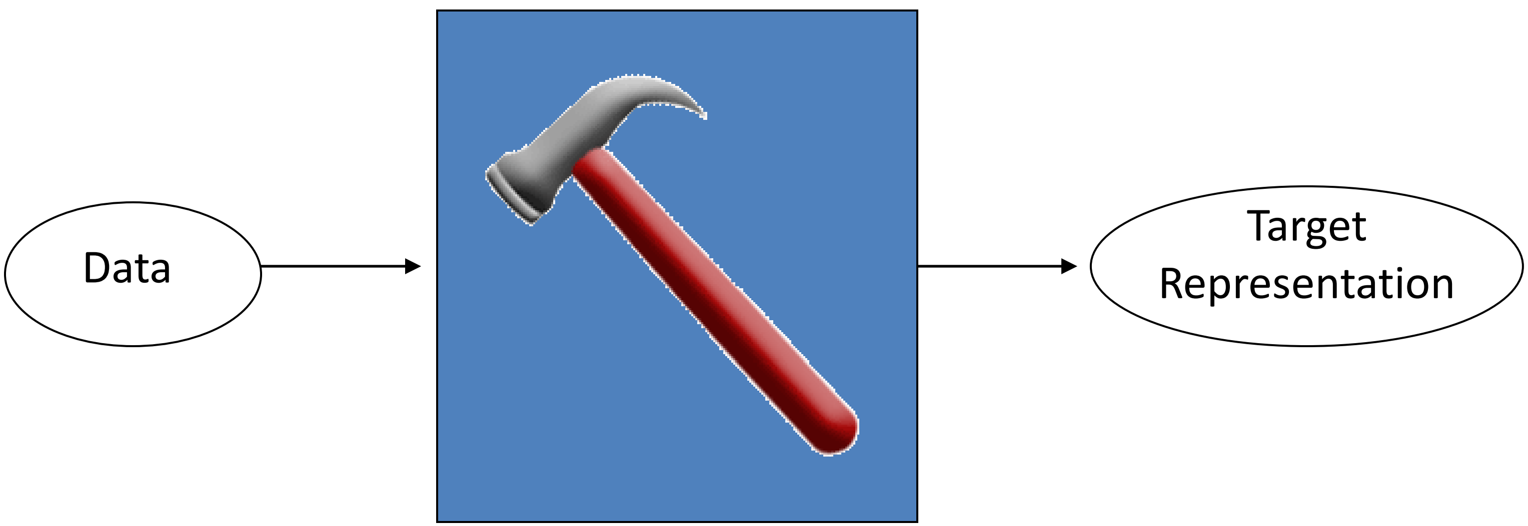


overview of machine learning process skills

naïve approach: when all you have is a hammer…

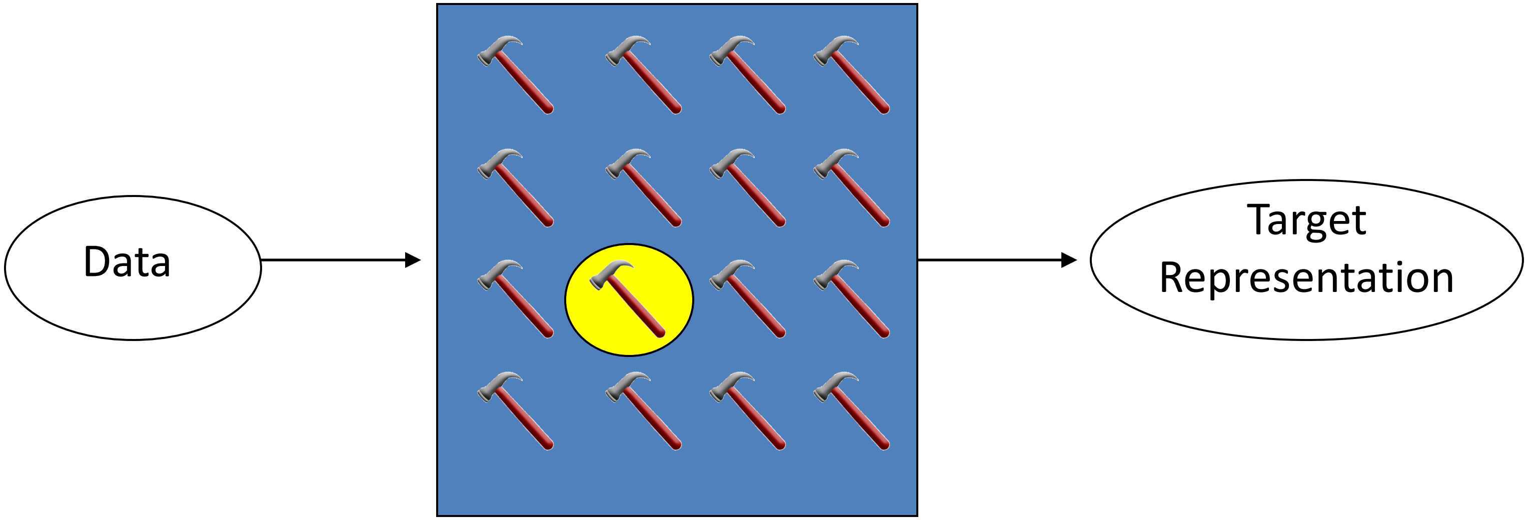


naïve approach: when all you have is a hammer…

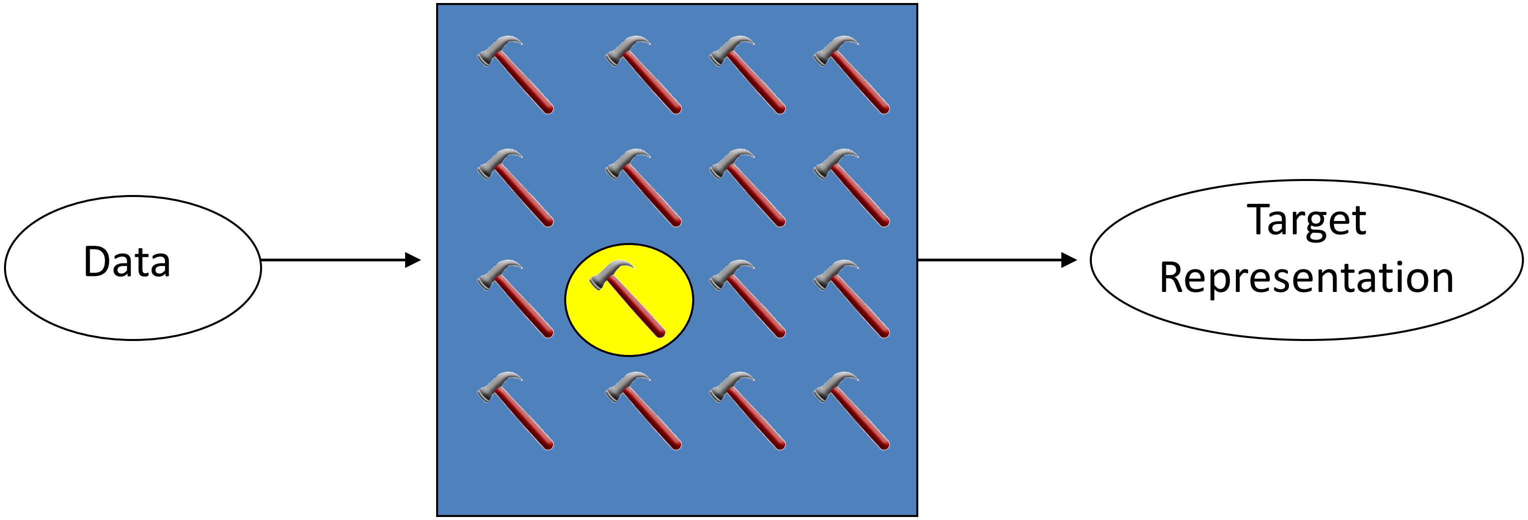


Problem: there isn't one universally best approach!!!!!

slightly less naïve approach: aimless wandering…

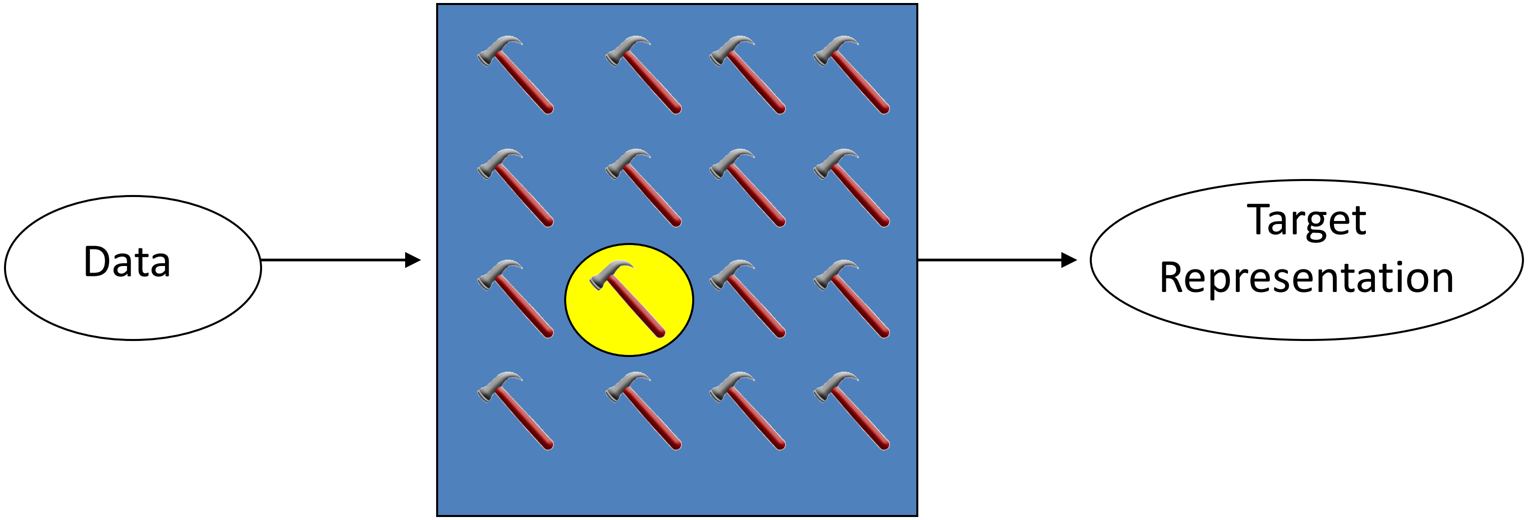


slightly less naïve approach: aimless wandering…



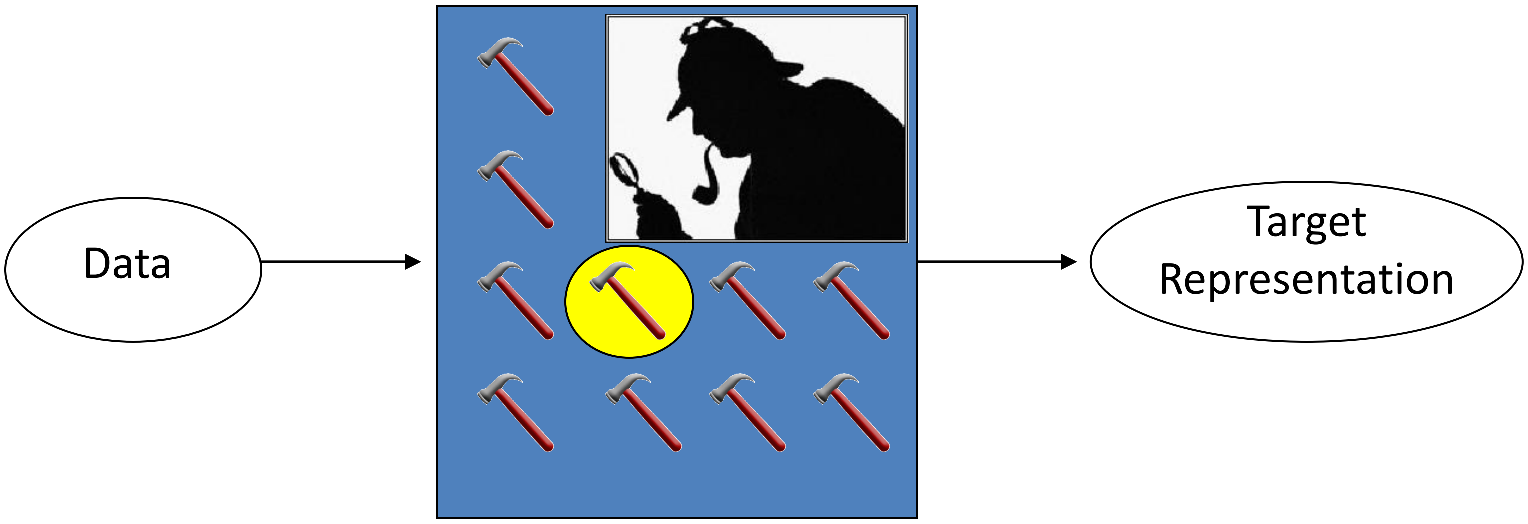
Problem 1: It takes too long!!!

slightly less naïve approach: aimless wandering…

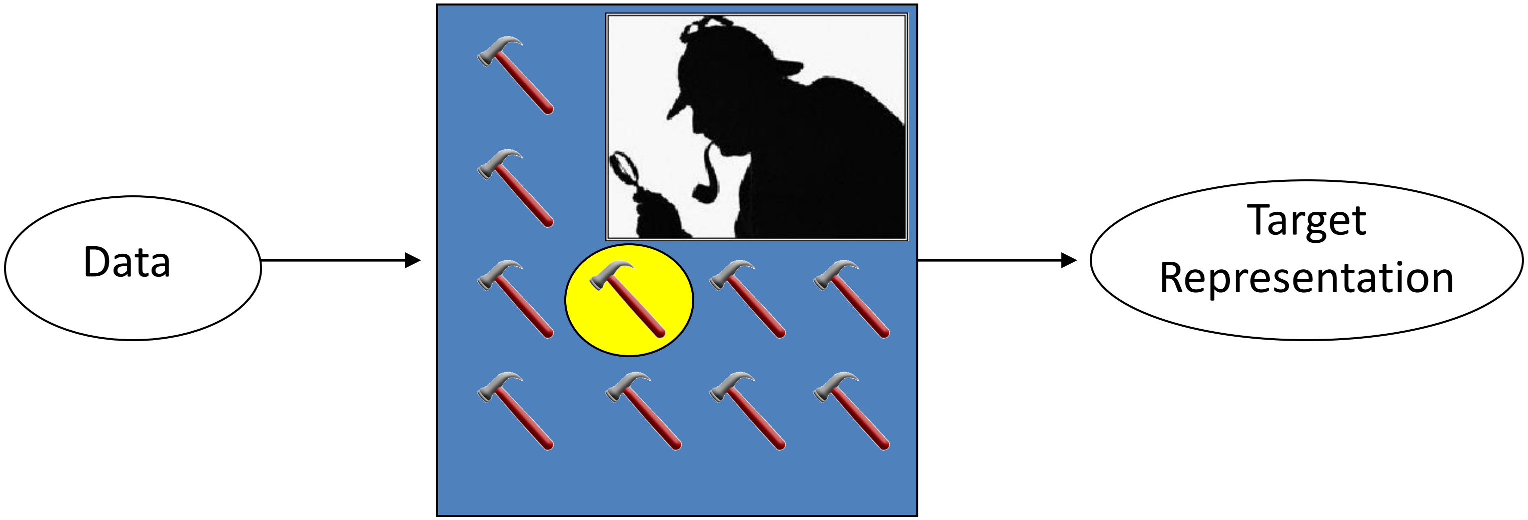


Problem 2: You might not realize all of the options that are available to you!

expert approach: hypothesis driven

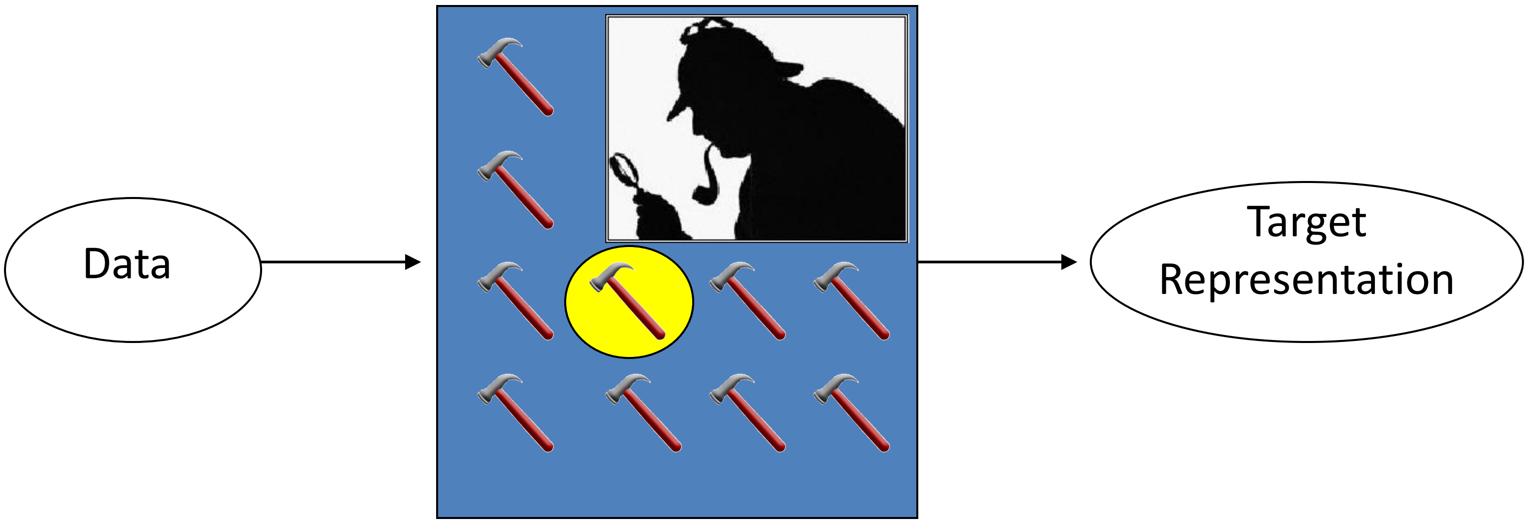


expert approach: hypothesis driven



You *might* end up with the same solution in the end, but you'll get there faster.

expert approach: hypothesis driven



Today we'll start to learn how!

ML paradigms

machine learning paradigms

- Supervised Learning
  - Classification and Regression (Naïve Bayes, SVM)
  - Sequence Learning (HMM, CRF)
- Unsupervised Learning
  - Clustering (K-means, HAC)
  - Dimensionality reduction (LDA, PCA)
- Semi-supervised Learning
  - Co-training (Multi-class probabilistic classification)
  - Active learning (domain and relation adoption)
- Deep Learning

supervised vs. unsupervised learning

- Supervised learning (e.g. classification)
  - Supervised learning from examples.
  - The data points (observations, measurements, etc.) are labeled with pre-defined classes.
  - Test data is classified into these classes.

- Unsupervised learning (e.g. clustering)
  - Class labels of the data are unknown
  - Given a set of data, the task is to establish the existence of classes or clusters in the data

fundamental assumption of learning

The distribution of training examples is identical to the distribution of test examples (including future unseen examples).

- In practice, this assumption is often violated to **certain degree**.
  - Strong violations will clearly result in poor accuracy.

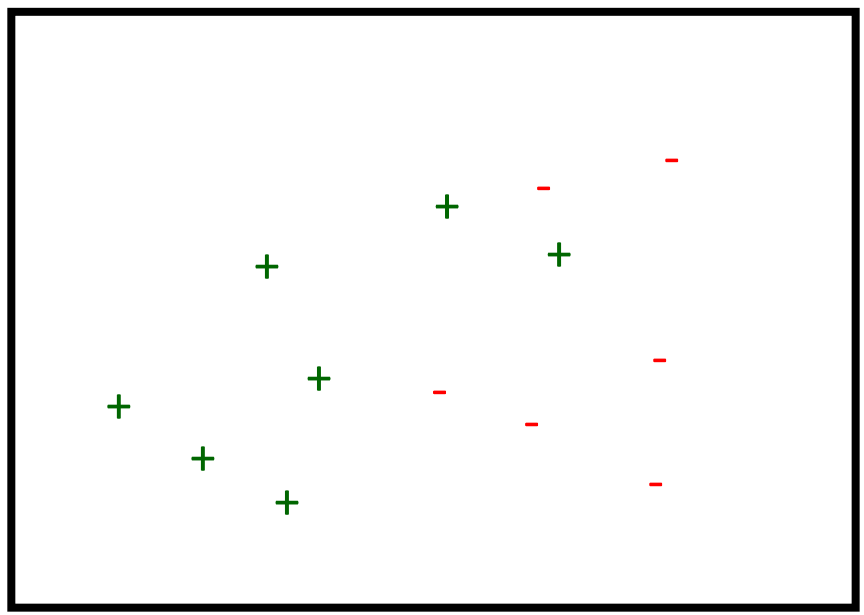
- To achieve good accuracy on the test data, training examples must be sufficiently representative of the test data.

classification

k-nearest neighbor (knn)

- In feature space, training examples are

Feature #2
(e.g.., roundness)

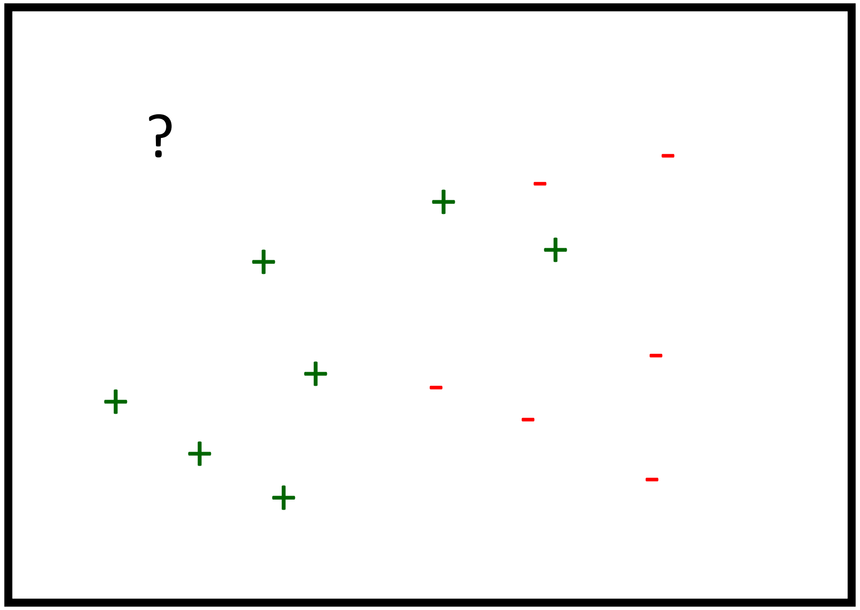


Feature #1 (e.g.., ‘area’)

k-nearest neighbor (knn)

- We want to label **'?'**

Feature #2
(e.g.., roundness)

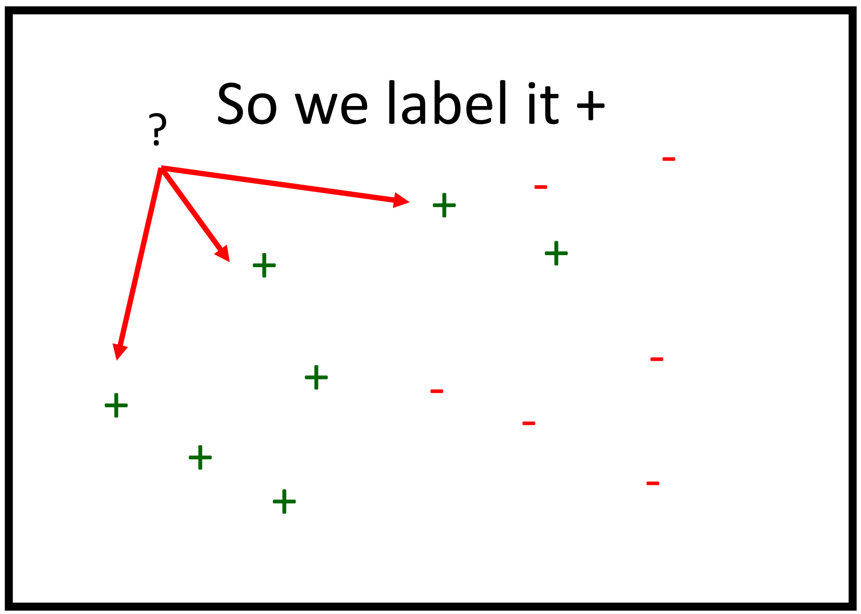


Feature #1 (e.g.., 'area')

k-nearest neighbor (knn)

- Find k nearest neighbors and vote

Feature #2
(e.g.., roundness)



for k=3,

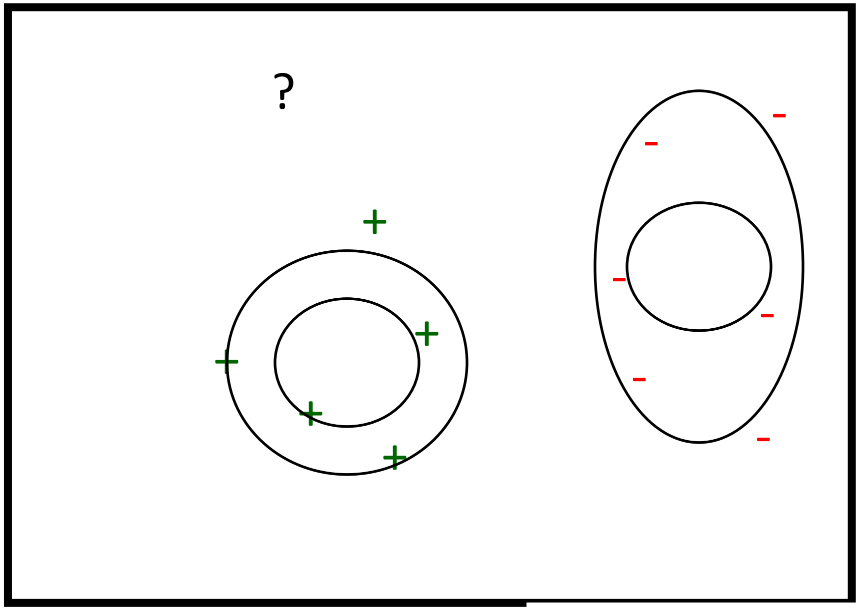
nearest neighbors are

Feature #1 (e.g.., ‘area’)

linear discriminants

- Fit multivariate Gaussian to each class
- Measure distance from “?” to each Gaussian

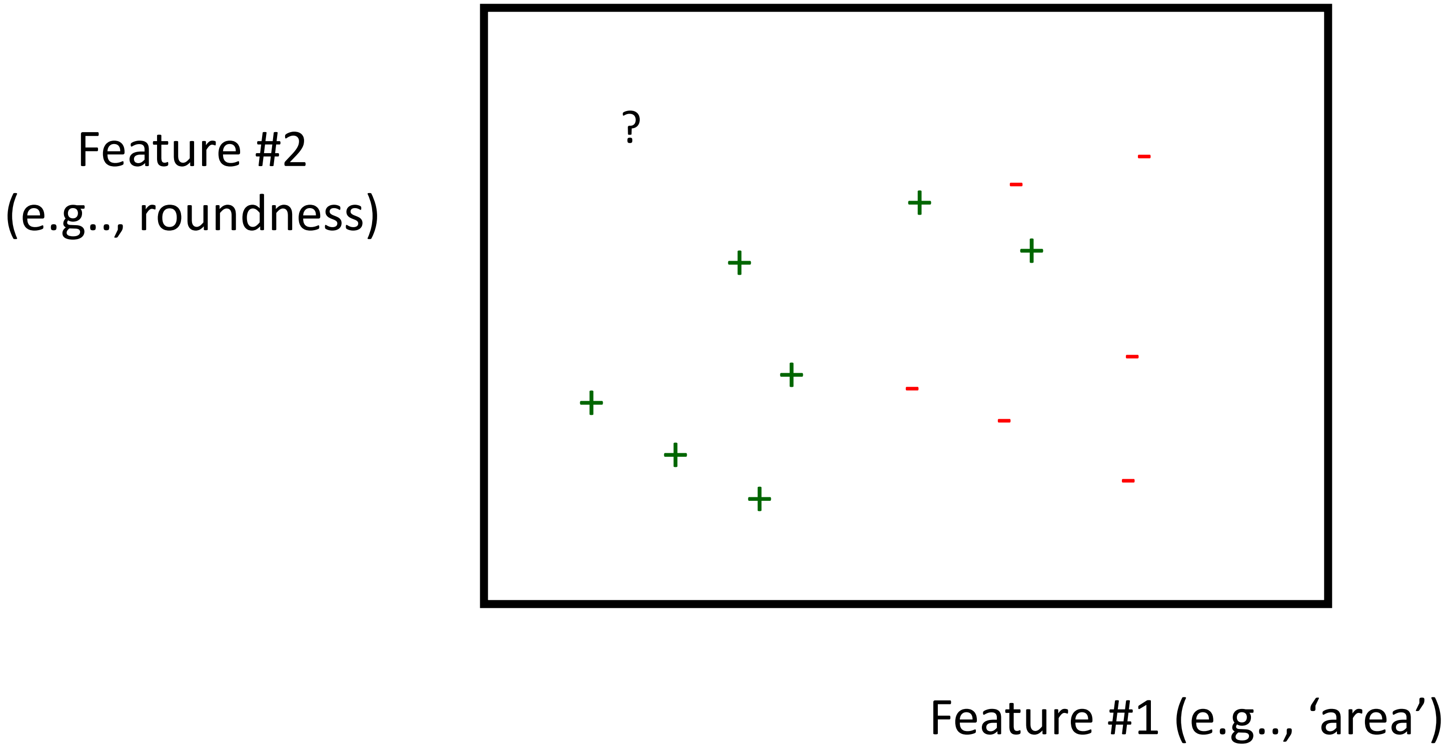
Feature #2
(e.g.., roundness)



Feature #1 (e.g.., ‘area’)

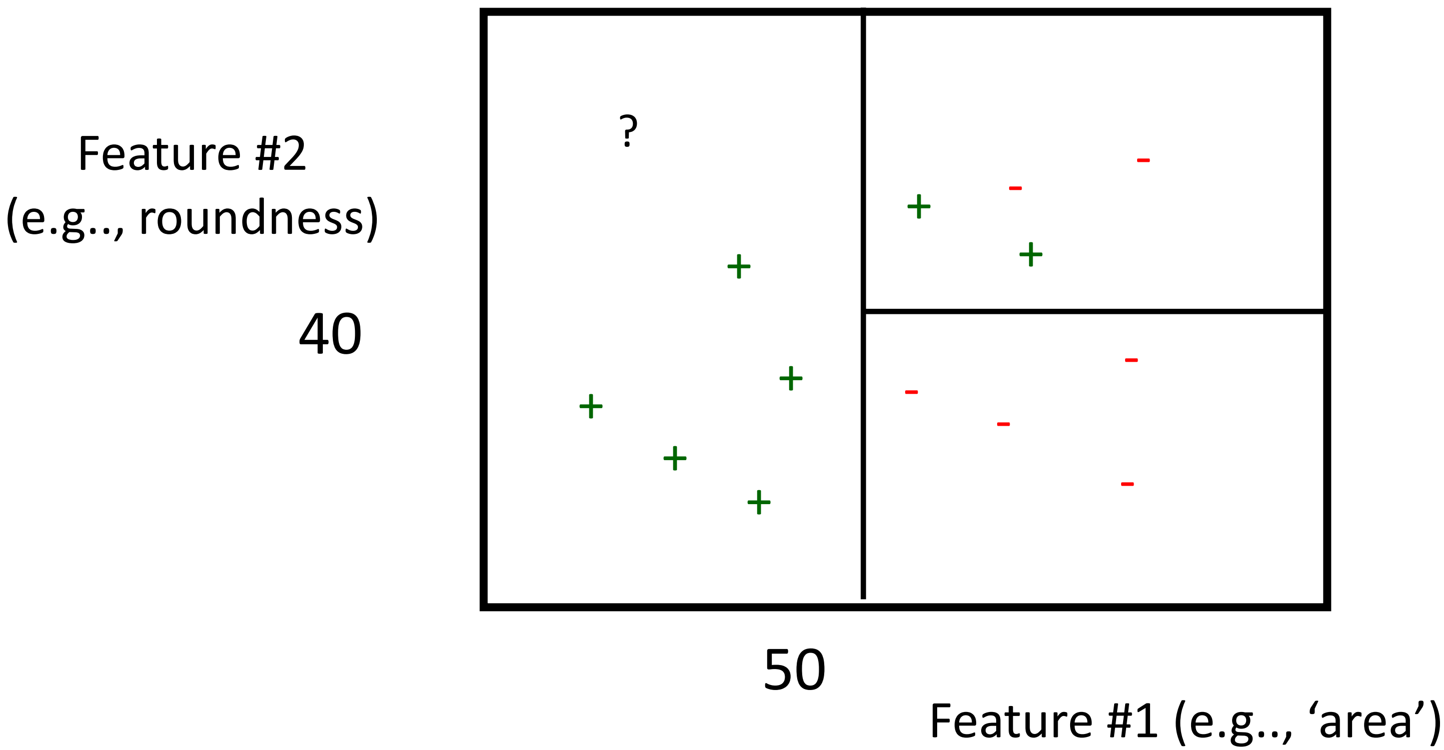
decision trees

- Again we want to label **‘?’**



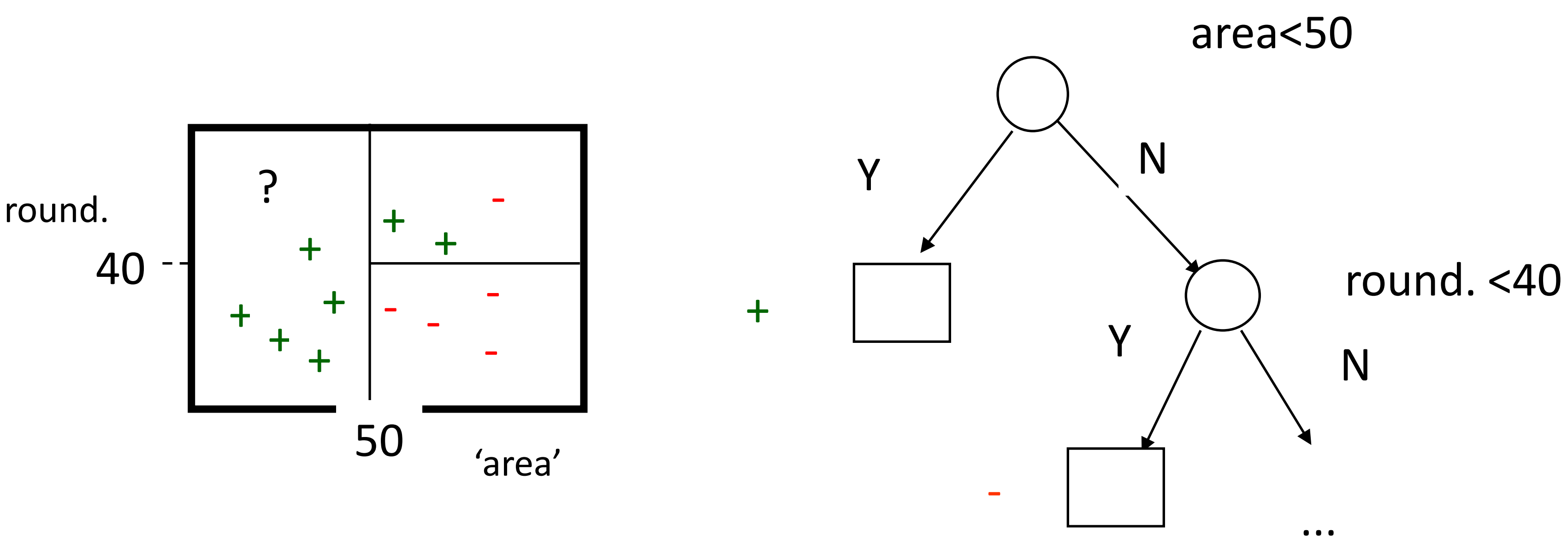
decision trees

- so we build a decision tree:



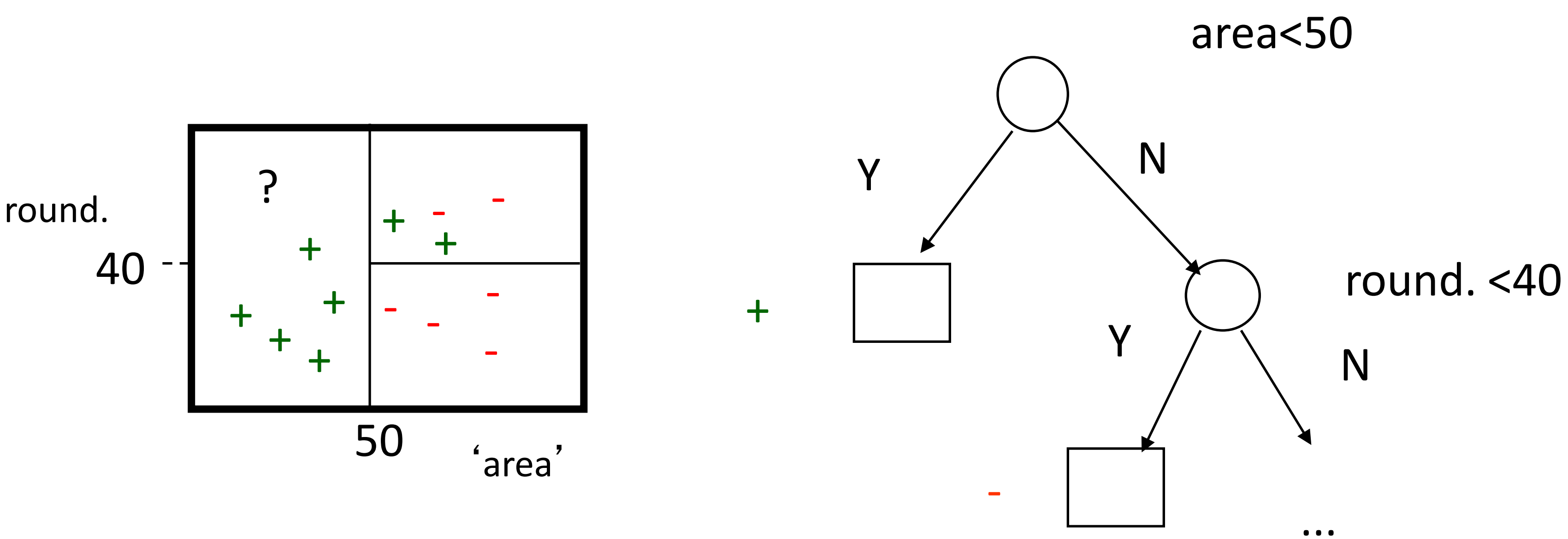
decision trees

- so we build a decision tree:



decision trees

- Goal: split address space in (almost) homogeneous regions



describing classifier errors

- For binary classifiers (positive or negative), define
  - TP = true positives, FP = false positives
  - TN = true negatives, FN = false negatives
  - Recall = TP / (TP + FN)
  - Precision = TP / (TP + FP)
  - F-measure= 2*Recall*Precision/(Recall + Precision)
  - Kappa
  - …

confusion matrix - binary

True \ Predicted	Positive	Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative

confusion matrix – multi-class

True Class	Output of the Classifier									
	DNA	ER	Gia	Gpp	Lam	Mit	Nuc	Act	TfR	Tub
DNA	98	2	0	0	0	0	0	0	0	0
ER	0	100	0	0	0	0	0	0	0	0
Gia	0	0	100	0	0	0	0	0	0	0
Gpp	0	0	0	96	4	0	0	0	0	0
Lam	0	0	0	4	95	0	0	0	0	2
Mit	0	0	2	0	0	96	0	2	0	0
Nuc	0	0	0	0	0	0	100	0	0	0
Act	0	0	0	0	0	0	0	100	0	0
TfR	0	0	0	0	2	0	0	0	96	2
Tub	0	2	0	0	0	0	0	0	0	98

Overall accuracy = 98%

document classification

document classification: problem definition

	d_1	...	...	d_j	...	...	d_n
c_1	a_{11}	...	...	a_{1j}	...	...	a_{1n}
...	...	...	...	...	...	...	...
c_i	a_{i1}	...	...	a_{ij}	...	...	a_{in}
...	...	...	...	...	...	...	...
c_m	a_{m1}	...	...	a_{mj}	...	...	a_{mn}

- Need to assign a boolean value {0,1} to each entry of the decision matrix
- C = {c_1,....., c_m} is a set of pre-defined categories
- D = {d_1,..... d_n} is a set of documents to be categorized
- 1 for a_{ij} : d_j belongs to c_i
- 0 for a_{ij} : d_j does not belong to c_i

flavors of classification

- Single Label
  - For a given d_i at most one (d_i, c_i) is true
  - Train a system which takes a d_i and C as input and outputs a c_i
- Multi-label
  - For a given d_i zero, one or more (d_i, c_i) can be true
  - Train a system which takes a d_i and C as input and outputs C', a subset of C
- Binary
  - Build a separate system for each c_i, such that it takes in as input a d_i and outputs a Boolean value for (d_i, c_i)
  - The most general approach
  - Based on assumption that decision on (d_i, c_i) is independent of (d_i, c_j)

classification methods

- Manual: Typically rule-based
  - Does not scale up (labor-intensive, rule inconsistency)
  - May be appropriate for special data on a particular domain
- Automatic: Typically exploiting machine learning techniques
  - Vector space model based
    - Prototype-based (Rocchio)
    - K-nearest neighbor (KNN)
    - Decision-tree (learn rules)
    - Neural Networks (learn non-linear classifier)
    - Support Vector Machines (SVM)
  - Probabilistic or generative model based
    - Naïve Bayes classifier

steps in document classification

- Classification Process
  - Data preprocessing
    - E.g., Term Extraction, Dimensionality Reduction, Feature Selection, etc.
  - Definition of training set and test sets
  - Creation of the classification model
    - using the selected classification algorithm
  - Classification model validation
  - Classification of new/unknown text documents

the bag of words representation

$\gamma($ I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet. $)=c$

the bag of words representation

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

$) = c$

bag of words representation

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS

BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: grain, wheat

bag of words representation

xxxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxx

xxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds xxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxx tonnes, xxxxxxxxxxxxxxx shipments xxxxxxxxxxx total xxxxxxxx total xxxxxxx xxxxxxxxxxxxxxxxxx:

- Xxxxx wheat xxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxx
- Sorghum xxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

→ Categories: grain, wheat

bag of words representation

xxxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds xxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxx tonnes, xxxxxxxxxxxxxxxxx shipments
xxxxxxxxxxxx total xxxxxxxxx total xxxxxxxx xxxxxxxxxxxxxxxxxxxx:

- Xxxxx wheat xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxxxx
- Sorghum xxxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

word	*freq*
grain(s)	3
oilseed(s)	2
total	3
wheat	1
maize	1
soybean	1
tonnes	1
...	...

Categories: grain, wheat

text classification with naive bayes

- Represent document x as set of (w_i,f_i) pairs:
  - $x = \{(grain,3),(wheat,1),...,(the,6)\}$
- For each y, build a probabilistic model Pr(X|Y=y) of "documents" in class y
  - Pr(X=*{(grain,3),...}*|Y=*wheat*) = ...
  - Pr(X=*{(grain,3),...}*|Y=*nonWheat*) = ...
- To classify, find the *y* which was most likely to *generate x—i.e.,* which gives *x* the best score according to Pr(x|y)
  - $f(x) = \text{argmax}_y \Pr(x/y)*\Pr(y)$

bayes rule

$$\Pr(y \mid x) \cdot \Pr(x) = \Pr(x, y) = \Pr(x \mid y) \cdot \Pr(y)$$

$$\Rightarrow \qquad \Pr(y \mid x) = \frac{\Pr(x \mid y) \cdot \Pr(y)}{\Pr(x)}$$

$$\Rightarrow \quad \arg\max_y \Pr(y \mid x) = \arg\max_y \Pr(x \mid y) \cdot \Pr(y)$$

text classification with naive bayes

- How to estimate Pr(X|Y) ?
- *Simplest useful* process to generate a bag of words:
  - pick word 1 according to Pr(W|Y)
  - repeat for word 2, 3, ...
  - each word is generated *independently* of the others (which is clearly not true) but means

$$\Pr(w_1,\dots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}_{\text{How to estimate Pr(W|Y)?}}$$

text classification with naive bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y = y) = \prod_{i=1}^{n} \Pr(w_i \mid Y = y)$$

Estimate *Pr(w|y)* by looking at the data...

$$\Pr(W = w \mid Y = y) = \frac{\text{count}(W = w \text{ and } Y = y)}{\text{count}(Y = y)}$$

This gives score of zero if x contains a brand-new word w_{new}

text classification with naive bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}$$

... and also imagine *m* examples with *Pr(w|y)=p*

$$\Pr(W=w \mid Y=y) = \frac{\text{count}(W=w \text{ and } Y=y)+mp}{\text{count}(Y=y)+m}$$

text classification with naive bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1,\dots,w_n \mid Y=y)=\prod_{i=1}^{n}\underbrace{\Pr(w_i \mid Y=y)}$$

for instance: *m=1, p=0.5*

$$\Pr(W=w \mid Y=y)=\frac{\mathrm{count}(W=w \text{ and } Y=y)+0.5}{\mathrm{count}(Y=y)+1}$$

text classification with naive bayes

- Putting this together:
  - for each document x_i with label y_i
    - for each word w_{ij} in x_i
      - count[w_{ij}][y_i]++
      - count[y_i]++
      - count++
  - to classify a new $x=w_1 \ldots w_n$, pick y with top *score*:

$$score(y, w_1 \ldots w_k) = \lg \frac{\text{count}[y]}{\text{count}} + \sum_{i=1}^{n} \lg \frac{\text{count}[w_i][y] + 0.5}{\text{count}[y] + 1}$$

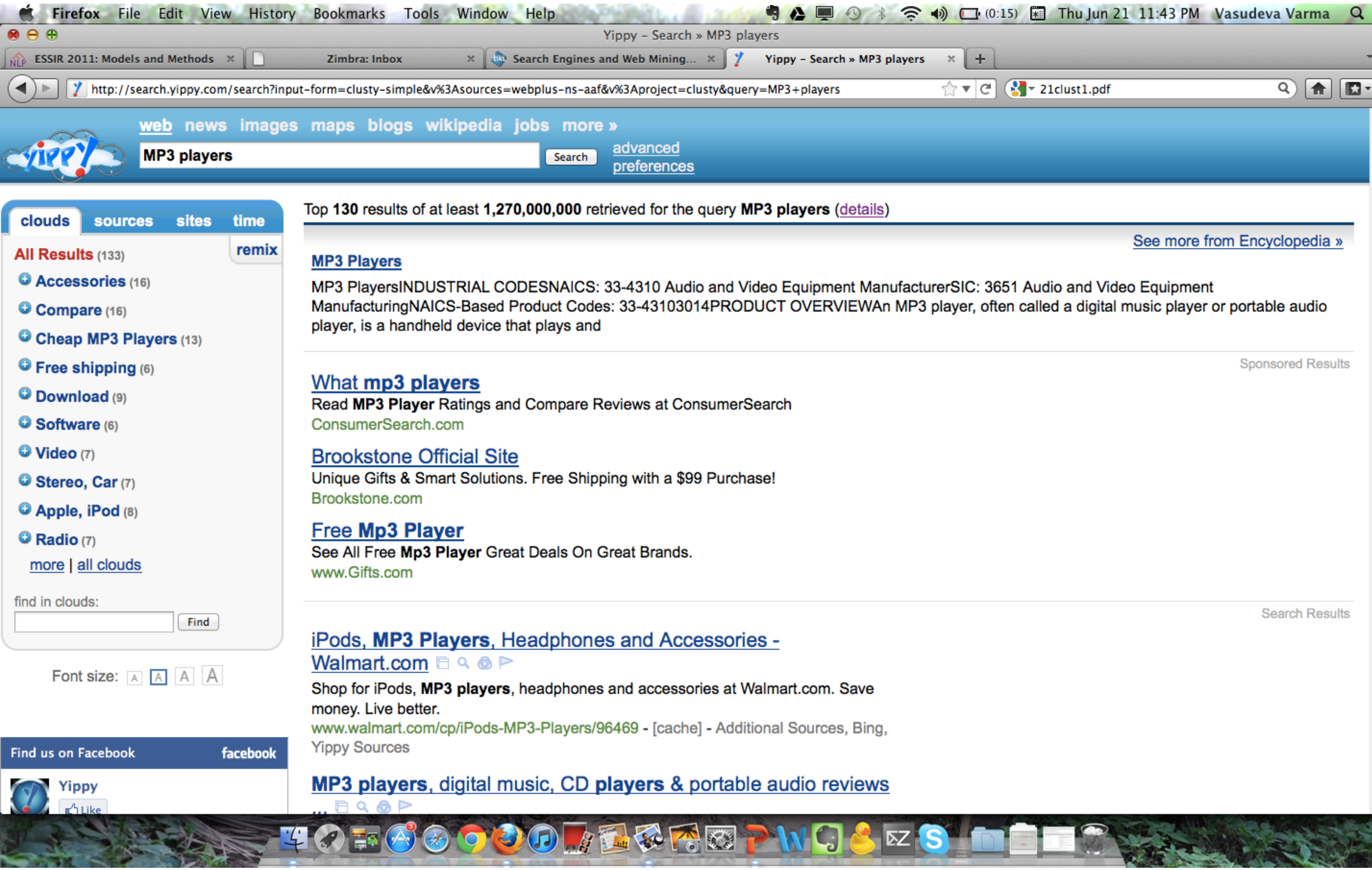
key point: we only need counts for words that actually appear in *x*

naive bayes summary

- Pros:
  - Very fast and easy-to-implement
  - Well-understood formally & experimentally
    - see "Naive (Bayes) at Forty", Lewis, ECML98
- Cons:
  - Seldom gives the very best performance
  - "Probabilities" *Pr(y|x)* are not accurate
    - e.g., Pr(y|x) decreases with length of *x*
    - Probabilities tend to be close to zero or one

unsupervised Learning

Clustering


Firefox File Edit View History Bookmarks Tools Window Help
Thu Jun 21 11:43 PM Vasudeva Varma
Yippy – Search » MP3 players
ESSIR 2011: Models and Methods
Zimbra: Inbox
Search Engines and Web Mining...
Yippy – Search » MP3 players
http://search.yippy.com/search?input-form=clusty-simple&v%3Asources=webplus-ns-aaf&v%3Aproject=clusty&query=MP3+players
21clust1.pdf
web news images maps blogs wikipedia jobs more »
MP3 players
Search
advanced
preferences
clouds sources sites time
remix
All Results (133)
Accessories (16)
Compare (16)
Cheap MP3 Players (13)
Free shipping (6)
Download (9)
Software (6)
Video (7)
Stereo, Car (7)
Apple, iPod (8)
Radio (7)
more | all clouds
find in clouds:
Find
Font size: A A A A
Find us on Facebook facebook
Yippy
Like
Top 130 results of at least 1,270,000,000 retrieved for the query MP3 players (details)
See more from Encyclopedia »
MP3 Players
MP3 PlayersINDUSTRIAL CODESNAICS: 33-4310 Audio and Video Equipment ManufacturerSIC: 3651 Audio and Video Equipment ManufacturingNAICS-Based Product Codes: 33-43103014PRODUCT OVERVIEWAn MP3 player, often called a digital music player or portable audio player, is a handheld device that plays and
Sponsored Results
What mp3 players
Read MP3 Player Ratings and Compare Reviews at ConsumerSearch
ConsumerSearch.com
Brookstone Official Site
Unique Gifts & Smart Solutions. Free Shipping with a $99 Purchase!
Brookstone.com
Free Mp3 Player
See All Free Mp3 Player Great Deals On Great Brands.
www.Gifts.com
Search Results
iPods, MP3 Players, Headphones and Accessories - Walmart.com
Shop for iPods, MP3 players, headphones and accessories at Walmart.com. Save money. Live better.
www.walmart.com/cp/iPods-MP3-Players/96469 - [cache] - Additional Sources, Bing, Yippy Sources
MP3 players, digital music, CD players & portable audio reviews

clustering overview

- Goals:
  - Assign similar objects to the same subset
  - Assign dissimilar objects to different subsets
- Secondary goals
  - Avoid very small and very large clusters
  - Define clusters that are easy to explain to the user
- Object Representation: Features and values
  - Example: Terms and term weights, for documents
  - Example: term co-occurrence, for words
- Similarity Metric:
  - Example: Cosine correlation, for documents

Types of Clustering Algorithms: Flat Vs. Hierarchical

- Hierarchical
  - Preferable for detailed analysis
  - Provides more information than flat
  - No single best algorithm
    - Top down (divisive)
    - Bottom up (Agglomerative)
  - Less efficient
- Flat:
  - Preferable for efficiency
  - K-means is very simple
  - K-means doesn't make sense for some types of data E.g., names

Types of Clustering Algorithms: Hard Vs. Soft

- Soft Vs. Hard
  - Soft: Each Object has a degree of membership in the cluster
    - $P(\text{Cluster}_i \ \text{Object}_j) = x$ where $x \ \varepsilon \ [0..1]$
  - Hard: Each Object is in a cluster or not in a cluster
    - $P(\text{Cluster}_i \ \text{Object}_j) = 0$ or 1

k-means

- Perhaps the best-known clustering algorithm
- Simple, works well in many cases
- Use as default / baseline for clustering documents
- Document representation: Vector space model.

69

k-means

- Each cluster in *K*-means is defined by a centroid.
- Objective/partitioning criterion: minimize the average squared difference from the centroid
- centroid:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

- where we use ω to denote a cluster.
- We try to find the minimum average squared difference by iterating two steps:
  - reassignment: assign each vector to its closest centroid
  - Re-computation: re-compute each centroid as the average of the vectors that were assigned to it in reassignment

k-means algorithm

K-MEANS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

for $k \leftarrow 1$ **to** K

do $\vec{\mu}_k \leftarrow \vec{s}_k$

while stopping criterion has not been met

do for $k \leftarrow 1$ **to** K

    do $\omega_k \leftarrow \{\}$

    for $n \leftarrow 1$ **to** N

    do $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$

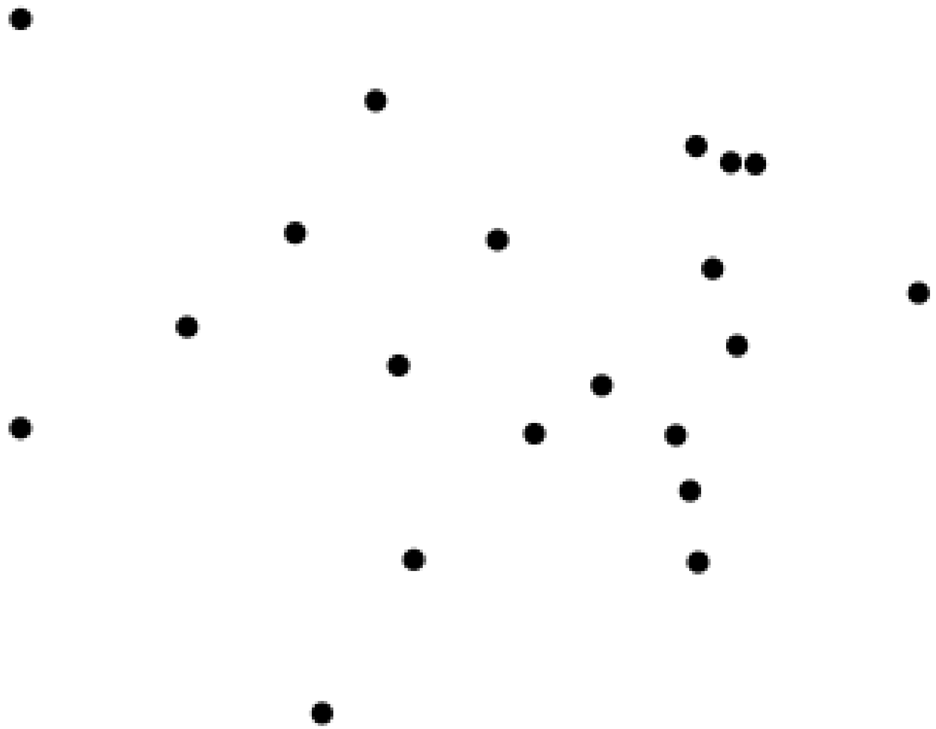
        $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*

    for $k \leftarrow 1$ **to** K

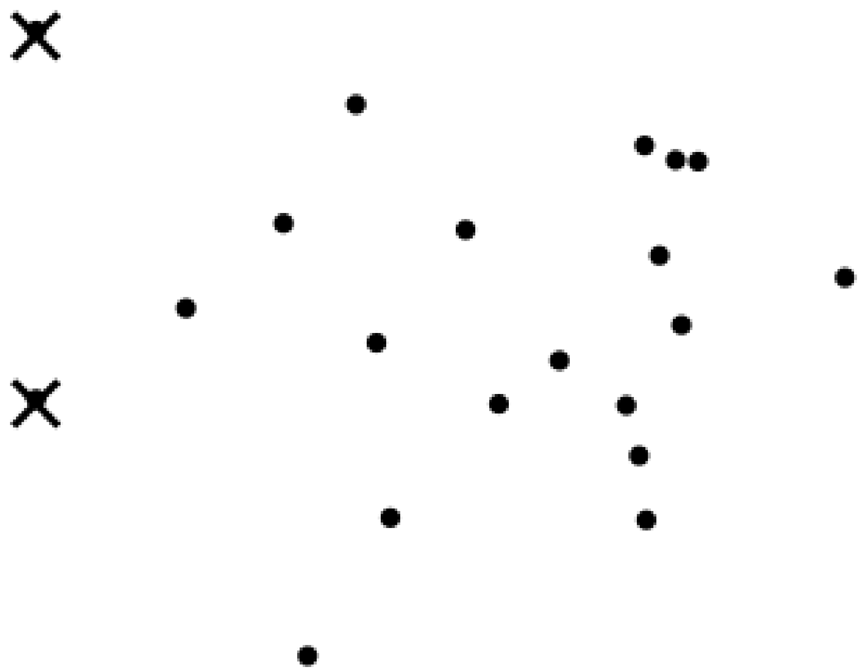
    do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*

return $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

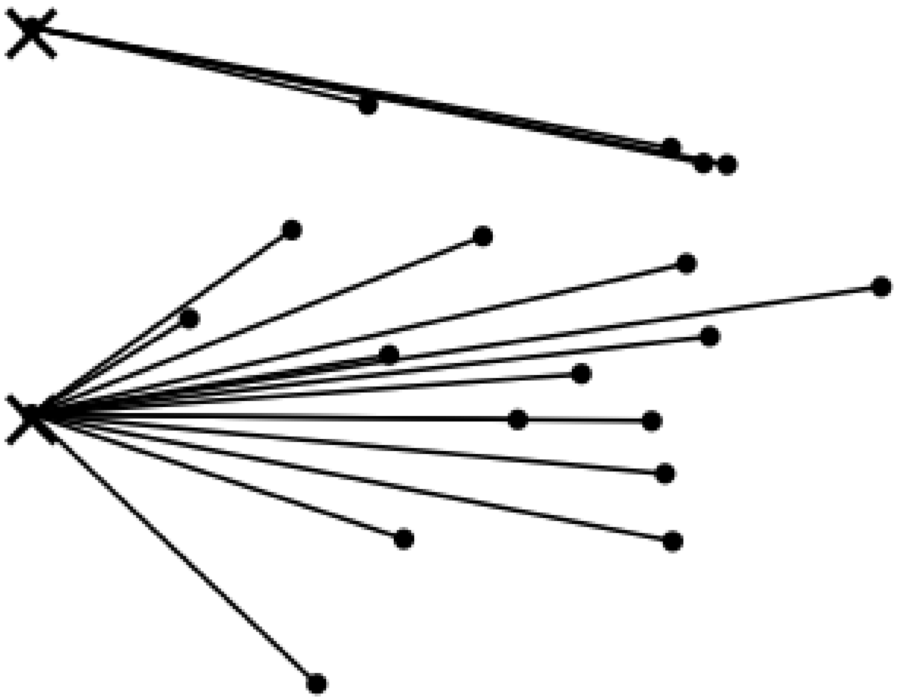
worked Example: set of to be clustered



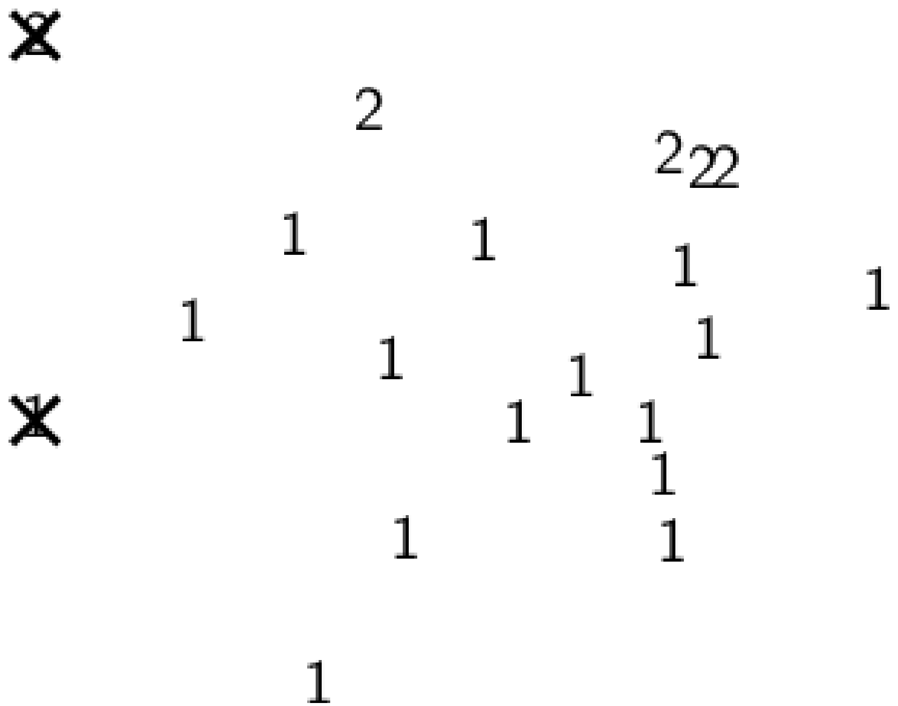
worked Example: random selection of initial centroids



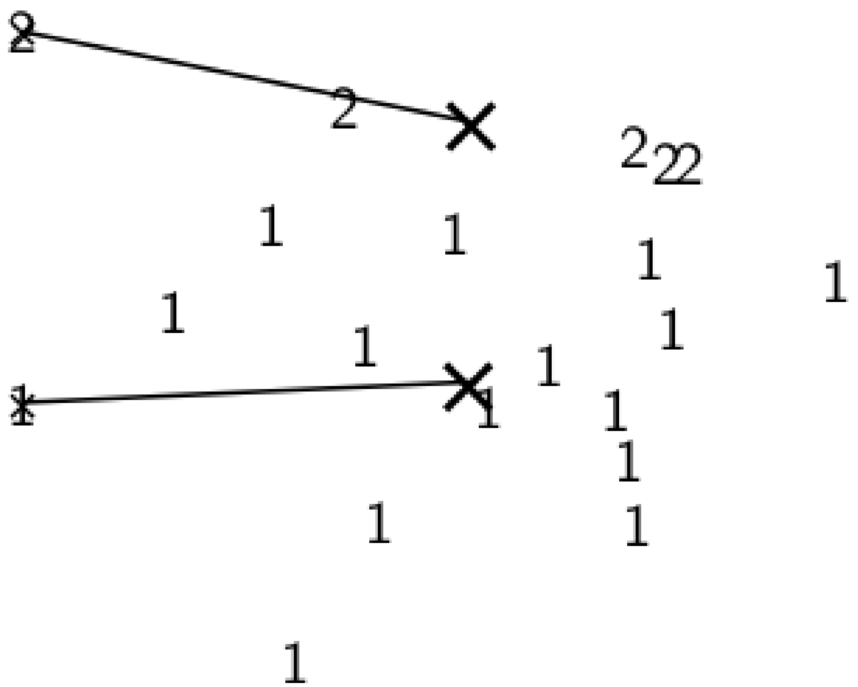
worked Example: Assign points to closest center



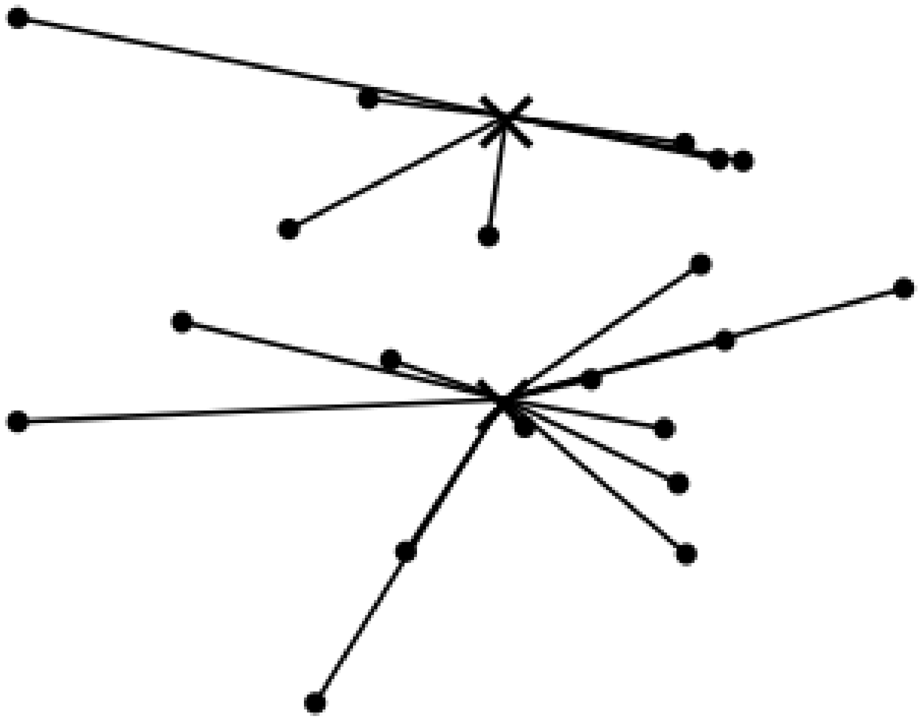
worked Example: Assignment



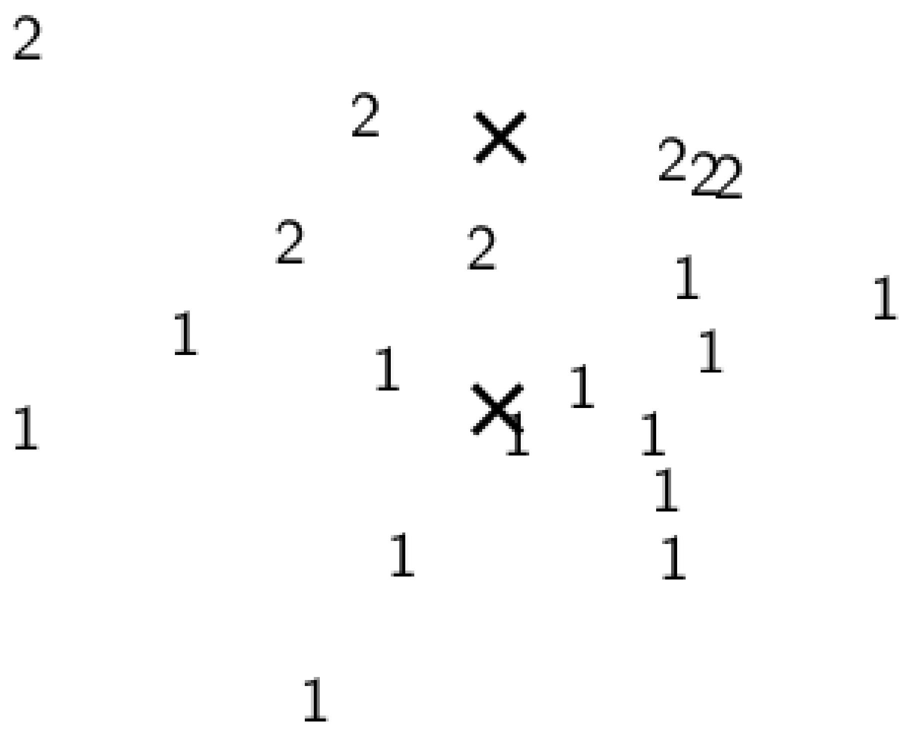
worked Example: Recompute cluster centroids



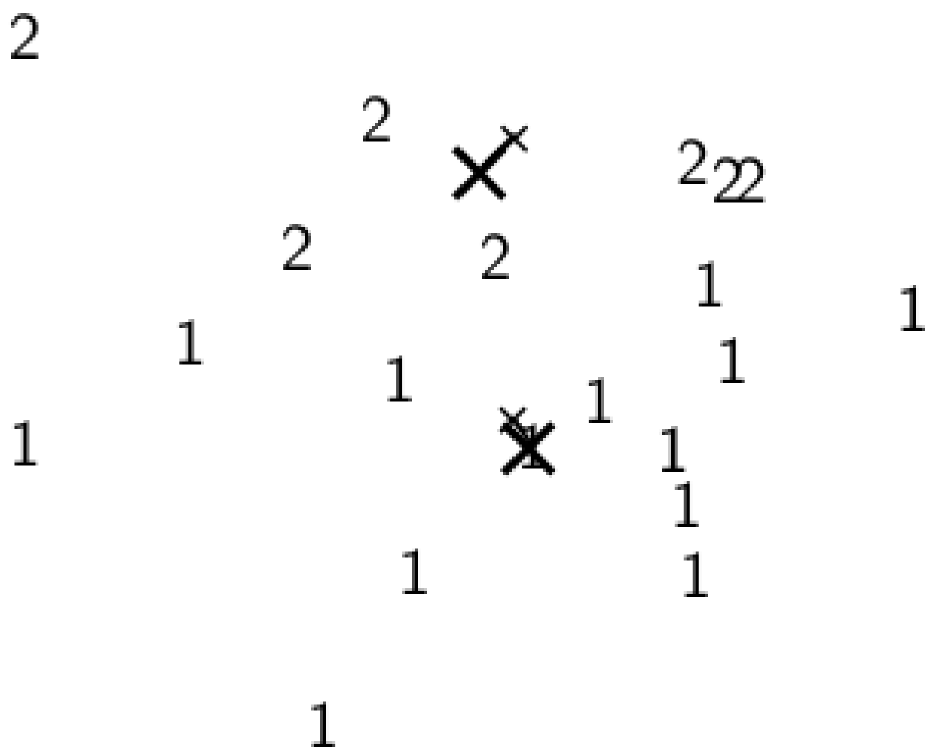
worked Example: Assign points to closest centroid



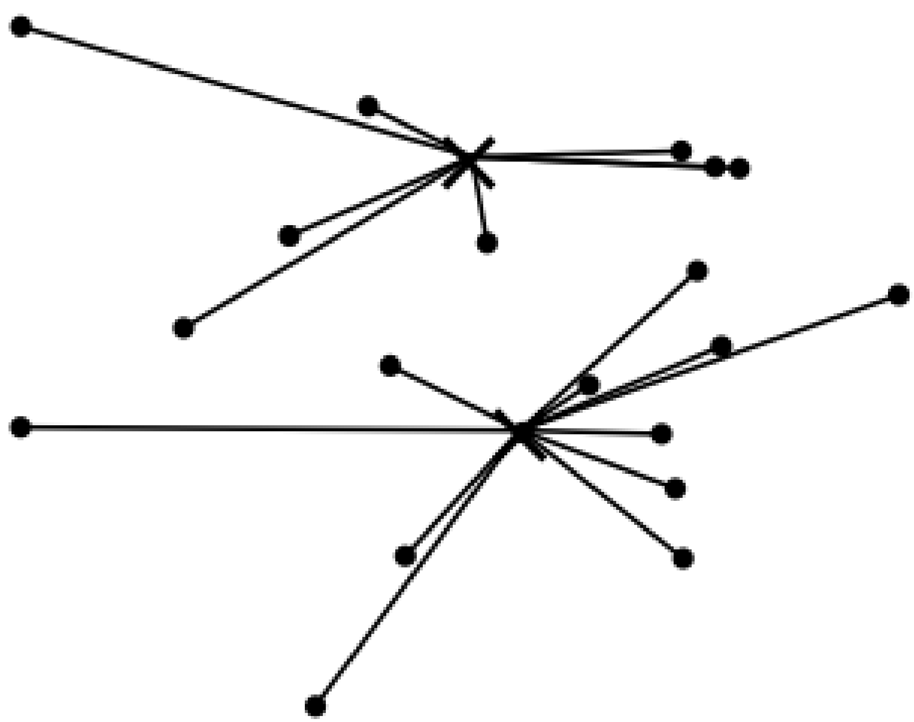
worked Example: assignment



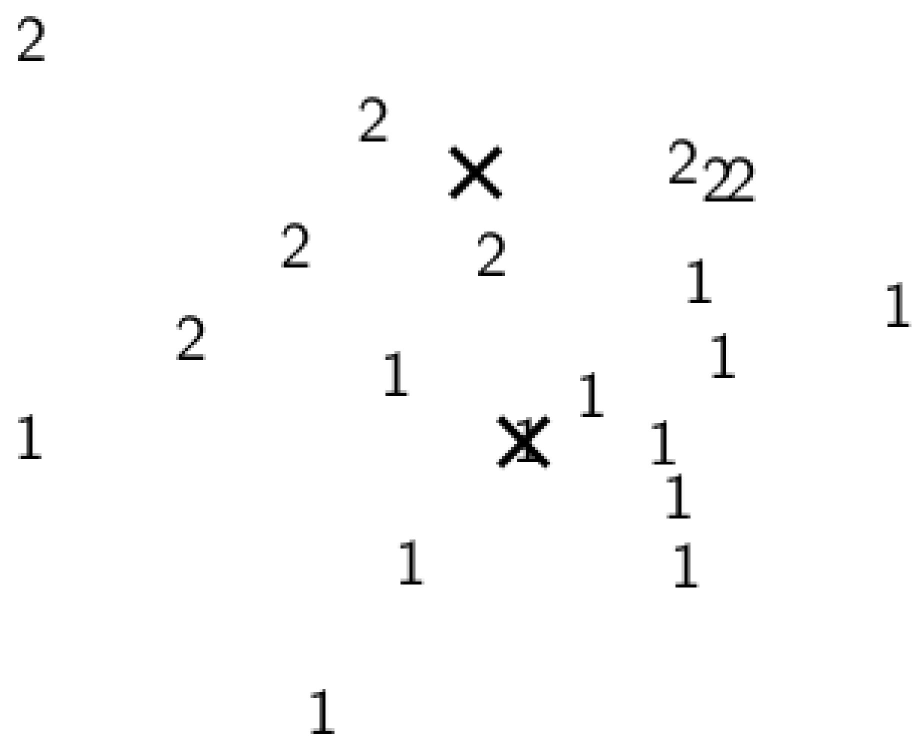
worked Example: recompute cluster centroids



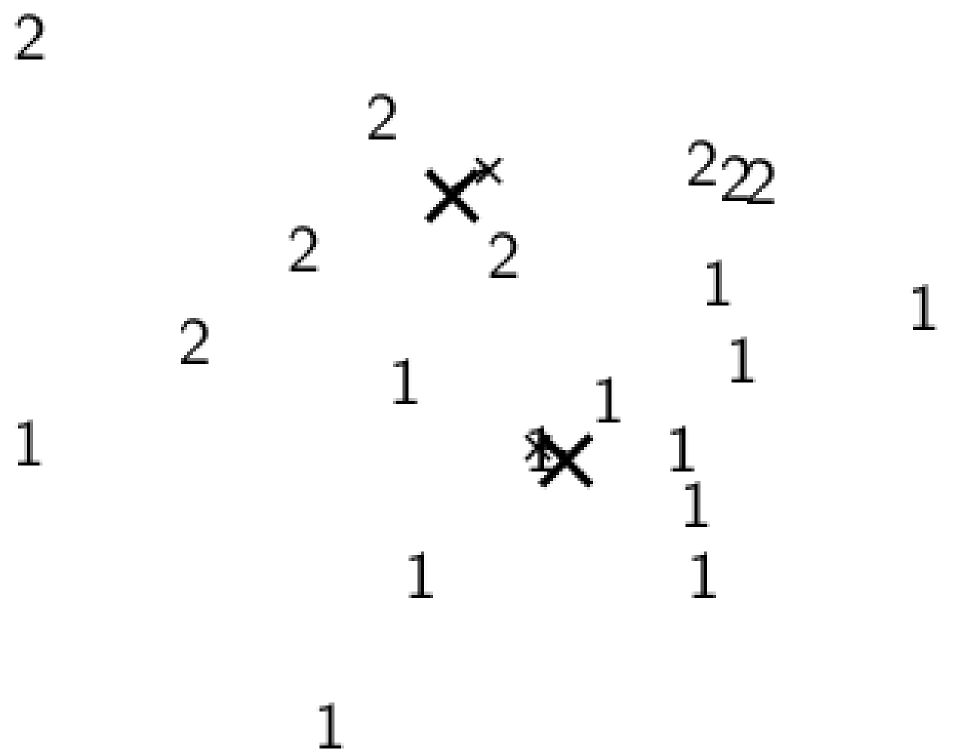
worked Example: assign points to closest centroid



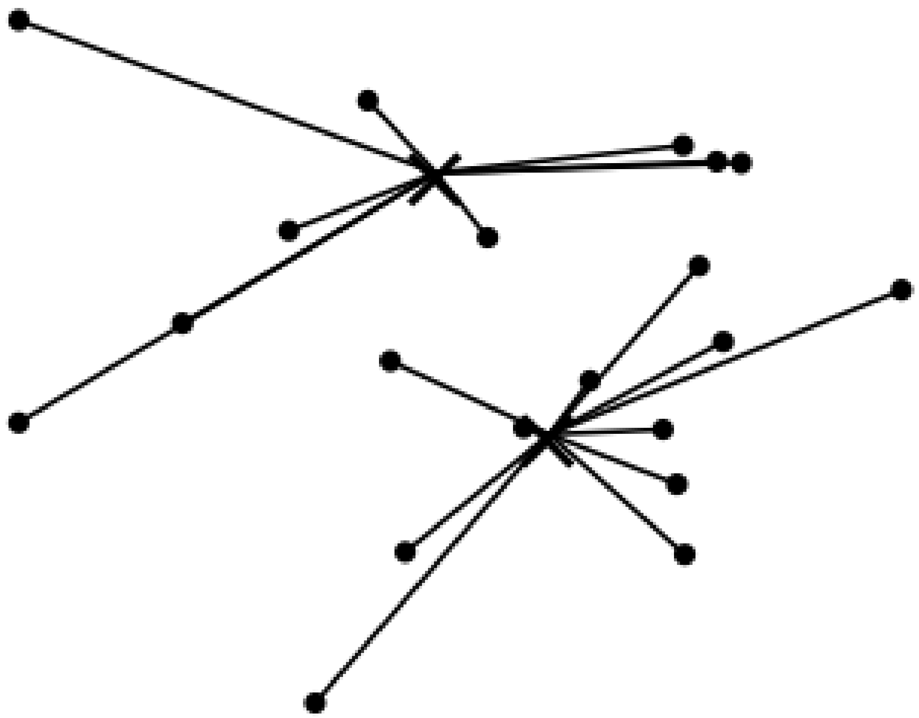
worked example: assignment



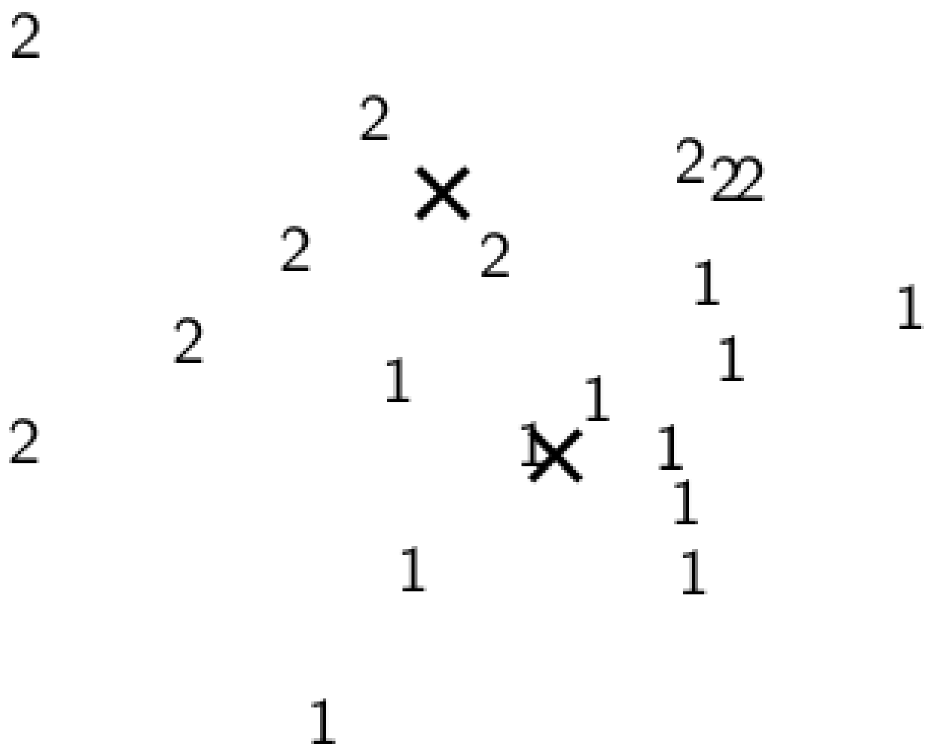
worked Example: recompute cluster centroids



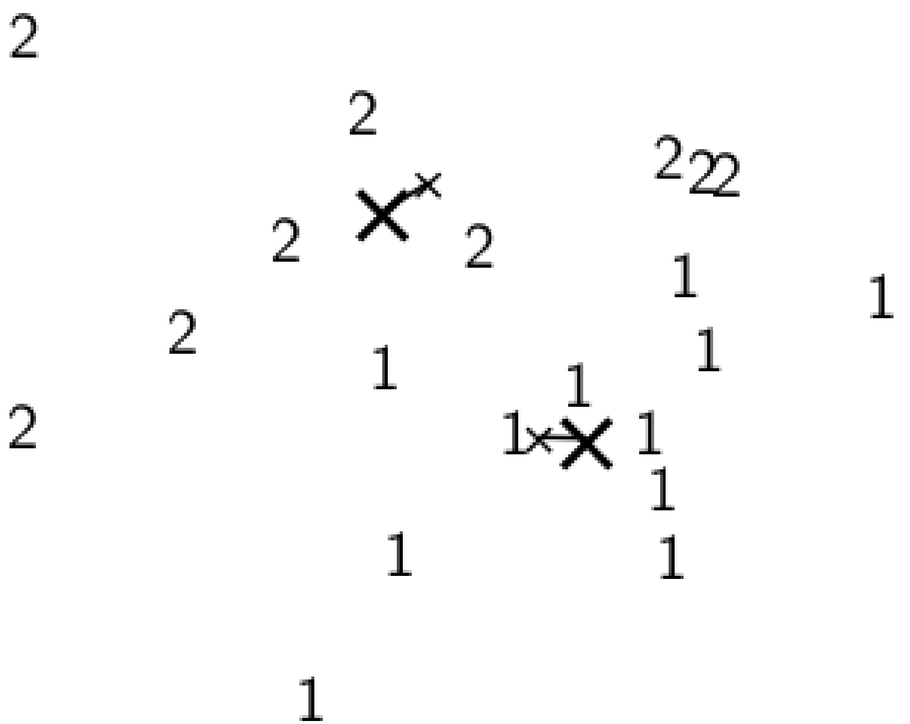
worked Example: assign points to closest centroid



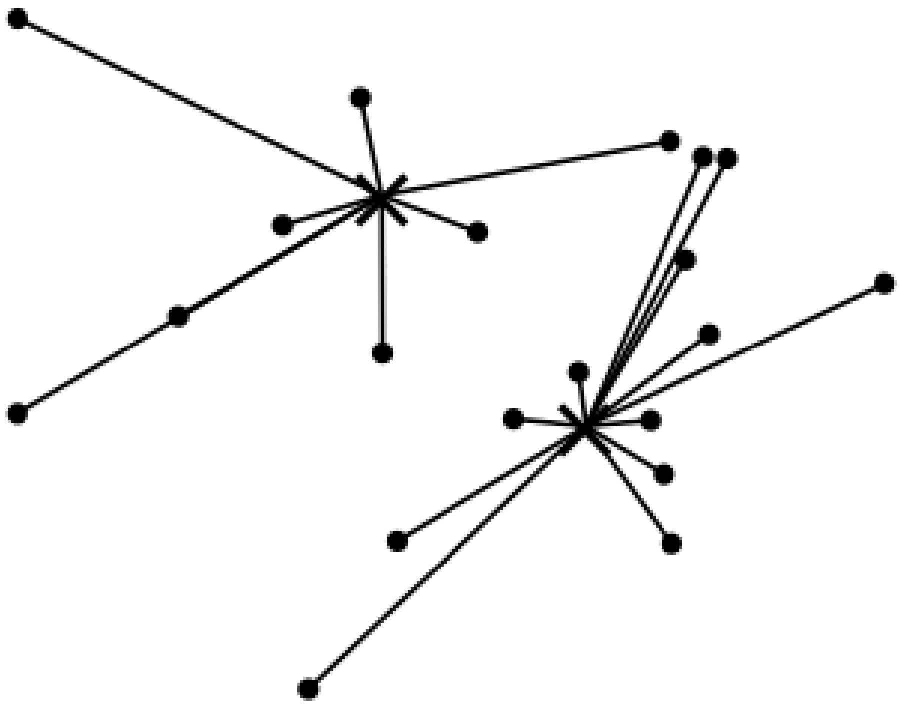
worked example: assignment



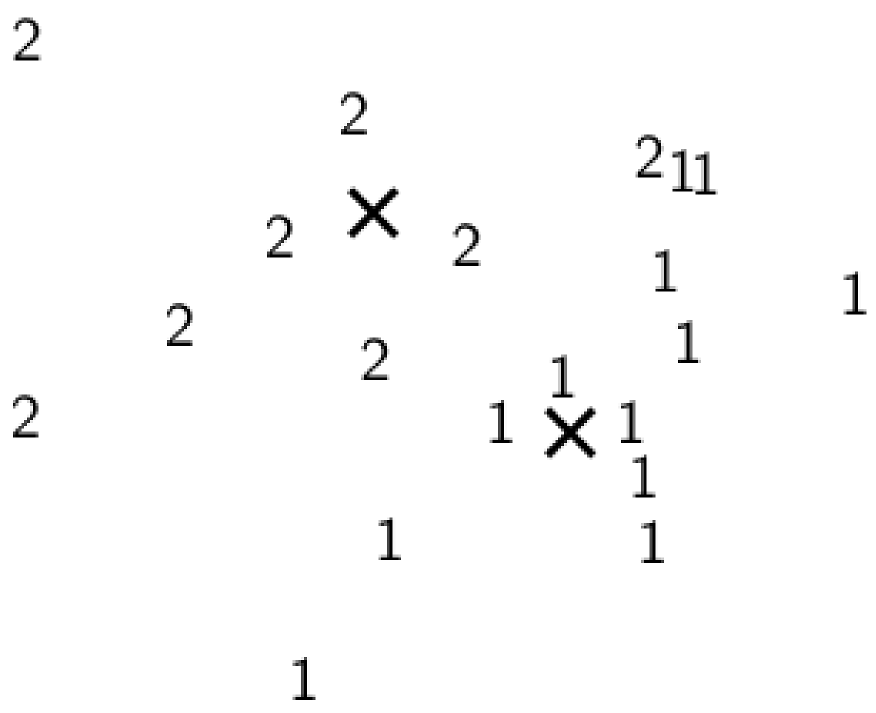
worked example: recompute cluster centroids



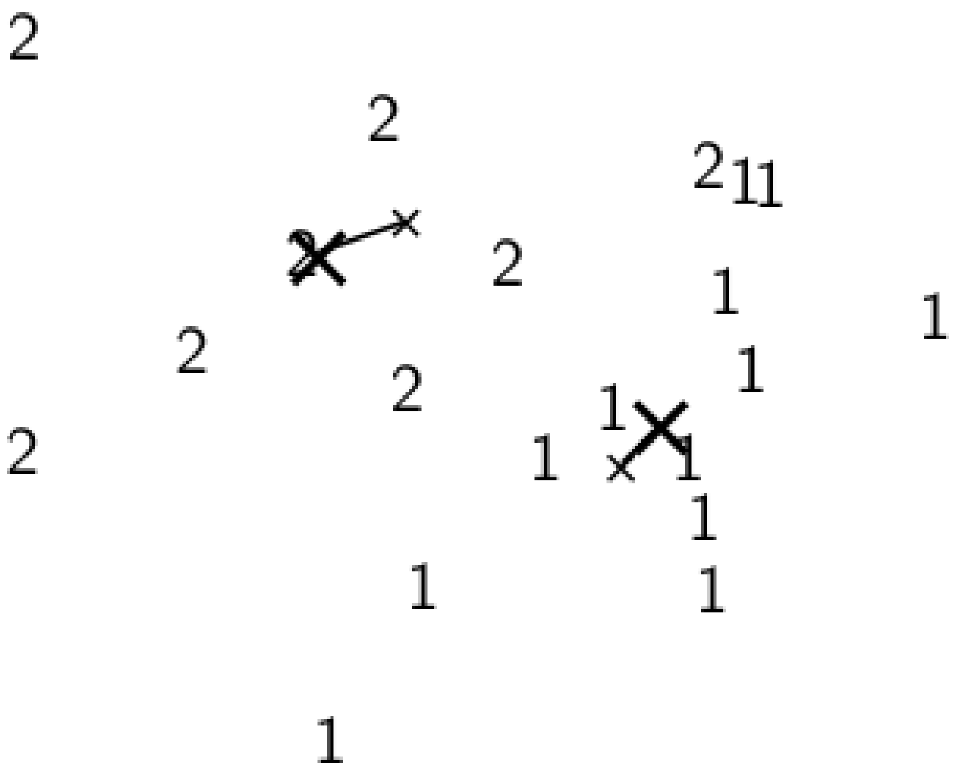
worked Example: assign points to closest centroid



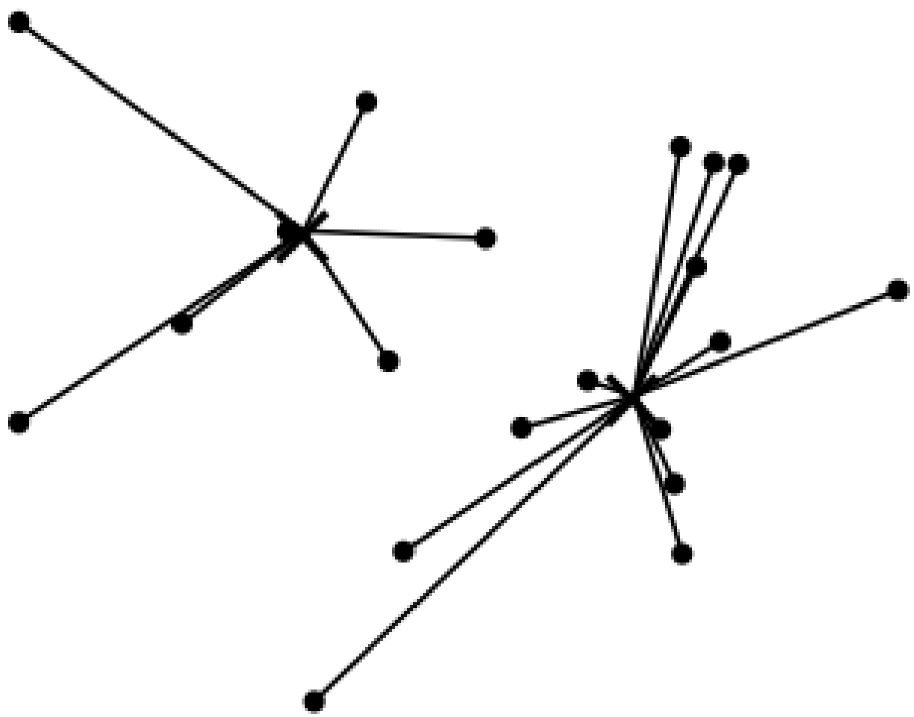
worked example: assignment



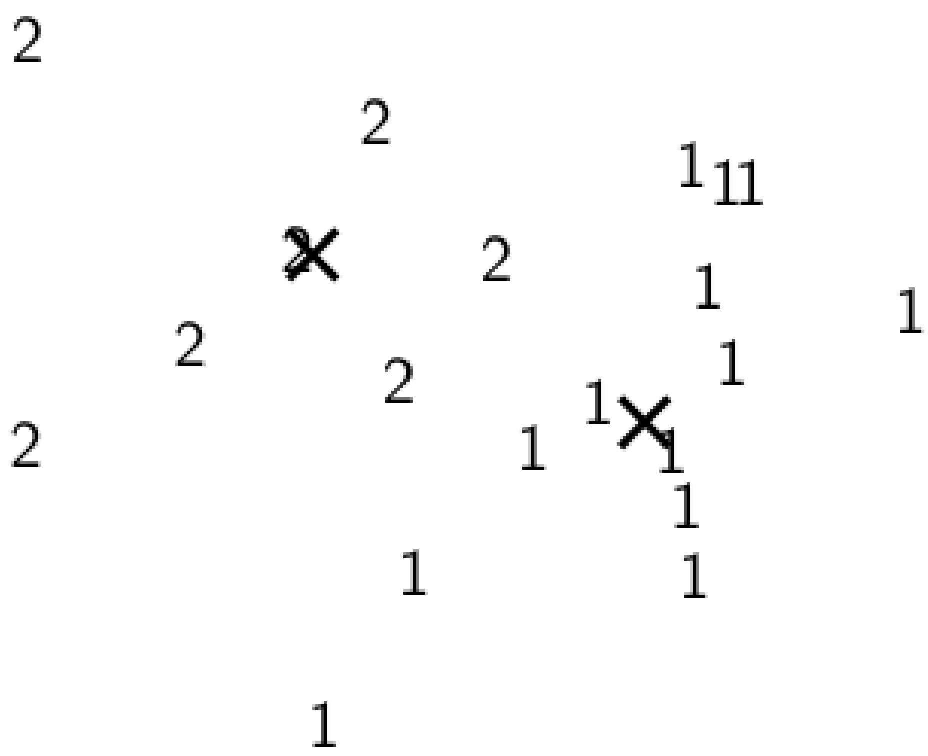
worked example: recompute cluster centroids



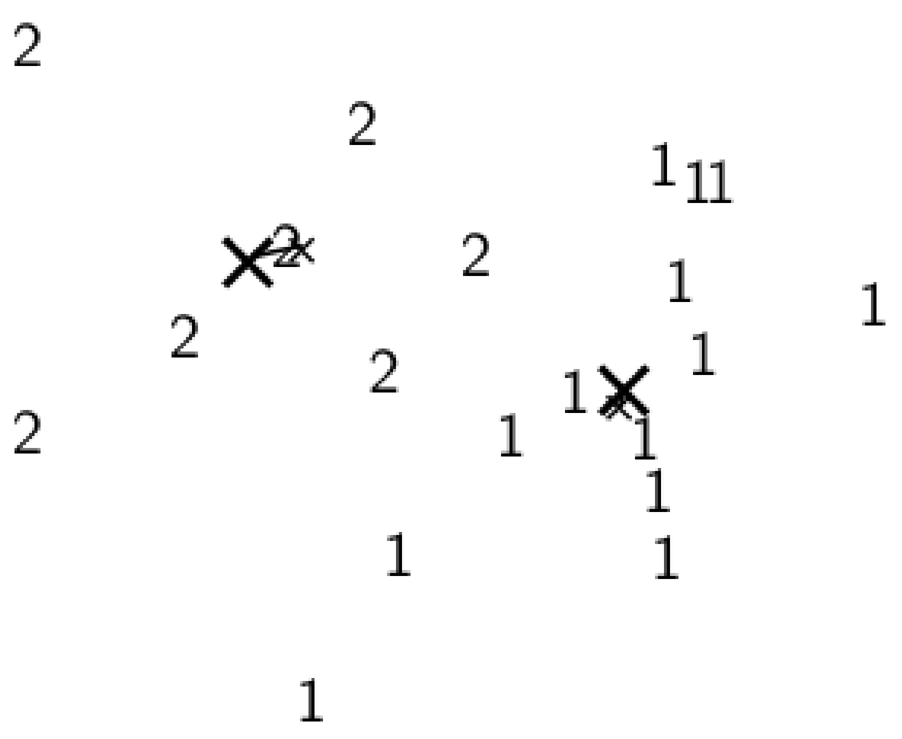
worked example: assign points to closest centroid



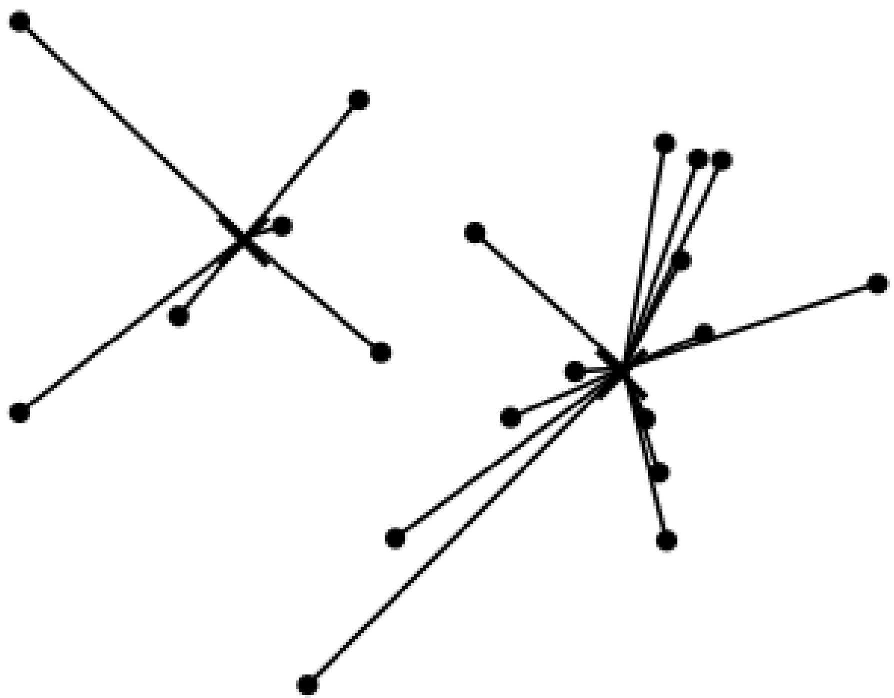
worked example: assignment



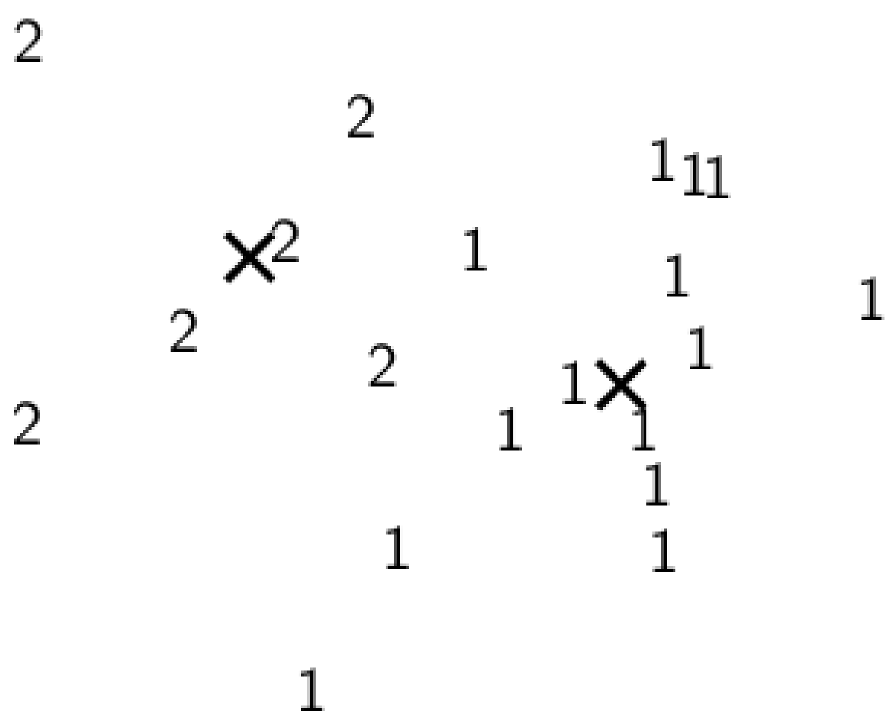
worked example: recompute cluster centroids



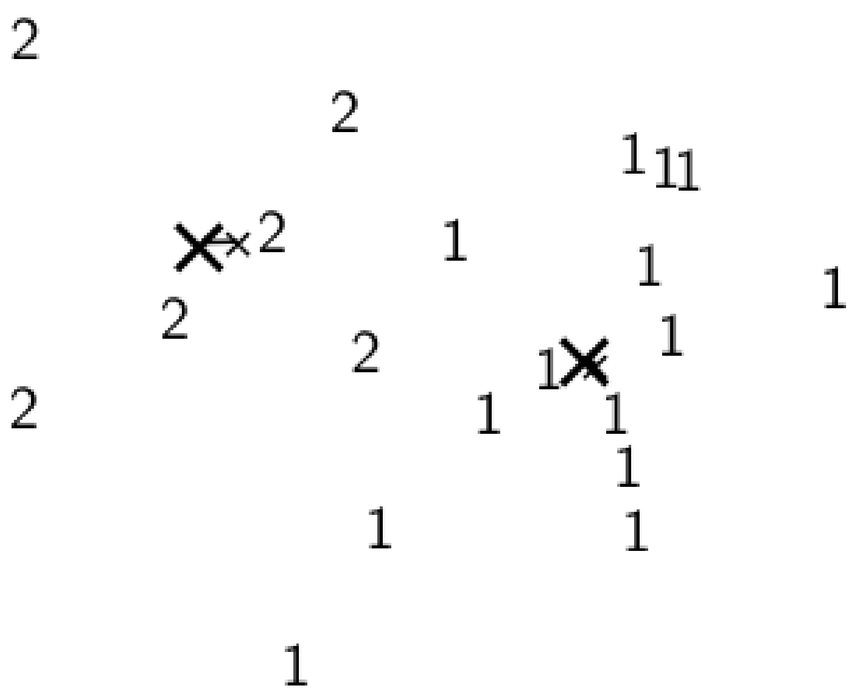
worked example: assign points to closest centroid



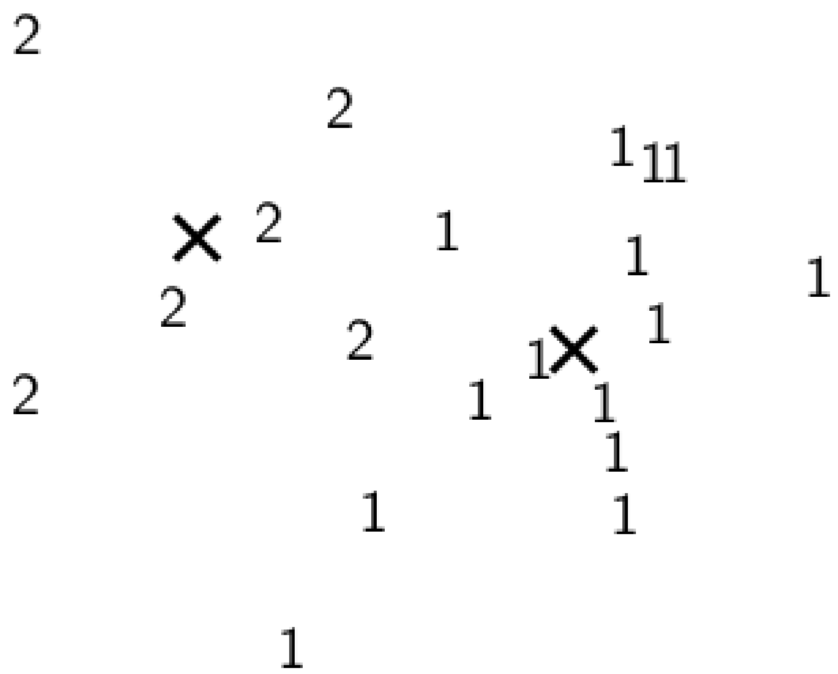
worked example: assignment



worked example: recompute cluster centroids



worked example: centroids and assignments after convergence



convergence

- K-means is guaranteed to converge
  - Proof available
    - RSS = sum of all squared distances between document vector and closest centroid
    - RSS decreases during each reassignment step
    - There is only a finite number of clusters
    - Thus: We must reach a fixed point
- But, Convergence may not be fast
  - But we don’t know how long convergence will take!
  - If we don’t care about a few docs switching back and forth, then convergence is usually fast (< 10-20 iterations).
  - However, complete convergence can take many more iterations
- But, Convergence may not be optimal
  - Convergence does not mean that we converge to the optimal clustering!
  - This is the great weakness of K-means.
  - If we start with a bad set of seeds, the resulting clustering can be horrible

how many clusters?

- Number of clusters K is given in many applications.
  - E.g., there may be an external constraint on K..
- What if there is no external constraint? Is there a "right" number of clusters?
- One way to go: define an optimization criterion
  - Given docs, find K for which the optimum is reached.
  - What optimization criterion can we use?
  - We can't use RSS or average squared distance from centroid as criterion: always chooses $K = N$ clusters.

thank you

questions?