Mining di Dati Web

Lezione 3 - Clustering and Classification

Introduction

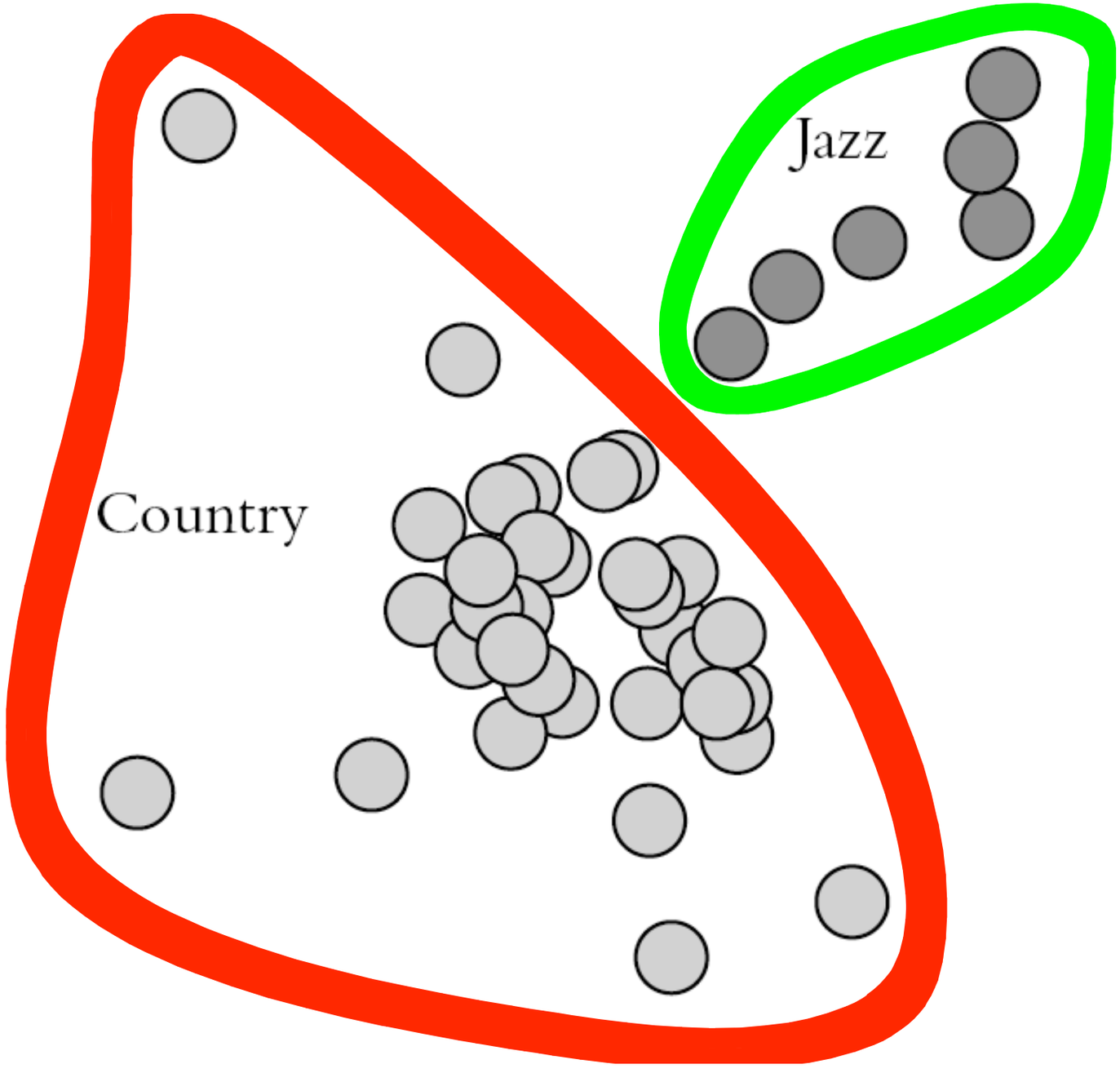
- Clustering and classification are both learning techniques
- They learn functions describing data
- Clustering is also known as Unsupervised Learning
- Classification is known as Supervised Learning

Clustering

- From Wikipedia, Data clustering:

  Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure

Clustering
A Toy Example



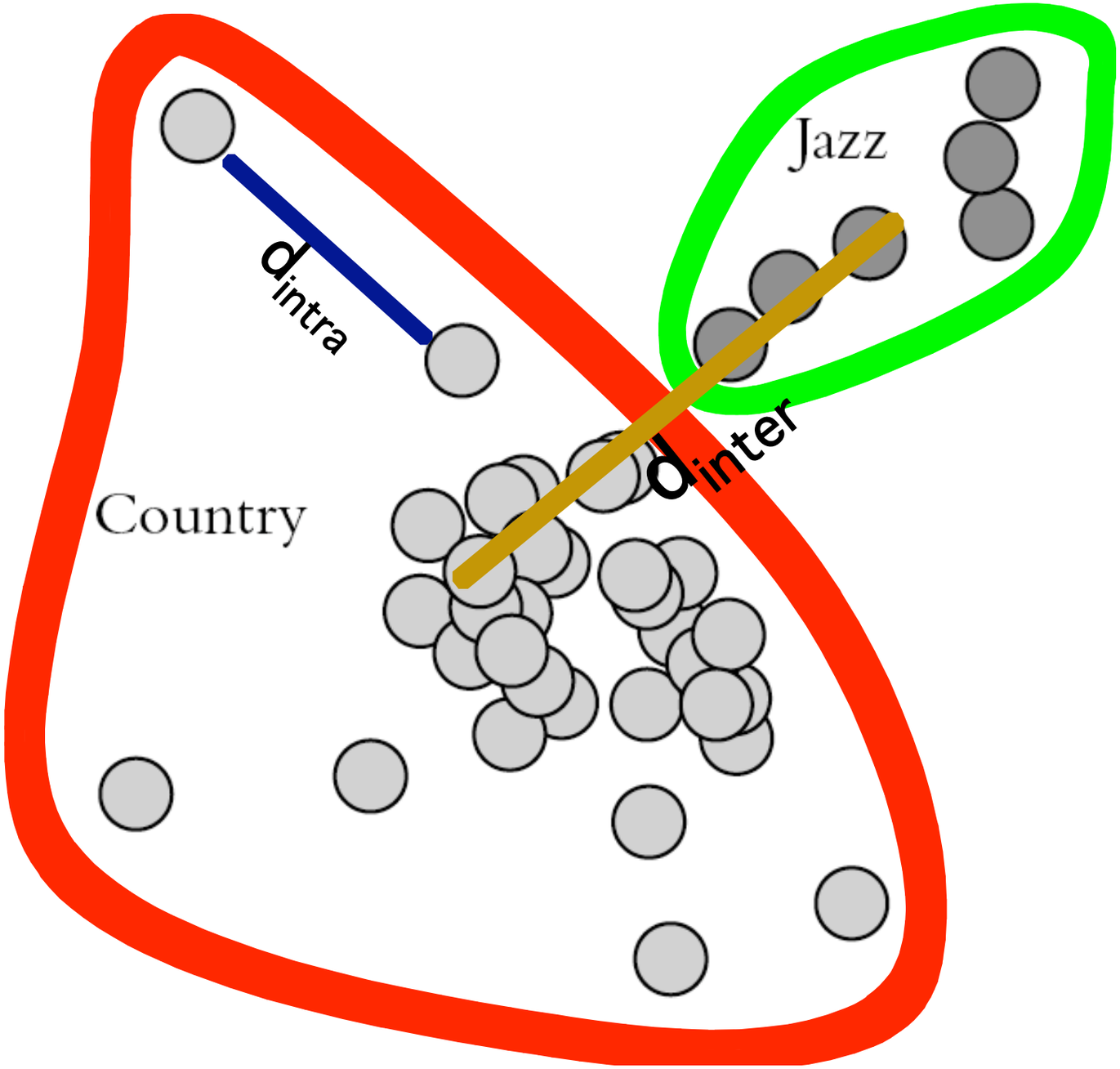
Sneak a Peak at Clustering: K-Means

- Randomly generate k clusters and determine the cluster centers or directly generate k seed points as cluster centers
- Assign each point to the nearest cluster center.
- Recompute the new cluster centers.
- Repeat until some convergence criterion is met (usually that the assignment hasn't changed).

d_intra or d_inter? That is the question!

- d_intra is the distance among elements (points or objects, or whatever) of the same cluster.
- d_inter is the distance among clusters.
- Questions:
  - Should we use distance or similarity?
  - Should we care about inter cluster distance?
  - Should we care about cluster shape?
  - Should we care about clustering?

d_intra, d_inter



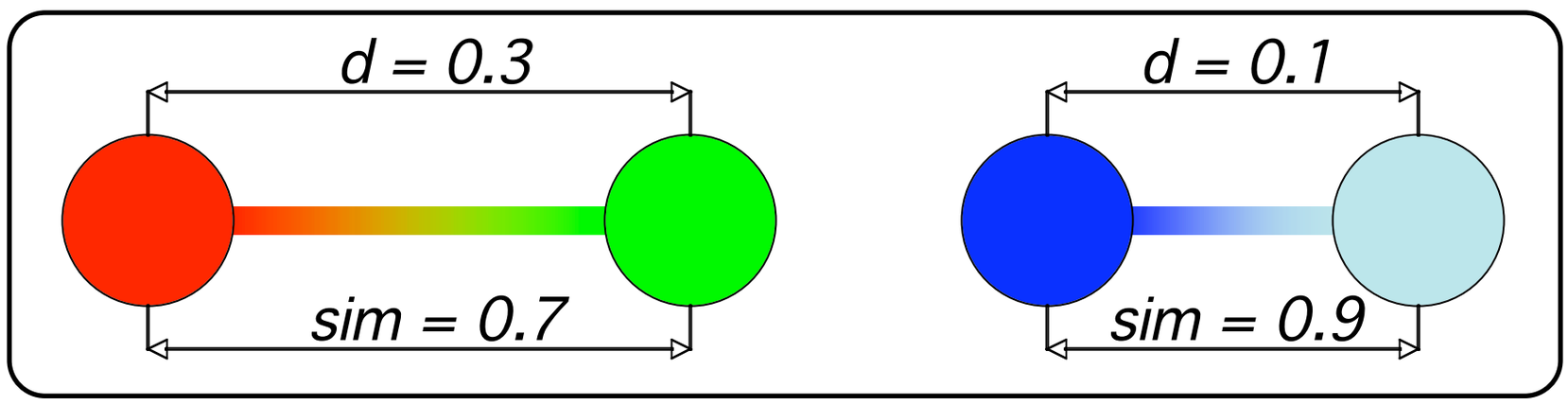
Distance Functions

- Informal definition:
- The distance between two points is the length of a straight line segment between them.
- A more formal definition:
  - A distance between two points P and Q in a metric space is d(P,Q), where d is the distance function that defines the given metric space.
  - We can also define the distance between two sets A and B in a metric space as being the minimum (or infimum) of distances between any two points P in A and Q in B.

Distance or Similarity?

- In a very straightforward way we can define the Similarity function
  sim: SxS => [0,1] as sim(o_1,o_2) = 1-d(o1,o2)

where o_1, o_2 are elements of the space S.



What does similar (or distant) really mean?

- Learning (either supervised or unsupervised) is impossible without ASSUMPTIONS
  - Watanabe's Ugly Duckling theorem
    - S. Watanabe, (1969). Knowing and Guessing: A Quantitative Study of Inference and Information. NY: Wiley.
  - Wolpert's No Free Lunch theorem
    - D.H. Wolpert, (2001). The supervised learning no-free-lunch theorems. Proc. 6th Online World Conference on Soft Computing.
- Learning is impossible without some sort of bias.

The Ugly Duckling theorems

- The theorem gets its fanciful name from the following counter-intuitive statement: assuming similarity is based on the number of shared predicates, an ugly duckling is as similar to a beautiful swan A as a beautiful swan B is to A, given that A and B differ at all.

- It was proposed and proved by Satoshi Watanabe in 1969.

Watanabe's Theorem

- Let n be the cardinality of the Universal set S.
- We have to classify them without prior knowledge on the essence of categories.
- The number of different classes, i.e. the different way to group the objects into clusters, is given by the cardinality of the Power Set of S:
  - $|Pow(S)|=2^n$
- Without any prior information, the most natural way to measure the similarity among two distinct objects is to count the number of classes they share.
- Oooops… They share exactly the same number of classes, namely 2^{n-2}.

The Ugly Duckling and 3 Beautiful Swans

- S={o_1,o_2,o_3,o_4}
- Pow(S) = { {}, {o_1}, {o_2}, {o_3}, {o_4},
  {o_1, o_2}, {o_1, o_3}, {o_1, o_4},
  {o_2, o_3}, {o_2, o_4}, {o_3, o_4},
  {o_1, o_2, o_3}, {o_1, o_2, o_4},
  {o_1, o_3, o_4}, {o_2, o_3, o_4},
  {o_1, o_2, o_3, o_4} }
- How many classes have in common o_i, o_j i<>j?
- o_1 and o_3 ===> 4 = 2^2
- o_1 and o_4 ===> 4 = 2^2

The Ugly Duckling and 3 Beautiful Swans

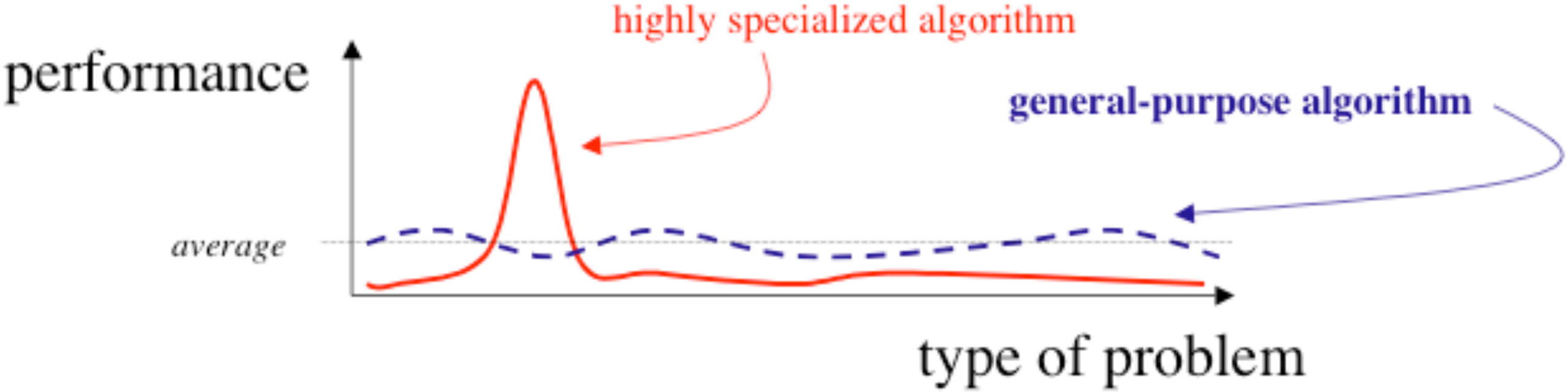
- In binary 0000, 0001, 0010, 0100, 1000,…
- Choose two objects
- Reorder the bits so that the chosen objects are represented by the first two bits.
- How many strings share the first two bits set to 1?
  - 2^{n-2}

Wolpert's No Free Lunch Theorem

- For any two algorithms, A and B, there exist datasets for which algorithm A outperform algorithm B in prediction accuracy on unseen instances.

Proof: Take any Boolean concept. If A outperforms B on unseen instances, reverse the labels and B will outperforms A.

No Free Lunch Theorem



So Let's Get Back to Distances

- In a metric space a distance is a function d:SxS=>R so that if a,b,c are elements of S:
  - $d(a,b) \geq 0$
  - $d(a,b) = 0$ iff $a=b$
  - $d(a,b) = d(b,a)$
  - $d(a,c) \leq d(a,b) + d(b,c)$
- The fourth property (triangular inequality) holds only if we are in a metric space.

Minkowski's Distance

- Let's consider two elements of a set S described by their feature vectors:
- $x=(x_1, x_2, \ldots, x_n)$
- $y=(y_1, y_2, \ldots, y_n)$
- The Minkowski's Distance is parametric in $p>1$

$$d^{(p)}(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Manhattan Distance (p=1)

- If p = 1 the distance is called Manhattan Distance.

$$d^{(1)}(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

- It is also called taxicab distance because it is the distance a car would drive in a city laid out in square blocks (if there are no one-way streets).

Euclidean Distance (p=2)

- If p = 2 the distance is the well know Euclidean Distance.

$$d^{(2)}(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

Chebyshev Distance (p=infinity)

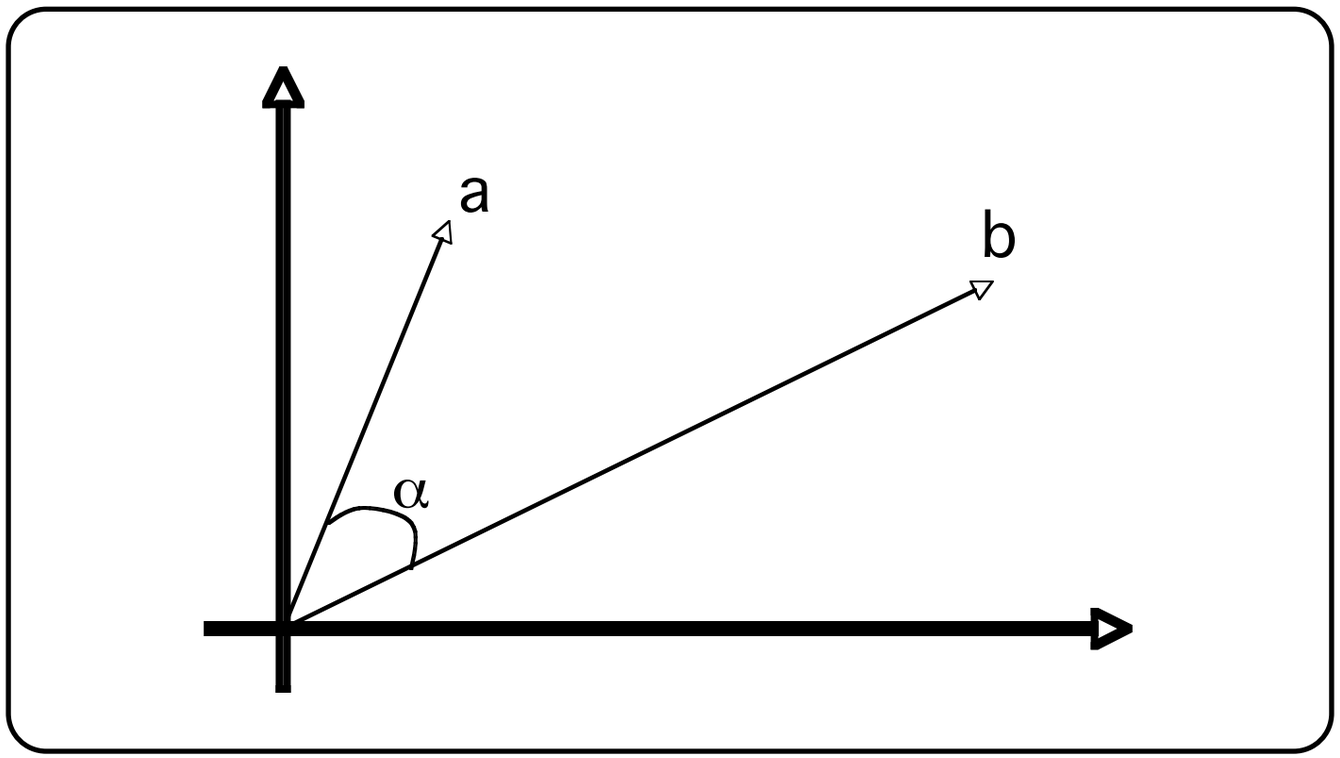
- If p = infinity then we must take the limit, the distance is called Chebyshev Distance.

$$d^{(\infty)}(x,y) = \lim_{p \to \infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} =$$

$$= \max\left\{ |x_1 - y_1|, |x_2 - y_2|, \ldots |x_n - y_n| \right\}$$

2D Cosine Similarity

- It's easy to explain in 2D.
- Let's consider $a=(x_1,y_1)$ and $b=(x_2,y_2)$.



Cosine Similarity

- Let's consider two points x, y in $\mathbf{R}^n$.

$$sim(x,y) = \cos\alpha = \frac{x \bullet y}{|x||y|} =$$

$$= \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^{\,2}} \sqrt{\sum_{i=1}^{n} y_i^{\,2}}}$$

Jaccard Distance

- Another commonly used distance is the Jaccard Distance:

$$d(x,y) = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i)}$$

Binary Jaccard Distance

- In the case of binary feature vector the Jaccard Distance could be simplified to:

$$d(x,y) = \frac{|x \cap y|}{|x \cup y|}$$

Binary Edit Distance

- The binary edit distance, d(x,y), from a binary vector x to a binary vector y is the minimum number of simple flips required to transform one vector to the other

$$d(x,y) = \sum_{i=1}^{n} \left(1 - x_1 y_i\right)$$

x=(0,1,0,0,1,1)
y=(1,1,0,1,0,1)

d(x,y)=3

- The binary edit distance is equivalent to the Manhattan distance (Minkowski p=1) for binary features vectors.

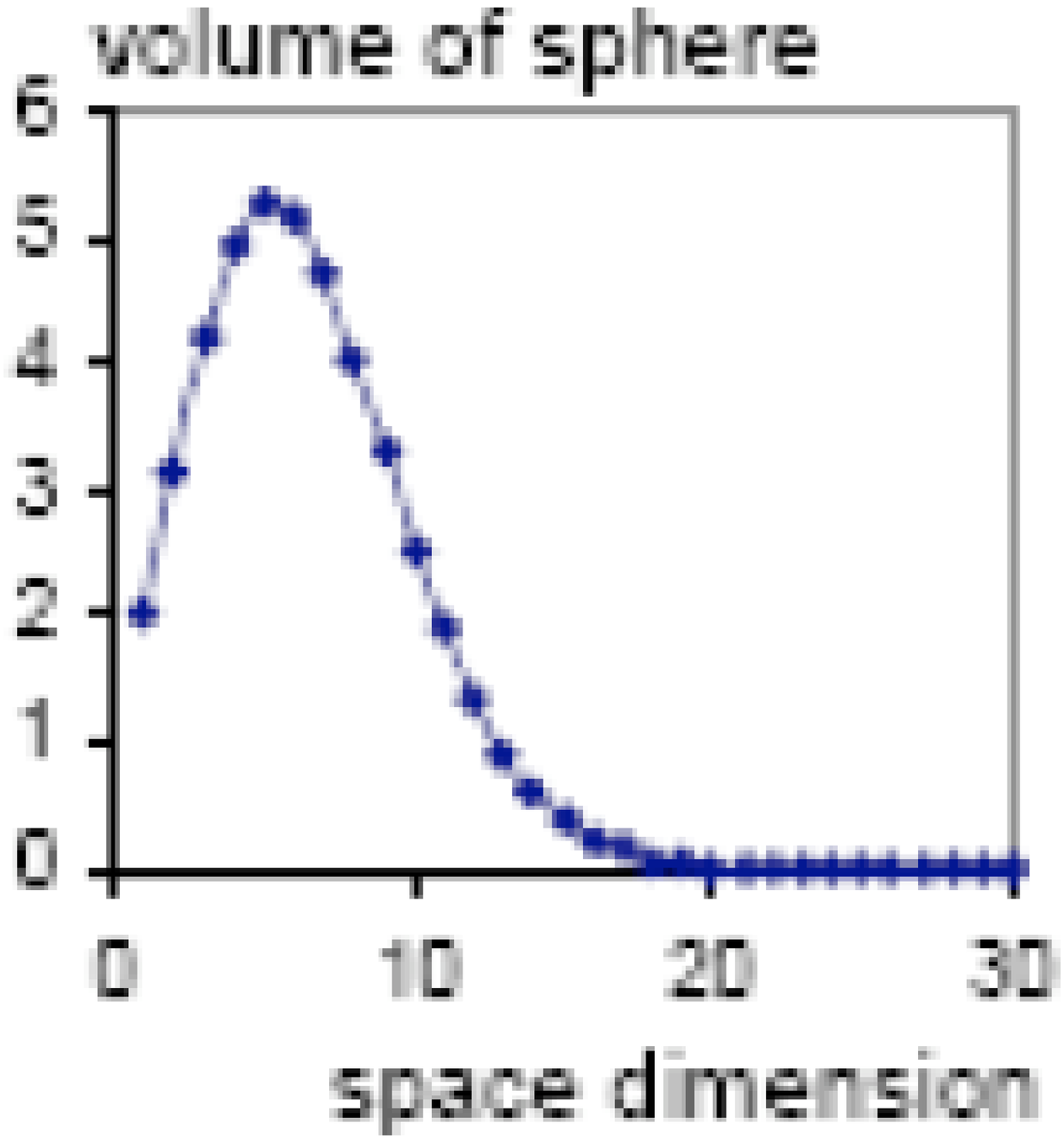
Now it Makes More Sense: The K-Means

- Randomly generate k clusters and determine the cluster centers or directly generate k seed points as cluster centers
- Assign each point to the nearest cluster center.
- Recompute the new cluster centers.
- Repeat until some convergence criterion is met (usually that the assignment hasn't changed).

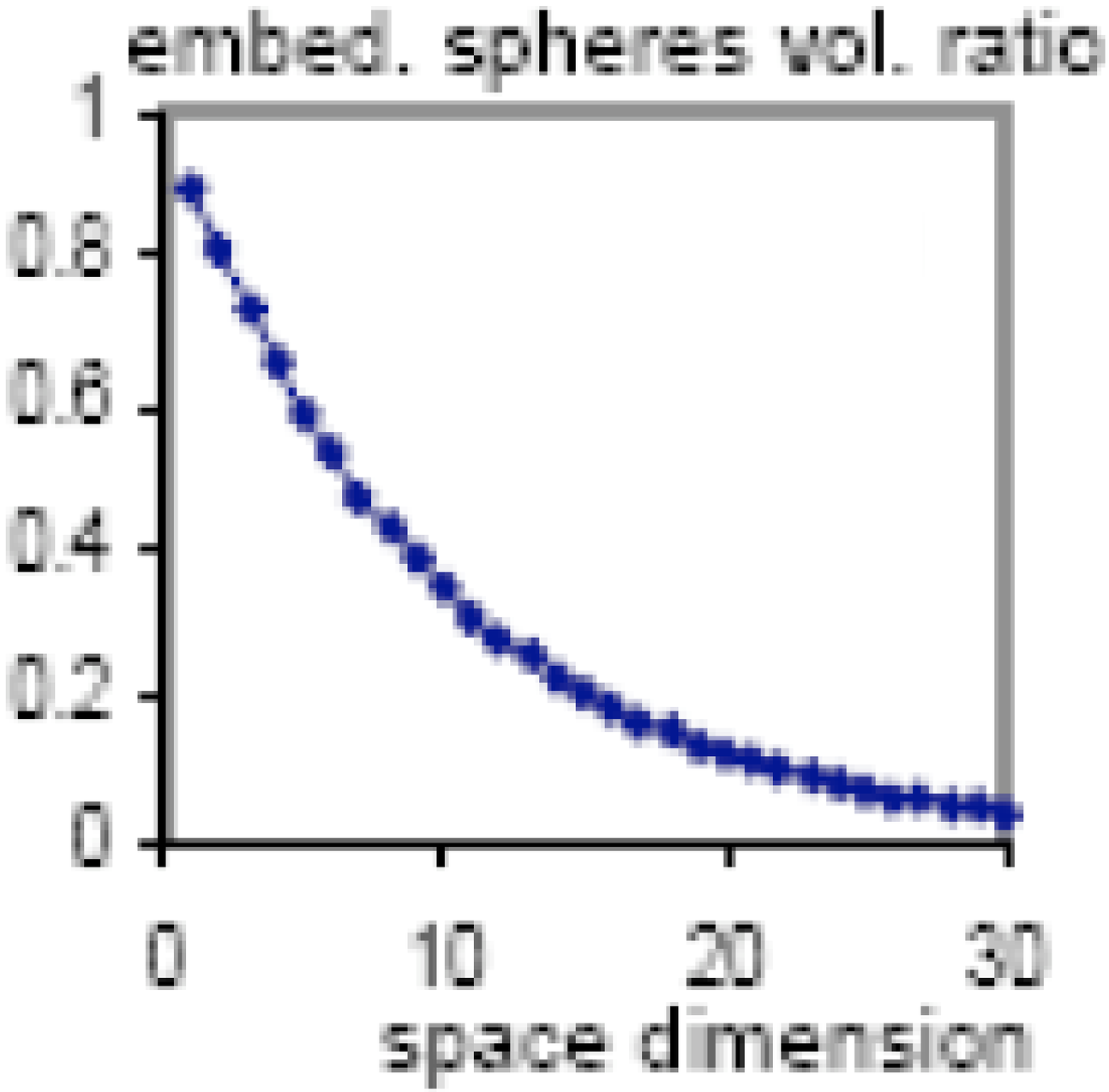
The Curse of High Dimensionality

- The dimensionality is one of the main problem to face when clustering data.
- Roughly speaking the higher the dimensionality the lower the power of recognizing similar objects.

A Visual Representation



Sphere/Sphere Volume Ratios



Clustering in Our Universe

- In our Universe, i.e. the Web, there are billions of documents made of millions of different words.
- Suppose 100 million different words.
- We have 100 million dimensions.
- We MUST face the curse of dimensionality!

Is K-Means Still Valid?

- We will see different approaches to Web Clustering.
- Anyway, YES!
  - K-Means is definitely still an option!

Categorization/ Classification

- Given:
  - A description of an instance, x in X, where X is the instance language or instance space.
    - E.g: how to represent text documents.
  - A fixed set of categories C = $\{c_1, c_2, \ldots, c_n\}$
- Determine:
- The category of x: c(x) in C, where c(x) is a categorization function whose domain is X and whose range is C.

Classification Task

- Input: a training set of tuples, each labeled with one class label
- Output: a model (classifier) that assigns a class label to each tuple based on the other attributes
- The model can be used to predict the class of new tuples, for which the class label is missing or unknown

What is Classification?

- Data classification is a two-step process
  - first step: a model is built describing a predetermined set of data classes or concepts.
  - second step: the model is used for classification.
- Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the class label attribute.
- Data tuples are also referred to as *samples*, *examples*, or *objects*.

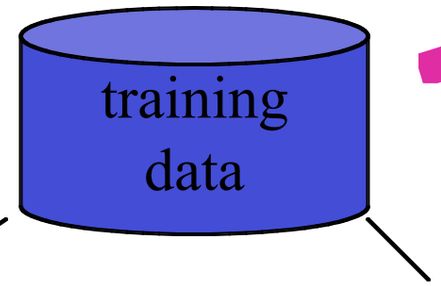
Train and Test

- The tuples (examples, samples) are divided into training set + test set
- Classification model is built in two steps:
  - training - build the model from the training set
  - test - check the accuracy of the model using test set

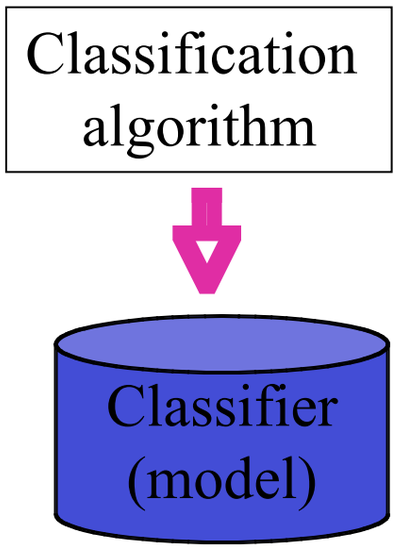
Train and Test

- Kind of models:
  - if - then rules
  - logical formulae
  - decision trees
- Accuracy of models:
  - the known class of test samples is matched against the class predicted by the model
  - accuracy rate = % of test set samples correctly classified by the model

Training Step

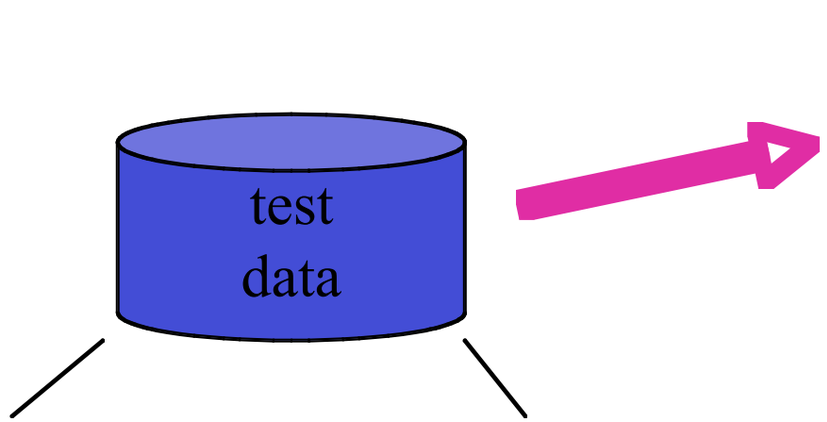


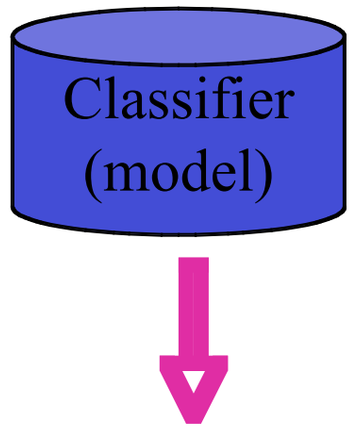
Age	Car Type	Risk
20	Combi	High
18	Sports	High
40	Sports	High
50	Family	Low
35	Minivan	Low
30	Combi	High
32	Family	Low
40	Combi	Low



if age < 31
or Car Type =Sports
then Risk = High

Test Step

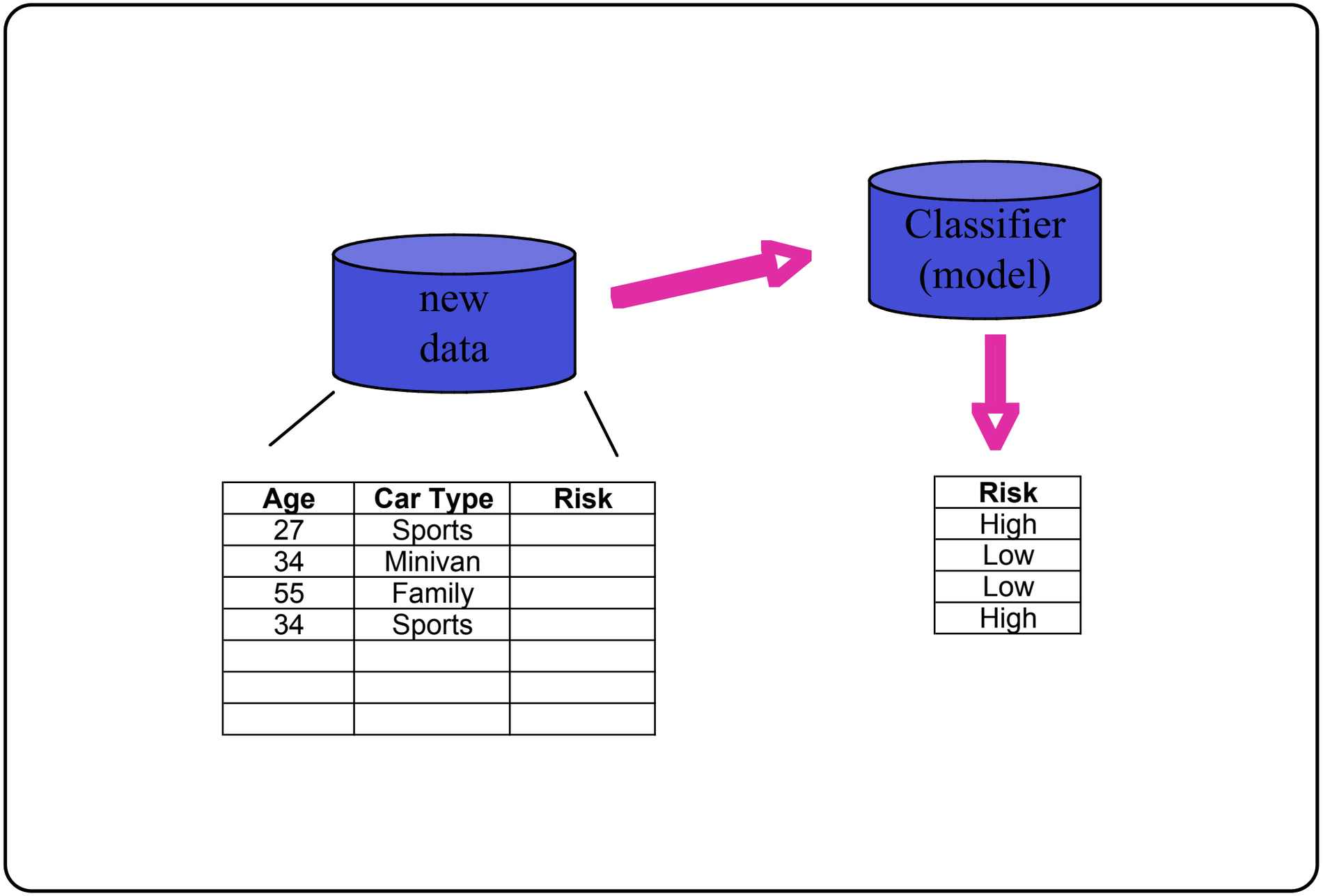




Age	Car Type	Risk
27	Sports	High
34	Family	Low
66	Family	High
44	Sports	High

Risk
High
Low
Low
High

Classification Step



Age	Car Type	Risk
27	Sports	
34	Minivan	
55	Family	
34	Sports	

Risk
High
Low
Low
High

Classification vs. Prediction

- There are two forms of data analysis that can be used to extract models describing data classes or to predict future data trends:
  - classification: predict categorical labels
  - prediction: models continuous-valued functions

Comparing Classification Methods

- Predictive accuracy: this refers to the ability of the model to correctly predict the class label of new or previously unseen data
- Speed: this refers to the computation costs involved in generating and using the model
- Robustness: this is the ability of the model to make correct predictions given noisy data or data with missing values

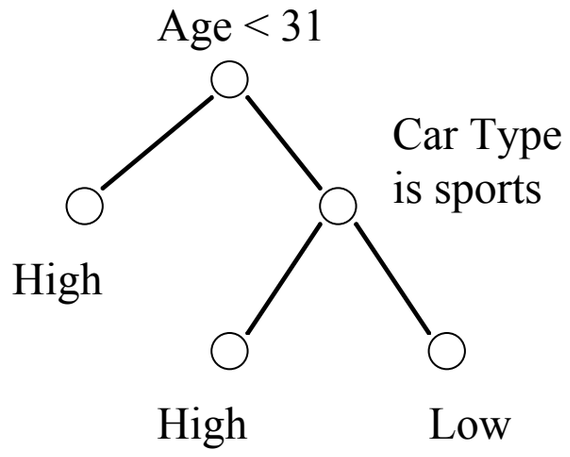
Comparing Classification Methods

- Scalability: this refers to the ability to construct the model efficiently given large amount of data
- Interpretability: this refers to the level of understanding and insight that is provided by the model
- Simplicity, e.g.:
  - decision tree size
  - rule compactness
- Domain-dependent quality indicators

Problem Formulation

- Given records in a database with a class label - find a model for each class.

Age	Car Type	Risk
20	Combi	High
18	Sports	High
40	Sports	High
50	Family	Low
35	Minivan	Low
30	Combi	High
32	Family	Low
40	Combi	Low



Classifier Accuracy

- The accuracy of a classifier on a given test set of samples is defined as the percentage of test samples correctly classified by the classifier, and it measures the overall performance of the classifier.

- Note that the accuracy of the classifier is not estimated on the training dataset, since it would not be a good indicator of the future accuracy on new data.

- The reason is that the classifier generated from the training dataset tends to overfit the training data, and any estimate of the classifier's accuracy based on that data will be overoptimistic.

Classifier Accuracy

- In other words, the classifier is more accurate on the data that was used to train the classifier, but very likely it will be less accurate on independent set of data.

- To predict the accuracy of the classifier on new data, we need to asses its accuracy on an independent dataset that played no part in the formation of the classifier.

- This dataset is called the test set

- It is important to note that the test dataset should not to be used in any way to built the classifier.

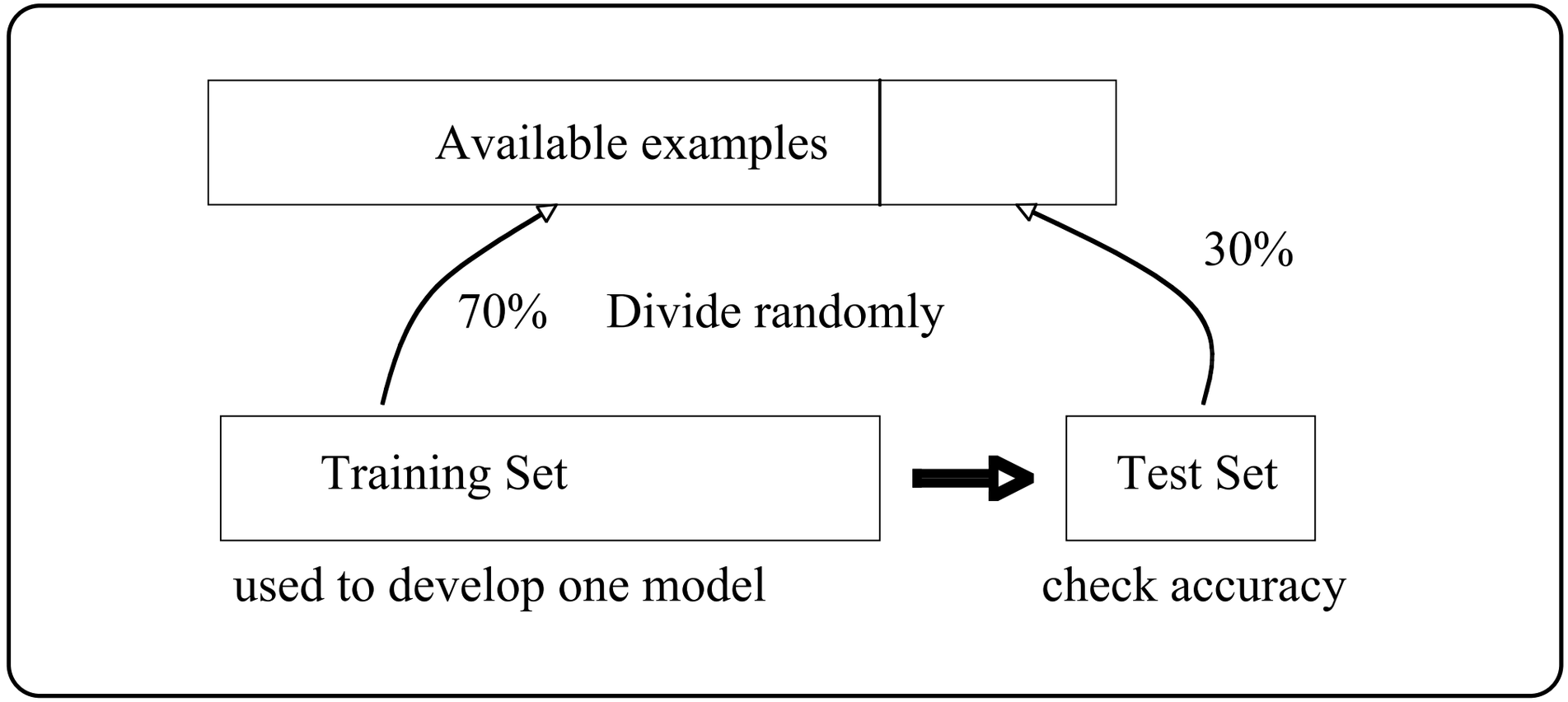
Classifier Accuracy

- There are several methods for estimating classifier accuracy. The choice of a method depends on the amount of sample data available for training and testing.
- If there are a lot of sample data, then the following simple holdout method is usually applied.
- The given set of samples is randomly partitioned into two independent sets, a training set and a test set (typically, 70% of the data is used for training, and the remaining 30% is used for testing)
- Provided that both sets of samples are representative, the accuracy of the classifier on the test set will give a good indication of accuracy on new data.

Stratification

- In general, it is difficult to say whether a given set of samples is representative or not, but at least we may ensure that the random sampling of the data set is done in such a way that the class distribution of samples in both training and test set is approximately the same as that in the initial data set.

Testing in Large Datasets



Classifier Accuracy

- If the amount of data for training and testing is limited, the problem is how to use this limited amount of data for training to get a good classifier and for testing to obtain a correct estimation of the classifier accuracy?

- The standard and very common technique of measuring the accuracy of a classifier when the amount of data is limited is k-fold cross-validation

- In k-fold cross-validation, the initial set of samples is randomly partitioned into k approximately equal mutually exclusive subsets, called folds, S_1, S_2, ..., S_k.

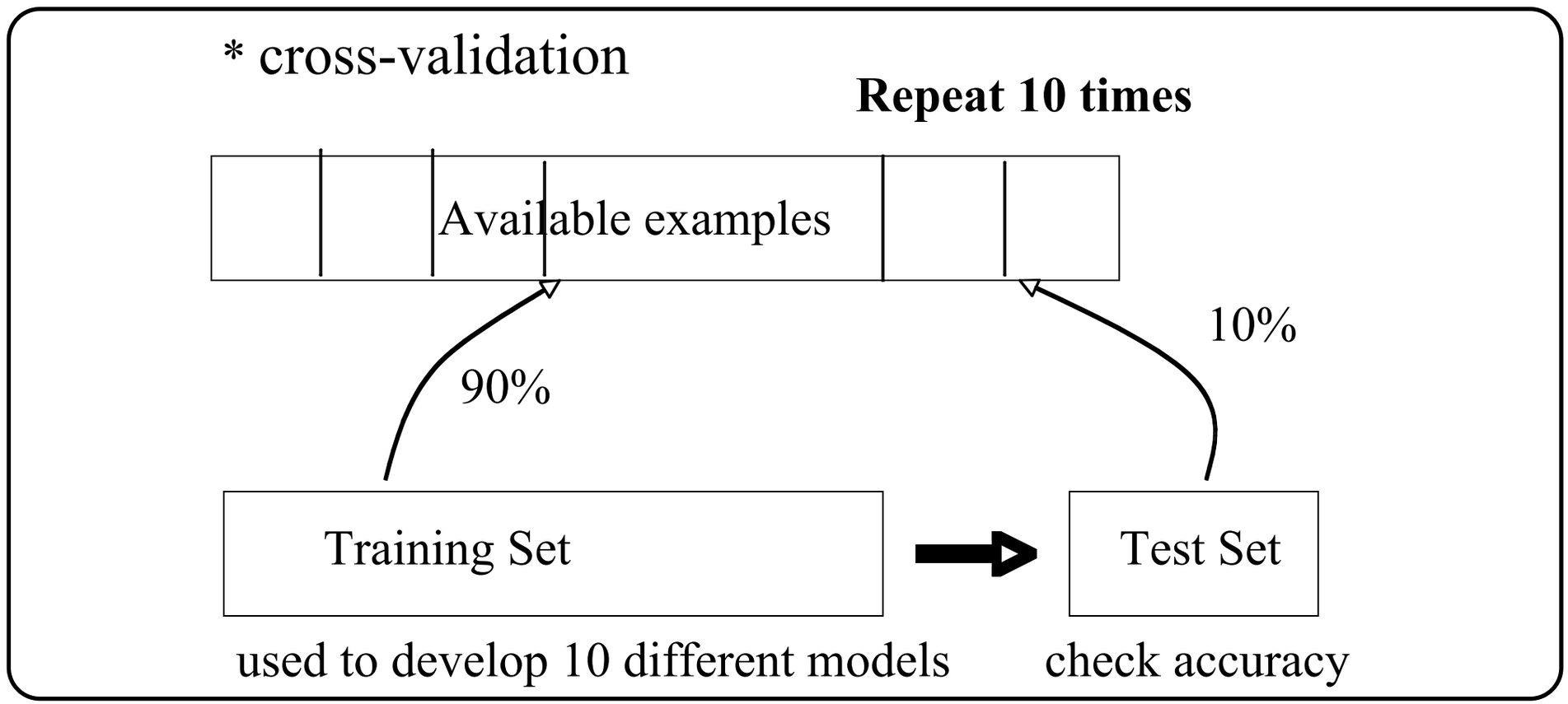
Classifier Accuracy

- Training and testing is performed k times. At each iteration, one fold is used for testing while remainder k-1 folds are used for training. So, at the end, each fold has been used exactly once for testing and k-1 for training.

- The accuracy estimate is the overall number of correct classifications from k iterations divided by the total number of samples N in the initial dataset.

- Often, the k-fold cross-validation technique is combined with stratification and is called stratified k-fold cross-validation.

Testing in Small Datasets



Classifier Accuracy

- There are many other methods of estimating classifier accuracy on a particular dataset
- Two popular methods are leave-one-out cross-validation and the bootstrapping
- Leave-one-out cross-validation is simply N-fold cross-validation, where N is the number of samples in the initial dataset
- At each iteration, a single sample from the dataset is left out for testing, and remaining samples are used for training. The result of testing is either success or failure.
- The results of all N evaluations, one for each sample from the dataset, are averaged, and that average represents the final accuracy estimate.

Classifier Accuracy

- Bootstrapping is based on the sampling with replacement
- The initial dataset is sampled N times, where N is the total number of samples in the dataset, with replacement, to form another set of N samples for training.
- Since some samples in this new "set" will be repeated, so it means that some samples from the initial dataset will not appear in this training set. These samples will form a test set.
- Both mentioned estimation methods are interesting especially for estimating classifier accuracy for small datasets. In practice, the standard and most popular technique of estimating a classifier accuracy is stratified tenfold cross-validation

Accuracy Measures

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

- Accuracy (A) =
  (a + d)/(a + b + c + d)
- Precision (P) =
  d / (b + d)
- Recall (R) =
  d / (c + d)
- F-Measure (F) =
  (2 * P * R) / (P + R)

The Lesson is Over