Ricardo Aler Mur

- First, preprocessing is introduced a a process in the Machine Learning pipeline
- One of the most important preprocessing processes is attribute selection.
- Attribute selection is important to remove redundant and irrelevant attributes, and ease the curse of dimensionality.
- The curse of dimensionality can happen even in linear classifiers.
- An important idea in attribute selection is that the unit of selection might not be an attribute but a set of attributes, because sometimes, two (or more) attributes do not work well when used in isolation, but work well when used together.

- The filter / wrapper and single / subset classification of selection algorithms are introduced.

- Some single attribute selection algorithms are introduced (ranking): based on entropy (which was already used to select the best attrilbute in decision trees), based on mutual-information, and based on chi-square.

- The correlation feature selection (CFS) and Wrapper algorithms are introduced as subset selection algorithm.

- Finally, it is explained the difference between attribute selection and attribute generation: selection selects the important attributes, while generation creates new attributes.

- Two main attribute generation algorithms are explained: PCA and random projections.

- PCA is a non-supervised algorithm and while it can be used to generate new attributes and reduce dimensionality, it has to be used with care when doing classification, given that PCA does not consider the class of the data. Random Projections is introduced as a faster method for doing PCA-like attribute generation and selection.

ATTRIBUTE SELECTION
ATTRIBUTE TRANSFORMATION

ML PIPELINE

- Get data into a matrix format (instances x attributes):
  - Feature extraction (from texts, for instance)
- Preprocessing:
  - Instances
  - Attributes
- Hyper-parameter tuning => best hyper-parameters + model
- Estimation of future performance via train / test or crossvalidation
- Use the model

PREPROCESSING

- Attributes:
  - Normalization (see recommended reading in Aula Global)
  - Dummy variables, one-hot encoding: for some algorithms, every categorical / discrete variable must be transformed into several binary variables
  - Imputation (what to do with missing values?)
  - **Attribute selection**
  - **Attribute transformation**
- Instances:
  - Remove outliers (strange instances)
  - Sampling (in order to have a smaller but representative training dataset)
  - Sampling in order to balance classes in imbalanced problems

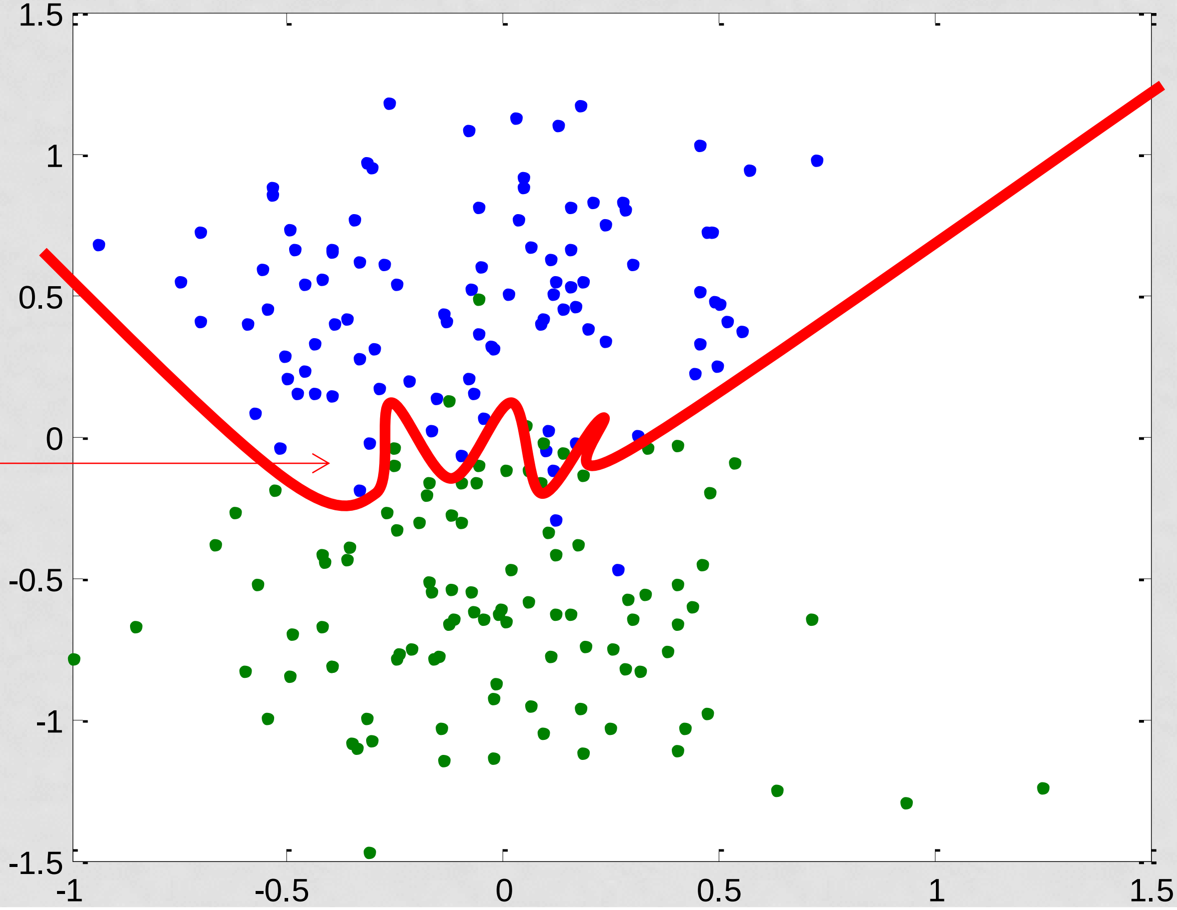
Attribute selection: motivation

- Some attributes can be **redundant** to some extent (such as “salary” and “social class”)
  - Learning is slower (e.g.: C4.5 is O(m*n^2) SVM is O(m*n))
  - Some classifiers can get confused (como el Naive Bayes)
- Some attributes can be **irrelevant** (such as “eye color” in order to predict payment of a loan)
  - In some studies, a single irrelevant attribute (random) dammages 5%-10% results from C4.5 (decision trees)
- **Curse of dimensionality**:
  - The number of required instances for learning can grow exponentiallly with the number of dimensions
- Having too many attributes may result in **overfitting**, because it increases the complexity of the model in relation to the available data.
- Sometimes it is useful to know which attributes are relevant (e.g. which genes are able to predict cancer?)
- The fewer attributes, the easier is to interpret the model

EXAMPLE OF A CLASSIFICATION MODEL NOT GENERALIZING WELL

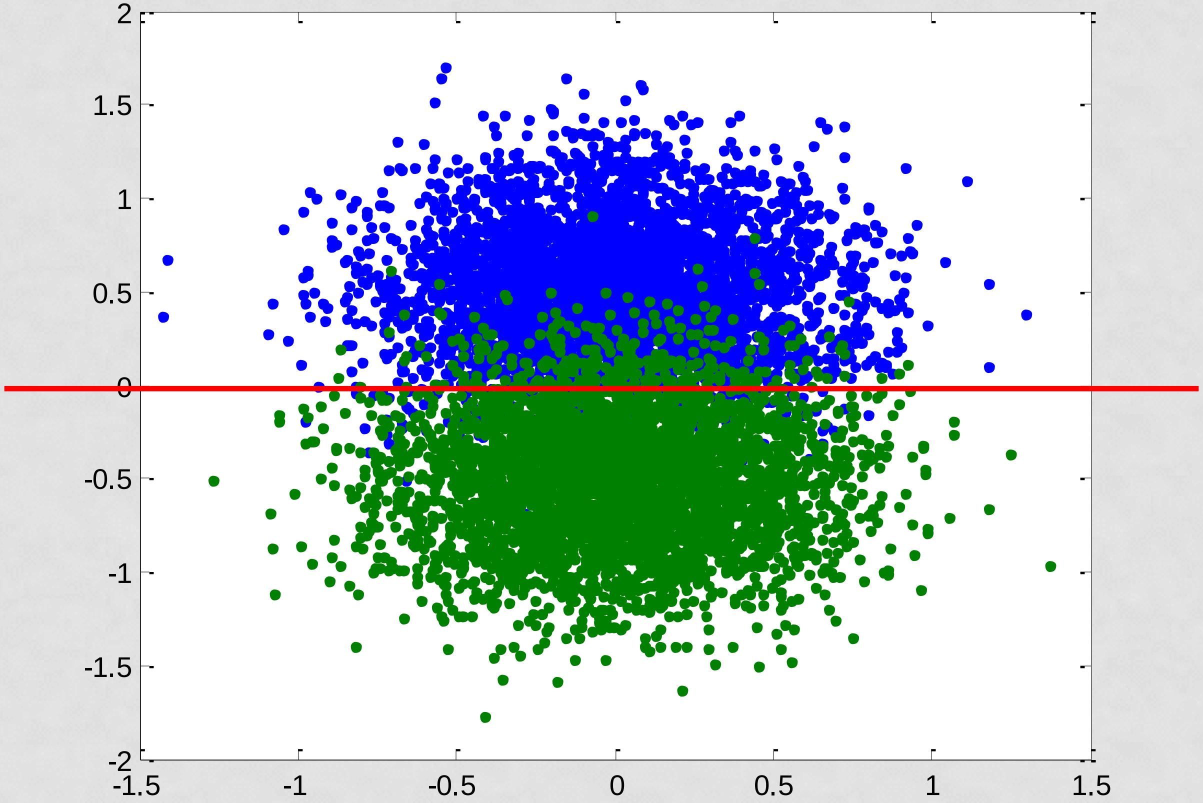
- But if we have few data for training, the following model might be learned
- The model is obviously not generalizing well. It is memorizing the data, or **overfitting** the data

This curve has been learned because there are no green instances here.
But this happened by chance. If we had more instances, probably there would be green instances in that region



EXAMPLE OF A CLASSIFICATION MODEL NOT GENERALIZING WELL

If we had lots of data, the following (correct) model would be learned



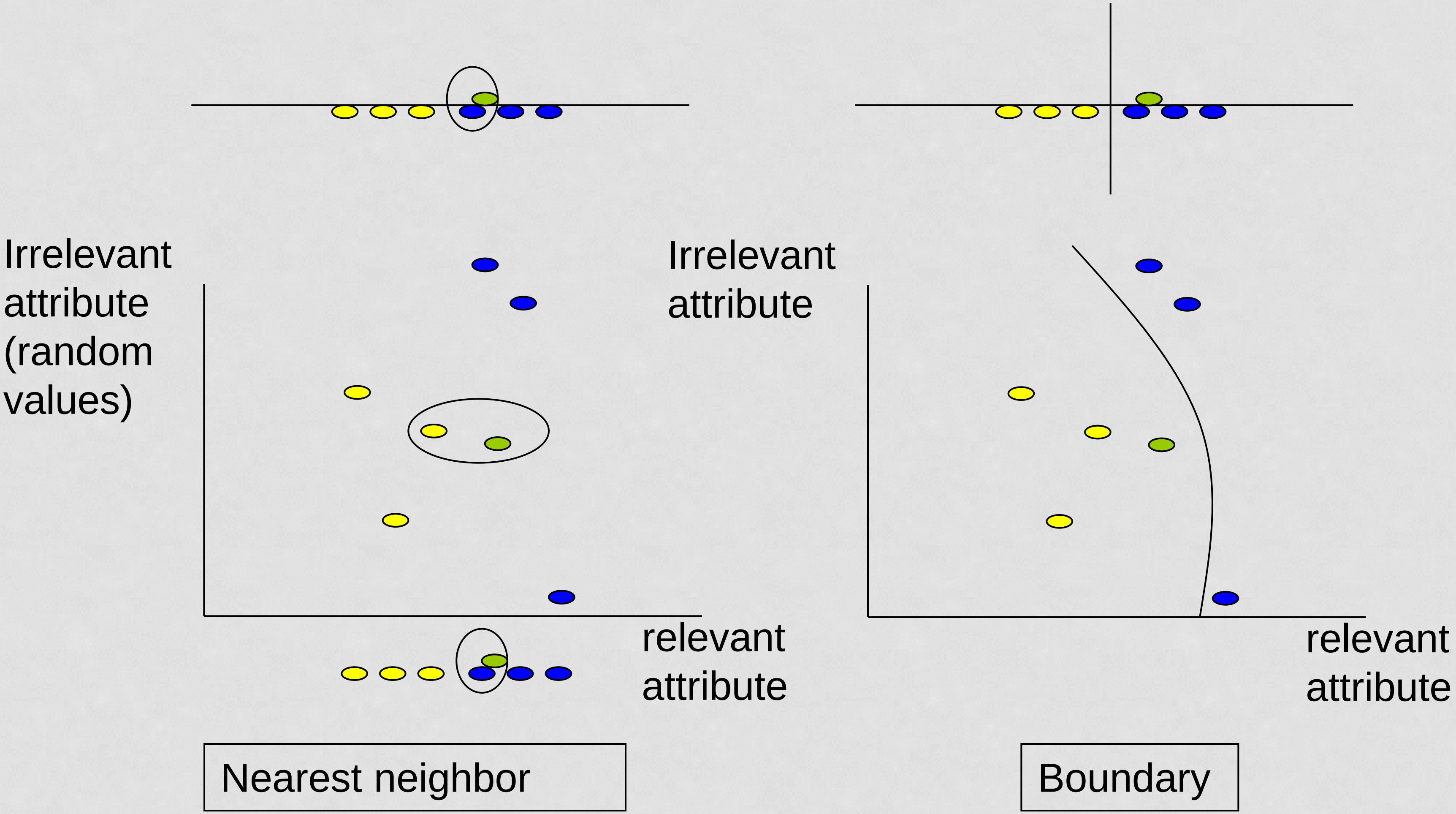
Redundant attributes

- Example in Naive Bayes. It assumes that attributes are (conditionally independent)
- Pr(Yes/sky = sunny, temp = cold, humidity = high, wind = yes)

  = k* pr(sky = sunny/yes) * pr(temp = cold /yes) * pr(humidity = high /yes) * pr(wind = yes /yes) * Pr(play tennis = yes)
- Let’s suppose that temperature and humidity are completely redundant. That is temperature = humidity
  - They are not, we are just assuming they are for the sake of the argument
  - Then, it is as if humidity was counted twice, as opposed to the rest of attributes:

  k* pr(sky = sunny/yes) * pr(humidity = high /yes)^2 * pr(wind = yes /yes) * Pr(play tennis = yes)

Irrelevant attributes



Irrelevant attribute (random values)

Irrelevant attribute

relevant attribute

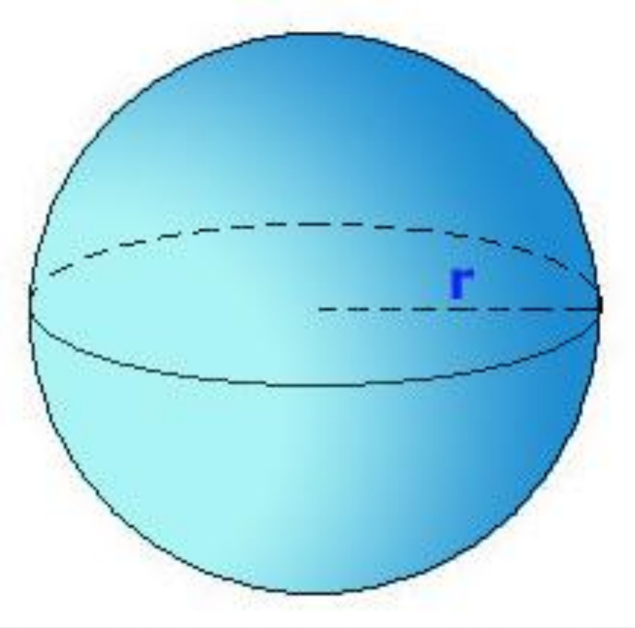
relevant attribute

Nearest neighbor

Boundary

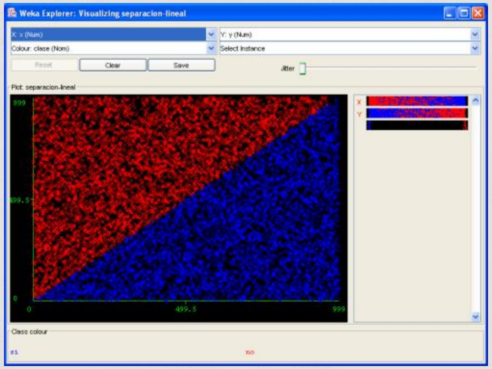
THE CURSE OF DIMENSIONALITY

- Volumes grow exponentially to the number of dimensions d.
- Example: surface of a sphere:
  - 2D: $2\pi r$ = 10 instances
  - 3D: $4\pi r^2$ = 100 instances
  - 4D: $2\pi^2 r^3$ = 1000 instances
  - 50D: = 10^{50} instances
  - dD: $O(r^{d-1})$



THE CURSE OF DIMENSIONALITY IN A LINEAR CLASSIFIER

- Two-class classification problem with **1000 attributes**
- Let's solve it with a linear classifier (i.e. the boundary is a hyper-plane)
- Let's assume we have **1001 training instances** (and 10000 test instances)
- What would be the training accuracy?
- What would be the test accuracy?



X: x (Num)

Y: y (Num)

Colour: clase (Nom)

$$A_1 * X_1 + A_2 * X_2 + A_3 * X_3 + \ldots + A_{1000} * X_{1000} > A_0$$

THE CURSE OF DIMENSIONALITY

- Conclusion: if there is not a good relation of available data to the number of attributes, classification may not be accurate, **even if all the attributes are relevant**

ADVANTAGES OF ATTRIBUTE / FEATURE SELECTION

- Alleviate the curse of dimensionality
- Improve generalization of the classifier (removing irrelevant and redundant attributes)
  - However, bear in mind that some classifier learning algorithms are able to deal with irrelevant attributes indirectly via hyper-parameters. For instance, shallow decision trees indirectly force the algorithm to choose the most relevant attributes. In other algorithms such as neural networks and SVMs, this is called "regularization"
- Speed up the learning process (but it is necessary to include the time for the attribute selection phase)
- Improve the interpretability of the model (by reducing the complexity / size of the model)

IMPORTANT IDEA

- Sometimes, two attributes are not predictive separately, but they are if they are used together (**attribute interaction**)
- Example:
  - Classification problem into two classes: computer science and philosophy
  - Binary attributes “intelligence” and “artificial” which are true if these words appear in the text and false otherwise (remember what we learned about feature extraction)
  - Separately, they do not allow to differenciate between computer science and anthropology, because both words appear in both types of books:

    IF inteliligence=yes **THEN** ?; **IF** artificial=yes **THEN** ?
  - But together they can

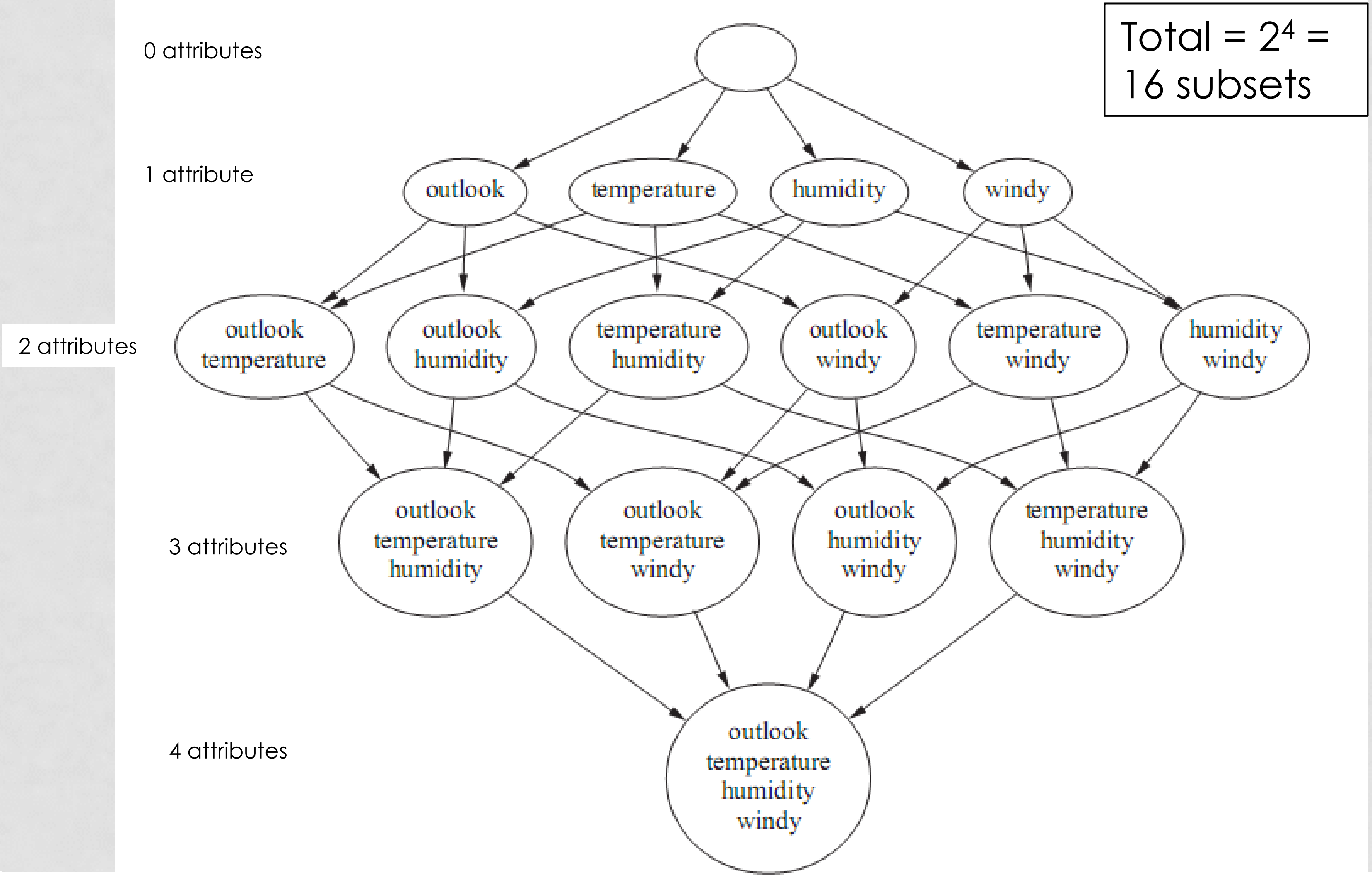
    IF intelligence=yes **AND** artificial=yes **THEN** “computer science”
- Therefore, the aim of attribute selection is to find the smallest **subset** of attributes for optimal prediction

EXHAUSTIVE SEARCH

- Test all possible subsets of attributes
- If there are 4 input attributes A, B, C, D
- The list of possible subsets to try is 2^4=16: {A, B, C, D}, {A, B, C}, {A, B, D}, {B, C, D}, {A, C, D}, {A, B}, {A, C}, …, {A}, {B}, {C}, {D}
- For n large, this is not feasible:
  - n = 10 => 2^{10} = 1024 subsets
  - n = 20 => 2^{20} = 1048576 subsets
  - n = 30 => 2^{30} = 1073741824 subsets
  - …

Space of subsets of attributes

Total = 2^4 = 16 subsets



TYPES OF ATTRIBUTE SELECTION METHODS

	Filter	Wrapper
Ranking (individual attributes)	Entropy (Information Gain), Chi-square, …	
Subset selection	Correlation Feature Selection (CFS)	Wrapper

Ranking: evaluation and ranking of attributes individually. Remove the less relevant attributes, below a threshold

Subset selection : search for the most relevant subset

Filter: evaluate attributes by using a simple expression

Wrapper: evaluate attributes by learning a model and testing its performance

- Search methods: different ways of traversing the space of attribute subsets
  - Greedy stepwise:
    - Sequential Forward Selection
    - Sequential Backward Selection
  - Best first (this is an artificial
  - Genetic algorithms
  - …

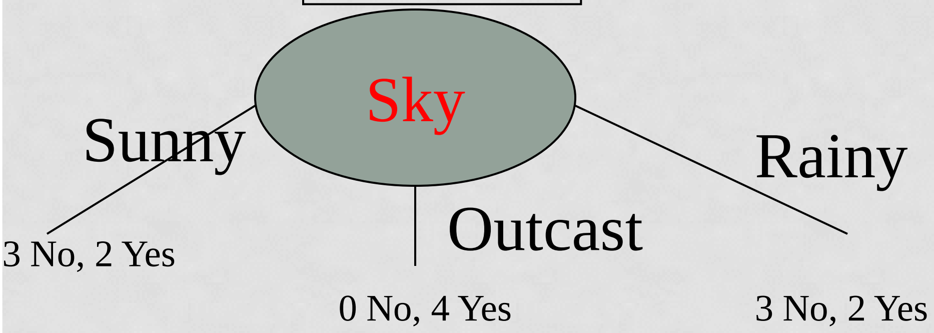
Ranking

- Given input attributes A_1, A_2, ..., A_n, each A_i is evaluated by itself, computing its correlation with the class, independently of the rest of attributes (i.e. attributes are considered individually, rather than subsets)
- An attribute A_1 is correlated with the class, if knowing its value implies that the class can be predicted more accurately
    - For instance, car speed is correlated with having an accident. But the Social Security Number of the driver is not.
    - For instance, salary is correlated with credit default
- How to evaluate / rank attributes (attribute/class correlation):
    - Entropy (information gain), like in decision trees
    - Chi-square
    - Mutual information
    - …
- Once evaluated and ranked, the worse attributes are removed (according to a threshold)

Entropy / Information Gain for ranking attributes

$$H(P) = -(p_{si}\log_2(p_{si}) + p_{no}\log_2(p_{no}))$$

HP=0.69



Sky

Sunny

Outcast

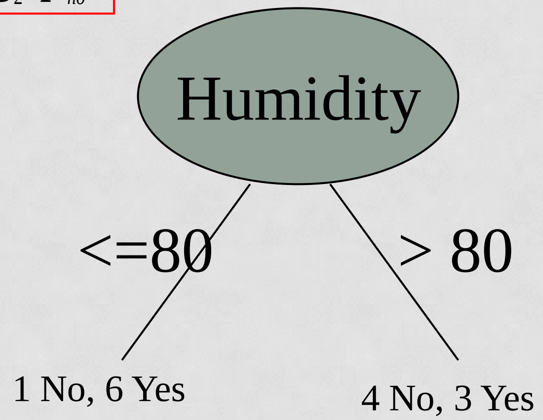
Rainy

3 No, 2 Yes

0 No, 4 Yes

3 No, 2 Yes

HP = 0.76



Humidity

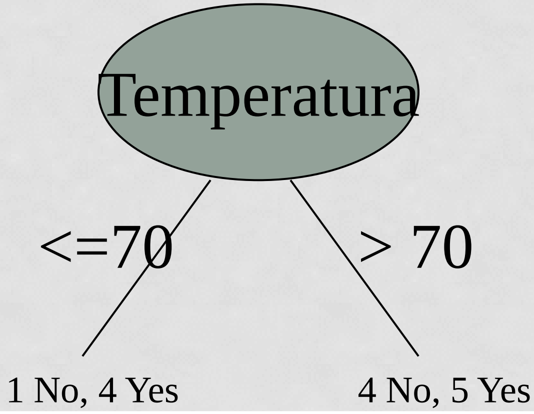
<=80

> 80

1 No, 6 Yes

4 No, 3 Yes

HP = 0.89



Temperatura

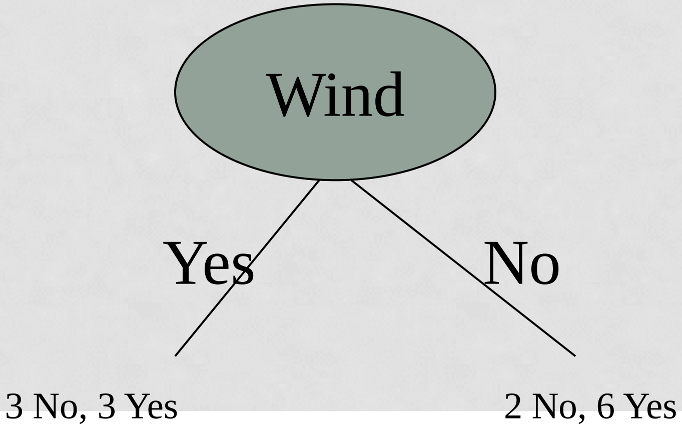
<=70

> 70

1 No, 4 Yes

4 No, 5 Yes

HP = 0.89



Wind

Yes

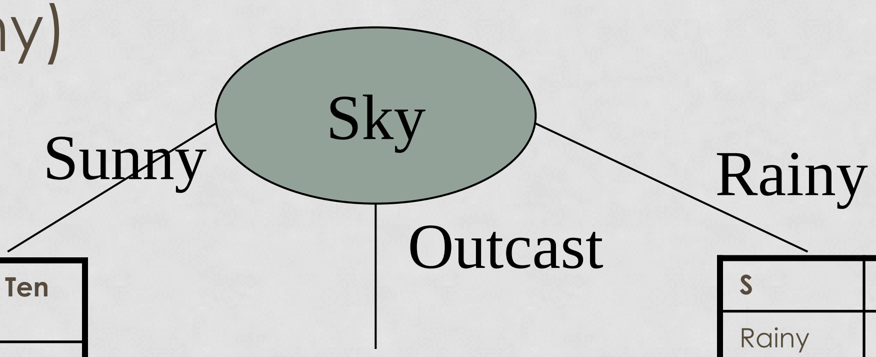
No

3 No, 3 Yes

2 No, 6 Yes

Entropy / Information Gain for ranking attributes

Sky generates as many partitions as values (3: sunny, outcast, rainy)



S	T	H	W	Ten
Sunny	85	85	No	No
Sunny	80	90	Yes	No
Sunny	72	95	No	No
Sunny	69	70	No	Yes
Sunny	75	70	Yes	Yes

“3 No, 2 Yes”

“no” majority

S	T	H	W	Ten
Outcast	83	86	No	Yes
Outcast	64	65	Yes	Yes
Outcast	72	90	Yes	Yes
Outcast	81	75	No	Yes

“0 No, 4 Yes”

Perfect partition

S	T	H	W	Ten
Rainy	70	96	No	No
Rainy	68	80	No	Yes
Rainy	75	80	No	Yes
Rainy	65	70	Yes	No
Rainy	71	91	Yes	No

“3 No, 2 Yes”

“no” majority

RANKING WITH MUTUAL INFORMATION

If x and y are independent,
p(x,y)=p(x)*p(y)

$$I(x,y)=\sum_{i}\sum_{j}p(x=i,y=j)\log\left[\frac{p(x=i,y=j)}{p(x=i)p(y=j)}\right]$$

•i means the values of attribute x, j means the values of class y

•I(x,y)=0 if x and y are independent (log(1) = 0)

•I(x,y)>= 0 (the more correlated, the larger is mutual information)

RANKING WITH CHI SQUARE

- Chi square is a statistical test that measures the association strength between two variables
- Liu, H., & Setiono, R. (1995, November). Chi2: Feature selection and discretization of numeric attributes. In *tai* (p. 388). IEEE.

Ranking

- Advantages: fast
- Disadvantages:
  - Redundant attributes are not removed
  - Attribute interaction is not detected: subsets of attributes that work well together but not individually are not detected. In fact, they are likely to be discarded.
    - E.g.: "inteligence" and "artificial" for anthropology/ computer science text classification

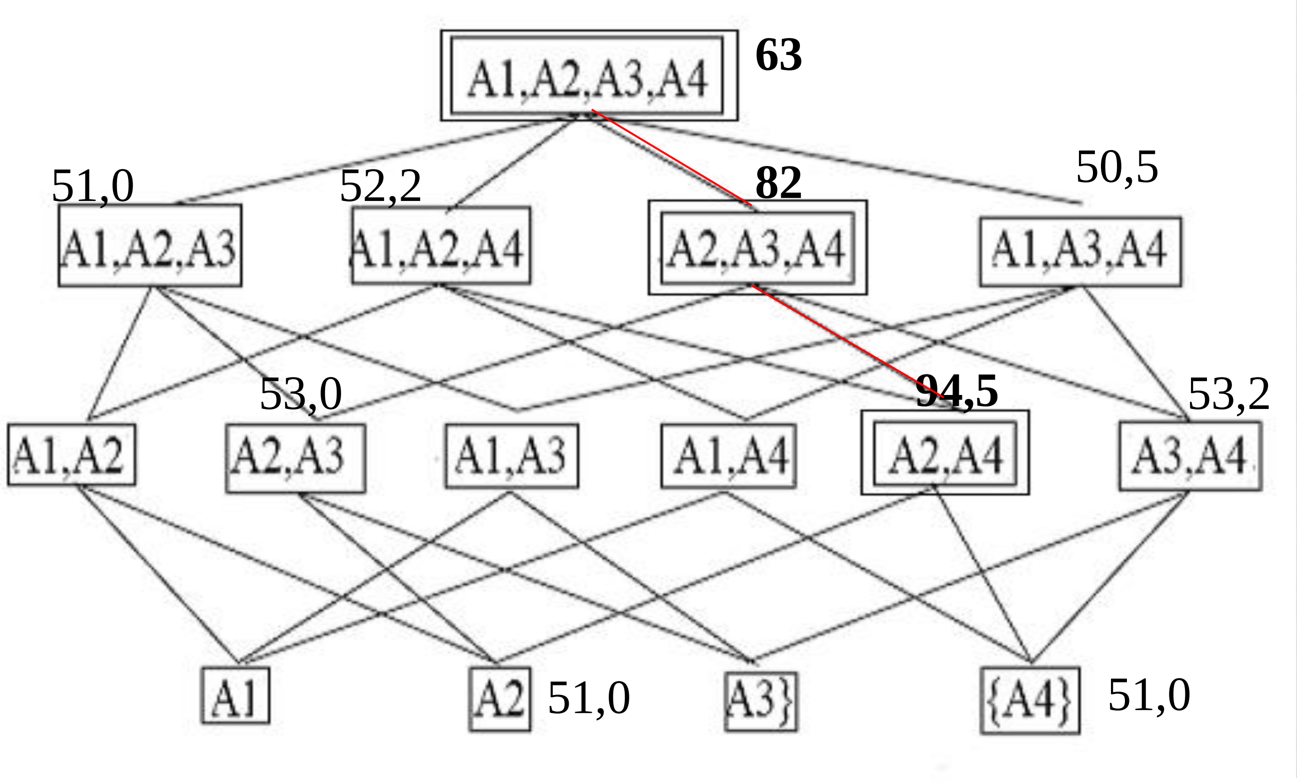
TYPES OF ATTRIBUTE SELECTION METHODS

	Filter	Wrapper
Ranking (individual attributes)	Entropy (Information Gain), Chi-square, …	✗
Subset selection	Correlation Feature Selection (CFS)	Wrapper

Subset selection : search for the most relevant subset

SUBSET SELECTION

- Subsets are evaluated (rather than individual attributes)
- But given that exhaustive search is not feasible,
- only a few subsets are evaluated

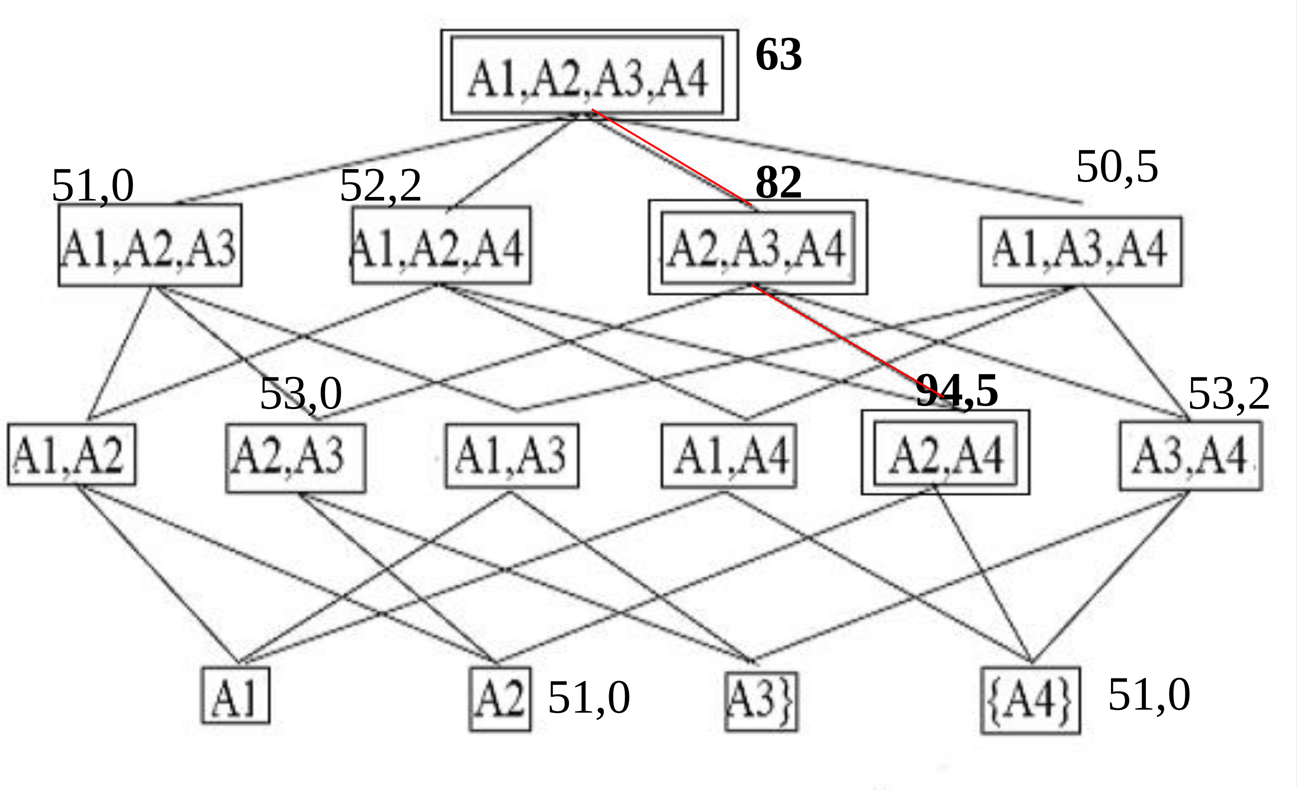


SUBSET SELECTION

- Two issues to be defined:
  - **A way of traversing the subset space:**
    - Forwards: start with the empty set of attributes, and add attributes one by one, until adding attributes makes results worse. **Sequential Forward Selection / Greedy forward**
    - Backwards: start with the whole set of attributes and remove them one by one, until removing attributes makes results worse. **Sequential backwards selecti**on / Greedy backward
    - …
  - A way of evaluating subsets of attributes:
    - CFS (Correlation Feature Seletion)
    - Wrapper
    - …

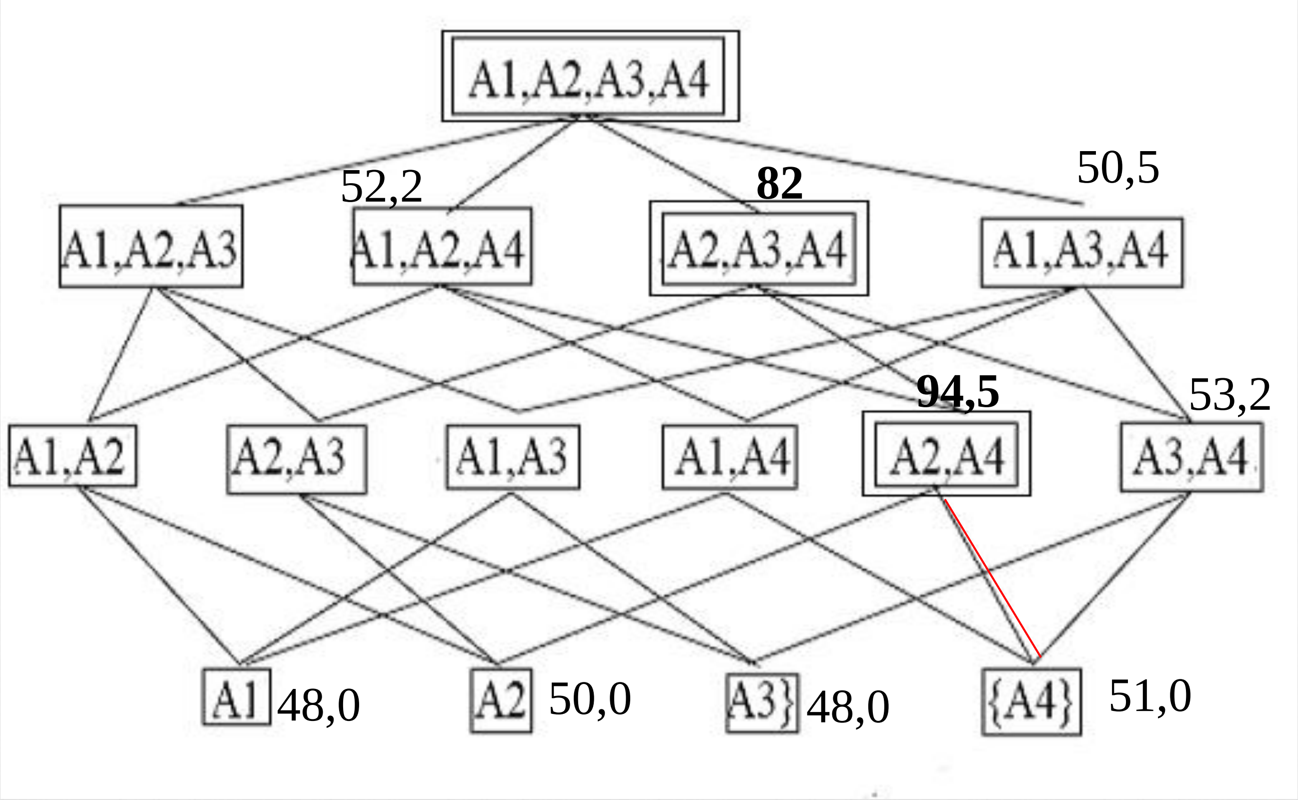
SEQUENTIAL BACKWARD SELECTION

1. Start with the whole set of attributes
2. Remove the attribute that produces the best reduced subset
3. Go to 2 until no improvements are found (or the size of the subset is small enough)



SEQUENTIAL FORWARD SELECTION

1. Start with the empty set of attributes
2. Add the attribute that best works with the current attribute set
3. Go to 2 until no improvements are found



SUBSET SELECTION

- Two issues to be defined:
  - A way of traversing the subset space:
    - Forwards: start with the empty set of attributes, and add attributes one by one, until adding attributes makes results worse. Sequential Forward Selection / Greedy forward
    - Backwards: start with the whole set of attributes and remove them one by one, until removing attributes makes results worse. Sequential backwards selection / Greedy backward
    - …
  - **A way of evaluating subsets of attributes:**
    - CFS (Correlation Feature Seletion)
    - Wrapper
    - …

TYPES OF ATTRIBUTE SELECTION METHODS

	Filter	**Wrapper**
Ranking (individual attributes)	Entropy (Information Gain), Chi-square, …	╳
Subset selection	**Correlation Feature Selection (CFS)**	Wrapper

Subset selection : search for the most relevant subset

Filter: evaluate attributes by using a simple expression

- Search methods: different ways of traversing the space of attribute subsets
  - Greedy stepwise:
    - Sequential Forward Selection
    - Sequential Backward Selection
  - Best first (this is an artificial
  - Genetic algorithms
  - …

SUBSET EVALUATION: Correlation Feature Selection (CFS)

- *CFS* evaluates a subset of attributes computing:
  - The average of each input attribute-class correlation
  - The correlations between input attributes. If two input attributes are correlated, that means they have some degree of redundancy

$$\text{Evaluation}(A_i) = \frac{\text{Average of correlation with the class}}{\text{correlations between input attributes}} = \sum_j U(A_j, C) \Big/ \sqrt{\sum_i \sum_j U(A_i, A_j)}$$

SUBSET EVALUATION :
Correlation Feature Selection (CFS)

- Advantage:
  - Quick
  - Removes redundant attributes
- Disadvantages: it removes redundant attributes, but it does not detect attribute interactions (i.e. it can remove attributes that are correlated with the class together, but not individually. e.g. “inteligence” and “artificial”)

TYPES OF ATTRIBUTE SELECTION METHODS

	Filter	**Wrapper**
Ranking (individual attributes)	Entropy (Information Gain), Chi-square, …	✗
Subset selection	Correlation Feature Selection (CFS)	**Wrapper**

Subset selection : search for the most relevant subset

Wrapper: evaluate attributes by learning a model and testing its performance

SUBSET EVALUATION: Wrapper

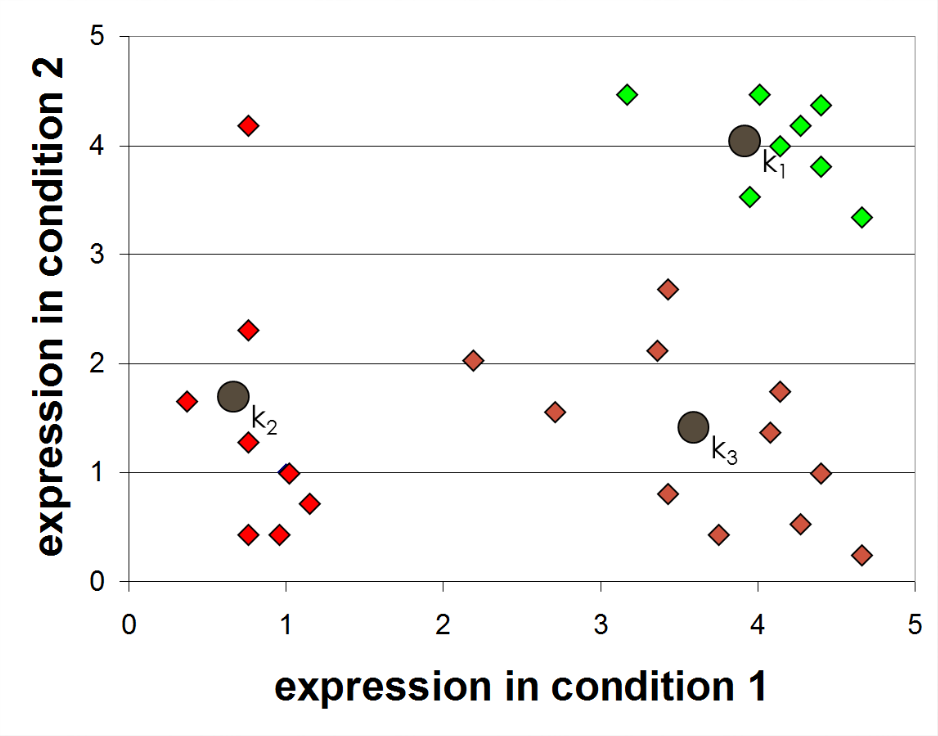
- *Wrapper* methods evaluate a subset of attributes by building a model (like a decision tree) and then computing its expected performance (e.g. accuracy for classification)
- Advantages:
  - They obtain subsets of attributes for particular machine learning algorithms (like decision trees)
  - They actually evaluate subsets of attributes
- Disadvantages:
  - Very slow (testing different attribute subsets involves building many models from training sets)
  - Although they are based on a good idea, they can produced overfitting

TRANSFORMATION (+ SELECTION) OF ATTRIBUTES

- Distances to prototypes
- Principal Component Analysis (PCA)
- Random Projections

DISTANCES TO K-MEANS PROTOTYPES

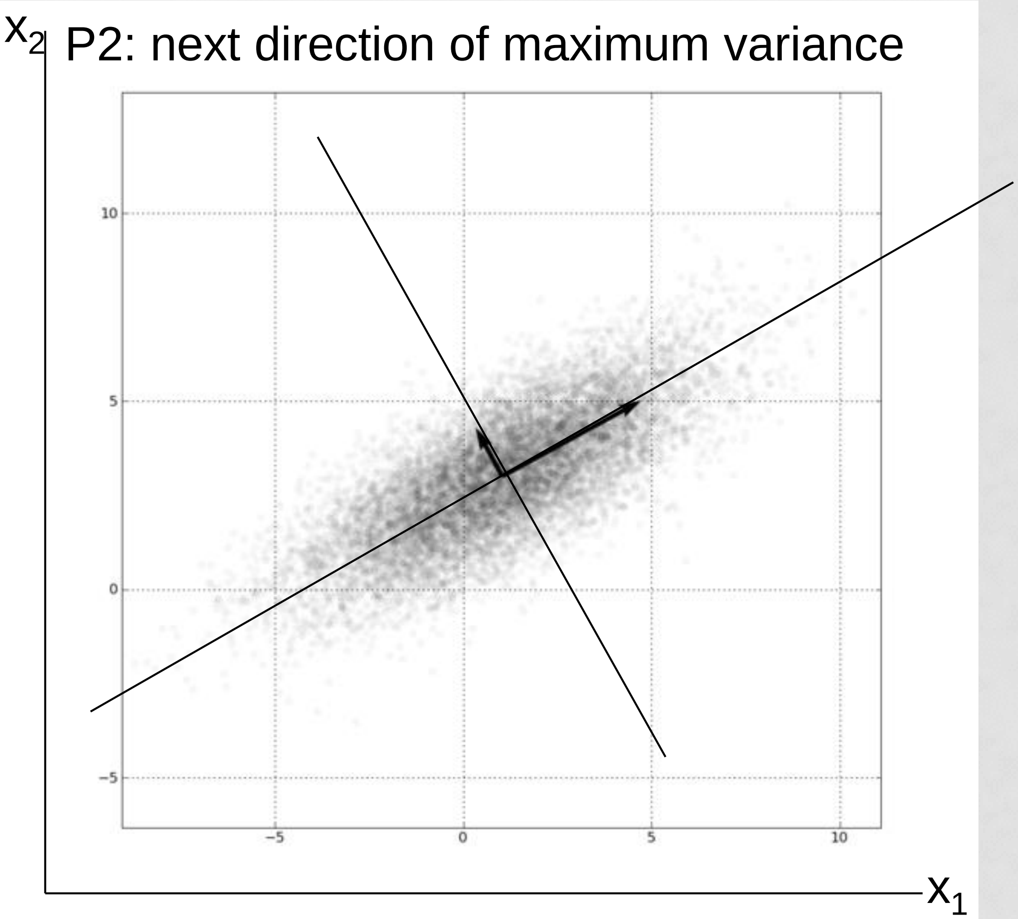
- Compute K prototypes with K-MEANS (unsupervised learning / clustering algorithm)
- Use the distances to the prototypes as additional attributes



TRANSFORMATION WITH PRINCIPAL COMPONENT ANALYSIS (PCA)

- This method constructs new attributes, as a linear combination of the original input attributes
- The new attributes are sorted by the variance of the new attributed (explained variance)
- Dimensionality can be reduced by choosing the attributes with more variance

PCA

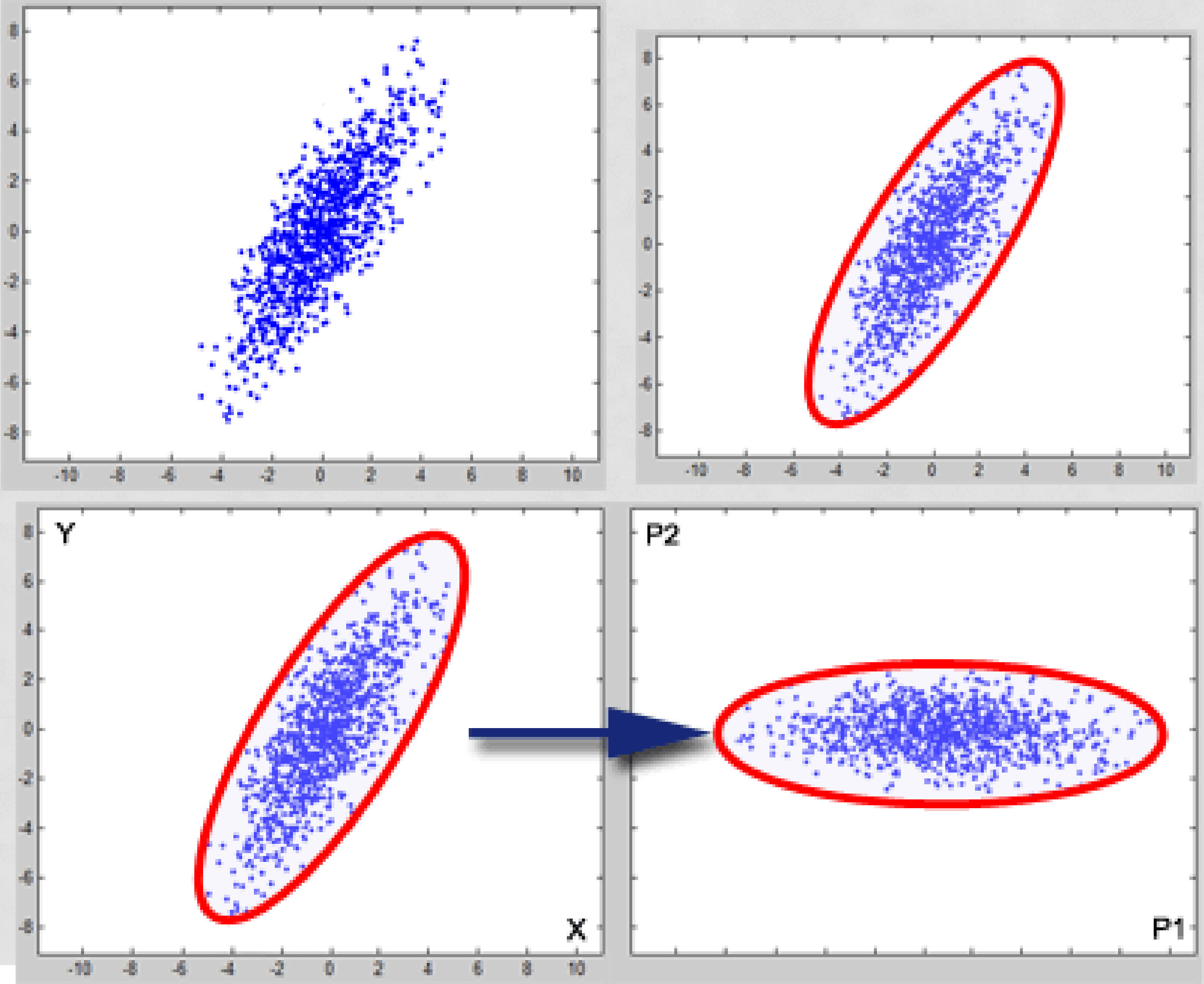


P2: next direction of maximum variance

P1: direction of maximum variance

Two new attributes: P1 and P2

PCA TRANSFORMATION



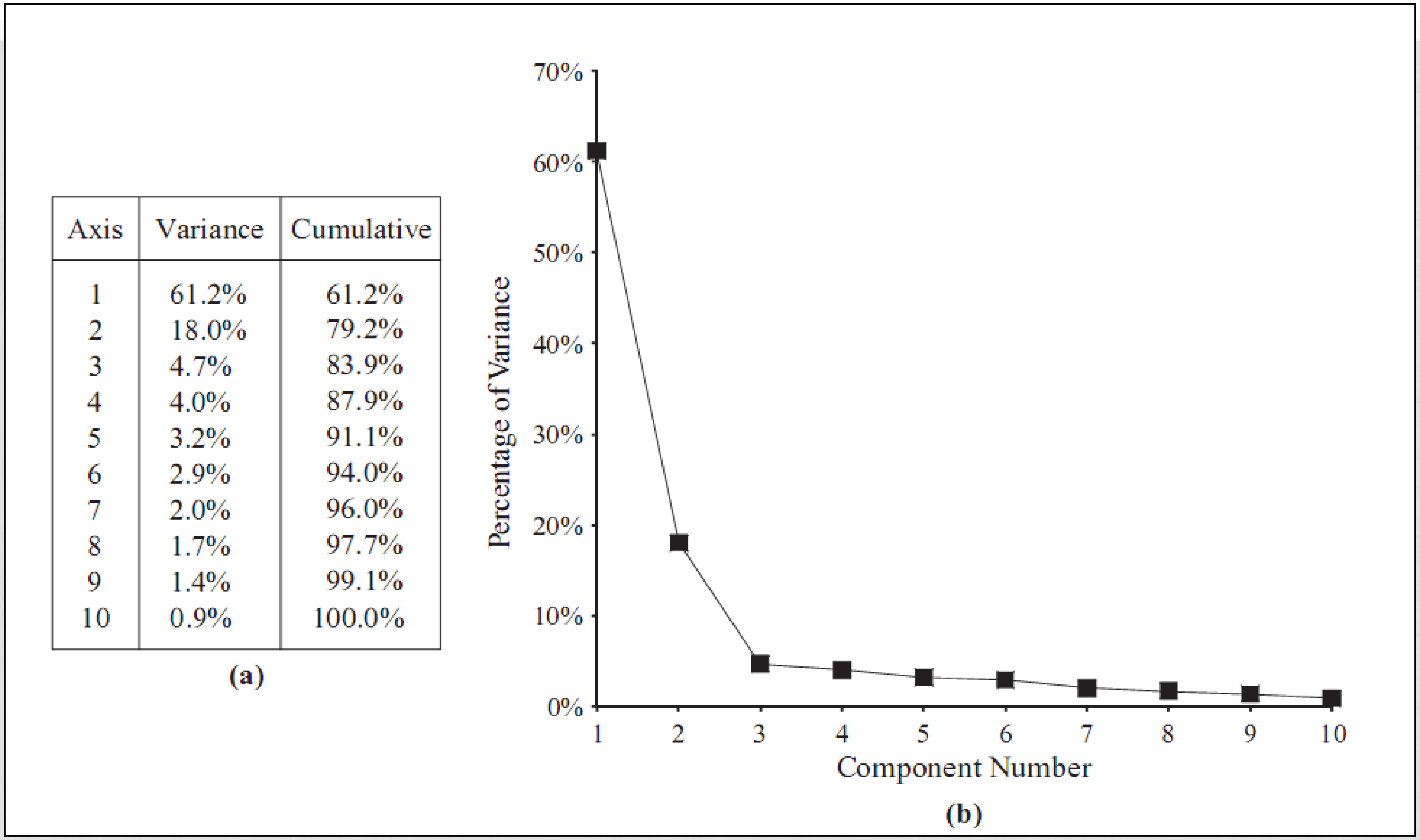
- Linear transformations
- It removes redundancy from attributes (correlation)

$P_1 = k_{11}*x_1+k_{12}*x_2$

$P_2 = k_{21}*x_1+k_{22}*x_2$

$P = \mathbf{X}*k$

PCA: TRANSFORMATION AND SELECTION



Axis	Variance	Cumulative
1	61.2%	61.2%
2	18.0%	79.2%
3	4.7%	83.9%
4	4.0%	87.9%
5	3.2%	91.1%
6	2.9%	94.0%
7	2.0%	96.0%
8	1.7%	97.7%
9	1.4%	99.1%
10	0.9%	100.0%

(a)

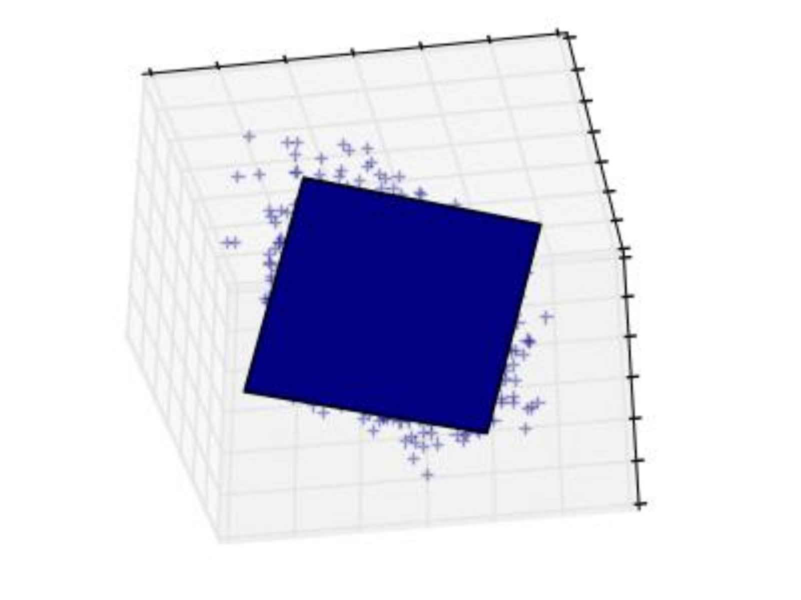
(b)

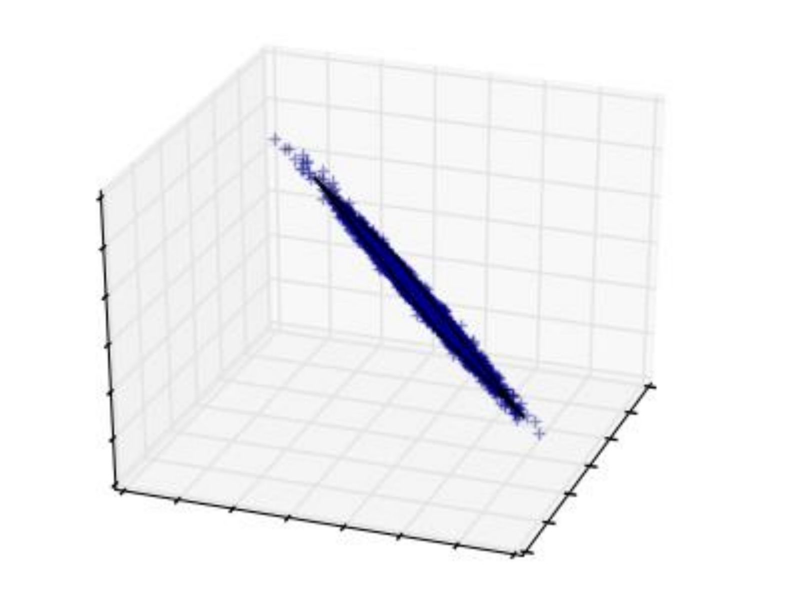
- Typically, a threshold is set so that the explained variance is larger than 95% (7 in this case)

- If only a few attributes explain most of the variance, the rest can be removed (e.g. imagine two dimensional data embedded in 20 dimensions)

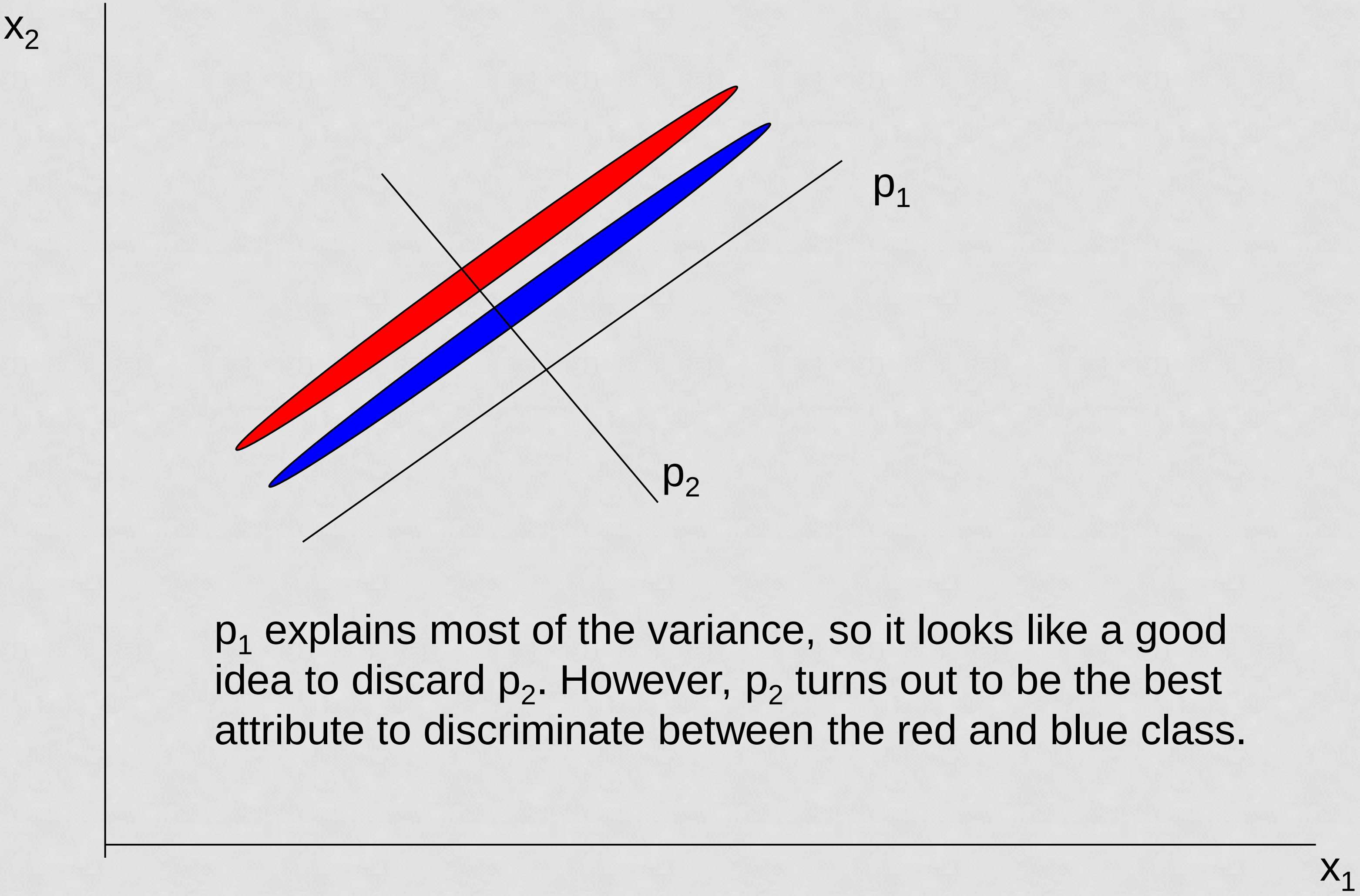
PCA AND ACTUAL DIMENSION OF DATA

- A two dimensional dataset embedded in three dimensions





BEWARE, PCA IS NOT SUPERVISED



p_1 explains most of the variance, so it looks like a good idea to discard p_2. However, p_2 turns out to be the best attribute to discriminate between the red and blue class.

ADVANTAGES / DISADVANTAGES OF PCA

- Advantage: it may find out the actual dimensionality of data
  - E.g.: let's imagine instances in 2D with an ellipsoid shape, but embedded in 20 dimensions. PCA will easily identify that only 2 dimensions are required.
- Advantage: decorrelates attributes (removes redundancy between attributes)
- Disadvantage: PCA is **not supervised**, so there is guarantee that it will find out the attributes that best discriminate between the classes.
- Disadvantage: Slow if lots of attributes.

RANDOM PROJECTIONS

- Projecting data to smaller dimensions by means of random matrices. They can usually obtain similar results to PCA but quickly, as far as the number of projected dimensions is not too small.
- X' = X*R
  - Dim(X) = num. instances x d
  - Dim(R) = dxd' ; d' << d
  - Dim(X') = num. instances x d'
- It can be shown that X' maintains to some extent the structure of instances in X. That is, distances between instances are approximately maintained

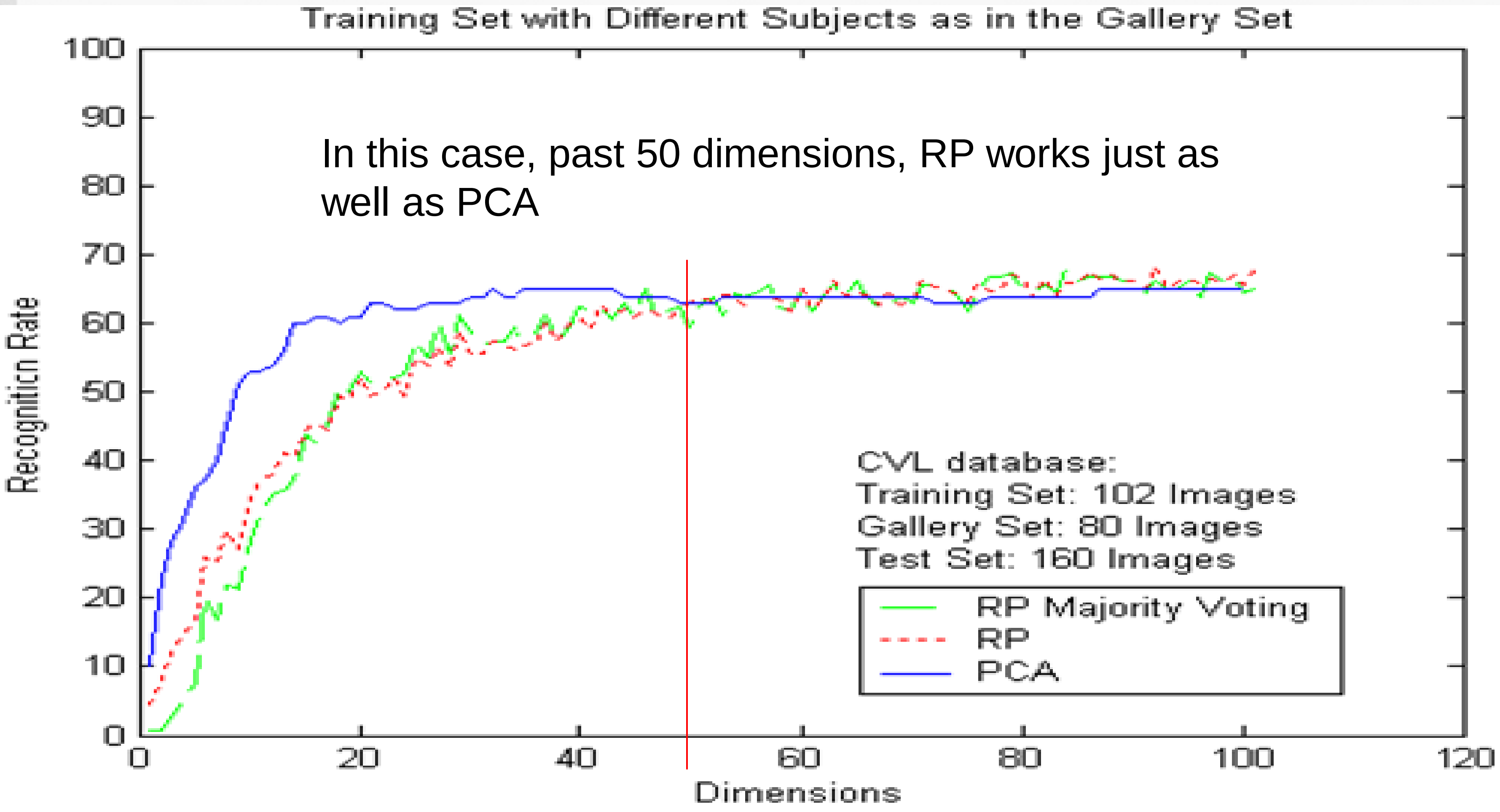
RANDOM PROJECTIONS

- Steps:

1. Generate a matrix R with gaussian random numbers: Normal(0,1)
2. Orthogonalize R (that is, R's columns become orthogonal vectors), by Gram-Schmidtt method (for instance)
3. Normalize R's columns to unity

- Step 2 can be removed, because in high dimensions random vectors are almost orthogonal

RANDOM PROJECTIONS



In this case, past 50 dimensions, RP works just as well as PCA

(c)