# My dear decision tree

# Working definitions

- Observation: a data point consisting of attributes and a class label
  - Very often also termed sample
- Sample (Singular): unfortunately very often used for two different things:
  - The complete set of all observations but also
  - One individual observation
- Samples (Plural): all available observations

# Questionaire

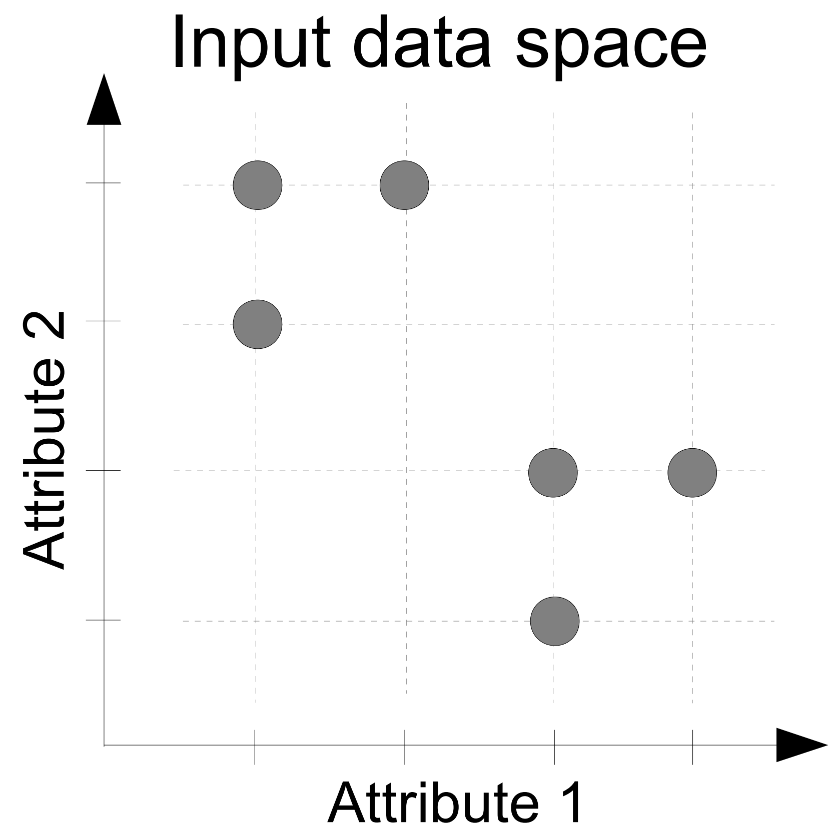
- Subjects answer a set of questions
- Most questions cannot be answered using number but sentences

- To make those answers comparable categories are introduced
- In most cases, the values of such categories cannot be ordered
  - They are nominal data


http://blog.mathsage.com/wp-content/uploads/2008/06/questionaire.jpg

# Nominal attributes

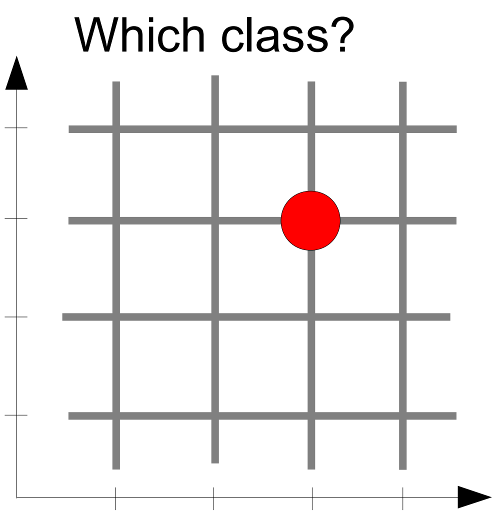
- Multi-dimensional
  - Attribute1={$a_{11}$,$a_{12}$,$a_{13}$,$a_{14}$}
  - Attribute2={$a_{21}$,$a_{22}$,$a_{23}$,$a_{24}$}
  - No natural order of $a_{ij}$
- Finit number of combinations
- Samples (e.g.)
  - $s_1$=[$a_{11}$,$a_{23}$]
  - $s_2$=[$a_{14}$,$a_{22}$,]
  - $s_3$=[$a_{11}$,$a_{24}$]

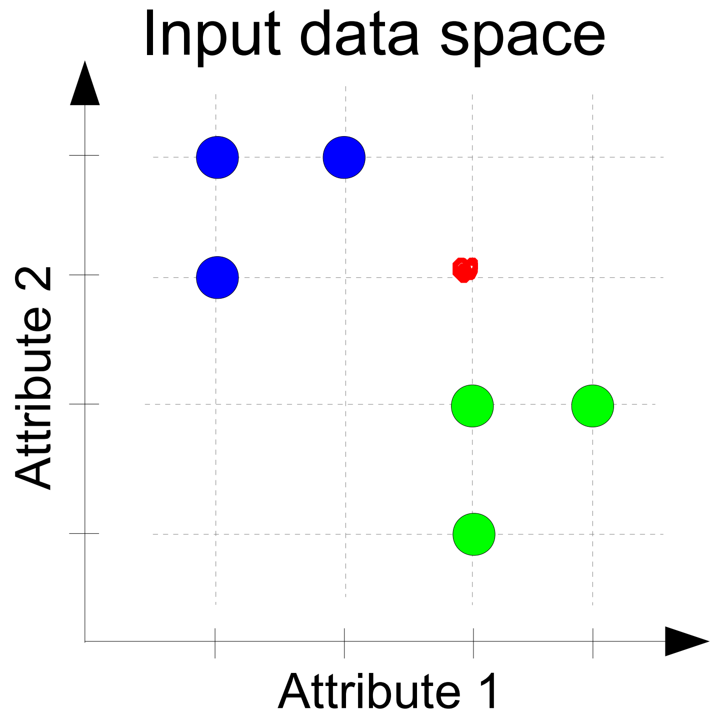


# Questionaire based prediction

- A prediction class is assigned to each sample
  - Two classes
  - How the sample looks like (questions)

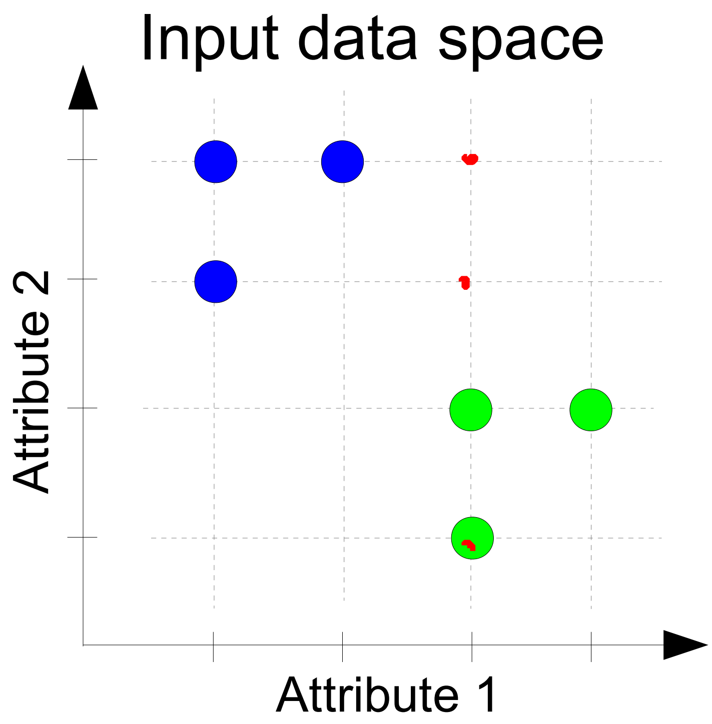
- Task: predict the class of an unseen sample

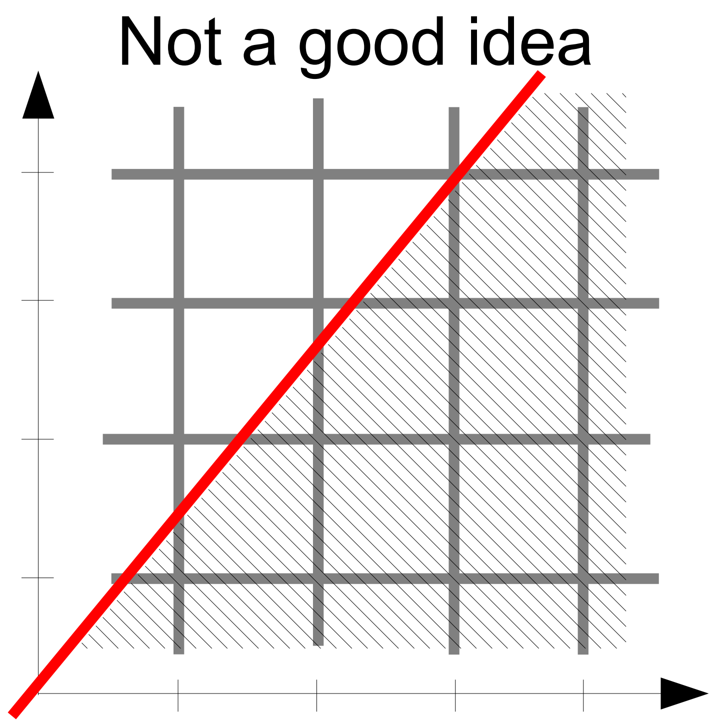


# Questionaire based prediction

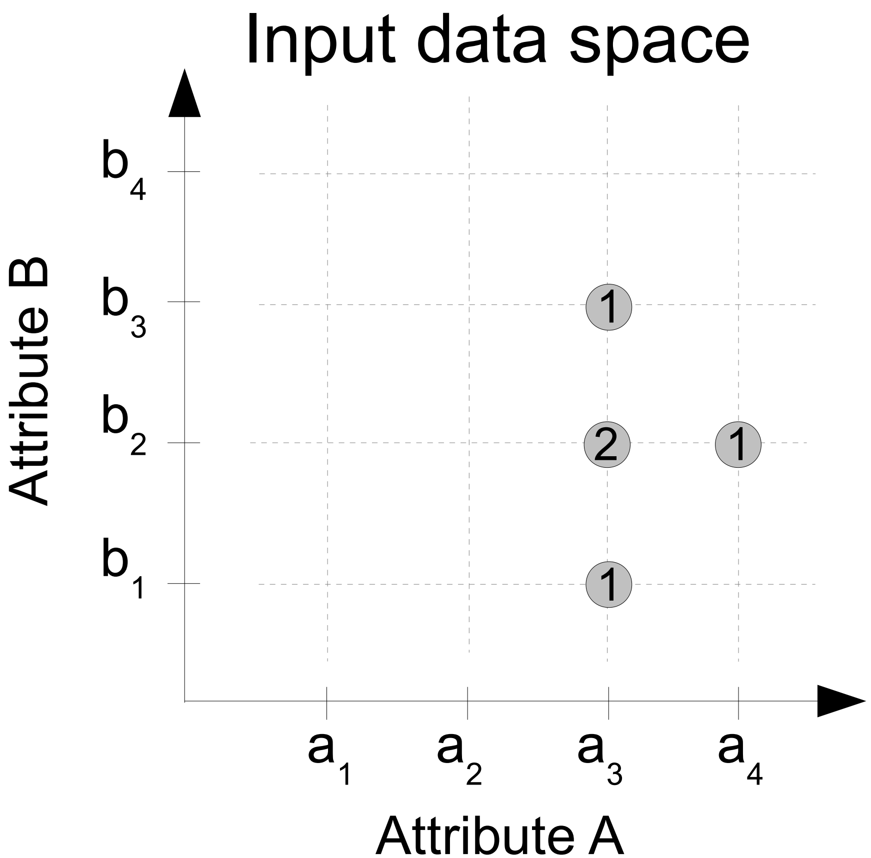
- Linear separation of input data space is not applicable



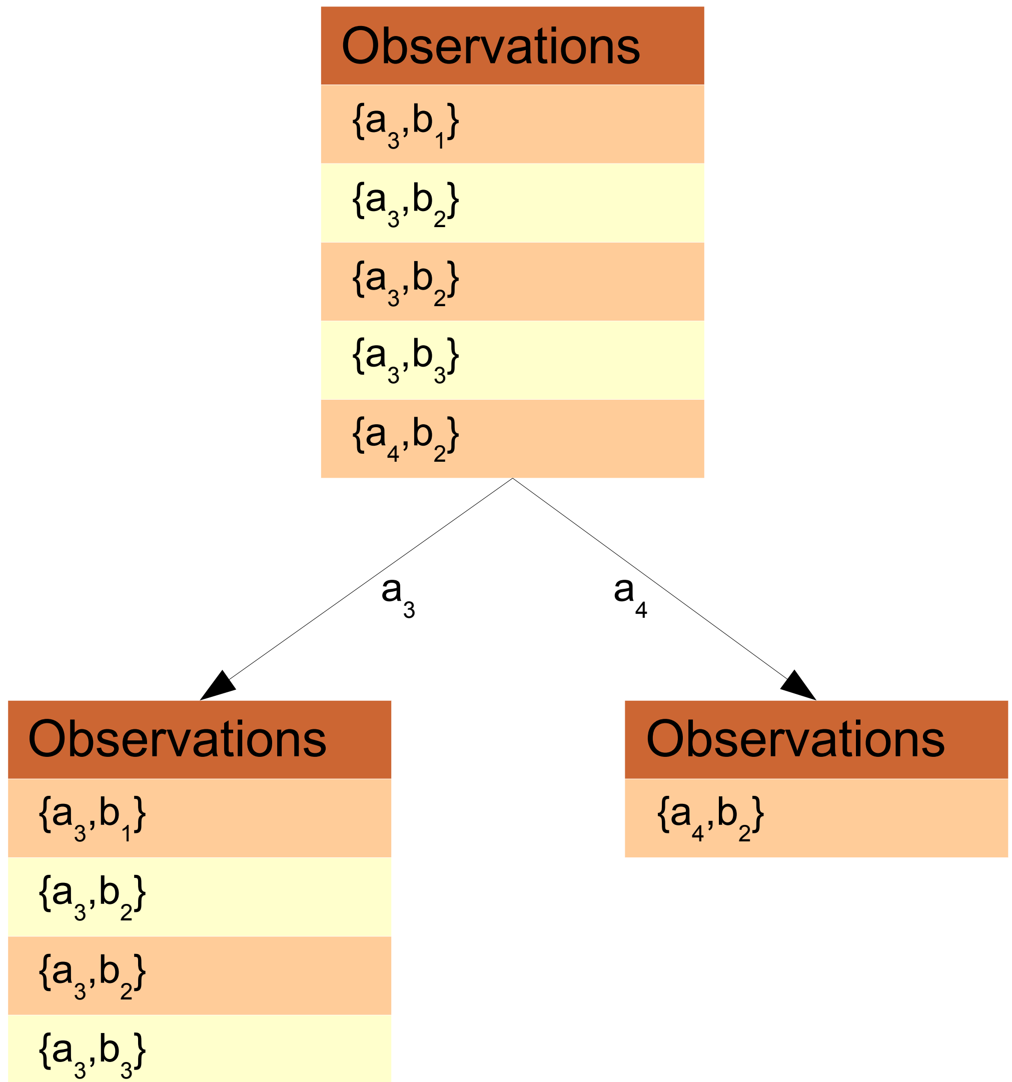
# Separation of nominal input space

- As attributes only have a limited number of values, we can use those values to split



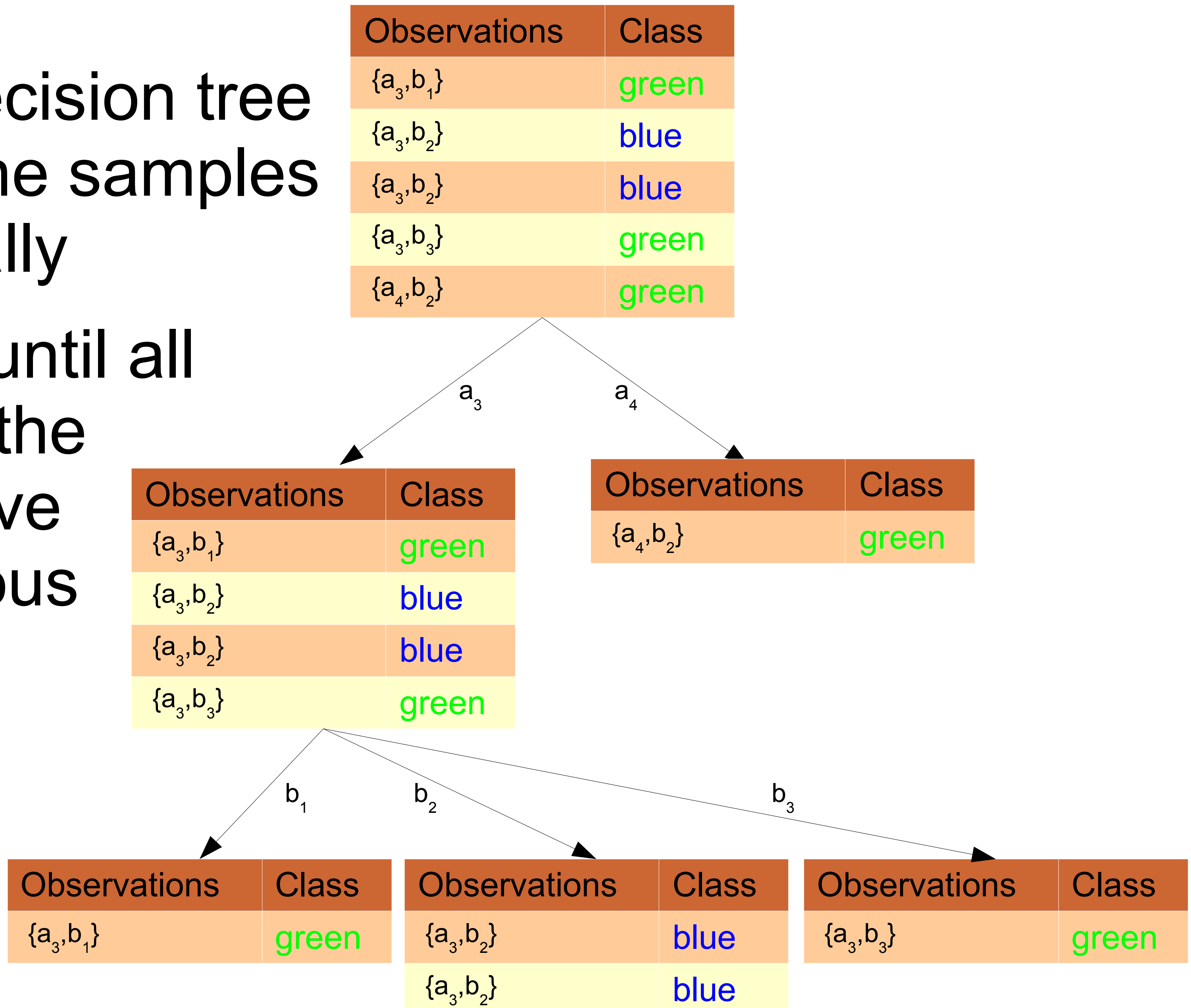

| Observations |
|---|
| $\{a_3,b_1\}$ |
| $\{a_3,b_2\}$ |
| $\{a_3,b_2\}$ |
| $\{a_3,b_3\}$ |
| $\{a_4,b_2\}$ |

# Splitting the samples using Attribute-Values



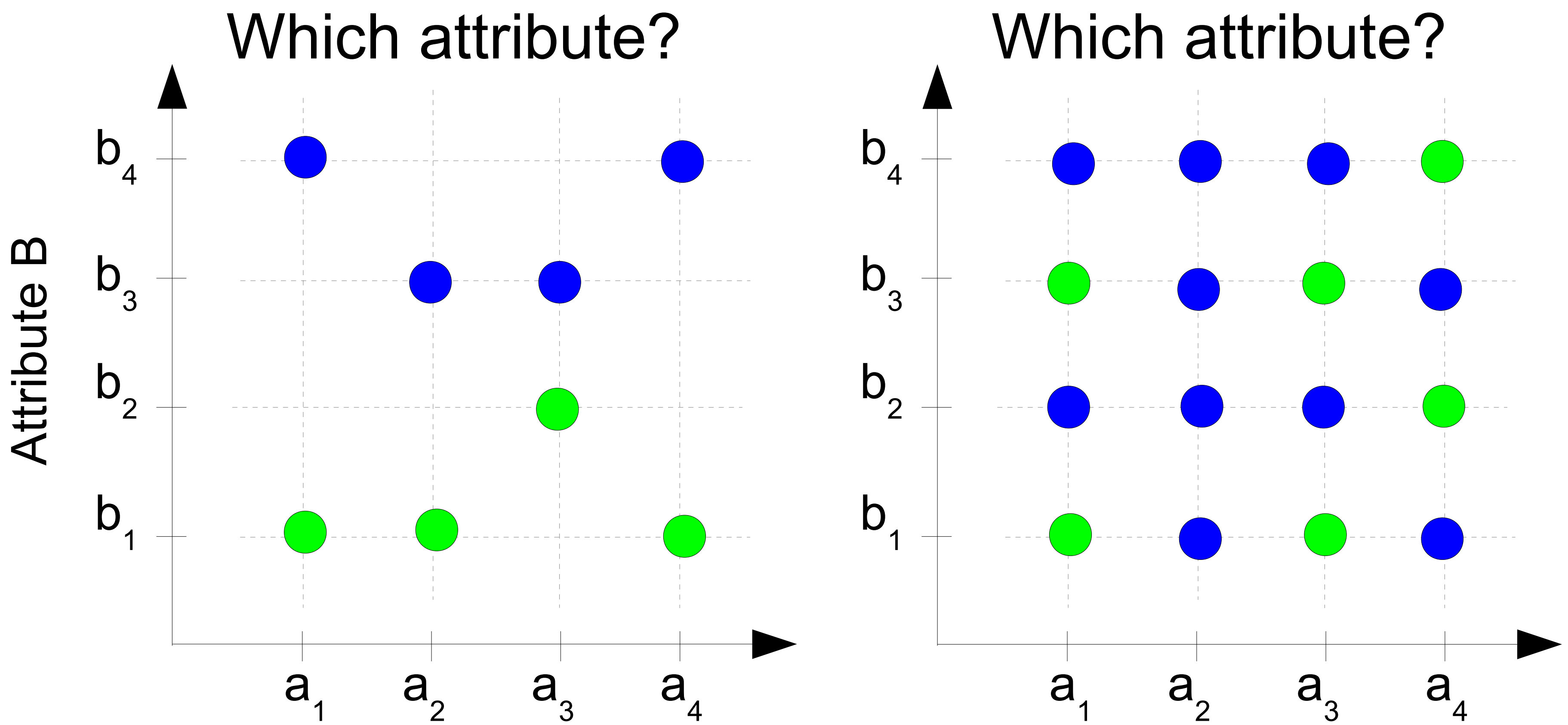
# Prediction using nominal attributes

- Create a decision tree that splits the samples hierar-chically
- Split again until all samples in the subTree have homogeneous class labels



# Wich attribute is best appropriate to separate the sample wrt to class

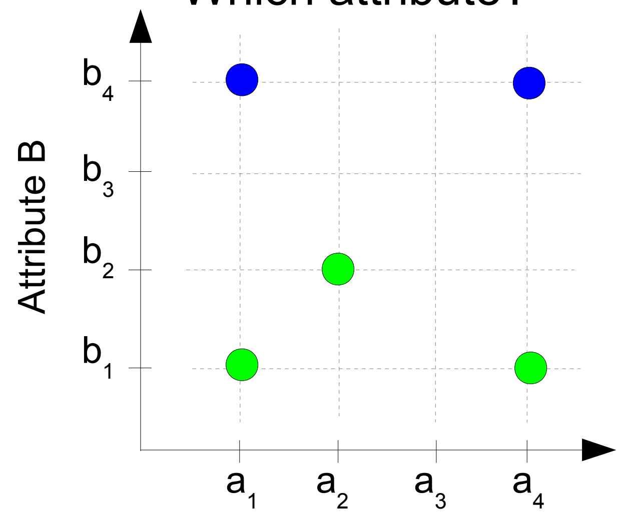
- Attribute A or B for splitting?



# ID3 Algorithm

- Split sample using an attribute
  - Such as the Subsets are mostly identical in their class labels
- A split using A returns $n_A$ subsets
  - $n_A$ is the number of values of A
- Here: use B for splitting as it produces homogeneous class labels in subsets

Which attribute?




Obervations

| Samples | Class |
|---|---|
| $\{a_1,b_1\}$ | green |
| $\{a_1,b_4\}$ | blue |
| $\{a_2,b_2\}$ | green |
| $\{a_4,b_1\}$ | green |
| $\{a_4,b_4\}$ | blue |

Split using A

| Samples | Class |
|---|---|
| $\{a_1,b_1\}$ | green |
| $\{a_1,b_4\}$ | blue |

| Samples | Class |
|---|---|
| $\{a_2,b_2\}$ | green |

| Samples | Class |
|---|---|
| $\{a_4,b_1\}$ | green |
| $\{a_4,b_4\}$ | blue |

Split using B

| Samples | Class |
|---|---|
| $\{a_1,b_1\}$ | green |
| $\{a_4,b_1\}$ | green |

| Samples | Class |
|---|---|
| $\{a_2,b_2\}$ | green |

| Samples | Class |
|---|---|
| $\{a_1,b_4\}$ | blue |
| $\{a_4,b_4\}$ | blue |

# ID3 algorithm

- Idea:
  - Use the increase of homogeneous class labels (also termed information gain) to decide which Attribute should be used for splitting the observations
  - If the class labels are not yet homogeneous in the resulting subSets (also termed subTrees) split them again.. and again .. and again until the class labels are homogeneous
  - Each split is a new branch and leaves “close” a path starting from the root (the root is where the first sample split was applied)
  - The leaves are then used as class labels for predicting the class of new observations

# Information Gain

- Compute the change of entropy when splitting the sample using Attribute A
  - S, the sample
  - A, the Attribute which is tested right now
  - $n_A$ the number of values in the attribute
  - $A_i$ the i-th value of Attribute A
  - $P_{Ai}$ the number of samples with $A=A_i$ divided by the number of all samples in the set
  - E(S) the entropy of set S regarding the class labels
  - $S_{Ai}$ the subset of S with values $A_i$ (all samples of S that have the value $A_i$ for their attribute A

$$G(S,A)=E(S)-\sum_{i=1}^{n_A} p_{A_i}\cdot E(S_{A_i})$$

# Information Gain

Difference between entropy before split and entropy after split measures the increase of homogeneity considering the class labels

Entropy of S:

A measure of how homogeneously the class labels are distributed in Sample S before splitting

Entropy of S after split by Attribute A:

A measure of how homogeneously the class labels are distributed in the subTrees, after the Sample S was split in $n_A$ subTrees according to the values of Attribute A

$$G(S,A) = E(S) - \sum_{i=1}^{n_A} p_{A_i} \cdot E(S_{A_i})$$

Entropy of $S_{Ai}$:

A measure of how homogeneously the class labels are distributed in Sample $S_{Ai}$

# ID3 algorithm

- ID3 (Examples, Target_Attribute, Attributes)
  - Create a root node for the tree
  - If all examples are positive, Return the single-node tree Root, with label = +.
  - If all examples are negative, Return the single-node tree Root, with label = -.
  - Otherwise Begin (to be continued on next slide)

# ID3 algorithm continued

- A = The Attribute that best classifies examples.
  - Decision Tree attribute for Root = A.
  - For each possible value, $v_i$, of A,
    - Add a new tree branch below Root, corresponding to the test (selection) A = $v_i$.
    - Let Examples($v_i$), be the subset of examples that have the value $v_i$ for A
      - below this new branch add the subtree **ID3 (Examples($v_i$), Target_Attribute, Attributes – {A})**

recursion

- End
- Return Root

# Practise

- New keyword: cell
- Create two samples with one attribute and one class
  - Ca 5-7 observations
  - Sample1: Attribute is highly correlated with class
  - Sample2:Attribute is not correlated with class label
- Create subsets (splitForAttributes.m) for both samples according to attribute
- Calculate the entropy of the respective subsets

getInformationGainAtt(data)

# Practise

- Create multidimensional sample (generateData)
- Use getInformationGainAtt to find out which attribute is most appropriate to split the sample
- Start ID3( data,level ) algorithms
  - Data the supervised data (attributes encoded in natural numbers and class labels)
  - Level is just a marker to trigger the recursion depth (use level=1 at call from matlab command line)

# Pruning

- Pruning (deutsch Gehölzschnitt)
- To avoid thousends of decision just cut the tree and generalize
- Can reduce the prediction error (over-fitting occurs as the tree contains to many decisions which may be driven by noise)

# What happens with noisy data

- In real world data, it can happen that the classes differ in observations even if the attribute-values are 100% identical
  - {a1,b3,c2,green}
  - {a1,b3,c2,blue}
  - {a1,b3,c2,blue}
- Predict the major class and hope for the best