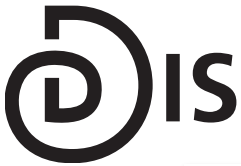
Learning and Data Mining

Abraham Bernstein

Universität Zürich[UZH]



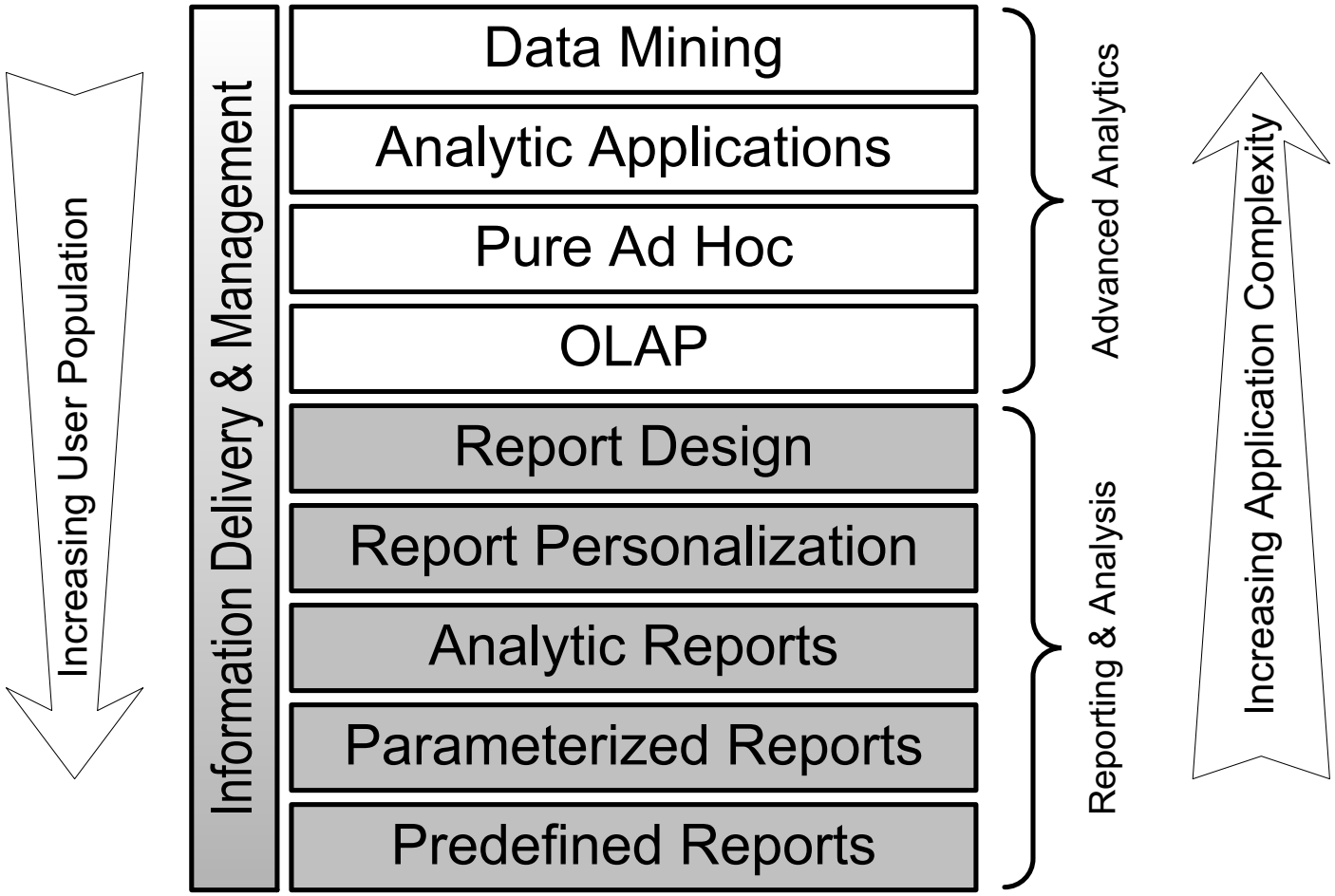
Dynamic and Distributed Information Systems

Note: Some slides were contributed by Andy Moore (CMU/Google); the lecture notes to Tan, Steinbach, and Kuma's Intro to DM book; and the slides for the Russel and Norvig AI book

Agenda

- **Machine Learning/Data Mining Basics**
  - **Why do we need mining or learning?**
  - **Learning as induction**
  - **Data Mining in a slide**
- Six Data Mining Tasks and their Evaluation
  - Classification - Learning Decision Trees
  - Class Probability Estimation – Naïve Bayes
  - Regression – Linear Regression & Neural Networks
  - Clustering – K-Means & Hierarchical Clustering
  - Association Rules – Apriori

The Data Analysis Technology Spectrum



Why Learning:
Here is a dataset…

age	*employmer*	*education*	*edun*	*marital*	…	*job*	*race*	*gender*	*hour*	*country*	*wealth*
39	State_gov	Bachelors	13	Never_mar	…	Adm_clerical	White	Male	40	United_States	poor
51	Self_emp_	Bachelors	13	Married	…	Exec_managerial	White	Male	13	United_States	poor
39	Private	HS_grad	9	Divorced	…	Handlers_cleaners	White	Male	40	United_States	poor
54	Private	11th	7	Married	…	Handlers_cleaners	Black	Male	40	United_States	poor
28	Private	Bachelors	13	Married	…	Prof_specialty	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	…	Exec_managerial	White	Female	40	United_States	poor
50	Private	9th	5	Married_sp	…	Other_service	Black	Female	16	Jamaica	poor
52	Self_emp_	HS_grad	9	Married	…	Exec_managerial	White	Male	45	United_States	rich
31	Private	Masters	14	Never_mar	…	Prof_specialty	White	Female	50	United_States	rich
42	Private	Bachelors	13	Married	…	Exec_managerial	White	Male	40	United_States	rich
37	Private	Some_coll	10	Married	…	Exec_managerial	Black	Male	80	United_States	rich
30	State_gov	Bachelors	13	Married	…	Prof_specialty	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	…	Adm_clerical	White	Female	30	United_States	poor
33	Private	Assoc_acc	12	Never_mar	…	Sales	Black	Male	50	United_States	poor
41	Private	Assoc_voc	11	Married	…	Craft_repair	Asian	Male	40	*MissingValue*	rich
34	Private	7th_8th	4	Married	…	Transport_moving	Amer_Indi	Male	45	Mexico	poor
26	Self_emp_	HS_grad	9	Never_mar	…	Farming_fishing	White	Male	35	United_States	poor
33	Private	HS_grad	9	Never_mar	…	Machine_op_inspct	White	Male	40	United_States	poor
38	Private	11th	7	Married	…	Sales	White	Male	50	United_States	poor
44	Self_emp_	Masters	14	Divorced	…	Exec_managerial	White	Female	45	United_States	rich
41	Private	Doctorate	16	Married	…	Prof_specialty	White	Male	60	United_States	rich
:	:	:	:	:	:	:	:	:	:	:	:

Why Learning: About this dataset

- It is a tiny subset of the 1990 US Census.
- It is publicly available online from the UCI Machine Learning Datasets repository

Used Attributes

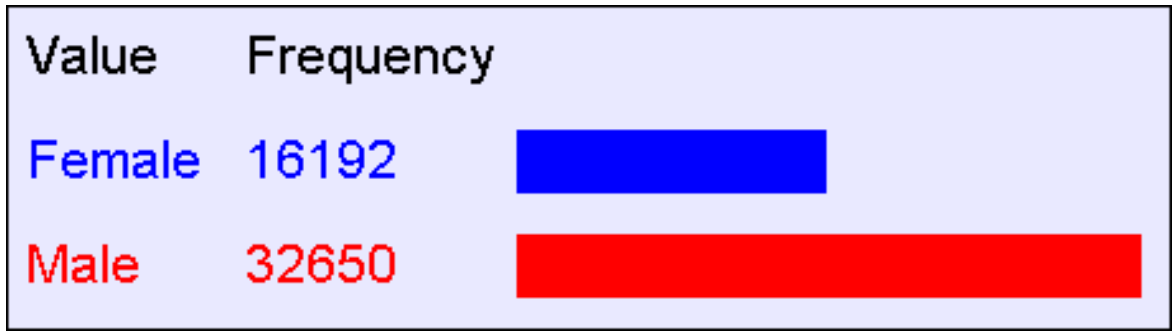
age	edunum	race	hours_worked
employment	marital	gender	country
taxweighting	job	capitalgain	wealth
education	relation	capitalloss	agegroup

This color = Real-valued This color = Symbol-valued

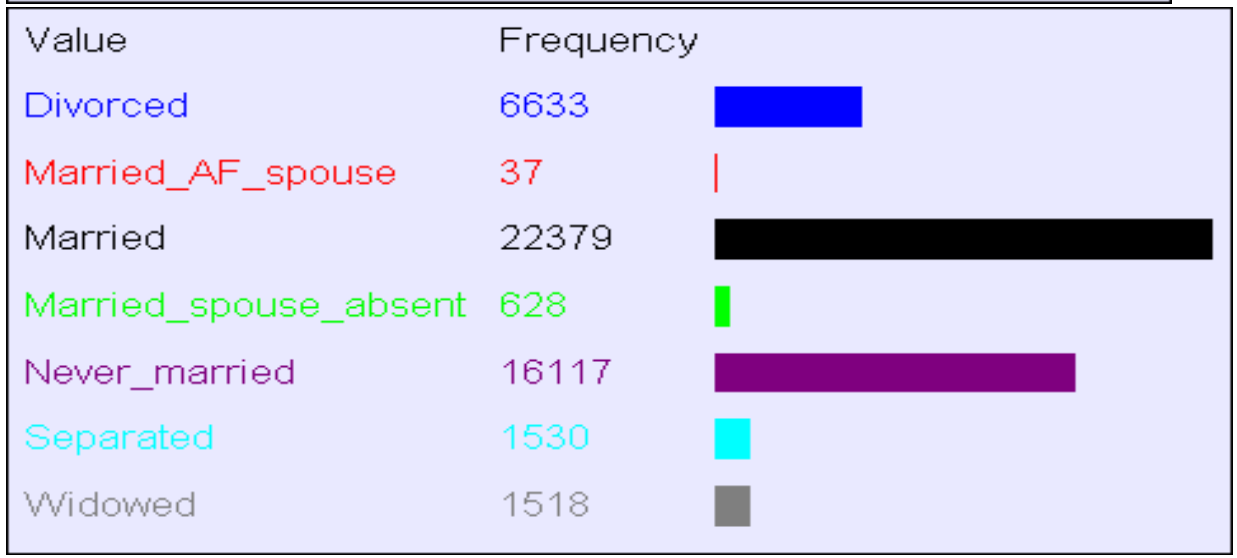
Successfully loaded a new dataset from the file \tadult.fds. It has 16 attributes and 48842 records.

Why Learning: What can you do with a dataset?

- Well, you can look at histograms…



Gender



Marital Status

Why Learning: Contingency Tables

- A better name for a histogram:

  A One-dimensional Contingency Table

- Recipe for making a k-dimensional contingency table:

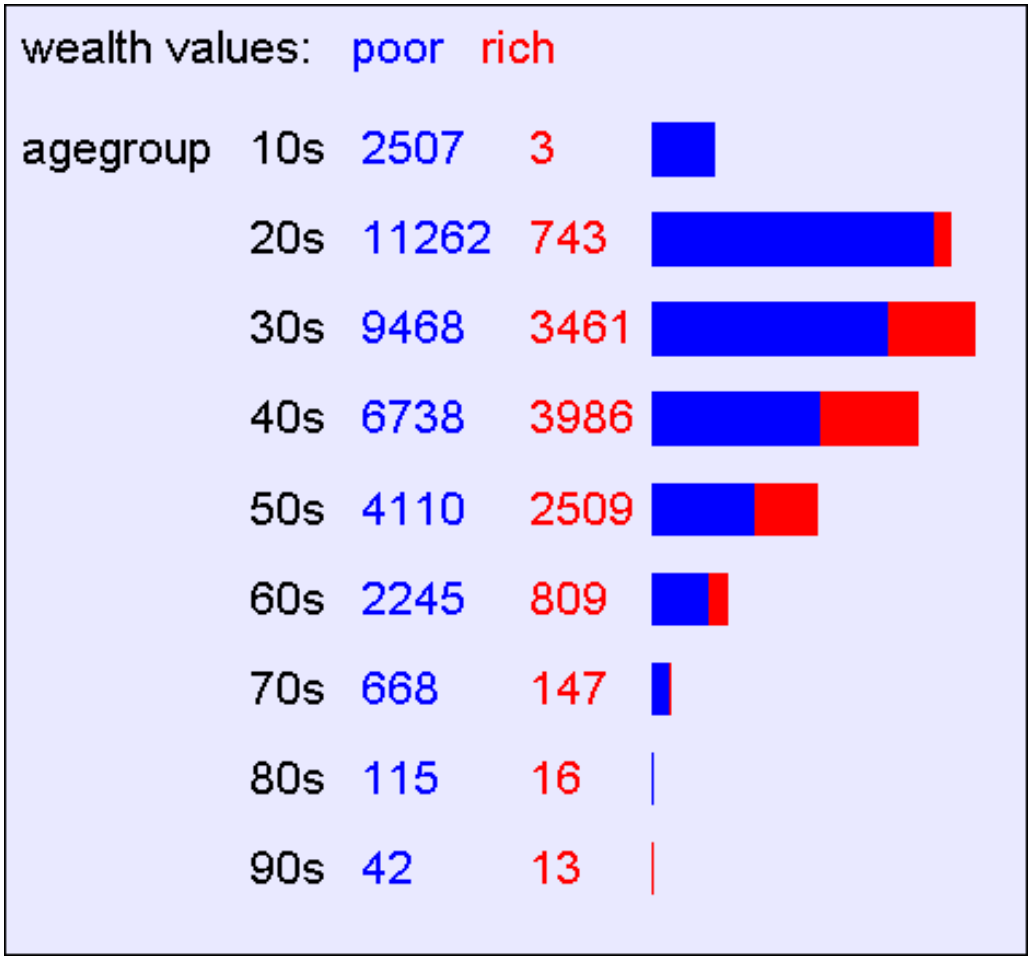
  1. Pick *k* attributes from your dataset. Call them $a_1, a_2, \ldots a_k$.
  2. For every possible combination of values, $a_1 = x_1$, $a_2 = x_2, \ldots a_k = x_k$, record how frequently that combination occurs

Why Learning: A 2-d Contingency Table

wealth values:		poor	rich
agegroup	10s	2507	3
	20s	11262	743
	30s	9468	3461
	40s	6738	3986
	50s	4110	2509
	60s	2245	809
	70s	668	147
	80s	115	16
	90s	42	13

- For each pair of values for attributes (agegroup,wealth) we can see how many records match.

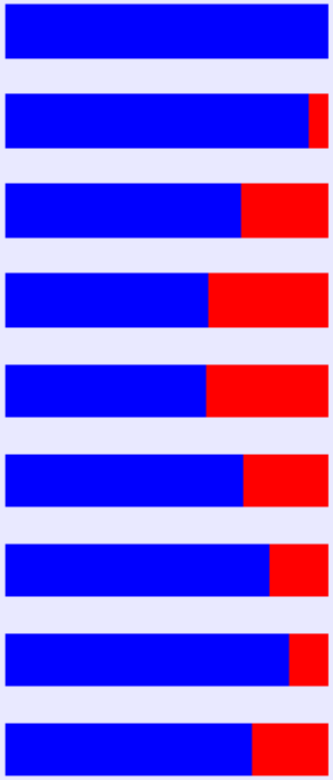
Why Learning: A 2-d Contingency Table



- Easier to appreciate graphically

Why Learning: A 2-d Contingency Table

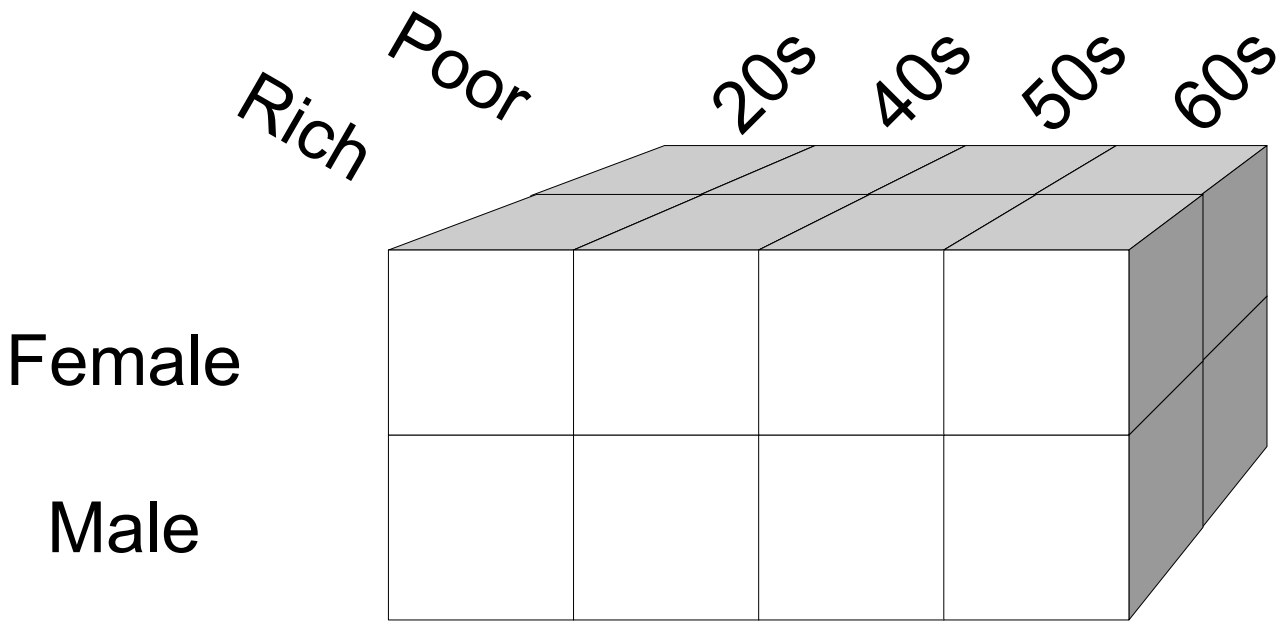
wealth values:	poor	rich
agegroup 10s	2507	3
20s	11262	743
30s	9468	3461
40s	6738	3986
50s	4110	2509
60s	2245	809
70s	668	147
80s	115	16
90s	42	13



- Easier to see “interesting” things if we stretch out the histogram bars

Why Learning: 3-d contingency tables

- These are harder to look at!



Why Learning: Let's think…

- With 16 attributes, how many 1-d contingency tables are there?
  - 16
- How many 2-d contingency tables?
  - 16-choose-2 = 16 * 15 / 2 = 120
- How many 3-d tables?
  - 560
- With 100 attributes how many 3-d tables are there?
  - 161,700

Manually looking at contingency tables

- Looking at one contingency table:
  - can be as much fun as reading an interesting book
- Looking at ten tables:
  - as much fun as watching CNN
- Looking at 100 tables:
  - as much fun as watching an infom
- Looking at 100,000 tables:
  - as much fun as a three-week Nov with a dying weasel.

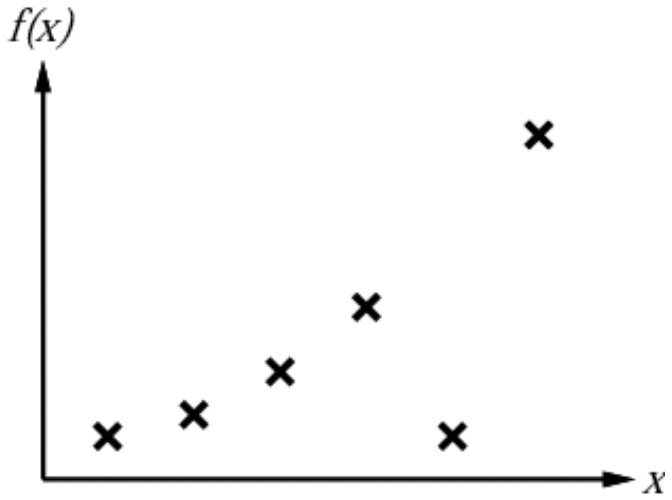
How about learning automatically where interesting structure is hiding?

So which contingency tables are interesting?

Inductive learning method

- Construct *h* to agree with *f* on training set
- (*h* is *consistent* if it agrees with *f* on all examples)

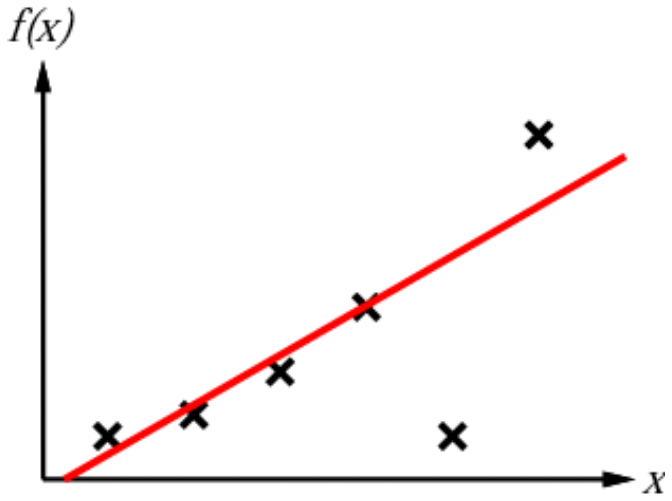
- E.g., curve fitting:



Inductive learning method

- Construct *h* to agree with *f* on training set
- (*h* is *consistent* if it agrees with *f* on all examples)

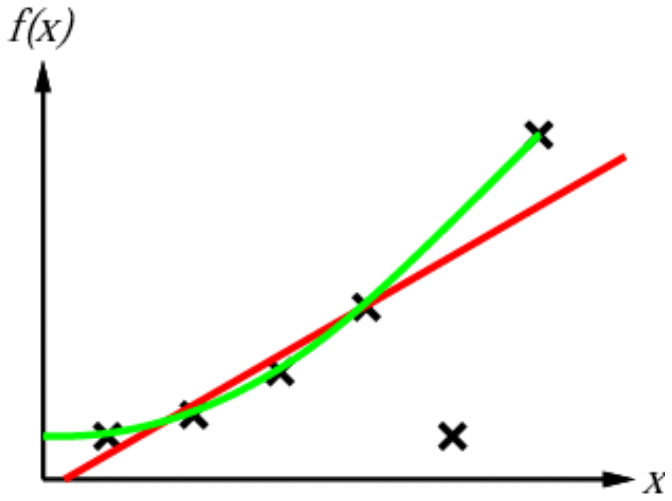
- E.g., curve fitting:



Inductive learning method

- Construct *h* to agree with *f* on training set
- (*h* is *consistent* if it agrees with *f* on all examples)

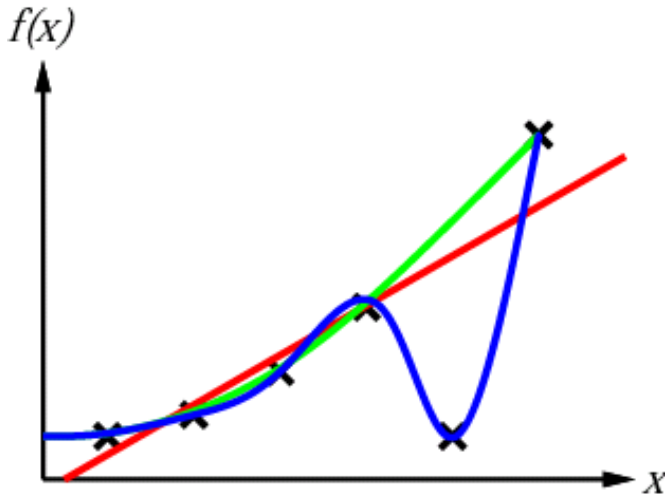
- E.g., curve fitting:



Inductive learning method

- Construct *h* to agree with *f* on training set
- (*h* is *consistent* if it agrees with *f* on all examples)

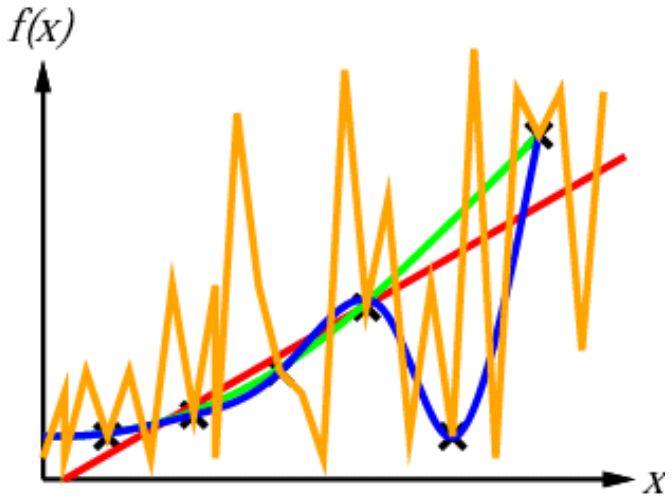
- E.g., curve fitting:



Inductive learning method

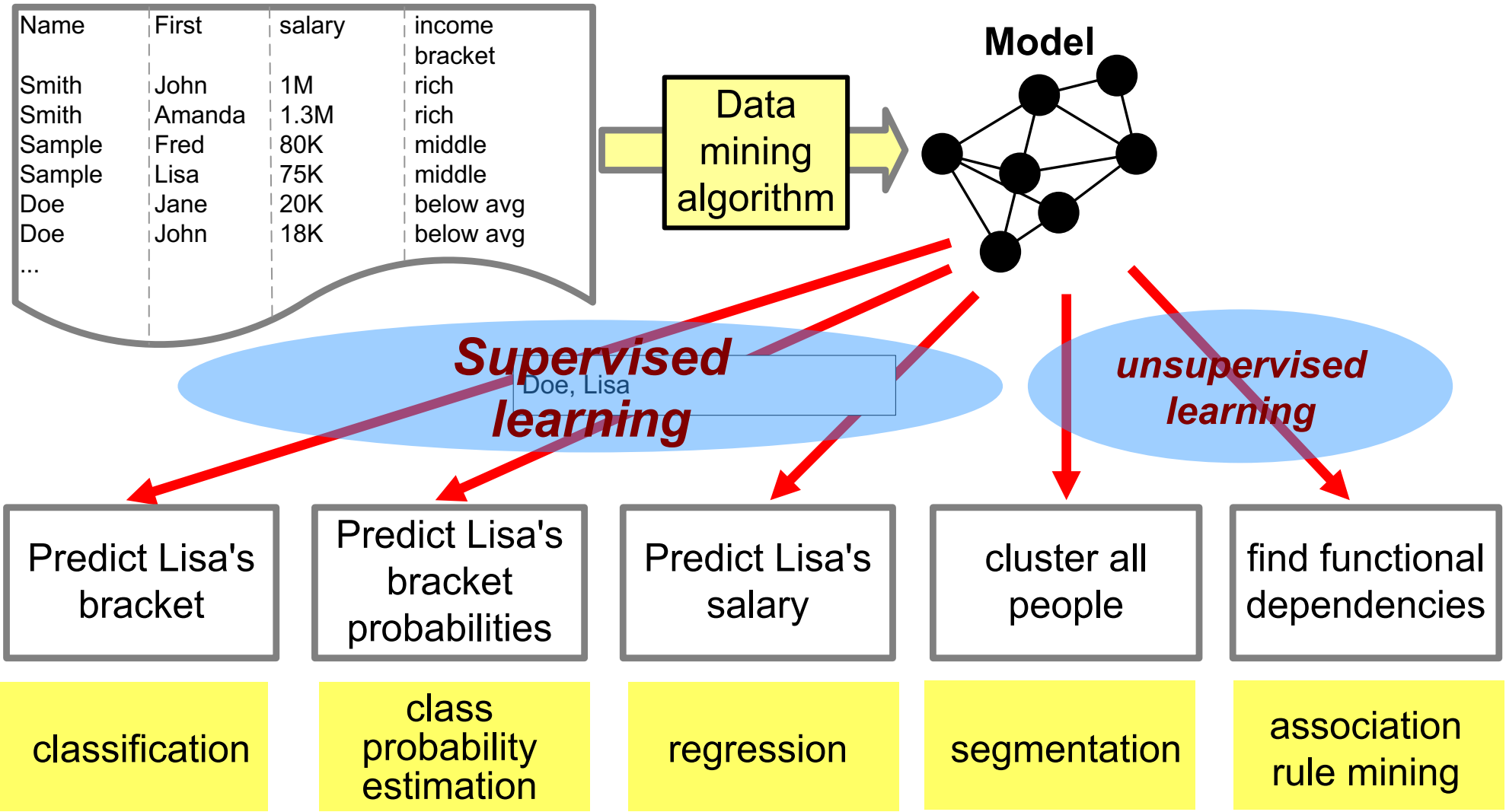
- Construct *h* to agree with *f* on training set
- (*h* is *consistent* if it agrees with *f* on all examples)

- E.g., curve fitting:

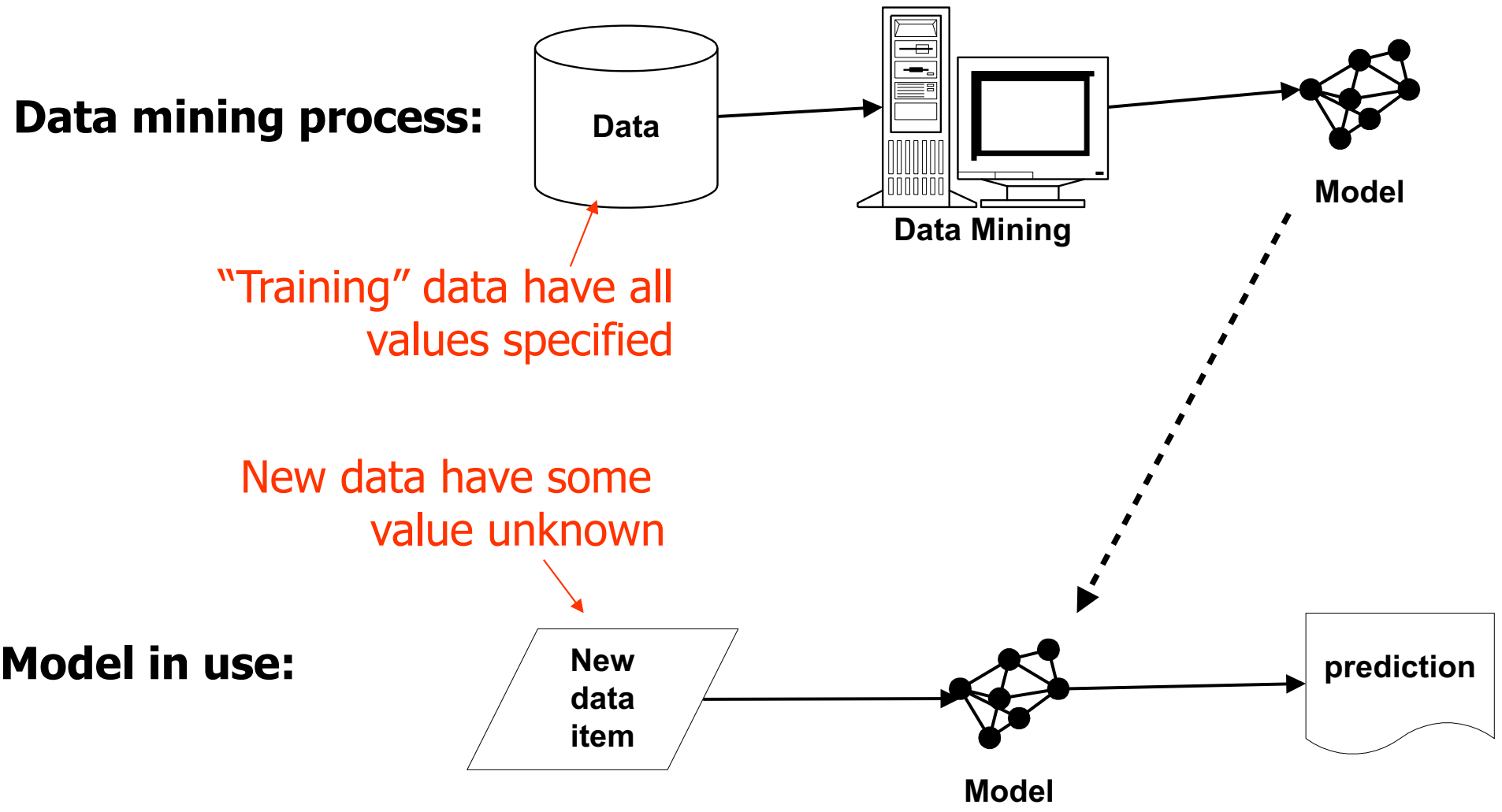


Ockham's razor: maximize a combination of consistency and simplicity

Data-Mining in a slide



Data mining versus use of model



Agenda

- Machine Learning/Data Mining Basics
  - Why do we need mining or learning?
  - Learning as induction
  - Data Mining in a slide
- **Six Data Mining Tasks and their Evaluation**
  - **Classification - Learning Decision Trees**
  - **Class Probability Estimation – Naïve Bayes**
  - **Regression – Linear Regression & Neural Networks**
  - **Clustering – K-Means & Hierarchical Clustering**
  - **Association Rules – Apriori**

Prediction: Classification

The Data Mining Workhorse

Universität Zürich[UZH]

DIS Dynamic and Distributed Information Systems

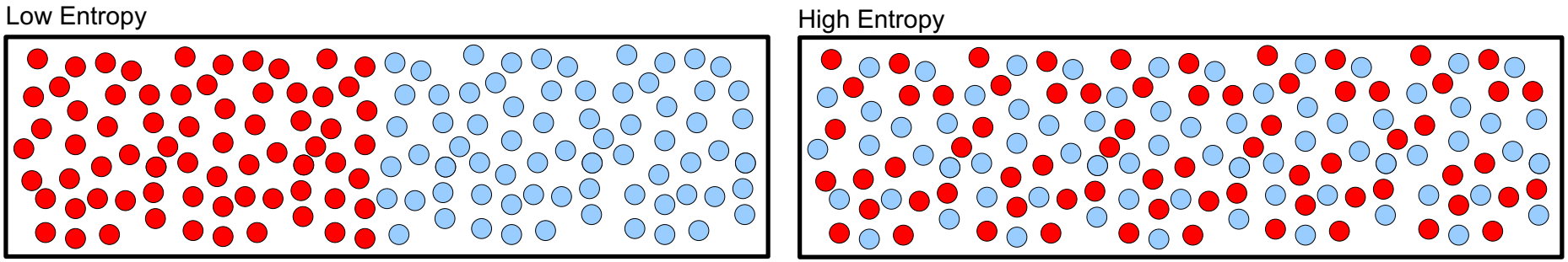
Manually looking at contingency tables

- Looking at one contingency table:
  - can be as much fun as reading an interesting book
- Looking at ten tables:
  - as much fun as watching CNN
- Looking at 100 tables:
  - as much fun as watching an infomercial
- Looking at 100,000 tables:
  - as much fun as a three-week Novembe with a dying weasel.

Information Theory to the rescue

So which contingency tables are interesting?

Entropy

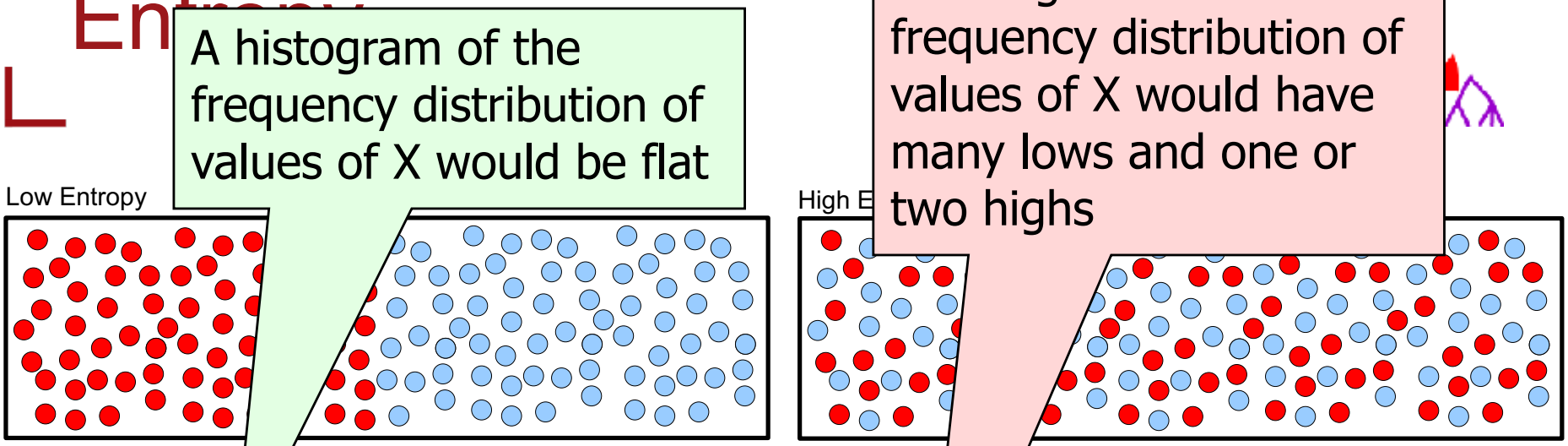


- "High Entropy" means X is from a uniform (boring) distribution
- "Low Entropy" means X is from varied (peaks and valleys) distribution
- Mathematically:

$$H(X) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_m \log_2 p_m$$

$$= -\sum_{j=1}^{m} p_j \log_2 p_j$$

Entropy

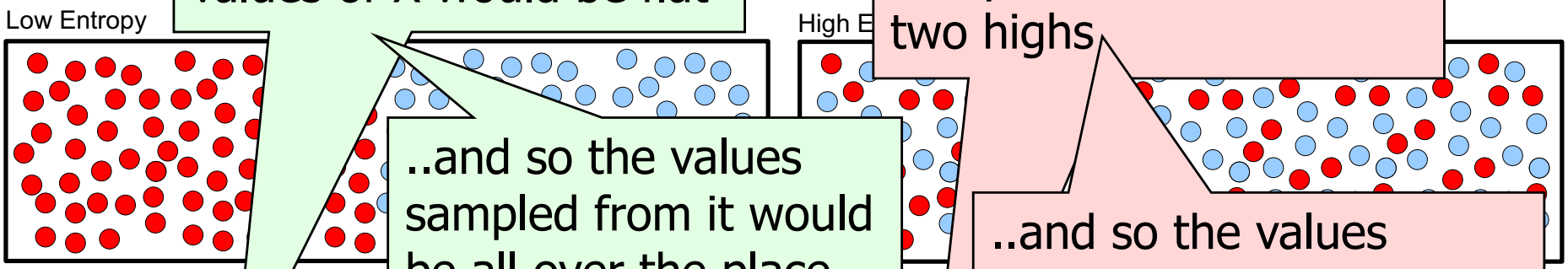


- "High Entropy" means X is from a uniform (boring) distribution
- "Low Entropy" means X is from varied (peaks and valleys) distribution
- Mathematically:

$$H(X) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_m \log_2 p_m$$

$$= -\sum_{j=1}^{m} p_j \log_2 p_j$$

Entropy



- "High Entropy" means X is from a uniform
- "Low Entropy" means X is from varied (peaks and valleys) distribution
- Mathematically:

$$H(X) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_m \log_2 p_m$$

$$= -\sum_{j=1}^{m} p_j \log_2 p_j$$

Conditional Entropy

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of general ***Conditional Entropy***:

$H(Y|X)$ = The average conditional entropy of Y

$= \Sigma_j Prob(X=v_j)\, H(Y \mid X = v_j)$

Example:

v_j	$Prob(X=v_j)$	$H(Y \mid X = v_j)$
Math	0.5	1
History	0.25	0
CS	0.25	0

$H(Y|X) = 0.5 * 1 + 0.25 * 0 + 0.25 * 0 = 0.5$

Information Gain

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of ***Information Gain***:

IG(Y|X) = I must transmit *Y*. How many bits on average would it save me if both ends of the line knew *X*?

$$IG(Y|X) = H(Y) - H(Y|X)$$

Example:

- H(Y) = 1
- H(Y|X) = 0.5
- Thus IG(Y|X) = 1 – 0.5 = 0.5

Information Gain

- Suppose you are trying to predict whether someone is going live past 80 years. From historical data you might find…
  - IG(LongLife | HairColor) = 0.01
  - IG(LongLife | Smoker) = 0.2
  - IG(LongLife | Gender) = 0.25
  - IG(LongLife | LastDigitOfSSN) = 0.00001
- IG tells you how interesting a 2-d contingency table is going to be.

Definition of general ***Conditional Entropy***:

$H(Y|X)$ = The average conditional entropy of Y

$= \Sigma_j Prob(X=v_j)\, H(Y \mid X = v_j)$

Definition of ***Information Gain***:

$IG(Y|X) = H(Y) - H(Y \mid X)$

Searching for High Info Gains

- Given something (e.g. wealth) you are trying to predict, it is easy to ask the computer to find which attribute has highest information gain for it.

wealth values: poor rich

relation	Husband	10870	8846		H(wealth | relation = Husband) = 0.992385
	Not_in_family	11307	1276		H(wealth | relation = Not_in_family) = 0.473439
	Other_relative	1454	52		H(wealth | relation = Other_relative) = 0.216617
	Own_child	7470	111		H(wealth | relation = Own_child) = 0.110192
	Unmarried	4816	309		H(wealth | relation = Unmarried) = 0.328606
	Wife	1238	1093		H(wealth | relation = Wife) = 0.997207

H(wealth) = 0.793844 H(wealth|relation) = 0.628421

IG(wealth|relation) = 0.165423

Learning Decision Trees

- A Decision Tree is a tree-structured plan of a set of attributes to test in order to predict the output.
- To decide which attribute should be tested first, simply find the one with the highest information gain.
- Then recurse…

A small dataset: Miles Per Gallon

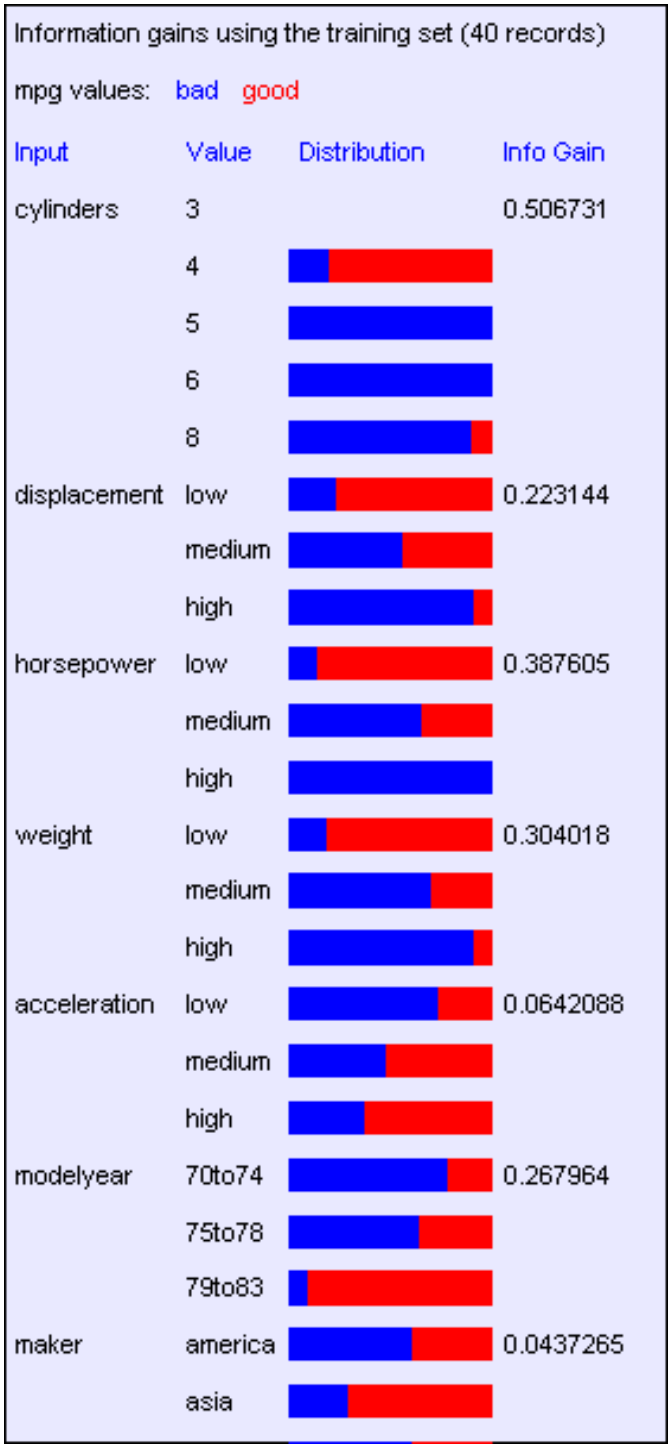
40 Records

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

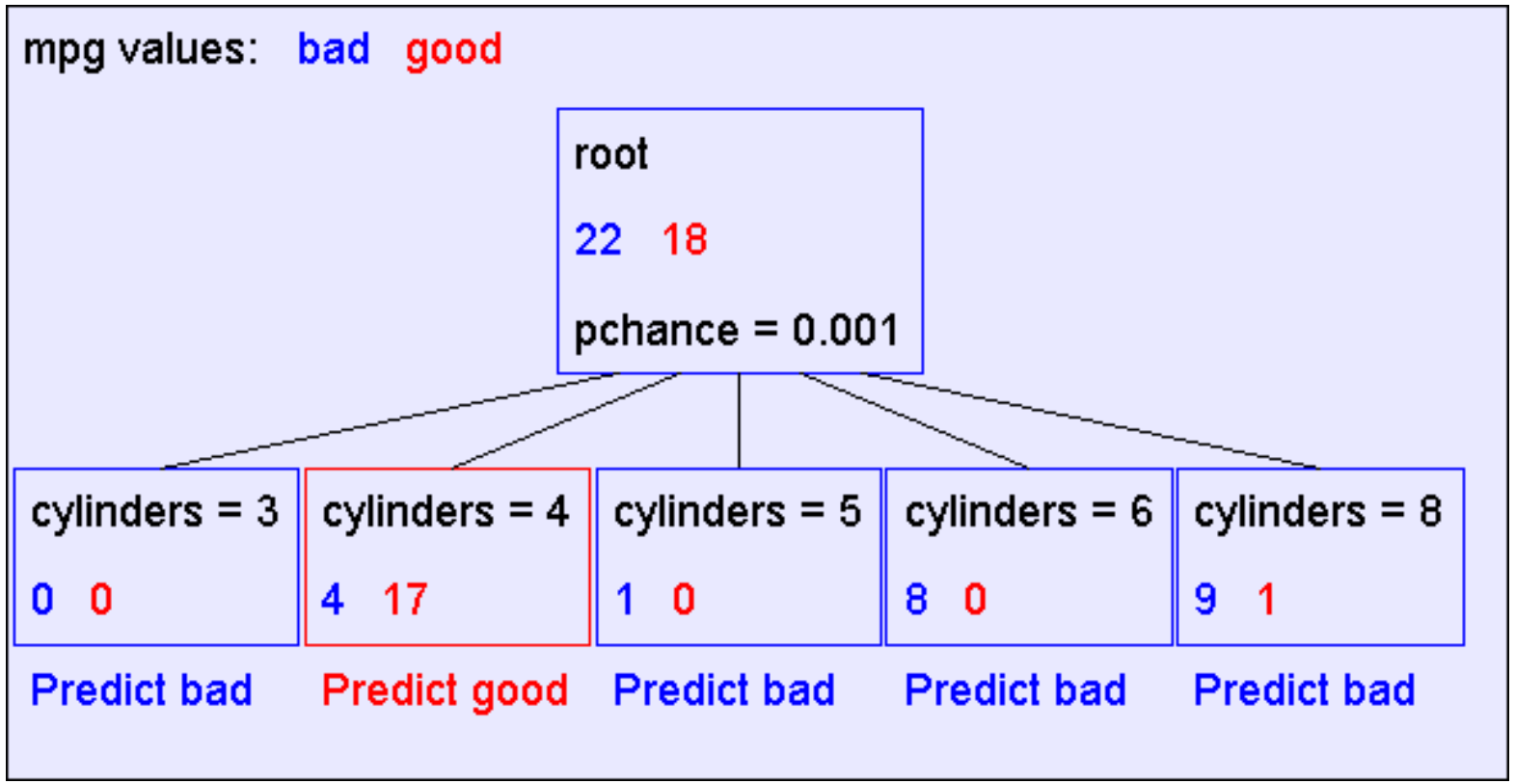
From the UCI repository (thanks to Ross Quinlan)

Suppose we want to predict MPG.

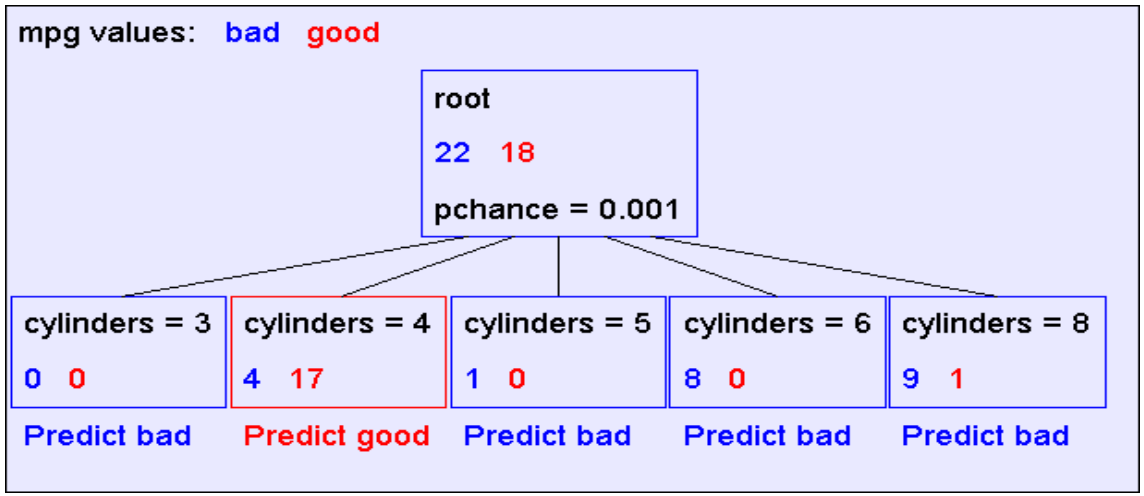
Look at all the information gains…



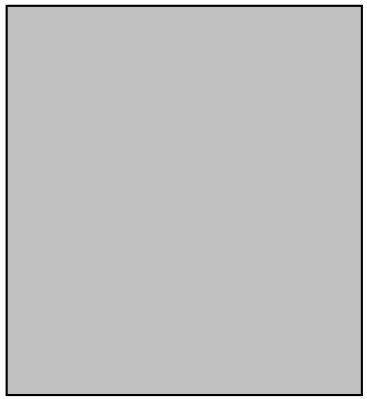
A Decision Stump



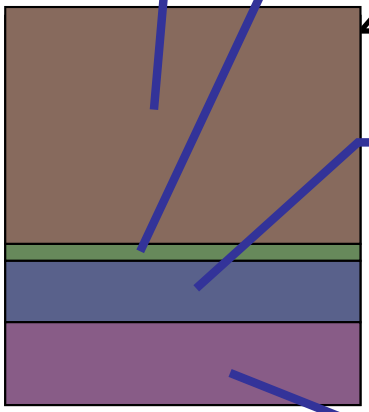
Recursion Step



Take the Original Dataset..



And partition it according to the value of the attribute we split on

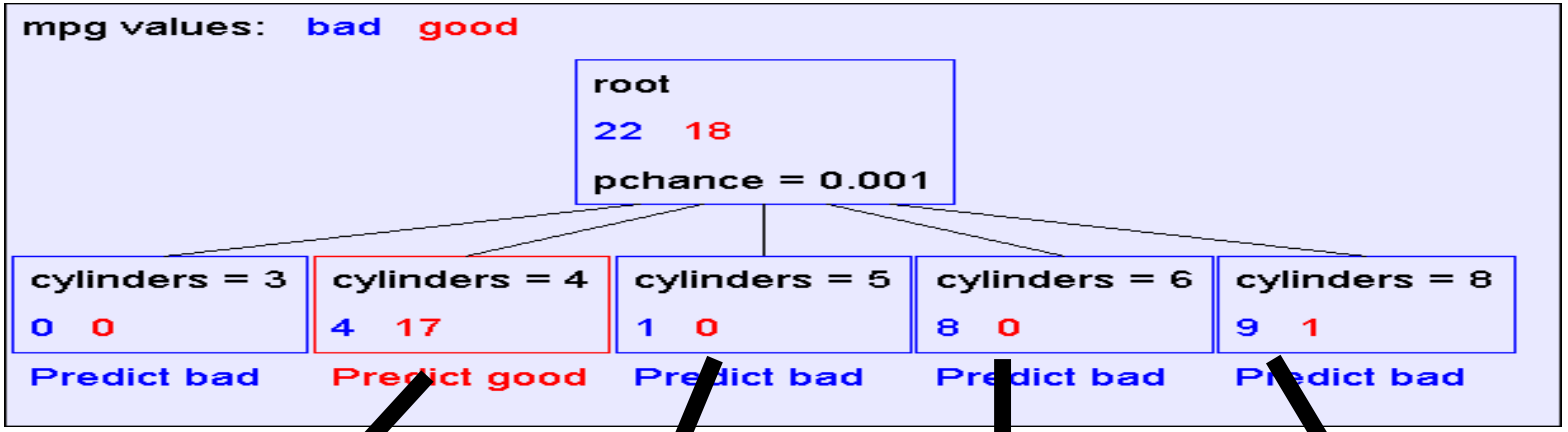


Records in which cylinders = 4

Records in which cylinders = 5

Records in which cylinders = 6

Recursion Step

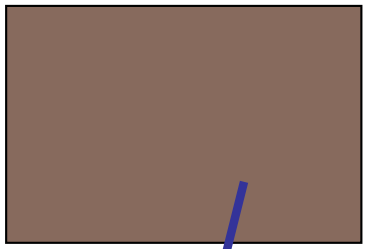


Build tree from These records..

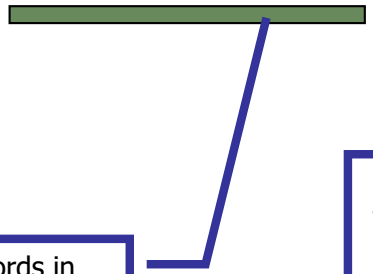
Build tree from These records..

Build tree from These records..

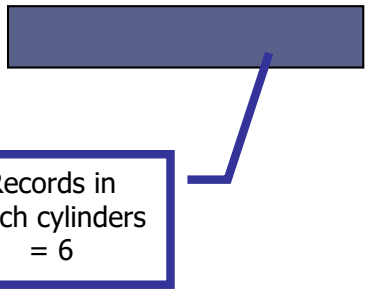
Build tree from These records..



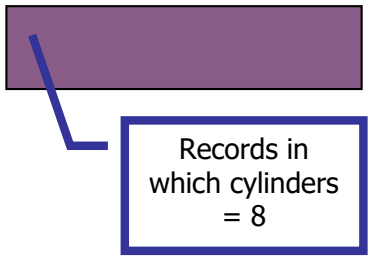
Records in which cylinders = 4



Records in which cylinders = 5

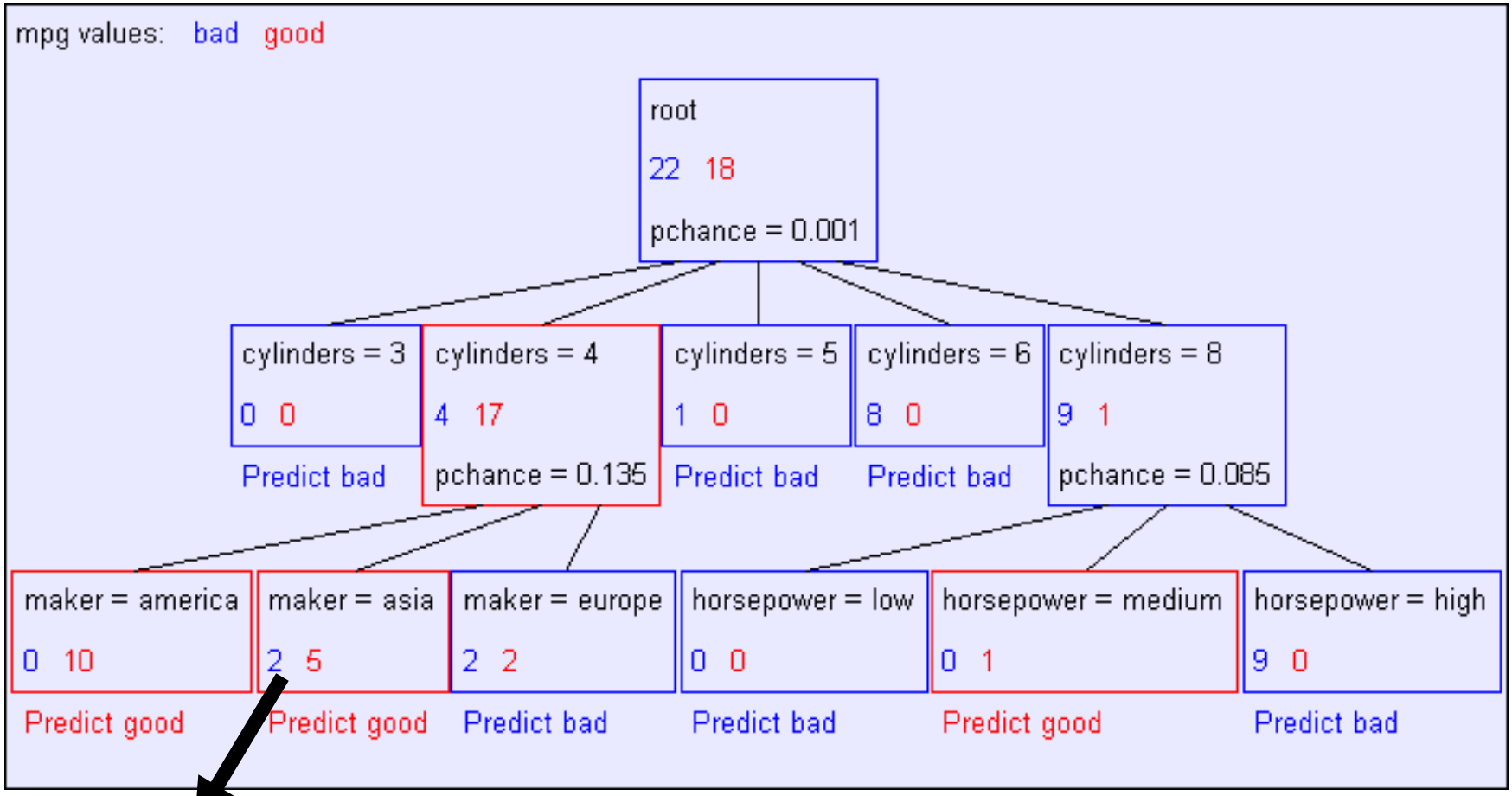


Records in which cylinders = 6



Records in which cylinders = 8

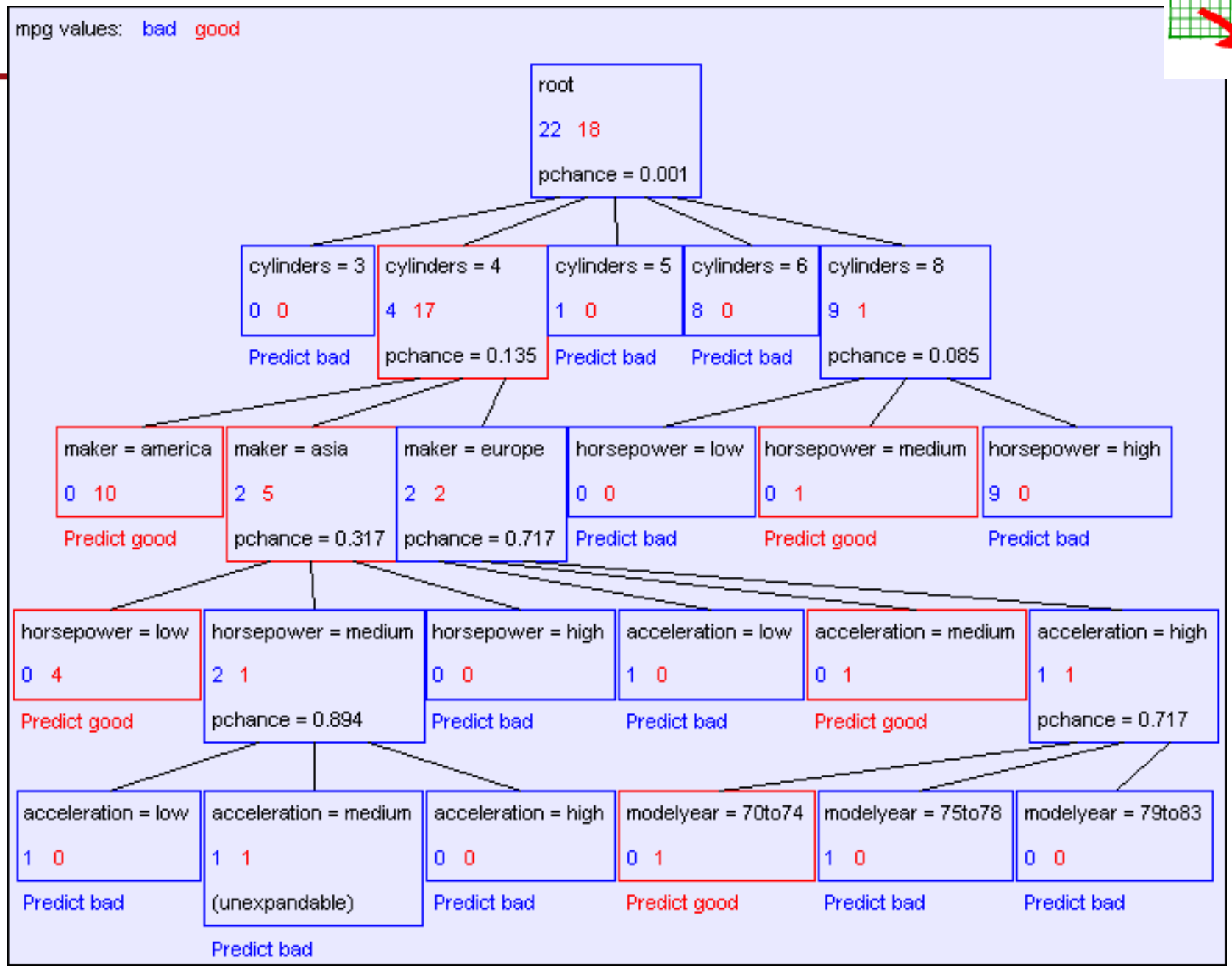
Second level of tree



Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)

The final tree



Basic Decision Tree Building Summarized

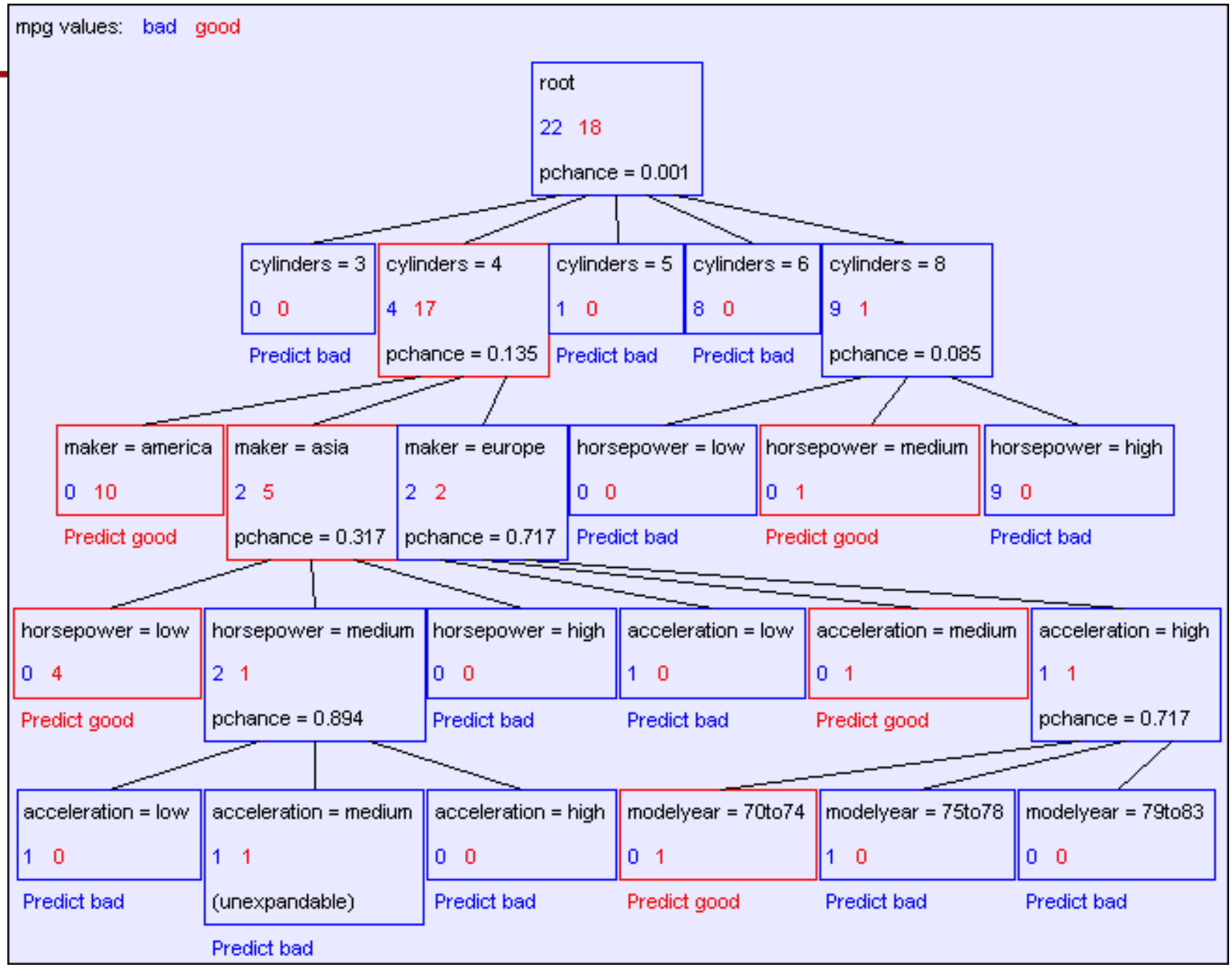
- BuildTree(DataSet,Output)
- If all output values are the same in DataSet, return a leaf node that says “predict this unique output”
- If all input values are the same, return a leaf node that says “predict the majority output”
- Else find attribute X with highest Info Gain

  Suppose X has nX distinct values
  (i.e. X has arity nX).
  - Create and return a non-leaf node with nX children.
  - The i’th child should be built by calling BuildTree(DSi,Output)
    - Where DSi built consists of all those records in DataSet for which X = ith distinct value of X.

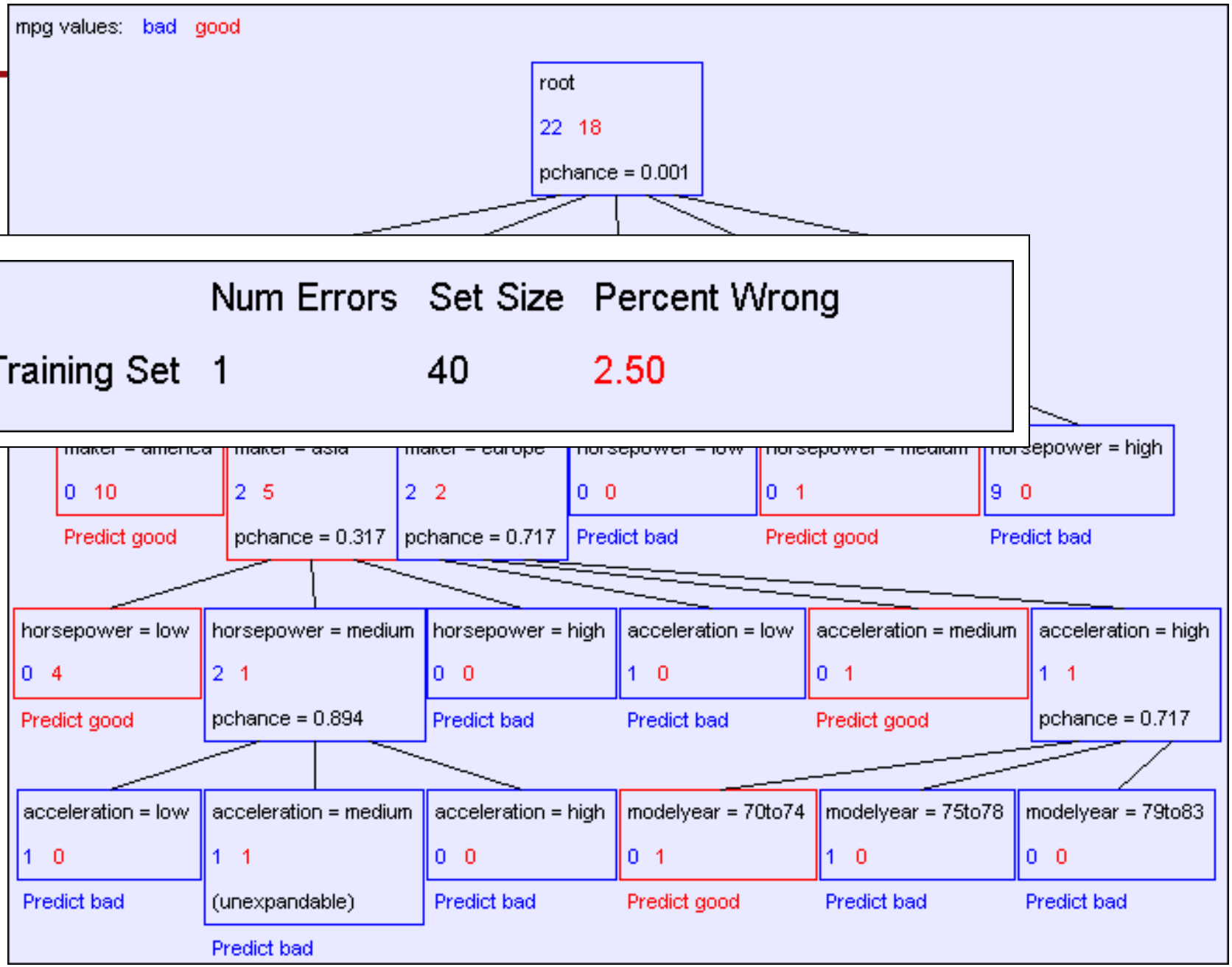
Training Set Error

- For each record, follow the decision tree to see what it would predict
  - For what number of records does the decision tree's prediction disagree with the true value in the database?
- This quantity is called the ***training set error***. The smaller the better.

MPG Training set error

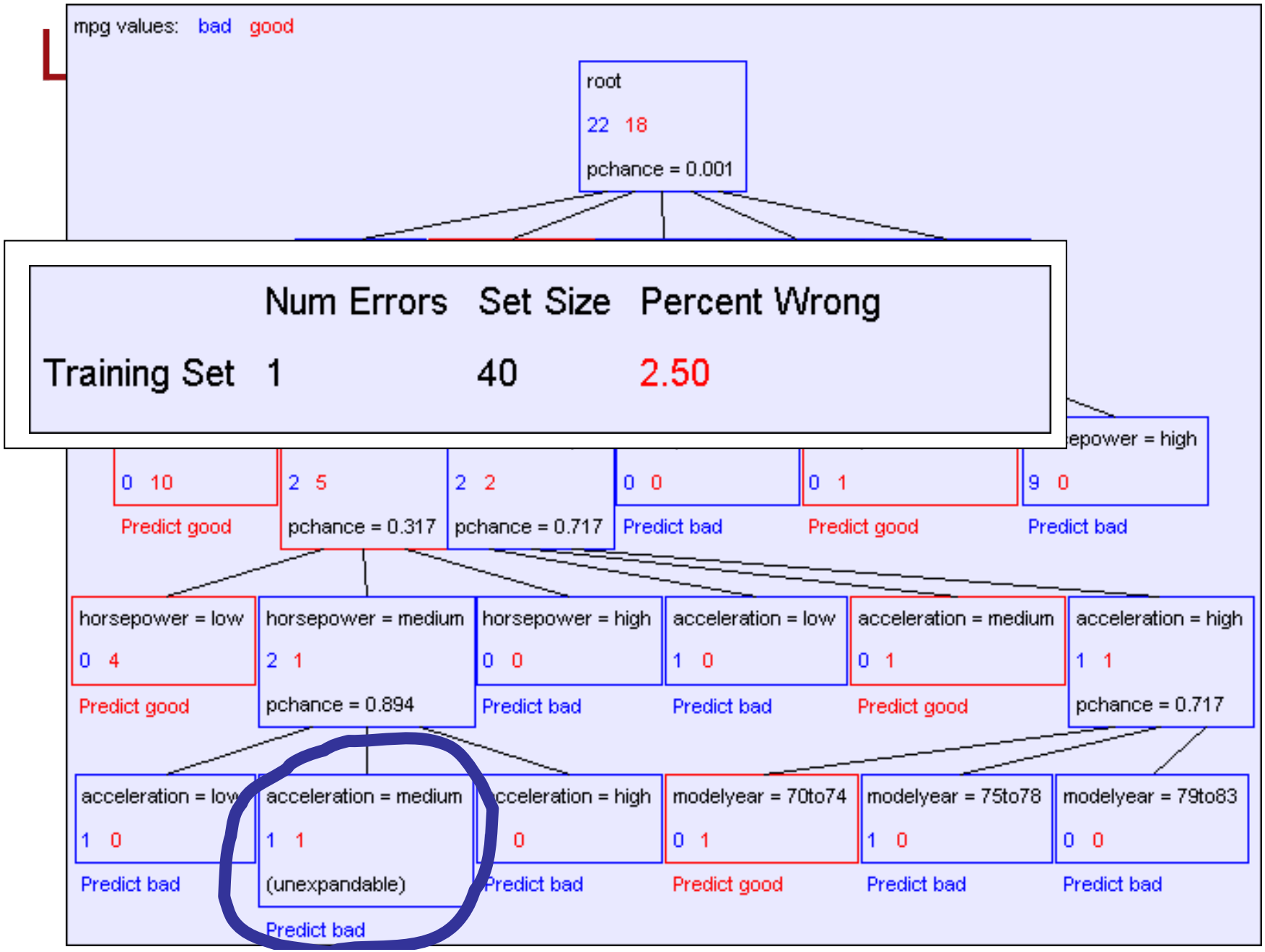


MPG Training set error



	Num Errors	Set Size	Percent Wrong
Training Set	1	40	2.50

MPG Training set error



Stop and reflect: Why are we doing this learning anyway?

- It is not usually in order to predict the training data's output on data we have already seen.

Stop and reflect: Why are we doing this learning anyway?

- It is not usually in order to predict the training data's output on data we have already seen.
- It is more commonly in order to predict the output value for ***future data*** we have not yet seen.

Stop and reflect: Why are we doing this learning anyway?

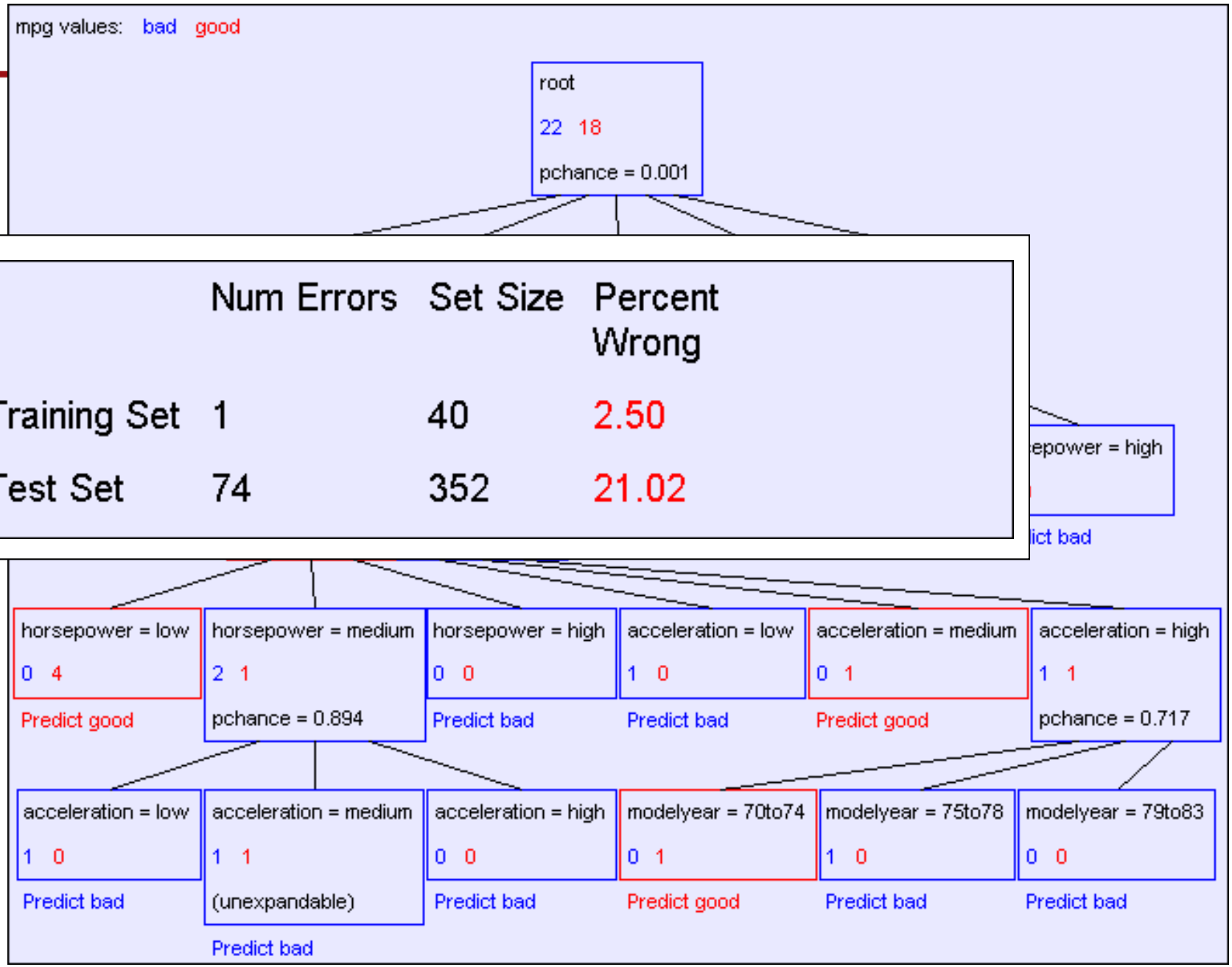
- It is not usually in order to predict the training data's output on data we have already seen.
- It is more commonly in order to predict the output value for ***future data*** we have not yet seen.

Warning: A common data mining misperception is that the above two bullets are the only possible reasons for learning. There are at least a dozen others.

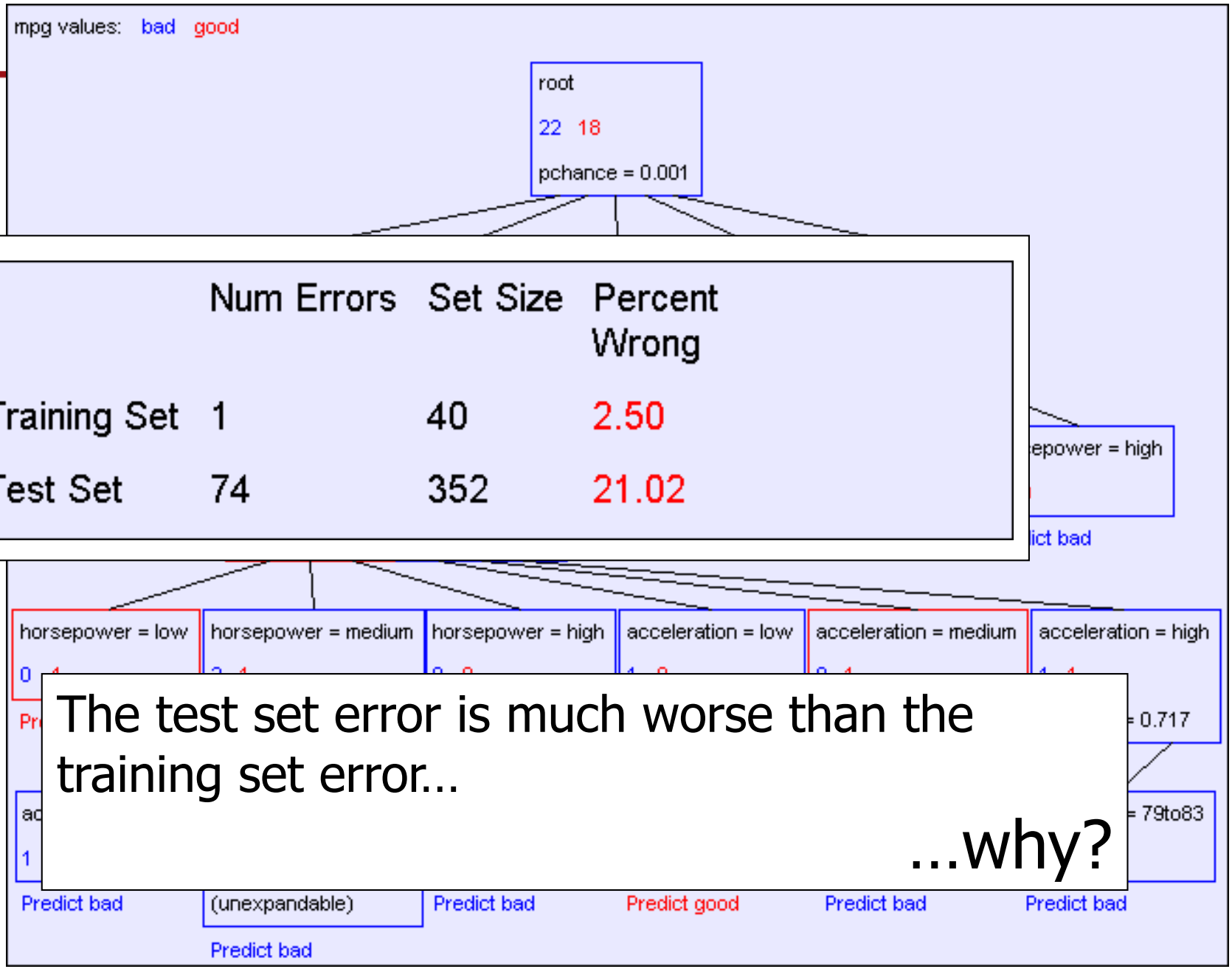
Test Set Error

- Suppose we are forward thinking.
- We hide some data away when we learn the decision tree.
- But once learned, we see how well the tree predicts that data.
- This is a good simulation of what happens when we try to predict future data.
- And it is called ***Test Set Error***.

MPG Test set error



MPG Test set error



An artificial example

- We'll create a training dataset

Five inputs, all bits, are generated in all 32 possible combinations

Output y = copy of e, Except a random 25% of the records have y set to the opposite of e

32 records

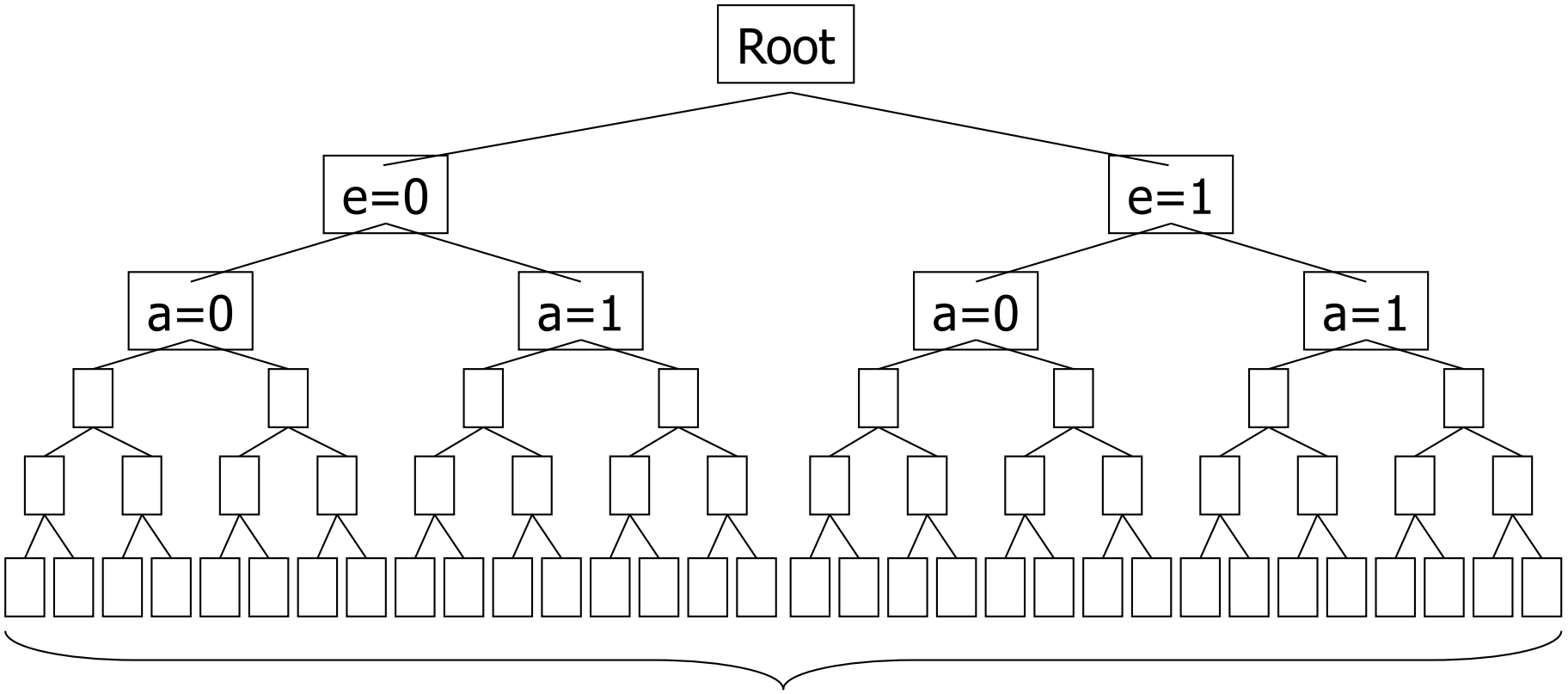
a	b	c	d	e	y
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	1
0	0	1	0	0	1
:	:	:	:	:	:
1	1	1	1	1	1

In our artificial example

- Suppose someone generates a test set according to the same method.
- The test set is identical, except that some of the y's will be different.
- Some y's that were corrupted in the training set will be uncorrupted in the testing set.
- Some y's that were uncorrupted in the training set will be corrupted in the test set.

Building a tree with the artificial training set

- Suppose we build a full tree (we always split until base case 2)



25% of these leaf node labels will be corrupted

Training set error for our artificial tree

All the leaf nodes contain exactly one record

and so…

→We would have a training set error of zero

Testing the tree with the test set

	1/4 of the tree nodes are corrupted	3/4 are fine
1/4 of the test set records are corrupted	1/16 of the test set will be correctly predicted for the wrong reasons	3/16 of the test set will be wrongly predicted because the test record is corrupted
3/4 are fine	3/16 of the test predictions will be wrong because the tree node is corrupted	9/16 of the test predictions will be fine

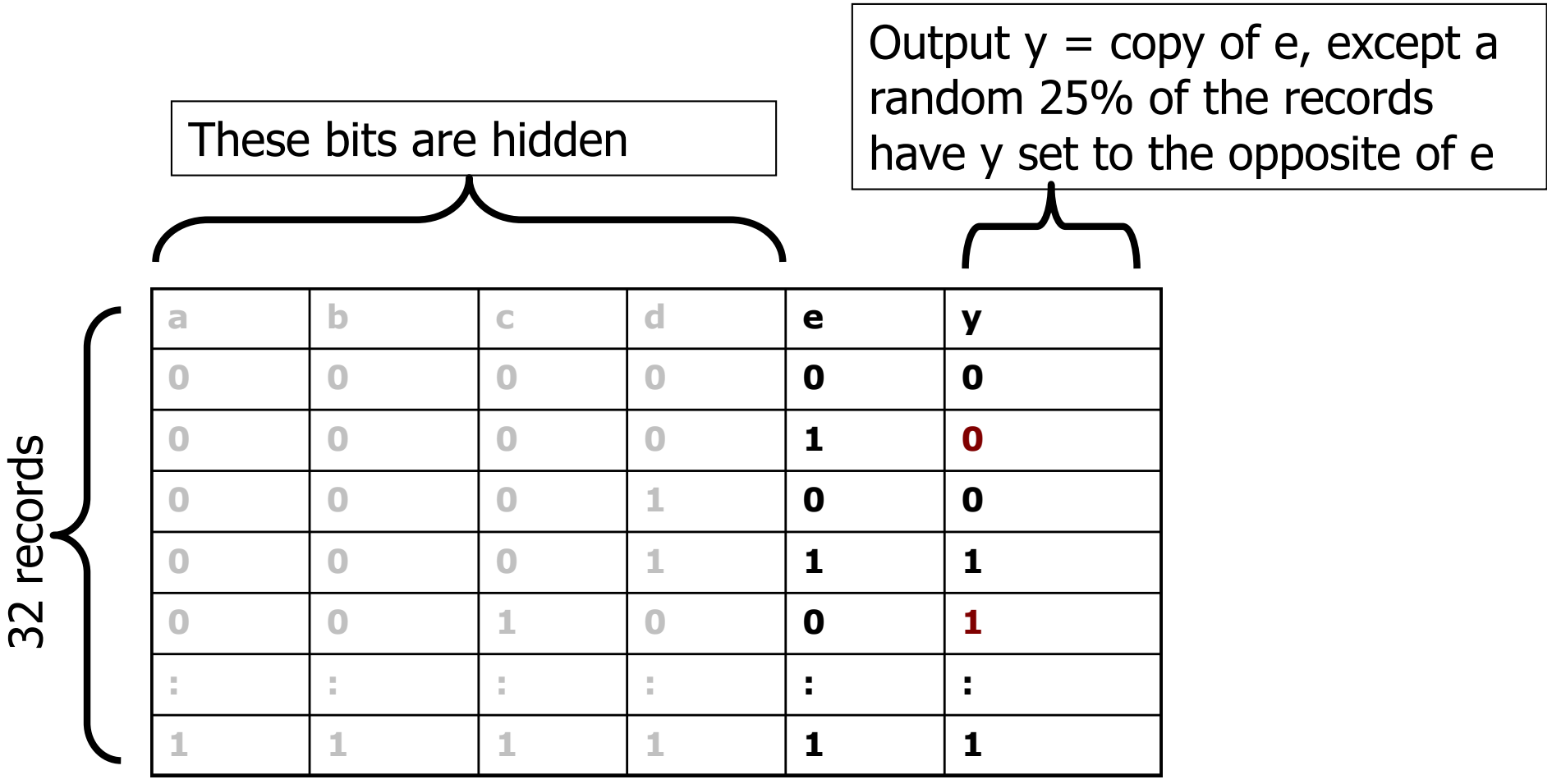
In total, we expect to be wrong on 3/8 of the test set predictions

What's this example shown us?

- This explains the discrepancy between training and test set error
- But more importantly… …it indicates there's something we should do about it if we want to predict well on future data.

Suppose we had less data

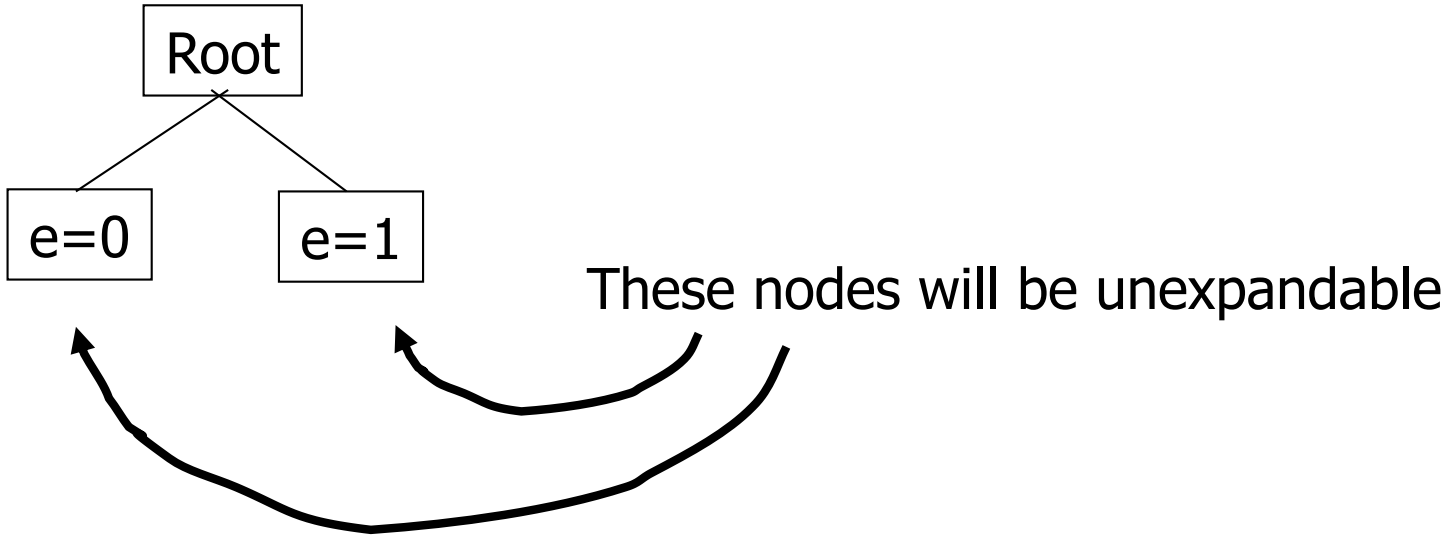
- Let's not look at the irrelevant bits



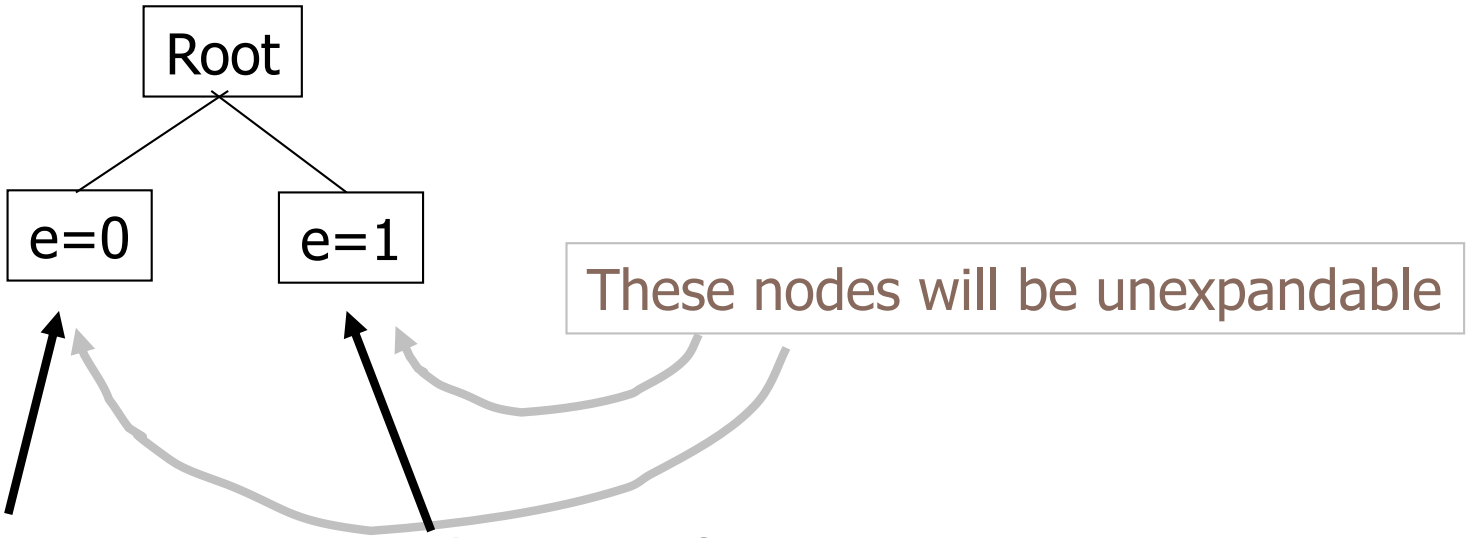
a	b	c	d	e	y
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	1
0	0	1	0	0	1
:	:	:	:	:	:
1	1	1	1	1	1

What decision tree would we learn now?

Without access to the irrelevant bits…



Without access to the irrelevant bits…



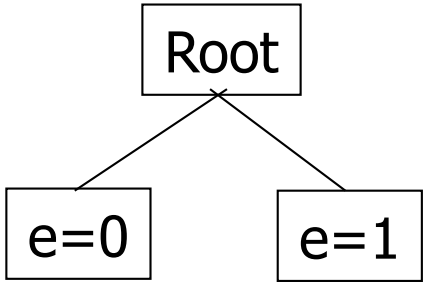
In about 12 of the 16 records in this node the output will be 0

So this will almost certainly predict 0

In about 12 of the 16 records in this node the output will be 1

So this will almost certainly predict 1

Without access to the irrelevant bits…



	almost certainly none of the tree nodes are corrupted	almost certainly all are fine
1/4 of the test set records are corrupted	n/a	1/4 of the test set will be wrongly predicted because the test record is corrupted
3/4 are fine	n/a	3/4 of the test predictions will be fine

In total, we expect to be wrong on only 1/4 of the test set predictions

Overfitting

- Definition: If your machine learning algorithm fits noise (i.e. pays attention to parts of the data that are irrelevant) it is ***overfitting***.
- Fact (theoretical and empirical): If your machine learning algorithm is overfitting then it may perform less well on test set data.

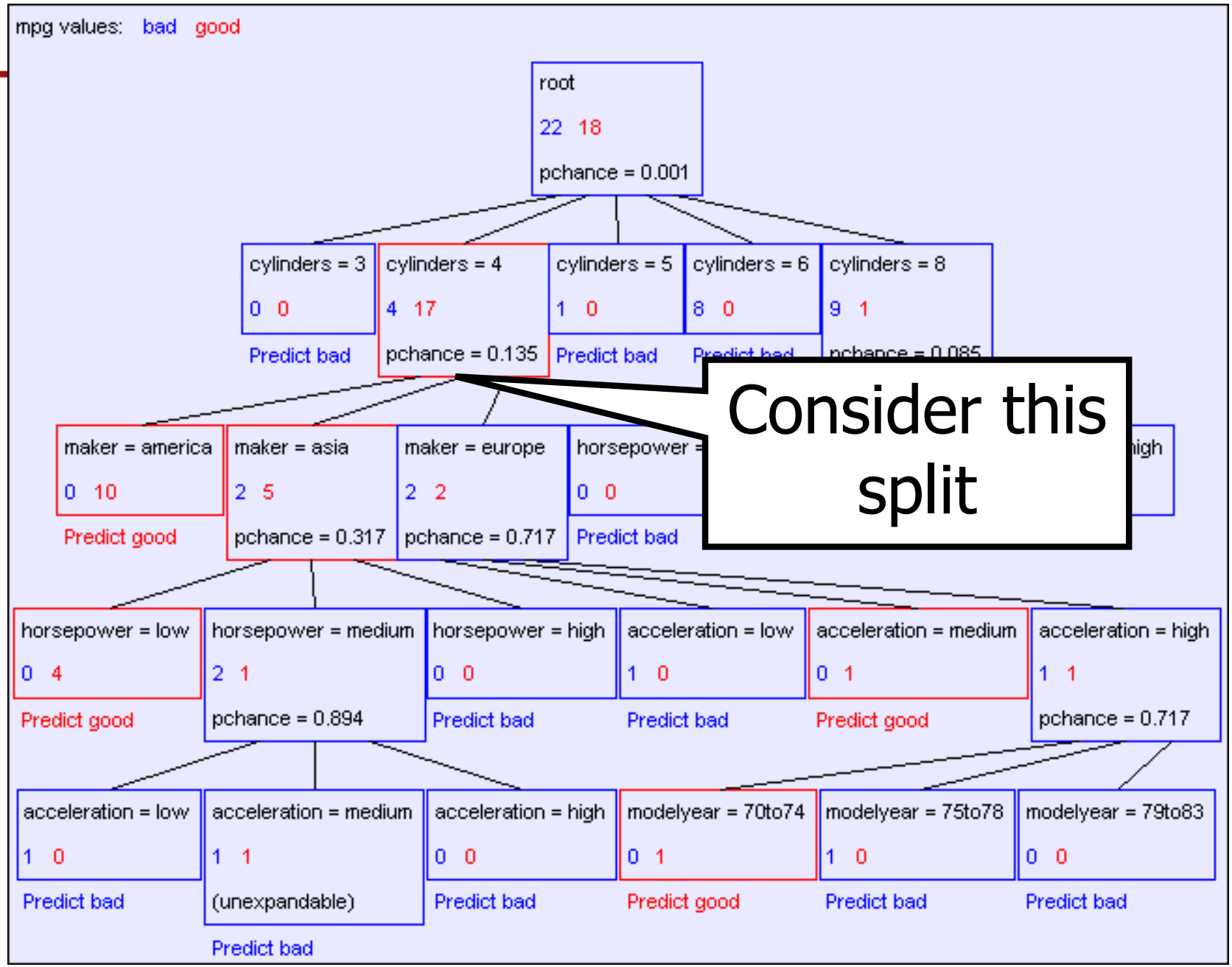
Avoiding overfitting

- Usually we do not know in advance which are the irrelevant variables
- …and it may depend on the context

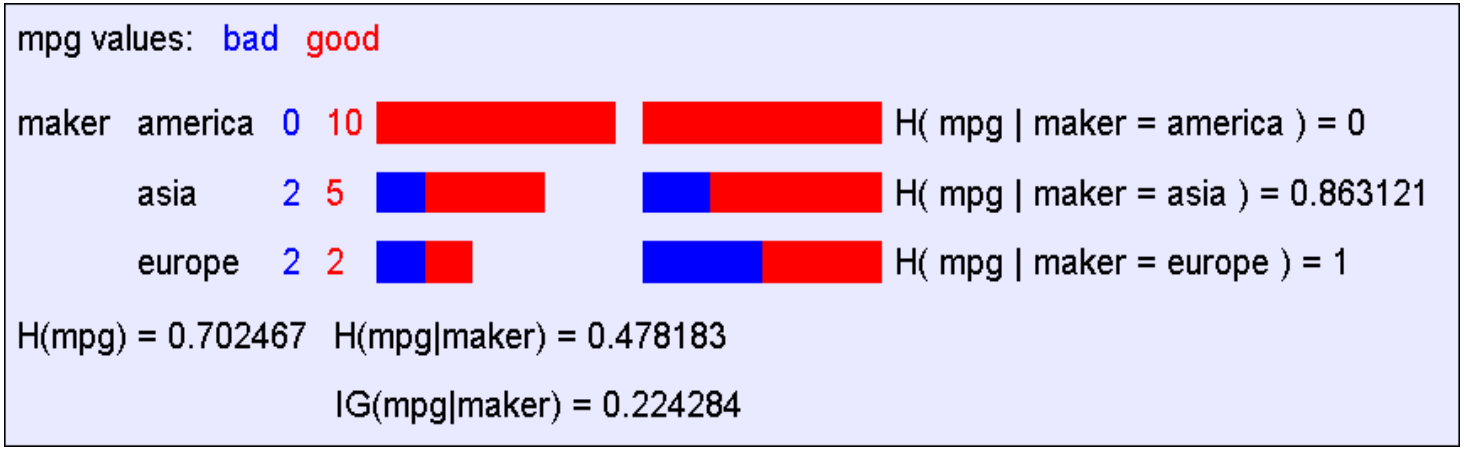
  For example, if y = a AND b then b is an irrelevant variable only in the portion of the tree in which a=0

But we can use simple statistics to warn us that we might be overfitting.

Avoiding overfitting cont.



A chi-squared test



- Suppose that mpg was completely uncorrelated with maker.
- What is the chance we'd have seen data of at least this apparent level of association anyway?
- By using a particular kind of chi-squared test, the answer is 13.5%.

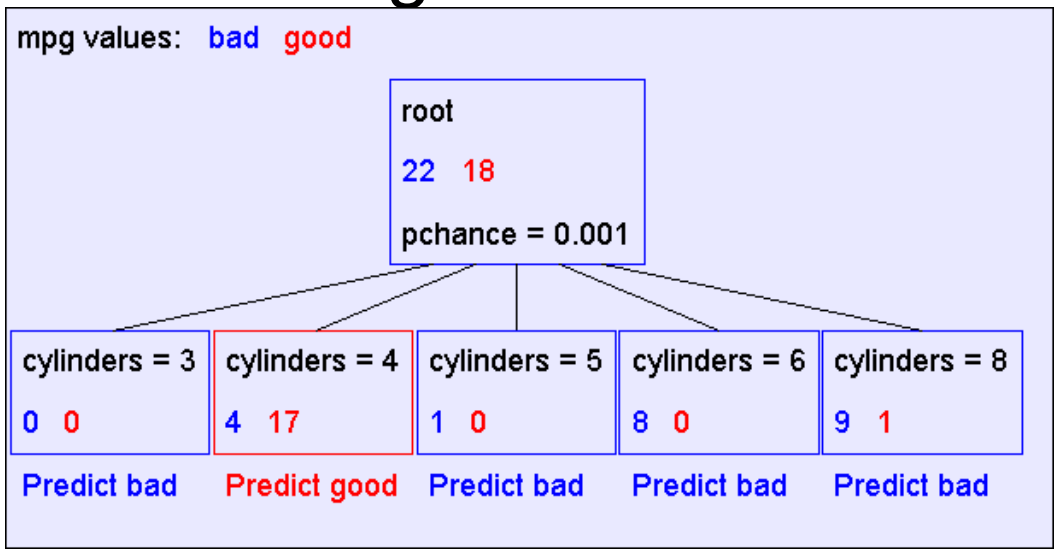
Using Chi-squared to avoid overfitting

- Build the full decision tree as before.
- But when you can grow it no more, start to prune:
  - Beginning at the bottom of the tree, delete splits in which $p_{chance} > MaxPchance$.
  - Continue working you way up until there are no more prunable nodes.

MaxPchance is a magic parameter you must specify to the decision tree, indicating your willingness to risk fitting noise.

Pruning example

- With MaxPchance = 0.1, you will see the following MPG decision tree:



Note the improved test set accuracy compared with the unpruned tree

	Num Errors	Set Size	Percent Wrong
Training Set	5	40	12.50
Test Set	56	352	15.91

MaxPchance

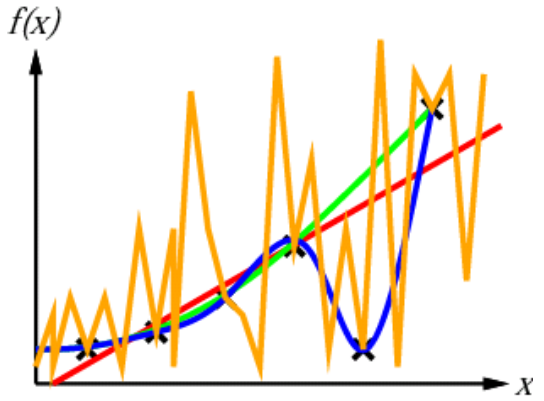
- Good news: The decision tree can automatically adjust its pruning decisions according to the amount of apparent noise and data.
- Bad news: The user must come up with a good value of MaxPchance. (Note, just use 0.05, everybody's favorite value for any magic parameter).
- Good news: But with extra work, the best MaxPchance value can be estimated automatically by a technique called cross-validation.

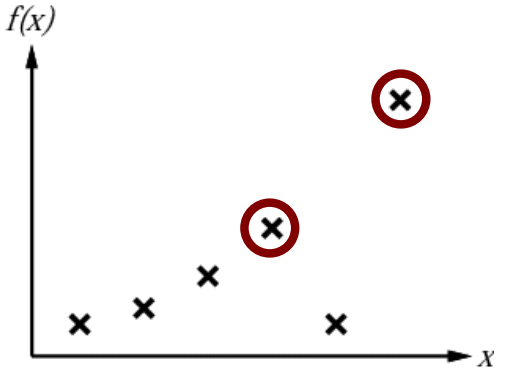
Evaluating Quality: Measures

- Classification
  - Accuracy
  - Confusion Matrix
- Problems of these measures:
  - Prior
  - Ignorance of near misses

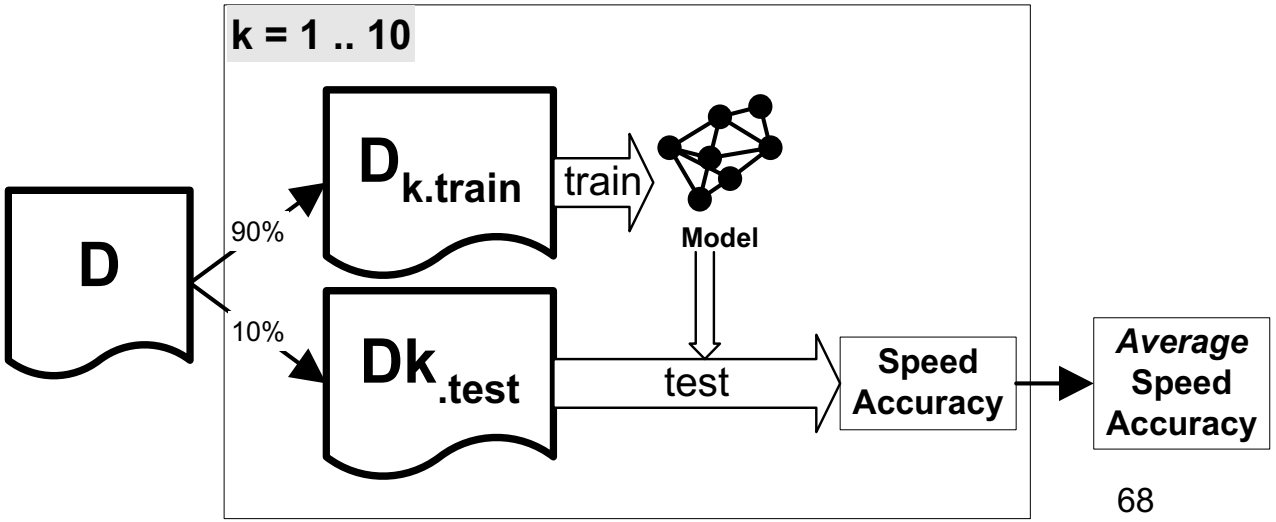
		Predicted	
		True	False
Given	True	14.1%	8.7%
	False	6.8%	70.4%
		Accuracy: 84.5	

Evaluating Quality: Cross-validation

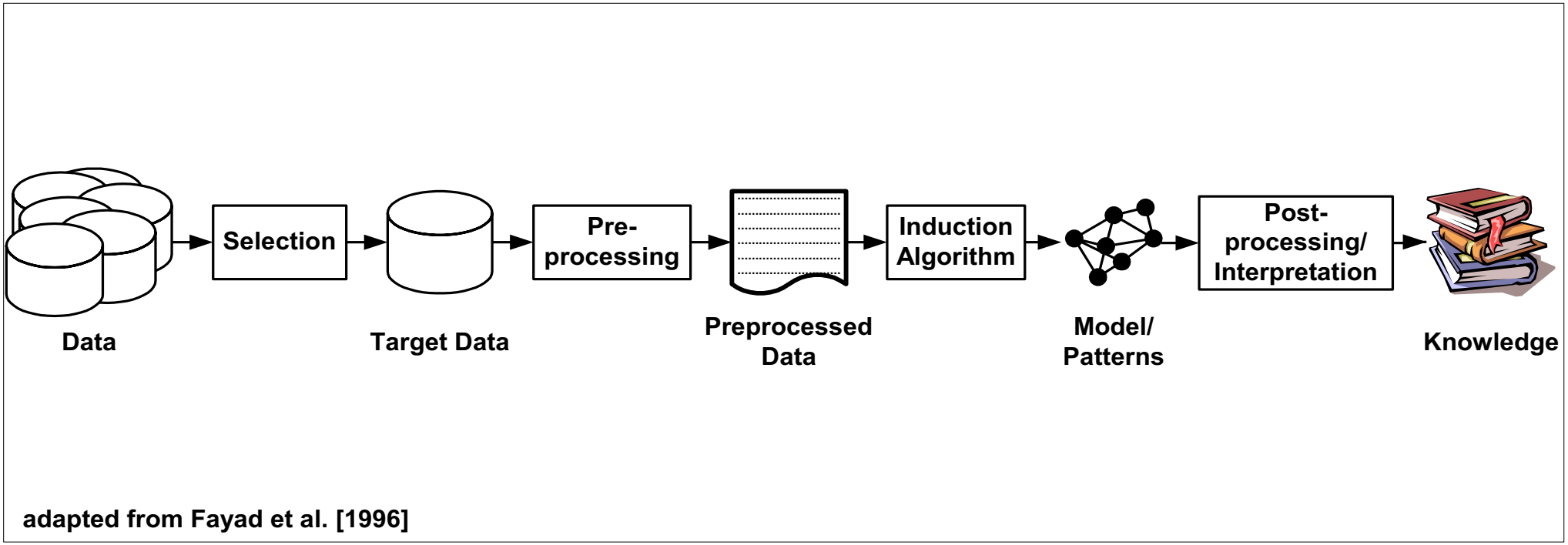




- Holdout
  - Randomly choose Test-set
- Leave-one-out
  - Learn model on all data items but one and test
- K-fold cross validation

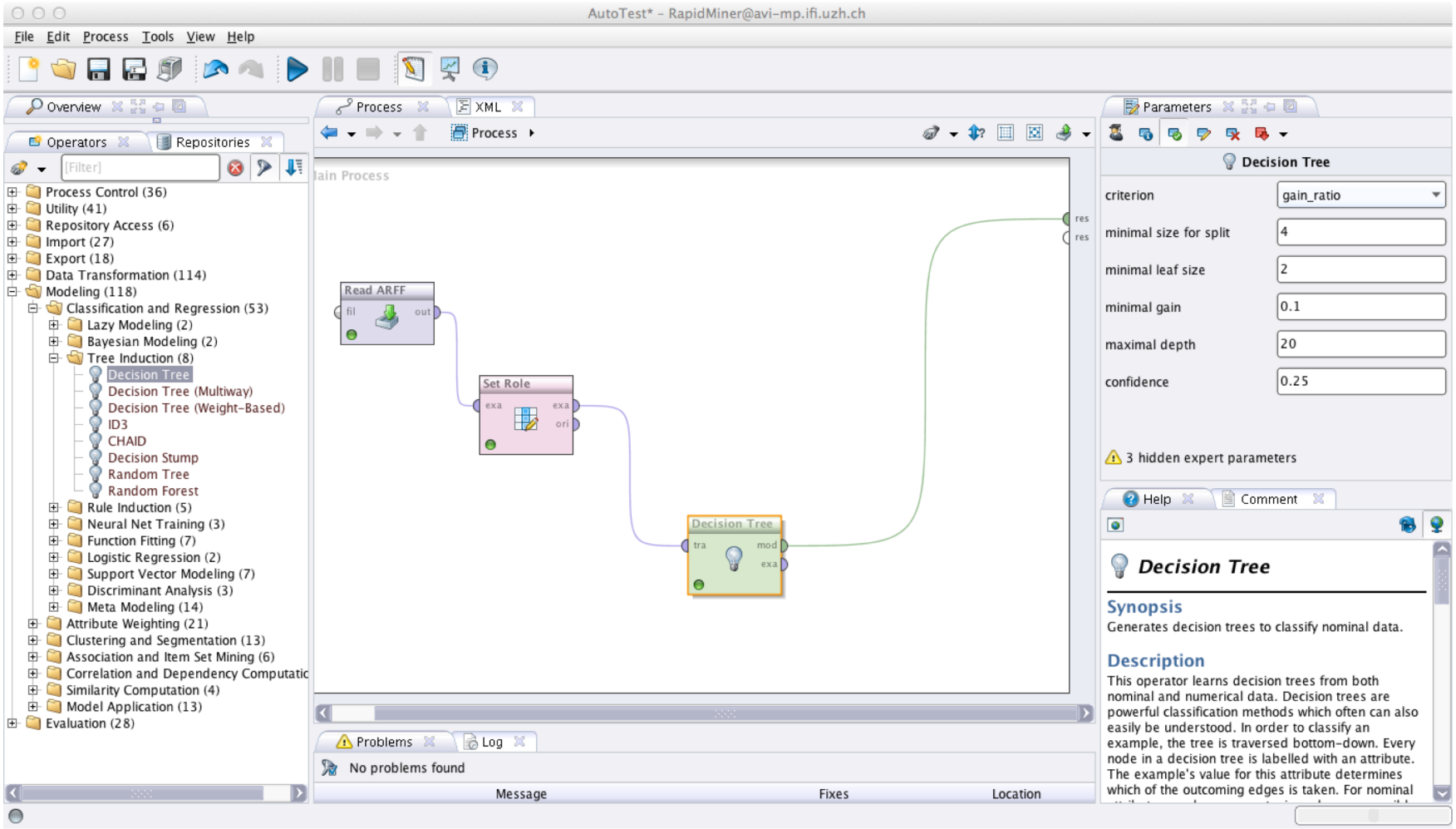


Evaluation and the Data Mining Process



- The act of Data Mining comprises a series of activities, which are repeatedly cycled through as understanding of data increases

- An open source Mining Toolkit: http://rapid-i.com/



Agenda

- Machine Learning/Data Mining Basics
  - Why do we need mining or learning?
  - Learning as induction
  - Data Mining in a slide
- Six Data Mining Tasks and their Evaluation
  - Classification - Learning Decision Trees
  - **Class Probability Estimation – Naïve Bayes**
  - Regression – Linear Regression & Neural Networks
  - Clustering – K-Means & Hierarchical Clustering
  - Association Rules – Apriori

Prediction:
Class Probability Estimation

Getting Probabilities

Universität Zürich[UZH]

DIS Dynamic and Distributed Information Systems

Using the Joint

$$Y^{predict} = \underset{v}{\operatorname{argmax}} \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)}{P(Y = v)}$$

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

Once you have the JD you can ask for the probability of any logical expression involving your attribute

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

$$Y^{predict} = \underset{v}{\operatorname{argmax}} \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)}{P(Y = v)}$$

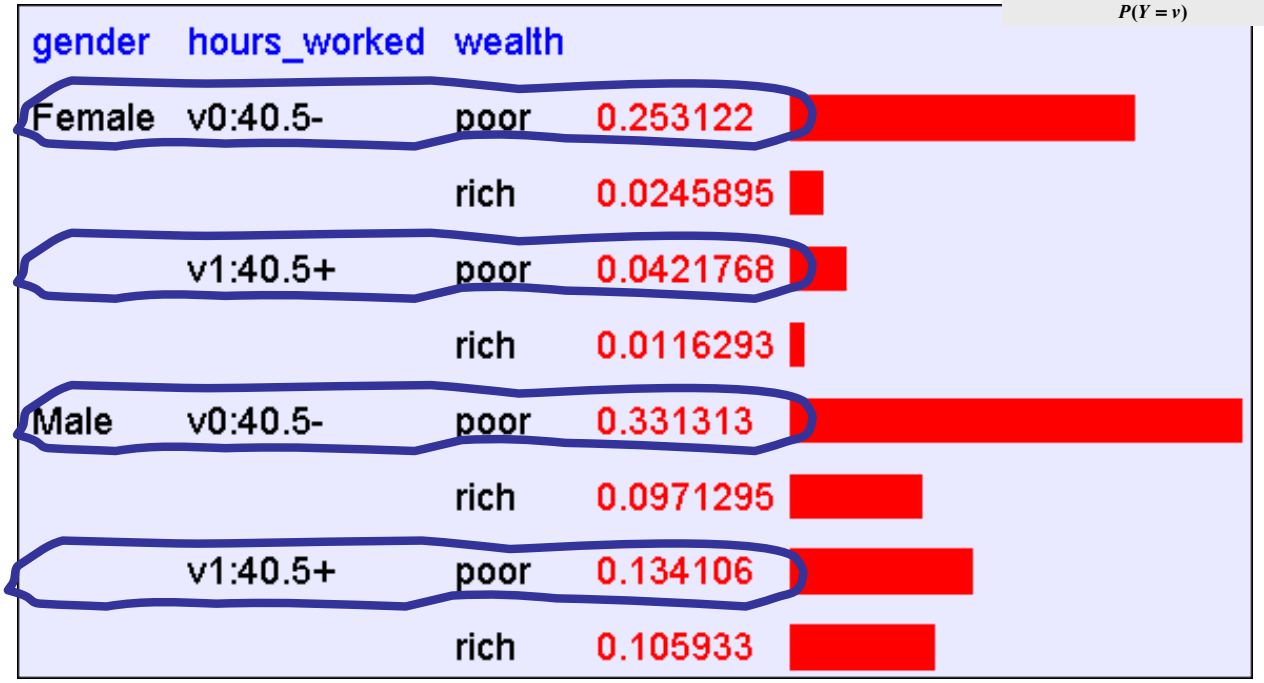
gender	hours_worked	wealth		
Female	v0:40.5-	poor	0.253122	
		rich	0.0245895	
	v1:40.5+	poor	0.0421768	
		rich	0.0116293	
Male	v0:40.5-	poor	0.331313	
		rich	0.0971295	
	v1:40.5+	poor	0.134106	
		rich	0.105933	

P(Poor Male) = 0.4654

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

$$Y^{predict} = \underset{v}{\operatorname{argmax}} \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)}{P(Y = v)}$$



P(Poor) = 0.7604

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Inference with the Joint

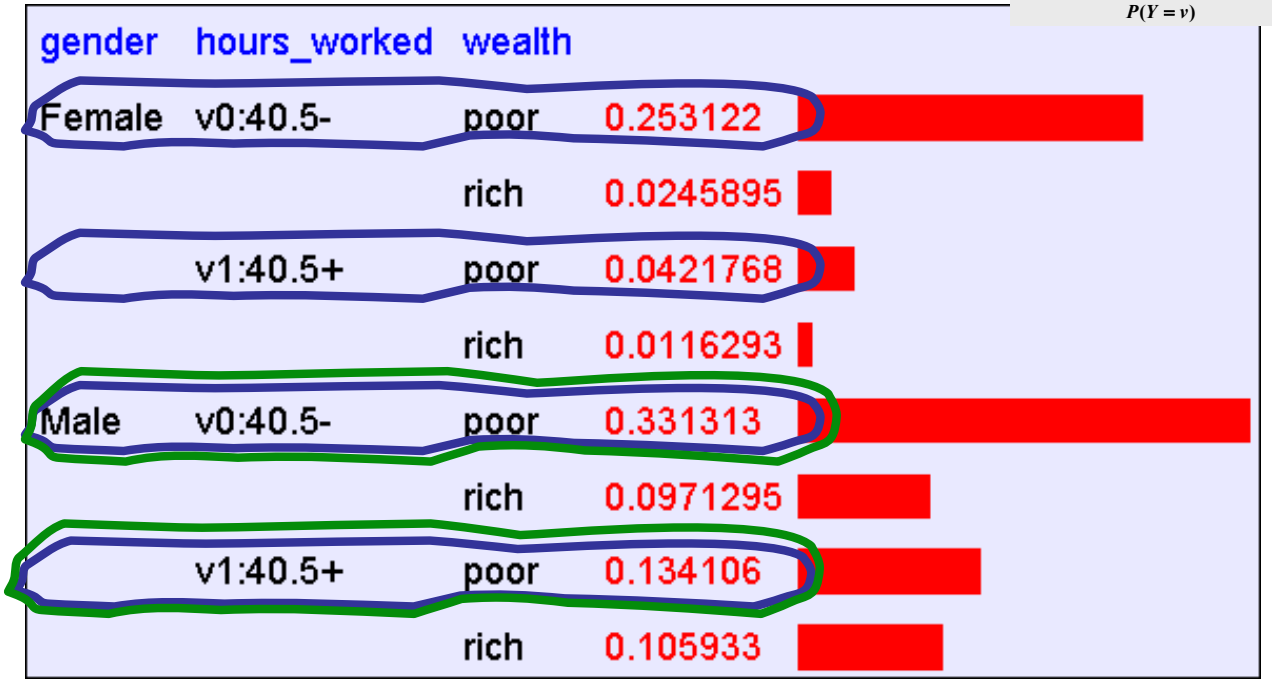
$$Y^{predict} = \operatorname*{argmax}_v P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)\, P(Y = v)$$

gender	hours_worked	wealth		
Female	v0:40.5-	poor	0.253122	
		rich	0.0245895	
	v1:40.5+	poor	0.0421768	
		rich	0.0116293	
Male	v0:40.5-	poor	0.331313	
		rich	0.0971295	
	v1:40.5+	poor	0.134106	
		rich	0.105933	

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

Inference with the Joint

$$Y^{predict} = \underset{v}{\operatorname{argmax}} \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)}{P(Y = v)}$$



$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

P(Male | Poor) = 0.4654 / 0.7604 = 0.612

$$Y^{predict} = \operatorname{argmax}_{v} P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)\, P(Y = v)$$

Naïve Density Estimation

The problem with the Joint Estimator is that it just mirrors the training data.

We need something which generalizes more usefully.

The naïve model generalizes strongly:

Assume that each attribute is distributed independently of any of the other attributes.

i.e., **P(A|B) = P(A)**

Learning a Naïve Density Estimator

$$Y^{predict} = \underset{v}{\operatorname{argmax}} \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)}{P(Y = v)}$$

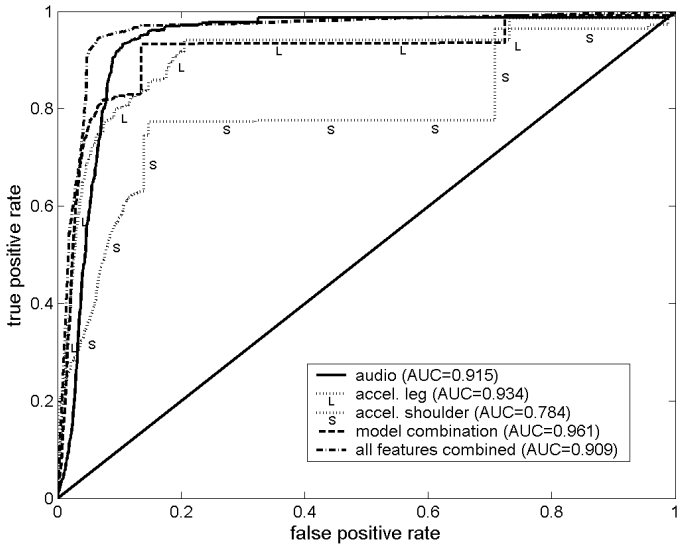
$$\hat{P}(x[i] = u \mid Cause) = \frac{\#\text{records in which } x[i] = u \mid Cause}{\text{total number of records} \mid Cause}$$

Another trivial learning algorithm!

Evaluating Quality: Measures

- Class Probability Estimation
  - Accuracy
  - Rank Accuracy
  - Confusion Matrix
  - ROC Curves
    (Provost & Fawcett)

		Predicted	
		True	False
Given	True	14.1%	8.7%
	False	6.8%	70.4%
		Accuracy: 84.5	



Agenda

- Machine Learning/Data Mining Basics
  - Why do we need mining or learning?
  - Learning as induction
  - Data Mining in a slide
- Six Data Mining Tasks and their Evaluation
  - Classification - Learning Decision Trees
  - Class Probability Estimation – Naïve Bayes
  - **Regression – Linear Regression & Neural Networks**
  - Clustering – K-Means & Hierarchical Clustering
  - Association Rules – Apriori
- Practical Considerations (Missing Values, …)
- Project A → Z

Prediction: Regression

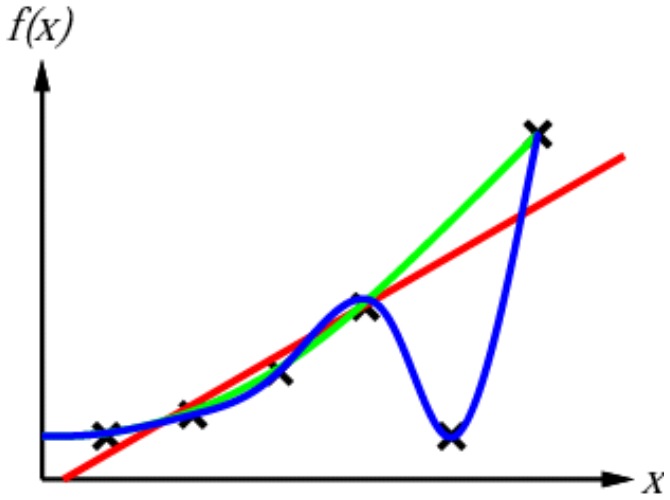
Predicting Continuous Values

Universität Zürich[UZH]

DIS Dynamic and Distributed Information Systems

Traditional Regression

- The regression equation deals with the following variables:
  - The *unknown parameters* denoted as β.
  - The *independent variables*, X
  - The *dependent variable*, Y.
- Equation: $Y = f(X, \beta)$
- Goal:
  - Find β_i (for all i) that minimize the error



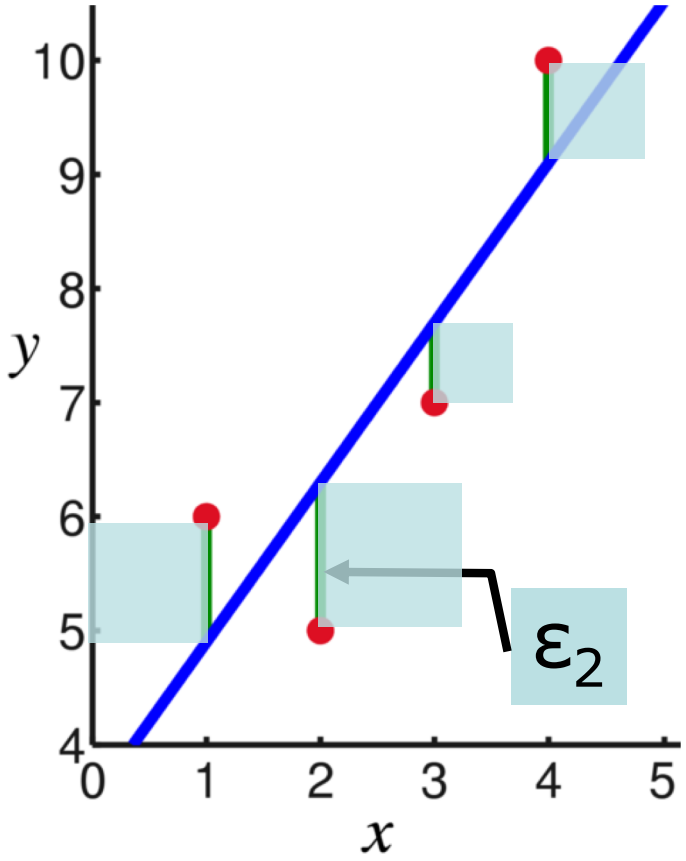
Linear Regression

- Learn β_i for the following formula

$$y_i = \beta_0 + \beta_1 x_{1,i} + \ldots + \beta_n x_{n,i} + \varepsilon_i$$

- While minimizing

$$\sum_{i=1}^{N} \varepsilon_i^2$$



Agenda

- Machine Learning/Data Mining Basics
  - Why do we need mining or learning?
  - Learning as induction
  - Data Mining in a slide
- Six Data Mining Tasks and their Evaluation
  - Classification - Learning Decision Trees
  - Class Probability Estimation – Naïve Bayes
  - Regression – Linear Regression & Neural Networks
  - **Clustering – K-Means & Hierarchical Clustering**
  - Association Rules – Apriori

Clustering

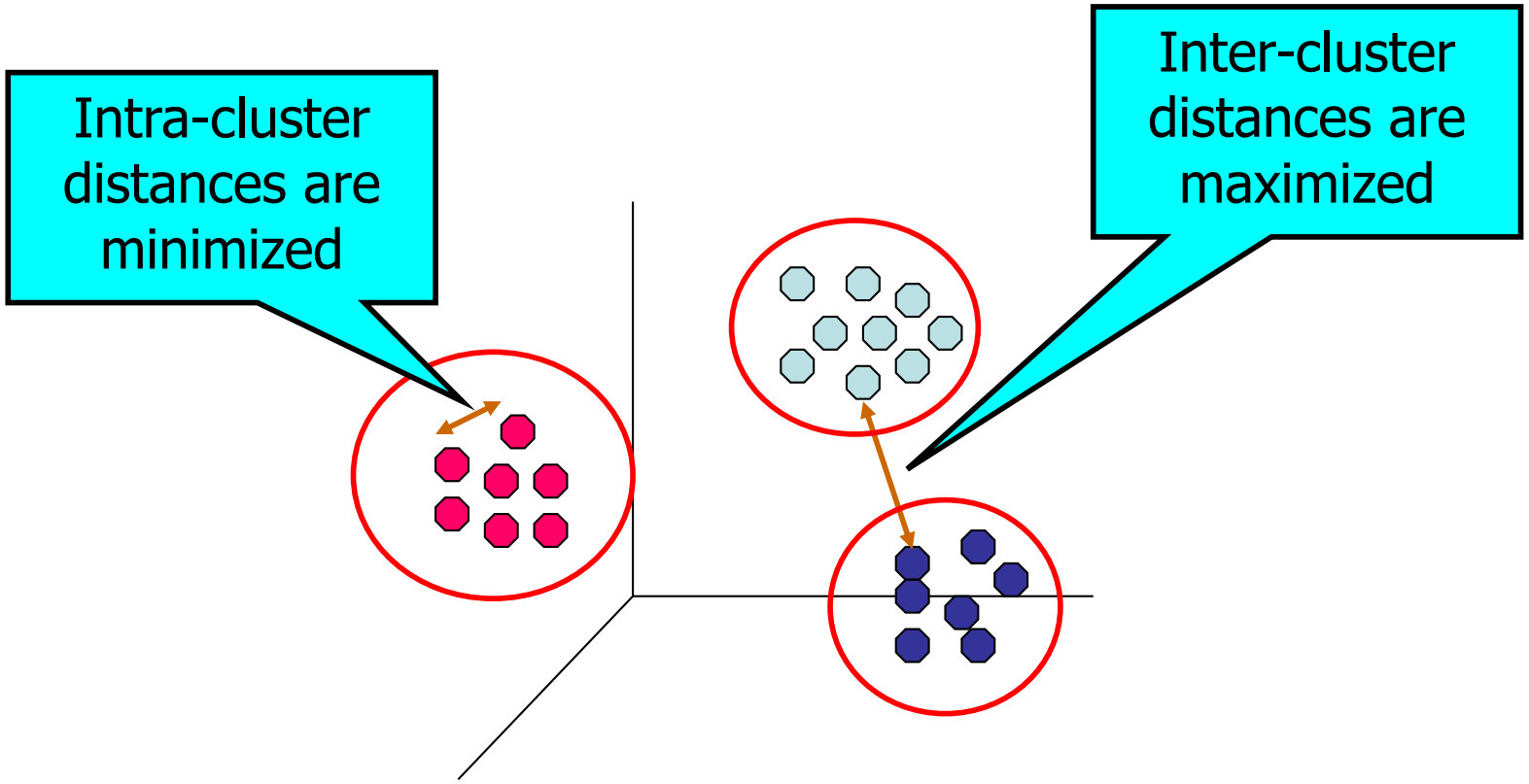
Grouping Entities

Universität Zürich[UZH]

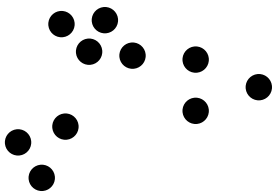
DIS Dynamic and Distributed Information Systems

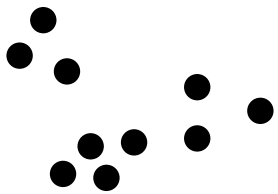
What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

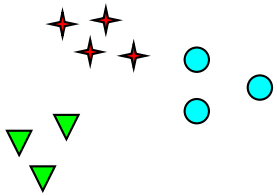


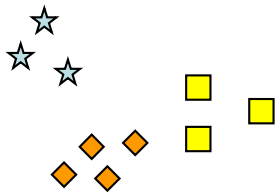
Notion of a Cluster can be Ambiguous



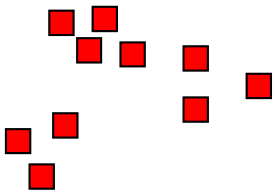


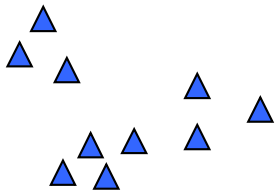
How many clusters?



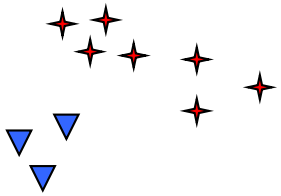


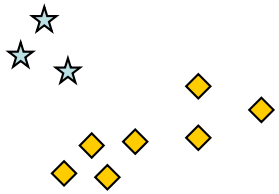
Six Clusters





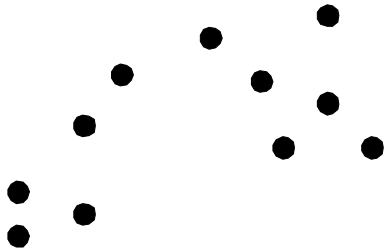
Two Clusters

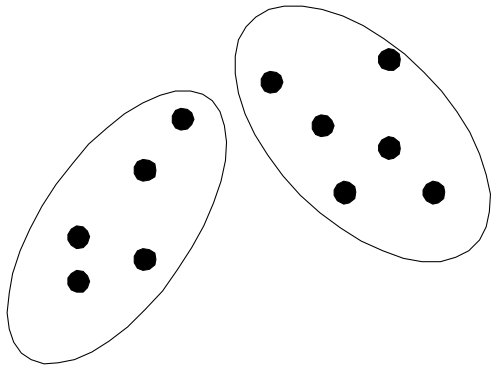


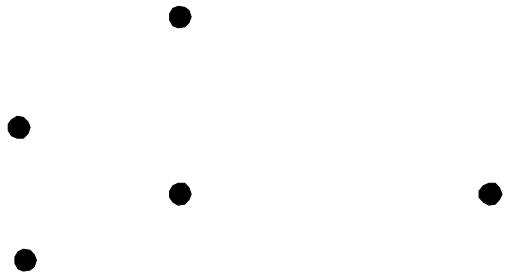


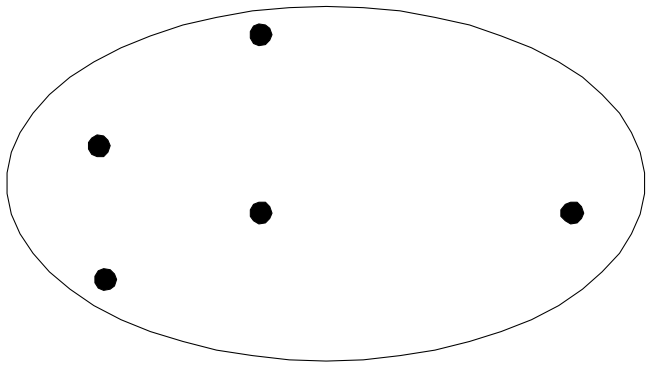
Four Clusters

Partitional Clustering





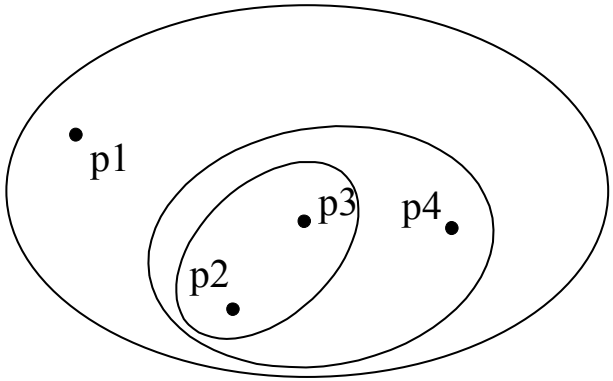




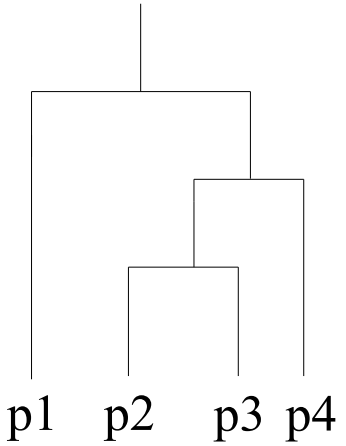
Original Points

A Partitional Clustering

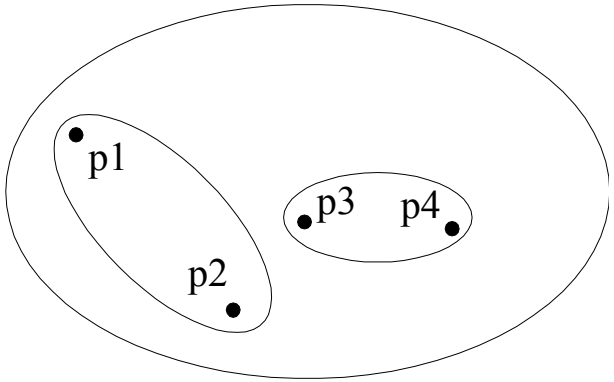
Hierarchical Clustering



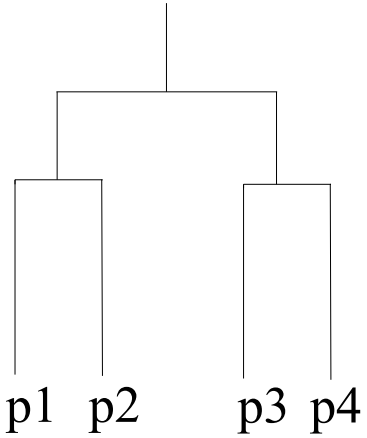
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



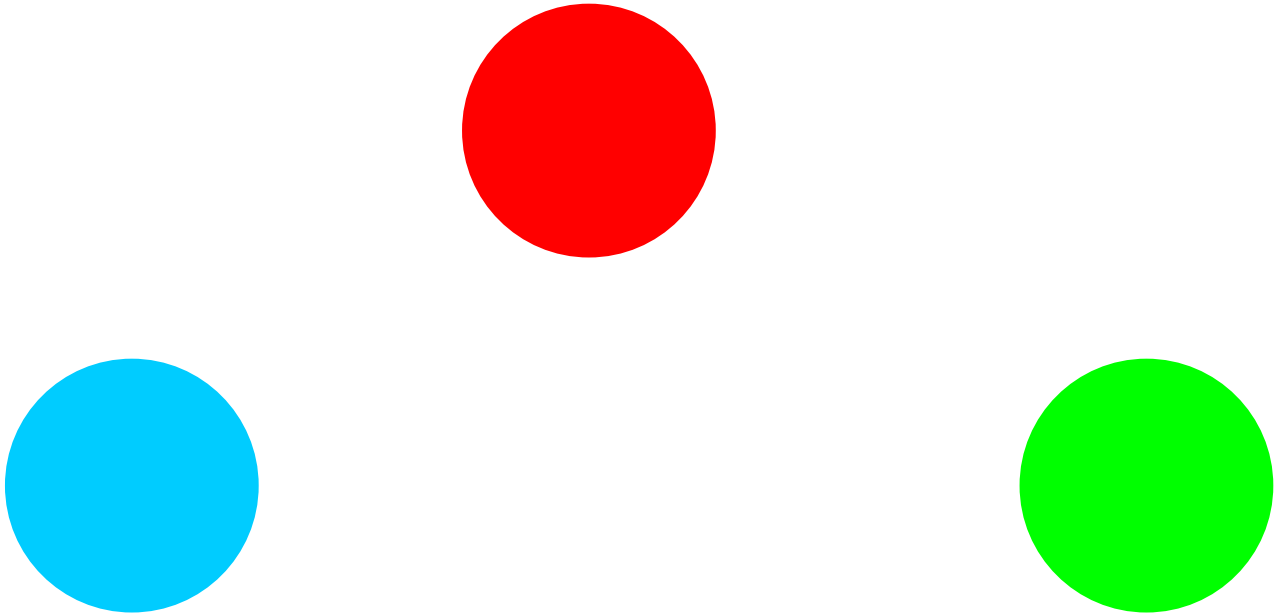
Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster of widely different sizes, shapes, and densities

Types of Clusters: Well-Separated

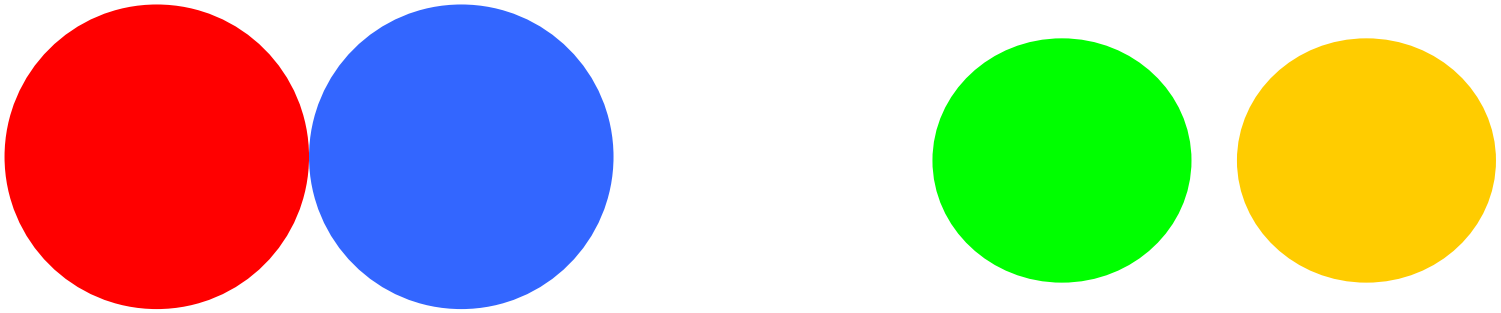
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Center-Based

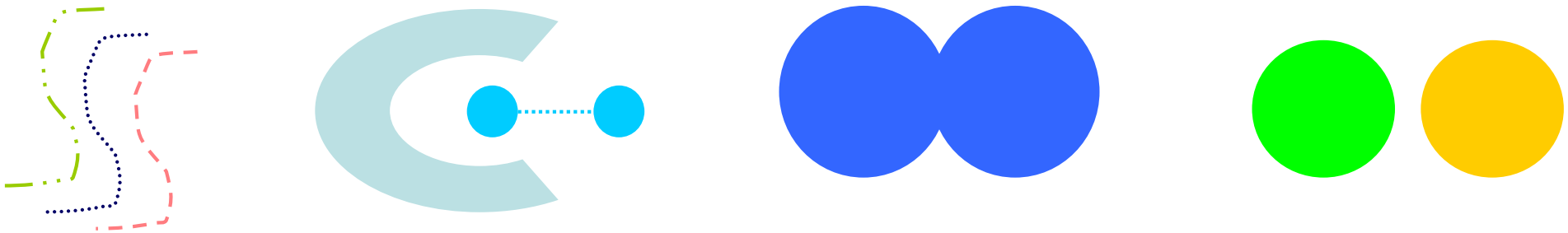
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most “representative” point of a cluster



4 center-based clusters

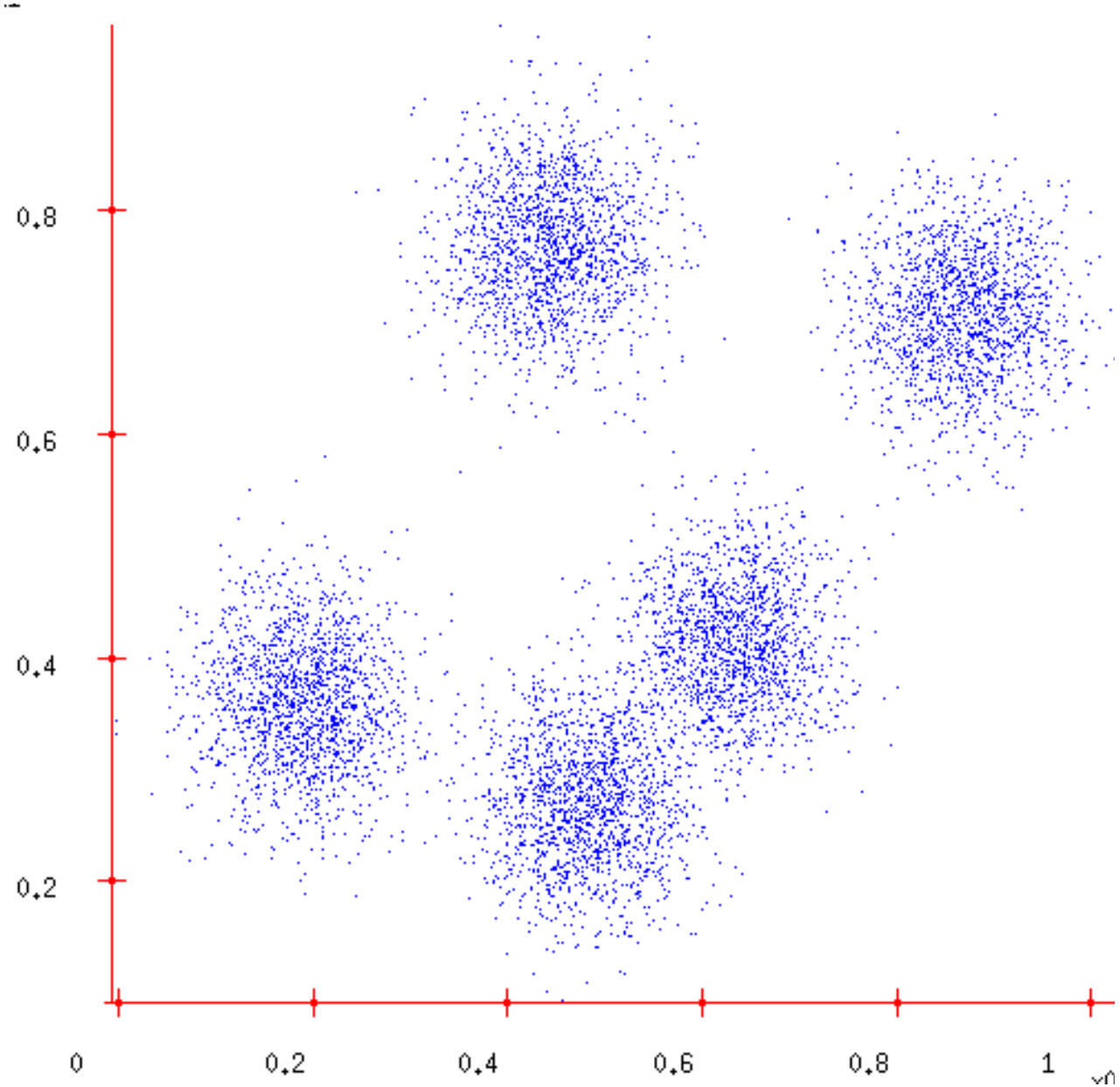
Types of Clusters: Contiguity-Based

- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



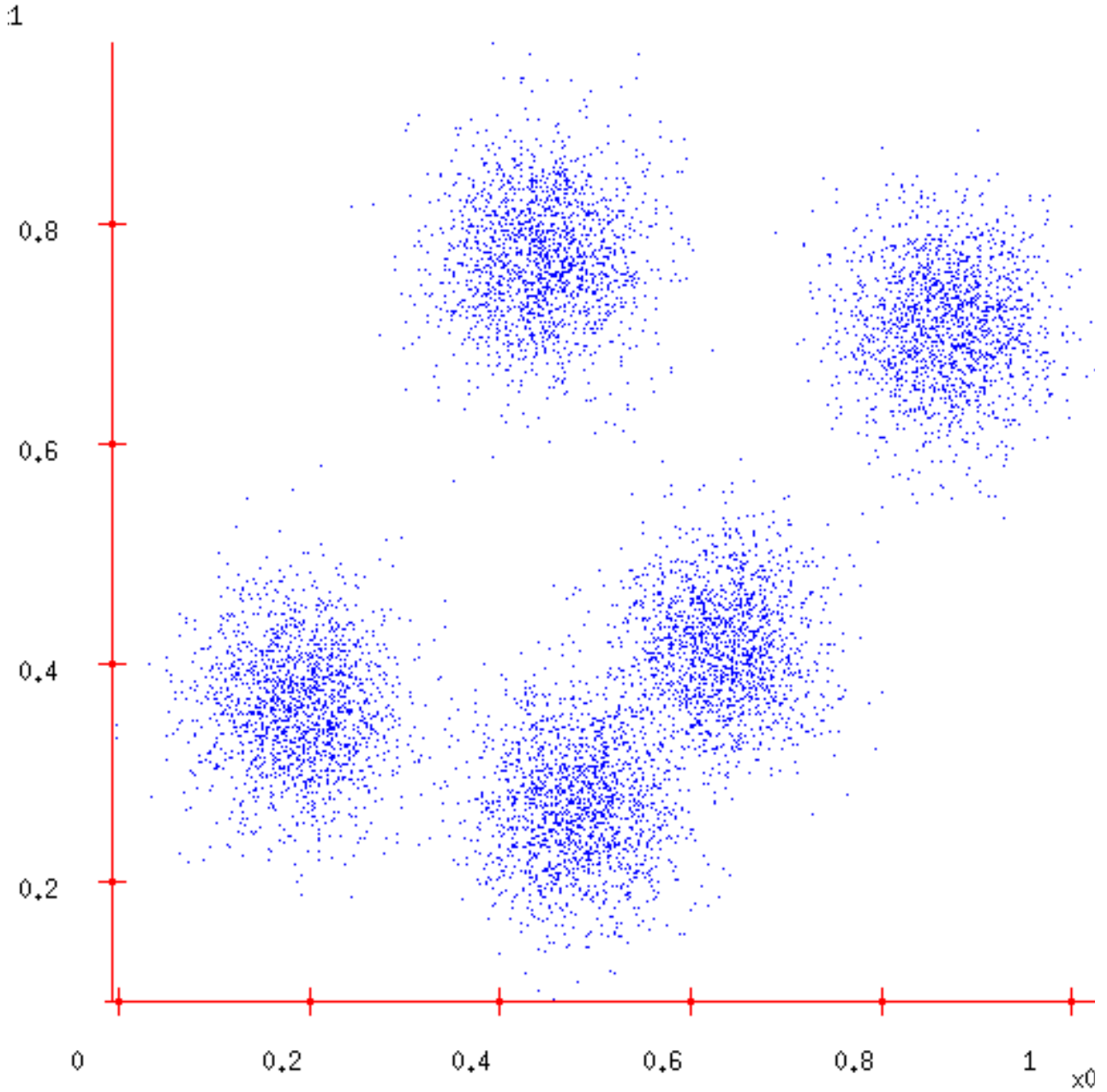
8 contiguous clusters

Clustering
K-means



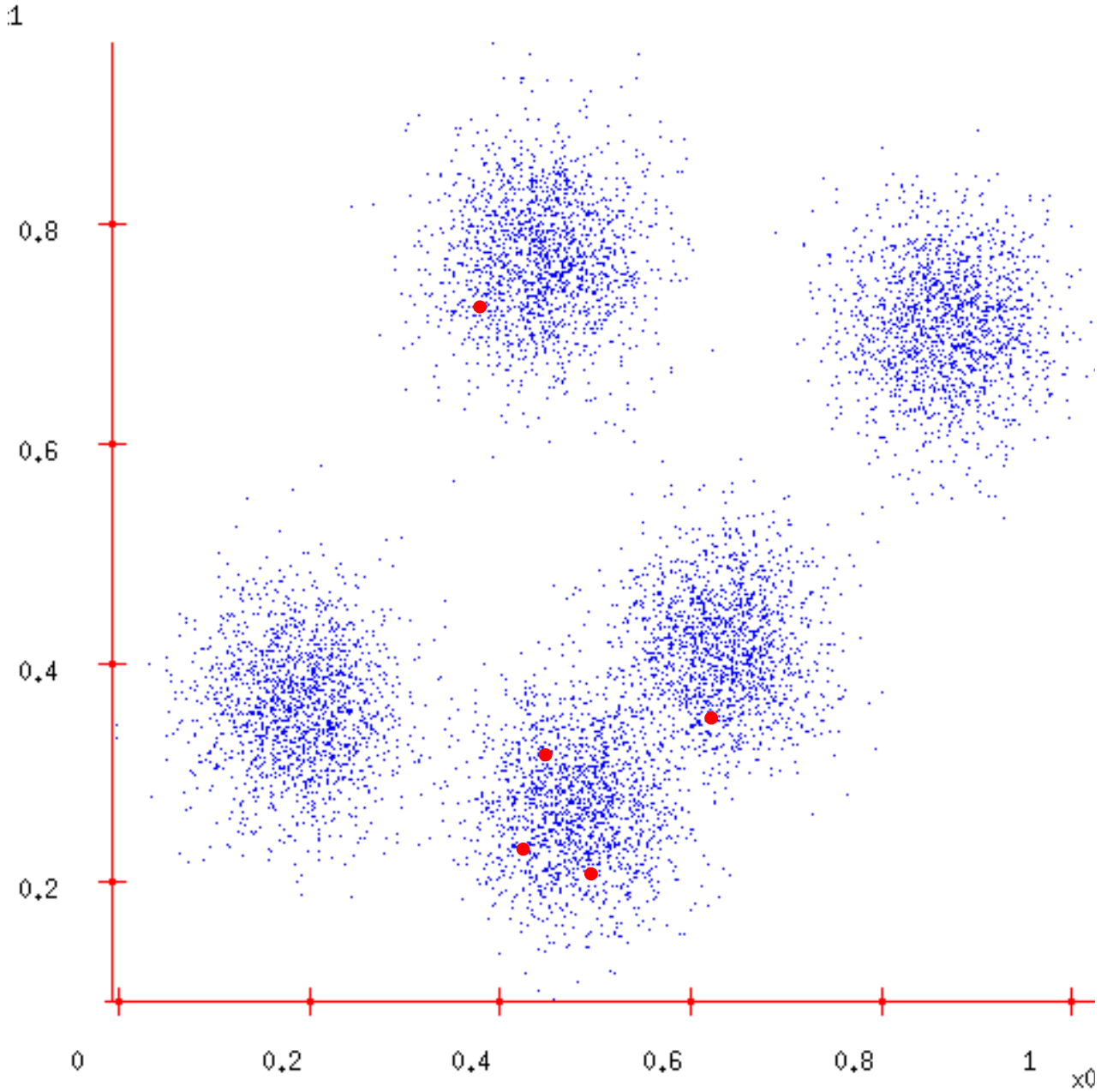
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2.
3.
4.
5.
6.



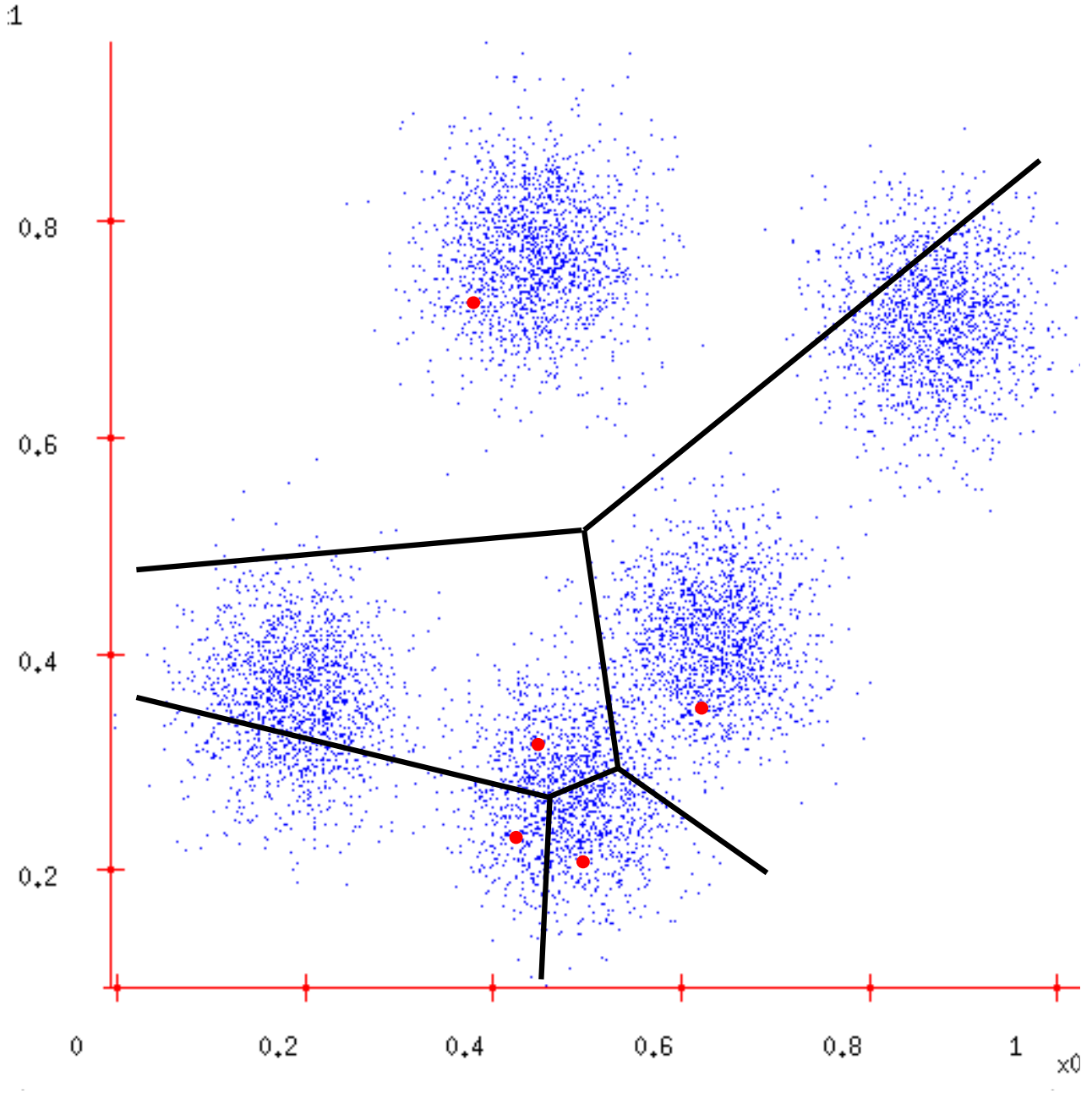
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3.
4.
5.
6.



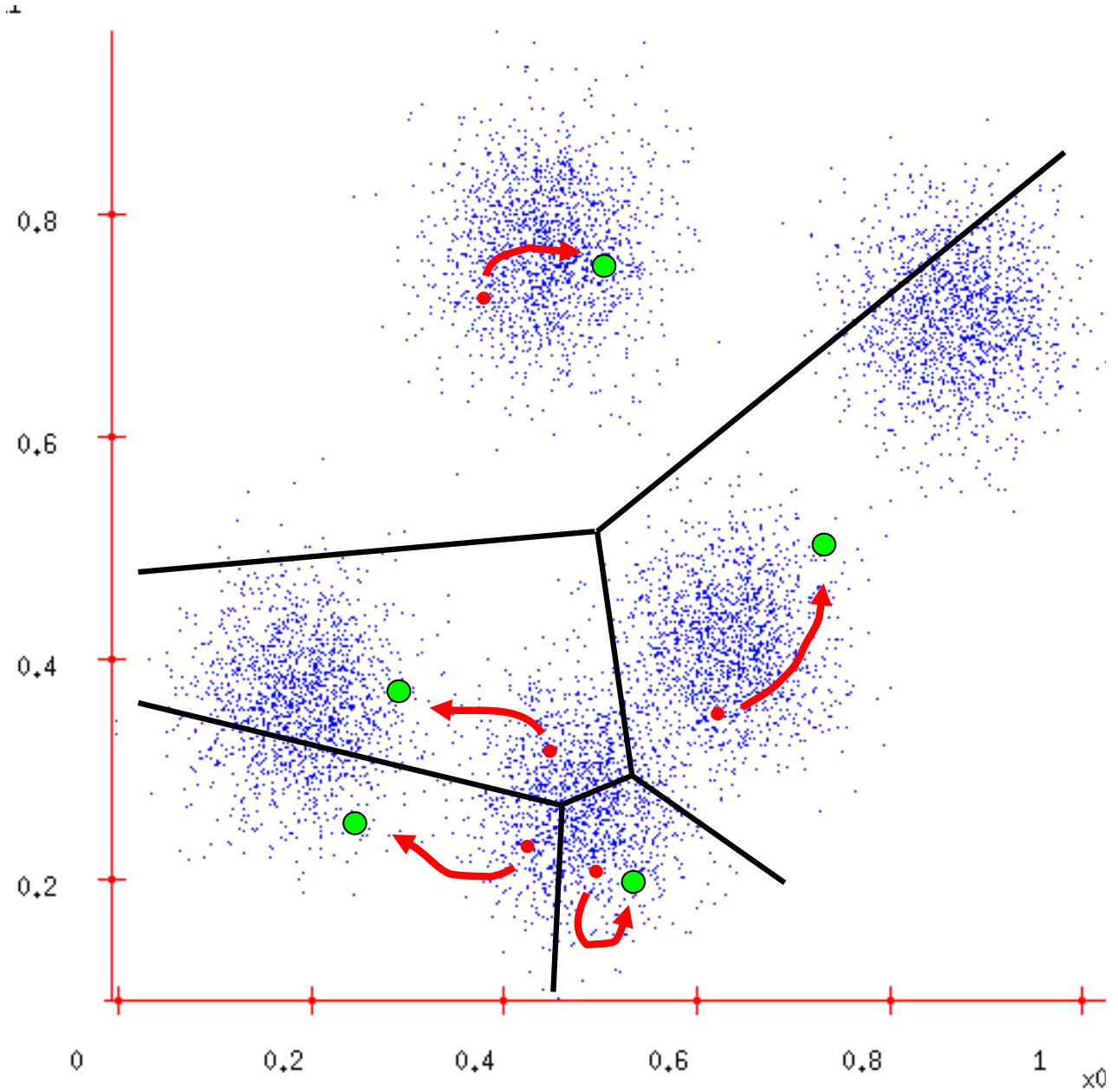
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4.
5.
6.



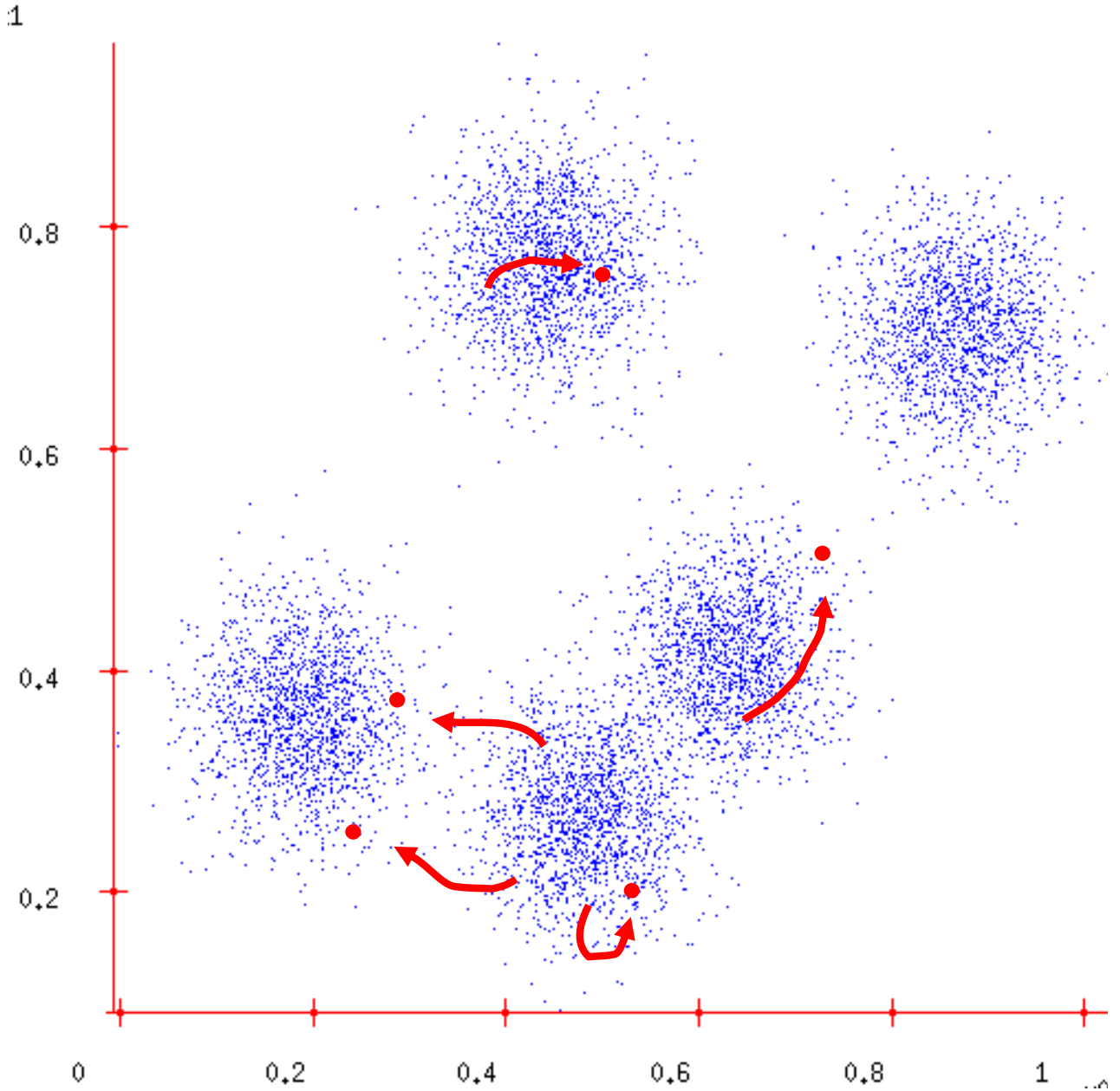
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5.
6.

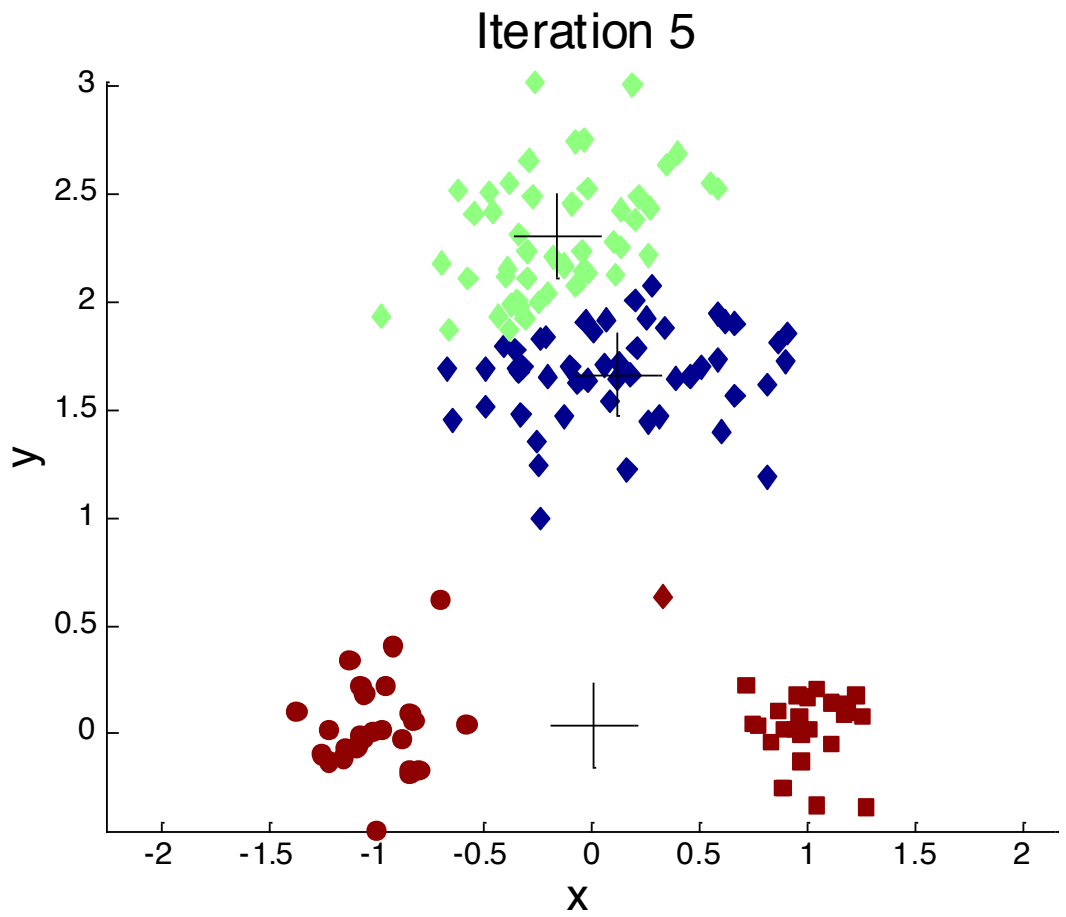


K-means

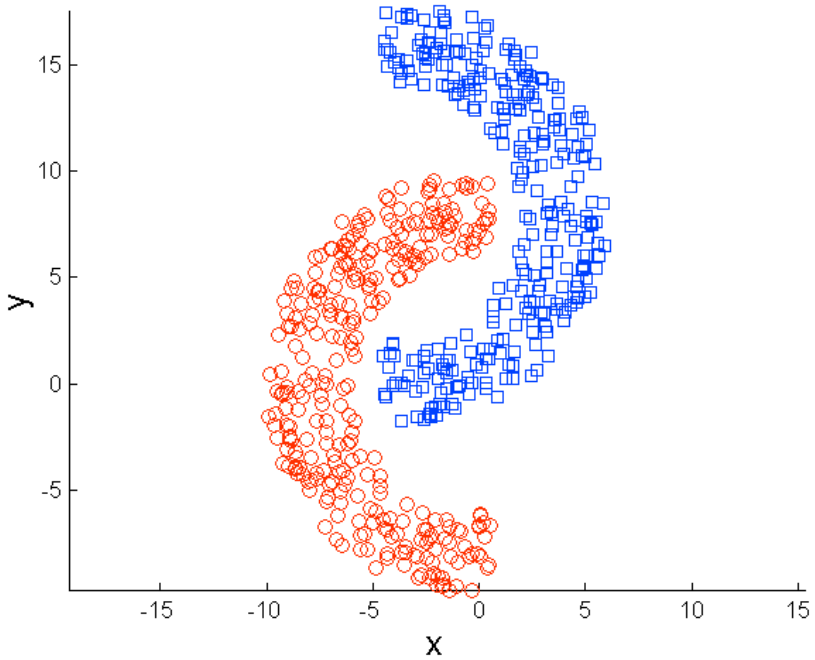
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



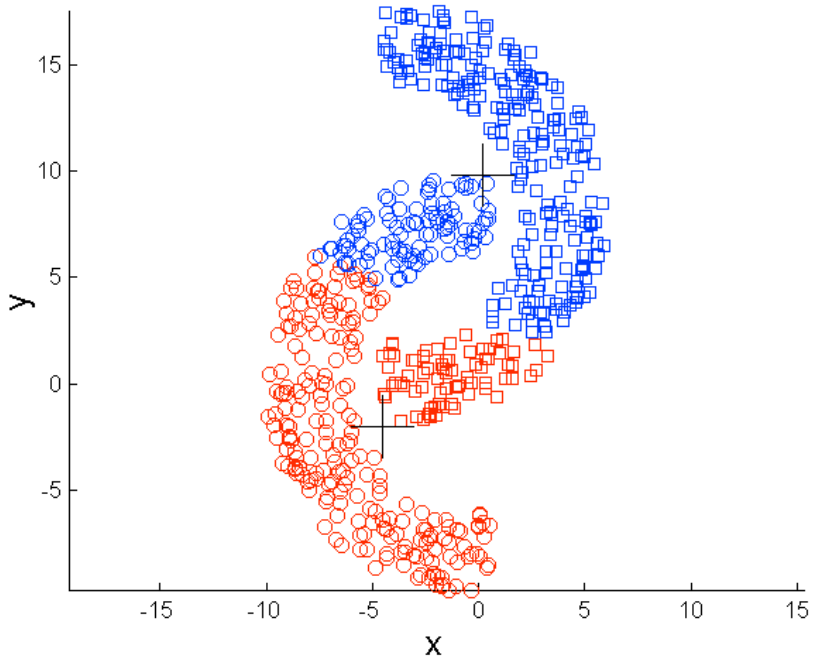
Importance of Choosing Initial Centroids …



Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

Agenda

- Machine Learning/Data Mining Basics
  - Why do we need mining or learning?
  - Learning as induction
  - Data Mining in a slide
- Six Data Mining Tasks and their Evaluation
  - Classification - Learning Decision Trees
  - Class Probability Estimation – Naïve Bayes
  - Regression – Linear Regression & Neural Networks
  - Clustering – K-Means & Hierarchical Clustering
  - **Association Rules – Apriori**

Association Rule Mining

Finding Dependencies

Universität Zürich[UZH]

DIS Dynamic and Distributed Information Systems

Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

Market-Basket transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

{Diaper} → {Beer},
{Milk, Bread} → {Eggs,Coke},
{Beer, Bread} → {Milk},

Implication means co-occurrence, not causality!

Definition: Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Bread, Milk, Diaper}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Bread, Milk, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Definition: Association Rule

- Association Rule
  - An implication expression of the form X → Y, where X and Y are itemsets
  - Example:
    {Milk, Diaper} → {Beer}

- Rule Evaluation Metrics
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

What we did not talk about…

- Pre-processing (e.g., Missing Values)
- Post-processing (e.g., Cost-sensitive learning)
- Reinforcement Learning
- Embedded Data Mining
- Relational Data Mining
- Additional Learning Algorithms
  - Support Vector Machines, Bayesian Networks, …

Further Sources/References

- References
  - Dhar, V. & Stein, R. (1997). *Intelligent Decision Support Methods.* Prentice Hall. Upper Saddle River
  - Witten, I. H., & Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufman Publishers. (much improved 3nd edition available)
  - Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw-Hill.
  - Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York: Wiley.
- Software (free)
  - Weka (java based DM toolkit with GUI) http://www.cs.waikato.ac.nz/ml/weka/
  - Rapid Miner http://rapid-i.com