

# Data Engineering

Data preprocessing and transformation

# JUST APPLY A LEARNER? NO!

---

- Algorithms are *biased*
  - No free lunch theorem: considering all possible data distributions, no algorithm is better than another
- Algorithms make *assumptions* about data
  - Conditionally independent features (naive Bayes)
  - All features relevant (e.g., kNN, C4.5)
  - All features discrete (e.g., 1R)
  - Little/no noise (many regression algorithms)
  - Little/no missing values (e.g., PCA)
- Given data:
  - Choose/adapt algorithm to data (selection/parameter tuning)
  - Adapt data to algorithm (data engineering)

# DATA ENGINEERING

---

- **Attribute selection (feature selection)**
  - Remove features with little/no predictive information
- **Attribute discretization**
  - Convert numerical attributes to nominal ones
- **Data transformations (feature generation)**
  - Transform data to another representation
- **Dirty data**
  - Remove missing values or outliers

# IRRELEVANT FEATURES CAN 'CONFUSE' ALGORITHMS

---

- kNN: curse of dimensionality
  - # training instances required increases exponentially with # (irrelevant) attributes
  - Distance between neighbors increases with every new dimension
- C4.5: data fragmentation problem
  - select attributes on less and less data after every split
  - Even random attributes can look good on small samples
  - Partially corrected by pruning
- Naive Bayes: redundant (very similar) features
  - Features clearly not independent, probabilities likely incorrect
  - But, Naive Bayes is insensitive to irrelevant features (ignored)

# ATTRIBUTE SELECTION

---

- Other benefits
  - Speed: irrelevant attributes often slow down algorithms
  - Interpretability: e.g. avoids huge decision trees
- 2 types:
  - Feature Ranking: rank by relevancy metric, cut off
  - Feature Selection: search for optimal subset

# ATTRIBUTE SELECTION

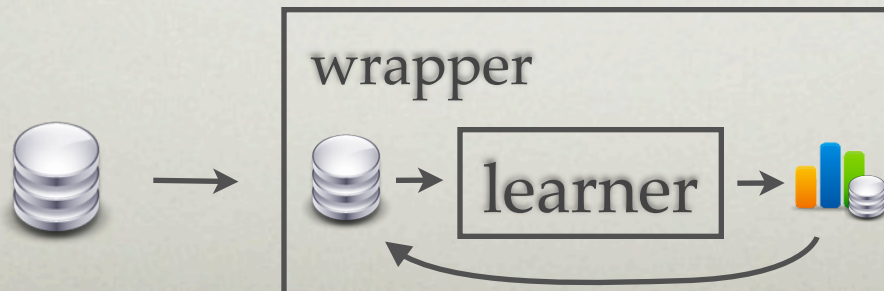
---

2 approaches (besides manual removal):

- Filter approach: Learner independent, based on data properties or simple models built by other learners



- Wrapper approach: Learner dependent, rerun learner with different attributes, select based on performance



# FILTERS

---

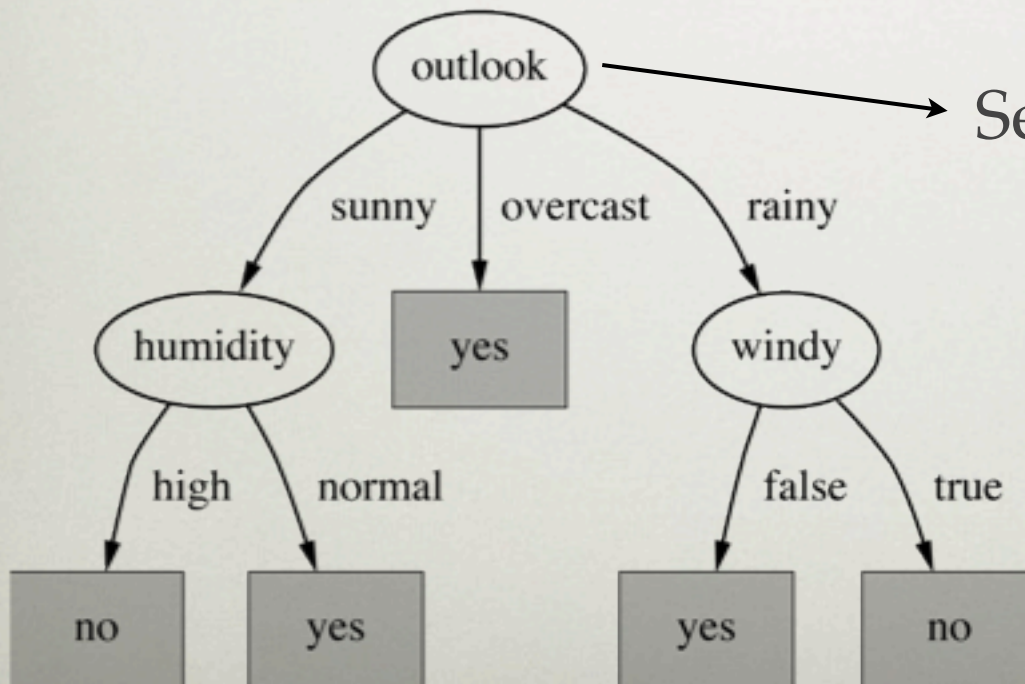
- Basic: find smallest feature set that separates data
  - Expensive, often causes overfitting
- Better: use another learner as filter
  - Many models show importance of features
    - e.g. C4.5, 1R, kNN, ...
  - Recursive: select 1 attribute, remove, repeat
  - Produces ranking: cut-off defined by user

# FILTERS

---

## Using C4.5

- select feature(s) tested in top-level node(s)
- `Decision stump' (1 node) sufficient



Select feature 'outlook',  
remove, repeat



# FILTERS

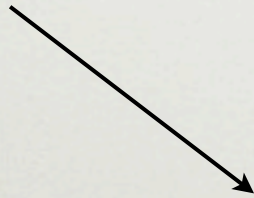
---

Using 1R

- select the 1R feature, repeat

Rule:

If(outlook=sunny) play=no, else play=yes

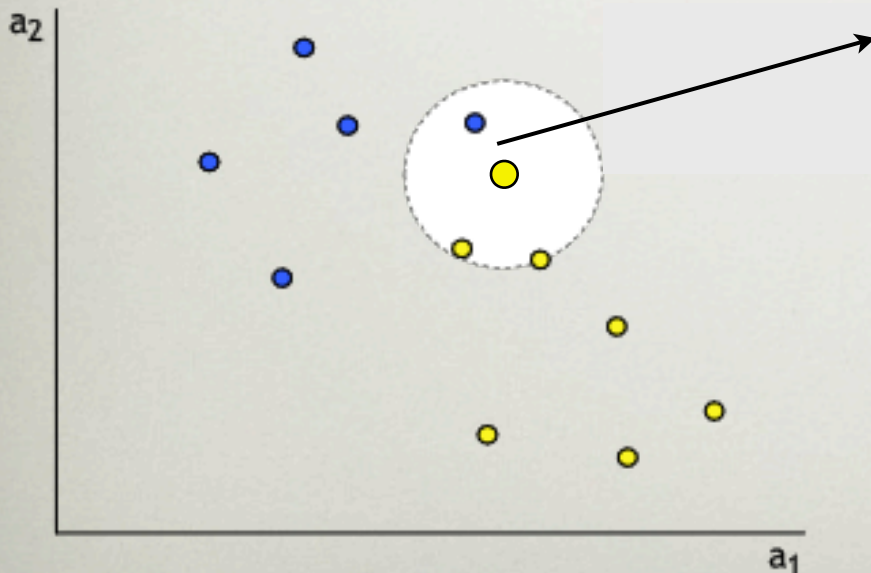


Select feature 'outlook',  
remove, repeat

# FILTERS

Using kNN: weigh features by capability to separate classes

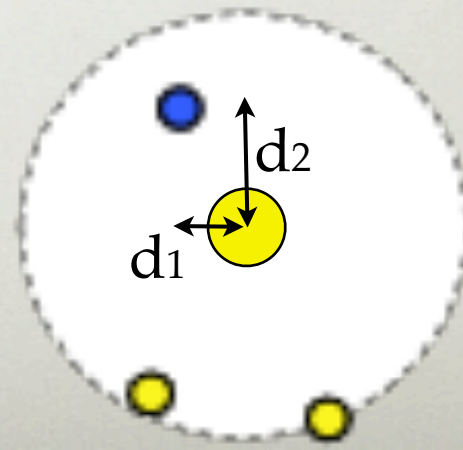
- same class: reduce weight of features with  $\neq$  value (irrelevant)
- other class: increase weight of features with  $\neq$  value (relevant)



Different classes:

increase weight of  $a_1 \propto d_1$

increase weight of  $a_2 \propto d_2$



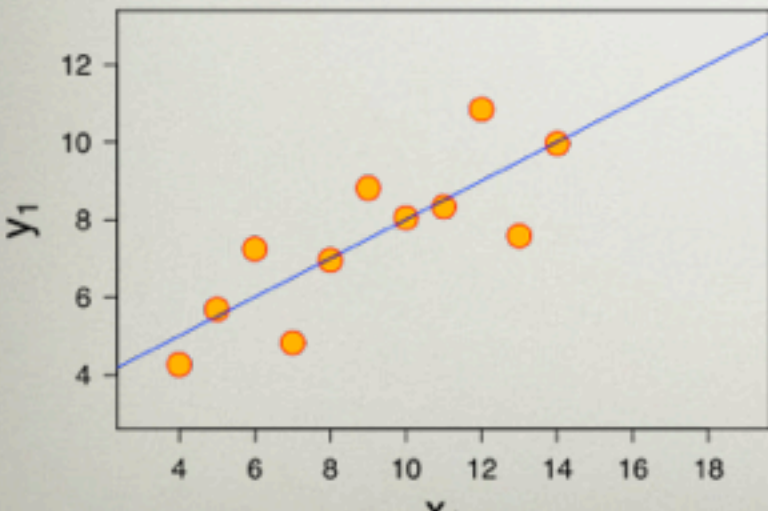
# FILTERS

---

Using Linear regression (simple or logistic)

- Select features with highest weights

$$x = w_0 + w_1 a_1 + w_2 a_2 + \dots + w_k a_k$$



Select  $w_i$ , so that  $w_i \geq w_j, i \neq j$   
remove, repeat

# Filters

---

- Direct filtering: use data properties
  - Correlation-based Feature Selection (CFS)

$$U(A, B) = 2 \frac{H(A) + H(B) - H(A, B)}{H(A) + H(B)} \in [0, 1]$$

H(): Entropy  
A: any attribute  
B: class attribute

- Select attributes with high class correlation, little intercorrelation
- Select subset by aggregating over attributes  $A_j$  for class C
  - Ties broken in favor of smaller subsets

$$\sum U(A_j, C) / \sqrt{\left( \sum \sum U(A_i, A_j) \right)}$$

- Fast, default in WEKA

# WRAPPERS

---

- Learner-dependent (selection for specific learner)
- Wrapper around learner
  - Select features, evaluate learner (e.g., cross-validation)
- Expensive
  - Greedy search:  $O(k^2)$  for  $k$  attributes
  - When using a prior ranking (only find cut-off):  $O(k)$









# WRAPPERS: SEARCH

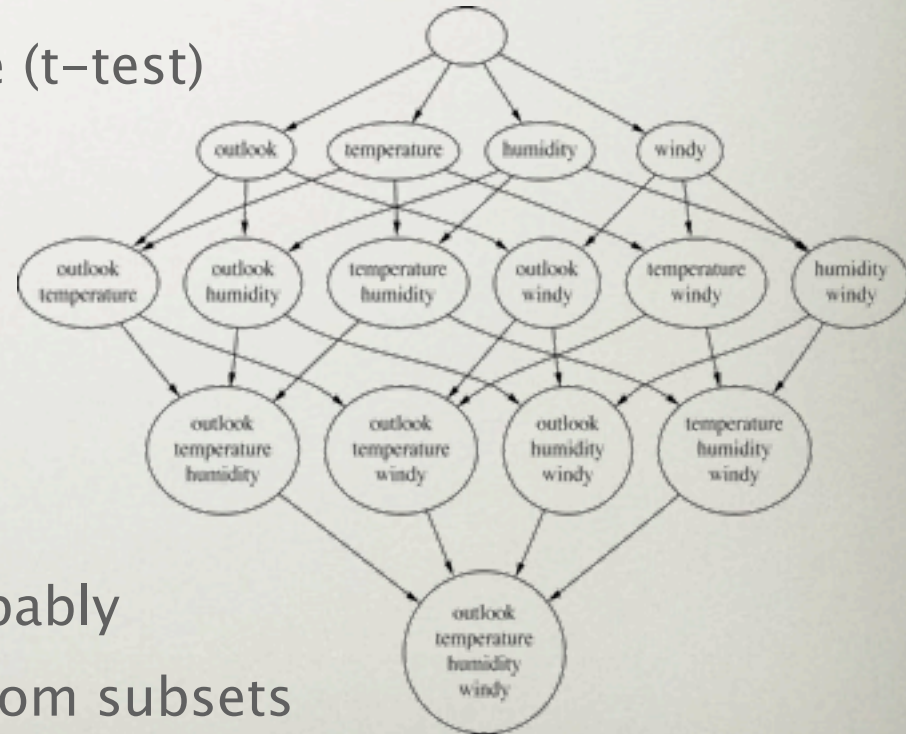
- Race search

- Stop cross-validation as soon as it is clear that feature subset is not better than currently best one
- Label winning subset per instance (t-test)

	outlook	temp	humid	windy
inst <sub>1</sub>	-1	0	1	-1
inst <sub>2</sub>	0	-1	1	-1

Selecting humid results in significantly better prediction for inst<sub>2</sub>

- Stop when one subset is better
  - better: significantly, or probably
- Schemata-search: idem with random subsets
  - if one better: stop all races, continue with winner



# PREPROCESSING WITH WEKA

---

- Attribute subset selection:
  - **ClassifierSubsetEval**: Use another learner as filter
  - **CfsSubsetEval**: Correlation-based Feature Selection
  - **WrapperSubsetEval**: Choose learner to be wrapped (with search)
- Attribute ranking approaches (with ranker):
  - **GainRatioAttributeEval, InfoGainAttributeEval**
    - C4.5-based: rank attributes by gain ratio/information gain
  - **ReliefFAttributeEval**: kNN-based: attribute weighting
  - **OneRAttributeEval, SVMAttributeEval**
    - Use 1R or SVM as filter for attributes, with recursive feat. elim.

# THE 'SELECT ATTRIBUTES' TAB

The screenshot shows the 'Select attributes' tab in Weka Explorer. The interface includes several sections: 'Attribute Evaluator' with a 'Choose' button and 'CfsSubsetEval' selected; 'Search Method' with a 'Choose' button and 'BestFirst -D 1 -N 5' selected; 'Attribute Selection Mode' with 'Use full training set' selected and 'Folds' set to 10; a dropdown menu showing '(Nom) surgical\_lesion'; 'Start' and 'Stop' buttons; a 'Result list' showing '15:46:35 - BestFirst + CfsSubsetEval'; and a large text area displaying the results of the attribute selection process. Blue callout boxes with white text point to these elements: 'Select attribute selection approach' points to 'CfsSubsetEval'; 'Select search strategy' points to 'BestFirst -D 1 -N 5'; 'Select class attribute' points to the '(Nom) surgical\_lesion' dropdown; and 'Selected attributes or ranked list' points to the result list entry.

Weka Explorer

Preprocess Classify Cluster Associate **Select attributes** Visualize

Attribute Evaluator  
Choose CfsSubsetEval

Select attribute selection approach

Search Method  
Choose BestFirst -D 1 -N 5

Select search strategy

Attribute Selection Mode  
 Use full training set  
 Cross-validation Folds 10 Seed 1

(Nom) surgical\_lesion

Select class attribute

Start Stop

Result list (right-click for options)  
15:46:35 - BestFirst + CfsSubsetEval

Selected attributes or ranked list

Attribute selection  
Evaluation mode:  
==== Attribute Selection====  
Search Method:  
Best first search  
Start seed: 1  
Search depth: 5  
Stale search: 0  
Total number of subsets evaluated: 115  
Merit of best subset found: 0.294  
Attribute Subset Evaluator (supervised, Class (nominal): 23 surgical\_lesion)  
CFS Subset Evaluator  
Including locally predictive attributes  
Selected attributes: 1,4,5,17,20 : 5  
surgery  
pulse  
respiratory\_rate  
abdomen  
abdominocentesis\_appearance

Status  
OK

Log x 0

# THE 'SELECT ATTRIBUTES' TAB

The screenshot shows the 'Select attributes' tab in Weka Explorer. The interface includes several sections:

- Attribute Evaluator:** Set to 'CfsSubsetEval'.
- Search Method:** Set to 'BestFirst -D 1 -N 5'.
- Attribute Selection Mode:** 'Use full training set' is selected. 'Folds' is set to 10 and 'Seed' is 1.
- Attribute Selection:** A dropdown menu shows '(Nom) surgical\_lesion'.
- Buttons:** 'Start' and 'Stop' buttons are visible.
- Result list (right):** Shows a timestamp '15:46:35 - Best'.
- Output Panel:** Displays the search method and results:

```
==== Attribute Selection====  
Search Method:  
Best first search  
Start search at 15:46:35  
Search completed at 15:46:35  
Stale search at 15:46:35  
Total number of subsets evaluated: 115  
Merit of best subset found: 0.294
```
- Selected attributes:** A list of selected attributes and their counts:

```
Selected attributes: 1,4,5,17,20 : 5  
surgery  
pulse  
respiratory_rate  
abdomen  
abdominocentesis_appearance
```
- Status:** Shows 'OK'.
- Log:** A 'Log' button is present.

Callouts from blue boxes point to the following elements:

- 'Select attribute selection approach' points to the 'Attribute Evaluator' section.
- 'Select search strategy' points to the 'Search Method' section.
- 'Select class attribute' points to the 'Attribute Selection' dropdown.
- 'Selected attributes or ranked list' points to the 'Selected attributes' output.

# THE 'PREPROCESS' TAB

The screenshot shows the Weka Explorer interface with the 'Preprocess' tab selected. The 'Filter' section shows 'AttributeSelection' as the active filter. The 'Current relation' is 'autos' with 205 instances and 26 attributes. The 'Attributes' section has buttons for 'All', 'None', 'Invert', and 'Pattern'. A list of attributes is shown with checkboxes: 1 normalized-losses (checked), 2 make (unchecked), 3 fuel-type (checked), 4 aspiration (checked), 5 num-of-doors (checked and highlighted), 6 body-style (unchecked), 7 drive-wheels (unchecked), 8 engine-location (unchecked), 9 wheel-base (unchecked), 10 length (unchecked), 11 width (unchecked). A 'Remove' button is at the bottom. The 'Selected attribute' section shows 'Name: num-of-doors' and 'Missing: 2 (1%)'. Below this are two bar charts with values 114 and 89. The status bar at the bottom shows 'Status OK' and a 'Log' button.

Use attribute selection feedback to remove unnecessary attributes (manually)

OR: select 'AttributeSelection' as 'filter' and apply it (will remove irrelevant attributes and rank the rest)

# DATA ENGINEERING

---

- Attribute selection (feature selection)
  - Remove features with little / no predictive information
- **Attribute discretization**
  - Convert numerical attributes to nominal ones
- Data transformations (feature generation)
  - Transform data to another representation
- Dirty data
  - Remove missing values or outliers

# ATTRIBUTE DISCRETIZATION

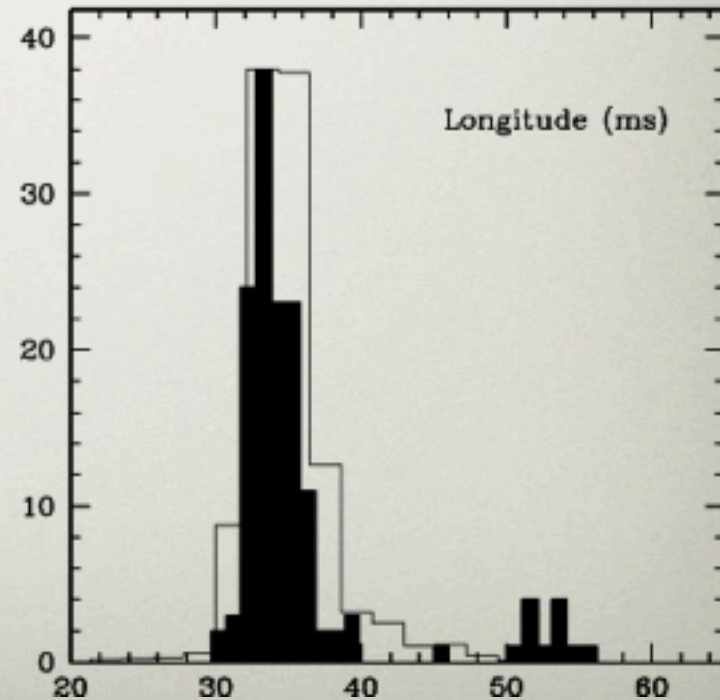
---

- Some learners cannot handle numeric data
  - ‘Discretize’ values in small intervals
  - Always loses information: try to preserve as much as possible
- Some learners can handle numeric values, but are:
  - Naive (Naïve Bayes assumes normal distribution)
  - Slow (1R *sorts* instances before discretization)
  - Local (C4.5 discretizes in nodes, on less and less data)
- Discretization:
  - Transform into one  $k$ -valued discretized attribute
  - Replace with  $k-1$  new **binary** attributes
    - values  $a, b, c$ :  $a \rightarrow \{0, 0\}$ ,  $b \rightarrow \{1, 0\}$ ,  $c \rightarrow \{1, 1\}$

# UNSUPERVISED DISCRETIZATION

---

- Determine intervals without knowing class labels
  - When clustering, the only possible way!
- Strategies:
  - **Equal-interval binning**: create intervals of fixed width
    - often creates bins with many or very few examples

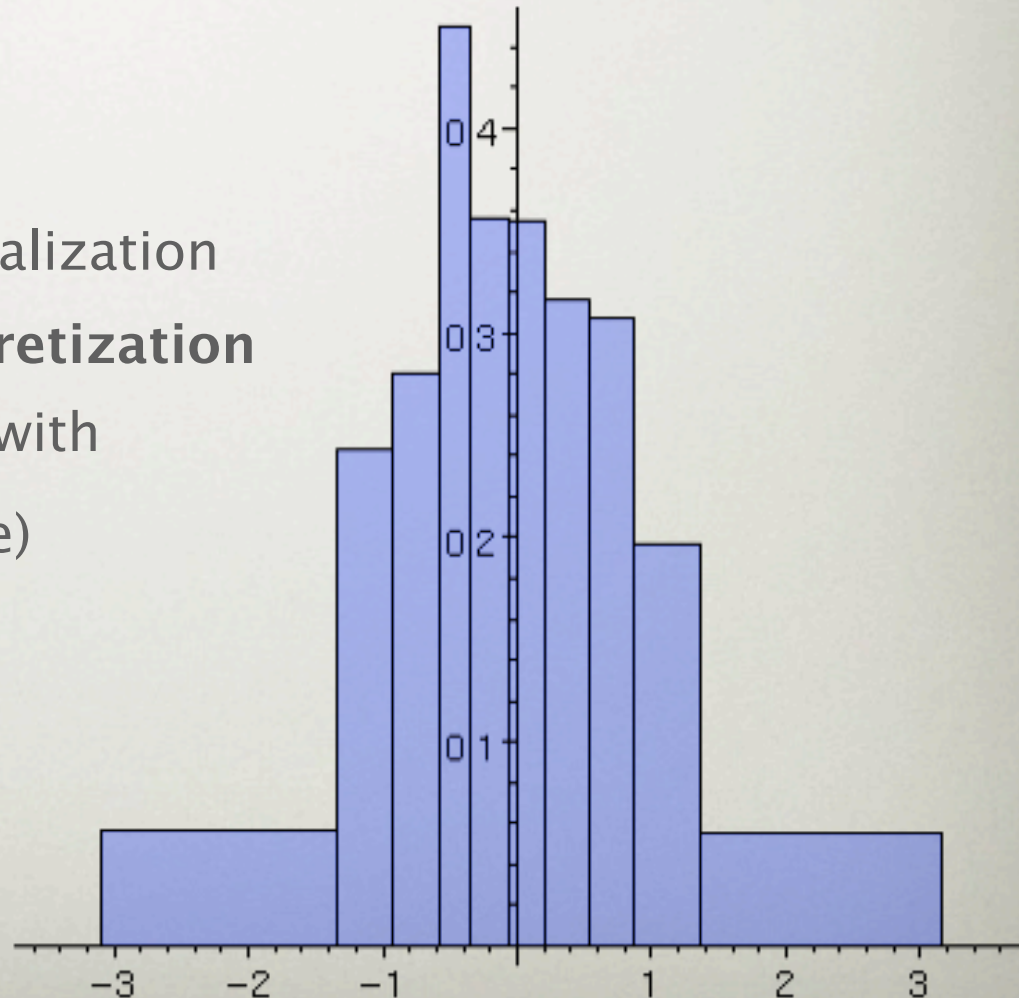




# UNSUPERVISED DISCRETIZATION

---

- Strategies:
  - **Equal-frequency binning:**
    - create bins of equal size
    - also called histogram equalization
  - **Proportional k-interval discretization**
    - equal-frequency binning with
    - # bins =  $\sqrt{\text{dataset size}}$



# SUPERVISED DISCRETIZATION

---

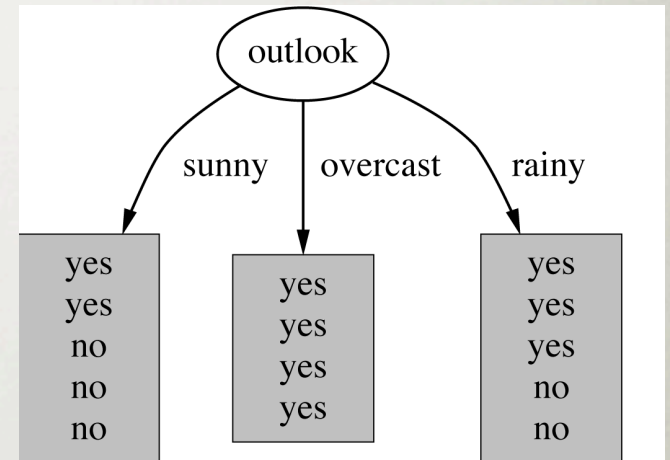
- Supervised approach usually works better
  - Better if all/most examples in a bin have same class
  - Correlates better with class attribute (less predictive info lost)
- Different approaches
  - Entropy-based
  - Bottom-up merging
  - ...

# ENTROPY-BASED DISCRETIZATION

Split data in the same way C4.5 would: each leaf = bin

- Use entropy as splitting criterion

$$H(p) = -p \log(p) - (1-p) \log(1-p)$$



Outlook = Sunny:

$$\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5 \log(2/5) - 3/5 \log(3/5) = 0.971 \text{ bits}$$

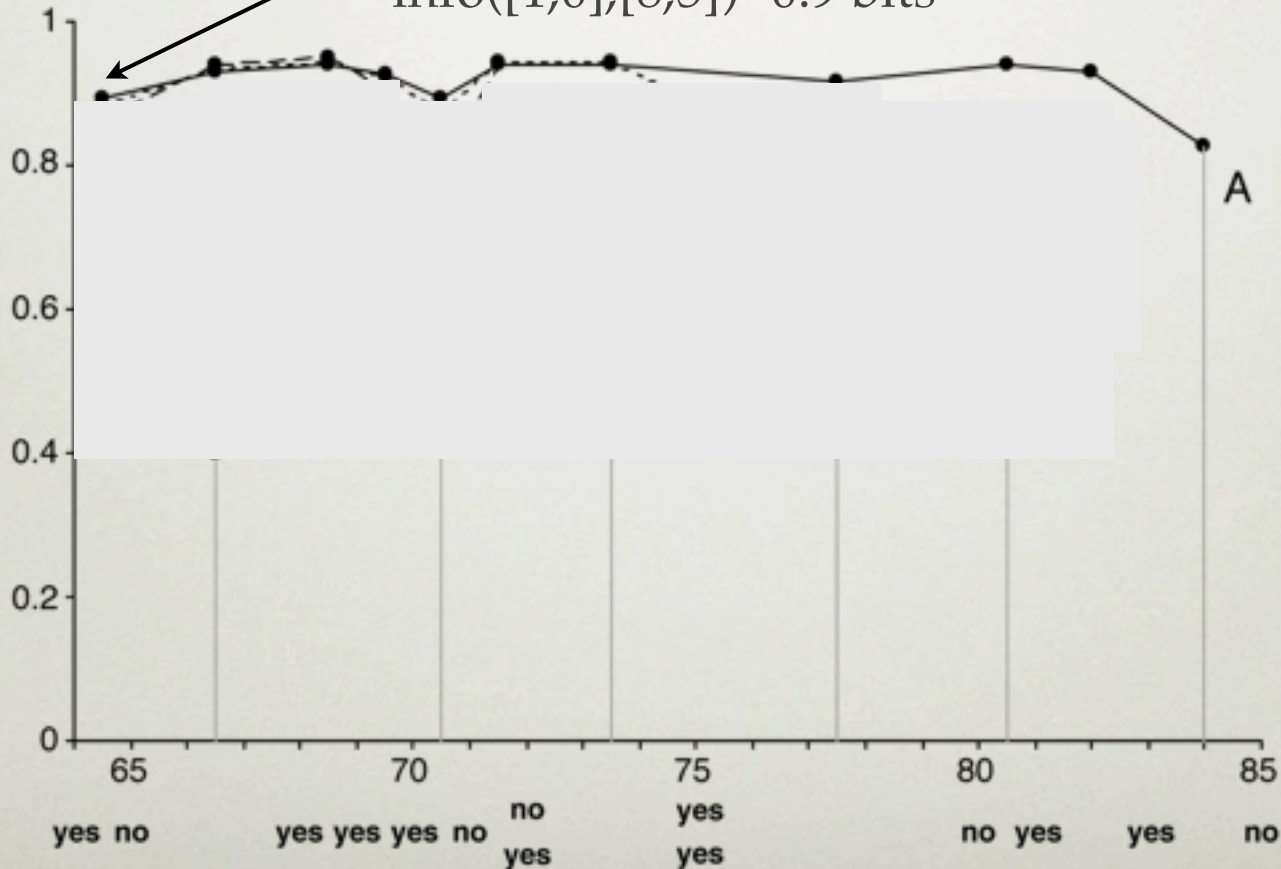
Expected information for outlook:

$$\begin{aligned} \text{info}([3,2],[4,0],[3,2]) &= (5/14) \times 0.971 + (4/14) \times 0 + (5/14) \times 0.971 \\ &= 0.693 \text{ bits} \end{aligned}$$

# EXAMPLE: TEMPERATURE ATTRIBUTE

Temperature	64	65	68	69	70	71	72	72	75	75	80	81	83	85
Play	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

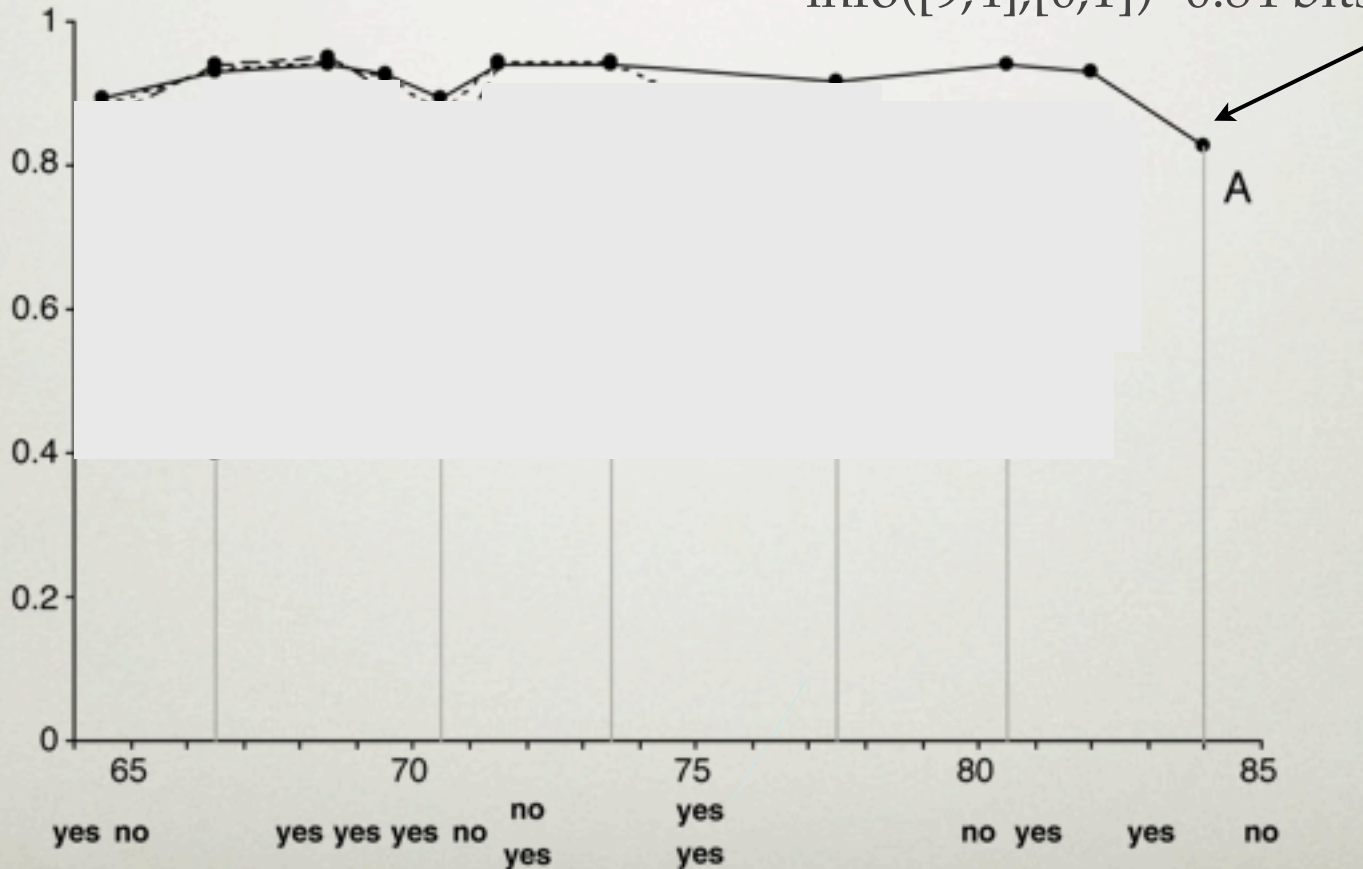
$\text{info}([1,0],[8,5])=0.9$  bits



# EXAMPLE: TEMPERATURE ATTRIBUTE

Temperature	64	65	68	69	70	71	72	72	75	75	80	81	83	85
Play	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

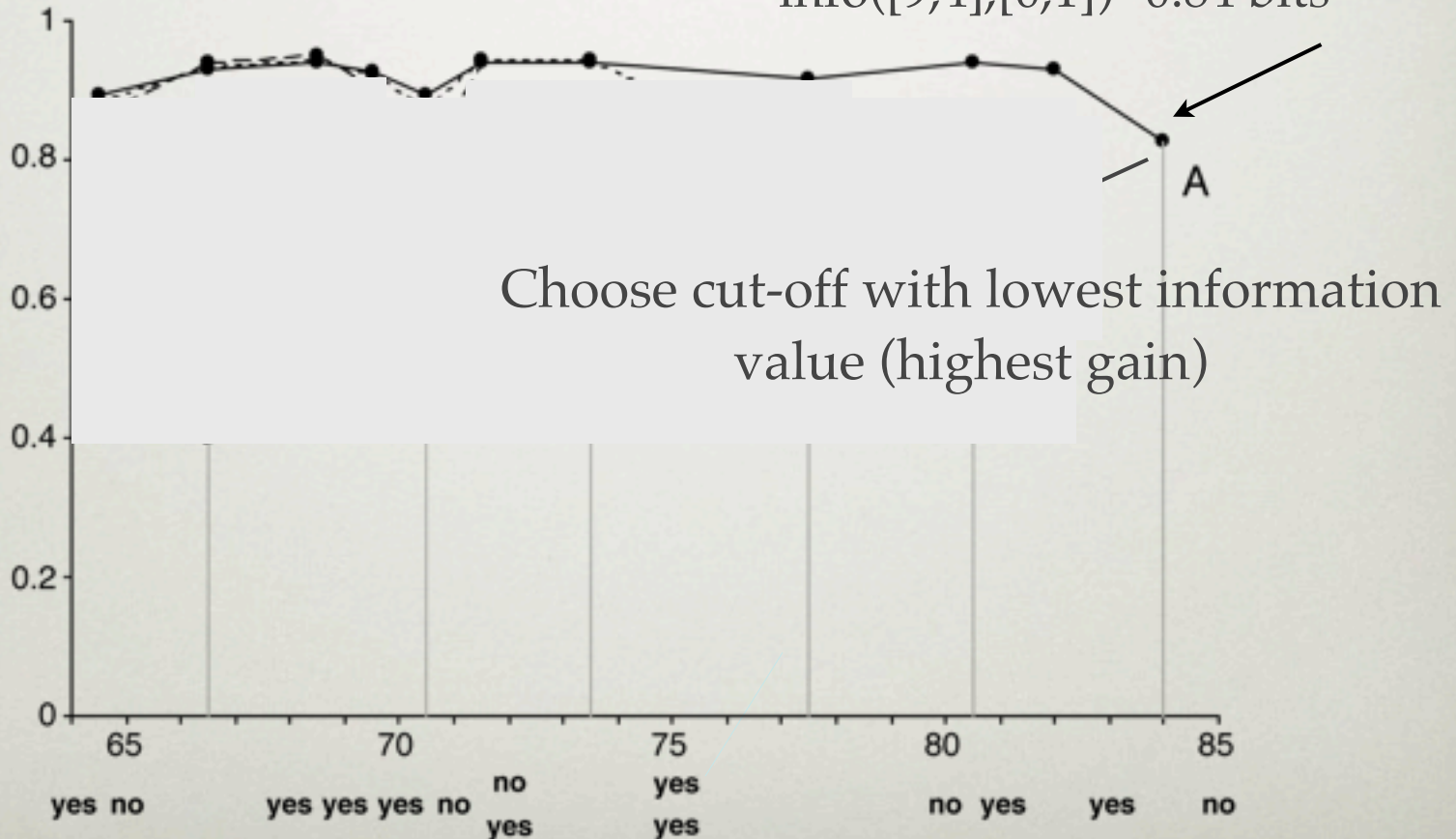
$info([9,4],[0,1])=0.84$  bits



# EXAMPLE: TEMPERATURE ATTRIBUTE

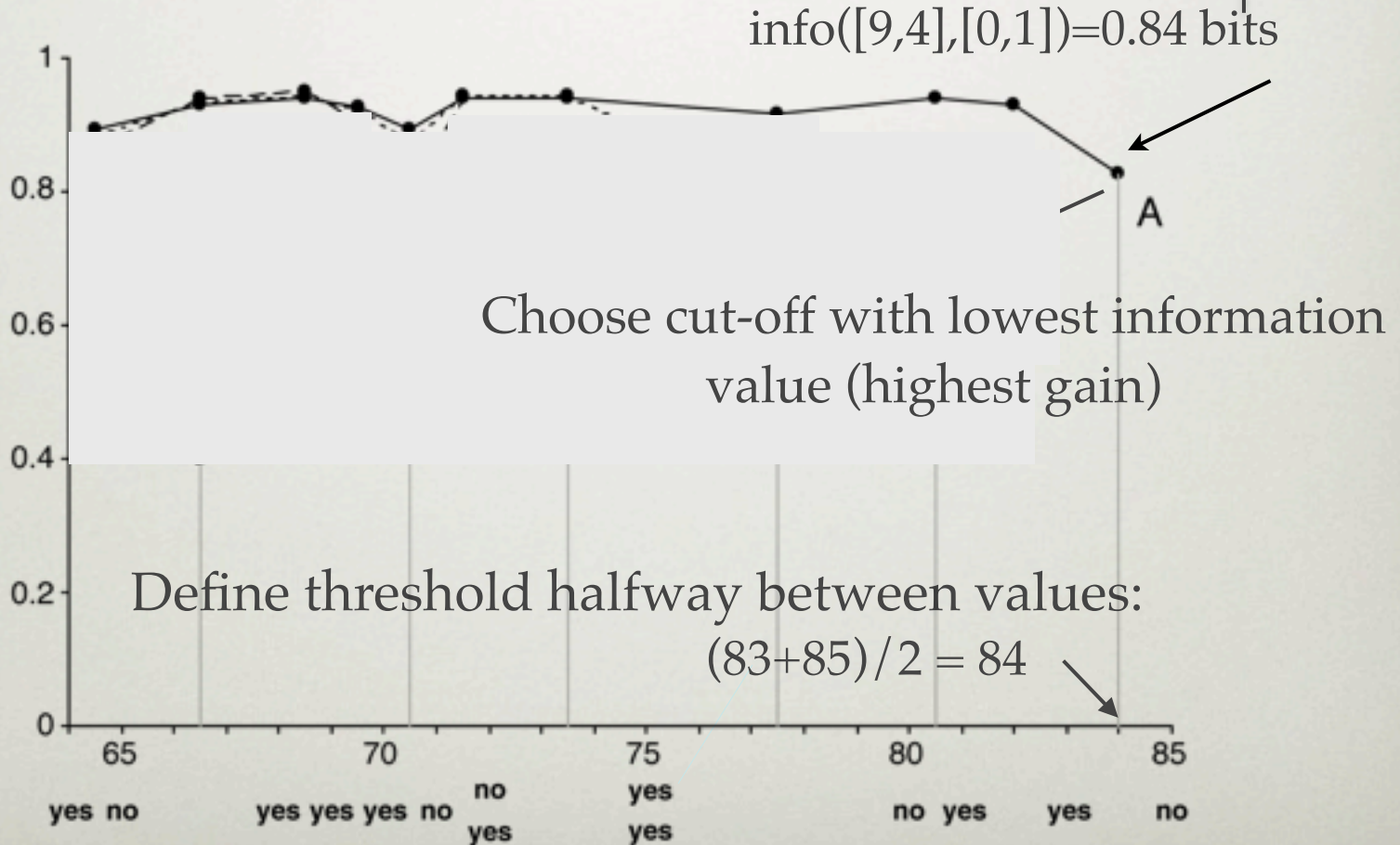
Temperature	64	65	68	69	70	71	72	72	75	75	80	81	83	85
Play	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

$info([9,4],[0,1])=0.84$  bits



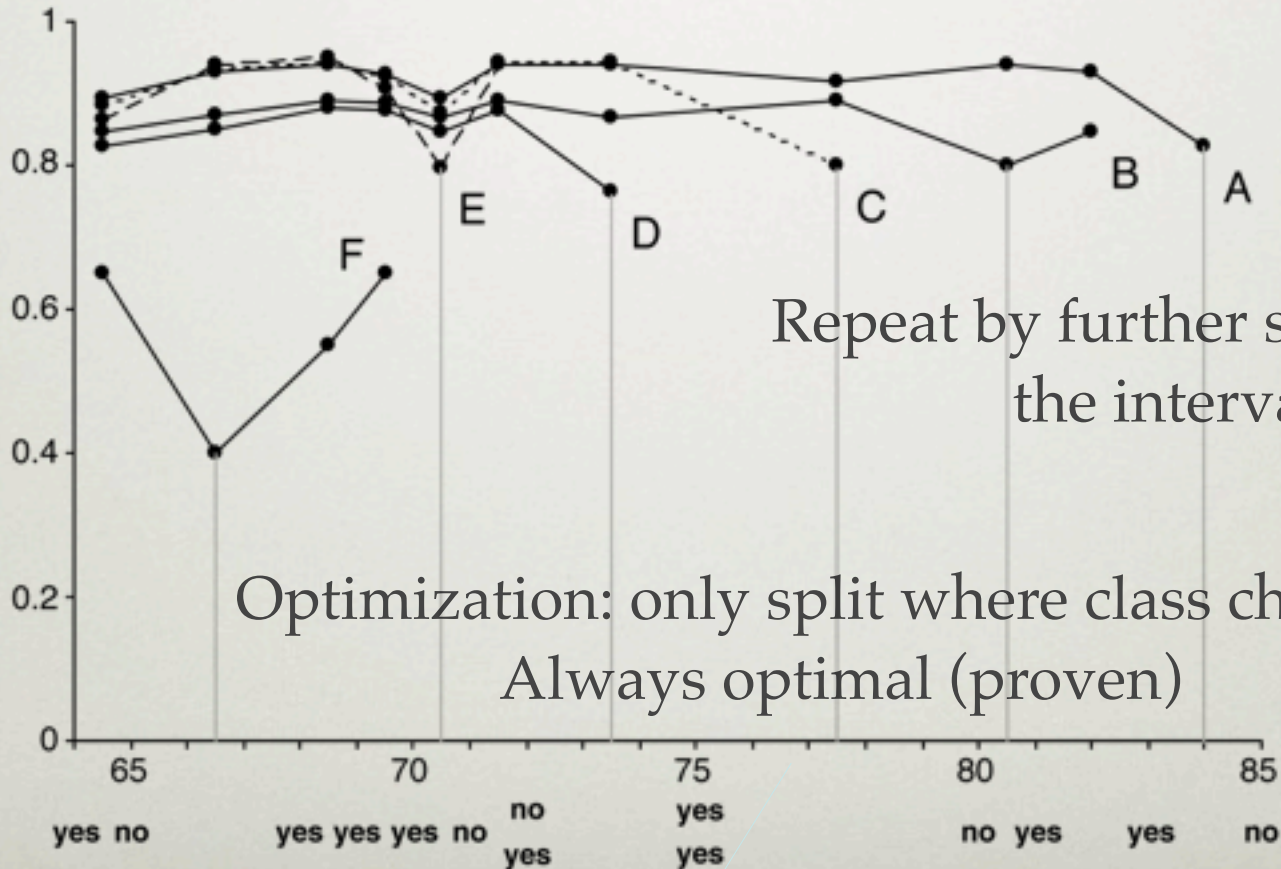
# EXAMPLE: TEMPERATURE ATTRIBUTE

Temperature	64	65	68	69	70	71	72	72	75	75	80	81	83	85
Play	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No



# EXAMPLE: TEMPERATURE ATTRIBUTE

Temperature	64	65	68	69	70	71	72	72	75	75	80	81	83	85
Play	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No



Repeat by further subdividing the intervals

Optimization: only split where class changes  
Always optimal (proven)



# ENTROPY-BASED DISCRETIZATION

---

Split data in the same way C4.5 would: each leaf = bin

- Use entropy as splitting criterion
- Use minimum description length principle as stopping criterion
  - Stop when description of attribute cannot be compressed more
    - Description of splitting points ( $\log_2[N - 1]$  bits) +  
Description of bins (class distribution)
  - Short if few thresholds, homogenous (single-class) bins
  - Split worthwhile if information gain >

$$\frac{\log_2(N-1)}{N} + \frac{\log_2(3^k - 2) - kE + k_1E_1 + k_2E_2}{N}$$

Entropy  $E$ , number of classes  $k$  in original set  $(E, k)$ ,  
subset before threshold  $(E_1, k_1)$ , after threshold  $(E_2, k_2)$

# SUPERVISED DISCRETIZATION: ALTERNATIVES

---

- Work bottom-up: each value in its own bin, then merge
  - Replace MDL by chi-squared test
  - Tests hypothesis that two adjacent intervals are independent of the class. If so, merge the intervals.
- Use dynamic programming to find optimum k-way split for given additive criterion
  - Requires time quadratic in the number of instances
  - Can be done in linear time if error rate is used (not entropy)

# MAKE DATA NUMERIC

---

- Inverse problem
- Some algorithms assume numeric features
  - e.g. kNN
- Classification
  - You could just number nominal values 1..k (a=0,b=1,c=2,...)
    - However, there isn't always a logical order
  - Replace attribute with k nominal values by k binary attributes ('indicator attributes')
  - Value '1' if example has nominal value corresponding to that indicator attribute, '0' otherwise:  $A \rightarrow \{1,0\}$ ,  $B \rightarrow \{0,1\}$

A		Aa	Ab
a	→	1	0
b		0	1

# MAKE DATA NUMERIC

---

- Regression
  - Value = average of all target values corresponding to same nominal attribute value

A	target
a	0.9
a	0.8
b	0.7
b	0.6

→

A'	target
0.85	0.9
0.85	0.8
0.65	0.7
0.65	0.6

# DISCRETIZATION WITH WEKA

---

- Discretization:
  - Unsupervised:
    - **Discretize**: Equal-width or equal-frequency
    - **PKIDiscretize**: equal-frequency with  $\# \text{bins} = \sqrt{\# \text{values}}$
  - Supervised:
    - **Discretize**: Entropy-based discretization

# DISCRETIZATION WITH WEKA

---

- Nominal to numerical:
  - Supervised:
    - **NominalToBinary**: for regression (use average target value)
  - Unsupervised:
    - **MakeIndicator**: replaces nominal with boolean attribute
    - **NominalToBinary**: creates 1 binary attribute for each value

# WEKA: DISCRETIZATION FILTER

The screenshot shows the Weka Explorer application window. The 'Preprocess' tab is active. In the 'Filter' section, the 'Discretize -R first-last' filter is selected. A blue arrow points from the 'Choose' button to the filter name. A blue callout box on the right contains the text: 'Select (un)supervised > attribute > Discretize'. The 'Current relation' section shows 'Relation: None', 'Instances: None', and 'Attributes: None'. The 'Attributes' section has buttons for 'All', 'None', 'Invert', and 'Pattern'. The 'Selected attribute' section shows 'Name: None', 'Missing: None', 'Distinct: None', 'Type: None', and 'Unique: None'. The 'Status' bar at the bottom says 'Welcome to the Weka Explorer' and has a 'Log' button and a small icon.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose Discretize -R first-last Apply

Current relation

Relation: None  
Instances: None  
Attributes: None

Selected attribute

Name: None  
Missing: None  
Distinct: None  
Type: None  
Unique: None

Attributes

All None Invert Pattern

Remove

Status

Welcome to the Weka Explorer

Log x 0

Select (un)supervised > attribute > Discretize

# DATA ENGINEERING

---

- Attribute selection (feature selection)
  - Remove features with little / no predictive information
- Attribute discretization
  - Convert numerical attributes to nominal ones
- **Data transformations (feature generation)**
  - Transform data to another representation
- Dirty data
  - Remove missing values or outliers



# DATA TRANSFORMATIONS

---

- Often, a data transformation can lead to new insights in the data and better performance
- Simple transformations:
  - Subtract two 'date' attributes to get 'age' attribute
  - If linear relationship is suspected between numeric attributes A and B: add attribute  $A/B$
- Clustering the data
  - add one attribute recording the cluster of each instance
  - add k attributes with membership of each cluster

# DATA TRANSFORMATIONS

---

- Other transformations:
  - Add noise to data (to test robustness of algorithm)
  - Obfuscate the data (to preserve privacy)

# DATA TRANSFORMATIONS

---

- Convert text to table data
  - *Bag of words:*
    - ♦ Each instance is a document or string
    - ♦ Attributes are words, phrases, n-grams (e.g., `to be`)
    - ♦ Attribute values: term frequencies ( $f_{ij}$ )
      - ♦ frequency of word  $i$  in document  $j$

Document	$f_i(\text{to})$	$f_i(\text{be})$	$f_i(\text{or})$	$f_i(\text{not})$
`To be or not to be`	2	2	1	1
`Or not`	0	0	1	1

# DATA TRANSFORMATIONS

---

Document	$f_i(\text{to})$	$f_i(\text{be})$	$f_i(\text{or})$	$f_i(\text{not})$
`To be or not to be'	2	2	1	1
`Or not'	0	0	1	1

- Language-dependent issues:
  - Delimiters (ignore periods in 'e.g.'?)
  - Stopwords (the, is, at, which, on, ...)
  - Low frequency words (ignore to reduce # features)
- Better alternatives:  $\log(1+f_{ij})$  or TFxIDF  
(*term frequency x inverse document frequency*)=

$$f_{ij} \log \frac{\# \text{ documents}}{\# \text{ documents\_that\_include\_word } i}$$

# DATA TRANSFORMATION FILTERS

The screenshot shows the Weka Explorer application window. The title bar reads "Weka Explorer". The main menu bar includes "Preprocess", "Classify", "Cluster", "Associate", "Select attributes", and "Visualize". Below the menu bar is a toolbar with buttons for "Open file...", "Open URL...", "Open DB...", "Generate...", "Undo", "Edit...", and "Save...".

The "Filter" section is active, showing a "Choose" button and a text field containing "Discretize -R first-last", with an "Apply" button to the right. Below this, the "Current relation" section displays "Relation: None", "Instances: None", and "Attributes: None". The "Selected attribute" section displays "Name: None", "Missing: None", "Distinct: None", "Type: None", and "Unique: None".

The "Attributes" section contains buttons for "All", "None", "Invert", and "Pattern", and a large empty list area. A "Remove" button is located at the bottom of this section. A blue arrow points from the "Choose" button to a blue callout box.

**Select unsupervised >  
attribute > ...**

The "Status" bar at the bottom left says "Welcome to the Weka Explorer". The bottom right corner features a "Log" button and a small icon with "x 0".

# SOME WEKA IMPLEMENTATIONS

---

- Simple transformations:
  - **AddCluster**: clusters data and adds attribute with resulting cluster for each data point
  - **ClusterMembership**: clusters data and adds k attributes with membership of each data point in each of k clusters
  - **AddNoise**: changes a percentage of attribute's values
  - **Obfuscate**: renames attribute names and nominal/string values to random name

# SOME WEKA IMPLEMENTATIONS

---

- Other transformations
  - **StringToWordVector**: produces bag of words (many options)
  - **RELAGGS**: propositionalization algorithm: converts relational data (e.g. relational database) to single table
  - **TimeSeriesDelta**: Replace attribute values with difference between current and past/previous instance
  - **TimeSeriesTranslate**: Replace attribute values with equivalent value in past/previous instance

# SOME WEKA IMPLEMENTATIONS

---

- Also data projections (out of scope):
  - **PrincipalComponents**: does PCA transformation (constructs new (smaller) feature set to maximize variance per feature)
  - **RandomProjection**: Random projection to lower-dimensional subspace
  - **Standardize**: standardizes all numeric attributes to have zero mean and unit variance



# DATA ENGINEERING

---

- Attribute selection (feature selection)
  - Remove features with little / no predictive information
- Attribute discretization
  - Convert numerical attributes to nominal ones
- Data transformations (feature generation)
  - Transform data to another representation
- **Dirty data**
  - Remove missing values or outliers

# SOME DATA 'CLEANING' METHODS IN WEKA

---

- Unsupervised > Instance:
  - **RemoveWithValues**: removes instances with certain value and/or with missing values
  - **RemoveMisclassified**: removes instances incorrectly classified by specified classifier, useful for removing outliers
  - **RemovePercentage**: removes given percentage of instances
- Supervised > Instance:
  - **Resample**: produces random subsample, with replacement
  - **SpreadSubSample**: produces random subsample, with given spread between class frequencies, with replacement

# SOME DATA 'CLEANING' METHODS

---

- Unsupervised > Attribute:
  - **ReplaceMissingValues**: replaces all missing values for nominal / numeric attributes with mode / mean of training data