Data Mining
(EECS 4412)

Data Preprocessing

Parke Godfrey

EECS

Lassonde School of Engineering

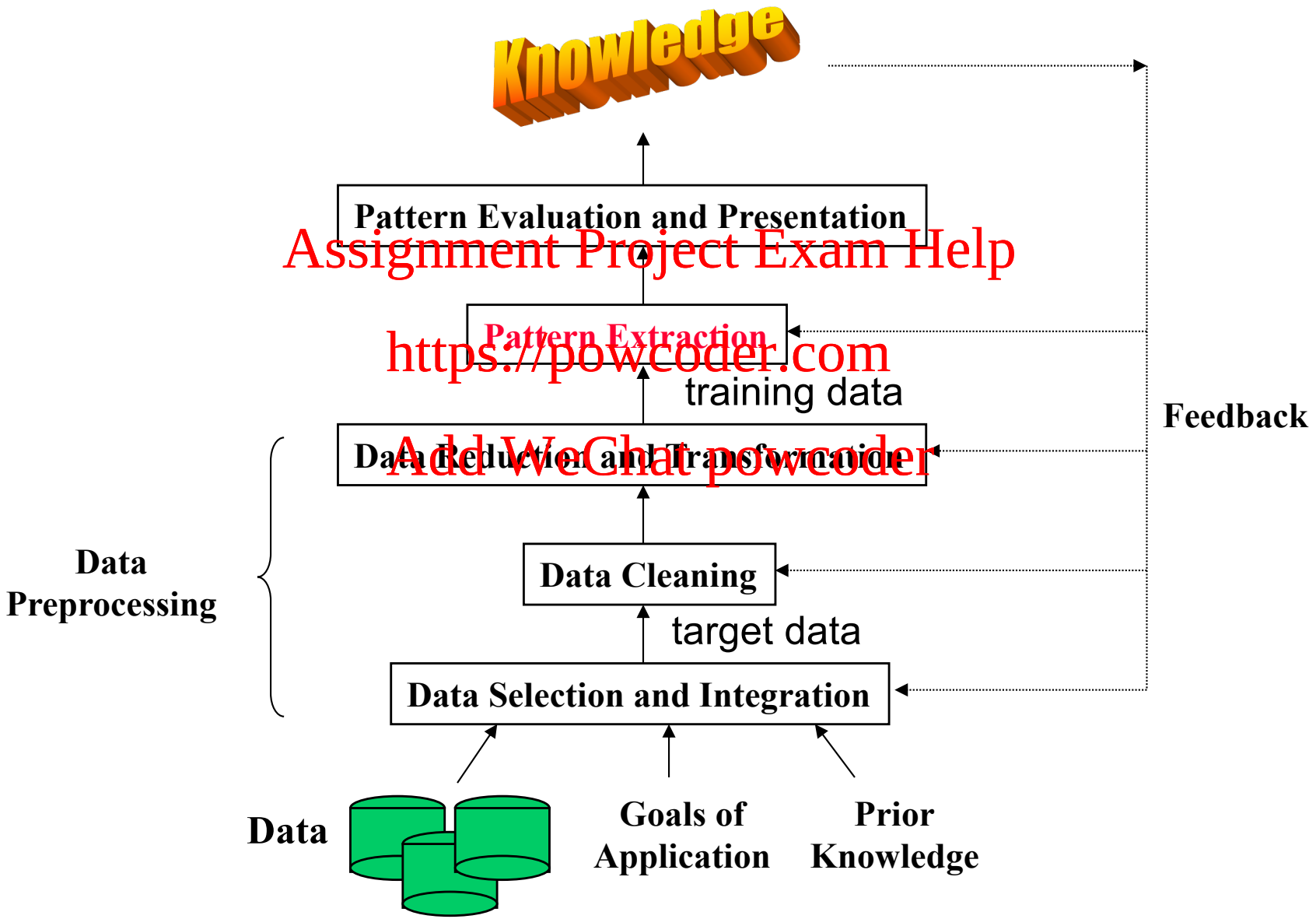
York University

Thanks to

Professor Aijun An

for curation & use of these slides.

Process of Data Mining and KDD



Outline

- Why preprocess the data?
- Data integration
- Data cleaning
- Data transformation
- Data reduction
- Discretization

Why Data Preprocessing?

- Heterogeneous data – data integration
  - From various departments, in various forms
- Dirty data – data cleaning
  - Incomplete data: missing attribute values
    - e.g., occupation=“”
  - Noisy data: containing errors
    - e.g., Salary=“-10”
  - Discrepancies in codes or names
    - e.g., US=USA
- Data not in the right format – data transformation
  - Normalization, discretization, etc.
- A huge amount of data – data reduction
  - Speed up mining

No quality data, no quality mining results!

Major Tasks in Data Preprocessing

- Data integration
  - Integration of multiple databases or files
- Data cleaning
  - Fill in missing values, identify outliers and smooth out noisy data, and resolve discrepancies
- Data transformation
  - Feed right data to the mining algorithm
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction and data transformation but with particular importance, transform numerical data into symbolic (discrete) data

Data Integration

- Data integration:
  - combines data from multiple sources into a coherent store
- Schema integration
  - integrate metadata from different sources
  - *Entity identification problem*: identify real world entities from multiple data sources, e.g., A.cust-id ≡ B.cust-#
- Detecting and resolving data value conflicts
  - for the same real world entity, attribute values from different sources are different
  - possible reasons: different representations, different scales,
    - e.g., hotel price in different currencies, metric vs. British units
    - e.g., Age="42" Birthday="03/07/1997"
    - e.g., Was rating "1,2,3", now rating "A, B, C"

Data Cleaning

- Why is data dirty?
  - Incomplete data come from
    - human/hardware/software problems (e.g. equipment malfunction)
    - different consideration between the time when the data was collected and when it is analyzed.
      - certain data may not be considered important at the time of entry
  - Noisy data come from the process of
    - data collection
    - data entry
    - data transmission
- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data

How to Handle Missing Values?

- Fill in the missing value manually: tedious + infeasible?
- Ignore the tuple containing missing values:

Cust-id	Age	Gender	Income	Credit
1	36	M	$54K	good
2	24	M	$20K	bad
3	37	M	$50K	?
4	28	F	$30K	good
5	55	F	$25K	good
6	65	?	$16K	bad
7	33	F	$10K	bad

  - usually done when class label is missing (assuming the task is classification)
  - not effective when missing values in attributes spread in many different tuples.
- Fill it in with a value "unknown"
  - patterns containing "unknown" is ugly

How to Handle Missing Values? (*Contd.*)

- Global estimation
  - the attribute mean/median for numeric attributes
  - the most probable value for symbolic (i.e. categorical) attributes
- Local estimation: smarter
  - the attribute mean/median for all the tuples belonging to the same class (for numeric attributes)
  - the most probable value within the same class (for symbolic attributes)

Cust-id	Age	Gender	Income	Credit
1	36	F	$55K	good
2	24	?	$20K	bad
3	37	F	$50K	good
4	23	F	$30K	good
5	55	F	$25K	good
6	35	M	?	bad
7	33	M	$10K	bad

- Use inference-based prediction techniques, such as
  - Nearest-neighbor estimator, decision tree, regression, neural network, etc.
  - good method with overhead

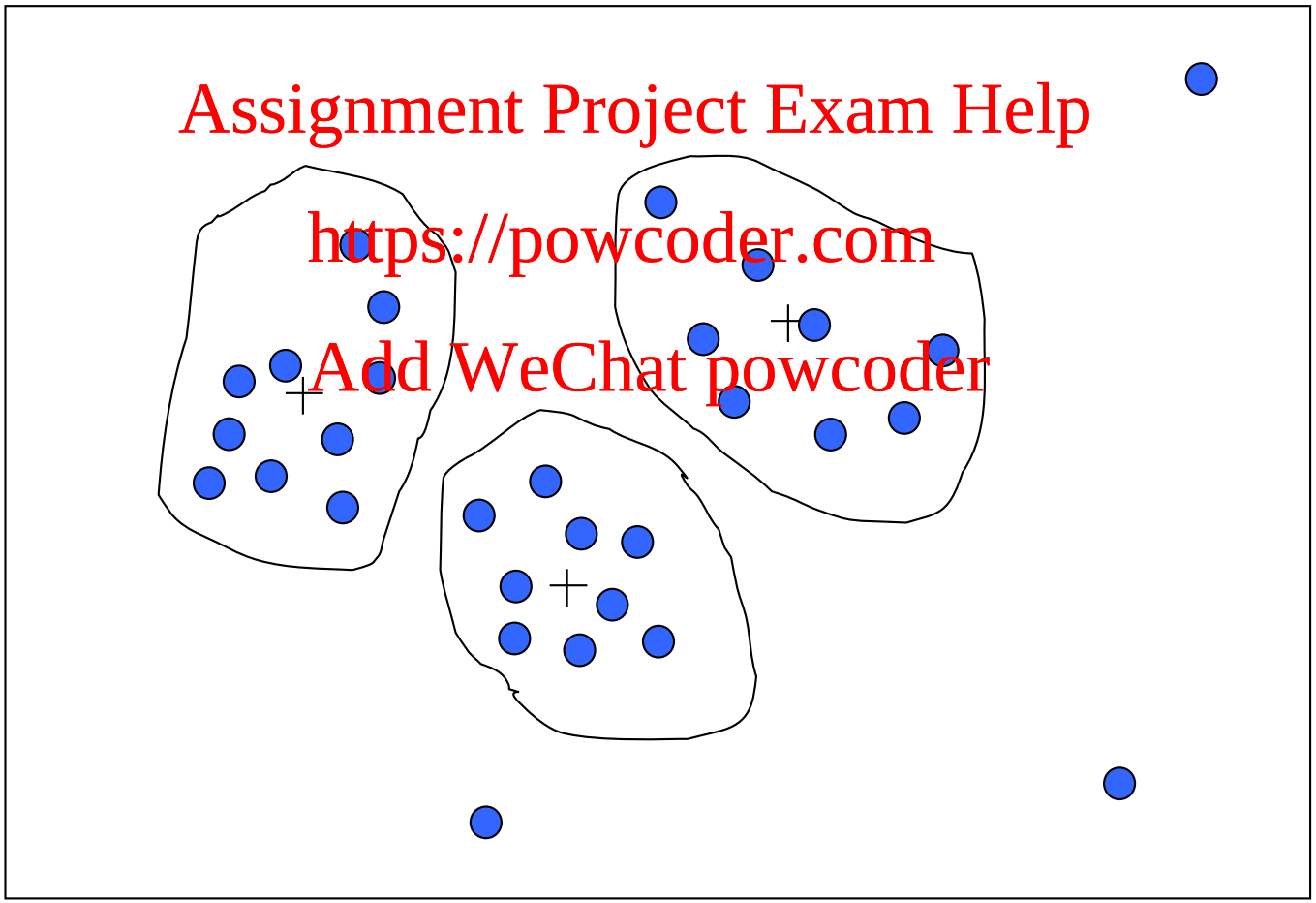
Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems

How to Handle Noisy Data?

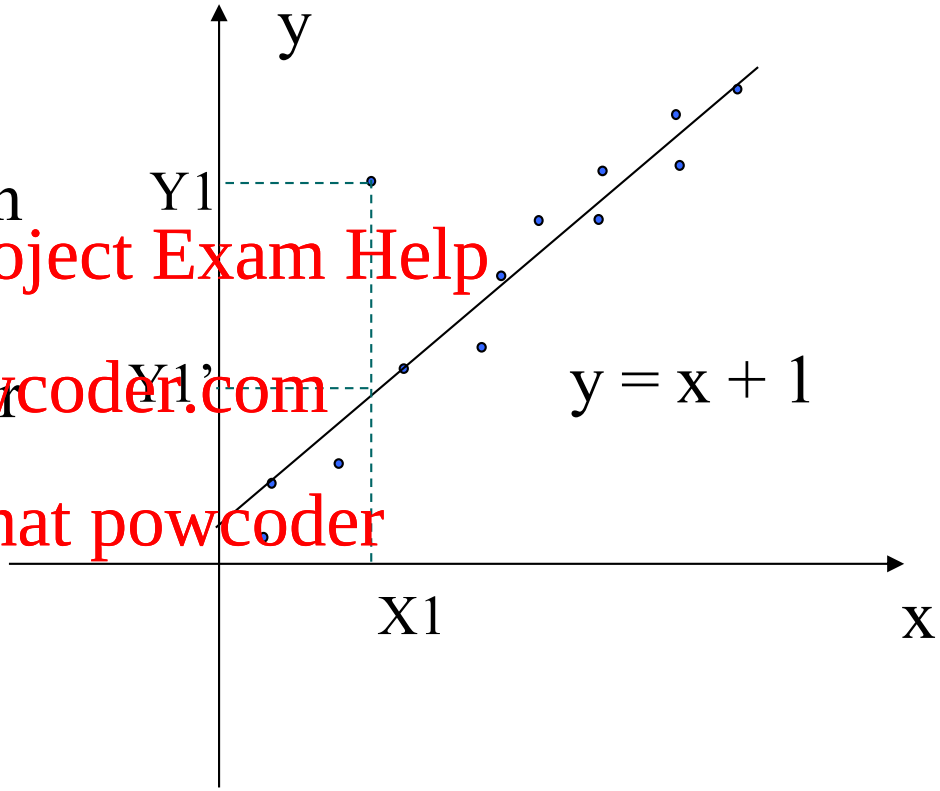
- Clustering
  - detect and remove outliers (An outlier is a value that does not follow the general pattern of the rest)
- Regression
  - smooth by fitting the data into regression functions
- Binning method:
  - first sort data and partition into bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Moving average
  - Use the arithmetic mean of neighborhood examples
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Cluster Analysis



Regression

- Fit the data to a function.
- Data points too far away from the function are outliers.
- A *single linear regression*, for instance, finds the line to fit data with 2 variables so that one variable can predict the other.
- More variables can be involved in *multiple linear regression*.



Binning:

- Equal-width (distance) partitioning:
  - It divides the range of an attribute into N intervals of equal size: uniform grid
  - if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward
  - But outliers may dominate presentation
  - Skewed data is not handled well.
- Equal-depth (frequency) partitioning:
  - It divides the range into N intervals, each containing approximately the same number of values
  - Good data scaling; better handle skewed data

Equal-width Binning Methods for Smoothing Data

* Sorted data for price (in dollars): 5, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
* Partition into (equi-width) bins: 3 intervals of equal size
  - Bin 1: 5, 8, 9
  - Bin 2: 15, 21, 21, 24
  - Bin 3: 25, 26, 28, 29, 34
* Smoothing by bin means:
  - Bin 1: 7, 7, 7
  - Bin 2: 20, 20, 20, 20
  - Bin 3: 28, 28, 28, 28, 28
* Smoothing by bin boundaries:
  - Bin 1: 5, 9, 9
  - Bin 2: 15, 24, 24, 24
  - Bin 3: 25, 25, 25, 25, 34

Equal-depth Binning Methods for Smoothing Data

* Sorted data for price (in dollars): 5, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34 (12 points in total)
* Partition into 3 (equi-depth) bins:
  - Bin 1: 5, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by bin boundaries:
  - Bin 1: 5, 5, 5, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Moving Average

- Use neighborhood values to smooth out noise
- Typically used for time-series data

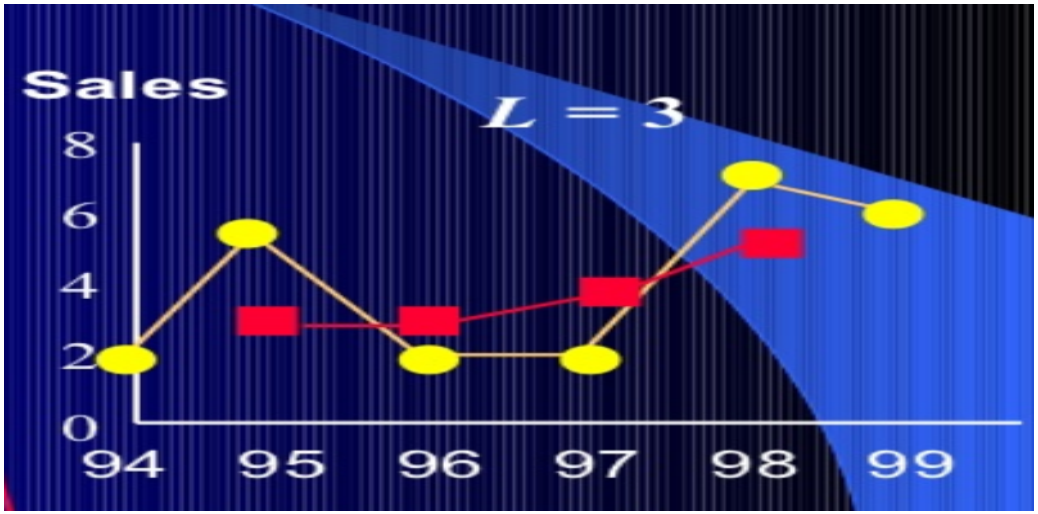
  - Use series of arithmetic means over time

  - Result depends on choice of length L for computing mean.

- Can also be used on spatial data such as images

Moving Average Example

Year	Sales	Moving Average
1994	2	NA
1995	5	3
1996	2	3
1997	2	3.67
1998	7	5
1999	6	NA



Outline

- Why preprocess the data?
- Data integration
- Data cleaning
- Data transformation
- Data reduction
- Discretization

Data Transformation

- Transform the data into appropriate form for mining
- Attribute/feature construction
  - New attributes constructed from the given ones
    - e.g., compute average sale amount using total sale amount divided by units sold.
- Normalization: scale attribute values to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Discretization
  - Transform numeric attributes into symbolic attributes

Data Transformation: Normalization

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

where min_A and max_A are the minimum and maximum values of attribute A, and $[new\_min_A, new\_max_A]$ is the new range

- Example: Attribute *income* has values

  - $12,000, $20,000, $25,000, $30,000, $45,000, $60,000, $73,600, $98,000
  - normalized into values in range [0, 1]:

    0, 0.093, 0.151, 0.209, 0.384, 0.558, 0.716, 1

- Problems:
  - "Out of bounds" error occurs if a future input case falls outside the original range for A
  - A too big or too small value could be noise. If they are used as min or max value for normalization, the results are not reliable.

Data Transformation: Normalization (Contd.)

- z-score normalization

$$v' = \frac{v - mean_A}{s_A}$$

where $mean_A$ is the mean of attribute A and s_A is the standard deviation of A (suppose values are : $v_1, v_2, \ldots, v_n$):

$$s_A = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(v_i - mean_A)^2}$$

- Example:
  - Attribute income has values:
    - \$12,000, \$20,000, \$25,000, \$30,000, \$45,000, \$60,000, \$73,600, \$98,000
  - The mean and standard deviation of the attribute *income* are
    - \$45,450 and \$29,735, respectively
  - With z-score normalization, the values are transformed into:

    -1.12, -0.86, -0.69, -0.52, -0.02, 0.49, 0.95, 1.77

- Advantages:
  - useful when the actual min and max are unknown
  - better deal with outliers than min-max normalization

Data Transformation: Normalization (Contd.)

- Normalization by decimal scaling

$$v_i' = \frac{v_i}{10^k}$$

where k is the smallest integer such that $\text{Max}(|v_i'|) \leq 1$

- Example:
  - Suppose the recorded values of A range from -986 to 97
  - The maximum absolute value of A is 986.
  - Then k= 3
  - -986 is normalized to -0.986 and 97 is normalized to 0.097

Outline

- Why preprocess the data?
- Data integration
- Data cleaning
- Data transformation
- Data reduction
- Discretization

Data Reduction

- What is data reduction?
  - A preprocessing step before applying learning or mining techniques to the data
  - Purpose: reduce the size of data
- Why data reduction?
  - A data set may be too large for a learning program. The dimensions exceed the processing capacity of the program.
  - The expected time for inducing a solution may be too long. Trade off accuracy for speed-up.
  - Sometimes, better answers are found by using a reduced subset of the available data. Too large data may cause the program to fit too many exceptions.

Data Reduction Operations

- Standard data form

Case/Example	$feature_1$	…	$feature_k$	*Class*
e_1	$V_{1,1}$	…	$V_{1,k}$	c_1
…	…	…	…	…
e_i	$V_{i,1}$	…	$V_{i,k}$	c_i
…	…	…	…	…
e_n	$V_{n,1}$	…	$V_{n,k}$	c_n

- Data reduction operations
  - Feature reduction (reduce the number of columns)
  - Case reduction (reduce the number of rows)
  - Value reduction (reduce the number of distinct values in a column)

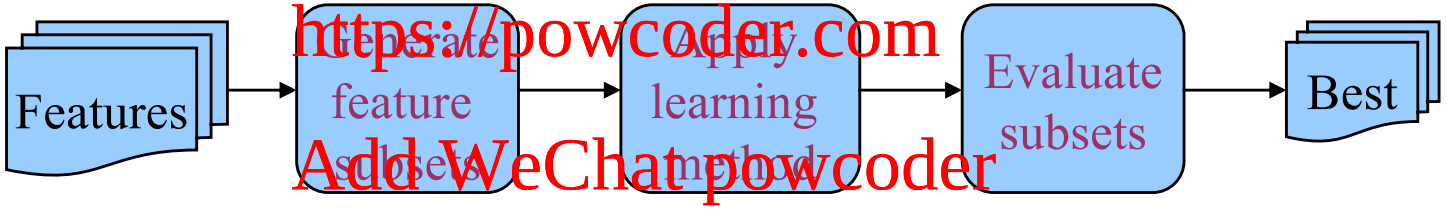
Types of Attributes (Features)

- Three types of attributes:
  - Nominal (symbolic, categorical) — values from an unordered set
    - Eg: {red, yellow, blue, ….}
  - Ordinal — values from an ordered set
    - Eg: {low, medium, high}
  - Continuous — real numbers
    - Eg: {-9.8, 3.9, 8.7, 19.1}

- Next: feature selection for classification tasks.

Feature Selection

- Objective
  - Find a subset of features with predictive performance comparable to the full set of features.
- An optimal subset selection



- A *practical objective* is to remove clearly extraneous features - leaving the table reduced to manageable dimensions - not necessarily to select the optimal subset.

Feature Selection Methods

- Filter Methods: select a subset of original features.
  - Feature Selection from Means and Variances (√)
  - Feature Selection by Mutual Information (√)
  - Feature Selection by Decision Trees (√)
  - Feature Selection by Rough Sets, etc.
- Merger Methods: merge features, resulting in a new set of fewer columns with new values.
  - Principal component analysis (PCA)
- Wrapper Methods: feature selection is being "wrapper around" a learning algorithm.
  - This is the optimal method in the last slide.
  - Running time is long; infeasible in practice if there are many features.

Feature Selection from Means and Variances

- Principle
  - Compute the means of a feature for each class, normalized by the variances;
  - If the means are far apart, interest in a feature increases (the feature has potential in terms of distinguishing between classes);
  - If the means are indistinguishable, interest wanes in that feature.
- Two intuitive methods
  - Independent feature analysis (√)
    - Assuming the features are independent. Features are examined individually.
  - Distance-based feature selection
    - Features are examined collectively.
- Limitation: only applied to continuous features.

Independent Feature Analysis

- For a problem with two classes: C_1 and C_2 :
  - Compute $mean_1(f)$ and $mean_2(f)$: the means of feature f measured for C_1 and C_2
  - Compute $var_1(f)$ and $var_2(f)$: the variances of feature f measured for C_1 and C_2
  - Significance test (t-test):

$$|mean_1(f) - mean_2(f)| > sig \times \sqrt{\frac{\text{var}_1(f)}{n_1} + \frac{\text{var}_2(f)}{n_2}}$$

    - n_1 and n_2 are the numbers of cases in C_1 and C_2
    - $sig = 2$ for the 95% confidence level.

  - If the comparison fails the test, the feature can be deleted.

- For k classes, k pairwise comparisons are conducted for f.
  - Each pairwise comparison compares feature means for class C_i and $\neg C_i$ ($i=1, \ldots, k$).
  - A feature is retained if it is significant for at least one of the pairwise comparisons.
- Limitations:
  - Only applies to numeric attributes
  - Treat each feature independently

Feature Selection by Mutual Information

- Objective: Select features according to the mutual information between a feature and the class variable.
- The mutual information (also called information gain) between the class variable y and a discrete feature x :

$$MI_x = \sum_v \sum_c [P(y=c, x=v) \times \log_2 \frac{P(y=c, x=v)}{P(y=c)P(x=v)}]$$

  - $P(y=c)$ is the probability of cases in class c.
  - $P(x=v)$ is the probability that feature x takes on value v.

- MI measures the degree to which x and y are not independent. The bigger the value, the more dependent y is on x.

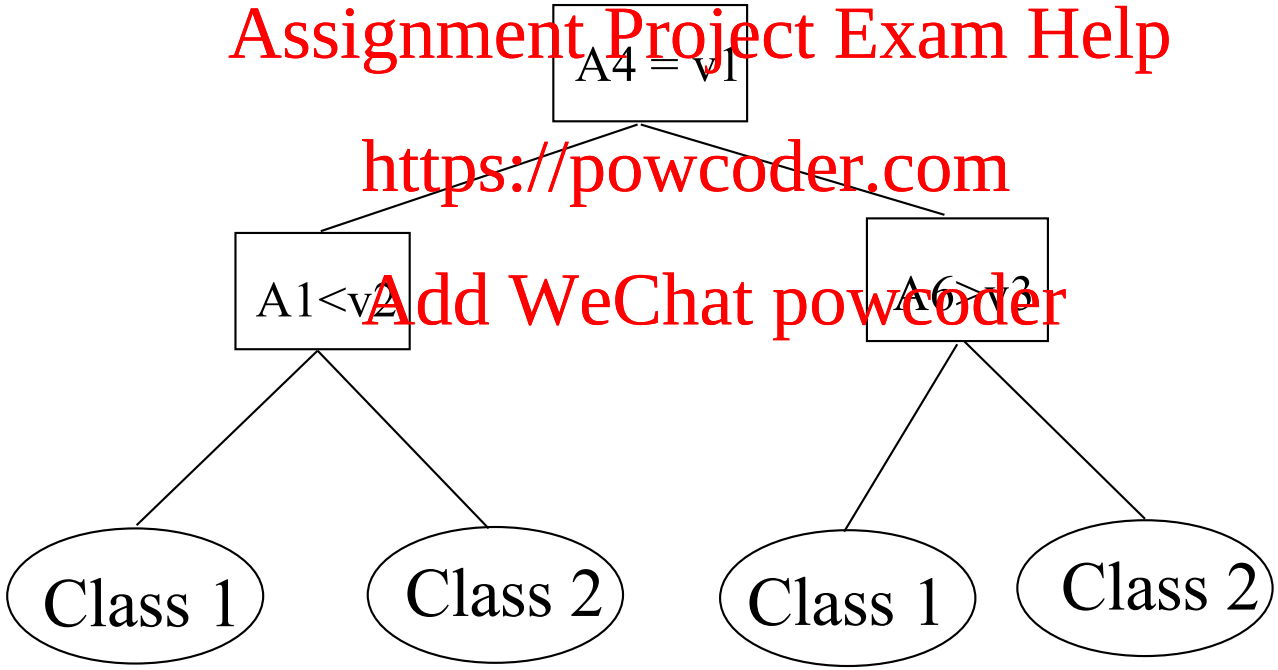
- MI is used to select or weight features.
  - You can select the top k features with the highest weights. Or some mining algorithms can take the feature weights and select features in the mining process.
- Suitable for nominal or discrete attributes.
  - For continuous features, a discretization algorithm can be applied first to convert a real-valued feature to a discrete-valued feature.
- Limitation: Treat each feature independently.

Feature Selection by Decision Trees

- Objective
  - Decision tree learning methods integrates feature selection to their algorithms and decision tree is a fast learning method.
  - Make use of the decision tree learning technique to select features from a data set for other learning methods, such as neural networks, that take substantially more time to search their solution space.
    - Decision tree learning is a relatively fast learning method.
- Method
  - Apply a decision tree learning algorithm to the data set to generate a decision tree.
  - Select features that appear in the tree.
- Advantage
  - *Context sensitive*. Tree methods evaluate candidate features in the context of related features that have already been selected.

Example of Feature Selection Using Decision Tree

Initial attribute set:
{A1, A2, A3, A4, A5, A6}



------> Reduced attribute set: {A1, A4, A6}

Data Reduction Outline

- Feature Selection
- Case Reduction
- Value Reduction

Case Reduction

- Objective: reduce the number of cases, the largest dimension in the data set
- How many cases are enough?
  - Application dependent - depends on the complexity of the patterns to be extracted from the data.
    - If the pattern is simple, the results are unlikely to change even with additional cases. For example, x>1 completely separates two classes.
    - For complex patterns, large volumes of data can supply more evidence for the correctness of the induced patterns.
- Some types of problems requiring more data than others:
  - Multiclass classification
  - Regression
  - Imbalanced data sets: almost all cases belong to the larger class, and far fewer cases to the smaller, usually more interesting class.

Case Reduction Methods

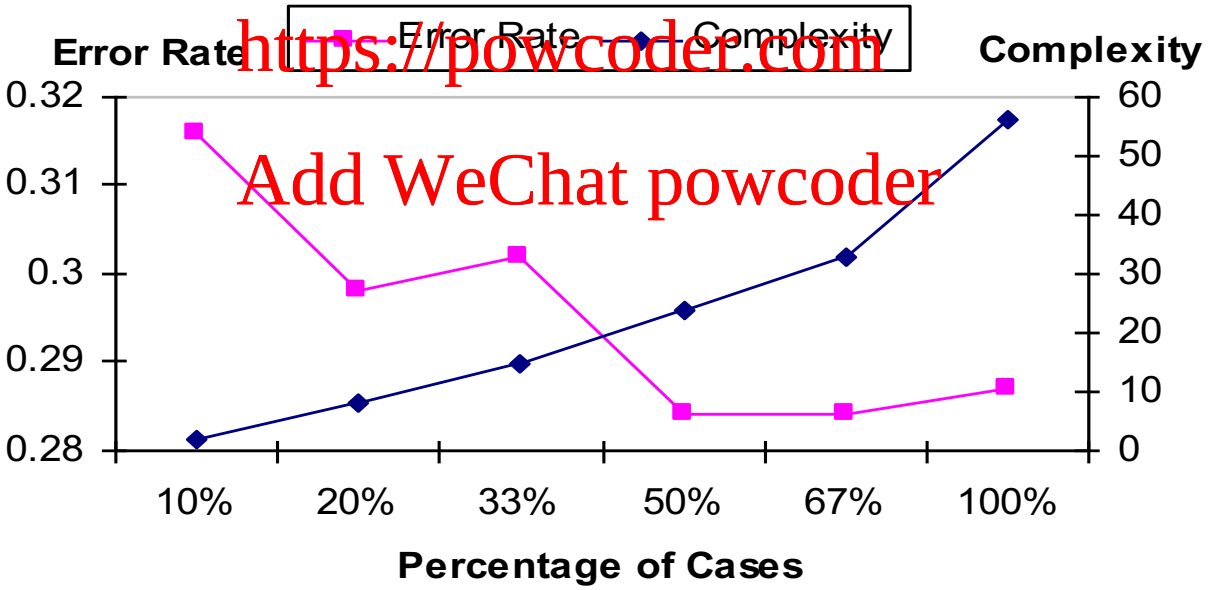
- Simple Random Sampling
  - A single sample
  - Incremental samples
  - Average samples
- Sampling by Adjusting Prevalence
- Stratified Sampling

Single Simple Random Sample

- Choose n objects randomly from a set D of N objects ($n<N$) so that each object has the same probability of being chosen.
- Two methods
  - Simple random sampling without replacement (SRSWOR)
    - Each object cannot be chosen more than once
  - Simple random sampling with replacement (SRSWR)
    - Each time an object is drawn, it is recorded and placed back to D so that it can be drawn again.

Problem with Single Sampling

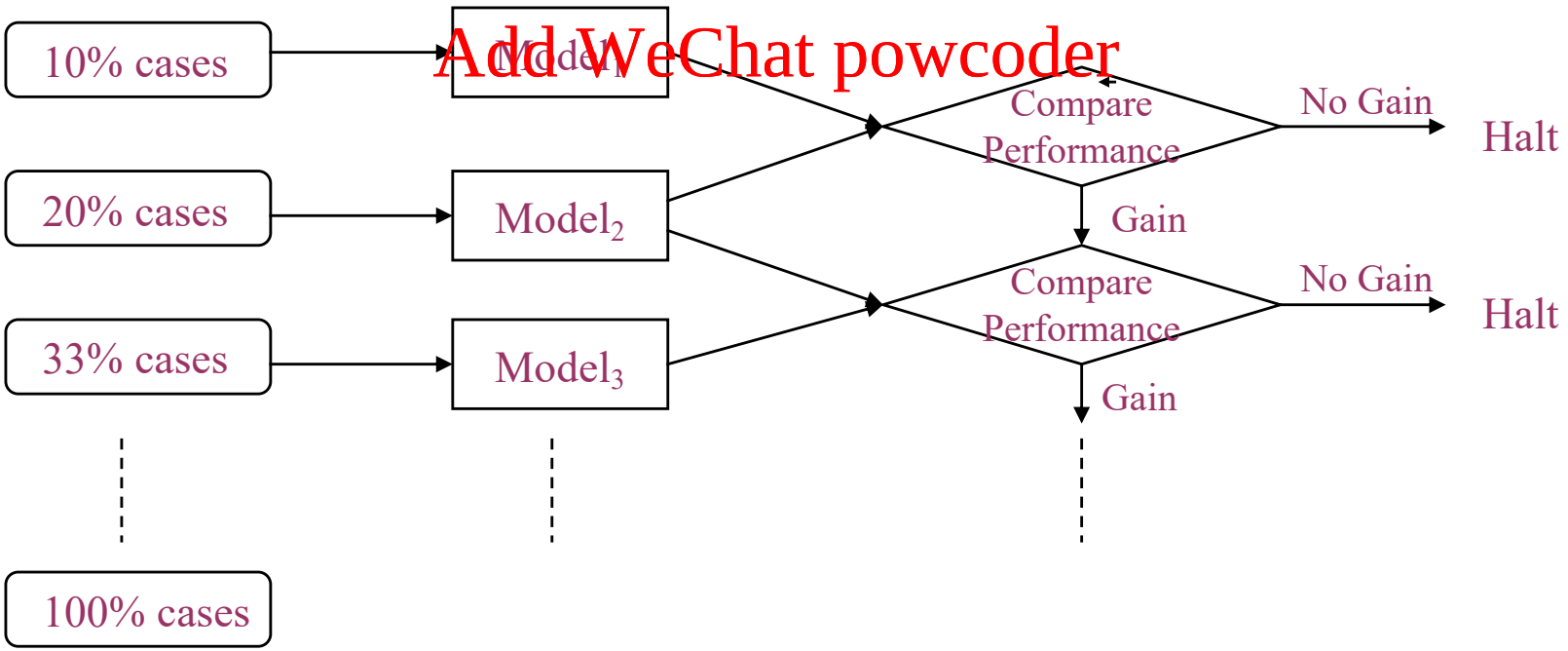
- We don't know the suitable sample size.
- Too small, the model may not be accurate enough; too big, the model may be more complex



Incremental Sampling

- Objective: Spot trends in error and complexity by *learning with incrementally larger random subsets of the data* to help produce a single solution.
- A typical pattern of incremental subsets is:

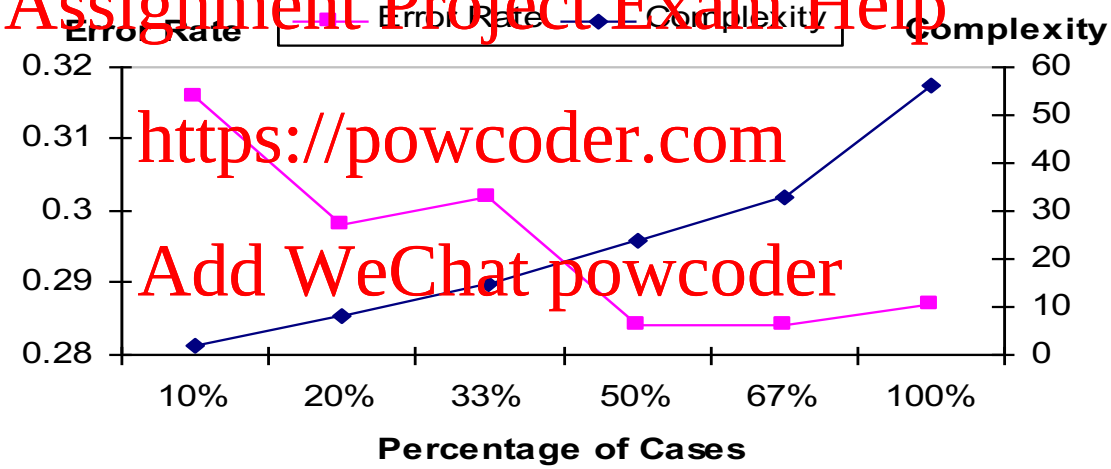
10%, 20%, 33%, 50%, 67%, 100%



Incremental Sampling (Cont.)

- Performance measures:
  - Error rate (test error, that is error on a test data set)
  - Complexity of the solution (e.g. number of nodes in a tree)
- Plot error and complexity relative to increasing sample size.

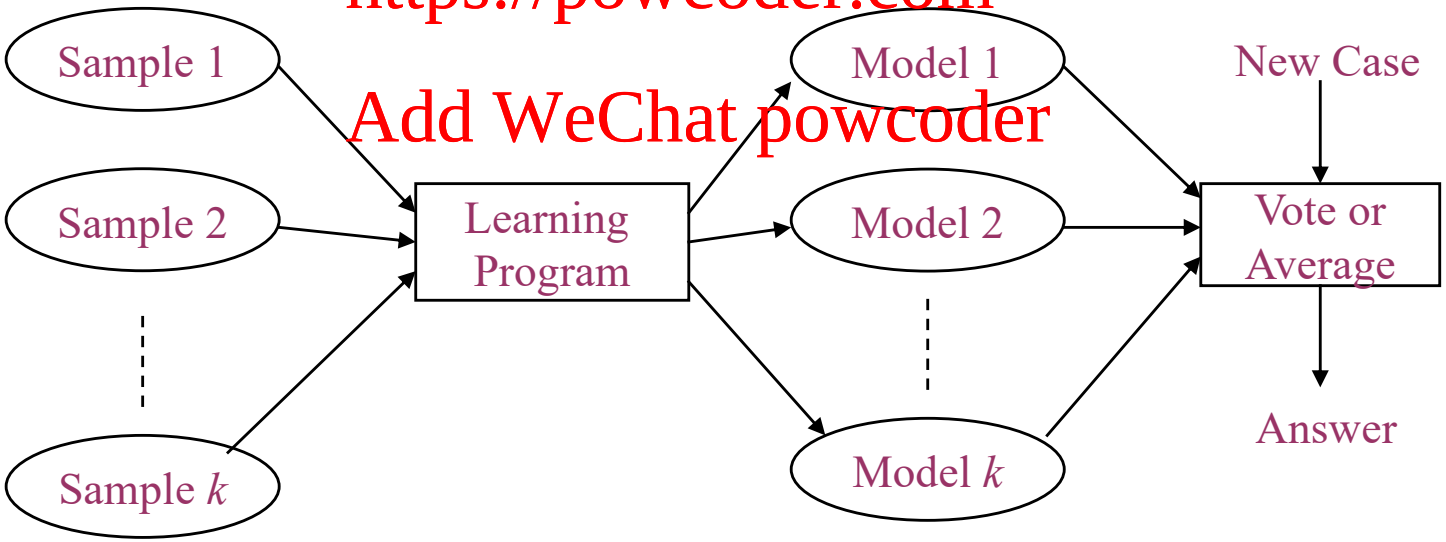
Example:



- Make decision on whether to do further sampling
  - Net changes in error and complexity are examined:
    - Is the error smaller?
    - Is the complexity acceptable?
    - Is complexity increasing much more than error is decreasing?

Average Sampling

- For a dataset containing a huge number of cases that exceed the maximum capacity of a learning program.
- Average sampling:
  - Select *k* random samples of *n* cases
  - Solutions from different samples are combined in the prediction phase.



- Averaged or voted solutions usually have less error than the single solution found on all cases in the database.

Sampling by Adjusting Prevalence

- Directly adjust the prevalence of cases over the classes.

- Suitable for classification problems with a very imbalance data set
  - In a bio-chemistry data set for predicting biological potency of chemical compounds,
    - Only 0.16% of the compounds belong to the class of highly active compounds, which is the most interesting class that can lead to discovery of new drugs.
    - Remaining 99.84% of compounds are inactive.
  - Low-prevalence class, usually the most interesting class.

Sampling by Adjusting Prevalence

- Two ways for boosting prevalence:
  - *Up-sampling*: repeat (or give higher weights to) the cases in the low-prevalence class in the sample - increase the sample size.
  - *Down-sampling*: keep the low-prevalence cases intact or randomly sample a high percentage of them, while including a low percentage random subset of a larger class in the training sample.
- Result: the predictive performance on the most interesting new cases may increase, while the overall predictive performance on new data of all classes may decrease.

Stratified Sampling

- The data set *D* is partitioned into mutually disjoint subsets, called *strata*.

- Then randomly sample data from each stratum

- Objective: ensure a representative sample, especially when the data are skewed.

Data Reduction Outline

- Feature Selection
- Case Reduction
- Value Reduction

Reducing and Smoothing Values

- Objective
  - Reduce the number of distinct values of a feature so that the size of the search space for patterns is reduced.
  - Smooth out noise
- Methods for reducing values
  - Nominal attributes
    - Generalization.

Toronto → Ontario → Central Canada → Canada

Reducing and Smoothing Values (Cont'd)

- Integer or real-valued attributes
  - Rounding
    - e.g. 462.4 can be rounded to 462, 460, or 500 according to requirements
  - Binning
    - Partition the value range of an attribute into bins

1, 1, 2, 3, 3, 3, 4, 5, 5, 7

bin1 bin2 bin3

    - Smooth values by bin medians, means or boundaries

1, 1, 1, 3, 3, 3, 5, 5, 5, 5

bin1 bin2 bin3

1, 1, 2, 3, 3, 3, 4, 4, 4, 7

bin1 bin2 bin3

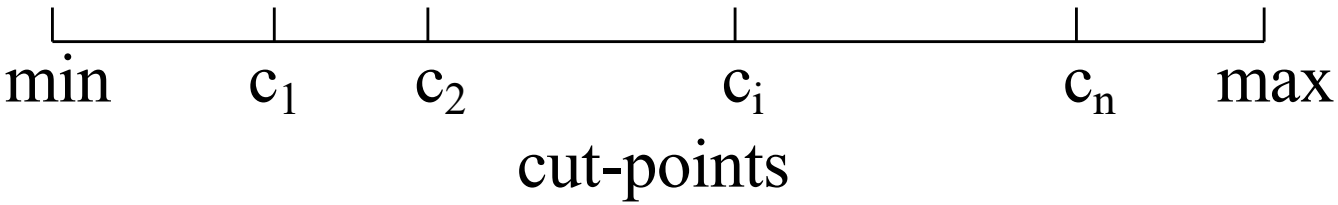
  - Discretization: label each bin by discrete values

Data Preprocessing

- Why preprocess the data?
- Data integration
- Data cleaning
- transformation
- Data reduction
  - Feature Selection
  - Case Reduction
  - Value Reduction
- Discretization

What is discretization?

- A discretization algorithm
  - converts continuous attributes into discrete attributes by partitioning the range of a continuous attribute into intervals.
  - Interval labels can then be used to replace actual data values.



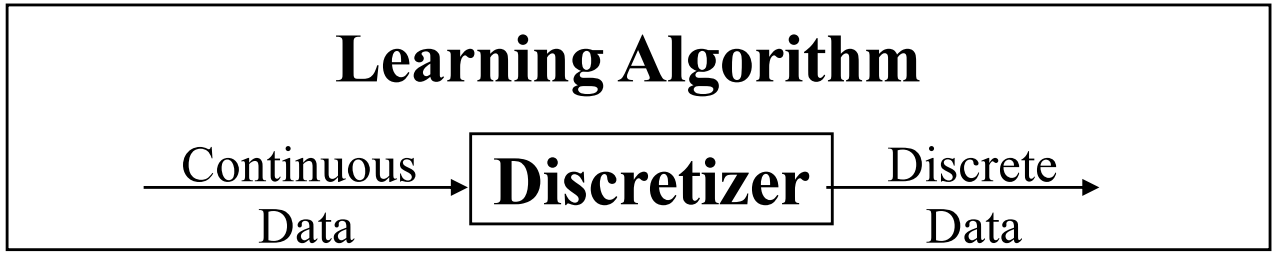
Why Need Discretization?

- Some learning algorithms are limited to discrete inputs.
- Efficiency: handling (lots of) continuous values tends to slow down learning considerably. (*Value reduction*)
- Accuracy: in the presence of noise good discretization can sometimes improve predictive accuracy. (*Smoothing out noise*)
- Intelligibility: discretization may lead to smaller sizes of induced trees or rule sets.

Two Architectures

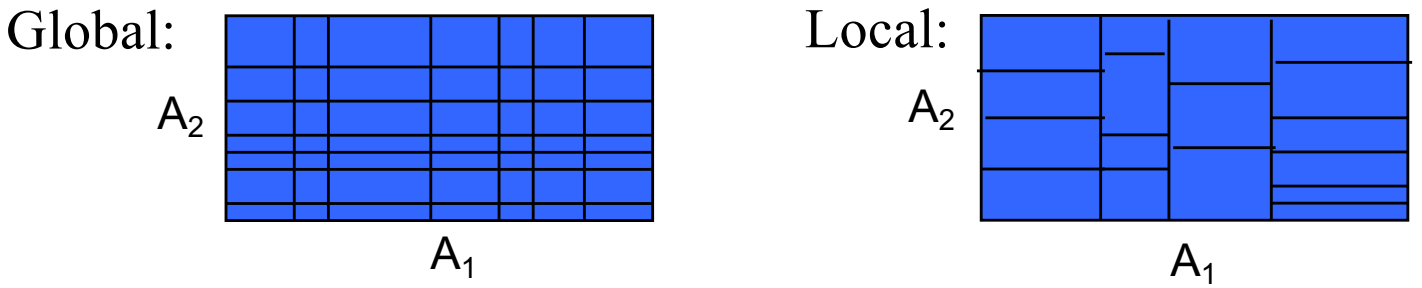
- Discretization before learning starts (Static discretization)

- Discretization during the learning process (Dynamic discretization)



Classification of Discretization Methods

- Supervised vs. unsupervised.
  - Supervised discretization uses class information.
  - Unsupervised does not use class labels.
- Bottom-up vs. top-down
  - Bottom-up: start from many intervals with one value each and repeatedly merge intervals until some stopping criterion is satisfied.
  - Top-down: start from one interval with all values and repeatedly split intervals until some stopping criterion is satisfied.
- Global vs. local
  - Global: an attribute is partitioned over the entire continuous range, using global information and independent of other attributes.
  - Local: partition is applied to local regions of an attribute range.



Unsupervised Discretization

- Equal-width binning
  - Use discrete values, such as 1, 2, 3, …, to represent intervals instead of bin means or boundaries
- Equal-depth/frequency binning
  - Use discrete values, such as 1, 2, 3, …, to represent intervals instead of bin means or boundaries
- k-means clustering
  - Given k bins, distribute the values in the bins to minimize the average distance of a value from its bin mean.

K-mean Clustering

- Input: (1) a set of values for an attribute
  (2) k = number of bins
- Sort the input values and keep the unique values
- Create k bins using equal-depth binning
- Compute bin means ($mean_1$, $mean_2$, ……, $mean_k$)
- Compute global distance: $D_{new} = \sum_i \sum_j (v_{ij} - mean_i)^2$

  where $mean_i$ is the mean in bin_i and v_{ij} is the jth value in bin_i.
- Repeat
  - $D_{old} = D_{new}$
  - for each bin_i
    - for each v_{ij} in bin_i
      - If ($|v_{ij}$- $mean_{i-1}|$) < ($|v_{ij}$-$mean_i|$), move v_{ij} to bin_{i-1}.
      - If ($|v_{ij}$- $mean_{i+1}|$) < ($|v_{ij}$-$mean_i|$), move v_{ij} to bin_{i+1}.
  - Compute new bin means and D_{new}
- Until D_{new} is not less than D_{old}.

Supervised Discretization

- ChiMerge
  - Based on chi-square test
- Entropy-based discretization method
  - Based on an entropy minimization heuristic

ChiMerge: a Bottom-up Supervised Method

- ChiMerge is based on the statistical χ^2 test
- The purpose of a χ^2 test is to determine whether two variables are related.
  - E.g., we want know if there is any relationship between the gender of undergraduate students in a university and their footwear preferences.
- Observations about the two variables in a sample are usually expressed in a contingency table:

	Sandals	Sneakers	Leather shoes	Boots	Other	Total
Male	6	17	13	9	5	50
Female	13	5	7	16	9	50
Total	19	22	20	25	14	100

Chi Square Significance Test

- The null hypothesis is that the two variables are unrelated (that is, only randomly related).
- χ^2 test determines whether we should reject the null hypothesis and *at what significance level* (***p-value***) we should reject the null hypothesis.
- For the example in the previous slide,
  - The null hypothesis is that gender is unrelated with footwear preference
  - But the χ^2 test shows that we should reject this hypothesis at the significance level of 0.01, which means that we are 99% sure that gender and footwear preferences are related.
  - Usually, p-value should be at most 0.05 in order to reject the null hypothesis.

How to Calculate χ^2

- Given the contingency table:

	Sandals	Sneakers	Leather shoes	Boots	Other	Total
Male	6	17	13	9	5	50
Female	13	5	7	16	9	50
Total	19	22	20	25	14	100

- Compute the expected frequency for each cell
  - The expected frequency of $\text{cell}_{i,j}$ is

$$E_{ij} = \frac{\text{the total of row i} \times \text{the total of column j}}{\text{sample size}}$$

  - For example, the expected frequency of the upper left cell is $\dfrac{50 \times 19}{100}$

How to Calculate χ^2 (Cont'd)

- Compute the chi-square value for the table

	Sandals	Sneakers	Leather shoes	Boots	Other	Total
Male	6	17	13	9	5	50
Female	13	5	7	16	9	50
Total	19	22	20	25	14	100

- Let O_{ij} denote the observed value in cell$_{i,j}$

$$\chi^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

- For example, the chi-square value of the above table is 14.026

How to Calculate χ^2 (Cont'd)

- Calculate the degrees of freedom for the table

	Sandals	Sneakers	Leather shoes	Boots	Other	Total
Male	6	17	13	9	5	50
Female	13	5	7	16	9	50
Total	19	22	20	25	14	100

$$df = (r-1)(c-1)$$

- where *r* is the number of rows and *c* is the number of columns
- For example, the degrees of freedom for the above table is 4.
- This is because, given row or column totals, all but one of the values in a given row or column are free to vary.

How to Calculate χ^2 (Cont'd)

- Using the chi-square table to determine the p-value for rejecting the null hypothesis

df	P = 0.05	P = 0.01	P = 0.001
1	3.84	6.64	10.83
2	5.99	9.21	13.82
3	7.82	11.35	16.27
4	9.49	13.28	18.47
5	11.07	15.09	20.52
…	…	…	…

  - The table lists the critical values (i.e., thresholds)
  - The calculated chi-square value for a contingency table must be greater than the critical value corresponding to the df of the table and a p-value (e.g., 0.05) in order to reject the null hypothesis at the significance level (p-value).

ChiMerge: a Bottom-up Supervised Method

- Sort all examples according to the values of the attribute to be discretized.
- Place each value in its own interval.
- Merge intervals repeatedly in the following manner:
  - For each pair of adjacent intervals:
    - Calculate the χ^2 value:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

      where k = # of classes, O_{ij}= # of examples in the ith interval and jth class, E_{ij}= expected frequency of O_{ij}= $\frac{R_i \times C_j}{N}$, in which N is # of examples, R_i=# of examples in the *ith* interval, and C_j= # of examples in the *jth* class.
  - If the lowest χ^2 value is smaller than a threshold, merge the two adjacent intervals with the lowest χ^2 value.
  - This process is repeated until all χ^2 values exceeds this threshold.
- The threshold can be obtained from the standard χ^2 table

Entropy-Based Discretization

- Supervised, top-down discretization
- Employs an entropy minimization heuristic for splitting the range of a continuous attribute.

- Given a set *S* of examples and *k* classes, the *entropy* of *S* with respect to the *k* classes is defined as:

$$Ent(S) = -\sum_{i=1}^{k} P(C_i)\log_2(P(C_i))$$

where $P(C_i)$ is the probability of examples in S that belong to C_i.

- The bigger *Ent(S)* is, the more impure *S* is.

Entropy-Based Discretization

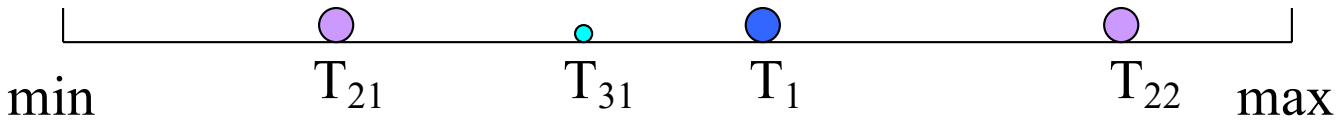
Given an attribute A and a set S of training examples:

- Sort the examples in a set S by increasing values of the attribute A: $\{v_1, v_2, \ldots, v_n\}$.
- A potential cut-point T: midpoint between v_i and v_{i+1} dividing S into S_1: $\{v_1, v_2, \ldots, v_i\}$ and S_2: $\{v_{i+1}, \ldots v_n\}$.
- A total of n-1 potential cut-points.
- Suppose a cut-point T partitions S into S_1 and S_2. Entropy (with respect to the class attribute) after the partition induced by cutpoint T:

$$Ent(T,S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

where $|S|$, $|S_1|$ and $|S_2|$ = # of examples in S, S_1 and S_2

- Select T_A for which $E(T_A, S)$ is minimal to split the range into two subranges
- The process is recursively applied to partitions obtained until some stopping criterion is met.



Stopping Criteria for Entropy-Based Discretization

- Stopping criteria in D-2 (Catlett, 1991):

  Recursive partitioning stops if any of the following is satisfied:

  - all the examples in the interval belong to the same class.
  - number of examples in an interval is below a given level;
  - maximum number of cut-points for an attribute is reached;
  - the entropy reduction on all possible cut-points is equal;

- Stopping criterion based on Minimum Description Length Principle (MDLP) (Fayyad and Irani, 1993):

  Recursive partitioning stops iff

$$Ent(S) - Ent(T,S) \le \frac{\log_2(N-1)}{N} + \frac{\Delta(T;S)}{N}$$

$$\Delta(T;S) = \log_2(3^k - 2) - [kEnt(S) - k_1 Ent(S_1) - k_2 Ent(S_2)]$$

  where T is the cut point leading to the smallest $Ent(T,S)$, k, k_1 and k_2 are the number of classes in S, S_1 and S_2, respectively, and N is the number of examples in S.

Summary

- Data preparation is a big issue for data mining
- Data preparation includes
  - Data integration
  - Data cleaning
    - Handle missing values
    - Detect and remove noise
  - Data transformation
  - Data reduction
    - feature selection, case reduction and value reduction
  - Discretization
- A lot of methods have been developed but still an active area of research

Readings

- Chapter 3 in Jiawei Han's book
- Chapters 3 and 4 in "Predictive Data Mining, a Practical Guide" by Sholom M. Weiss and Nitin Indurkhya.
- U. M. Fayyad and K. B. Irani, "*Multi-interval discretization of continuousvalued attributes for classification learning*," Proc. of the 13th Int. Joint Conf. on Artificial Intelligence, pp. 1022--1027, Morgan Kaufmann, 1993.