

Data Wrangling and Data Analysis

Data Preparation (2)

Hakim Qahtan

Department of Information and Computing Sciences

Utrecht University



Utrecht University

Reading Material for Today

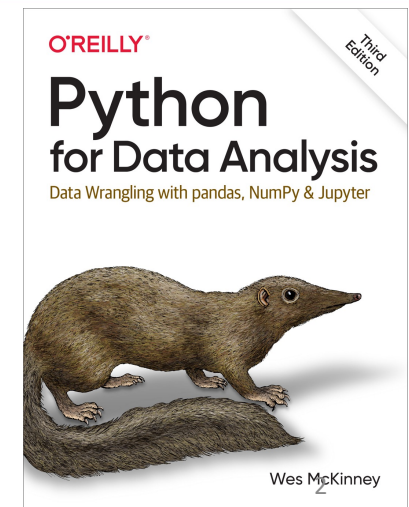
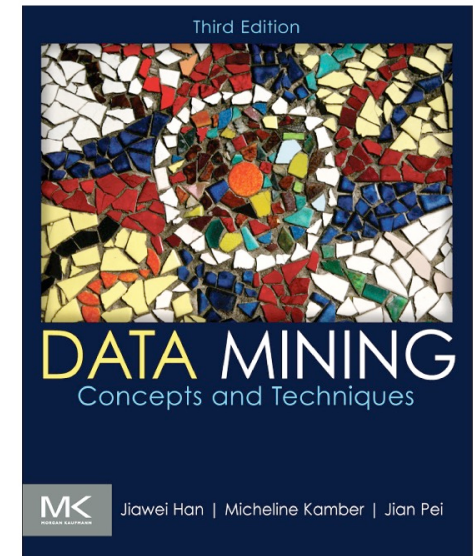
- Data Mining: Concepts and Techniques Book

CH 3.3 – 3.5

- Python for Data Analysis, 3E
CH 7.2



Utrecht University



Topics for Today

- Data transformation
- Data Integration
- Data reduction
- Data discretization



Data Transformation



Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values
- Methods for data transformation
 - Smoothing: Remove noise from data
 - Attribute/feature construction
 - New attributes constructed from the given ones
 - Aggregation: Summarization, data cube construction



Data Transformation (Cont.)

- Methods for data transformation
 - Normalization: Scaled to fall within a smaller, specified range
 - Min-max normalization
 - Z-score normalization
 - Normalization by decimal scaling
 - Data reformatting:
 - E.g. Jack Wilsher → Wilsher, J.
 - Use the same unit:
 - Records in inches and cm
 - Records with prices in Euros and Dollars



Data Normalization (Standardization)

The goal of standardization or normalization is to make an entire set of values have a particular property.

Data Normalization – Min-Max Normalization

- Transform the data from a given range with $[min_A, max_A]$ to a new interval $[new_min_A, new_max_A]$ for a given attribute A :

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

where v is the current value of attribute A .



Data Normalization – Min-Max Normalization

- Example:

Suppose that the minimum and the maximum in the attribute income are €12,000 and €98,000, respectively

We would like to map the income into the interval [0,1]

Using min-max normalization, a value of €73,600 for income is transformed into:

$$\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0.0) + 0.0 = 0.716$$



Data Normalization – Z-Score Normalization

- Transform the data by converting the values to a common scale with an average of zero and a standard deviation of one.
- A value, v , of attribute A is normalized to v' by computing:

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

where $\bar{A}$ and σ_A are the mean and standard deviation of attribute A , respectively.



Data Normalization – Z-Score Normalization

- Example:
 - Suppose that the mean and standard deviation of the values for the feature income are 54,000 and 16,000, respectively. With z-score normalization, a value of €73,600 for income is transformed to:

$$\frac{73,600 - 54,000}{16,000} = 1.225$$



Data Normalization – Decimal Scaling Normalization

- Transform the data by moving the decimal points of values of attribute A .
- The number of decimal points moved depends on the maximum absolute value of A .
- A value v of A is normalized to v' by computing: $v' = \frac{v}{10^j}$
where j is the smallest integer such that $\max(|v'|) < 1$



Data Normalization – Decimal Scaling Normalization

- Example:
 - Suppose that the recorded values of A range from -986 to 917 .
 - The maximum absolute value of A is 986 .
 - To normalize by decimal scaling, we divide each value by $1,000$ (i.e., $j = 3$) so that -986 normalizes to -0.986 and 917 normalizes to 0.917



Data Transformation (Cont.)

- Exercise
 - Use flash fill of Microsoft Excel to convert set of names from the format Family_name, First_name middle_initial. to First_name Family_name
 - E.g. Wilsher, John K. to John Wilsher



Data Integration (Revisited)



Data Integration

- Combines data from multiple sources into a coherent store
- Schema integration: e.g., $A.cust-id \equiv B.cust-No$
 - Integrate metadata from different sources
- Entity identification problem:
 - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- 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., metric vs. British units



Handling Redundancy when Integrating Data

- Handling data redundancy is an important task of the data integration
- Duplicate records can be identified by applying entity resolution techniques
- Redundant attributes can be detected by correlation and covariance analysis
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining quality



Entity Resolution

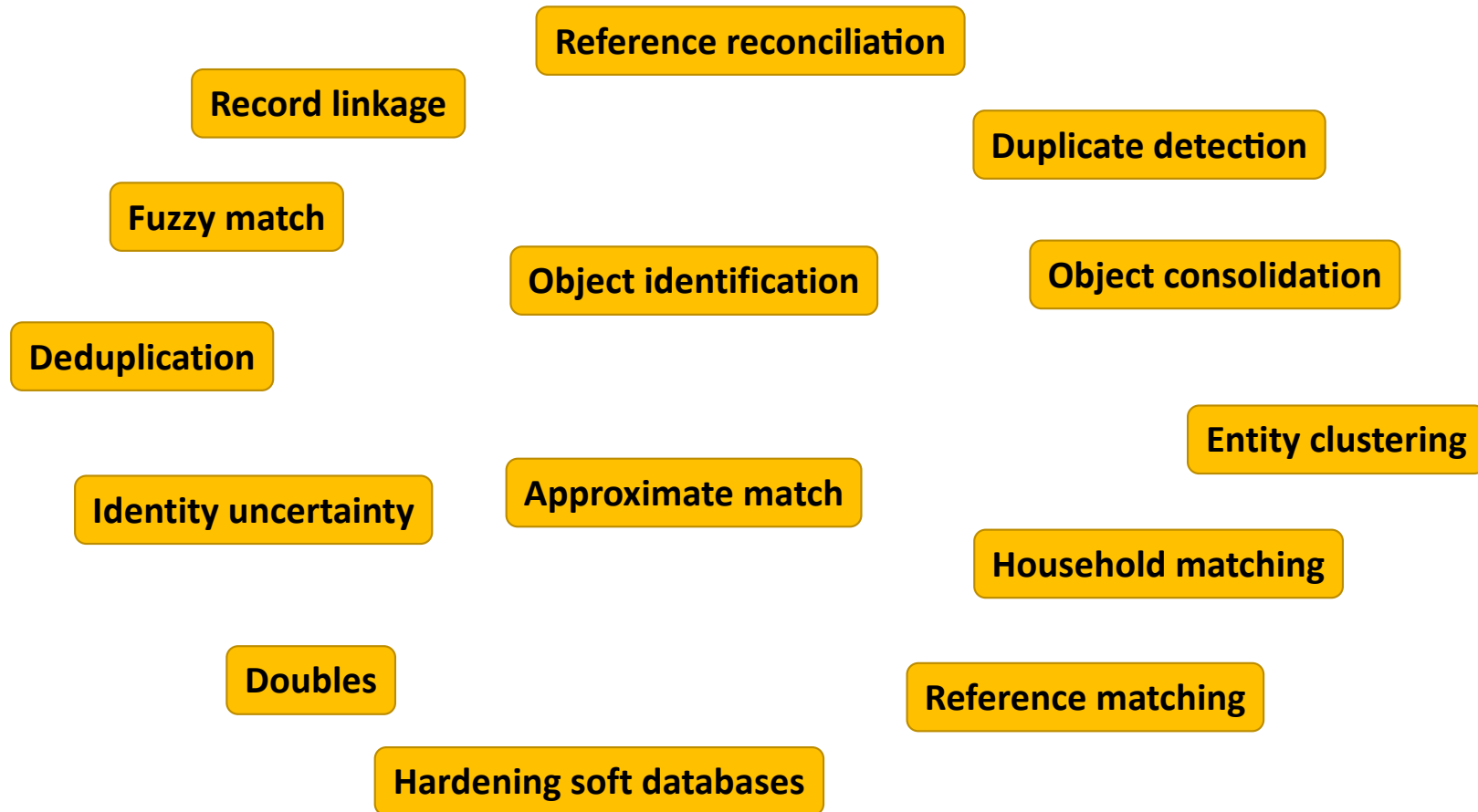
Problem of identifying and linking/grouping different representations of the same real-world object.

Examples:

- Different ways of addressing (names, email addresses, FaceBook accounts) the same person in text.
- Web pages with differing descriptions of the same business.
- Different photos of the same object.
- ...



Ironically, Entity Resolution has Many Duplicate Names



Entity Resolution – Normalization

- Schema Normalization
 - Schema matching: e.g., Contact No. vs. Phone
 - Compound attributes: e.g., address vs. (street, city, zip)
- Data Normalization
 - Capitalization, white-space normalization
 - Correcting typos, replacing abbreviations, variations, nick names
 - Usually done by employing dictionaries



Entity Resolution – Matching Features

- Given two records, compute a vector of similarity scores for corresponding features
- E.g., to match two bibliographical references, compute:
 - First author match score
 - Title match score
 - Venue match score
 - Year match score
 - ...
- Score can be Boolean (match/mismatch) or a continuous value based on specific similarity measure (distance function).



Entity Resolution – Similarity Measures

- Atomic similarity measures
 - Difference between numerical values
 - Jaro for comparing names
 - Edit distance for typos
 - Phonetic-based
 - Soundex
- Set similarity
 - Jaccard similarity
- Vector-based
 - Cosine similarity



Entity Resolution – Issues

- Similarity measures has different scales
 - How to combine the different values to compare the tuples
- Pairwise similarity between records is expensive
 - Takes $O(n^2)$
- Blocking to reduce the quadratic time complexity
 - Divide the records into blocks
 - Perform pairwise comparison between records within the same block only
 - When $\#blocks = k$ and block size $\approx (n/k)$, the time complexity $O(k(n/k)^2)$



Data Reduction



Data Reduction

- **Data reduction:** obtain a reduced representation of the dataset
 - Much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction?
 - A dataset could be extremely large – Complex data analysis may take a very long time to run on the complete dataset.
- Data reduction strategies (attribute level)
 - Dimensionality reduction, e.g., remove unimportant attributes
 - Principal Components Analysis (PCA)
 - Singular Value Decomposition (SVD)
 - Feature subset selection, feature creation



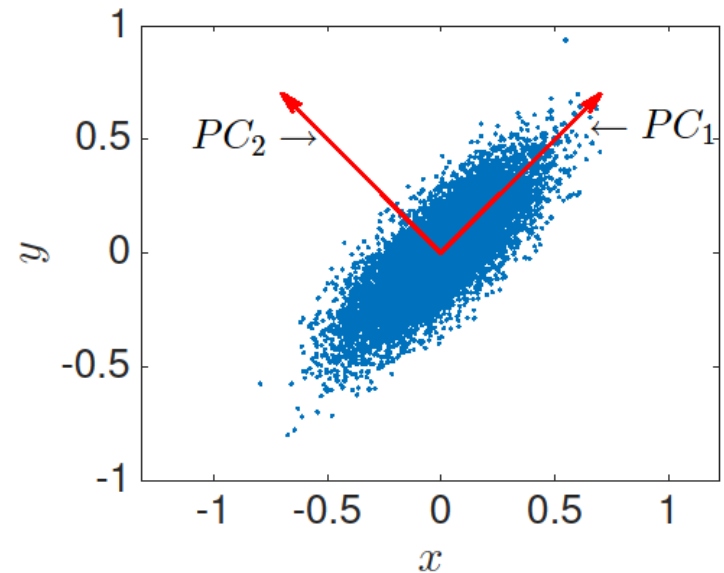
Dimensionality Reduction

- Curse of dimensionality
 - When dimensionality increases, data becomes increasingly sparse
 - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
 - The possible combinations of subspaces will grow exponentially
- Dimensionality reduction
 - Avoid the curse of dimensionality
 - Help eliminate irrelevant features and reduce noise
 - Reduce time and space required in data mining
 - Allow easier visualization



Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction.
- We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



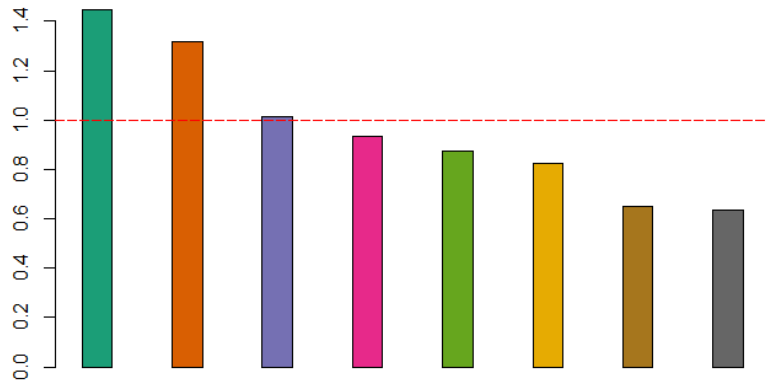
Principal Component Analysis (Steps)

- Given: N data vectors from d -dimensions, find $k \leq d$ principal components that can accurately represent the data
 - Normalize input data: each attribute falls within the same range
 - Compute k orthonormal (unit) vectors, i.e., principal components
 - Each input data (vector) is a linear combination of the k principal component vectors
 - The principal components are sorted in order of decreasing “significance” or strength
 - The components are sorted
 - Reduce the data dimensionality by eliminating the weak components
 - Weak components have low variance
 - We can reconstruct a good approximation of the original data using strong components
- Works for numeric data only



PCA for Dimensionality Reduction

- Can ignore the components of lesser significance.



- You **lose some information**, but if the eigenvalues are small, it is not much
 - d dimensions in original data
 - calculate d eigenvectors and eigenvalues
 - choose only the first k eigenvectors, based on their eigenvalues (eigenvectors with eigenvalues greater than 1 are considered important)
 - final data set has only k dimensions



Attribute Subset Selection for Data Reduction

- Another way to reduce dimensionality of data
- Redundant attributes
 - Duplicate much or all of the information contained in one or more other attributes
 - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
 - Contain no information that is useful for the data mining task at hand
 - E.g., students' ID is often irrelevant to the task of predicting students' GPA



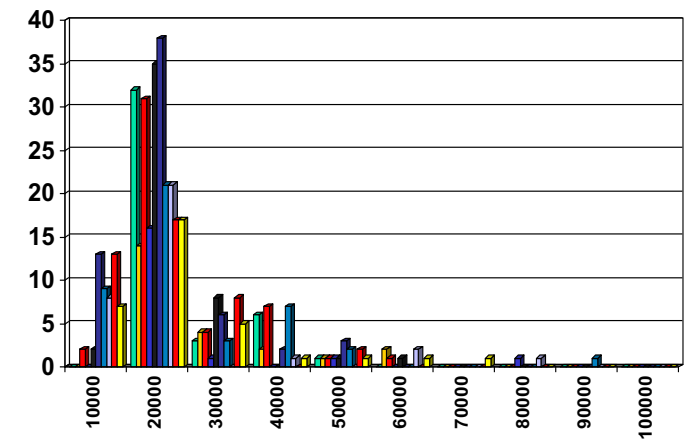
Data Reduction (instances or examples level)

- Linear regression
 - Data modeled to fit a straight line
 - Often uses the least-square method to fit the line
- Multiple regression
 - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model
 - Approximate data by a function whose logarithm is linear



Histograms for Instance Level Data Reduction

- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
 - Equal-width: equal bucket range
 - Equal-frequency (or equal-depth)



Clustering-Based Data Reduction

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- There are many choices of clustering definitions and clustering algorithms
- Clustering will be studied in more details in weeks 7 & 8



Sampling-Based Data Reduction

- Sampling: obtaining a small sample s to represent the whole data set N
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
 - Simple random sampling may have very poor performance in the presence of skew
 - Develop adaptive sampling methods, e.g., stratified sampling
- Note: Sampling may not reduce database I/Os (page at a time)

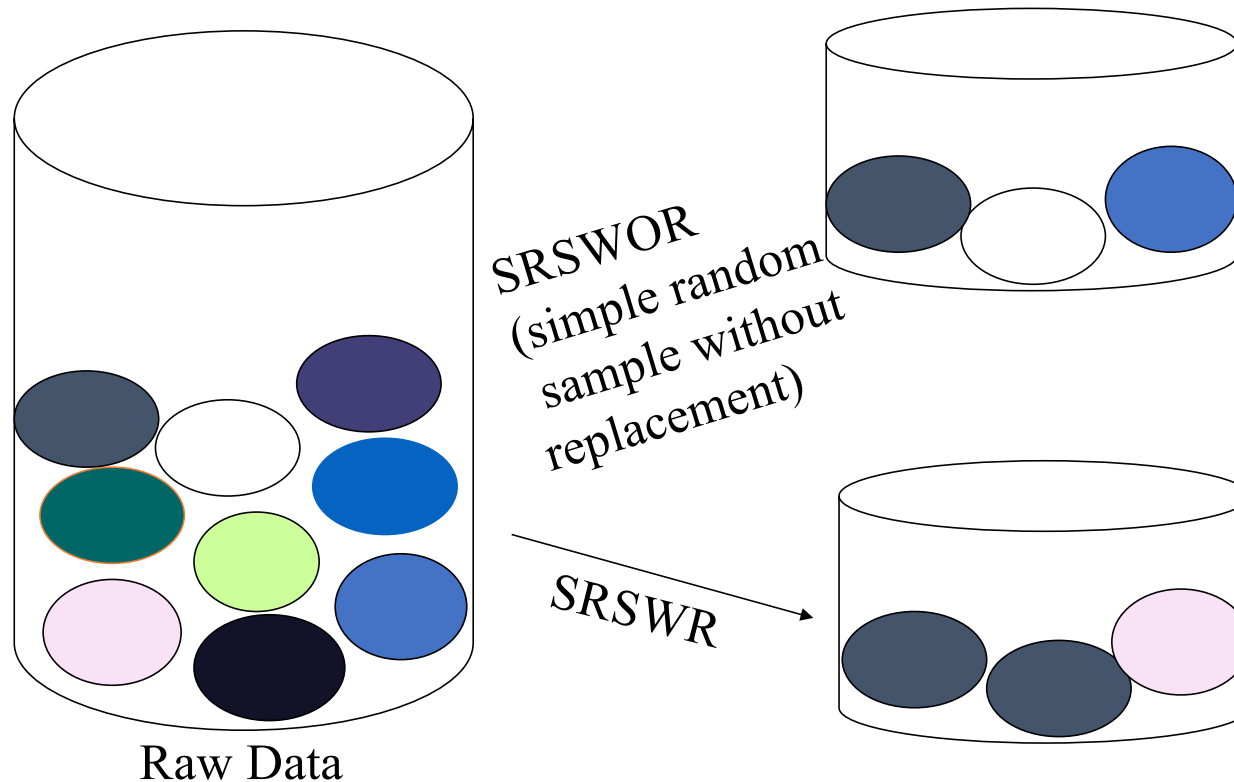


Types of Sampling

- Simple random sampling
 - There is an equal probability of selecting any particular item
- Sampling without replacement
 - Once an object is selected, it is removed from the population
- Sampling with replacement
 - A selected object is not removed from the population
- Stratified sampling:
 - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)
 - Used in conjunction with skewed data

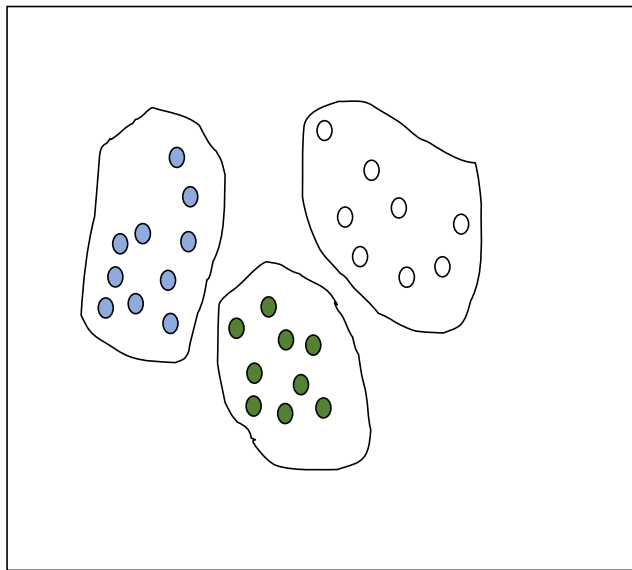


Sampling – with or without Replacement

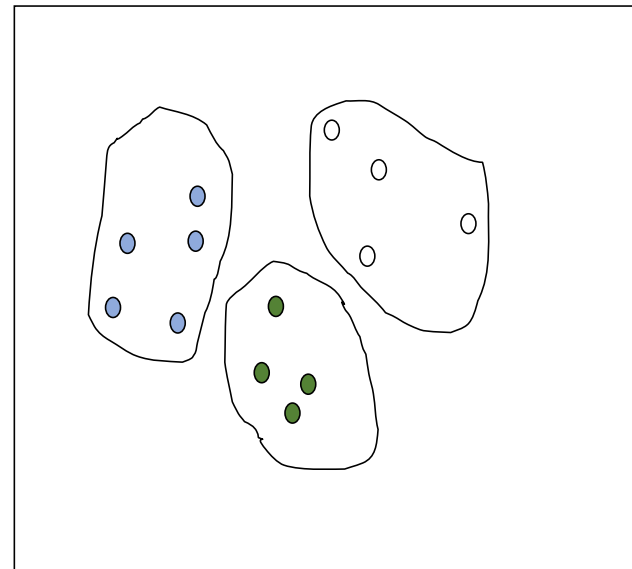


Sampling – Cluster or Stratified Sampling

Raw Data

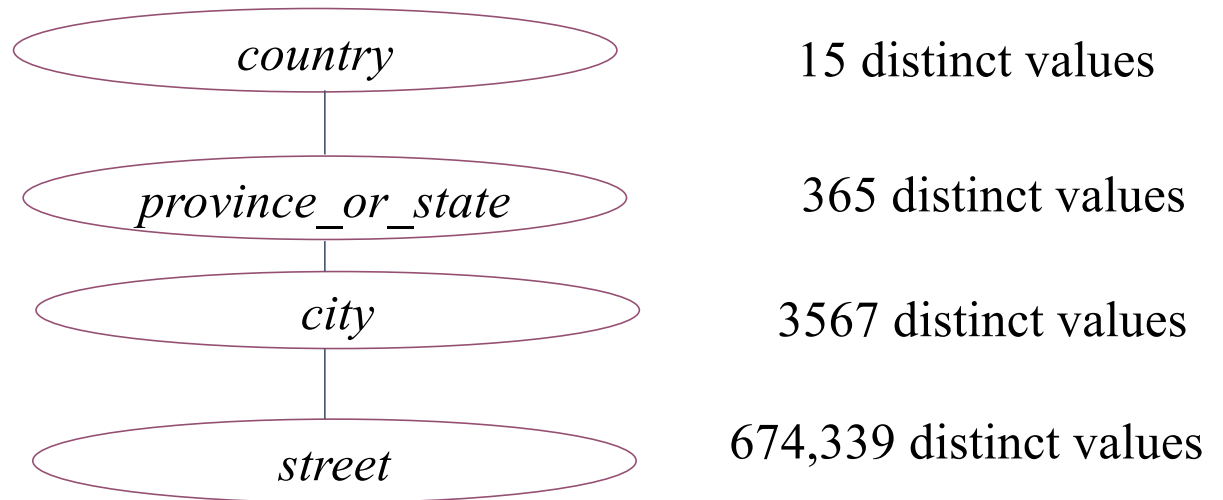


Cluster/Stratified Sample



Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
 - The attribute with the most distinct values is placed at the lowest level of the hierarchy



Data Discretization



Data Discretization

- Three types of attributes
 - Nominal: values from an unordered set, e.g., color, profession
 - Ordinal: values from an ordered set, e.g., military or academic rank
 - Numeric: real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
 - Interval labels can be used to replace actual data values
 - Supervised vs. unsupervised
 - Split (top-down) vs. merge (bottom-up)



Discretization Methods

- Typical methods: All the methods can be applied recursively
 - Binning – Histograms
 - Clustering
 - Classification (e.g. Decision-trees)
 - Correlation



Discretization Methods – Binning

- Equal-width (distance) partitioning
 - Divides the range into N intervals of equal size: uniform grid
 - If A and B are the smallest and largest 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

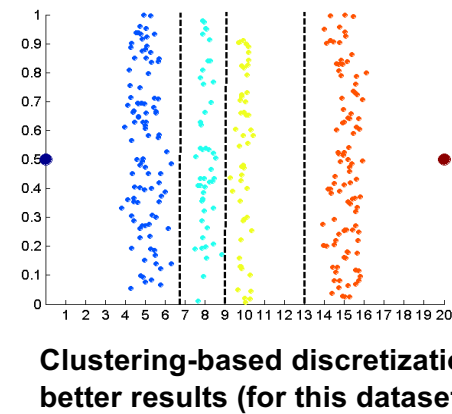
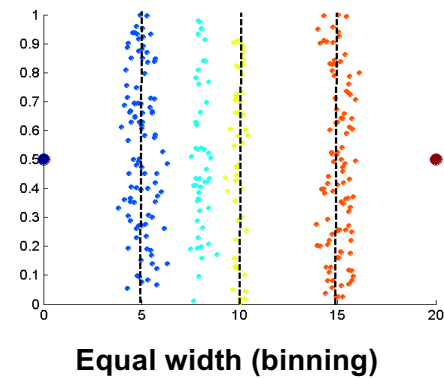
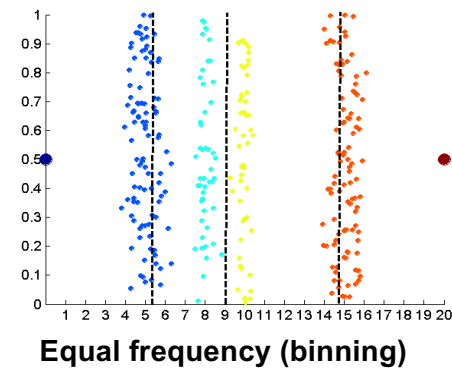
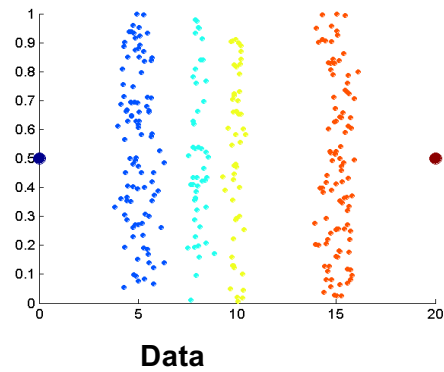


Discretization Methods – Binning (Cont.)

- Equal-depth (frequency) partitioning
 - Divides the range into N intervals, each containing approximately same number of samples
 - Good data scaling
 - Managing categorical attributes can be tricky



Discretization Methods – Binning (Cont.)

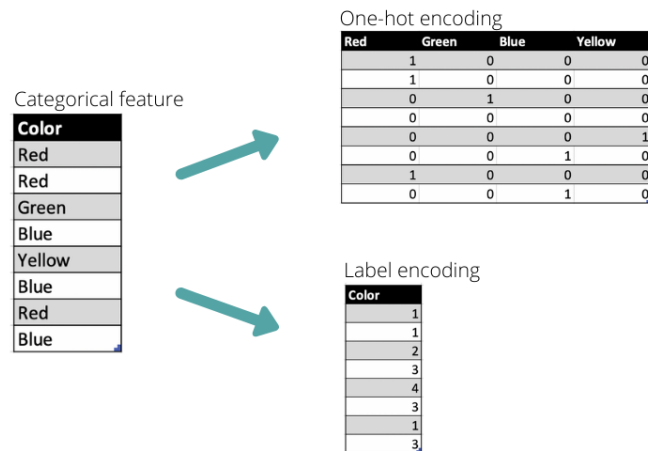


Discretization Methods – Classification & Correlation

- Classification (e.g., decision tree analysis)
 - Supervised: Given class labels, e.g., cancerous vs. benign
 - Using entropy to determine split point (discretization point)
 - Top-down, recursive split
- Correlation analysis
 - Supervised: use class information
 - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes) to merge
 - Merge performed recursively, until a predefined stopping condition is satisfied



Other Data Preparation Techniques



- Transforming categorical data into numerical data (See the Figure)
 - Label encoding
 - One-hot encoding
- Data Enrichment
 - Augmentation
 - Join & Union
- Data Validation
 - Check Permitted Characters
 - Find Type-Mismatched Data



Wrap-Up

Summarize what you learned
today in 2-minutes





The information in this presentation has been compiled with the utmost care,
but no rights can be derived from its contents.