Data Processing

Some slides are adapted from data mining course of UIUC

Outline

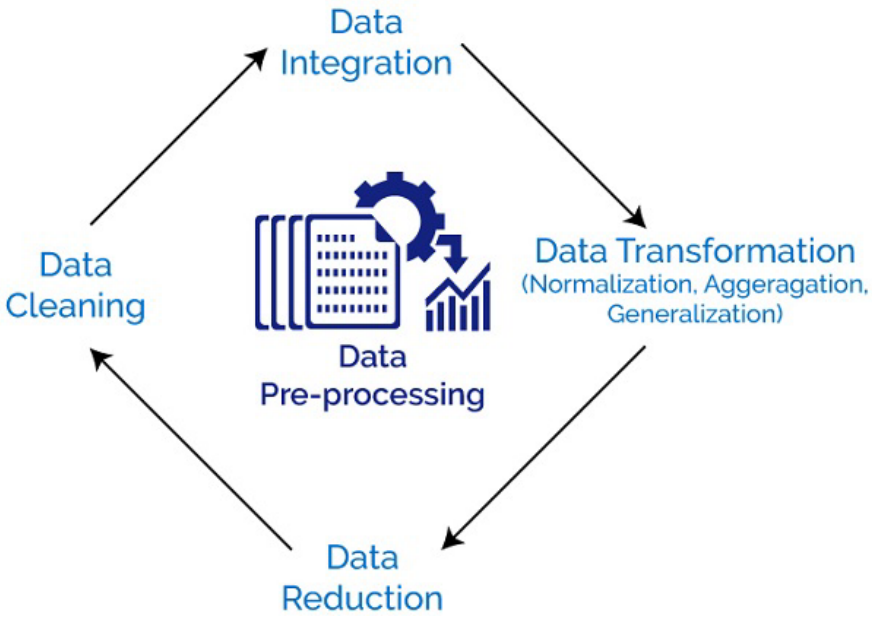
- Data Processing
  - Data Integration
  - Data Cleaning
  - Data Reduction
  - Data transformation and data discretization

Overview of data preprocessing

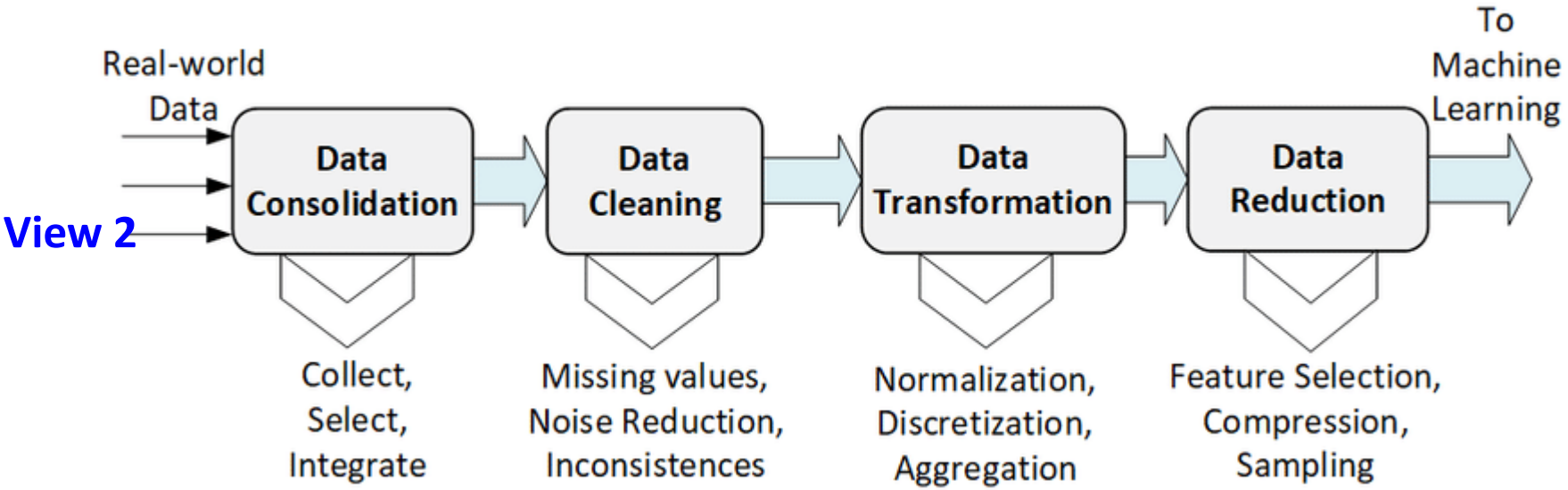
- Goal: get data into a structured form suitable for analysis
  - Variously called: data preparation, data munging, data curation
  - Also often called ETL (Extract-Transform-Load) process
- **Often the step where majority of time (80-90%) is spent**
- Main tasks:
  - **Scraping**: extracting information from sources, e.g., webpages, spreadsheets
  - **Data transformation and data discretization** : to get it into the right structure
  - **Information extraction**: extracting structured information from unstructured/text sources
  - **Data integration**: combine information from multiple sources
  - **Data cleaning**: Handle missing data, smooth noisy data, identify or remove outliers, and resolve inconsistencies
  - **Data reduction:** Dimensionality reduction, Numerosity reduction, Data compression

Relationship among main tasks

View 1



View 2



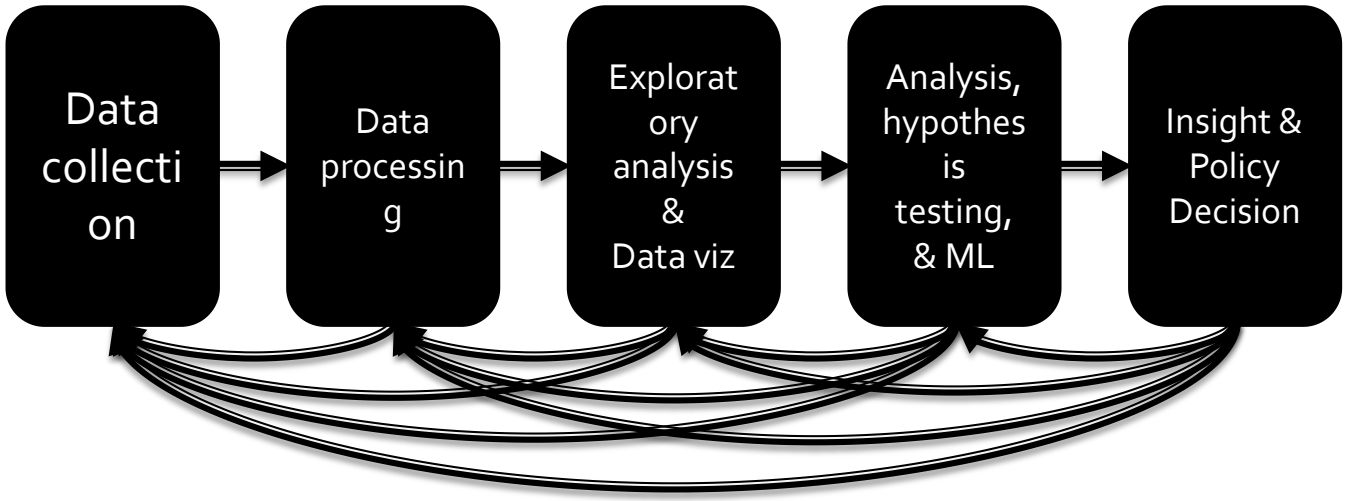
Data Preprocessing

Big Data Borat
@BigDataBorat

Following

In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

The Data Science LifeCycle



Data Integration

Goal: Combine data residing in different sources and provide users with a unified view of these data for querying or analysis

- Each data source has its own schema called **local schemas** (much work assumes relational schemas, but some work on XML as well)
- The unified schema is often called **mediated schema** or **global schema**

Two different setups:

1. Bring the data together into a single repository (often called data warehousing)
2. Keep the data where it is, and send queries back and forth

1. Data Warehousing

From Data Cleaning: Problems and Current Approaches

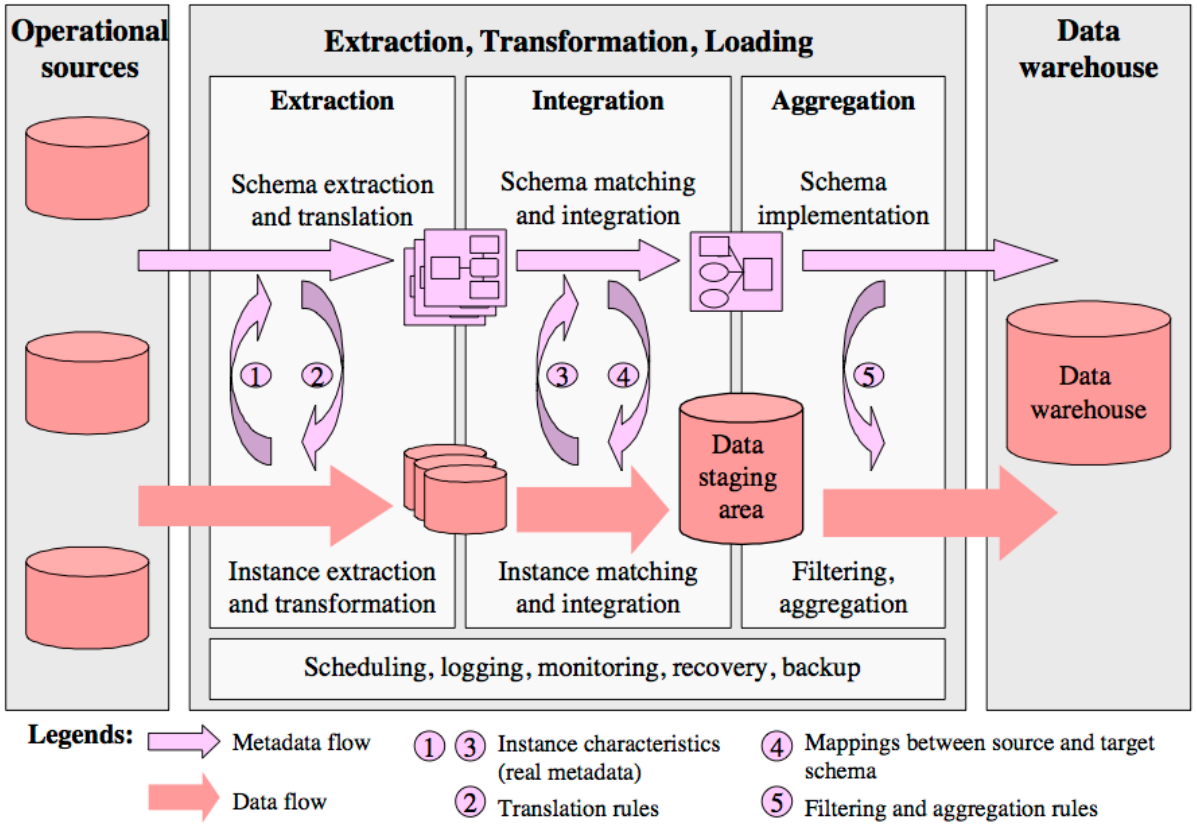
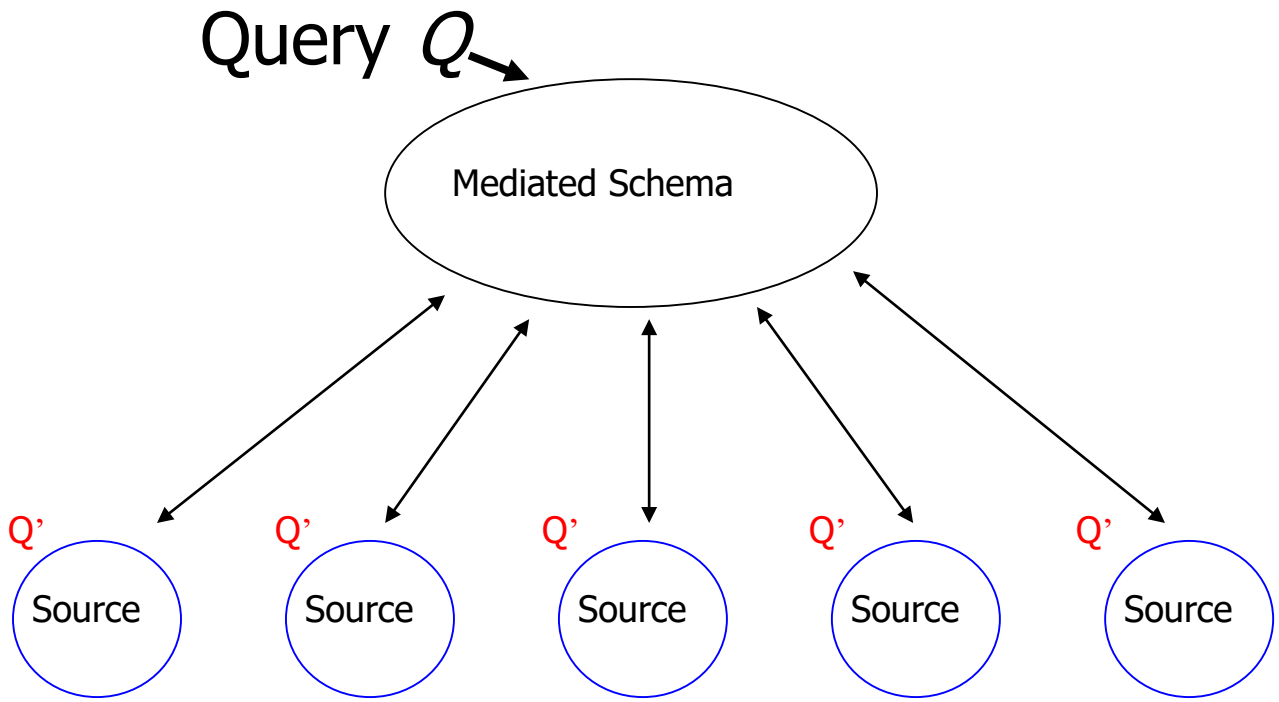


Figure 1. Steps of building a data warehouse: the ETL process

2. In-place integration



Data Integration

Two different setups:

1. Bring the data together into a single repository (often called data warehousing)
   - Relatively easier problem - only need one-way-mappings
   - Query performance predictable and under your control

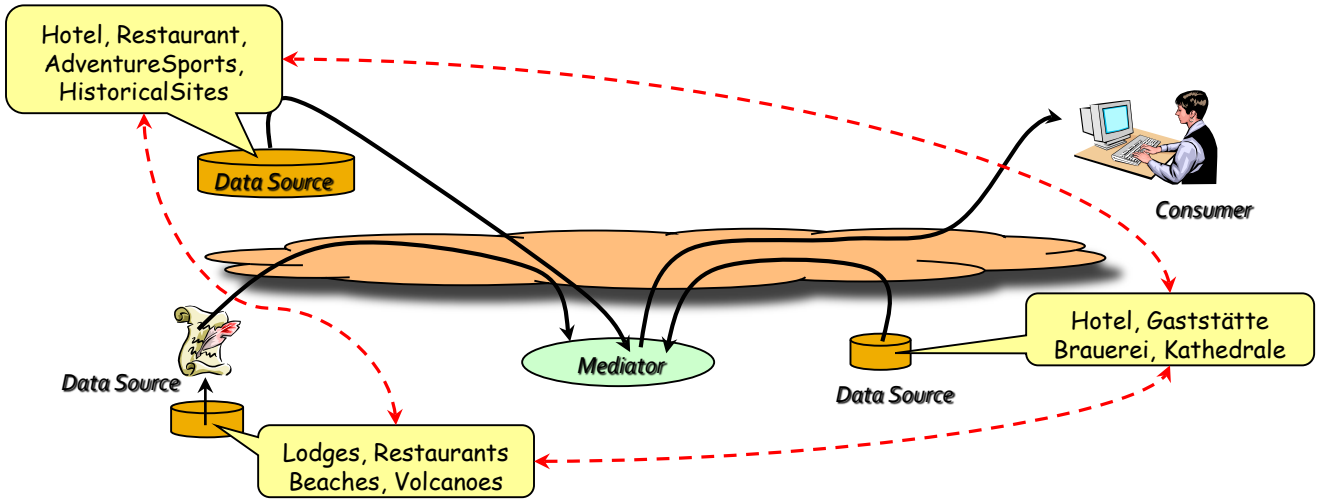
2. Keep the data where it is, and send queries back and forth
   - Need two-way mappings -- a query on the mediated schema needs to be translated into queries over data source schemas
   - Not as efficient and clean as data warehousing, but a better fit for dynamic data
   - Or when data warehousing is not feasible

Data Integration: Key Challenges

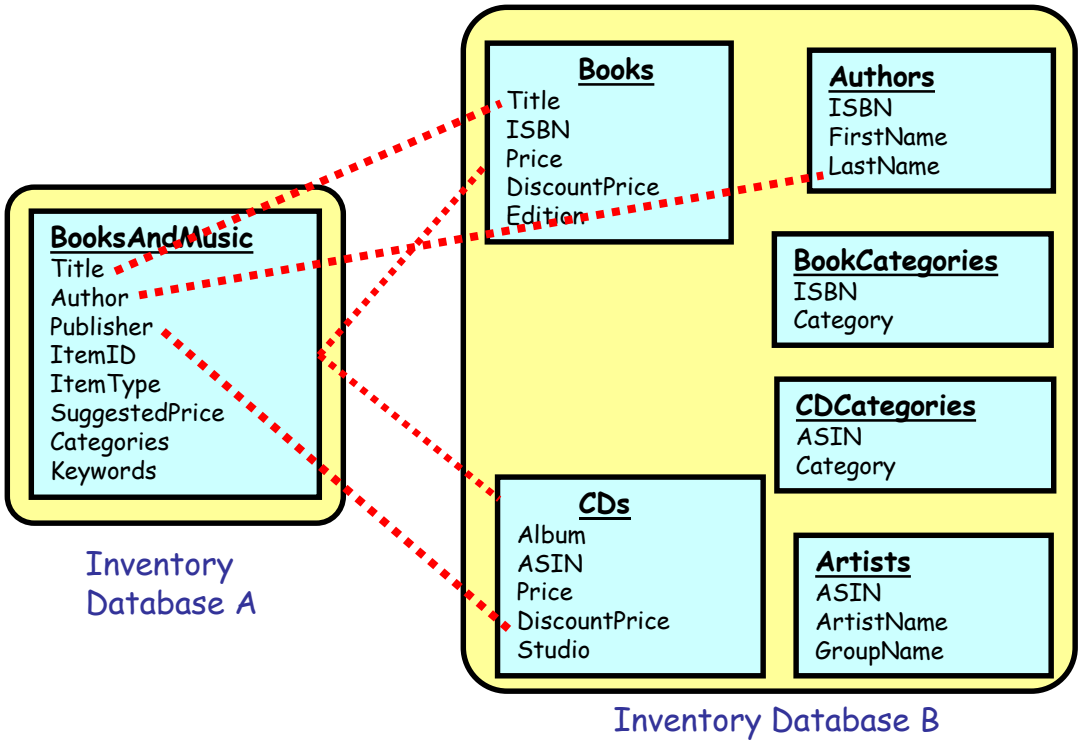
- Data extraction, reconciliation, and cleaning
  - Get the data from each source in a structured form
  - Often need to use wrappers to extract data from web sources
  - May need to define a schema
- Schema alignment and mapping
  - Decide on the best mediated schema
  - Figure out mappings and matchings between the local schemas and the global schema
- Answer queries over the global schema
  - In the second scenario, need to figure out how to map a query on global schema onto queries over local schemas
  - Also need to decide which sources contain relevant data
- Limitations in mechanisms for accessing sources
  - Many sources have limits on how you can access them
  - Limits on the number of queries you can issues (say 100 per min)
  - Limits on the types of queries (e.g., must enter a zipcode to get information from a web source)

Schema Matching or Alignment

- Goal: Identify corresponding elements in two schemas
  - As a first step toward constructing a global schema
  - Schema heterogeneity is a key roadblock
    - Different data sources speak their own schema



Schema Matching or Alignment



More words on data integration

- Data integration continues to be a very active area in research and increasingly industry
- Solutions still somewhat ad hoc and manual, although tools beginning to emerge
- Need to minimize the time needed to integrate a new data source
  - Crucial opportunities may be lost otherwise
  - Can take weeks to do it properly
- Dealing with changes to the data sources a major headache
  - Especially for data sources not under your control

Entity Resolution

- Identify different manifestations of the same real world object
  - Also called: identity reconciliation, record linkage, deduplication, fuzzy matching, Object consolidation, Coreference resolution, and several others
- Motivating examples: ??????????????
  - Postal addresses
  - Entity recognition in NLP/Information Extraction
  - Identifying companies in financial records
  - Comparison shopping
  - Author disambiguation in citation data
  - Connecting up accounts on online networks
  - Crime/Fraud Detection
  - Census
  - …

Entity Resolution

- Important to correctly identify references
  - Often actions taken based on extracted data
  - Cleaning up data by entity resolution can show structure that may not be apparent before
- Challenges
  - Such data is naturally ambiguous (e.g., names, postal addresses)
  - Abbreviations/data truncation
  - Data entry errors, Missing values, Data formatting issues complicate the problem
  - Heterogeneous data from many diverse sources
- No magic bullet here !!
  - Approaches fairly domain-specific
  - Be prepared to do a fair amount of manual work

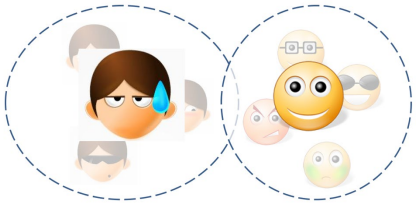
Entity Resolution: Three Slightly Different Problems

Setup:

- Real world: there are entities (people, addresses, businesses)
- We have a large collection of noisy, ambiguous "references" to those entities (also called "mentions")
- Somewhat different techniques, but a lot of similarities

Deduplication

- Cluster records/mentions that correspond to the same entity
- Choose/construct a cluster representative
  - This is in itself a non-trivial task (e.g., averaging may work for numerical attributes, but what about string attributes?)



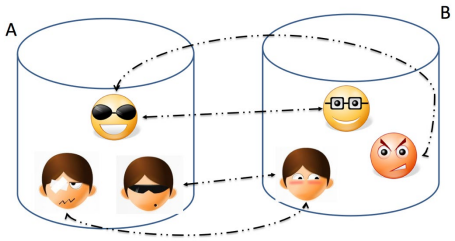
Entity Resolution: Three Slightly Different Problems

Setup:

- Real world: there are entities (people, addresses, businesses)
- We have a large collection of noisy, ambiguous "references" to those entities (also called "mentions")
- Somewhat different techniques, but a lot of similarities

Record Linkage (or Entity Matching)

- Match records across two different databases (e.g., two social networks, or financial records w/ campaign donations)
- Typically assume that the two databases are fairly clean



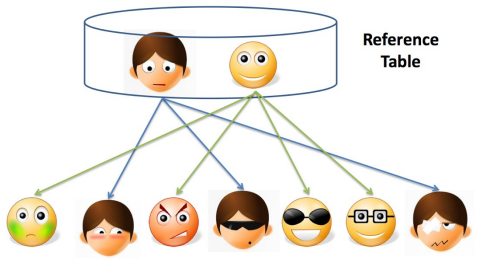
Entity Resolution: Three Slightly Different Problems

Setup:

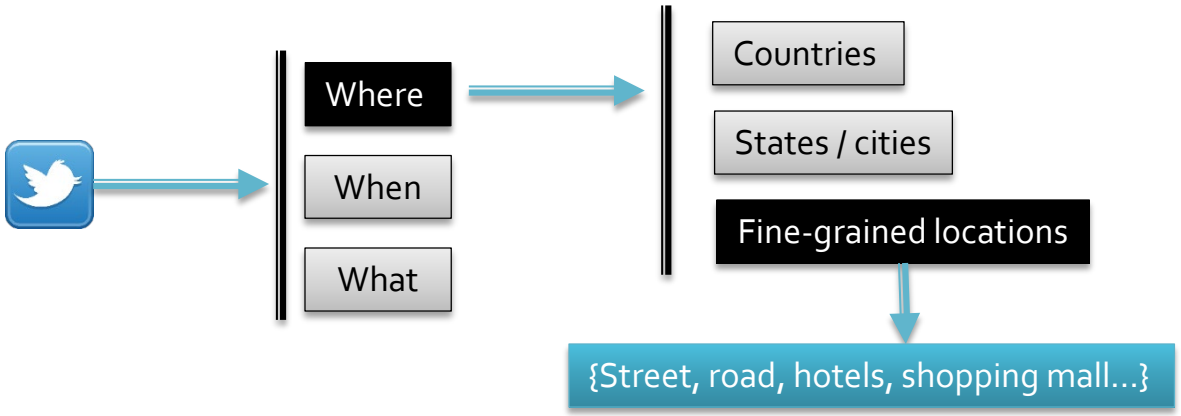
- Real world: there are entities (people, addresses, businesses)
- We have a large collection of noisy, ambiguous "references" to those entities (also called "mentions")
- Somewhat different techniques, but a lot of similarities

Reference Matching

- Match "references" to clean records in a reference table
- Commonly comes up in "entity recognition" (e.g., matching newspaper article mentions to names of people)



An example of Reference Matching: Fine-grained location recognition and linking



Zongcheng Ji, Aixin Sun, Gao Cong, Jialong Han: **Joint Recognition and Linking of Fine-Grained Locations from Tweets**. WWW 2016: 1271-1281

Entity Resolution: Entity Matching

Comprehensive treatment: Data Matching; P. Christen; 2012 (Springer Books -- not available for free)

- One of the key issues is finding similarities between two references
  - What similarity function to use?
- Edit Distance Functions
  - Levenstein: min number of changes to go from one reference to another
    - A change is defined to be: a single character insertion or deletion or substitution
    - May add transposition
  - Many adjustments to the basic idea proposed (e.g., higher weights to changes at the start)
  - Not cheap to compute, especially for millions of pairs
- Set Similarity
  - Some function of intersection size and union size
  - E.g., Jaccard distance = size of intersection/size of union
  - Much faster to compute
- Vector Similarity
  - Cosine similarity

Entity Resolution: Entity Matching

- Q-Grams
  - Find all length-q substrings in each string
  - Use set/vector similarity on the resulting set
- Several approaches that combine the above (especially q-grams and edit distance)
- Soundex: Phonetic Similarity Metric
  - Homophones should be encoded to the same representation so spelling errors can be handled
- May need to use Translation Tables
  - To handle abbreviations, nicknames, other synonyms
- Different types of data requires more domain-specific functions
  - E.g., geographical locations, postal addresses
  - Also much work on computing distances between XML documents etc.

Entity Resolution: Algorithms

- Simple threshold method
  - If the distance below some number, the two references are assumed to be equal
  - May review borderline matches manually
- Can be generalized to rule-based:
  - Example from Christen, 2012

$$(s(\text{GivenName})[r_i, r_j] \geq 0.9) \wedge (s(\text{Surname})[r_i, r_j] = 1.0) \wedge (s(\text{BMonth})[r_i, r_j] = 1.0) \wedge (s(\text{BYear})[r_i, r_j] = 1.0) \Rightarrow [r_i, r_j] \rightarrow \text{Match}$$

$$(s(\text{GivenName})[r_i, r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i, r_j] \geq 0.8) \wedge (s(\text{BDay})[r_i, r_j] = 1.0) \wedge s(\text{BMonth})[r_i, r_j] = 1.0) \wedge (s(\text{BYear})[r_i, r_j] = 1.0) \Rightarrow [r_i, r_j] \rightarrow \text{Match}$$

$$(s(\text{GivenName})[r_i, r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i, r_j] \geq 0.8) \wedge (s(\text{StrName})[r_i, r_j] \geq 0.8) \wedge (s(\text{Suburb})[r_i, r_j] \geq 0.8) \Rightarrow [r_i, r_j] \rightarrow \text{Match}$$

$$(s(\text{GivenName})[r_i, r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i, r_j] \geq 0.8) \wedge (s(\text{BDay})[r_i, r_j] \leq 0.5) \wedge (s(\text{BMonth})[r_i, r_j] \leq 0.5) \wedge (s(\text{BYear})[r_i, r_j] \leq 0.5) \Rightarrow [r_i, r_j] \rightarrow \text{Non-Match}$$

$$(s(\text{GivenName})[r_i, r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i, r_j] \geq 0.8) \wedge (s(\text{StrName})[r_i, r_j] \leq 0.6) \wedge (s(\text{Suburb})[r_i, r_j] \leq 0.6) \Rightarrow [r_i, r_j] \rightarrow \text{Non-Match}$$

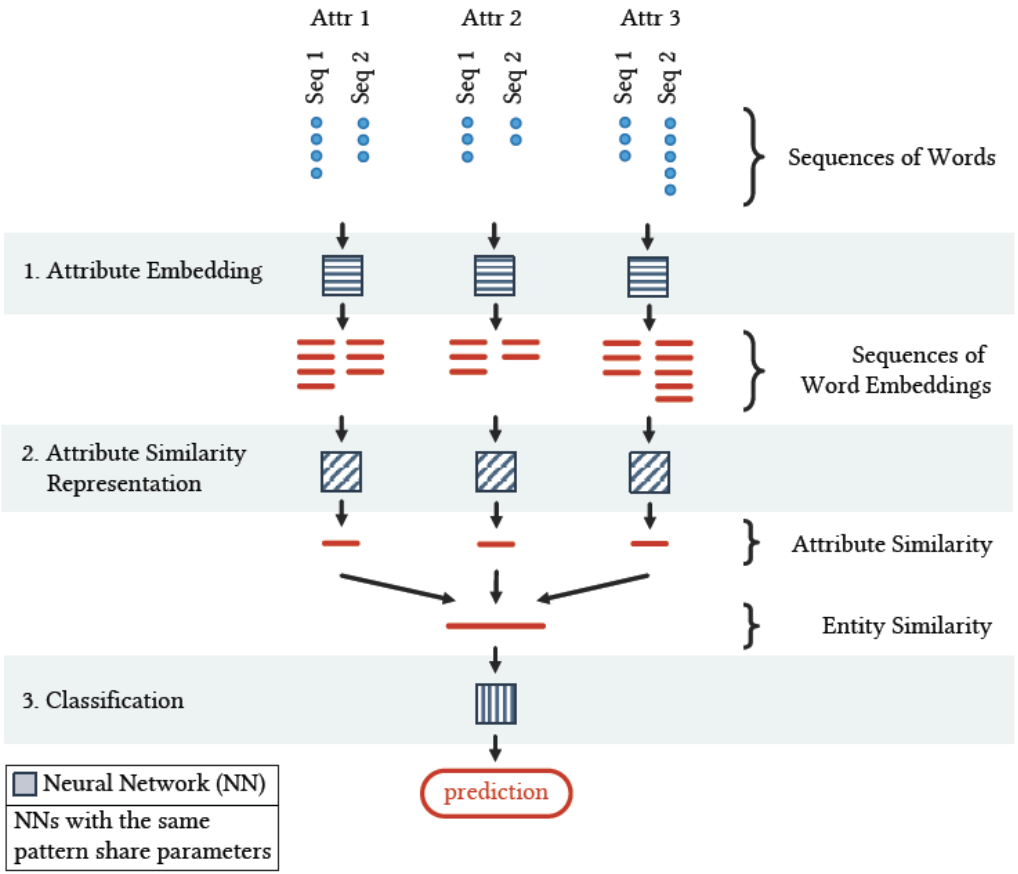
Entity Resolution: Algorithms

- May want to give more weight to matches involving rarer words
  - More naturally applicable to record linkage problem
  - If two records match on a rare name like "Machanavajjhala", they are likely to be a match
  - Can formalize this as "probabilistic record linkage”
- Constraints: May need to be satisfied, but can also be used to find matches
  - Often have constraints on the matching possibilities
  - Transitivity: M1 and M2 match, and M2 and M3 match, and M1 and M3 must match
  - Exclusivity: M1 and M2 match --> M3 cannot match with M2
  - Other types of constraints:
    - E.g., if two papers match, their venues must match

Entity Resolution: Scaling to Big Data

- One immediate problem
  - There are $O(N^2)$ possible matches
  - Must reduce the search space
- Use some easy-to-evaluate criterion to restrict the pairs considered further
  - May lead to false negative (i.e., missed matches) depending on how noisy the data is
- Much work on this problem as well, but domain-specific knowledge likely to be more useful in practice

Deep learning based entity matching



Mudgal et al. Deep Learning for Entity Matching: A Design Space Exploration, SIGMOD18

Outline

- Data Processing
  - Data Integration
  - Data Cleaning
  - Data Reduction
  - Data transformation and data discretization

Data Cleaning: Single-source problems

- Databases can enforce constraints, whereas data extracted from files or spreadsheets, or scraped from webpages is much more messy
- Types of problems:
  - Ill-formatted data, especially from webpages or files or spreadsheets
  - Missing or illegal values, Misspellings, Use of wrong fields, Extraction issues (not easy to separate out different fields)
  - Duplicated records, Contradicting Information, Referential Integrity Violations
  - Unclear default values (e.g., data entry software needs something)
  - Evolving schemas or classification schemes (for categorical attributes)
  - Outliers

Examples of Data quality Problems

Scope/Problem		Dirty Data	Reasons/Remarks
Attribute	Missing values	phone=9999-999999	unavailable values during data entry (dummy values or null)
	Misspellings	city="Liipzig"	usually typos, phonetic errors
	Cryptic values, Abbreviations	experience="B"; occupation="DB Prog."	
	Embedded values	name="J. Smith 12.02.70 New York"	multiple values entered in one attribute (e.g. in a free-form field)
	Misfielded values	city="Germany"	
Record	Violated attribute dependencies	city="Redmond", zip=77777	city and zip code should correspond
Record type	Word transpositions	$name_1$= "J. Smith", $name_2$="Miller P."	usually in a free-form field
	Duplicated records	emp_1=(name="John Smith",...); emp_2=(name="J. Smith",...)	same employee represented twice due to some data entry errors
	Contradicting records	emp_1=(name="John Smith", bdate=12.02.70); emp_2=(name="John Smith", bdate=12.12.70)	the same real world entity is described by different values
Source	Wrong references	emp=(name="John Smith", deptno=17)	referenced department (17) is defined but wrong

Table 2. Examples for single-source problems at instance level

Data Cleaning: Multi-source problems

- Different sources are developed separately, and maintained by different people
- Issue 1: Mapping information across sources (schema mapping/transformation)
  - Naming conflicts: same name used for different objects
  - Structural conflicts: different representations across sources
- Issue 2: Entity Resolution: Matching entities across sources
- Issue 3: Data quality issues
  - Contradicting information, Mismatched information, etc.

Constraint based data cleaning

- Inconsistencies: violations of functional dependencies (FDs)and conditional functional dependencies (CFDs).
- Example: Customer database: (country code (CC), area code (AC), phone number (PN)), name (NM), and address (street (STR), city (CT), zip code (ZIP)).
  - functional dependencies (FDs) on a cust relation: (hold on all the tuples)
    - *f: [CC, AC, PN]→[STR, CT, ZIP]*
    - *f: [CC, AC]→[CT]*
  - CFD: The following constraint is supposed to hold only when the country code is 44. That is, for customers in the UK, ZIP determines STR:
    - *φ: [CC = 44, ZIP]→[STR]*
- Problem: Given a set of FDs, CFDs and data, we want to repair the data to make it consistent with CFDs.

Gao Cong et al. **Improving Data Quality: Consistency and Accuracy.** VLDB 2007: 315-326

Outline

- Data Processing
  - Data Integration
  - Data Cleaning
  - Data Reduction
  - Data transformation and data discretization

Data Reduction

- **Data reduction**:
  - Obtain a reduced representation of the data set
    - much smaller in volume but yet produces *almost* the same analytical results
- Why data reduction?—A database/data warehouse may store terabytes of data
  - Complex analysis may take a very long time to run on the complete data set
- **Methods for data reduction** (also *data size reduction* or *numerosity reduction*)
  - Regression and Log-Linear Models
  - Histograms, clustering, sampling
  - Data cube aggregation
  - Data compression
  - …

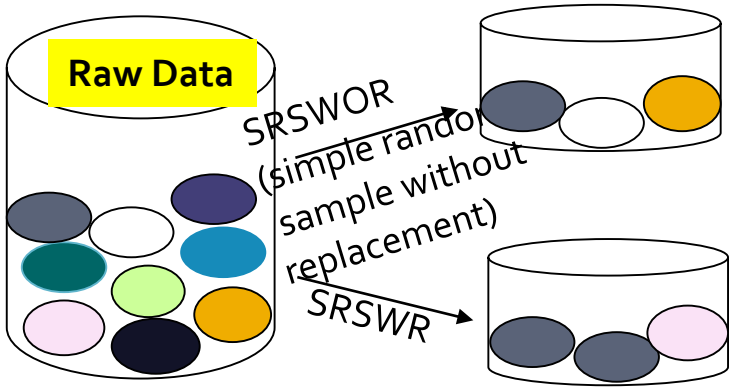
Sampling

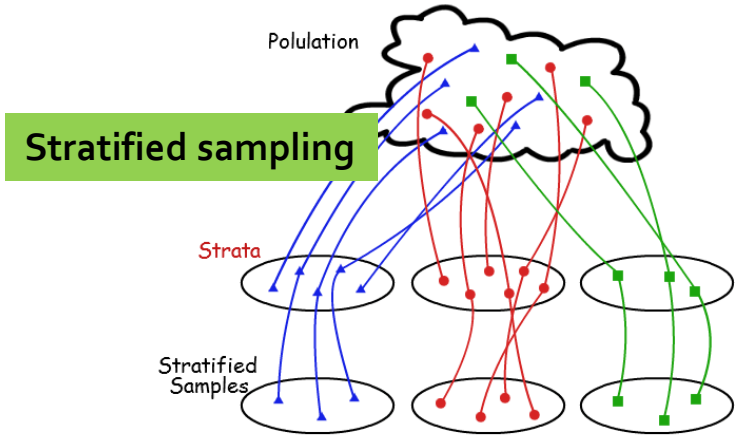
- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity

- 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

Types of Sampling

- **Simple random sampling:** 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 (or cluster) the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)





Outline

- Data Processing
  - Data Integration
  - Data Cleaning
  - Data Reduction
  - Data transformation and data discretization

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
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization
  - Normalization: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - Discretization

Discretization

- Discretization: Divide the range of a continuous attribute into intervals
  - Decide how many intervals/categories, and where to split into intervals
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Prepare for further analysis, e.g., classification

Simple Discretization: Binning

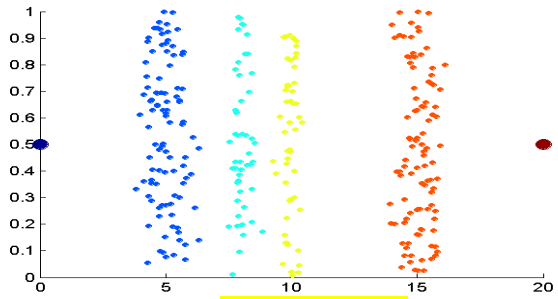
- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples

Example: Binning Methods for Data Smoothing

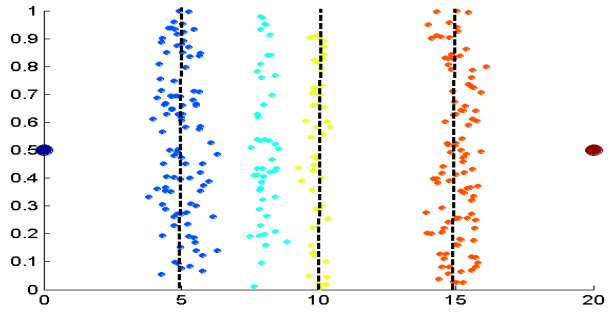
- Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (**equi-depth**) bins:
  - Bin 1: 4, 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: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

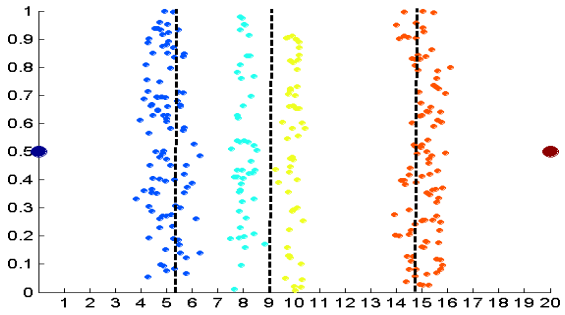
Discretization Without Supervision: Binning vs. Clustering



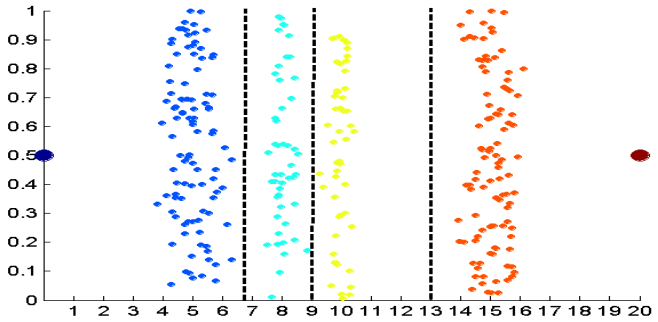
Data



Equal width (distance) binning



Equal depth (frequency) (binning)



K-means clustering leads to better results

Data Discretization Methods

- Binning
  - Top-down split, unsupervised
- Histogram analysis
  - Top-down split, unsupervised
- Clustering analysis
  - Unsupervised, top-down split or bottom-up merge
- Decision-tree analysis
  - Supervised, top-down split
- Correlation (e.g., χ^2) analysis
  - Unsupervised, bottom-up merge
- Note: All the methods can be applied recursively

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
- **Dimensionality reduction**
  - Reducing the number of random variables under consideration, via obtaining a set of principal variables
- **Advantages of 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

Dimensionality Reduction Techniques

- Dimensionality reduction methodologies
  - **Feature selection**: Find a subset of the original variables (or features, attributes)
  - **Feature extraction**: Transform the data in the high-dimensional space to a space of fewer dimensions
- Some typical dimensionality reduction methods
  - Principal Component Analysis
  - Supervised and nonlinear techniques
    - Feature subset selection
    - Feature creation

Summary

- Data Processing
  - Data Integration
  - Data Cleaning
  - Data Reduction
  - Data transformation and data discretization
- Expectation: understand the basic concept. No computation is required