Mining of Massive Datasets

2023 Spring

Ki Yong Lee, Ph.D.

Professor

Division of Computer Science

Sookmyung Women's University

Before the Lecture…

- What is data mining?
  - Knowledge discovery from data
- But to extract the knowledge, data needs to be …
  - Stored
  - Managed
  - And ***Analyzed*** ← this class
- Data mining ≈ Big data ≈ Data analytics ≈ Data science

What is Data Mining?

- Given ***lots*** of data, discover ***patterns*** and ***models*** that are:
  - **Valid**: hold on new data with some certainty
  - **Useful**: should be possible to act on the item
  - **Unexpected**: non-obvious to the system
  - **Understandable**: humans should be able to interpret the pattern

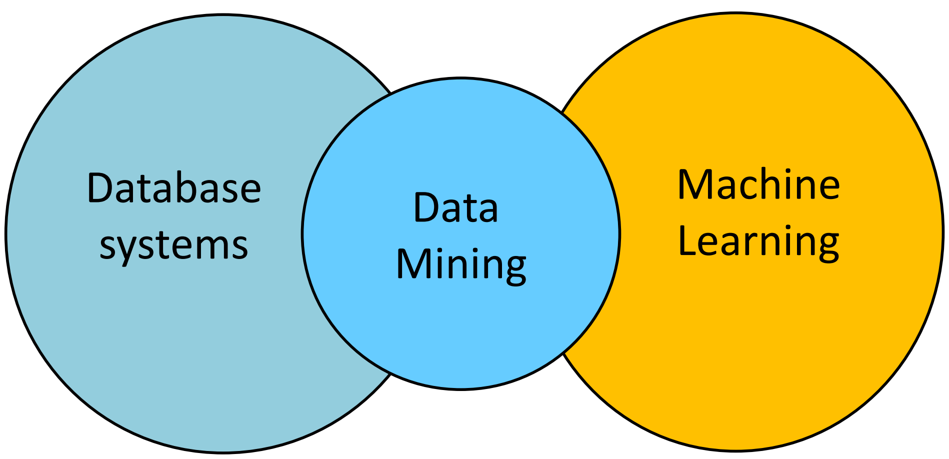
Data Mining Tasks

- Descriptive methods
  - Find human-interpretable patterns that describe the data
  - (ex) clustering

- Predictive methods
  - Use some variables to predict unknown or future values of other variables
  - (ex) recommender systems

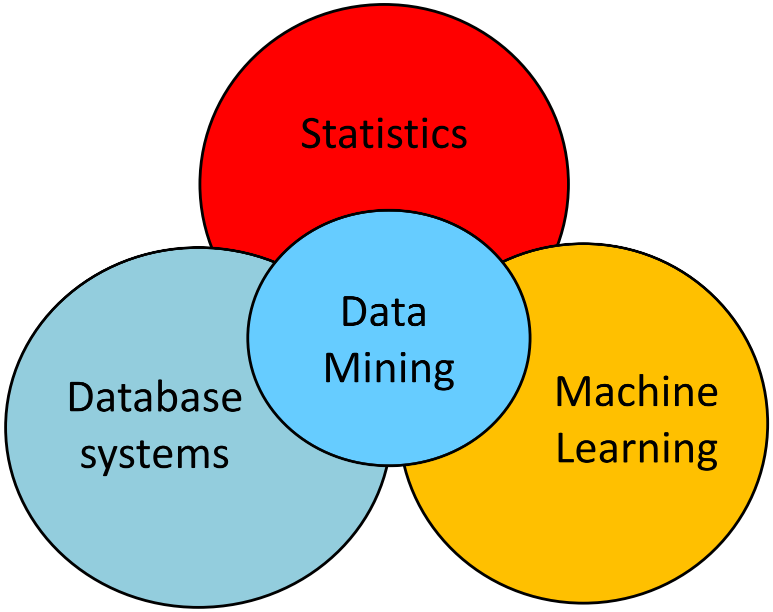
Data Mining: Cultures

- Data mining overlaps with
  - **Databases**: large-scale data, simple queries
    - To a DB person, data mining is an extreme form of analytic processing (i.e., queries that examine large amount of data)
    - Result = the query answer
  - **Machine learning**: small data, complex models
    - To a ML person, data mining is the inference of models
    - Result = the parameters of the model



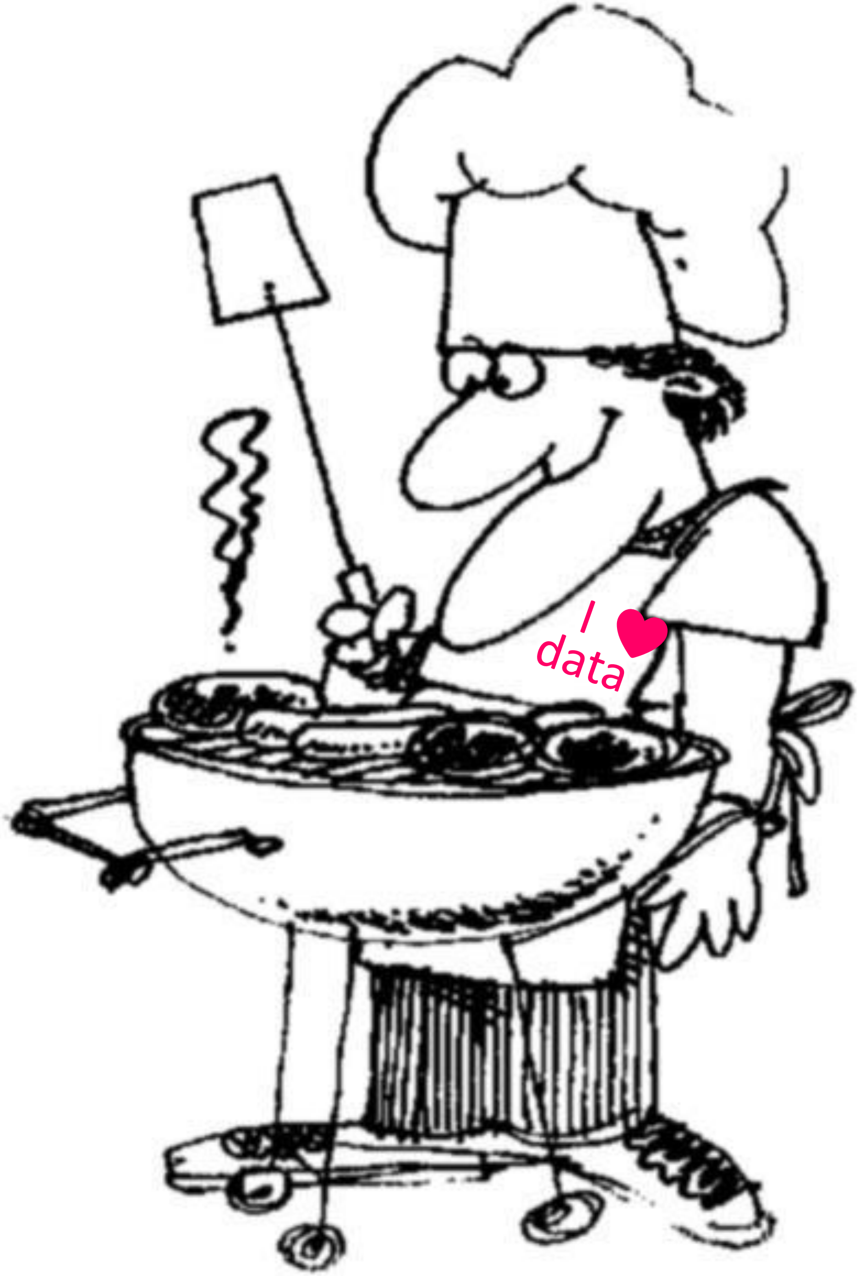
This Class

- This class overlaps with machine learning, statistics, artificial intelligence, databases, but more stress on
  - Scalability (Big data)
  - Algorithms
  - Computing architectures
  - Automation for handling large data



What Will We Learn?

High dim. data	Graph data	Infinite data	Machine learning	Apps
Locality sensitive hashing	PageRank, SimRank	Filtering data streams	SVM	Recommender systems
Clustering	Community Detection	Web advertising	Decision Trees	Association Rules
Dimensionality reduction	Spam Detection	Queries on streams	Perceptron, kNN	Duplicate document detection



How do you want that data?

Course Information (1/3)

- Instructor
  - Ki Yong Lee (Professor, Division of Computer Science)
    - Office: Saehim Hall 406
    - Phone: 02-2077-7583m 010-… (upon request)
    - Email: kiyonglee@sookmyung.ac.kr
    - Homepage: http://cs.sookmyung.ac.kr/~kylee
    - Office hour: You are ***always*** welcome! (but prior appointment is recommended)

- Course homepage
  - SnowBoard → 전산학특강I (001)

Course Information (2/3)

- Main topics
  - Data mining (of very large amounts of data)
    - MapReduce
    - Finding similar items
    - Mining data streams
    - Link analysis
    - Frequent itemsets
    - Clustering ?
    - Advertising on the Web
    - Recommendation systems
    - Mining social-network graphs
    - Dimensionality reduction
    - Large-scale machine learning

Course Information (3/3)

- Textbook
  - Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, "Mining of Massive Datasets," 3rd Edition, Cambridge University Press, 2020
  - Free online: http://www.mmds.org

- Grading policy
  - Midterm Exam : 40%
  - Final Exam : 40%
  - Project or Paper : 20% (subject to change)

  - Total : 100%

Chapter 1

Data Mining

What is Data Mining?

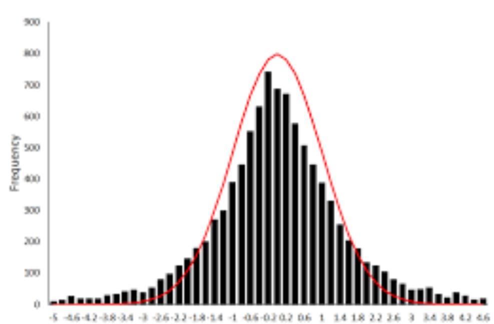
- The most commonly accepted definition of "data mining"
  - The discovery of "***models***" for data

- Model?
  1. Statistical modeling
  2. Machine learning
  3. Summarization
  4. Feature extraction

1. Statistical Modeling

- Statisticians view data mining as the construction of a ***statistical model***
  - An underlying distribution from which the visible data is drawn

- Example
  - Suppose our data is a set of numbers
  - You decide that the data comes from a Gaussian distribution
  - You use a formula to compute the most likely parameters of this Gaussian
  - The mean and standard deviation completely characterize the distribution and would become the ***model*** of the data



$$f(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

2. Machine Learning

- Machine-learning practitioners use the data as a training set, to train a ***machine-learning algorithm***
  - (ex) Bayes nets, support-vector machines, decision trees, hidden Markov models, etc.

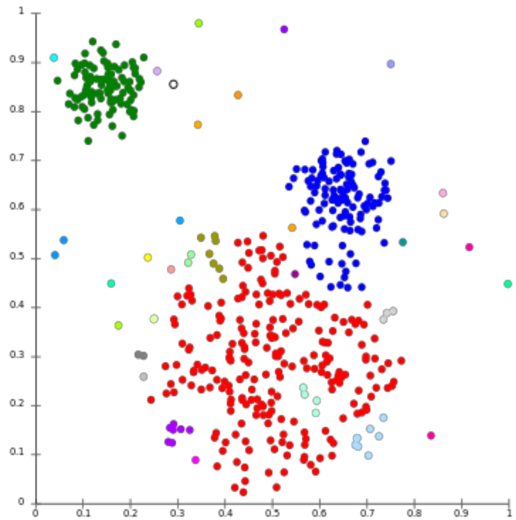
- Good approach when we have little idea of what we are looking for in the data

- Example
  - It is rather unclear what it is about movies that makes certain movie-goers like or dislike it
  - We can devise a machine-learning algorithm that predicts the ratings of movies by users, based on a sample of their responses

3. Summarization

- ***Summarizing*** the data succinctly and approximately

- Example
  - PageRank
    - The entire complex structure of the Web is summarized by a single number for each page
    - This number (the "PageRank" of the page) reflects the ***importance*** of the page
  - Clustering
    - Data is views as points in a multidimensional space
    - Points that are "close" in this space are assigned to the same cluster
      - Cluster summary = (centroid, average distance)
    - These cluster summaries becomes the summary of the entire data set



4. Feature Extraction

- Extracting the most prominent ***features*** of the data and ignore the rest

- Examples
  - Frequent itemsets
    - We look for small sets of items that appear together in many baskets
    - These "frequent itemsets" are the characterization of the data that we seek
  - Similar items
    - Given a collection of sets, find pairs of sets that have a relatively large fraction of their elements in common
    - (ex) in Amazon, customers are treated as the set of items they have bought, and Amazon can look for "similar" customers to recommend something many of these customers have bought

Statistical Limits on Data Mining

- A common sort of data-mining problem
  - Discovering ***unusual*** events hidden within massive amounts of data

- A risk with data mining is that an analysis can discover patterns that are ***meaningless***

- Statisticians call it "Bonferroni's Principle"
  - We may only discover meaningful patterns by looking for events that are so ***rare*** that they are unlikely to occur in ***random*** data

Bonferroni's Principle

- Helps us avoid treating random occurrences as if they were real

- Basic steps
  - Calculate the expected number of occurrences of the events you are looking for, ***on the assumption that data is random***
  - If this number is larger than the number of real instances you hope to find, then you must expect almost anything you find to be ***bogus***
    - i.e., a statistical artifact rather than evidence of what you are looking for
  - In other words, random data will always have some number of unusual events that look significant but aren't

(Ex) Bonferroni's Principle (1/2)

- Suppose there are some "evil-doers" and we want to detect them

- To find evil-doers, we shall find people who ***at least twice have stayed at the same hotel on the same day***

- Assumptions
  - There are 10^9 people
  - Everyone goes to a hotel 1 day in 100
  - A hotel holds 100 people, so there are 10^5 hotels
  - We shall examine hotel records for 1000 days

- If everyone behaves randomly (i.e., no terrorists), will the data mining detect anything suspicious?

(Ex) Bonferroni's Principle (2/2)

- The expected number of "suspicious" pairs of people
  - The probability of two people visiting a hotel on a given day = 10^{-4}
  - The probability of two people visiting the ***same*** hotel on a given day = $10^{-4}/10^{5} = 10^{-9}$
  - The probability of two people visiting the same hotel on the two different given days = $10^{-9} \times 10^{-9} = 10^{-18}$
  - The number of pairs of people = ${}_{10^9}C_2 \approx 5 \times 10^{17}$
  - The number of pairs of days = ${}_{1000}C_2 \approx 5 \times 10^{5}$
  - The expected number of events that any two people were at the same hotel on two different days = $5 \times 10^{17} \times 5 \times 10^{5} \times 10^{-18}$ = **250,000**

- That is, there are too many pairs of people who look like evil-doers, even though they are not
  - We need to have some additional evidence to find "suspicious" pairs of people in some more efficient way

Things Useful to Know

- Each will be useful in the study of data mining

1. TF.IDF measure of word importance
2. Hash functions
3. Indexes
4. Secondary storage

1. TF.IDF Measure of Word Importance (1/3)

- How can we categorize documents by their topic?
  - We often find the significant words in those documents

- Significant (or important) words?

① Words appearing frequently in a document

- Except stop words such as “the” or “and”

→ ***Term Frequency (TF)***

② Words appearing only in just a few documents

- “albeit” vs. “chukker”

→ ***Inverse Document Frequency (IDF)***

1. TF.IDF Measure of Word Importance (2/3)

- Term frequency (TF_{ij})

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

  - f_{ij}: the frequency of term (word) i in document j
  - $\max_k f_{kj}$: the maximum number of occurrence of any term in document j
  - The most frequent term in document j gets a TF of 1

- Inverse document frequency (IDF_i)

$$IDF_i = \log_2(N/n_i)$$

  - N: the number of documents
  - n_i: the number of documents in which term i appears

1. TF.IDF Measure of Word Importance (3/3)

- The TF.IDF score for term i in document j

$$TF_{ij} \times IDF_i$$

  - The terms with the highest TF.IDF score are often terms that ***best characterize*** the topic of document

- Example
  - The number of documents $= 2^{20} = 1{,}048{,}576$
  - Suppose word w appears in $2^{10} = 1{,}024$ documents
  - Suppose word w appears in document j 20 times, and that is the maximum number of times in which any word appears
  - The TF.IDF score for w in j = $TF_{wj} \times IDF_w = 1 \times \log_2(20^{20}/2^{10}) = 10$

2. Hash Functions (1/2)

- A hash function $h(x)$
  - Takes a hash-key value x as an argument
  - Produces a bucket number $(0, 1, \ldots, B - 1)$ as a result
    - B: the number of buckets
- Property of hash functions
  - They "randomize" hash-keys
  - If hash-keys are drawn randomly, then h will send approximately equal numbers of hash-keys to each of the B buckets
- Example: $h(x) = x \bmod B$
  - If the population of x is all positive integers, h works fine
  - If the population of x is even integers and $B = 10$, then h is nonrandom
    - What if $B = 11$?

2. Hash Functions (2/2)

- What if hash-keys are not integers?
  - The values of all data types are composed of bits, and sequences of bits can always be interpreted as ***integers***
  - For a record type, each of whose components has its own type
    - Recursively convert the value of each component to an ***integer***
    - ***Sum*** the integers for the components
    - Convert the integer sum to buckets by dividing by B
  - For an array, set, or bag type
    - Convert the values of the elements' type to ***integers***
    - ***Sum*** the integers
    - Divide the integer sum by B

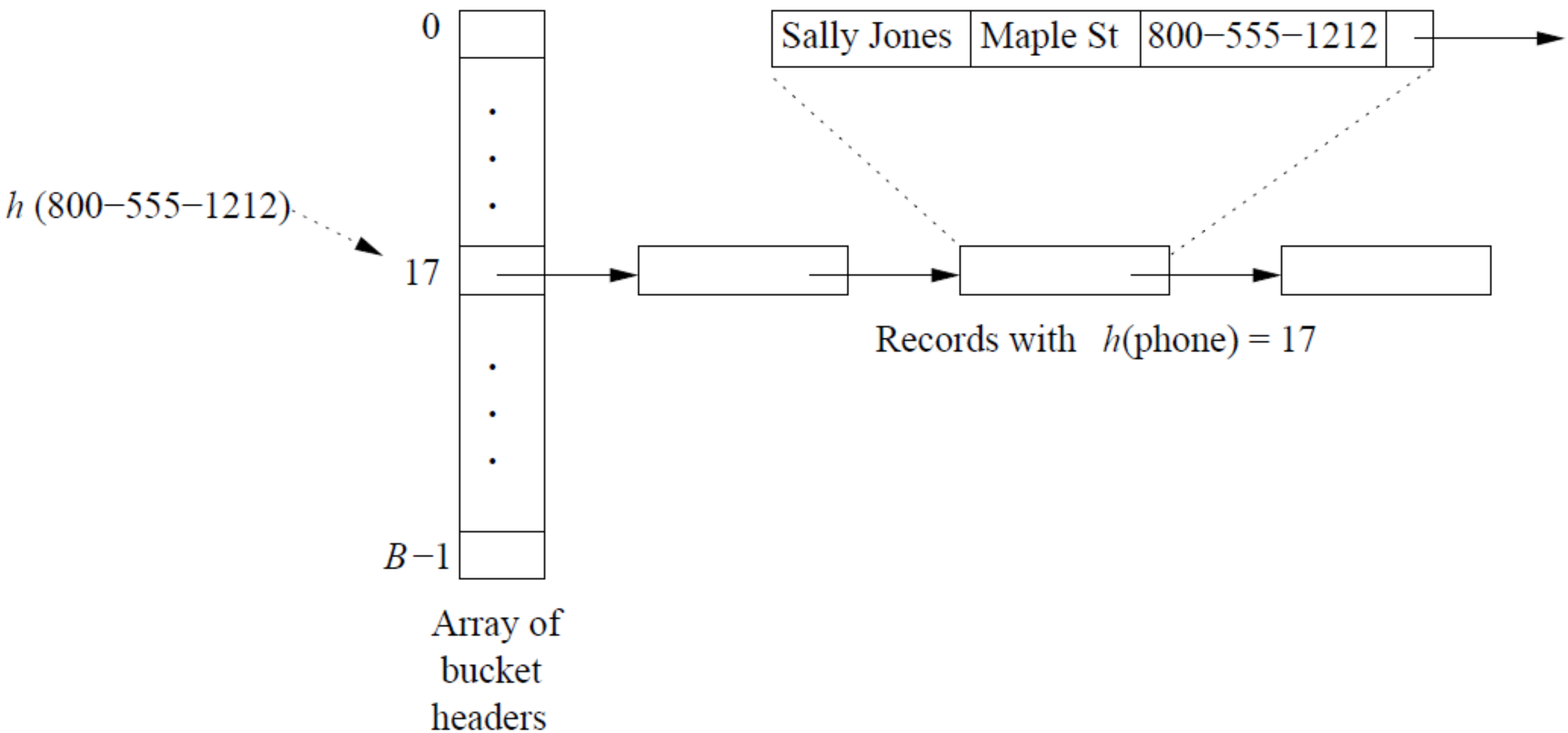
3. Indexes (1/2)

- A data structure that makes it efficient to retrieve objects given the value of one or more elements of those objects
  - (ex) given a value v for a field, an index on the field lets us retrieve all the records with value v in that field

- There are many ways to implement indexes
  - (ex) Hash table
    - The value v of the field of a record is hashed by a hash function h
    - The record is placed in the bucket whose number is $h(v)$
    - To find the records with a given value v, we need to search only the bucket whose number is $h(v)$

3. Indexes (2/2)

- (ex) a hash table as an index
  - Phone numbers are hashed to buckets
  - The entire record is placed in the bucket whose number is the hash value of the phone number



4. Secondary Storage (1/2)

- Secondary storage = storage on ***disk***
- The time taken to perform computations when the data is initially on disk is ***different*** from the time needed if the data is initially in main memory
- Disks are organized into ***blocks***
  - Blocks: the minimum units that the operating system uses to move data between main memory and disk
- The time taken to read a block from disk vs. main memory
  - 10^{-3} seconds vs. 10^{-8} seconds

4. Secondary Storage (2/2)

- Usually, the time taken to move a block from disk to main memory is ***far larger*** than the time taken to do the computation

- When your dataset is larger than 100 GB or 1 TB, just accessing it presents problems, let alone doing anything useful with it

Summary of Chapter 1

- Data mining
- Bonferroni's Principle
- TF.IDF
- Hash functions
- Indexes
- Secondary storage