Introduction to Data Mining

Preliminaries

U Kang
Seoul National University

In This Lecture

- Learn basic tools for data mining
  - Text analysis
  - Searching
  - Storage
  - Approximation
  - Power law

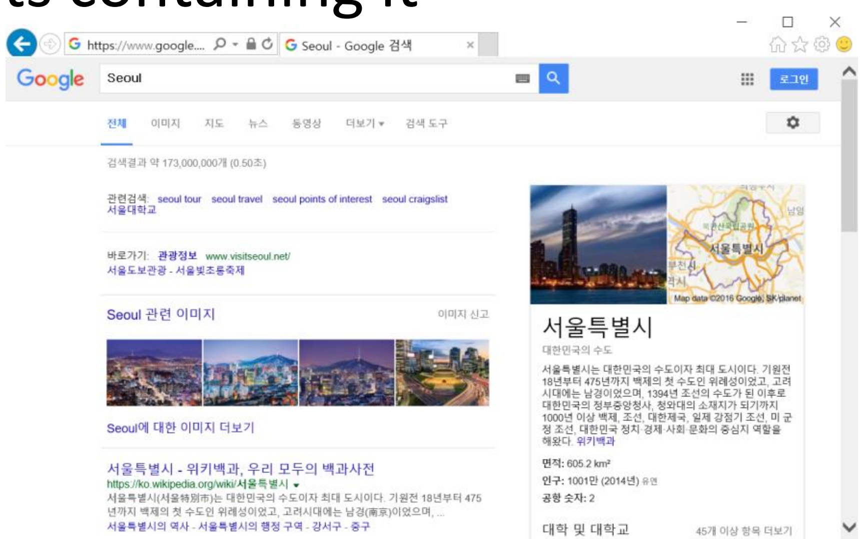
- These tools will help you better analyze data

Outline

- **Preliminaries**
  - Importance of Words in Documents
  - Hash Functions
  - Index
  - Secondary Storage
  - Base of Natural Log
  - Power Law

Importance of Words in Documents

- **How important is a word to a document?**
  - E.g., “ball”, “bat”, “pitch”, “run” in a document related to baseball
- Application: Search Engine
  - Given a query word “Seoul”, how to rank 173 million documents containing it



Importance of Words in Documents

- **How important is a word to a document?**
  - E.g., “ball”, “bat”, “pitch”, “run” in a document related to baseball
- **The most famous measure is TF.IDF**
  - Main idea 1 (TF) : a word is important to a document if the word occurs frequently
    - What about words like “a”, “the”, …?
  - Main idea 2 (IDF) : a word is important to a document if it occurs ***only in the document***

Importance of Words in Documents

- **Term Frequency (TF)**
  - Let f_{ij} be the frequency of term i in document j
  - $TF_{ij} = \frac{f_{ij}}{max_k f_{kj}}$
- **Inverse Document Frequency (IDF)**
  - Suppose term i appears in n_i of N documents
  - $IDF_i = log_2(\frac{N}{n_i})$
- TF.IDF score of term i in doc. j = $TF_{ij} \times IDF_i$

Hash Functions

- **Hash function**
  - Takes a key as an input, and outputs a bucket number in the range of 0 ~ B-1 (B: total # of buckets)
  - E.g. h(x) = x mod 19
- Why do we need it?
  - Typically, hash function is used for quickly finding an item of interest (=indexing, to be explained soon)

Hash Functions

- **Good hash function?**
  - A function which sends approximately equal numbers of hash-keys to each of the B buckets
  - E.g.) modular hash function h(x) = x mod k
  - Assume x = 2, 4, 6, 8, 10, 12, ….
  - What if k = 10?
  - What if k = 11?

Hash Functions

- **Good hash function?**
  - A function which sends approximately equal numbers of hash-keys to each of the B buckets
  - E.g.) modular hash function h(x) = x mod k
  - Assume x = 2, 4, 6, 8, 10, 12, ….
  - What if k = 10?
  - What if k = 11?

  - It's best to choose a prime number for k

Index

- **Problem**
  - Assume we are given a file of (name, address, phone) triples
  - Given a phone number, how can we find out the name and address of the person quickly, without scanning all the contents of the file?
- Answer: index

Index

- **Index**
  - A data structure that makes it efficient to retrieve objects given the value of one or more elements of the objects
  - Several ways to build an index
    - Hash table, B-tree, …

Index

- **Index**
  - Example of an index based on hash-table

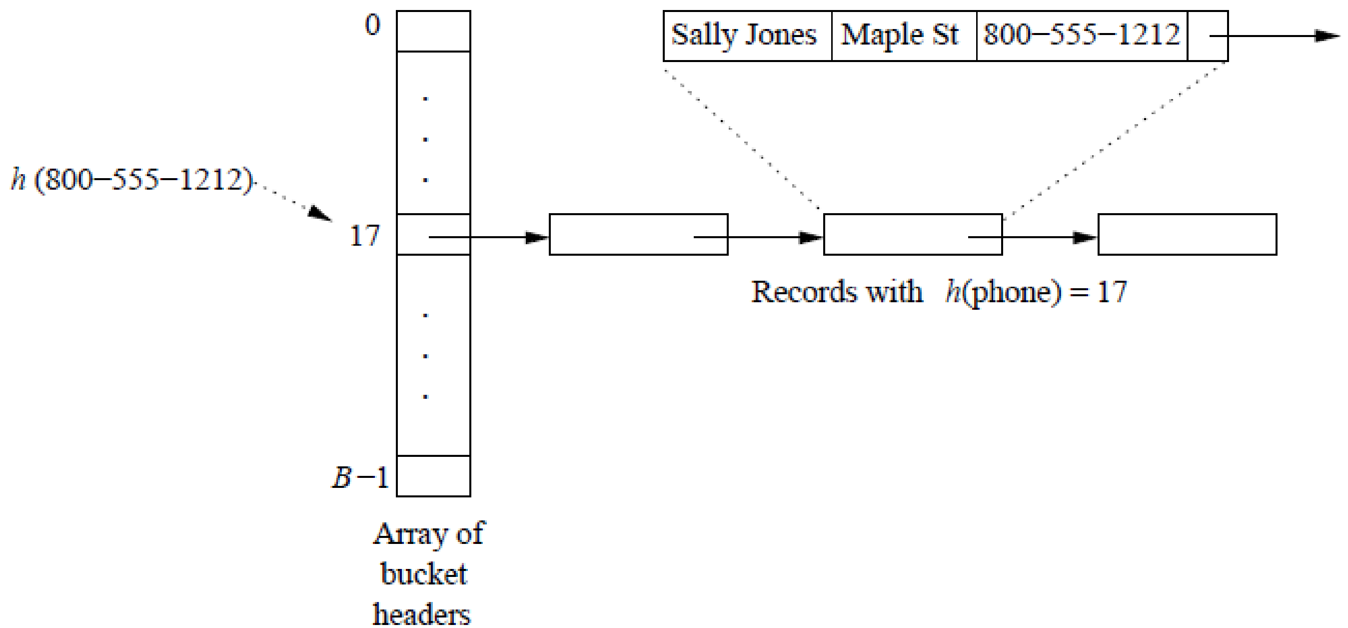


Figure 1.2: A hash table used as an index; phone numbers are hashed to buckets, and the entire record is placed in the bucket whose number is the hash value of the phone

Secondary Storage

- **Memory vs. Disk**
  - Price, Speed, Capacity
- Disk
  - Organized into blocks (=minimum units that OS uses to move data between main memory and disk)
  - Typical block size ~ 4 Kbytes
  - Time to access and read a block: ~ 10 milliseconds
  - Sequential access is much faster (~ 10^5 times) than random access

Base of Natural Logarithms

- e = 2.7182818… = $\lim_{x\to\infty}(1+\frac{1}{x})^x$
- Using the above fact, we can obtain useful approximations
  - $(1+a)^b = (1+a)^{\frac{1}{a}ab} \sim e^{ab}$
- Similarly, $\lim_{x\to\infty}(1-\frac{1}{x})^x = e^{-1}$
  - $(1-a)^b = (1-a)^{\frac{1}{a}ab} \sim e^{-ab}$

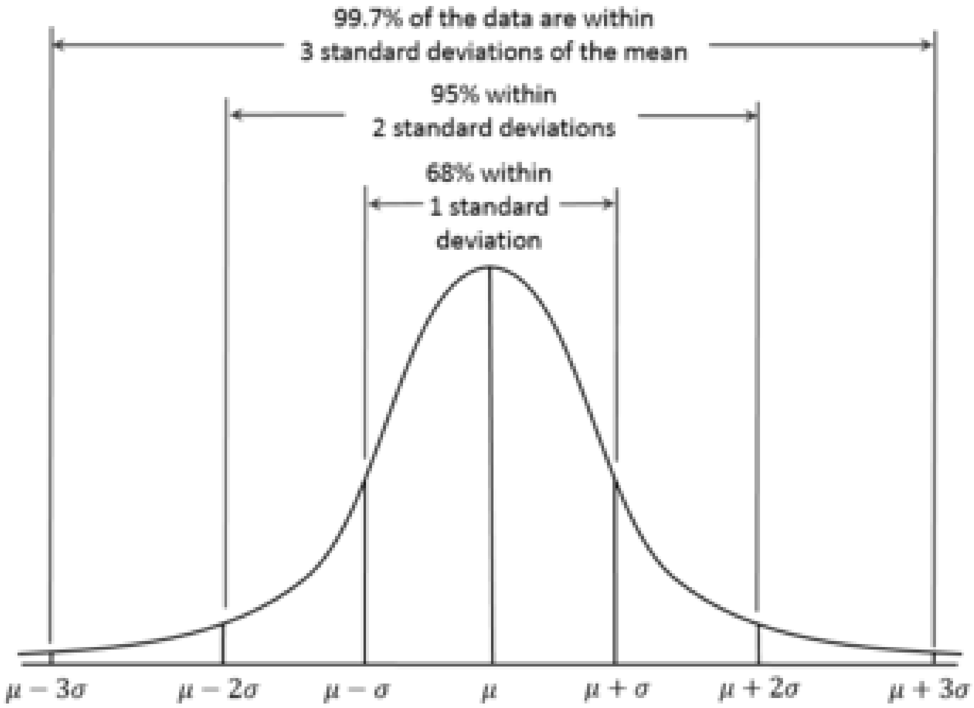
These approximations work well when a is small

Base of Natural Logarithms

- $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots$

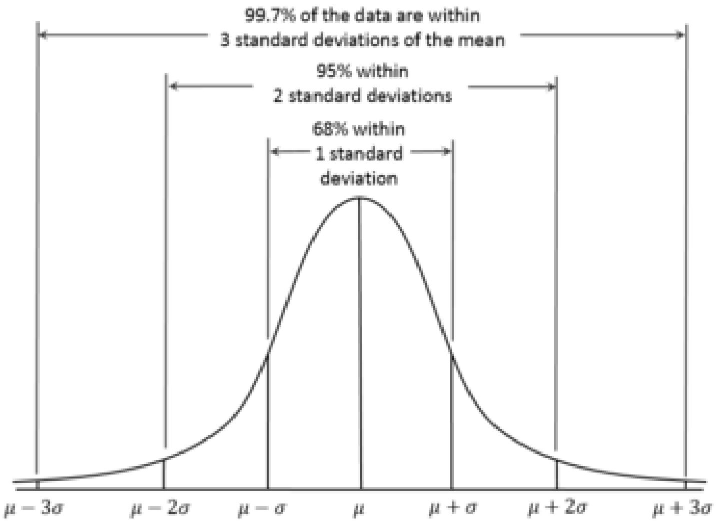
Power Laws

- Assume that students have 70.0 average score in an exam. What would be the distribution of scores?
  - You would answer this with Normal distribution

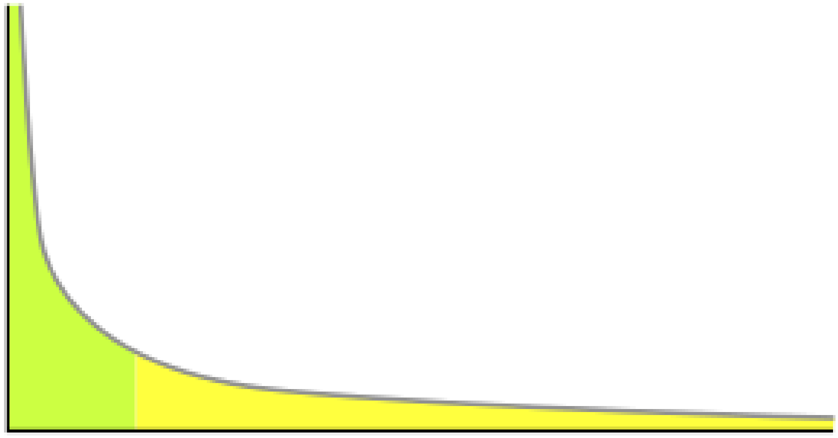


Power Laws

- Average # of friends in Facebook at 2014 is ~300. What would be the distribution of # of friends?



Gaussian



Power law

Power Laws

- Linear relationship between the logarithms of two variables

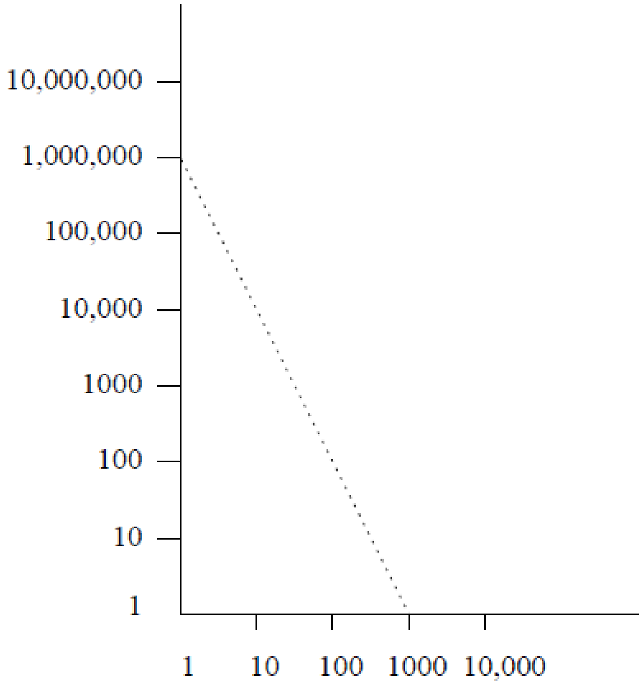


Figure 1.3: A power law with a slope of −2

Power Laws

- What about in linear scale?

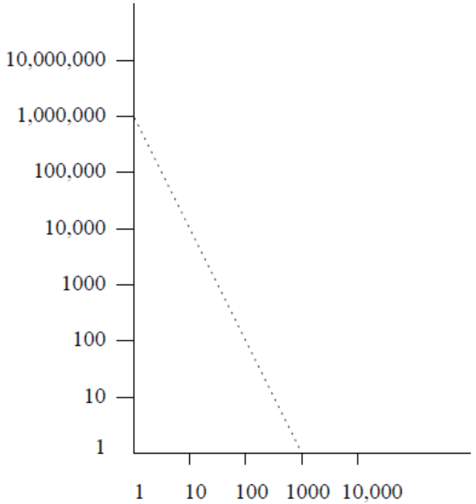
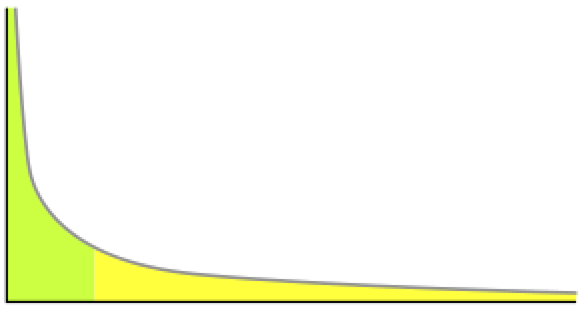
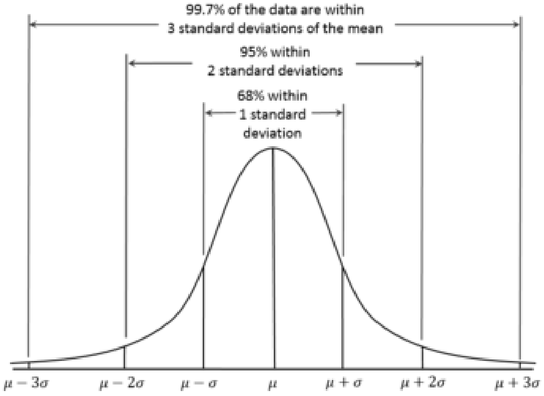


Figure 1.3: A power law with a slope of −2



Vs.



Power law distribution (Log-Log scale)

Power law distribution (Linear scale)

Gaussian distribution (Linear scale)

Power Laws

- In general, x and y are in a power law relationship if log y is linear to log x
  - (log y) = b + a (log x)

    $\Leftrightarrow y = e^{b} x^{a} = c x^{a}$

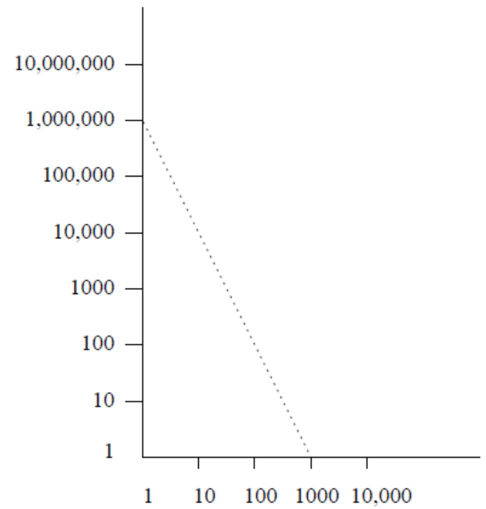


Figure 1.3: A power law with a slope of −2

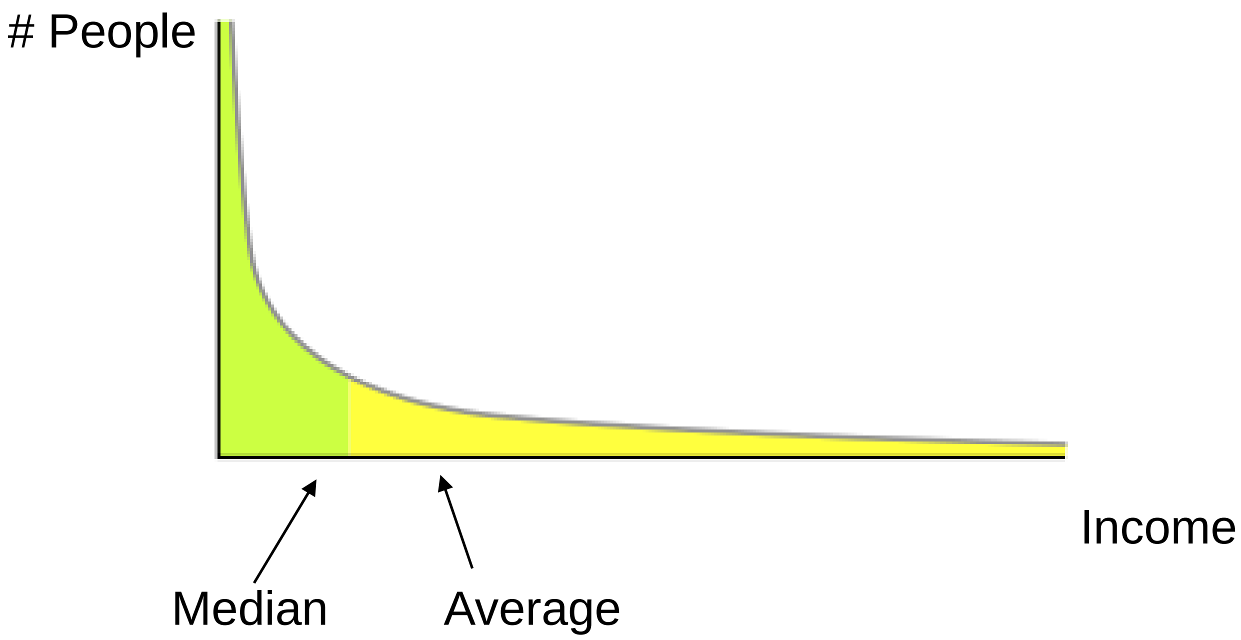
Power Laws

- Why is power-law important?
  - It helps better understand the characteristic of real world data
  - "Matthew Effect" : the rich gets richer
  - E.g.) If a person is popular in a social network, she/he will get more popular in the future

Barack Obama's Facebook

Power Laws

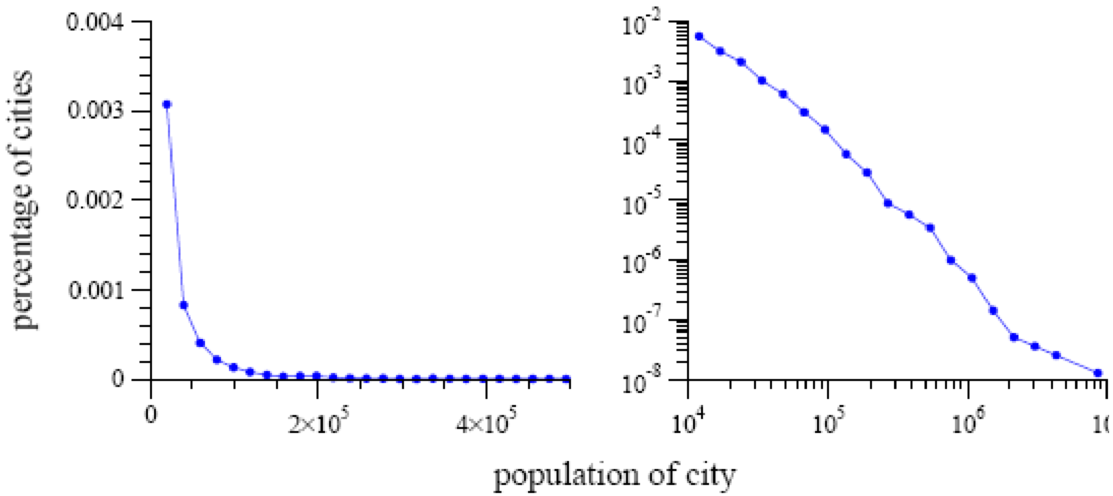
- Why is power-law important?



Why do governments like to report average?

Power Laws

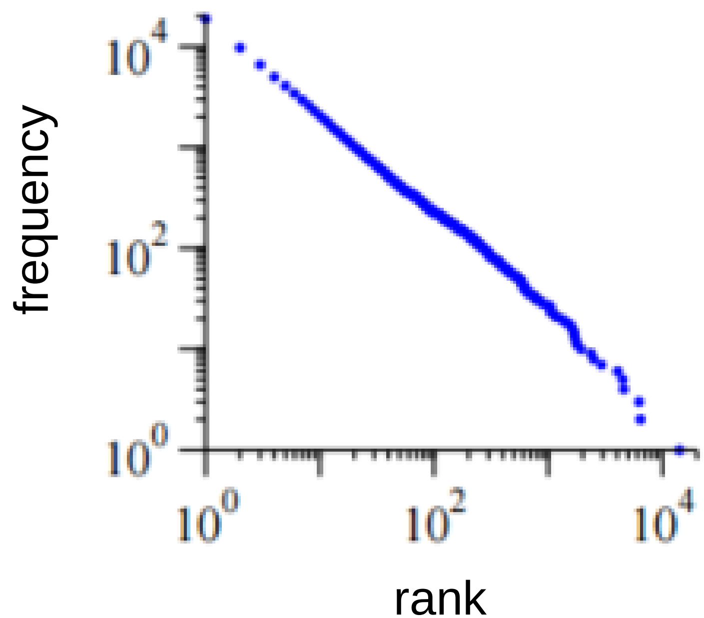
- Examples of Power Laws
  - Node Degrees in the Web Graph
  - Sales of Products
  - Sizes of Web sites
  - Population of cities



[Mark Newman] Power laws, Pareto distributions and Zipf's Law, 2005

Power Laws

- Examples of Power Laws
  - Zipf's Law: $y = cx^{-1/2}$
    - Word frequencies in text



[Mark Newman] Power laws, Pareto distributions and Zipf's Law, 2005

What You Need to Know

- How to measure the importance of words in documents
  - TF/IDF
- Hash functions: definition, and how to design a good hash function
- Index: search data quickly
- Memory vs. disk in terms of price, speed, and capacity
- Approximations
- Power law: powerful tool to understand data

Questions?