# DATA EXPLORATION

Xin Li
xin.li@helsinki.fi

# DATA EXPLORATION

- Introduction
- Three facets of data exploration
  - User Interaction
  - Middleware
  - Database Engine
- An example of data exploration
- Integrating data mining

# DEFINITION

- Data exploration is about efficiently extracting knowledge from data even if we do not know exactly what we are looking for.

- With exploratory data analysis the researcher explores the data in many possible ways, including the use of graphical tools like boxplots or histograms, gaining knowledge from the way data are displayed

# PURPOSE

- Helping users to make sense of huge amount of data

- Enable non-professional users to analyze the data

# TRADITIONAL SYSTEMS

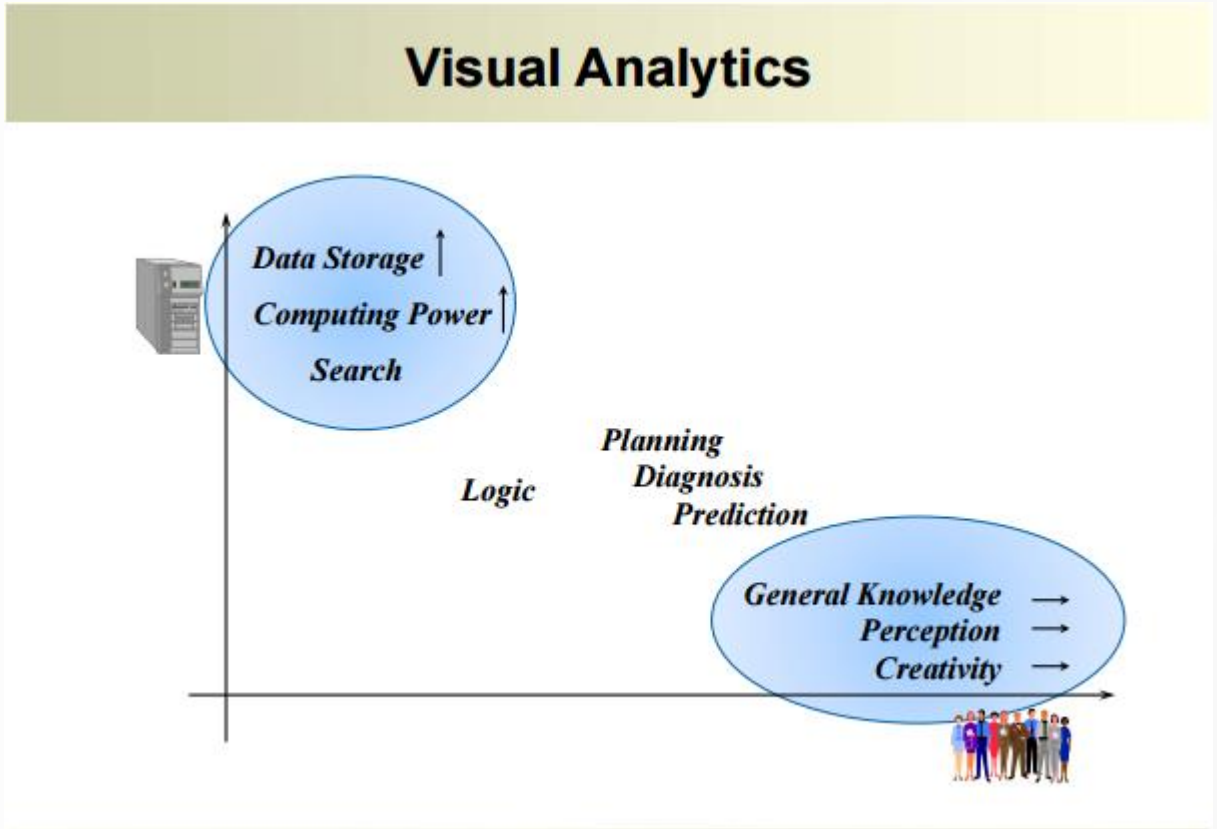
- Users have a good understanding of the structure and contents of the database.
- Users are certain about what they want and the results when they pose a particular query.
- Designed for static scenario where tuning phase is needed for administrator to tune the configuration for the expected workload.

# DATA-DRIVEN APPLICATIONS

- Increasing amount of data.
- Dynamic data-driven applications that address different requirements from traditional DBMS.
- The exploration process is like a session between user and system where several queries could be involved.

# USER INTERACTION

- User Interface Layer: The goal is to assist users, who are not database experts, to explore the data sets.




http://in.bgu.ac.il/en/engn/ise/dmbi2014/Documents/Exploring%20Big%20Data%20using%20Visual%20Analytics.pdf

# QUERY RESULT VISUALIZATION

- Visualization tools that assist users in navigating the underlying data structures.
- Incorporate new types of interactions such as collaborative annotations and searches as well as recommendations of visualizations.
- Optimization techniques that aim to support large-scale visual analytics.

# EXPLORATION INTERFACES

- Automate the data exploration process by discovering relevant data to user.
- Assist users to formulate their exploratory queries.
- Query learning based on example tuples and solutions for tuning imprecise queries.
- SQL recommendation and data segmentation.

# MIDDLEWARE

- Middleware is between the user interaction layer and the database engine.
- Assist users to formulate their exploratory queries.
- Query learning based on example tuples and solutions for tuning imprecise queries.
- Improve the exploratory properties of existing systems without changing the underlying architecture.

# DATA PREFETCHING

- Reduce the overall exploration time.
- Multidimensional windows, data cubes, and spatial queries.
- Trade-off between introducing new results and re-using cached ones and optimization methods for diverse query results.
- Improve the exploratory properties of existing systems without changing the underlying architecture.

# QUERY APPROXIMATION

- Offer approximate results to get a quick glance at whether a particular query reveals interesting information.

- Achieve fast response by processing queries on sampled data.

# DATABASE LAYER

- Reconsiders the fundamental methods to store and access data to match exploration patterns.
- Achieve fast response by processing queries on sampled data.
- Adaptive indexing, adaptive loading, adaptive storage, and flexible architectures

# ADAPTIVE INDEXING

- Creating indexes incrementally and adaptively during query processing.
- As more queries arrive indexes are continuously fine-tuned.
- Adaptive indexing has been studied for supporting exploration in time-series processing and in Hadoop.

# ADAPTIVE LOADING

- Start querying even before all data is loaded since not all data are used for the query.

# ADAPTIVE STORAGE

- Traditional systems rely on particular static layouts and build the whole architecture based on it since there is no perfect layout for all systems instead a particular layout may match a system perfectly.

- In data exploration scenario, no prior decisions about what is a good layout is made.

# FLEXIBLE ARCHITECTURES

- Flexible database architectures where dynamically tune the system according to tasks is possible.

- Incorporate sampling feature for query approximation inside the core of data engine.

# AN EXAMPLE OF DATA EXPLORATION

- (Buoncristiano et al., 2015) introduced a conversation-like model for exploratory computing.
- A fitness tracker application.
- Technically, the conversation model is built as a lattice of nodes that are actually views of database. Each view is a representation of a conjunctive query in the database.

# AN EXAMPLE OF DATA EXPLORATION

($i$) a AcmeUser(id, name, sex, age, cityId) table with user data;

($ii$) a Location(id, cityName, state, region) table to record location data about users (here region may be east, west, north or south);

($iii$) an Activity(id, type, date, start, length, userId) table to record step counts for user activities of the various kinds (like walks, runs, cycling etc.);

($iv$) a Sleep(id, date, start, length, quality, userId) table to record user sleep and its quality (like deep sleep, restless etc.).

($\mathcal{S}_1$) "It might be interesting to explore the types of activities. In fact: *running* is the most frequent activity (over 50%), *cycling* the least frequent one (less than 20%)";

($\mathcal{S}_2$) "It might be interesting to explore the sex of users with running activities. In fact: more than 65% of the runners are male";

($\mathcal{S}_3$) "It might be interesting to explore differences in the distribution of the length of the running activities between male and female. In fact: male users generally have longer running activities".

Buoncristiano, M., Mecca, G., Quintarelli, E., Roveri, M., Santoro, D. and Tanca, L., 2015. Database challenges for exploratory computing. *ACM SIGMOD Record*, *44*(2), pp.17-22.

# DATA MINING PERSPECTIVE

- Data mining concentrates on discovering interesting frequent patterns and rule generation, whose core idea partly overlap with data exploration.
- Integrating data mining algorithms into data exploration.
- How to handle the computational overhead is a crucial issue.

# THANK YOU!