CMU SCS

15-826: Multimedia Databases and Data Mining

Lecture#1: Introduction
Christos Faloutsos
CMU
www.cs.cmu.edu/~christos

CMU SCS

Outline

Goal: 'Find similar / interesting things'
- Intro to DB
- Indexing - similarity search
- Data Mining

15-826   Copyright: C. Faloutsos (2013)   2

CMU SCS

Problem

Given a large collection of (multimedia) records, or graphs, find similar/interesting things, ie:
- Allow fast, approximate queries, and
- Find rules/patterns

15-826   Copyright: C. Faloutsos (2013)   3

CMU SCS

Sample queries

- Similarity search
  - Find pairs of branches with similar sales patterns
  - find medical cases similar to Smith's
  - Find pairs of sensor series that move in sync
  - Find shapes like a spark-plug
  - (nn: 'case based reasoning')

15-826   Copyright: C. Faloutsos (2013)   4

CMU SCS

Sample queries –cont'd

- Rule discovery
  - Clusters (of branches; of sensor data; ...)
  - Forecasting (total sales for next year?)
  - Outliers (eg., unexpected part failures; fraud detection)

15-826   Copyright: C. Faloutsos (2013)   5

CMU SCS

Outline

Goal: 'Find similar / interesting things'
- Intro to DB
- Indexing - similarity search
- Data Mining

15-826   Copyright: C. Faloutsos (2013)   6

Detailed Outline

Intro to DB
- Relational DBMS - what and why?
  - inserting, retrieving and summarizing data
  - views; security/privacy
  - (concurrency control and recovery)

15-826   Copyright: C. Faloutsos (2013)   7

What is the goal of rel. DBMSs

15-826   Copyright: C. Faloutsos (2013)   8

What is the goal of rel. DBMSs

Electronic record-keeping:

Fast and convenient access to information.

Eg.: students, taking classes, obtaining grades;

- find my gpa
- <and other ad-hoc queries>

15-826   Copyright: C. Faloutsos (2013)   9

CMU SCS

Why Databases?

 10

CMU SCS

Why Databases?

- Flexibility
- data independence (can add new tables; new attributes)
- data sharing/concurrency control
- recovery

 11

CMU SCS

Why NOT Databases?

 12

CMU SCS

Why NOT Databases?

- Price
- additional expertise (SQL/DBA)
- over-kill for small data sets

15-826   Copyright: C. Faloutsos (2013)   13

CMU SCS

Main vendors/products

Commercial   Open source

15-826   Copyright: C. Faloutsos (2013)   14

CMU SCS

Main vendors/products

Commercial   Open source
- Oracle   Postgres (UCB)
- IBM/DB2   mySQL, sqlite,
- MS SQL-server   miniBase (Wisc)
- Sybase   (www.sigmod.org)
- (MS Access,
- ...)

15-826   Copyright: C. Faloutsos (2013)   15

CMU SCS

Detailed Outline

Intro to DB
- Relational DBMS - what and why?
  - inserting, retrieving and **summarizing** data
  - views; security/privacy
  - (concurrency control and recovery)

15-826   Copyright: C. Faloutsos (2013)   16

CMU SCS

How do DBs work?

We use **sqlite3** as an example, from
http://www.sqlite.org

15-826   Copyright: C. Faloutsos (2013)   17

CMU SCS

How do DBs work?

%sqlite3 mydb   # mydb: file
sql>create table student (
    ssn fixed;
    name char(20));

student	
ssn	name

15-826   Copyright: C. Faloutsos (2013)   18

CMU SCS

How do DBs work?

sql>insert into student
    values (123, "Smith");

student	
ssn	**name**
123	Smith

sql>select * from student;

15-826   Copyright: C. Faloutsos (2013)   19

CMU SCS

How do DBs work?

sql>create table takes (
    ssn fixed,
    c_id char(5),
    grade fixed));

takes		
ssn	**c_id**	**grade**

15-826   Copyright: C. Faloutsos (2013)   20

CMU SCS

How do DBs work - cont'd

More than one tables - joins
Eg., roster (names only) for 15-826

student	
ssn	**name**

takes		
ssn	**c_id**	**grade**

15-826   Copyright: C. Faloutsos (2013)   21

CMU SCS

How do DBs work - cont'd

sql> select name
    from student, takes
    where student.ssn = takes.ssn
    and takes.c_id = "15826"

15-826   Copyright: C. Faloutsos (2013)   22

CMU SCS

SQL-DML

General form:
    select a1, a2, … an
    from r1, r2, … rm
    where P
    [**order by** ….]
    [**group by** …]
    [**having** …]

15-826   Copyright: C. Faloutsos (2013)   23

CMU SCS

Aggregation

Find ssn and GPA for each student

student	
ssn	name

takes		
ssn	c_id	grade
123	603	4
123	412	3
234	603	3

15-826   Copyright: C. Faloutsos (2013)   24

CMU SCS

Aggregation

DM!

sql> select ssn, avg(grade)
    from takes
    group by ssn;

takes

ssn	c_id	grade
123	603	4
123	412	3
234	603	3

ssn	avg(grade)
123	3.5
234	3

15-826   Copyright: C. Faloutsos (2013)   25

CMU SCS

Detailed Outline

Intro to DB
- Relational DBMS - what and why?
  – inserting, retrieving and summarizing data
  – views; security/privacy
  – (concurrency control and recovery)

15-826   Copyright: C. Faloutsos (2013)   26

CMU SCS

Views - what and why?

- suppose you ONLY want to see ssn and GPA (eg., in your data-warehouse)
- suppose secy is only allowed to see GPAs, but not individual grades
- (or, suppose you want to create a short-hand for a query you ask again and again)
- -> VIEWS!

15-826   Copyright: C. Faloutsos (2013)   27

CMU SCS

Views

sql> create view fellowship as (
     select ssn, avg(grade)
     from takes   group by ssn);

takes

ssn	c_id	grade
123	603	4
123	412	3
234	603	3

ssn	avg(grade)
123	3.5
234	3

15-826   Copyright: C. Faloutsos (2013)   28

CMU SCS

Views

sql> create view fellowship as (
     select ssn, avg(grade)
     from takes   group by ssn);

takes

ssn	c_id	grade
123	603	4
123	412	3
234	603	3

ssn	avg(grade)
123	3.5
234	3

15-826   Copyright: C. Faloutsos (2013)   29

CMU SCS

Views

Views = 'virtual tables'

15-826   Copyright: C. Faloutsos (2013)   30

Views

sql> select * from fellowship;

takes

ssn	c_id	grade
123	603	4
123	412	3
234	603	3

ssn	avg(grade)
123	3.5
234	3

15-826   Copyright: C. Faloutsos (2013)   31

Views

sql> grant select on fellowship to secy;

takes

ssn	c_id	grade
123	603	4
123	412	3
234	603	3

ssn	avg(grade)
123	3.5
234	3

15-826   Copyright: C. Faloutsos (2013)   32

Detailed Outline

Intro to DB
- Relational DBMS - what and why?
  - inserting, retrieving and summarizing data
  - views; security/privacy
  - (concurrency control and recovery)
- What if slow?
- Conclusions

15-826   Copyright: C. Faloutsos (2013)   33

CMU SCS

What if slow?

sqlite> select * from irs_table where
  ssn='123';

Q: What to do, if it takes 2hours?

15-826   Copyright: C. Faloutsos (2013)   34

CMU SCS

DM!

What if slow?

sqlite> select * from irs_table where
  ssn='123';

Q: What to do, if it takes 2hours?
A: build an index
  Q': on what attribute?
  Q'': what syntax?

15-826   Copyright: C. Faloutsos (2013)   35

CMU SCS

What if slow - #2?

sqlite> create table friends (p1, p2);

Facebook-style: find the 2-step-away people

15-826   Copyright: C. Faloutsos (2013)   36

CMU SCS

What if slow - #2?

sqlite> create table friends (p1, p2);
sqlite> select f1.p1, f2.p2
  from friends f1, friends f2
  where f1.p2 = f2.p1;

Q: too slow – now what?

CMU SCS

DM!

What if slow - #2?

sqlite> create table friends (p1, p2);
sqlite> select f1.p1, f2.p2
  from friends f1, friends f2
  where f1.p2 = f2.p1;

Q: too slow – now what?
A: '**explain**': sqlite> explain select ….

CMU SCS

Long term answer:

- Check the query optimizer (see, say, Ramakrishnan + Gehrke 3rd edition, chapter15)

CMU SCS

Conclusions

- (relational) DBMSs: electronic record keepers
- customize them with **create table** commands
- ask SQL queries to retrieve info

Copyright: C. Faloutsos (2013)   40

CMU SCS

Conclusions cont'd

main advantages over flat files & scripts:
- logical + physical data independence (ie., flexibility of adding new attributes, new tables and indices)
- (concurrency control and recovery for free)

Copyright: C. Faloutsos (2013)   41

CMU SCS

Conclusions cont'd

D.M. practitioner's guide
- Data mining: `group by` + aggregates
- If a query runs slow:
  - `explain select` – to see what happens
  - `create index` – often speeds up queries

Copyright: C. Faloutsos (2013)   42

CMU SCS

For more info:

- Sqlite3: www.sqlite.org - @ linux.andrew
- postgres:
  http://www.postgresql.org/docs/
- Microsoft Access: available on andrew
  clusters (PC)
- Ramakrishnan + Gehrke, 3rd edition
- 15-415 web page, eg,
  – http://www.cs.cmu.edu/~christos/courses/dbms.S13/

15-826   Copyright: C. Faloutsos (2013)   43