Today

We will introduce the idea of a relational database, discuss its underlying data model and present a slightly simplified view of how to access its information.

As with all new technologies in this class, we will consider databases in the context of an applied problem, the so-called Reality Mining project at the MIT Media Lab.

But, keeping with the theme of the last few weeks, we will also use R as our “window” into the inner-workings of databases.

Reality Mining

The project starts with the idea that just about everyone has or will have a mobile phone; these devices act as a kind of wearable sensor.

When using a mobile phone, your location is roughly known because you “associate” (to borrow the language from a previous project) with a cell tower.

In addition, many mobile phones have the ability to other local wireless devices via the Bluetooth protocol; this means that it is possible to passively track social interactions and exhibit daily patterns of contact.

Last time

We introduced some tools for debugging functions written in R; the facilities range from simply throwing exceptions to a more informative browsing tool.

We then considered the performance of code in terms of either space or time, and illustrated simple tools to help compare programming options.

Finally, we looked at the structure of packages in R and talked a bit about software licenses.
Reality Mining

In addition, mobile phones provide information about users’ communication patterns: Who did you call? Who did you text?

Here, for example, is a representation of user locations with links indicating current calls.

We can also collect information about the device itself; Is it idle? Is the battery charging?

Bluetooth

This protocol was introduced by Ericsson in 1994 and by 2006 is estimated to be available on about 90% of PDAs, 80% of laptops and 75% of mobile phones.

You might be own Bluetooth wireless headset, keyboard, or mouse; the protocol was initially designed to form ad hoc networks so that nearby devices could cooperate in some sense.

One feature of this protocol is something called device discovery; a Bluetooth phone can identify information on other Bluetooth devices within 5-10 meters.

The Media Lab group created a small piece of monitoring software that runs on certain mobile phones and records data on devices carried by people nearby.

The data

The Reality Mining study consisted of tracking 100 people for 9 months using specially prepared Nokia 6600 smart phones.

75 of the users were students or faculty in the Media Lab, while the remaining 25 come from the Sloan Business School.

Researchers collected call logs, records of Bluetooth devices nearby and the cell towers the user “associated” with, and statistics related to application usage and phone status.

Machine Perception and Learning of Complex Social Systems

Reality Mining defines the collection of machine-sensed environmental data pertaining to human social behavior. This new paradigm of data mining makes possible the modeling of conversation context, proximity sensing, and temporospatial location throughout large communities of individuals. Mobile phones (and similarly innocuous devices) are used for data collection, opening social network analysis to new methods of empirical stochastic modeling.

The original Reality Mining experiment is one of the largest mobile phone projects attempted in academia. Our research agenda takes advantage of the increasingly widespread use of mobile phones to provide insight into the dynamics of both individual and group behavior. By leveraging recent advances in machine learning we are building generative models that can be used to predict what a single user will do next, as well as model behavior of large organizations.

We have captured communication, proximity, location, and activity information from 100 subjects at MIT over the course of the 2004-2005 academic year. This data represents over 350,000 hours (~40 years) of continuous data on human behavior. Such rich data on complex social systems have implications for a variety of fields. The research questions we are addressing include:

- How do social networks evolve over time?
- How entropic (predictable) are most people’s lives?
- How does information flow?
- Can the topology of a social network be inferred from only proximity data?
- How can we change a group’s interactions to promote better functioning?

If you have a Nokia Symbian Series 60 Phone (such as the Nokia 6600) with a data plan, you can participate. Additionally, we have cleaned the 2004-2005 data of identifiable information and are making it available to other researchers within the academic community. Both the mobile phone application and the resultant dataset can be downloaded here.
The data

So far, we haven’t talked very much about formal methods for data organization; we mainly kept to flat files, scraped through data on the disk, or (at best) data frames in R.

In industry, in large (and even not-so-large) applications, you will be called upon to access data from relational databases; these are often commercial systems from vendors like Oracle or even SAS.

The data from the Reality Mining project were provided to us in the form of a MySQL database.

What is a database?

- an organized body of related information
- A database is a collection of information stored in a computer in a systematic way, such that a computer program can consult it to answer questions. The software used to manage and query a database is known as a database management system (DBMS). The properties of database systems are studied in information science.
- Data stored on computer files or on CD-ROM. A database may contain bibliographic, textual or numeric data. The data are usually structured so that they may be searched in a number of ways. A variety of databases is accessible via this website.
- A database is an organised collection of information records that can be accessed electronically.
- is an organized collection of information stored on a computer.
- A database is a collection of data that is organized so that its contents can easily be accessed, managed and updated.
- A collection of information that has been systematically organized for easy access and analysis. Databases typically are computerized.
- A collection of information arranged into individual records to be searched by computer.
- Any organized collection of information; it may be paper or electronic.
- a standardized collection of information in computerized format, searchable by various parameters; in libraries often refers to electronic catalogs and indexes.
- A collection of electronic records having a standardized format and using specific software for computer access.
- A collection of information organized and presented to serve a specific purpose. A computerized database is an updated, organized file of machine readable information that is rapidly searched and retrieved by computer.
- A set of data that is structured and organized for quick access to specific information.

Benefits of a database

- The data can be shared and accessed by many users; it is also possible to regulate the kind of access granted to each user, introducing a layer of security.
- Redundancy is reduced in the sense that not everyone has to have their own private copy of the data; as a corollary we also have the opportunity to reduce inconsistencies in the data and maintain better control over data integrity.
- Finally, the act of creating a database and deciding on how data are to be represented often forces discussions by the users about what services they require from the database; it also allows for the introduction of data standards which could enable the interchange of data with other systems and organizations.

MySQL

At a technical level, MySQL is a multithreaded, multi-user, structured query language (SQL) database management system (DBMS).

- It is distributed free under the GPL (remember our discussion about software licenses last time).
- It is owned and sponsored by a for-profit company that sells support and service contracts, as well as commercially-licensed copies of MySQL.
Why not load the data into R?

Just as flat files are not ideal for data organization and storage, R and its data frames are also lacking.

As a main memory program, there’s a definite limit to the size of your data.

R doesn’t support concurrent access to its data; rather, if more than one user is changing or deleting records, only that user “sees” the updates.

The format for data stored in R is somewhat specialized; it is not directly readable by other systems.

The relational model

A relational database contains multiple tables, each similar to the flat model; the word relation is borrowed from mathematics where it essentially means table.

Next, a series of operators exist through which users query the database, deriving new tables from old ones; for example, we might extract a subset of a given table, perform simple computations, or merge the data in several tables.

Data models

To ground our discussion somewhat, a database provides a kind of organizational structure to the data it contains and defines operations that can be performed on the data.

We can think of a data frame as having a simple (the simplest) “flat” data model; it is essentially a two-dimensional array of data elements where all members of a given column have similar values, and members of each row are related in some way.

Database access through R

There are three ingredients to accessing a relational database in R.

**The driver:** This facilitates the communication between the R session and the database management system (DBMS) by defining, say, data type mappings.

**The connection:** This object encapsulates the actual connection with the DBMS and carries out the requested queries.

**The result:** This tracks the status of a query, such as the number of rows that have been fetched and whether or not the query has completed.
Some terminology

Each table is called a relation; rows in the table are called tuples and columns are called attributes.

The degree of a table refers to the number of columns, and the cardinality the number of rows.

An entity is an abstraction of an object will be represented in the database; an instance is a particular occurrence of the object.

For the person relation, this means some facts about their phone and a series of survey results; in our database we have 97 instances, separate users.
Relational databases

In the Reality Mining example, we see several tables that are related in the sense that data in one refers might refer to data in another.

Keys are used to match up rows of data in different tables; a key is simply a collection of one or more attributes.

In our example, this structure is pretty simple...

DBMSs are optimized to handle a set of queries made by users; again, these queries essentially let us make new tables out of old by subsetting, merging or performing simple computations.

A query language allows users to interact with the database, reducing the data and summarizing it before retrieving the results.

The Structured Query Language (SQL) is widely used, and is supported by most commercial databases.
SQL

The SELECT statement is used to retrieve data from a database; you specify the table you want to draw from and various conditions you want to impose to extract a subset of the data

SELECT * FROM person;

This will return the complete person table and is equivalent to calling `dbReadTable` (Note that our SQL statements are always in caps; the language is not case sensitive, but we do so for readability)

SQL

We can understand some of what SELECT does by comparing it to subsetting on data frames; compare

SELECT name, person_oid FROM cellname
WHERE celltower_oid = 1;

and an operation on our R data frame

cell[cell$"celltower_oid"==1,
c("name","person_oid")]
The general form of this command is

```
SELECT column(s) FROM relation(s)
[WHERE constraints];
```

For example, we get closer to our familiar subsetting with statements like

```
SELECT * FROM callspan
WHERE duration > 1200 AND person_oid = 1;
```
SQL

Certain functions also exist to operate on the data before you retrieve it; COUNT, SUM, AVG, MIN, MAX

Additional clauses can be added to the statement, to function like our various flavors of apply

```
SELECT person_oid, AVG(duration)
FROM callspan GROUP BY person_oid;
```

This will find the average duration of calls for each ID

We can take it a step further with the following construction

```
SELECT person_oid, AVG(duration)
FROM callspan
GROUP BY person_oid
HAVING MAX(duration) < 10000;
```

This very very brief introduction is meant solely to whet your appetite for relational databases; entire year-long courses are devoted to this topic in CS

Our goal was simply to draw some parallels between this form of data organization and what we have been studying in class

We also note that R can be used to access data from a database; that is, R can serve as one of the many clients able to pull data from a database