From relational databases to linked data: R for the semantic web

Jose Quesada,
Max Planck Institute, Berlin
Who this talk targets

• You have big data; you use a database

• You have an evolving schema definition. Sometimes at runtime

• You are interested in alternative ways to present your data

• You would thrive by using data out there, if only they were more accessible
Semantic web
The **Semantic web**

- Ontology as Barad-dur (Sauron’s tower)
  - Extremely powerful
  - Patrolled by Orcs
    - Let one little hobbit in it, and the whole thing could come crashing down
  - OWL
The **Semantic web**

- Ontology as Barad-dur (Sauron’s tower)
  - Extremely powerful
  - Patrolled by Orcs
    - Let one little hobbit in it, and the whole thing could come crashing down
  - OWL
Inconsistency

1537 classes, 1 modeling error = failure!
The semantic web

- The tower of Babel
  - We will build a tower to reach the sky
  - We only need a little ontological agreement
    - Who cares if we all speak different languages?

This is RDFS
Statistics matter here
Web-scale
Lots of data; finding anything in the mess can be a win
Approaches to data representation

- Objects
- Tables (relational databases)
- Non-relational databases
- Tables (data.frame)
- Graphs
What one can do with semantic web data, now:

People that died in Nazi Germany and if possible, any notable works that they might have created

SELECT *
WHERE {
    OPTIONAL {
        ?subject dbpedia-owl:notableworks ?works
    }
}
<table>
<thead>
<tr>
<th>subject</th>
<th>works</th>
</tr>
</thead>
<tbody>
<tr>
<td>:Anne_Frank</td>
<td>:The_Diary_of_a_Young_Girl</td>
</tr>
<tr>
<td>:Martin_Bormann</td>
<td>-</td>
</tr>
<tr>
<td>:lr%C3%A8ne_N%C3%A9mirovsky</td>
<td>-</td>
</tr>
<tr>
<td>:Erich_Fellgiebel</td>
<td>-</td>
</tr>
<tr>
<td>:Friedrich_Ferdinand%2C_Duke_of_Schleswig-Holstein</td>
<td>-</td>
</tr>
<tr>
<td>:Friedrich_Olbricht</td>
<td>-</td>
</tr>
<tr>
<td>:Ludwig_Beck</td>
<td>-</td>
</tr>
<tr>
<td>:Erwin_Rommel</td>
<td>-</td>
</tr>
<tr>
<td>:Maurice_Bavaud</td>
<td>-</td>
</tr>
<tr>
<td>:Early_Years_of_Adolf_Hitler</td>
<td>-</td>
</tr>
<tr>
<td>:Emil_Zegad%C5%82owicz</td>
<td>-</td>
</tr>
<tr>
<td>:Friedrich_Fromm</td>
<td>-</td>
</tr>
<tr>
<td>:Helmuth_James_Graf_von_Moltk</td>
<td>-</td>
</tr>
</tbody>
</table>
• Scale to the entire web
• Do reasoning with open word assumption
• Retrieval in real-time
• Go beyond logics

• Use cases:
  – Real time city
  – Cancer monographs for WHO
  – Gene expression finding
RDF is a graph

• We have lots of interesting statistics that run on graphs

• In many Semantic Web (SW) domains a tremendous amount of statements (expressed as triples) might be true but, in a given domain, **only a small number of statements** is known to be true or **can be inferred to be true**. It thus makes sense to attempt to estimate the truth values of statements by **exploring regularities** in the SW data with machine learning
Scale

• You cannot use the entire thing at once: subsetting

• Are there patterns in knowledge structures that we can use for subsetting?
Idea

• Graph theory applied to subsetting large graphs

• Developing Semantic Web applications requires handling the RDF data model in a programming language

• Problem: current software is developed in the object-oriented paradigm, programming in RDF is currently triple-based.
Data

IMDB is a big graph:
- 1.4 m movies
- 1.7 m actors
- 11 M connections
  - Movies have votes
- Bipartite network

Packages: *igraph*:
- Nice functions that you cannot find anywhere else
- Uses Sparse Matrices
- Implemented in C
- Some support for bipartite networks

*Rmysql, Matrix (sparse m)*
# Centrality

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>$C_d(v) = \frac{\text{deg}(v)}{n-1}$</td>
<td>$O(E)$</td>
</tr>
<tr>
<td></td>
<td>Where $\text{deg}(v)$ is the number of connections that $v$ has.</td>
<td></td>
</tr>
<tr>
<td>Closeness</td>
<td>$C_v = \frac{</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>Where $d_{vi}$ is the distance between vertex $d$ and $i$ and $</td>
<td>v</td>
</tr>
<tr>
<td>Betweenness</td>
<td>$B_v = \sum_{i \neq j, i \neq v, j \neq v} g_{ivj}/g_{ij}$</td>
<td>$O(VE)$ time using Brandes’ (2001) algorithm; parallelizable (Bader &amp; Madduri, 2006).</td>
</tr>
<tr>
<td></td>
<td>Where $g_{ivj}$ is the number of shortest paths between $i$ and $j$ that pass through $v$, and $g_{ij}$ is the number of paths between $i$ and $j$ that do not go through $v$.</td>
<td></td>
</tr>
</tbody>
</table>
Centrality

**Eigenvector**

\[ E_v = \frac{1}{\lambda} A_{iv} E_i, \quad Ax = \lambda x \]

Where \( A \) is a matrix that represents a linear transformation, and \( \lambda \) is the scaling factor (eigenvalue) in the eigenvalue equation (right).

**PageRank**

\[ E_v = \frac{1 - d}{|V|} + d \sum_{i=1}^{|V|} A_{iv} E_i \]

\( d \) is the damping factor (set at .85 in our experiment, as in the original Brin and Page paper).

\( O(|E|\log(1/e)), \ e \) is the precision required.

The complexity is independent of the number of vertices (Bianchini, 2005, p. 100).
Pagerank

• The pagerank vector is the stationary distribution of a markov chain in a link matrix

• Some assumptions to warrant convergence

• The typical value of \( d \) is .85

\[
E_v = \frac{1 - d}{|V|} + d \sum_{i=1}^{|V|} A_{iv} E_i
\]

\[
A = \begin{bmatrix}
0 & 0 & 1 & \frac{1}{2} \\
\frac{1}{3} & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\
\frac{1}{3} & \frac{1}{2} & 0 & 0
\end{bmatrix}
\]

\[
norm \leftarrow \text{function}(x) \ x/\text{sum}(x)
\]

\[
norm(\text{eigen}(0.15/n\text{Vertices} + 0.85 * \text{t}(A))\text{vectors}[,1])
\]
Proportion of interest captured when subsetting by pagerank
Top movies by pageRank in the actor->movie network

<table>
<thead>
<tr>
<th>degree</th>
<th>pagerank</th>
<th>cluster</th>
<th>imdbID</th>
<th>title</th>
<th>rank</th>
<th>votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1298252192870</td>
<td>0.000243688</td>
<td>0</td>
<td>822609</td>
<td>Around the World in Eighty Days (1956)</td>
<td>40031</td>
<td>6134</td>
</tr>
<tr>
<td>313862390464</td>
<td>0.000103540</td>
<td>0</td>
<td>76352</td>
<td>&quot;Beyond Our Control&quot; (1968)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2910099912811</td>
<td>0.000091669</td>
<td>0</td>
<td>993780</td>
<td>Gone to Earth (1950)</td>
<td>7.0</td>
<td>291</td>
</tr>
<tr>
<td>2855923652847</td>
<td>0.000089025</td>
<td>0</td>
<td>915626</td>
<td>Deadlands 2: Trapped (2008)</td>
<td>39971</td>
<td>15</td>
</tr>
<tr>
<td>424328163772</td>
<td>0.000083882</td>
<td>0</td>
<td>1282574</td>
<td>Stuck on You (2003)</td>
<td>6.0</td>
<td>19709</td>
</tr>
<tr>
<td>6291101098043</td>
<td>0.000080824</td>
<td>0</td>
<td>622100</td>
<td>&quot;Shortland Street&quot; (1992)</td>
<td>39850</td>
<td>225</td>
</tr>
</tbody>
</table>
Problems

- Graphs have advantages over RDBMS/tables[1]. But we are used to think in tables.
- There is no direct way to handle RDF in R. Worth an R package?

ActiveRDF: Object-Oriented Semantic Web Programming

Eyal Oren  eyal.oren@deri.org
Renaud Delbru  renaud.delbru@deri.org
Sebastian Gerke  sebastian.gerke@deri.org
Armin Haller  armin.haller@deri.org
Stefan Decker  stefan.decker@deri.org
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
Linked data are out there for the grabs

We need to start thinking in terms of graphs, and slowly move away from tables

Thanks for your attention

Jose Quesada, quesada@workingcogs.com, http://josequesada.name
Twitter: @Quesada