Overview, Lecture I

Data Mining

What’s Data?

Record data, numerical data, data matrix, document data, graph data, chemical data, etc.

What’s Data Mining?

Why Data Mining?

Commercial viewpoint
Scientific viewpoint
Overview, Lecture I

Mining Data sets
  Association Rules
  Classification
  Clustering
  Forecasting
Challenges of Data Mining
References

Information Retrieval, Lecture II

What’s Information Retrieval
Information Retrieval and Business Intelligence
Data preparation - Parsing
  - Tokenisation
  - Stop words removal
  - Stemming
  - Entity detection
  - Part of speech
Data storage - Indexing
  - Index construction
Exploitation - Querying
  - Exploiting data repositories
Questions
Information Retrieval

What’s Information Retrieval

In the science of searching for information in heterogeneous data sources such as:

- Documents
- Images
- Audio
- Video
What I.R. systems do?

- Take documents in any format
  - Break into words
- Create an index
- Search it very quickly
- Update index when collection changes

Information Retrieval and B.I.

Business Intelligence transforms data into valuable information that can be accessed to help decision makers develop strategic, tactical and operational planning initiatives.

Business Intelligences deals with data integration from structured and unstructured data sources.
Enterprise Search

Enterprise search characteristics:
- Diversity of content sources and formats, and not necessarily HTTP based
- Secure access
- Combined structured and unstructured search
- Dedicated search (e.g. email search)
- Intranet search
- Ranking and categorisation problem
- Social forces behind the creation of Internet and Intranet content are quite different
- Deployment environments for these domains also differs

Enterprise Search comparison with Web search

Similarities
- Crawling
- Indexing and ranking
- User interface

Differences
- Diverse repositories
- Fewer documents
- Access control
- Diverse doc type
- Different user needs
- Less filtering
Data preparation and data source

Different document types required different access methods. PDF, MS Word, HTML, Open formats, email, etc.

Good Formatted file

name="Ron Whitney"
email="RFW@math.ams.org"
sent="Sat, 20 Apr 1996 17:44:20 -0400 (EDT)"
inreplyto="199604201303.6479@uvea.wolfram.com"

*I was unable to attend last week, but am available this Monday. If the group plans to meet, I’ll attend, but I have no specific items for the agenda now.

-Ron

Bad formatted file

name="*
email="mail137@163.com"
sent="Wed, 8 May 2002 14:39:51 -0400"

I was unable to attend last week, but am available this Monday. If the group plans to meet, I’ll attend, but I have no specific items for the agenda now.

-Ron

Parsing, tokenisation

Tokenisation is the process of splitting a stream of words into units or "tokens". Normally this process does not included the following symbols:

- period (.)
- comma (,)
- semicolon (;)
- quotation marks ("")
- colon (:)
- brackets [ ]
- braces {}
- parentheses ()
- mathematical operators + - / * = < >
- special characters | & ~
- the at sign @
- underscores and other rare characters

e.g.

The “brown” fox jumps, quickly over the lazy dog*.

The brown fox jumps quickly over the lazy dog
Parsing, stop words removal

**Stop Word** is the name given to a word that will be filtered and is not consider relevant by an information retrieval system. Some of the more frequently used stop words for English include: “a”, “of”, “the”, “I”, “it”, “you”, and “and”. These are generally regarded as functional words that do not carry meaning for the system.

**Stop List** is the list or set of stop words, there are as many stop lists as there are languages. I.e. if a system processes text that includes English, French, German and Spanish it also should be a stop list for each of these languages.

Parsing, Stemming

Stemming is the process to obtain a word’s root (also called normalisation) through eliminating suffixes.

- running → run
- ran
- laughing → laugh
- laughs
- laughed
- languished → languish
- seals
- connection → connect
- connections
- connective
- connected
- connecting

**Benefits of Stemming**
- Takes care of morphological variants
- Reduces index size

**Known Limitations**
- Impact on advanced syntax and exact match
- Accented characters are not supported
- Short words are not stemmed

Parsing, entity detection

The problem here is to find various structured data within unstructured documents, e.g.

- people’s names
- project’s names
- places
- amounts

Algorithms for entity detection normally are either rule or statistical based.

see: Special Interest Group on Natural Language Learning on the Association for Computational Linguistics (CoNLL). http://cnts.uia.ac.be/signll/conll.html

Parsing, part of speech

Part of speech is a categorisation process that take in consideration phrase function. Each part of speech explains not what the word is, but how the word is used. POS is very useful when dealing with Natural Language Processing in IR.

Parts of speech: the verb, the noun, the pronoun, the adjective, the adverb, the preposition, the conjunction, and the article.

e.g.

<table>
<thead>
<tr>
<th>can</th>
<th>I think I can do it.</th>
<th>verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>can</td>
<td>Don’t open that can of beans.</td>
<td>noun</td>
</tr>
<tr>
<td>only</td>
<td>This is my only pen.</td>
<td>adjective</td>
</tr>
<tr>
<td>only</td>
<td>He was only joking.</td>
<td>adverb</td>
</tr>
<tr>
<td>his</td>
<td>That book is his.</td>
<td>pronoun</td>
</tr>
<tr>
<td>his</td>
<td>That is his book.</td>
<td>adjective</td>
</tr>
<tr>
<td>English</td>
<td>Can you speak English?</td>
<td>noun</td>
</tr>
<tr>
<td>English</td>
<td>I am reading an English novel.</td>
<td>adjective</td>
</tr>
</tbody>
</table>
Data storage: indexing

After parsing

- Tokenisation
- Stop words removal
- Stemming
- Entity detection
- Part of speech

we can start the indexation process, which consist in storing the tokens in a DB, usually in a vector space fashion

Index design factors

- Merge factors
- Storage techniques
- Index size (compression)
- Lookup speed
- Maintenance
- Fault tolerance
- Scalability
Index construction, inverted index

T1: bab(y, ies, y’s)  D1: Infant & Toddler First Aid
T2: child(ren’s)     D2: Babies & Children’s Rooms (For Your Home)
T3: guide           D3: Child Safety at Home
T4: health          D4: Your Baby’s Health and Safety: From Infant to Toddler
T5: home            D5: Baby Proofing Basics
T6: infant          D6: Your Guide to Easy Rust Proofing
T7: safety          D7: Beanie Babies Collector’s Guide
T8: toddler

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

Vector Space Model (VSM)

\[ t_i^T = [x_{i,1}, \ldots, x_{i,n}] \quad d_j = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{m,j} \end{bmatrix} \quad t_i^T \rightarrow \begin{bmatrix} x_{1,1} & \ldots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \ldots & x_{m,n} \end{bmatrix} \]

Where:

\[ f_{i,j} : \text{number of times term } i \text{ in document } j \]

<table>
<thead>
<tr>
<th>Binary</th>
<th>( \chi(f_{ij}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logarithmic</td>
<td>( \log(1 + f_{ij}) )</td>
</tr>
<tr>
<td>Normal</td>
<td>( \frac{1}{\sqrt{\sum_{j} f_{ij}^2}} )</td>
</tr>
<tr>
<td>Inverse Document Frequency</td>
<td>( \log(n/\sum_{j} \chi(f_{ij})) )</td>
</tr>
<tr>
<td>etc.</td>
<td>.............</td>
</tr>
</tbody>
</table>
Similarity

Similarity help to identifying the closeness between different vectors or in our case documents (or between a query and documents).

Similarity is based in a metric that defines distance

Euclidean distances in the classical example

Vector Space geometrically explained (2 dimensions)

Dot product
\[ A \cdot B = x_1 x_2 + y_1 y_2 \]

Euclidean distance
\[ a, b = d_{ab} = ((x_1 - x_0)^2 + (y_1 - y_0)^2)^{1/2} = (x_{12} + y_{12})^{1/2} \]
Similarity, how far is A from B?

We normalise the dot product by the Euclidean distance. This ratio defines the cosine.

\[ \text{Dot product} \]

\[ \text{Distance} \]

Distance

Similarity between a query and a document

\[
\text{Sim}(Q, D_i) = \frac{\sum_{i,j} w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}}
\]

System exploitation, querying

**Types of Queries**

- **Boolean**: AND, OR, NOT
- **Natural Language Queries**: query is formulated as a question or a statement
- **Thesaurus Queries**: the user selects the term from a previous term-set provided by the IR system
- **Fuzzy Queries**: threshold of relevance is expanded to include additional documents
- **Term Searches**: based in a few words or phrases provided by the user
- **Probabilistic Queries**: IR systems based in a computed probability to retrieve documents
An IR system overview

Sorting the query’s result

To present a query’s result to the user the output should be sorted in a meaningful way.

For example we can use the cosine similarity measure to rank result.

Also we can use clustering!
Ranking

For ranking we can use different mathematical functions based in word frequencies to determine the importance of a query’s result.

Then we can sort this result in ascent order.

Clustering

Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that:

– Data points in one cluster are more similar to one another
– Data points in separate clusters are less similar to one another

Unlike the classification problem here we do not know the labels or data categories, for this reason this is also called unsupervised learning.
Evaluation of I.R. systems

Recall = \( \frac{\text{No. of relevant docs retrieved}}{\text{Total relevant in the collection}} \)

Precision = \( \frac{\text{No. of relevant docs retrieved}}{\text{Total retrieved in the collection}} \)

Demo 1: text classification

Here we have an ontology and we would like to assign or classify documents on a given category

http://thames.cs.rhul.ac.uk/wstalk
Demo 1: text classification

http://thames.cs.rhul.ac.uk/wstalk/prototype.html

Demo 2: email search

file:<keyword>
from: <name>
to: <email to>
cc: <email cc>
subject: <subject>
<keyword>
Search engines

Commercial

Google
Yahoo!
Ask
WebCrawler
Live Search
altavista

Open source

Lucene
Terrier
Xapian

and so on...

The web needs a lot of servers
Enterprise search

Not just docs. Multimedia I.R.

Content is not just text-based, there is more than text, i.e.

- Images
- Video
- Audio

But there again there isn’t a perfect technique... let’s see some examples
Text based image retrieval

e.g. Using information from the web site to annotate images
  - Neighbouring text
  - Same paragraph text
  - Title
  - Heading
  - etc.

Colour histogram based image retrieval

Take an image, convert to histogram and use to query image index
Colour based image retrieval

http://www.hermitagemuseum.org/

Features based image retrieval

http://amazon.ece.utexas.edu/~qasim/cires.htm
Questions

References


