Software Data Analytics

part 1: Introduction to
SVM in Information Retrieval

Mohammad Ghafari
Software Composition Group
Fall 2016
Agenda

- Problem definition;
- The notion of Information Retrieval;
- Document scoring;
- Similarity in the vector space model.
Structured data refers to information with a high degree of organization that is readily searchable. Think about your Mom’s cabinets in the kitchen.
Lack of structure in data makes it unstructured. The searching will be a time and energy-consuming task.
Information retrieval

It is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

See “Introduction to Information Retrieval” by Christopher D. Manning et al., 2008.

This is the conventional definition. 
NB: document could be a software, its bug report, developers discussion, and etc.
The bigger the circle is, the more a term exists in a document. Think about one of your developed applications that needs a database. How you would look for specific information? How do you show the results to users?
Boolean retrieval

• It works based on the absence or presence of a term.
• Queries are formulated by combining the terms and Boolean operators such as AND, OR, NOT.
• The system returns all documents that satisfy the query.
Two main issues in Boolean retrieval: [1] Either it returns too few or too many results. [2] No scoring function to rank the findings.

Though 75% of the terms in both queries are identical, the first query returns only one record (that is also irrelevant) and the second one yields 819 records.
We would rank documents based on their relevance score to a query.
Assume we have a one term query, then the more frequent that term is in a document, we give a higher score to that document. What are the issues?
Another problem would be exact word matching (ignoring synonyms).

### Issues with rough frequency ranking

- All terms have equal weight;
- The frequency in larger documents may be much more than small documents;
- N-times more frequent does not necessarily mean n-times more relevant;

Another problem would be exact word matching (ignoring synonyms).
Each document is a bag of words.

Jaccard coefficient

It represents the overlap between two sets, a document and a query.

\[ \text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \]
• Query: actionable software visualization

• Doc₁: towards actionable visualization in software development → 3/6

• Doc₂: actionable visualization → 2/3
  \[ A \cap B = \{ \text{actionable, visualization} \} \]
  \[ A \cup B = \{ \text{actionable, software, visualization} \} \]

The second document gets a higher score, while the first one looks more relevant. Thus, Jaccard coefficient biases towards shorter documents.
We need a better normalization of the length.

Issues

- Jaccard has a bias towards shorter documents.
- It does not take into account the term frequency.
Each document is a binary vector. It only shows the absence or presence of the terms.
Each document is a count vector. In raw term frequency, relevance increases proportionally with term frequency, however, e.g., 10 times more frequent does not actually mean 10 times more relevant.
The term frequency $t_{f,d}$ of term $t$ in document $d$ is the number of times that $t$ occurs in $d$. The base in the Log depends on how much we want to dampen the effect of frequency.

$$w_{t,d} = \begin{cases} 1 + \log t_{f,t,d} & \text{if } t_{f,t,d} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

<table>
<thead>
<tr>
<th></th>
<th>Mining</th>
<th>Test-case</th>
<th>Visualization</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VISSoft'16</td>
<td>1.3</td>
<td>0</td>
<td>2.56</td>
<td>2.14</td>
</tr>
<tr>
<td>JSEM'16</td>
<td>2.49</td>
<td>2.41</td>
<td>0</td>
<td>2.25</td>
</tr>
<tr>
<td>IWST'16</td>
<td>0</td>
<td>1.47</td>
<td>1</td>
<td>1.60</td>
</tr>
<tr>
<td>SANER'17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Score $= \sum_{t \in d \cap \mathcal{D}} (1 + \log t_{f,t,d})$
Frequency is certainly one indicator, but we also need a high weight for rare terms. If a term is repeated in many documents it might be noise and have not a discriminative power for ranking, but if for example Duqu (a collection of computer malware discovered on 1 September 2011) is a rare term in the documents, the one containing this term is very likely to be more relevant to the query.
Inverse document frequency

The document frequency of term $t$, $df_t$, is the number of documents that contain $t$. Accordingly, the informativeness of a term is:

$$\text{idf}_t = \log \frac{N}{df_t}$$

If a term occurs in all documents then it has no discriminative power.
We consider both the frequency and the informativeness of a term during ranking. In fact, the scoring increases as the frequency of a term increases as well when a term is more informative (the rarity of a term).

**TF-IDF scoring**

\[
\text{tfidf}(t, d, D) = tf(t, d) \times idf(t, D)
\]

\[
\text{Score}(q, d) = \sum_{t \in q \cap d} \text{tfidf}_t, d
\]

Therefore, if a term occurs in all documents, its idf will be zero, and that term won't affect the ranking.
We have a m-dimensional vector space; terms are the axes of the space and documents are m-dimensional vectors in this space. This matrix becomes the index to compute the relevant score.
Length indicates how frequent is a term in a document. For example, take a document $d$ and append it to itself. Call this new document $d'$. Semantically $d$ and $d'$ have the same content but the Euclidean distance between $d$ and $d'$ can be quite large. However, the angle between them is 0, corresponding to maximal similarity. Thus, it is the angle that captures the distribution of terms.
Cosine similarity (proximity measure)

The cosine similarity between two vectors is the cosine of the angle between them.

\[ \cos \theta = similarity (A, B) = \frac{\vec{A} \cdot \vec{B}}{\|A\| \|B\|} \]

As long as vectors point to the same direction, the angle between them will still be small.
I wish I could just google anything...

Googled

where the f**k are my glasses?

and it will be like, “under the bed, you dumbass!”

HAPPINESS!

HEY FOUND MY GLASSES!

hellosixthuh