UNIVERSIDAD POLITÉCNICA DE MADRID

Search Engines

Machine Learning Ranking

information retrieval, extraction and integration

AUTHORS:
Álvaro Arranz
Ignacio Martínez
Sofia Smolianinova

Master's Programme in ICT Innovation: Data Science
&
Master of Science in Health & Medical Data Analytics (EIT Health)

escuela técnica superior de ingeniería de informática

1 Introduction

Every day, people make millions of queries to the search engines and get some ranked results. The most important question is "Which is the page or the set of the pages the user wants to receive as a result for his query?".

The word "ranked" in the previous paragraph plays an important part in the search engine algorithms. There are typically thousands of pages that contain words from the user query, but the user is interested in a much smaller subset. The end-user doesn't want to scroll down to the page with number 100 to find a meaningful article or relevant site; he wants to see the most relevant results at the top of the list. And the key point for resolving this problem is "user feedback". If we knew the set of pages actually relevant to the user's query, we could use this as training data for optimizing (and even personalizing) the retrieval function. Unfortunately, users don't want to give explicit feedback for each query that they do. However, information about relevant links is already hidden in the log files of search engines.

This document will give the definition of clickthrough data, how it can be recorded, how it can be used to generate training examples and how a support vector machine (SVM) approach can be trained with clickthrough data.

2 Pairwise Algorithm

The algorithm that we have chosen is the one presented by Thorstem Joachins in the paper "Optimizing Search Engines using Clickthrough Data". It uses a support vector machine (SVM) approach that is trained with clickthrough data (query, ranking, clicks).

Clickthrough data can be easily obtained. The query and the ranking are obtained whenever the search engine shows its results to the user. The documents clicked are recorded using a proxy system that keeps a logfile associated to the queries unique IDs.

The chosen algorithm does not classify the documents as relevant or non-relevant, since a user never reads all the results presented in the ranking and only selects a few, usually in the highest positions. A binary approach would mean that the algorithm could achieve a high degree of accuracy by marking all documents as non-relevant.

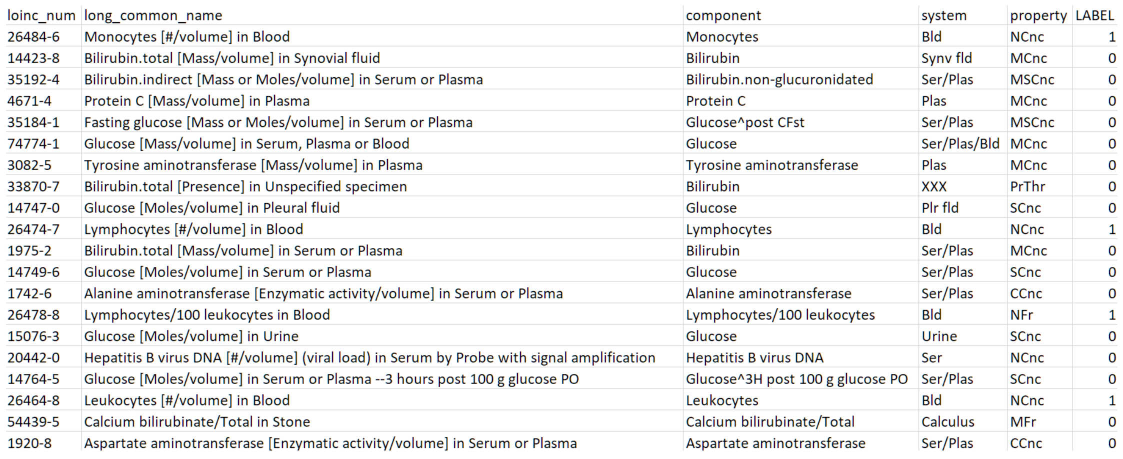
3 Training Dataset

The training data for the algorithm makes use of the information conveyed in click-through data. It is composed of triples of the following elements:

1. Query: The set of words the user looks for.
2. Ranking: The set of documents presented to the user as result of the query search.
3. Clicks: The set of documents that the user clicks on.

The training set constructed for this algorithm consists on three clickthrough triplets associated to the following queries: "glucose in blood", "bilirrobin in plasma", "white blood cells count". For each query, an unordered ranking with documents marked as clicked (1) and unclicked (0) is obtained. Figure 1 shows an example.

The clicked and non-clicked marks act as labels for the algorithm to order the results. Clickthrough data is noisy, as users not always click on the best results and often click in several documents for the same queries. However, the clicks contain some information, since they are not made at random but using the user's judgement.



loinc_num	long_common_name	component	system	property	LABEL
26484-6	Monocytes [#/volume] in Blood	Monocytes	Bld	NCnc	1
14423-8	Bilirubin.total [Mass/volume] in Synovial fluid	Bilirubin	Synv fld	MCnc	0
35192-4	Bilirubin.indirect [Mass or Moles/volume] in Serum or Plasma	Bilirubin.non-glucuronidated	Ser/Plas	MSCnc	0
4671-4	Protein C [Mass/volume] in Plasma	Protein C	Plas	MCnc	0
35184-1	Fasting glucose [Mass or Moles/volume] in Serum or Plasma	Glucose^post CFst	Ser/Plas	MSCnc	0
74774-1	Glucose [Mass/volume] in Serum, Plasma or Blood	Glucose	Ser/Plas/Bld	MCnc	0
3082-5	Tyrosine aminotransferase [Mass/volume] in Plasma	Tyrosine aminotransferase	Plas	MCnc	0
33870-7	Bilirubin.total [Presence] in Unspecified specimen	Bilirubin	XXX	PrThr	0
14747-0	Glucose [Moles/volume] in Pleural fluid	Glucose	Plr fld	SCnc	0
26474-7	Lymphocytes [#/volume] in Blood	Lymphocytes	Bld	NCnc	1
1975-2	Bilirubin.total [Mass/volume] in Serum or Plasma	Bilirubin	Ser/Plas	MCnc	0
14749-6	Glucose [Moles/volume] in Serum or Plasma	Glucose	Ser/Plas	SCnc	0
1742-6	Alanine aminotransferase [Enzymatic activity/volume] in Serum or Plasma	Alanine aminotransferase	Ser/Plas	CCnc	0
26478-8	Lymphocytes/100 leukocytes in Blood	Lymphocytes/100 leukocytes	Bld	NFr	1
15076-3	Glucose [Moles/volume] in Urine	Glucose	Urine	SCnc	0
20442-0	Hepatitis B virus DNA [#/volume] (viral load) in Serum by Probe with signal amplification	Hepatitis B virus DNA	Ser	NCnc	0
14764-5	Glucose [Moles/volume] in Serum or Plasma --3 hours post 100 g glucose PO	Glucose^3H post 100 g glucose PO	Ser/Plas	SCnc	0
26464-8	Leukocytes [#/volume] in Blood	Leukocytes	Bld	NCnc	1
54439-5	Calcium bilirubinate/Total in Stone	Calcium bilirubinate/Total	Calculus	MFr	0
1920-8	Aspartate aminotransferase [Enzymatic activity/volume] in Serum or Plasma	Aspartate aminotransferase	Ser/Plas	CCnc	0

Figure 1: Clickthrough triplet for the query "glucose in blood". The LABEL column represents the relevance of the document (0: ignored, 1: clicked).

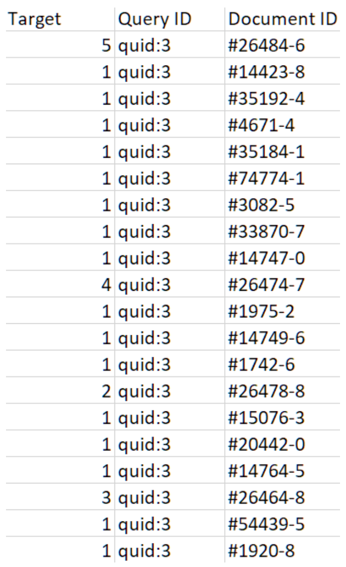
The dataset must be adapted to the adequate format before being fed to the algorithm. The correct format is composed of a file with one line per document, per query. Each line is composed of several values:

1. Target: The desired order in the ranking. The higher its value the higher the document most be placed in the ranking.
2. Query ID: Query's unique identifier. Each query has its own target values.

3. Feature:Value pairs: Pairs of numbers representing the features and their associated values. Features are integers and must be ordered. Values are floats.

4. Additional information: Preceded by ””, it is saved and used to add some information that could be useful for the user.

Figure 2 shows an example of this format. The main difference is that the information contained by the label is replaced by the target value. This way, no document is binary marked as relevant or non-relevant, but a preferred order is proposed.



Target	Query ID	Document ID
5	quid:3	#26484-6
1	quid:3	#14423-8
1	quid:3	#35192-4
1	quid:3	#4671-4
1	quid:3	#35184-1
1	quid:3	#74774-1
1	quid:3	#3082-5
1	quid:3	#33870-7
1	quid:3	#14747-0
4	quid:3	#26474-7
1	quid:3	#1975-2
1	quid:3	#14749-6
1	quid:3	#1742-6
2	quid:3	#26478-8
1	quid:3	#15076-3
1	quid:3	#20442-0
1	quid:3	#14764-5
3	quid:3	#26464-8
1	quid:3	#54439-5
1	quid:3	#1920-8

Figure 2: Expected format for the SVM algorithm. Note that no feature:value pairs have been included.

4 Working with the model

4.1 Training

The process of training the model is simple. First, model receives as input the training data in the appropiate format. Second, the model solves and optimization problem (see Figure 3). This way, we obtain a ranking function that has a low number of discordant pairs with respect to the target ranking.

OPTIMIZATION PROBLEM 2. (RANKING SVM (PARTIAL))

$$\textit{minimize:} \quad V(\vec{w}, \vec{\xi}) = \frac{1}{2}\, \vec{w} \cdot \vec{w} + C \sum \xi_{i,j,k} \tag{21}$$

subject to:

$$\forall (d_i, d_j) \in r_1' : \vec{w}\Phi(q_1, d_i) > \vec{w}\Phi(q_1, d_j) + 1 - \xi_{i,j,1}$$

$$\ldots \tag{22}$$

$$\forall (d_i, d_j) \in r_n' : \vec{w}\Phi(q_n, d_i) > \vec{w}\Phi(q_n, d_j) + 1 - \xi_{i,j,n}$$

$$\forall i \forall j \forall k : \xi_{i,j,k} \geq 0 \tag{23}$$

Figure 3: Optimization problem to be solved by the model. Source: "Optimizing Search Engines using Clickthrough Data", by Thorstem Joachins.

4.2 Ranking

The ranking function obtained in the training process is applied to the queries. For each query the documents with different target values should be ranked from highest to lowest target value. On the other hand,those documents with equal target values should be ordered according to their features.

There are different possible features to be used:

1. Rank in other search engines:ranking 1st in X search engine, ranked in the top 50 in X search engine, rank in the top 10 in X, Y and Z search engines, etc.
2. Query/Content match: cosine between the url words and the query, if the query includes the name of the document's domain, etc.
3. Popularity attributes: url length, country code of the url, etc.