An Analysis of Factors Used in Search Engine Ranking

Albert Bifet [1] Carlos Castillo [2] Paul-Alexandru Chirita [3]
Ingmar Weber [4]

[1]Technical University of Catalonia [2]University of Chile

[3]L3S Research Center [4]Max-Planck-Institute for Computer Science

Adversarial Information Retrieval Workshop AIRWeb
14th IWWW Conference Tokyo

Outline

1. Motivation
   - Introduction
2. Retrieval Information
   - Introduction
   - Features
   - Estimating a Ranking Function
3. Experimental Results
   - Architecture of the system
   - Main Results

Outline

1. Motivation
   - Introduction
2. Retrieval Information
   - Introduction
   - Features
   - Estimating a Ranking Function
3. Experimental Results
   - Architecture of the system
   - Main Results

Search Engine Ranking.

> *Un buscador es alguien que busca,*
> *no necesariamente alguien que encuentra.*
> **JORGE BUCAY**

Why it's Search Engine Ranking so important?

- 88% of the time we use it, given a new task
- E-commerce depends on a high ranking

How we can get a good ranking?

- Ranking depends on certain features
- Don't forget : CONTENT

An Analysis of Factors Used in Search Engine Ranking

Influence of different page features on the ranking of search engine results:

- We use Google
- Binary classification problem
- Linear and non-linear methods
- Training set and a test set

Outline

1. Motivation
   - Introduction
2. Retrieval Information
   - Introduction
   - Features
   - Estimating a Ranking Function
3. Experimental Results
   - Architecture of the system
   - Main Results

IR Model

Definition

An information retrieval model is a quadruple $[\mathbf{D}, \mathbf{Q}, \mathcal{F}, R(q_i, d_j)]$ where

1. $\mathbf{D}$ is a set composed of logical representations for the documents in the collection
2. $\mathbf{Q}$ is a set composed of logical representations for the user information needs (Queries)
3. $\mathcal{F}$ is a framework for modeling documents representations, queries, and their relationships
4. $R(q_i, d_j)$ is a ranking function wich associates a real number with a query $q_i \in \mathbf{Q}$ and a document representation $d_j \in \mathbf{D}$

IR Model

Definition

Let t be the number of index terms in the system and k_i be a generic index term.

1. $K = \{k_1, \ldots, k_t\}$ is the set of all index terms
2. A weight $w_{i,j}$ is associated with each k_i of a document d_j
3. With the document d_j is associated an index term vector $\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$

Example

1. Document 1 =" My tailor is rich", Document 2 ="Your chair is red"
2. $K = \{$"tailor","chair","rich","red" $\}$
3. $\vec{d_1} = (1, 0, 1, 0), \vec{d_2} = (0, 1, 0, 1)$

Vector Model

Definition

For the vector model, the weight $w_{i,j}$ associated with a pair (k_i, d_j) is positive and non-binary. The index terms in the query are also weighted

1. Let $w_{i,q}$ be the weight associated with the pair $[k_i, q]$
2. We define the query vector $\vec{q}$ as $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$
3. With the document d_j is associated an index term vector $\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$
4. $$Sim(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{\left|\vec{d_j}\right| \left|\vec{q}\right|} = \frac{\sum_i w_{i,j} \times w_{i,q}}{\sqrt{\sum_i w_{i,j}^2}\sqrt{\sum_i w_{i,q}^2}}$$

Vector Model

Example

1. Document 1 ="My tailor is rich"
2. Document 2 ="Your chair is red"
3. $K = \{$ "tailor", "chair", "rich" $\}$
4. Query= "rich"

Example

1. $\vec{q} = (0, 0, 1)$
2. $\vec{d_1} = (1, 0, 2), \vec{d_2} = (0, 1, 0)$
3. $Sim(d_1, q) = \frac{\vec{d_1} \cdot \vec{q}}{|\vec{d_1}||\vec{q}|} = \frac{2}{\sqrt{5}}$, $Sim(d_2, q) = 0$

Vector Model

Definition

1. Let N be the total number of documents in the system
2. Let n_i be the number of documents in wich k_i appears
3. The best known term-weighting schemes use weights wich are given by
$$w_{i,j} = f_{i,j} \times idf_i$$
4. The inverse document frequency is
$$idf_i = \log \frac{N}{n_i}$$

PageRank

The Anatomy of a Search Engine **Sergey Brin and Lawrence Page**

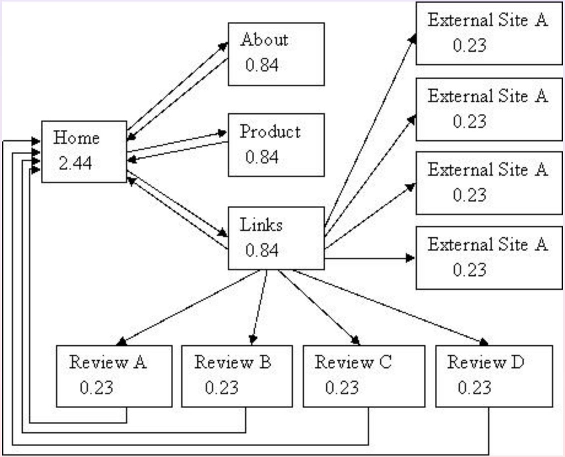
Definition

We assume page A has pages $T_1 ... T_n$ which point to it. d is a damping factor between 0 and 1. Also $C(A)$ is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

$$PR(A) = (1 - d) + d(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)})$$

Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.

PageRank


About
0.84
Product
0.84
Home
2.44
Links
0.84
External Site A
0.23
External Site A
0.23
External Site A
0.23
External Site A
0.23
Review A
0.23
Review B
0.23
Review C
0.23
Review D
0.23

Outline

1. Motivation
   - Introduction
2. Retrieval Information
   - Introduction
   - Features
   - Estimating a Ranking Function
3. Experimental Results
   - Architecture of the system
   - Main Results

Content Features

- Content features, query independent:
  - `FNEN` Fraction of terms in the documents which can not be found in an English dictionary
  - `NBOD` Number of bytes of the original document
  - `RFFT` Relative frequency of the more frequent term
  - `ATLE` Average term length
- Content features, query dependent:
  - `SIMT` Similarity of the term to the document.
  - `AMQT` Average matches of the query terms
  - `FATT` Anchor text term frequency

Formatting, Link and Metadata Features

- Formatting features, query-dependent.
  - `TMKY` Term in the meta keywords or description (N)
- Link features, query-independent:
  - `ILNK` Number of pages linking to a page, in-degree approximated using Google API `link:` queries
  - `PRNK` PageRank of the page, or the approximation of the PageRank in a 0-10 scale obtained from Google's toolbar.
- Metadata features:
  - `TURL` Term is in the page's URL or not.
  - `TDIR` Term is listed in a web directory or not.

Outline

1. Motivation
   - Introduction
2. Retrieval Information
   - Introduction
   - Features
   - Estimating a Ranking Function
3. Experimental Results
   - Architecture of the system
   - Main Results

Data Set

- **Arts** : Albertinelli, Bacchiacca, Botticelli
- **States** : Arizona, Arkansas, Connecticut
- **Spam** : buy cds, buy dvds, cheap software
- **Multiple** : anova bootstrap feature missing principal squared, analysis frequent likelihood misclassification pruning statistical,

- **Training Set** (7 queries): learn a linear scoring function or a decision tree.
- **Validation Set** (2 queries): used for feature selection and pruning the decision tree.
- **Test Set** (3 queries): To estimate the generalization error of our ranking function,

Binary classification problem

- Let q be a query, and $\mathbf{u}, \mathbf{v}$ be feature vector of pages.
- Let $\mathbf{u} <_q \mathbf{v}$ represent the ordering returned by the ranking function of a search engine for the given query.
- We want to find f such that $f(\mathbf{u}) < f(\mathbf{v})$ whenever $\mathbf{u} < \mathbf{v}$. If we assume that f is linear, then there exists a vector $\mathbf{w}$ such that $f(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u}$. Then

$$\begin{aligned} & f(\mathbf{u}) < f(\mathbf{v}) \\ \Leftrightarrow \quad & \mathbf{w} \cdot \mathbf{u} < \mathbf{w} \cdot \mathbf{v} \\ \Leftrightarrow \quad & \mathbf{w} \cdot (\mathbf{v} - \mathbf{u}) > 0 \, . \end{aligned}$$

Logistic Regression and SVM

Logistic regression models the posterior probabilities of the classes. In the case of only two classes “–” and “+” the model has the form

$$\log \frac{P(\text{class = “+”}|X = x)}{P(\text{class = “–”}|X = x)} = \beta_0 + \mathbf{w} \cdot \mathbf{x} \tag{1}$$

- Support Vector Machines typically use linear decision boundaries in a *transformed* feature space.
- In this higher dimension space data becomes separable by hyperplanes

Binary classification trees

- f might not be linear
- Search engine might use several layers of índices
- A classification tree is built through binary recursive partitioning.
- Pruning the tree leads to a better performance on general data

Outline

1. Motivation
   - Introduction
2. Retrieval Information
   - Introduction
   - Features
   - Estimating a Ranking Function
3. Experimental Results
   - Architecture of the system
   - Main Results

Architecture.

- **Downloader**: software that executes a query and downloads the returned pages using the data set queries
- **Feature Extractor**: software that computes the features of the pages downloaded.
- **Analyzer**: software that analyzes the features of the returned pages and estimates a function

Outline

1. Motivation
   - Introduction
2. Retrieval Information
   - Introduction
   - Features
   - Estimating a Ranking Function
3. Experimental Results
   - Architecture of the system
   - Main Results

Precision values obtained using only individual features.

Feature	Arts	States	Spam	Multiple
NBOD	(-)53.8%	**54.4%**	51.1%	50.9%
FNEN	**(-)59.4%**	53.5%	51.6%	**(-)59.8%**
RFFT	52.1%	(-)54.3%	50.0%	53.9%
ATLE	(-)54.2%	54.0%	50.0%	54.4%
FATT	56.2%	50.3%	53.0%	51.8%
AMQT	55.4%	50.5%	52.1%	56.9%
SIMT (N)	56.5%	(-)52.0%	52.7%	**59.0%**
SIMT	55.4%	(-)50.9%	52.6%	**69.7%**
TMKY (N)	51.4%	51.4%	54.5%	50.0%
TMKY	53.0%	51.4%	**55.1%**	50.0%
ILNK	**58.0%**	**66.3%**	**57.0%**	53.9%
PRNK	**58.7%**	**60.6%**	**55.0%**	57.2%
TDIR	52.3%	53.5%	54.3%	51.9%

Best precision achieved on all.

Table: Best precision achieved on all, “shifted” and “top” pairs. We include the performance on the test data as well as on the whole data set, including training, validation and test sets.

	% all		% “shifted”		% “top”		
	pairs correct		pairs correct		pairs correct		
Dataset	Test	All	Test	All	Test	All	Best model
Arts	63.7%	61.8%	69.1%	66.4%	47.6%	48.0%	Log. regr., strongest 3 features
States	64.6%	66.3%	73.2%	73.8%	97.6%	98.5%	Class. tree, only ILINK feature
Spam	62.5%	59.5%	70.5%	62.1%	98.2%	74.8%	Log. regr., strongest 10 features
Multiple	67.5%	70.9%	78.1%	81.3%	81.0%	87.0%	Log. reg., strongest 3 features

Relevant Features hidden

- The query logs, which Google obtains through its toolbar.
- The age of the incoming links and other information related to web link dynamics.
- The rate of change at which a website changes, obtained by repeated web crawls.
- The “true” number of ingoing links, as Google’s `link:www.abc.com` only gives a lower bound.
- The “true” PageRank used by Google, as the one displayed in its toolbar is only an approximation, and furthermore, seems to be too strongly correlated to the number of in-links .

Summary

- Influence of different page features on the ranking of search engine results:
  - We use Google
  - Binary classification problem
  - Training set and a test set
- Ranking only according to the strongest feature for a category gives is able to predict the order in which any pair of pages will appear in the results with a precision of between 57% and 70% .