MASTER'S PROJECT IN INFORMATION RETRIEVAL

2014

By Vinothini Ramachandran

Department of Computer Science

and

Information Technology

Montclair State University

***Advisor*: Dr. H.M. Hubey**

Advising Committee

Dr. Jing Peng

Dr. Dajin Wang

Table of Contents

Project Report

I. Introduction

Information retrieval (IR) is the process of getting information resources related to a query. Information retrieval searches up on various algorithm to optimize various parameter.

These methods can be used to access the library books, journals and other materials. Web search engines are the most visible IR (Information Retrieval) applications.

This method is used when a user needs some information from the Internet, he/ she will type the queries on any web search engines to get the needed information. Then the Information retrieval method will display not only the relevant information, it will also display information related for the user typed queries. It will display all the objects matched for the query. The queries can be typed in any format.

The documents are represented in the meta data instead of storing directly in the information retrieval. An object can be of anything that can be used in database as information.

In IR, the information for the query is shown using ranking system. The Matching data for the query from the database that is selected will be give a numeric score. Based on the score, the data will be retrieved and shown to the user.

II. Model types

The documents are transformed into a particular representation for the effective use of information system. Each retrieval process include a specific model for its document representation purposes. The models are generally categorized into two types,

Mathematical basis

Properties of the model

a) Mathematical basis

a. 1) Set-theoretic models. Set-theoretic models required documents as sets of words or phrases. It is similar to the set-theoretic operations. Common models are:

- Standard Boolean model
- Extended Boolean model
- Fuzzy retrieval

a.2) Algebraic model. In this model vectors, matrices and tuples are used as documents and queries. This method similar to the representation of scalar value.

Examples are given below,

- Vector space model
    - Generalized vector space model
    - Topic-based Vector Space Model
    - Extended Boolean model
    - Latent semantic indexing or latent semantic analysis

a.3) Probabilistic models. Similarities are computed as probabilities that a document is relevant for a given query.

- Binary Independence Model
    - Uncertain inference
    - Language models
    - Divergence-from-randomness model
    - Latent Dirichlet allocation

a.4) Feature-based retrieval models. This model analyze the best feature from the document and gives the ranking. Feature functions are arbitrary functions of document and query, and as such can easily incorporate almost any other retrieval model as just a yet another feature.

b) Properties of the model

Properties of model is classified into following types,

Models without term inter dependencies treat different terms/words as independent. It uses the vector space model for the vectors and probabilistic model for variables.

Models with immanent term inter dependencies allow a representation of inter dependencies between terms. However the degree of the interdependency between two terms is defined by the model itself. It is usually directly or indirectly derived from the co-occurrence of those terms in the whole set of documents.

Models with transcendent term inter dependencies allow a representation of inter dependencies between terms, but they do not allege how the interdependency between two terms is defined. They relay an external source for the degree of interdependency between two terms.

III. Vector Space Model

The vector space model procedure can be divided in to three stages. (a) First stage is the document indexing where content bearing terms are extracted from the document text. (b) Second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. (c) Last stage ranks the document with respect to the query according to a similarity measure.

a) Document Indexing

It is obvious that many of the words in a document do not describe the content, words like *the, is*. By using automatic document indexing those non significant words are removed from the document vector, so the document will only be represented by content bearing words. This indexing can be based on term frequency, where terms that have both high and low frequency within a document are considered to be function words. In practice, term frequency has been difficult to implement in automatic indexing. Instead the use of a stop list which holds common words to remove high frequency words , which makes the indexing method language dependent. In general, 40-50% of the total number of words in a document are removed with the help of a stop list.

Non linguistic methods for indexing have also been implemented. Probabilistic indexing is based on the assumption that there is some statistical difference in the distribution of content bearing words, and function words. Probabilistic indexing ranks the terms in the collection of the term frequency in the whole collection. The function words are modeled by a Poisson distribution over all documents, as content bearing terms cannot be modeled. The use of Poisson model has been expand to Bernoulli model. Recently, an automatic indexing method which uses serial clustering of words in text has been introduced. The value of such clustering is an indicator if the word is content bearing.

b) Term Weighting

Term weighting has been explained by controlling the exhaustivity and specificity of the search, where the exhaustivity is related to recall and specificity to precision. The

term weighting for the vector space model has entirely been based on single term statistics. There are three main factors term weighting: term frequency factor, collection frequency factor and length normalization factor. These three factor are multiplied together to make the resulting term weight.

A common weighting scheme for terms within a document is to use the frequency of occurrence. The term frequency is somewhat content descriptive for the documents and is generally used as the basis of a weighted document vector. It is also possible to use binary document vector, but the results have not been as good compared to term frequency when using the vector space model.

There are used various weighting schemes to discriminate one document from the other. In general this factor is called collection frequency document. Experimentally it has been shown that these document discrimination factors lead to a more effective retrieval, i.e., an improvement in precision and recall.

The third possible weighting factor is a document length normalization factor. Long documents have usually a much larger term set than short documents, which makes long documents more likely to be retrieved than short documents.

Different weight schemes have been investigated and the best results, to recall and precision, are obtained by using term frequency with inverse document frequency and length normalization.

c) Similarity Coefficients

The similarity in vector space models is determined by using associative coefficients based on the inner product of the document vector and query vector, where word overlap indicates similarity. The inner product is usually normalized. The most popular similarity measure is the cosine coefficient, which measures the angle between the a document vector and the query vector. Other measures are e.g., Jaccard and Dice coefficients. See below in section on Distance. See also the section on Dissimilarity Metrics.

IV. Distance

In everyday usage, distance may refer to a physical length, or an estimation based on other criteria. In mathematics, a distance function or metric is a generalization of the concept of physical distance. A metric is a function that behaves according to a specific set of rules, and is a concrete way of describing what it means for elements of

some space to be "close to" or "far away from" each other. In most cases, "distance from A to B" is interchangeable with "distance between B and A".

E.1) $d(x,z) \leq d(x,y) + d(y,z)$

a) Euclidean Distance

This metric represents the straight-line distance between observations in variable space, and is the most commonly used metric in many disciplines.

E.2) $\sum_{i=1}^{n} \left| x_i - y_i \right|$

b) Squared Euclidean Distance

This metric is simply the Euclidean Distance squared, and will give you the same results in terms of boundary delineation as the Euclidean Distance. We include this metric because if you have very large data sets, the processing time can be lower if the program does not have to calculate the square root for Euclidean Distance.

c) Geometry

In analytic geometry, the distance between two points of the xy-plane can be found using the distance formula. The distance between ($x1$, $y1$) and ($x2$, $y2$) is given by:

E.3) $d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

If given points ($x1$, $y1$, $z1$) and ($x2$, $y2$, $z2$) in three-space, the distance between them is

E.4) $d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2 - z_1)^2}$

V. Dissimilarity Metric

Dissimilarity metrics is defined as how close two sets of observations are in *conceptual space i*nstead of physical space but the dissimilarity metric has to obey the Distance Triangular Inequality in order to qualify as a “metric” The variables for each location being plotted in a many-dimensional space, and then imagine estimating

"distances" between these points. Both Euclidean distance and Manhattan distance can be used as metrics of dissimilarity as well as proximity, as can many other metrics. Dissimilarity metrics are closely related to similarity metrics; the range of values for both is often between 0 and 1. In many cases, you can convert between a measure of similarity and one of dissimilarity by subtracting the first metric from 1 to get the other

E.5) **S(x,y) = 1- D(x,y)**

VI. Information Refinement

a) Precision

Precision is otherwise called as positive predictive value. It is a basic measures used in evaluating search strategies. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage.

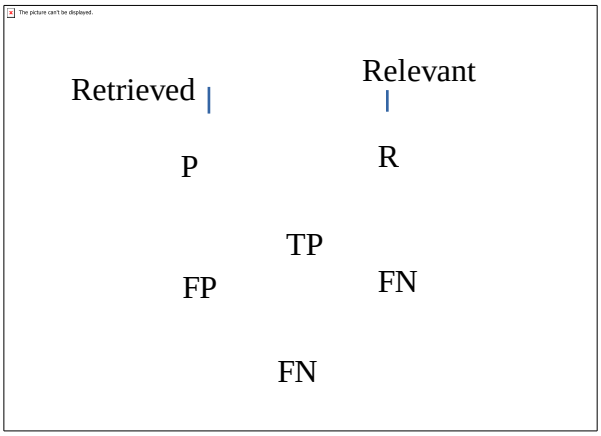


Figure 1: Precision and Recall

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})}$$

$$= P(\text{relevant}/\text{retrieved})$$

$$= \frac{TP}{TP + FP}$$

$$= \frac{P \cap R}{P}$$

b) Recall

Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

Recall= #(relevant items retrieved)
#(relevant items)

=P(retrieved/relevant)
= TP
TP + FN
= P∩R
R

Type I and Type II Errors in Prediction: Confusion Matrix

	Actual Positive	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

True Positive Rate = TP/ TP + FN
False Positive Rate = FP/ FP + TN

VII. Specific Models

a) TF- IDF

TF–IDF, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

One of the functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

term frequency tf(*t*,*d*), the simplest choice is to use the *raw frequency* of a term in a document, i.e. the number of times that term *t* occurs in document *d* . If we denote the raw frequency of *t* by f(*t*,*d*), then the simple tf scheme is tf(*t*,*d*) = f(*t*,*d*).

E.6) $tf(t, d) = 0.5 + 0.5 * f(t,d) / \max \{f(w,d) : w \epsilon d\}$

The **inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically scaled fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

E.7) $idf(t, D) = \log N / |\{d \epsilon D: t \varepsilon d\}|$

Example:

Consider a document containing 100 words wherein the word *cat* appears 3 times. The term frequency (i.e., tf) for *cat* is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word *cat* appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.

b) Tf-idf weighting

We now combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document. The *tf-idf* weighting scheme assigns to term 't' a weight in document 'd' given by,

E.8) $tf\text{-}idf_{t,d} = tf_{t,d} * idf_t.$

VIII. Simulating Documents Approximating the Zipf Density

Rejection sampling is a well-known method to generate random samples from arbitrary target probability distributions. It demands the design of a suitable proposal probability density function (pdf) from which candidate samples can be drawn. These samples are either accepted or rejected depending on a test involving the ratio of the target and proposal densities. The rejection sampling method is an efficient algorithm to sample from a log-concave target density, that attains high acceptance rates by improving the proposal density whenever a sample is rejected. In this project the rejection sampling procedure that can be applied with a broad class of target probability distributions, possibly non-log concave and exhibiting multiple modes. The proposed technique yields a sequence of proposal densities that converge toward the target pdf, thus achieving very high acceptance rates.

Introduction

Generating random numbers deals with computational task. It has many challenges. A good way to generate random numbers in computational statistics involves analyzing various distributions using computational methods. As a result, the probability distribution of each possible number appears to be uniform or pseudo-random.

Generating random number is based upon Rejection Sampling. The main idea is when you generate a number in the certain range, output that number immediately. If the number is out of the certain range, reject it and re-sample again. As each number in the desired range has the same probability of being chosen, a uniform distribution is produced.

Suppose if we want to sample from a target distribution $f(x)$ that is difficult or impossible to sample from directly. Suppose also that we have a proposal distribution $g(x)$ from which we have a reasonable method of sampling (e.g. the uniform distribution).

Step 1: Generate X,Y, on uniform density [0,1].

Step 2: Reject if Y> f(x)

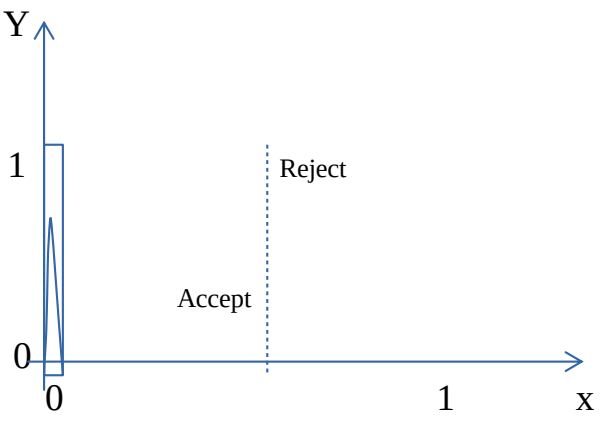


Figure 2: Rejection Sampling

The concept of rejection sampling is, imagine the large rectangular board, in that darts are throwing on the board to graphing the density function of a random variable. Assume that the darts are uniformly distributed around the board. Now remove or reject the dart which are throwing outside of the boundary. The remaining darts will be distributed uniformly within the area under the boundary, and the x-positions of these darts will be distributed according to the random variable's density. This is because there is the most

room for the darts to land where the curve is highest and thus the probability density is greatest.

The general form of rejection sampling assumes that the board is not necessarily rectangular but it can be of any shape according to the distribution method that we know how to sample from and which is at least as high at every point as the distribution we want to sample from, so that the former completely encloses the latter.

The fundamental figure of merit of a rejection sampler is the mean acceptance rate, which means the expected number of accepted samples over the total number of proposed candidates.

Zipf's Distribution/Density

Zipf's distribution is also referred to as Zeta distribution. It is a discrete probability distribution. The probability of occurrence of words or other items starts high and tapers off. Thus, a few occur very often while many others occur rarely. The general definition is, if $P_n \sim 1/n^a$, where P_n is the frequency of occurrence of the n^{th} ranked item and a is close to 1.

Zipf's law is most easily observed by plotting the data on a log-log graph, with the axes being log and log. For example, the word "the" would appear at $x = \log(1), y = \log(69971)$. The data conform to Zipf's law to the extent that the plot is linear.

- *N* be the number of elements;
- *k* be their rank;
- *s* be the value of the exponent characterizing the distribution.
-

Zipf's law then predicts that out of a population of *N* elements, the frequency of elements of rank *k*, $f(k;s,N)$, is:

E. 9) $$f(K; s, N) = 1/ k^s / \left(\sum_{n=1}^{N} (1/ n^s) \right)$$

Zipf's law holds if the number of occurrences of each element are independent and identically distributed random variables with power law distribution.

E.10) $\mathbf{p\ (f) = \alpha\ f^{-1-1/s}}$

In the example of the frequency of words in the English language, *N* is the number of words in the English language and, if we use the classic version of Zipf's law, the exponent *s* is 1. *f*(*k*; *s*,*N*) will then be the fraction of the time the *k*th most common word occurs.

The derivation of Zipf Distribution can be found in Smith .

Reginald Smith Density

The distinct words are calculated using files from WinEdt iSpell spell check dictionaries for a given language. Spell check dictionaries are not comprehensive by design since making them too large will include many rare or archaic words that should not be passed as correct in most writing. However, they give a good sample of commonly used terms in a language and the population of distinct words found in most texts excluding some proper nouns. The exception is the extinct language of Meroitic which used a corpus adopted from a previous paper .

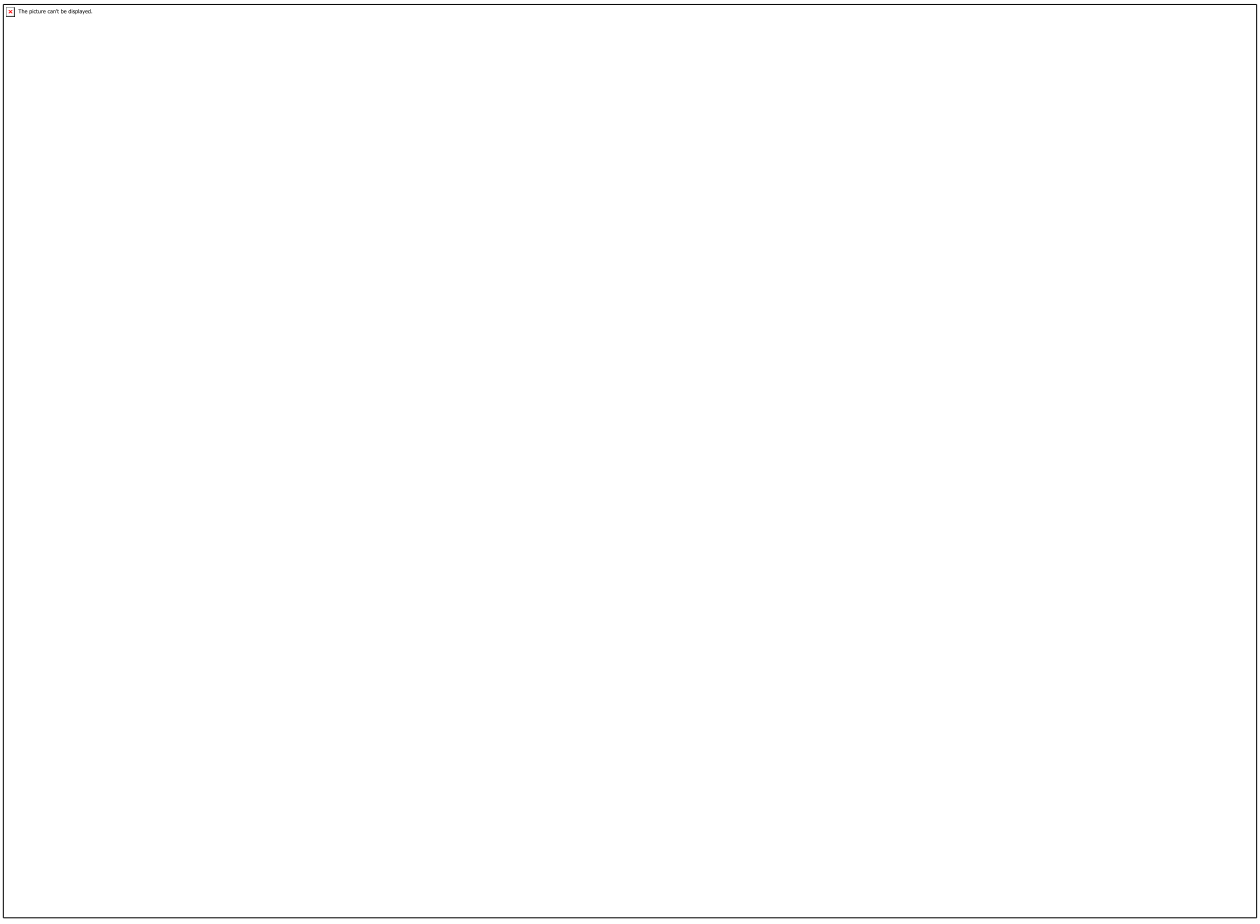
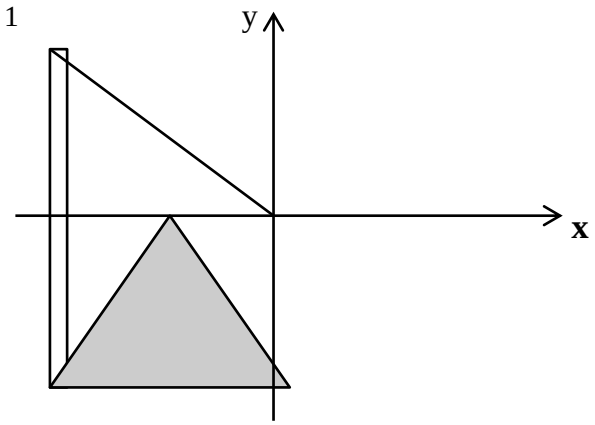


Figure 3: http://arxiv.org/ftp/arxiv/papers/1207/1207.2334.pdf

IX. Simple Simulation

The easiest approximation to the Zipf Distribution is probably the Triangular Distribution (density) shown below. However it is not asymmetric, hence we must improve it.



0

0 **1**

Figure 4: Simulation

x=1-y
2x=1
x= ½
Reject if (x>1/2)&& (y>1-x)
Reject if (x<1/2)&& (y>x)

X. Improved Triangular Density

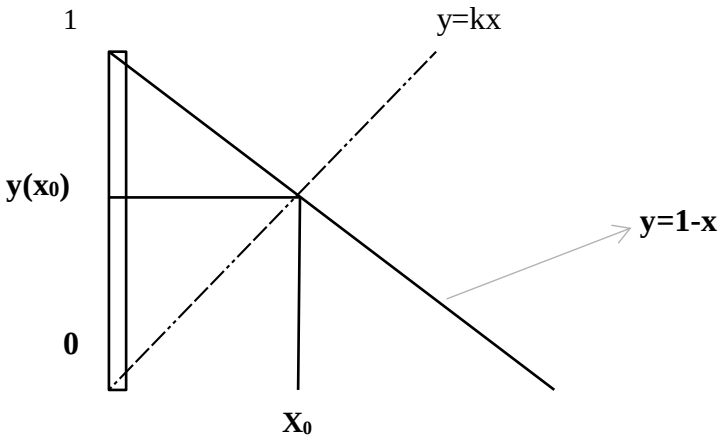


Figure 5: Improved Triangular Density

$kx=1-x$

$kx+x=1$

$x(1+k)=1$

$x_0=x=1/(1+k)$

if(x>1/(1+k)) && (y<1-x) ||
(x<1/(1+k)) && (y<x)

XI. Simulation

Program

Histogram.jar

XII. Output

XIII. References

http://cogsys.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html

http://www.biomedware.com/files/documentation/boundaryseer/Preparing_data/Choosing_a_dissimilarity_metric.htm

http://www.biomedware.com/files/documentation/boundaryseer/Preparing_data/About_dissimilarity_metrics.htm

http://machinelearning.wustl.edu/mlpapers/paper_files/icml2006_DavisG06.pdf

http://machinelearning.wustl.edu/mlpapers/paper_files/icml2006_DavisG06.pdf

http://en.wikipedia.org/wiki/Distance

http://arxiv.org/ftp/arxiv/papers/1207/1207.2334.pdf

http://en.wikipedia.org/wiki/Tf%E2%80%93idf

http://www-nlp.stanford.edu/IR-book/