Document Routing by Discriminant Projection : TREC-4

Kok F. Lai [1] Vincent A.S. Lee Jeremy P. Chew [2]

Information Technology Institute
11 Science Park Road, Singapore 0511.

Abstract

We present document routing as a standard problem in discriminant analysis. The standard solution involves the inversion of a large matrix whose dimension is the number of indexed terms. Typically, the solution does not exist because the number of training documents are much smaller compared to the number of terms. We show that one can project this raw document space into a lower dimensional space where solution is possible. Our projection algorithm exploits the characterisitics of the empty space, using only the training documents for efficient coding of the relevance information. Its complexity is linear with respect to the number of terms, and second order with respect to the number of training documents. We can therefore fully exploit the power of discriminant analysis without imposing severe computational and storage constraints.

1 Introduction

This paper describes the experiments performed by the Information Technology Institute for TREC-4. We are participating in TREC for the first time and have elected to perform only the main routing experiment.

We adopt a vector space model and attempt to construct the optimum routing queries using only the training documents. The subsequent sections contain the details of our *discriminant projection algorithm* which was used for this purpose.

We present the routing task as a standard problem in discriminant analysis. The standard solution involves the inversion of a large matrix whose dimension is the number of indexed terms. However, it is usually not possible to invert this matrix because the number of training documents are much smaller compared to the number of terms. This phenonenum is commonly called the *empty space* syndrome.

Consequently, it is desirable to project the raw document space into a lower dimensional space where each axis possesses higher representational richness. We show that linear orthonormal projections that efficiently encode the relevance information can be constructed using training documents only. We describe a projection algorithm that exploits the peculiarity of the empty space to construct the orthogonal basis on-the-fly. The complexity of this algorithm is linear with respect to the number

[1] Please direct all enquiries and correspondence to Kok F. Lai, Information Technology Institute, 11 Science Park Road, Singapore. Email address : kflai@iti.gov.sg.

[2] On attachment from Nanyang Technological University, Singapore, from Jan 8, 95 to Jun 18, 95.

of terms and second order with respect to the number of training documents. It first constructs a series of orthogonal axis using the Gram-Schmidt procedure, and then rotates the reduced space so that the axis are ordered by their importance. One can then discard insignificant axis to obtain an efficient representation of relevance information.

We then describe how one can perform discriminant analysis in this reduced space, and obtain the routing queries by projecting the results back into the original space. As we construct the projection basis on-the-fly, it is not necessary to alter the indexing structures, and no additional storage is required.

Due to time and resource contraints, we made several approximations in our TREC-4 implementations and these are described in this report. We will attempt to lift these approximations to improve the system's performance in subsequent participations.

We begin in the next section by describing the discriminant projection algorithm.

2 The Discriminant Projection Algorithm

2.1 Construction of Routing Query using Discriminant Analysis

We denote a document as a vector as follows :

$$\mathbf{d} = [f_1, f_2, \ldots f_N]^T \tag{1}$$

where f_j is a function of the term frequency for term j. N is the total number of unique indexed terms, and is typically very large. We shall assume that $\mathbf{d}$ is normalized, *i.e.*, $\|\mathbf{d}\| = 1$, in subsequent discussions.

Denote a document collection as $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_M]$. For a typical routing task, $\mathbf{D}$ consists of three separate groups : $\mathbf{D}_r$, $\mathbf{D}_{nr}$, and $\mathbf{D}_{na}$. $\mathbf{D}_r$ contains the set of M_r relevant documents; $\mathbf{D}_{nr}$ contains the set of M_{nr} non-relevant documents; and $\mathbf{D}_{na}$ contains the set of M_{na} documents where relevance information is unavailable.

Note that the total number of document, $M = M_r + M_{nr} + M_{na}$. Without loss of generality, let

$$\mathbf{D} = [\mathbf{D}_r : \mathbf{D}_{nr} : \mathbf{D}_{na}] \tag{2}$$

We want to find a *routing query* $\mathbf{q}$, such that $\mathbf{D}_r^T\mathbf{q} = \mathbf{1}$ and $\mathbf{D}_{nr}^T\mathbf{q} = \mathbf{0}$. Thus, we write

$$[\mathbf{D}_r : \mathbf{D}_{nr}]^T\mathbf{q} = [\mathbf{1} : \mathbf{0}]^T \tag{3}$$

Pre-multiplying both sides of the equation by $[\mathbf{D}_r : \mathbf{D}_{nr}]$, we have

$$\mathbf{R}\mathbf{q} = \mathbf{D}_r\mathbf{1} \tag{4}$$

where

$$\mathbf{R} = [\mathbf{D}_r : \mathbf{D}_{nr}]\begin{bmatrix} \mathbf{D}_r^T \\ \mathbf{D}_{nr}^T \end{bmatrix} \tag{5}$$

Now, observe that

$$\mathbf{D}_r\mathbf{1} = \sum_{i=1}^{M_r} \mathbf{d}_{r_i} = M_r\mathbf{m} \tag{6}$$

where $\mathbf{m}$ is the *mean* or *centroid* of the relevant documents. Suppose that the NxN matrix $\mathbf{R}$ is non-singular, we may then solve for $\mathbf{q}$ as follows:

$$\mathbf{q} = M_r \mathbf{R}^{-1} \mathbf{m} \tag{7}$$

Note that equation (7) is equivalent to the well-established *discriminant analysis* in statistical pattern recognition.

However, in most routing applications, the number of terms, N, is very much larger than the number of training documents $M_r + M_{nr}$. Thus $N >> M_r + M_{nr}$ and $\mathbf{R}^{-1}$ does not exist. Consequently it is impossible to solve for $\mathbf{q}$ from equation (7).

2.2 Solving Discriminant Analysis in a Reduced Space

Recall from the previous section that $N >> M_r + M_{nr}$ and $\mathbf{R}$ is singular. To solve for $\mathbf{q}$, it is necessary to project the raw document space into a lower dimensional space where each axis possesses higher representational richness.

We can project the raw document space $\mathbf{D}$ into a lower dimensional space $\mathbf{V}$ as follows:

$$\mathbf{V} = \mathbf{U}^T [\mathbf{D}_r : \mathbf{D}_{nr}] \tag{8}$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$ are the first k orthonormal basis that account for most information in the raw space. Using $\mathbf{V}$, we can then obtain the transformed routing query $\mathbf{q}_v$ as follows:

$$\mathbf{q}_v = M_r \mathbf{\Lambda}^{-1} \mathbf{m}_v \tag{9}$$

where $\mathbf{m}_v$ is the mean vector of relevant documents in the transformed space. Now

$$\mathbf{\Lambda} = \mathbf{V}\mathbf{V}^T = diag(\lambda_1, \lambda_2, \ldots, \lambda_k) \tag{10}$$

is a diagonal matrix, where λ_i are the eigenvalues of $\mathbf{R}$ in equation (5) in descending order. Note that $\mathbf{\Lambda}^{-1} = diag(\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_k^{-1})$ exists and thus $\mathbf{q}_v$ can be solved via equation (9).

We can then obtain the original query $\mathbf{q}$ in the raw space via

$$\mathbf{q} = \mathbf{U}\mathbf{q}_v = \sum_{i=1}^{k} q_{vi} \mathbf{u}_i \tag{11}$$

The above formulation is related to singular value decomposition[3] (SVD) employed by Latent Semantic Indexing [2]. However, the order of complexity of the standard SVD algorithm [3] is $O(N^2 + k^3)$. Since the total number of terms N is very large ($> 10^6$) for a typical application, this is very computational intensive. Moreover, SVD is usually pre-computed based on the entire document collection, which includes documents that carry no relevance information. As a result, the reduced dimensional space does not efficiently represent the relevance information contained in the training documents. The pre-computed SVD projection also substantially increases storage requirements [4].

We present an algorithm below, called *semantic projection*, that uses only the training documents to obtain the orthonormal basis. The algorithm reduces the order of computation substantially to approximately $O(N(M_r + M_{nr})^2)$. This allows it to compute the orthonormal basis on-the-fly and thus no additional storage requirement is necessary.

[3]Please refer to [1] for the relationships between singular values and eigenvalues.

2.3 Semantic Projection

Recall from the previous section that we wish to find the orthonormal basis $\mathbf{U}$ such that

$$\mathbf{V} = \mathbf{U}^T[\mathbf{D}_r : \mathbf{D}_{nr}] \tag{12}$$

It can be shown [1] that $\mathbf{U}$ can be computed via eigen-decomposition of the NxN matrix $\mathbf{R} = [\mathbf{D}_r : \mathbf{D}_{nr}][\mathbf{D}_r : \mathbf{D}_{nr}]^T$. However, due to the peculiar nature of text routing applications where frequently $N >> M_r + M_{nr}$, it will be advantageous to find an alternative method to compute $\mathbf{U}$.

The *semantic projection* algorithm uses only the $M_r + M_{nr}$ documents with relevance information to compute the optimum projection. It first contructs a subspace based on the training documents using the Gram-Schmidth Procedure. The dimension of this subspace will be at most $M_r + M_{nr}$. It then performs eigen-decomposition on this subspace to extract orthonormal basis which are ordered by their importance in capturing the subspace information. By specifying the desirable representation richness, only $k < M_r + M_{nr}$ orthonormal basis needs to be extracted.

2.3.1 Gram-Schmidt Procedure

The procedure makes use of training documents $\mathbf{d} \in [\mathbf{D}_r : \mathbf{D}_{nr}]$. We begin the procedure with the initialization

$$\mathbf{y}_1 = \mathbf{d}_1 \tag{13}$$

We then construct $\mathbf{y}_2$ as follows :

$$\begin{aligned} \mathbf{x}_2 &= \mathbf{d}_2 - \mathbf{d}_2^T\mathbf{y}_1\mathbf{y}_1 \\ \mathbf{y}_2 &= \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|} \end{aligned} \tag{14}$$

The vector $\mathbf{y}_2 = \mathbf{0}$ if and only if $\mathbf{d}_2 = a\mathbf{d}_1$, meaning that $\mathbf{d}_2$ is linearly dependent upon $\mathbf{d}_1$. Assume that $\mathbf{y}_2 \neq \mathbf{0}$, then $\mathbf{y}_1$ and $\mathbf{y}_2$ are linearly independent and orthogonal :

$$\mathbf{y}_2^T\mathbf{y}_1 = \mathbf{0} \tag{15}$$

We continue this procedure, generating each new vector $\mathbf{y}_j$ as follows :

$$\begin{aligned} \mathbf{x}_j &= \mathbf{d}_j - \sum_{i=1}^{j-1}(\mathbf{d}_j^T\mathbf{y}_i)\mathbf{y}_i \\ \mathbf{y}_j &= \frac{\mathbf{x}_j}{\|\mathbf{x}_j\|} \end{aligned} \tag{16}$$

This procedure maps the vectors $(\mathbf{d}_1, \mathbf{d}_2, \ldots \mathbf{d}_{M_r+M_{nr}})$ into vectors $(\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_{M_r+M_{nr}})$. In the latter set, there will be $r \leq M_r + M_{nr}$ nonzero vectors that the linearly independent which are retained. The discarded vectors are linearly dependent on the r linearly independent vectors and do not carry additional information on the training documents.

Note that the complexity of this algorithm is $O(Nr^2)$ and is linear with respect to the number of terms N. In the worst case, the complexity will be $O(N(M_r + M_{nr})^2)$.

2.3.2 Eigen-Decomposition of the Resultant Subspace

Without loss of generality, we write the resultant nonzero vectors as follows : $(\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_r)$. We can obtain the rxr correlation matrix $\mathbf{R}_y$ in this space by simply computing the projection on each training documents on the $\mathbf{y}$ vectors :

$$\mathbf{R}_y(i,j) = \sum_{m=1}^{r} \mathbf{d}_m^T \mathbf{y}_i \cdot \mathbf{d}_m^T \mathbf{y}_j \tag{17}$$

We can then apply an eigen-decomposition algorithm to efficiently extract eigenvectors of this r-dimensional subspace. In our implementation, we use the Hotelling's Iterative Procedure [5] to solve the eigen-decomposition problem. The procedure iteratively obtain eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \ldots$ that corresponds to eigenvalues $\lambda_1, \lambda_2, \ldots$ in decending order. One can simply terminate the procedure if a pre-specified threshold t has been exceeded :

$$\sum_{i=1}^{k} \lambda_i / tr(\mathbf{R}_y) \geq t \tag{18}$$

$tr(\mathbf{R}_y)$ is the trace of the matrix $\mathbf{R}_y$ that denotes the *total* variance available in the training documents. For example, if $t = 0.75$, then the first k eigenvectors account for more than 75% of the information in the training documents.

In the Hotelling's procedure, the iteration of each eigenvectors typically take only a few steps to converge. Since the algorithm operates on $\mathbf{R}_y$, its complexity is $O((M_r + M_{nr})^2)$ in the worst case.

Finally, the transformation vectors $\mathbf{u}$ can be obtained via

$$\mathbf{u}_i = \sum_{j=1}^{k} w_{ij} \mathbf{y}_j \tag{19}$$

Each $\mathbf{u}_i$, in diminishing importance, captures an orthogonal axis containing the semantics of the training documents. Thus $\mathbf{u}_1$ is a linear combination of terms containing the highest information contents in the training documents. $\mathbf{u}_1$ is orthogonal to $\mathbf{u}_2$ *i.e.*, $\mathbf{u}_1^T \mathbf{u}_2 = 0$: it captures information contents missed by $\mathbf{u}_1$, and so on. This set of orthonormal basis projects the original raw space into a reduced dimensional space while retaining the semantic information :

$$\mathbf{V} = \mathbf{U}^T [\mathbf{D}_r : \mathbf{D}_{nr}] \tag{20}$$

Thus the reduced space $\mathbf{V}$ is called the *semantic projection* of the training documents.

2.4 The Discriminant Projection Algorithm : Summary

In summary, the proposed routing algorithm is based on the well-established *discriminant analysis* as shown in equation (7). However, equation (7) by itself is unsolvable due to the *empty space* syndrome. The algorithm projects the training documents into a reduced space using a method called *semantic projection*, and solves for the routing query in this reduced space. The resulting query vector is then projected back into the original space.

The complexity of this algorithm is linear with respect to the number of terms N, and square with respect to the number of training documents $M_r + M_{nr}$. This makes it particularly attractive in relevance feedback environment where the number of feedback documents is typically very small.

We list the steps of the proposed *discriminant projection algorithm* below. Note that with a little housekeeping, steps 2 and 3 can be combined into a single operation.

1. Collect training documents $[\mathbf{D}_r : \mathbf{D}_{nr}]$. Obtain the mean vector $\mathbf{m}$ of the relevant documents by $\mathbf{m} = \frac{1}{M_r}\sum_{i=1}^{M_r} \mathbf{d}_{r_i}$.
2. Perform Gram-Schmidt procedure to obtain the r linearly independent vectors.
3. Obtain the rxr matrix $\mathbf{R}_y$ by computing the projection of each training documents with the r linearly independent vectors.
4. Use Hotelling's procedure to extract eigenvectors until required threshold is reached. Put the resulting k eigenvectors ($\mathbf{u}$) into the projection matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_k]$, and the eigenvalues ($\lambda$) into the diagonal matrix $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_k)$.
5. Project $\mathbf{m}$ to the reduced space : $\mathbf{m}_v = \mathbf{U}^T\mathbf{m}$.
6. Obtain the transformed routing query $\mathbf{q}_v = M_r\mathbf{\Lambda}^{-1}\mathbf{m}_v$.
7. Finally, obtain the routing query $\mathbf{q}$ in the original space by $\mathbf{q} = \mathbf{U}\mathbf{q}_v = \sum_{i=1}^{k} q_{vi}\mathbf{u}_i$.

3 TREC IV Implementations

As a first-time participant, we decided to take part in the main routing task only. Due to time and resource constraints, we experimented only with documents that appear on the relevant list of at least one query. All other documents that do not belong to this group are simply ignored. Consequently our training documents are drawn only from approximately 18,000 documents, rather than from the entire TREC collection.

Each document is automatically parsed and index terms representing the document content are retrieved. A simple token recognizer written in flex is used to identify words within the document structure that contributes to the content. Most words are indexed except very common words, maintained in a stopword list, and numbers are removed. The remaining words are then stemmed using the Porter's stemming algorithm. The document indexing process next creates the inverted files using Faircom's c-tree Plus file management product. Currently, without compression, the overhead for the indices amounts to about 80% of the original database size.

Using the training documents, we compute the *idf* weights for each term, *i.e.* $idf_i = log(N/n_i)$. Weights of terms seen only in test documents but not in training documents are set to 1.

For our first experiment, *itidp1*, we made no use of the topics and used only the relevant documents to construct the queries ($M_{nr} = 0$). Using this special case, it can be shown that step 5 of the algorithm reduces to

$$\mathbf{m}_v = \mathbf{U}^T\mathbf{m} = [1, 0, 0, \ldots, 0]^T \tag{21}$$

and therefore $\mathbf{q} = \mathbf{m}$. Thus the optimum routing query is simply the mean (centroid) of the relevant documents.

For our second experiment, *itidp2*, we enriched the training documents in *itidp1* by drawing from the 18,000 documents not marked relevant for the particular query. To further reduce the computational complexity, we did not invoke the Gram-Schmidt procedure for the non-relevant documents, and used these only to compute the projection into $\mathbf{R}_y$. This caused the complexity of the training algorithm to remain as $O(NM_r^2)$, in the expense of reduced representational richness.

For both sets of queries, we compute the projection scores $\mathbf{q}^T\mathbf{d}$ for the test documents $\mathbf{d}$, and ranked the result list in decending order.

4 Results

The following are the results of our implementation :

```
Queryid (Num):       all  itidp1
Total number of documents over all queries
    Retrieved:    42000
    Relevant:      5091
    Rel_ret:       3537
Interpolated Recall - Precision Averages:
    at 0.00       0.6882
    at 0.10       0.4499
    at 0.20       0.3763
    at 0.30       0.3209
    at 0.40       0.2641
    at 0.50       0.2257
    at 0.60       0.1821
    at 0.70       0.1304
    at 0.80       0.0660
    at 0.90       0.0234
    at 1.00       0.0002
Average precision (non-interpolated) over all rel docs
                  0.2282
Precision:
  At    5 docs:   0.5000
  At   10 docs:   0.4714
  At   15 docs:   0.4206
  At   20 docs:   0.4071
  At   30 docs:   0.3794
  At  100 docs:   0.2919
  At  200 docs:   0.2270
  At  500 docs:   0.1388
  At 1000 docs:   0.0842
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2782
```

```
Queryid (Num):       all  itidp2
Total number of documents over all queries
    Retrieved:    42000
    Relevant:      5091
    Rel_ret:       3104
```

```
Interpolated Recall - Precision Averages:
    at 0.00       0.6528
    at 0.10       0.4252
    at 0.20       0.3123
    at 0.30       0.2510
    at 0.40       0.1961
    at 0.50       0.1457
    at 0.60       0.1041
    at 0.70       0.0604
    at 0.80       0.0249
    at 0.90       0.0096
    at 1.00       0.0007
Average precision (non-interpolated) over all rel docs
                  0.1768
Precision:
  At    5 docs:   0.4714
  At   10 docs:   0.4524
  At   15 docs:   0.4254
  At   20 docs:   0.4083
  At   30 docs:   0.3738
  At  100 docs:   0.2664
  At  200 docs:   0.1973
  At  500 docs:   0.1177
  At 1000 docs:   0.0739
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2221
```

5 Conclusions

We presented a method for document routing using the discriminant projection algorithm. As a first-time participant in TREC, we faced time and resource contraints and had to make several approximations in our implementation. We wish to lift these approximations in our subsequent participation in TREC.

References

[1] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation and Time Series Analysis*, Addison-Wesley, 1991.

[2] S. T. Dumais, G. W. Furnas & T. K. Landauer, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, 1990, pp. 391-407.

[3] R. A. Harshman & M. E. Lundy, "Data preprocessing and the extended PARAFAC model," *Research Methods for MultiMode Data Analysis*, edited by H. G. Law et. al., 1984.

[4] D. Hull, "Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing," *Proc SIGIR '94*, 1994, pp. 282-291.

[5] H. Hotelling, "Analysis of a Complex Statistical Variables into Principal Components," *Journal of Educational Psychology*, 1933, pp. 417-441, 498-520.