An Algorithm for Text Categorization

Anestis Gkanogiannis
Athens University of Economics and Business
Informatics Department
76, Patission Str.
10434 Athens, Greece
utumno@aueb.gr

Theodore Kalamboukis
Athens University of Economics and Business
Informatics Department
76, Patission Str.
10434 Athens, Greece
tzk@aueb.gr

ABSTRACT

A novel and efficient learning algorithm is proposed for the binary linear classification problem. The algorithm is trained using the Rocchio's relevance feedback technique and builds a classifier by the intermediate hyperplane of two common tangent hyperplanes for the given category and its complement. Experimental results presented are very encouraging and justify the need for further research.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval;

General Terms: Algorithms, Experimentation, Measurement, Performance.

Keywords: Relevance Feedback, Text Categorization.

1. INTRODUCTION

Rocchio's relevance feedback technique[1, 2] is a query modification process that has been extensively investigated in the literature and used in information retrieval. Relevance feedback improves the query with terms that are considered relevant to the information seek. This is done iteratively either manually or automatically by selecting a predefined set of the top retrieved documents as relevant (pseudo-relevance feedback).

The aim is to find the optimum query, that is the query that maximizes the similarity with relevant documents while minimizing similarity with non-relevant documents. If $R(\bar{R})$ is the set of relevant(non-relevant) documents, then we wish to find, Q_{opt}, such that:

$$Q_{opt} = argmax_q \left(sim(q, R) - sim(q, \bar{R}) \right) \qquad (1)$$

where $sim(q, R)$ denotes the similarity of the query to the set of relevant documents. Using as similarity measure the *cosine* formula, we get that:

$$\overrightarrow{Q}_{opt} = \frac{1}{|R|} \sum_{d_j \in R} \overrightarrow{d}_j - \frac{1}{|\bar{R}|} \sum_{d_j \in \bar{R}} \overrightarrow{d}_j = \overrightarrow{C}_R - \overrightarrow{C}_{\bar{R}} \qquad (2)$$

that is, the optimum query is a vector defined by the difference of the centroids of the relevant and non-relevant documents, as it is shown in Figure 1a. In other words Q_{opt} defines a hyperplane, $h : \overrightarrow{Q}_{opt} \cdot \overrightarrow{x} - \theta = 0$ that separates the sets, R and $\bar{R}$, for an appropriate value of θ.

SIGIR'08, July 20–24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

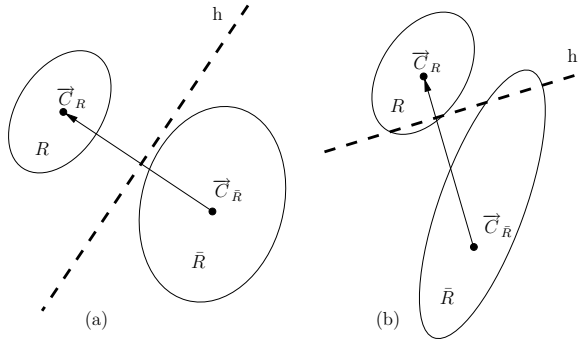

Figure 1: Vector $C_R - C_{\bar{R}}$ is not always the Optimum

This is not, however, always the case as it is shown in Figure 1b, where the query defined by equation (2) is not optimum although the sets R, $\bar{R}$ are linearly separable and as a consequense there is a hyperplane that separates them.

Algorithmically the relevance feedback process has been implemented by updating the query iteratively according to the following equation:

$$\overrightarrow{Q}_{i+1} = \kappa \overrightarrow{Q}_i + \frac{\lambda}{|R|} \sum_{d_j \in R} \overrightarrow{d}_j - \frac{\mu}{|\bar{R}|} \sum_{d_j \in \bar{R}} \overrightarrow{d}_j \qquad (3)$$

for a given initial query vector, $\overrightarrow{Q}_0$, where the constants κ, λ, μ are control parameters defined empirically. From (3) follows the conjecture that the optimal query is, in general, a linear combination of the relevant and non-relevant documents. In the following we shall use the Rocchio's feedback technique in constructing a classifier for a pair of linearly separable sets. The algorithm starts with an initial classifier, defined in (2), which is improved iteratively applying relevance feedback on the misclassified examples (Figure 1b). In the next section we describe the algorithm in more detail.

2. THE PROPOSED ALGORITHM

The algorithm constructs two common tangent hyperplanes for sets R, $\bar{R}$ such that these sets lie on opposite sides and the classifier is determined by the intermediate hyperplane of those tangent hyperplanes. If the two tangent hyperplanes are $\overrightarrow{W}_1 \cdot \overrightarrow{x} - \theta_1 = 0$ and $\overrightarrow{W}_2 \cdot \overrightarrow{x} - \theta_2 = 0$ respectively, then the classifier is $\overrightarrow{W} \cdot \overrightarrow{x} - \theta = 0$, where $\overrightarrow{W} = \frac{\overrightarrow{W}_1 + \overrightarrow{W}_2}{2}$ and $\theta = \frac{\theta_1 + \theta_2}{2}$.

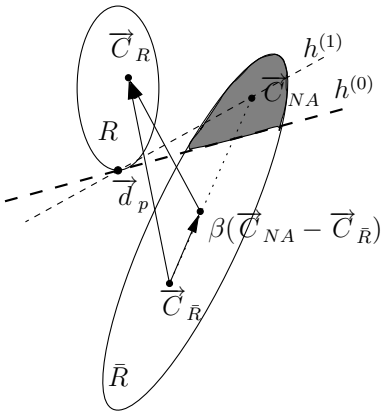

Figure 2: Rotation of hyperplane $h^{(0)}$ towards the misclassified examples

Initialization of the algorithm:

- Select initial vector $\overrightarrow{W}^{(0)} = \overrightarrow{C}_R^{(0)} - \overrightarrow{C}_{\bar{R}}^{(0)}$
- Calculate $s_j = \overrightarrow{W}^{(0)} \cdot \overrightarrow{d}_j \ , \ \forall \ d_j \in R \cup \bar{R}$
- Find s_p such that $s_p = min(\overrightarrow{W}^{(0)} \cdot \overrightarrow{d}_j) \ , \ \forall d_j \in R$

The hyperplane defined by $h^{(0)} : \overrightarrow{W}^{(0)} \cdot \overrightarrow{x} - \theta = 0$, with $\theta = s_p = \overrightarrow{W}^{(0)} \cdot \overrightarrow{d}_p$ by construction is vertical to the vector $\overrightarrow{W}^{(0)}$ and $\overrightarrow{d}_p$ lies on it (θ was defined at the value of $recall = 1$). By construction all the relevant documents lie on the same side of h, the one pointed by the vector $\overrightarrow{W}^{(0)}$. If it happens all the non-relevant examples to lie on the other side of h, then h is a separating hyperplane and the algorithm stops. This is not generally the case, as it is shown in Figure 2, where the hyperplane $h^{(0)}$, defined by the above process, intersects the set of non-relevant documents and the examples in the gray area are misclassified. In this case we rotate the hyperplane $h^{(0)}$ towards the misclassified examples (NA set, set of Negative Accepted examples) until all the negative examples lie on the same side of the hyperplane.

Rotation of $h^{(0)}$ towards the misclassified examples: This rotation is performed stepwise. At each step we determine the misclassified, NA, examples by h_1, construct their centroid vector, $\overrightarrow{C}_{NA}$, and then rotate the hyperplane towards the misclassified examples until it passes from the centroid $\overrightarrow{C}_{NA}$. This is equivalent to the process of moving $\overrightarrow{C}_{\bar{R}}$ towards $\overrightarrow{C}_{NA}$ by adding $\beta(\overrightarrow{C}_{NA} - \overrightarrow{C}_{\bar{R}})$, i.e. $\overrightarrow{C}_{\bar{R}} \leftarrow \overrightarrow{C}_{\bar{R}} + \beta(\overrightarrow{C}_{NA} - \overrightarrow{C}_{\bar{R}})$, such that the plane defined by $\overrightarrow{W}^{(1)}$, $\overrightarrow{W}^{(1)} = \overrightarrow{C}_R - (\overrightarrow{C}_{\bar{R}} + \beta(\overrightarrow{C}_{NA} - \overrightarrow{C}_{\bar{R}}))$ passes through the points $\overrightarrow{d}_p$ and $\overrightarrow{C}_{NA}$. With little algebra we estimate the value of β:

$$\beta = \frac{\overrightarrow{W}^{(0)} \cdot (\overrightarrow{C}_{NA} - \overrightarrow{d}_p)}{(\overrightarrow{C}_{NA} - \overrightarrow{C}_{\bar{R}}) \cdot (\overrightarrow{C}_{NA} - \overrightarrow{d}_p)} \tag{4}$$

This rotation however may cause examples in the set R to be misclassified (PR set, set of Positive Rejected examples). Thus we repeat the process on the set R by moving $\overrightarrow{C}_R$ towards the $\overrightarrow{C}_{PR}$, i.e. $\overrightarrow{C}_R \leftarrow \overrightarrow{C}_R + \alpha(\overrightarrow{C}_{PR} - \overrightarrow{C}_R)$, such that the plane defined by $\overrightarrow{W}^{(2)}$, $\overrightarrow{W}^{(2)} = (\overrightarrow{C}_R + \alpha(\overrightarrow{C}_{PR} - \overrightarrow{C}_R)) - \overrightarrow{C}_{\bar{R}}$, passes through the points $\overrightarrow{C}_{PR}$ and $\overrightarrow{C}_{NA}$. In the same way we determine α, by:

$$\alpha = -\frac{\overrightarrow{W}^{(1)} \cdot (\overrightarrow{C}_{PR} - \overrightarrow{C}_{NA})}{(\overrightarrow{C}_{PR} - \overrightarrow{C}_R) \cdot (\overrightarrow{C}_{PR} - \overrightarrow{C}_{NA})} \tag{5}$$

This alternation of the rotation towards either the NA or the PR continues until $|NA| = |PR| = 0$ or the number of iteration exceeds a predetermined value. This process converges to a common tangent hyperplane that leaves the sets R and $\bar{R}$ on opposite sides. In the same way we built a second common tangent and then we take as classifier their intermediate hyperplane.

Table 1: Performance results of the Proposed Algorithm, SVM and Generalized Centroid.

Dataset Name	Generalized Centroid $MicroF_1$	SVM $MicroF_1$	Proposed Algorithm $MicroF_1$
Reuters-10	0.8761	0.9166	0.9139
Reuters-90	0.8215	0.8536	0.8627
Reuters-10-x	0.9105	0.9441	0.9412
Reuters-90-x	0.8282	0.8656	0.8737
ReutersBIG-103	0.7241	0.8008	0.7951
Ohsumed-23	0.6121	0.5849	0.6449
OhsumedBIG-96	0.4934	0.5522	0.5750
OhsumedBIG-49	0.4971	0.5678	0.5797
OhsumedBIG-96-x	0.5827	0.6155	0.6250
OhsumedBIG-49-x	0.5951	0.6363	0.6411
Trec-20	0.5816	0.6742	0.6039
NG-20	0.6972	0.7853	0.8016
Reuters-8	0.9375	0.9659	0.9611
Reuters-52	0.9023	0.9283	0.9274

3. NUMERICAL RESULTS

For evaluation of the performance, scalability and robustness of the algorithm, five main corpuses were used; namely the Reuters-21578 corpus, the Reuters-RCV1 corpus, the OHSUMED corpus, the TREC-AP corpus and the 20 Newsgroup corpus. From these corpuses, we have constructed various subsets in order to explore the behavior of the algorithm under different synthesis and size of both the train and the testing set. These files have been constructed on a preprocessing step, applying stopword removal and stemming. The results are presented in Table 1 together with results obtained by SVM, a state of art classification algorithm. These results are very encouraging and justify the need for further research.

Acknowledgments

This work is part of a research project, co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%).

4. REFERENCES

[1] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART retrieval system: experiments in automatic document processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, USA, 1971.

[2] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. pages 355–364, 1997.