A Study on Optimal Parameter Tuning for Rocchio Text Classifier

Alessandro Moschitti

University of Rome *Tor Vergata*
Department of Computer Science Systems and Production
00133 Rome (Italy)
moschitti@info.uniroma2.it

Abstract. Current trend in operational text categorization is the designing of fast classification tools. Several studies on improving accuracy of *fast* but less accurate classifiers have been recently carried out. In particular, enhanced versions of the Rocchio text classifier, characterized by high performance, have been proposed. However, even in these extended formulations the problem of tuning its parameters is still neglected.
In this paper, a study on parameters of the Rocchio text classifier has been carried out to achieve its maximal accuracy. The result is a model for the automatic selection of parameters. Its main feature is to bind the searching space so that optimal parameters can be selected quickly. The space has been bound by giving a feature selection interpretation of the Rocchio parameters. The benefit of the approach has been assessed via extensive cross evaluation over three corpora in two languages. Comparative analysis shows that the performances achieved are relatively close to the best TC models (e.g. Support Vector Machines).

1 Introduction

Machine learning techniques applied to text categorization (TC) problems have produced very accurate although computationally complex models. In contrast, systems of real scenario such as Web applications and large-scale information management necessitate fast classification tools. Accordingly, several studies (e.g. [4, 6, 7]) on improving accuracy of low complexity classifiers have been carried out. They are related to the designing of efficient TC models in Web scenarios: feature space reduction, probabilistic interpretation of k-Nearest Neighbor and hierarchical classifiers are different approaches for optimizing speed and accuracy.

In this perspective, there is a renewed interest in the Rocchio formula. Models based on it are characterized by a low time complexity for both training and operative phases. The Rocchio weakness in TC application is that accuracy is often much lower than other more computationally complex text classifiers [19, 9]. Cohen and Singer [5] have suggested that a suitable tuning of parameters can improve the Rocchio text classifier accuracy. However, they did not propose

F. Sebastiani (Ed.): ECIR 2003, LNCS 2633, pp. 420–435, 2003.

a procedure for their estimation, as the parameters chosen to optimize the classification accuracy over the training documents were, in general, different from those optimizing the *test-set* classification. A possible explanation is that the searching in parameter space was made at random: a bunch of values for parameters was tried without applying a specific methodology.

Another attempt to enhance the Rocchio classifier is described in [14]. There, Schapire et al. show that Rocchio standard classifier can achieve the state-of-the art performances, although its efficiency is penalized. Improvements in accuracy are achieved by using more effective weighting schemes and *query zoning* methods, but a methodology for estimating Rocchio parameters was not considered.

Thus, the literature confirms the need of designing a methodology that automatically derives optimal parameters. Such a procedure should search parameters in the set of all feasible values. As no analytical procedure is available for deriving optimal Rocchio parameters, some heuristics are needed to limit the searching space. Our idea to reduce the searching space is to consider the feature selection property of the Rocchio formula. We will show that:

1. The setting of Rocchio parameters can be reduced to the setting of the rate among parameters.
2. Different values for the rate induce the selection of feature subsets ranked by relevance.
3. Only the features in the selected subset affect the accuracy of Rocchio classifier parameterized with the target parameter rate.
4. The parameter rate is inverse-proportional to the cardinality of the feature subset.

Therefore, increasing the parameter rate produces a subset collection of decreasing cardinality. Rocchio classifier, trained with these subsets, outcomes different accuracies. The parameter rate seems affect accuracy in the same way a standard feature selector [11] would do. From this perspective, the problem of finding optimal parameter rate can be reduced to the feature selection problem for TC and solved as proposed in [20].

Section 2 defines the Text Categorization problem and its accuracy measurements. The parameter setting algorithm and the underlying idea is presented in Section 3. The resulting system has been experimented via cross-validation over three different collections in two different languages (Italian and English) in Section 4. Finally, conclusions are derived in Section 5.

2 Profile-Based Text Classification

The classification problem is the derivation of a decision function f that maps documents ($d \in D$) into one or more classes, i.e. $f : D \rightarrow 2^C$, once a set of classes $C = \{C_1, ..., C_n\}$, i.e. topics labels (e.g. *Politics* and *Economics*), is given. The function f is usually built according to an extensive collection of examples classified into C_i, called *training-set*.

Profile-based text classifiers are characterized by a function f based on a similarity measure between the synthetic representation of each class C_i and the incoming document d. Both representations are vectors, and similarity is traditionally estimated as the cosine angle between the two. The description $\boldsymbol{C}_i$ of each target class C_i is usually called *profile*, that is, a vector summarizing all training documents d such as $d \in C_i$. Vector components are called *features* and refer to independent dimensions in the similarity space. Traditional techniques (e.g. [13]) employ words or stems as basic features. The i-th component of a vector representing a given document d is a numerical value. It is the weight that the i-th feature of the *training-set* assumes in d (usually evaluated as $TF \cdot IDF$ product [13]). Similarly, profiles are derived from the grouping of positive and negative instances d for the target category C_i. A newly incoming document is considered a member for a given class *iff* the similarity estimation overcomes established thresholds. The latter are parameters that adjust the trade-off between *precision* and *recall*. In the next section, the performance measures to derive text classifier accuracy are shown.

2.1 Accuracy Measurements

We have adopted the following performance measurements:

$$recall = \frac{\texttt{categ. found and correct}}{\texttt{total categ. correct}} = \frac{\texttt{cfc}}{\texttt{tcc}} \tag{1}$$

$$precision = \frac{\texttt{categ. found and correct}}{\texttt{total categ. found}} = \frac{\texttt{cfc}}{\texttt{tcf}} \tag{2}$$

To maintain a single performance measurement, the interpolated *Breakeven point* (BEP) could be adopted. This is the point in which the *recall* is equal to the *precision*. It can be evaluated starting the threshold from 0 and increasing it until the *precision* is equal to the *recall*. The mean is applied to interpolates the BEP if it does not exist. However, this may provide artificial results [15] when *precision* is not *close* enough to *recall*. The f_1-measure improves the BEP definition by using the harmonic mean between *precision* and *recall*, i.e. $f_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$.

In our experiments we have evaluated the thresholds associated with the maximal BEP on a *validation-set*[1]. Then, the performances have been derived from the *test-set* by adopting the previous thresholds. Finally, as global measures of a set of classifiers, we apply the *microaverage* to the target performance measures (i.e. *precision*, *recall*, BEP and f_1) over all categories. In this case, the correct and found categories of Eq. 1 and 2 are summed[2] before the microaverage computation of the *recall* and *precision* for a pool of k binary classifiers[3]. These measures are then used for evaluating the *microaverage* of BEP and f_1 (MicroAvg-f_1).

[1] A separate portion of the *training-set* used for parameterization purposes.

[2] The microaverage *recall* R and the microaverage *precision* P are respectively $\sum_{i=1}^{k} \texttt{cfc}_i / \sum_{i=1}^{k} \texttt{tcc}_i$ and $\sum_{i=1}^{k} \texttt{cfc}_i / \sum_{i=1}^{k} \texttt{tcf}_i$. The $MicroAvg\text{-}f_1$ is then evaluate as $\frac{2P \cdot R}{P+R}$.

[3] A binary classifier is a decision function that assigns or not a document to a single category.

3 Study on the Rocchio Parameter Spaces

The Rocchio's formula has been successfully used for building profiles in text classification [8] as follows. Given the set of training documents R classified under the topic C, the set $\bar{R}$ of training documents not classified in C, a document d and a feature f, the weight Ω_f assumed by f in the profile of C is:

$$\Omega_f = \max\left\{0, \frac{\beta}{|R|}\sum_{d\in R}\omega_f^d - \frac{\gamma}{|\bar{R}|}\sum_{d\in \bar{R}}\omega_f^d\right\} \tag{3}$$

where ω_f^d represents the weights[4] of features f in documents d. In Eq. 3, the parameters β and γ control the relative impact of positive and negative examples and determine the weight of f in the target profile. In [8], Eq. 3 has been used with values $\beta = 16$ and $\gamma = 4$ for the categorization task of low quality images. The success of these values possibly led to a wrong reuse of them in the TC task (e.g. [19]). In fact, as it has been pointed out in [5], these parameters greatly depend on the training corpus and different settings produce a significant variation in performances. Recently, some researchers [17] have found that $\gamma = \beta$ is a good setting for the document space, however, systematic methodologies for parameter setting were not definitively proposed.

3.1 Searching Space of Rocchio Parameters

As claimed in the previous section, to improve the accuracy of the Rocchio text classifier, parameter tuning is needed. The exhaustive search of optimal values for β and γ parameters is not a feasible approach as it requires the evaluation of Rocchio accuracy for all the pairs in the $\Re^2$ space.

To reduce the searching space, we observe that not both γ and β parameters are needed. In the following we show how to bind the parameter β to the threshold parameter. The classifier accepts a document d in a category C if the scalar product between their representing vectors is greater than a threshold s, i.e. $\boldsymbol{C}\cdot\boldsymbol{d} > s$. Substituting $\boldsymbol{C}$ with the original Rocchio's formula we get:

$$\left(\frac{\beta}{|R|}\sum_{d\in R}\omega_f^d - \frac{\gamma}{|\bar{R}|}\sum_{d\in \bar{R}}\omega_f^d\right)\cdot\boldsymbol{d} > s$$

and dividing by β,

$$\left(\frac{1}{|R|}\sum_{d\in R}\omega_f^d - \frac{\gamma}{\beta|\bar{R}|}\sum_{d\in \bar{R}}\omega_f^d\right)\cdot\boldsymbol{d} > \frac{s}{\beta} \Rightarrow \left(\frac{1}{|R|}\sum_{d\in R}\omega_f^d - \frac{\rho}{|\bar{R}|}\sum_{d\in \bar{R}}\omega_f^d\right)\cdot\boldsymbol{d} > s'.$$

Once ρ has been set, the threshold s' can be automatically assigned by the algorithm that evaluates the BEP. Note that, to estimate the threshold from a *validation-set*, the evaluation of BEP is always needed even if we maintain both parameters. The new Rocchio formula is:

[4] Several methods are used to assign weights to a feature, as widely discussed in [13].

$$\Omega_f = \max\left\{0, \frac{1}{|R|}\sum_{d\in R}\omega_f^d - \frac{\rho}{|\bar{R}|}\sum_{d\in \bar{R}}\omega_f^d\right\} \quad (4)$$

where ρ represents the *rate* between the original Rocchio parameters, i.e. $\frac{\gamma}{\beta}$.

Our hypothesis for finding *good* ρ value is that it deeply depends on the differences among classes in term of document contents. This enables the existence of different optimal ρ for different categories. If a correlation function between the category similarity and ρ is derived, we can bound the searching space.

We observe that in Equation 4, features with negative difference between positive and negative weights are set to 0. This aspect is crucial since the 0-valued features do not contribute in the similarity estimation (i.e. they give a null contribution to the scalar product). Thus, the Rocchio model does not use them. Moreover, as ρ is increased *smoothly*, only the features having a *high* weight in the negative documents will be eliminated (they will be set to 0 value). These features are natural candidates to be irrelevant for the Rocchio classifier. On one hand, in [11, 20] it has been pointed out that classifier accuracy can improve if irrelevant features are removed from the feature set. On the other hand, the accuracy naturally decreases if relevant and some weak relevant features are excluded from the learning [11]. Thus, by increasing ρ, irrelevant features are removed until performance improves to a maximal point, then weak relevant and relevant features start to be eliminated, causing Rocchio accuracy to decrease. From the above hypothesis, we argue that:

The best setting for ρ can be derived by increasing it until Rocchio accuracy reaches a maximum point.

In Section 4.2, experiments show that the Rocchio accuracy has the above behavior. In particular, the ρ/accuracy relationship approximates a convex curve with a single max point.

An explanation of linguistic nature could be that a target class C has its own specific set of terms (i.e. features). We define *specific-terms* as the set of words typical of one domain (i.e. very frequents) and at the same time they occur infrequently in other domains. For example, *byte* occurs more frequently in a *Computer Science* category than a *Political* one, so it is a *specific-term* for *Computer Science* (with respect to the *Politic* category).

The Rocchio formula selects *specific-terms* in C also by *looking* at their weights in the other categories C_x. If negative information is emphasized enough the *non specific-terms* in C (e.g., terms that occur frequently even in C_x) are removed. Note that these *non specific-terms* are misleading for the categorization. The term *byte* in political documents is not useful for characterizing the political domain. Thus, until the *non specific-terms* are removed, the accuracy increases since noise is greatly reduced. On the other hand, if negative information is too much emphasized, some *specific-terms* tend to be eliminated and accuracy starts to decrease. For example, *memory* can be considered *specific-terms* in *Computer Science*, nevertheless it can appears in *Political* documents; by emphasizing its negative weight, it will be finally removed, even from the Computer Science pro-

file. This suggests that the specificity of terms in C depends on C_x and it can be captured by the ρ parameter.

In the next section a procedure for parameter estimation of ρ over the *training-set* is presented.

3.2 Procedure for Parameter Estimation

We propose an approach that takes a set of training documents for profile building and a second subset, the *estimation-set*, to find the ρ value that optimizes the Breakeven Point. This technique allows parameter estimation over data independent of the *test-set* (TS), and the obvious bias due to the training material is avoided as widely discussed in [11]. The initial corpus is divided into a first subset of training documents, called *learning-set* LS, and a second subset of documents used to evaluate the performance, i.e. TS.

Given the target category, estimation of its optimal ρ parameter can be carried out according to the following *held-out* procedure:

1. A subset of LS, called *estimation set* ES is defined.
2. Set $i = 1$ and $\rho_i =$ Init_value.
3. Build the category profile by using ρ_i in the Eq. 4 and the *learning-set* $LS - ES$.
4. Evaluate the BEP_i for the target classifier (as described in Section 2.1) over the set ES.
5. Optionally: if $i > 1$ and $BEP_{i-1} \geq BEP_i$ go to point 8.
6. if $\rho_i >$ Max_limit go to point 8.
7. Set $\rho_{i+1} = \rho_i + \Sigma$, $i = i + 1$ and go to point 3.
8. Output ρ_k, where $k = argmax_i(BEP_i)$.

The minimal value for ρ (i.e. the Init_value) is 0 as a negative rate makes no sense in the feature selection interpretation. The maximal value can be derived considering that: (a) for each ρ, a different subset of features is used in the Rocchio classifier and (b) the size of the subset decrease by increasing ρ. Experimentally, we have found that $\rho = 30$ corresponds to a subset of 100 features out of 33,791 initial ones for the *Acquisition* category of Reuters corpus. The above feature reduction is rather aggressive as pointed out in [20] so, we chose 30 as our maximal limit for ρ.

However, in the feature selection interpretation of ρ setting, an objective maximal limit exists: it is the value that assigns a null weight to all features that are also present in the negative examples. This is an important result as it enables the automatic evaluation of the maximum ρ limit on training corpus in a linear time. It can be obtained by evaluating the rate between the negative and the positive contributions in Eq. 4 for each feature f and by taking the maximum value. For example we have found a value of 184.90 for the *Acquisition* category.

The values for Σ also (i.e. the increment for ρ) can be derived by referring to the feature selection paradigm. In [20, 19, 9] the subsets derived in their feature selection experiments have a decreasing cardinality. They start from the total number of unique features n and then select $n - i \cdot k$ features in the i-th subset; k varies between 500 and 5,000. When $\Sigma = 1$ is used in our estimation algorithm,

subsets of similar sizes are generated. Moreover, some preliminary experiments have suggested that smaller values for Σ do not select better ρ (i.e., they do not produce better Rocchio accuracy).

A more reliable estimation of ρ can be applied if steps 2-8 are carried out according to different, randomly generated splits ES_k and $LS - ES_k$. Several values $\rho(ES_k)$ can thus be derived at step k. A resulting $\bar{\rho}$ can be obtained by averaging the $\rho(ES_k)$. Hereafter we will refer to the Eq. 4 parameterized with estimated ρ values as the *Parameterized Rocchio Classifier* (PRC).

3.3 *PRC* Complexity

The evaluation of Rocchio classifier time complexity can be divided in to three steps: *pre-processing*, *learning* and *classification*. The *pre-processing* includes the document formatting and the extraction of features. We will neglect this extra time as it is common in almost all text classifiers.

The learning complexity for original Rocchio relates to the evaluation of weights in all documents and profiles. Their evaluation is carried out in three important steps:

1. The IDF is evaluated by counting for each feature the number of documents in which it appears. This requires the ordering of the pair set $<$*document, feature*$>$ by feature. The number of pairs is bounded by $m \cdot M$, where m is the maximum number of features in a documents and M is the number of training documents. Thus, the processing time is $O(m \cdot M \cdot log(m \cdot M))$.
2. The weight for each feature in each document is evaluated in $O(m \cdot M)$ time.
3. The profile building technique, i.e. Rocchio formula, is applied. Again, the tuple set $<$*document, feature, weight*$>$ is ordered by feature in $O(m \cdot M \cdot log(m \cdot M))$ time.
4. All weights that a feature f assumes in positive (negative) examples are summed. This is done by scanning sequentially the $<$*document, feature, weight*$>$ tuples in $O(M \cdot m)$ time. As result, the overall learning complexity is $O(m \cdot M \cdot log(m \cdot M))$.

The classification complexity of a document d depends on the retrieval of weights for each feature in d. Let n be the total number of unique features; it is an upperbound of the number of features in a profile. Consequently, the classification step takes $O(m \cdot log(n))$.

In the PRC algorithm, an additional phase is carried out. The accuracy produced by ρ setting has to be evaluated on a *validation-set* V. This requires the re-evaluation of profile weights and the classification of V for each chosen ρ. The re-evaluation of profile weights is carried out by scanning all $<$*document, feature, weight*$>$ tuples. Note that the tuples need to be ordered only one time. Consequently, the evaluation of one value for ρ takes $O(m \cdot M) + O(|V|m \cdot log(n))$. The number of values for ρ, as described in the previous section, is $k = Max\_limit/\Sigma$. The complexity to measure k values is $O(mM \cdot log(mM)) + k(O(m \cdot M) + |V| \cdot O(m \cdot log(n)))$. The cardinality of the *validation-set* $|V|$ as well

as k can be considered constants. In our interpretation, k is an intrinsic property of the target categories. It depends on feature distribution and not on the number of documents or features. Moreover, n is never greater than the product $M \cdot m$. Therefore, the final PRC learning complexity is $O(mM \cdot log(mM)) + k \cdot O(mM) + k|V| \cdot O(m \cdot log(mM)) = O(mM \cdot log(mM))$, i.e. the complexity of the original Rocchio learning.

The document classification phase of PRC does not introduce additional steps with respect to the original Rocchio algorithm, so it is characterized by a very efficient time complexity, i.e. $O(m \cdot log(n))$.

3.4 Related Work

The idea of parameter tuning in the Rocchio formula is not completely new. In [5] it has been pointed out that these parameters greatly depend on the training corpus and different settings of their values produce a significant variation in performances. However, a procedure for their estimation was not proposed as the parameters chosen to optimize the classification accuracy over the training documents were, in general, different from those optimizing the *test-set* classification. A possible explanation is that the searching in parameter space was made at random: a group of values for parameters was tried without applying a specific methodology. Section 4.3 shows that, when a systematic parameter estimation procedure is applied (averaging over a sufficient number of randomly generated samples), a reliable setting can be obtained.

Another attempt to improve Rocchio classifier has been provided via probabilistic analysis in [10]. A specific parameterization of the Rocchio formula based on the $TF \cdot IDF$ weighting scheme is proposed. Moreover, a theoretical explanation within a vector space model is provided. The equivalence between the probability of a document d in a category C (i.e. $P(C|d)$) and the scalar product $\boldsymbol{C} \cdot \boldsymbol{d}$ is shown to hold. This equivalence implies that the following setting for the Rocchio parameters: $\gamma = 0$ and $\beta = \frac{|C|}{|D|}$, where $|D|$ is the number of corpus documents. It is worth noting that the main assumption, at the basis of the above characterization, is $P(d|w, C) = P(d|w)$ (for words w descriptors of d). This ensures that $P(C|d)$ is approximated by the expectation of $\sum_{w \in d} P(C|w)P(w|d)$. The above assumption is critical as it assumes that the information brought by w subsumes the information brought by the pair $<w, C>$. This cannot be considered generally true. Since the large scale empirical investigation, carried out in Section 4.2, proves that the relevance of negative examples (controlled by the γ parameter) is very high, the approach in [10] (i.e., $\gamma = 0$) cannot be assumed generally valid.

In [17, 16] an enhanced version of the Rocchio algorithm has been designed for the problem of document routing. This task is a different instance of TC. The concept of category refers to the important document for a specific query. In that use of the Rocchio's formula, β parameter cannot be eliminated as it has been in Section 3.1. Moreover, an additional parameter α is needed. It controls the impact of the query in routing the relevant documents. The presence of three

parameters makes difficult an estimation of a good parameter set. The approach used in [17] is to try a number of values without a systematic exploration of the space. The major drawback is that the selected values could be only the local max of some document sets. Moreover, no study was done about the parameter variability. A set of values that maximize Rocchio accuracy on a *test-set* could minimize the performance over other document sets.

In [14] an enhanced version of Rocchio text classifier has been designed. The Rocchio improvement is based on better *weighting schemes* [1], on *Dynamic Feedback Optimization* [3] and on the introduction of *Query Zoning* [17]. The integration of the above three techniques has shown that Rocchio can be competitive with state-of-the art filtering approaches such as *Adaboost*. However, the problem of parameter tuning has been neglected. The simple setting $\beta = \gamma$ is adopted for every category. The justification given for such choice is that the setting has produced good results in [17]. The same reason and parameterization has been found even in [2] for the task of document filtering in TREC-9.

In summary, literature shows that improvements can be derived by setting the Rocchio parameters. However, this claim is neither proven with a systematic empirical study nor is a methodology to derive the good setting given. On the contrary, we have proposed a methodology for estimating parameters in a bound searching space. Moreover, in the next section we will show that our approach and the underlying hypotheses are supported by the experimental data.

4 Extensive Evaluation of *PRC*

The experiments are organized in three steps. First, in Section 4.2 the relationship between the ρ setting and the performances of Rocchio classifier has been evaluated. Second, PRC as well as original Rocchio performances are evaluated over the Reuters (fixed) *test-set* in Section 4.3. These results can be compared to other literature outcomes, e.g., [9, 19, 18, 5].

Additionally, experiments of Section 4.4 over different splits as well as different corpora in two languages definitely assess the viability of the PRC and the related estimation proposed in this paper. Finally, an evaluation of SVM on Ohsumed and Reuters corpora is given. This enables a direct comparison between PRC and one state-of-the art text classification model.

4.1 The Experimental Set-Up

Three different collections have been considered: The Reuters-21578[5] collection Apté split. It includes 12,902 documents for 90 classes, with a fixed splitting between *test-set* (here after RTS) and learning data LS (3,299 vs. 9,603); the Ohsumed collection[6], including 50,216 medical abstracts. The first 20,000 documents, categorized under the 23 *MeSH diseases* categories, have been used in

[5] Once available at `http://www.research.att.com/~lewis` and now available at `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`.

[6] It has been compiled by William Hersh and it is currently available at `ftp://medir.ohsu.edu/pub/ohsumed`.

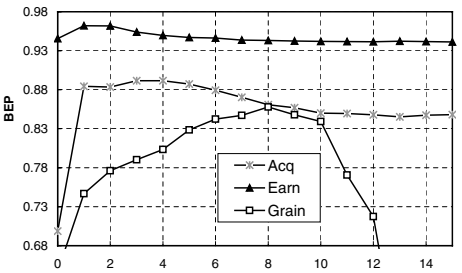

Fig. 1. Break-even point performances of the Rocchio classifier according to different ρ values for *Acq*, *Earn* and *Grain* classes of Reuters Corpus

all our experiments. The ANSA collection, which includes 16,000 news items in Italian from the ANSA news agency. It makes reference to 8 target categories (2,000 documents each). ANSA categories relate to typical newspaper contents (e.g. Politics, Sport and Economics).

Performance scores are expressed by means of *breakeven point* and f_1 (see Section 2.1). The global performance of systems is always obtained by *microaveraging* the target measure over all categories of the target corpus. The sets of features used in these experiments are all tokens that do not appear in the *SMART* [13] stop list[7]. They are 33,791 for Reuters, 42,234 for Ohsumed and 55,123 for ANSA. No feature selection has been applied. The feature weight in a document is the usual product between the logarithm of the feature frequency (inside the document) and the associated inverse document frequency (i.e. the SMART *ltc* weighting scheme [13]).

4.2 Relationship between Accuracy and ρ Values

In this experiments we adopted the fixed split of the Reuters corpus as our *test-set* (*RTS*). The aim here is simply to study as ρ influences the Rocchio accuracy. This latter has been measured by systematically setting different values of $\rho \in \{0, 1, 2, ..., 15\}$ and evaluating the BEP for each value.

Figures 1 and 2 shows the BEP curve on some classes of Reuters corpus with respect to ρ value. For *Earn*, *Acq* and *Grain* there is available a large number of training documents (i.e. from 2,200 to 500). For them, the BEP increases according to ρ until a max point is reached, then it begins to decrease for higher values of the parameter. Our hypothesis is that after BEP reaches the max point, further increase of ρ produces relevant or weakly relevant features to be removed. In this perspective, the optimal ρ setting would correspond to a quasi-optimal feature selection.

The *Trade*, *Interest* and *Money Supply* categories have a smaller number of documents available for training and testing (i.e. from 500 to 100). This reflects less regularity in ρ/BEP relationship. Nevertheless, it is still possible to identify

[7] No stop list was applied for Italian corpus.

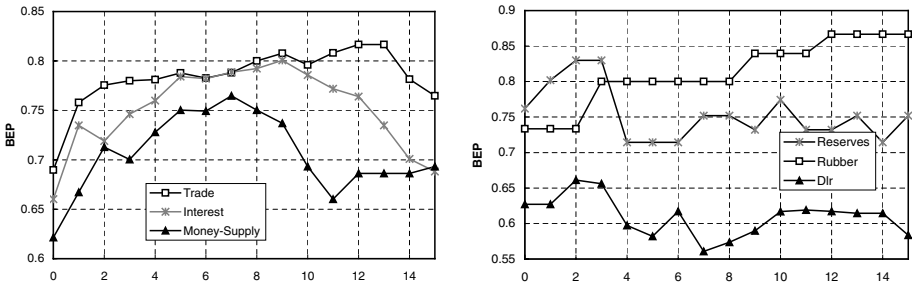

Fig. 2. Break-even point performances of the Rocchio classifier according to different ρ values for *Trade*, *Interest*, *Money Supply*, *Reserves*, *Rubber* and *Dlr* classes of Reuters corpus

convex curves in their plots. This is important as it allows us to infer that the absolute max is into the interval $[0, 15]$. The very small categories (i.e. less than 50 training documents) *Reserves*, *Rubber* and *Dlr* show a more chaotic relationship, and it is difficult to establish if the absolute maximum is in the target interval.

It is worth noting that the optimal accuracy is reached for $\rho > 1$. In contrast, is a common belief that for a classifier the positive information should be more relevant than negative information. This suggests that (a) in Rocchio classifier, the contribute of the feature weights in negative examples has to be emphasized and (b) the γ of Eq. 3 should not be interpreted as negative information control but as a simple parameter.

4.3 Performance Evaluation on Reuters Fixed *Test-Set*

In this experiment the performance of PRC model over the fixed Reuters *test-set* (RTS) has been measured. The aim is to provide direct comparison with other literature results (e.g. [19, 9, 5, 12]).

Twenty estimation sets $ES_1, ..., ES_{20}$ have been used to estimate the optimal rate as described in Section 3.2. Once $\bar{\rho}$ is available for the target category, its profile can be built and the performance can be measured. The PRC accuracy on RTS is a MicroAvg-f_1 of 82.83%. This score outperforms all literature evaluations of the original Rocchio classifier: 78% obtained in [5, 12], 75% in [19] and 79.9% in [9]. It is worth noting that this latter result has been obtained optimizing the parameters on RTS as the aim was to prove the SVM superiority independently on the parameters chosen (e.g. γ, β and thresholds) for Rocchio.

To investigate the previous aspect we have measured directly the original Rocchio parameterized as in literature: $\gamma = 4$ and $\beta = 16$ ($\rho = .25$) and with $\gamma = \beta$ ($\rho = 1$). The results are shown in columns 2 and 3 of Table 1. When $\rho = 1$ is used, the global performance (78.79%) replicates the results in [5, 12] while for $\rho = .25$, it is substantially lower (72.61%). The explanation is the high number of features used in our experiments without applying any feature

selection algorithm. A low rate ρ cannot filter an adequate number of irrelevant features and, consequently, the performances are low. As ρ increases, a high number of noised features is removed and the performances improve. PRC, by determining the best parameter ρ for each category, improves the Rocchio performance at least by 5 percent points.

To confirm the generality of the above results, cross validation experiments on Reuters and other corpora are presented in next section.

4.4 Cross Evaluation of Parameterized Rocchio Classifier

With the aim to assess the general performances of the PRC and of the original Rocchio classifier, wider empirical evidences are needed on different collections and languages. Moreover, to estimate the best TC accuracies achievable on the target corpora, we have also evaluated the Support Vector Machine (SVM) classifier [9].

Performance figures are derived for each category via a cross validation technique applied as follows:

1. Generate $n = 20$ random splits of the corpus: about 70% for training (LS^σ) and 30% for testing (TS^σ).
2. For each split σ
   (a) Extract 20 sample[8] ${ES^\sigma}_1 ... {ES^\sigma}_{20}$ from LS^σ.
   (b) Learn the classifiers on $LS^\sigma - {ES^\sigma}_k$ and for each ${ES^\sigma}_k$ evaluate: (i) the thresholds associated to the BEP and (ii) the optimal parameters ρ.
   (c) Learn the classifiers Rocchio, SVM and PRC on LS^σ: in case of PRC use the estimated $\bar{\rho}$.
   (d) Use TS_σ and the estimated thresholds to evaluate f_1 for the category and to account data for the final processing of the global MicroAvg-f_1.
3. For each classifier evaluate the mean and the Standard Deviation for f_1 and MicroAvg-f_1 over the TS_σ sets.

It is worth noting that the fixed *test-set* (RTS) and the *learning-set* of Reuters corpus have been merged in these experiments to build the new random splits.

Again, original Rocchio classifier has been evaluated on two different parameter settings selected from the literature (i.e. $\gamma = \beta$ and $\gamma = 4$ and $\beta = 16$). Table 1 reports the MicroAvg-f_1 over 90 categories and the f_1 (see Section 2.1) for the top 10 most populated categories. Original Rocchio accuracy is shown in columns 2, 3, 4 and 5. Columns 6 and 7 refer to PRC while columns 8 and 9 report SVM accuracy. The RTS label indicates that only the Reuters fixed *test-set* has been used to evaluate the results. In contrast, the TS^σ label means that the measurements have been derived averaging the results on 20 splits.

The symbol $\pm$ precedes the Std. Dev. associated to the mean. It indicates the variability of data and it can be used to build the confidence limits. We observe that our SVM evaluation on Reuters RTS (85.42%) is in line with the literature (84.2 %) [9]. The slight difference in [9] is due to the application of a stemming algorithm, a different weighting scheme, and a feature selection (only 10,000

[8] Each ES_k includes about 30-40% of training documents.

Table 1. Rocchio, SVM and PRC performance comparisons via f_1 and the MicroAvg-f_1 on the Reuters corpus. RTS is the Reuters fixed *test-set* while TS^{σ} indicates the evaluation over 20 random samples

	Rocchio				PRC		SVM	
Category	RTS		TS^{σ}		RTS	TS^{σ}	RTS	TS^{σ}
	$\rho = .25$	$\rho = 1$	$\rho = .25$	$\rho = 1$				
earn	95.69	95.61	92.57±0.51	93.71±0.42	95.31	94.01±0.33	98.29	97.70±0.31
acq	59.85	82.71	60.02±1.22	77.69±1.15	85.95	83.92±1.01	95.10	94.14±0.57
money-fx	53.74	57.76	67.38±2.84	71.60±2.78	62.31	77.65±2.72	75.96	84.68±2.42
grain	73.64	80.69	70.76±2.05	77.54±1.61	89.12	91.46±1.26	92.47	93.43±1.38
crude	73.58	80.45	75.91±2.54	81.56±1.97	81.54	81.18±2.20	87.09	86.77±1.65
trade	53.00	69.26	61.41±3.21	71.76±2.73	80.33	79.61±2.28	80.18	80.57±1.90
interest	51.02	58.25	59.12±3.44	64.05±3.81	70.22	69.02±3.40	71.82	75.74±2.27
ship	69.86	84.04	65.93±4.69	75.33±4.41	86.77	81.86±2.95	84.15	85.97±2.83
wheat	70.23	74.48	76.13±3.53	78.93±3.00	84.29	89.19±1.98	84.44	87.61±2.39
corn	64.81	66.12	66.04±4.80	68.21±4.82	89.91	88.32±2.39	89.53	85.73±3.79
MicroAvg. (90 cat.)	72.61	78.79	73.87±0.51	78.92±0.47	82.83	83.51±0.44	85.42	87.64±0.55

features were used there). It is worth noting that the global PRC and SVM outcomes obtained via cross validation are higher than those evaluated on the RTS (83.51% vs. 82.83% for PRC and 87.64% vs. 85.42% for SVM). This is due to the non-perfectly random nature of the fixed split that prevents a good generalization for both learning algorithms.

The cross validation experiments confirm the results obtained for the fixed Reuters split. PRC improves about 5 point (i.e. 83.51% vs. 78.92%) over Rocchio parameterized with $\rho = 1$ with respect to all the 90 categories (MicroAvg-f_1 measurement). Note that $\rho = 1$ (i.e. $\gamma = \beta$) is the best literature parameterization. When a more general parameter setting [5] is used, i.e. $\rho = .25$, PRC outperforms Rocchio by ~ 10 percent points. Table 1 shows a high improvement even for the single categories, e.g. 91.46% vs. 77.54% for the *grain* category. The last two columns in Table 1 reports the results for the linear version of SVM[9].

Tables 2 and 3 report the results on other two corpora, respectively Ohsumed and ANSA. The new data on these tables is the BEP evaluated directly on the TS^{σ}. This means that the estimation of thresholds is not carried out and the resulting outcomes are upperbounds of the real accuracies. We have used these measurements to compare the f_1 values scored by PRC against the Rocchio upperbounds. This provides a strong indication of the superiority of PRC as both tables show that Rocchio BEP is always 4 to 5 percent points under f_1 of PRC. Finally, we observe that PRC outcome is close to SVM especially for the Ohsumed corpus (65.8% vs. 68.37%).

[9] We have tried to set different polynomial degrees (1,2,3,4 and 5). As the linear version has shown the best performance we have adopted it for the cross validation experiments.

Table 2. Performance Comparisons among Rocchio, SVM and PRC on Ohsumed corpus

	Rocchio (BEP)		PRC		SVM
Category	$\rho = .25$	$\rho = 1$	BEP	f_1	f_1
Pathology	37.57	47.06	48.78	50.58	48.5
Cardiovasc.	71.71	75.92	77.61	77.82	80.7
Immunolog.	60.38	63.10	73.57	73.92	72.8
Neoplasms	71.34	76.85	79.48	79.71	80.1
Dig.Syst.	59.24	70.23	71.50	71.49	71.1
MicroAvg. (23 cat.)	54.4±.5	61.8±.5	66.1±.4	65.8±.4	68.37±.5

Table 3. Performance comparisons between Rocchio and PRC on ANSA corpus

	Rocchio (BEP)		PRC	
Category	$\rho = 0.25$	$\rho = 1$	BEP	f_1
News	50.35	61.06	69.80	68.99
Economics	53.22	61.33	75.95	76.03
Foreign Economics	67.01	65.09	67.08	66.72
Foreign Politics	61.00	67.23	75.80	75.59
Economic Politics	72.54	78.66	80.52	78.95
Politics	60.19	60.07	67.49	66.58
Entertainment	75.91	77.64	78.14	77.63
Sport	67.80	78.98	80.00	80.14
MicroAvg	61.76±.5	67.23±.5	72.36±.4	71.00±.4

5 Conclusions

The high efficiency of Rocchio classifier has produced a renewed interest in its application to operational scenarios. In this paper, a study on Rocchio text classifier parameters aimed to improve performances and to keep the same efficiency of the original version has been carried out. The result is a methodology for reducing the searching space of parameters: first, in TC only one parameter is needed, i.e., the rate ρ between γ and β. Secondly, ρ can be interpreted as a feature selector. This has allowed us to bind the searching space for the rate values since the ρ maximal value corresponds to the selection of 0 features. Moreover, empirical studies have shown that the ρ/BEP relationship can be described by a convex curve. This suggests a simple and fast estimation procedure for deriving the optimal parameter (see Section 3.1).

The Parameterized Rocchio Classifier (PRC) has been validated via cross validation, using three collections in two languages (Italian and English). In particular, a comparison with the original Rocchio model and the SVM text classifiers has been carried out. This has been done in two ways: (a) on the Reuters fixed split that allows PRC to be compared with literature results on TC and (b) by directly deriving the performance of Rocchio and SVM on the same data used for PRC. Results allow us to draw the following conclusions:

- First, PRC systematically improves original Rocchio parameterized with the best literature setting by at least 5 percent points, and it improves the general setting by 10 percent points. Comparisons with SVM show the performances to be relatively close (-4% on Reuters and -2.5% on Ohsumed).
- Second, the high performance, (i.e., 82.83%) on the Reuters fixed *test-set* collocates PRC as one of the most accurate classifiers on the Reuters corpus (see [15]).
- Third, the low time complexity for both training and classification phase makes the PRC model very appealing for real (i.e. operational) applications in Information Filtering and Knowledge Management.

Finally, the feature selection interpretation of parameters suggests a methodology to discover the *specific-term* of a category with respect to the other ones. A short-term future research may be the application of our methodology to estimate parameters in the enhanced Rocchio proposed in [14].

Acknowledgments

I would like to tank the AI-NLP group at *Tor Vergata* University of Rome and in especial way Roberto Basili for the fruitful discussions and suggestions. Thanks to the reviewers of ECIR for their punctual and careful reviews. Many thanks to the Technical Communication Lecturers, Kathy Lingo and Margaret Falersweany, that helped me to revise the English syntax of this article.

References

[1] Pivoted document length normalization. Technical Report TR95-1560, Cornell University, Computer Science, 1995. 428

[2] Avi Arampatzis, Jean Beney, C. H. A. Koster, and T. P. van der Weide. Incrementality, half-life, and threshold optimization for adaptive document filtering. In *the Nineth Text REtrieval Conference (TREC-9),Gaithersburg, Maryland*, 2000. 428

[3] Christopher Buckley and Gerald Salton. Optimization of relevance feedback weights. In *Proceedings of SIGIR-95*, pages 351–357, Seattle, US, 1995. 428

[4] Wesley T. Chuang, Asok Tiyyagura, Jihoon Yang, and Giovanni Giuffrida. A fast algorithm for hierarchical text classification. In *Proceedings of DaWaK-00*, 2000. 420

[5] William W. Cohen and Yoram Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999. 420, 423, 427, 428, 430, 432

[6] Harris Drucker, Vladimir Vapnik, and Dongui Wu. Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Neural Networks*, 10(5), 1999. 420

[7] Norbert Gövert, Mounia Lalmas, and Norbert Fuhr. A probabilistic description-oriented approach for categorising Web documents. In *Proceedings of CIKM-99*. 420

[8] David J. Ittner, David D. Lewis, and David D. Ahn. Text categorization of low quality images. In *Proceedings of SDAIR-95*, pages 301–315, Las Vegas, US, 1995. 423

[9] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *In Proceedings of ECML-98*, pages 137–142, 1998. 420, 425, 428, 430, 431

[10] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of ICML97 Conference*. Morgan Kaufmann, 1997. 427

[11] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997. 421, 424, 425

[12] Wai Lam and Chao Y. Ho. Using a generalized instance set for automatic text categorization. In *Proceedings of SIGIR-98*, 1998. 430

[13] G: Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988. 422, 423, 429

[14] Robert E. Schapire, Yoram Singer, and Amit Singhal. Boosting and Rocchio applied to text filtering. In W. Bruce Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98*, pages 215–223, Melbourne, AU, 1998. ACM Press, New York, US. 421, 428, 434

[15] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. 422, 434

[16] Amit Singhal, John Choi, Donald Hindle, and Fernando C. N. Pereira. ATT at TREC-6: SDR track. In *Text REtrieval Conference*, pages 227–232, 1997. 427

[17] Amit Singhal, Mandar Mitra, and Christopher Buckley. Learning routing queries in a query zone. In *Proceedings of SIGIR-97*, pages 25–32, Philadelphia, US, 1997. 423, 427, 428

[18] K. Tzeras and S. Artman. Automatic indexing based on bayesian inference networks. In *SIGIR 93*, pages 22–34, 1993. 428

[19] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval Journal*, 1999. 420, 423, 425, 428, 430

[20] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97*, pages 412–420, Nashville, US, 1997. 421, 424, 425