Semi-supervised Single-label Text Categorization using Centroid-based Classifiers

Ana Cardoso-Cachopo
IST — TULisbon / INESC-ID
Av. Rovisco Pais, 1
1049-001 Lisboa — Portugal
acardoso@ist.utl.pt

Arlindo L. Oliveira
INESC-ID / IST — TULisbon
Rua Alves Redol, 9
1000-029 Lisboa — Portugal
aml@inesc.pt

ABSTRACT

In this paper we study the effect of using unlabeled data in conjunction with a small portion of labeled data on the accuracy of a centroid-based classifier used to perform single-label text categorization. We chose to use centroid-based methods because they are very fast when compared with other classification methods, but still present an accuracy close to that of the state-of-the-art methods. Efficiency is particularly important for very large domains, like regular news feeds, or the web.

We propose the combination of Expectation-Maximization with a centroid-based method to incorporate information about the unlabeled data during the training phase. We also propose an alternative to EM, based on the incremental update of a centroid-based method with the unlabeled documents during the training phase.

We show that these approaches can greatly improve accuracy relatively to a simple centroid-based method, in particular when there are very small amounts of labeled data available (as few as one single document per class).

Using one synthetic and three real-world datasets, we show that, if the initial model of the data is sufficiently precise, using unlabeled data improves performance. On the other hand, using unlabeled data degrades performance if the initial model is not precise enough.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms

Algorithms, Experimentation, Performance

Keywords

Single-label Text Categorization, Centroid-based Models, Semi-supervised Learning, Online Learning

1. INTRODUCTION

Text Categorization (TC) is concerned with finding methods that, given a document, can automatically classify it into one or more of a predefined set of categories (or classes) [20]. When there is no overlap between classes, that is, when each document belongs to a single class, it is called single-label TC. In this paper, we are concerned with single-label TC.

A very efficient class of methods for TC is that of centroid-based methods [5, 10, 11, 8, 22, 13]. These methods are very efficient during the classification phase, because time and memory are proportional to the number of classes, rather than to the number of training documents. Despite their computational simplicity, these methods are very effective, even when compared with state-of-the-art methods like Support Vector Machines (SVM) [12, 3].

A characteristic common to many TC applications is that it is expensive to classify data for the training phase, while it is relatively inexpensive to find unlabeled data. In other situations, only a small portion of the document space is available initially, and new documents arrive incrementally. The web is a good example of a large, changing environment, with both these characteristics.

It has been shown that the use of large amounts of unlabeled data in conjunction with small amounts of labeled data can greatly improve the performance of some TC methods [17]. This has been done using a well known iterative algorithm called Expectation-Maximization (EM) [7].

In this paper, we propose the combination of EM with a centroid-based method to incorporate information about the unlabeled data during the training phase. We show that this approach can greatly improve accuracy relatively to a simple centroid-based method, in particular when there are very small amounts of labeled data.

For the situations where only a small portion of the document space is available initially, we incrementally update a centroid-based method with the unlabeled documents during the training phase.

It is particularly interesting that, by using a centroid-based method as the underlying classification method, we are able to compare the accuracy of semi-supervised and incremental learning directly, and choose the most adequate approach for each situation.

Using one synthetic and three real-world datasets, we show that, if the initial model of the data is sufficiently precise, using unlabeled data improves performance. On the other hand, using unlabeled data degrades performance if the initial model is not precise enough.

This paper is structured as follows: Section 2 briefly de-

scribes centroid-based methods and puts in context some of the work that has been done in this area. Section 3 describes how EM can be combined with a centroid-based method to incorporate information about the unlabeled data and refers to some other applications of EM to semi-supervised learning. Section 4 describes our proposal for incrementally incorporating information about the unlabeled data, as well as some of the work done on online learning. Section 5 describes the experimental setup that was used for this paper. Section 6 discusses the results that we obtained and compares them to published work. Finally, in Section 7, we conclude and refer some future directions for our work.

2. CENTROID-BASED METHODS

Centroid-based methods combine documents represented using a vector-space model [18], to find a representation for a "prototype" document that summarizes all the known documents for a given class, which is called the centroid. Given a set of n document vectors $D = \{\overrightarrow{d_1}, \ldots, \overrightarrow{d_n}\}$, classified along a set C of m classes, $C = \{C_1, \ldots, C_m\}$, we use D_{C_j}, for $1 \leq j \leq m$, to represent the set of document vectors belonging to class C_j. The centroid of a particular class C_j is represented by a vector $\overrightarrow{c_j}$, which is a combination of the document vectors $\overrightarrow{d_i}$ belonging to that class, sometimes combined with information about vectors of documents that are not in that class. There are several ways to calculate this centroid during the training phase and several proposals for centroid-based methods are available in the literature. Each proposal uses one possible way of calculating the centroids, which are similar, but produce different results. The most common are:

- The *Rocchio* formula, where each centroid, $\overrightarrow{c_j}$, is represented by the sum of all the document vectors for the positive training examples for class C_j, minus the sum of all the vectors for the negative training examples, weighted by control parameters β and γ, respectively. The application of this method to TC was first proposed by Hull [9] and it has been used in other works where the role of negative examples is deemphasized, by setting β to a higher value than γ (usually $\beta = 16$ and $\gamma = 4$) [5, 10, 11].

$$\overrightarrow{c_j} = \beta \cdot \frac{1}{|D_{C_j}|} \cdot \sum_{\overrightarrow{d_i} \in D_{C_j}} \overrightarrow{d_i} - \gamma \cdot \frac{1}{|D - D_{C_j}|} \cdot \sum_{\overrightarrow{d_i} \notin D_{C_j}} \overrightarrow{d_i} \quad (1)$$

- The *average* formula [8, 22], where each centroid, $\overrightarrow{c_j}$, is represented by the average of all the vectors for the positive training examples for class C_j:

$$\overrightarrow{c_j} = \frac{1}{|D_{C_j}|} \cdot \sum_{\overrightarrow{d_i} \in D_{C_j}} \overrightarrow{d_i} \quad (2)$$

- The *sum* formula [4], where each centroid, $\overrightarrow{c_j}$, is represented by the sum of all the vectors for the positive training examples for class C_j:

$$\overrightarrow{c_j} = \sum_{\overrightarrow{d_i} \in D_{C_j}} \overrightarrow{d_i} \quad (3)$$

- The *normalized sum* formula [13], where each centroid, $\overrightarrow{c_j}$, is represented by the sum of all the vectors for the

positive training examples for class C_j, normalized so that it has unitary length:

$$\overrightarrow{c_j} = \frac{1}{\| \sum_{\overrightarrow{d_i} \in D_{C_j}} \overrightarrow{d_i} \|} \cdot \sum_{\overrightarrow{d_i} \in D_{C_j}} \overrightarrow{d_i} \quad (4)$$

It is fairly obvious that these ways of calculating each class's centroid make centroid-based methods very efficient during the training phase, because there is little computation involved, unlike methods based on SVMs, which build a more sophisticated model of the data. Centroid-based methods also have the advantage that they are very easy to modify in order to perform incremental learning during their training phase, as we shall show in Section 4.

During the classification phase, each test document (or query) is represented by its vector, $\overrightarrow{d_i}$, and is compared to each of the class's centroids $\overrightarrow{c_j}$. The document will be classified as belonging to the class to whose centroid it has the greatest cosine similarity:

$$sim(\overrightarrow{d_i}, \overrightarrow{c_j}) = \frac{\overrightarrow{d_i} \cdot \overrightarrow{c_j}}{||\overrightarrow{d_i}|| \times ||\overrightarrow{c_j}||} \quad (5)$$

Centroid-based methods are very efficient during the classification phase because time and memory spent are proportional to the number of classes that exist, rather than to the number of training documents as is the case for the vector method and other related methods.

3. INCORPORATING UNLABELED DATA WITH EM

It has been shown that the use of large amounts of unlabeled data in conjunction with small amounts of labeled data can greatly improve the performance of some TC methods [17]. The combination of the information contained in the labeled and unlabeled data can be done using Expectation-Maximization (EM) [7].

Inputs: A set of labeled document vectors, L, and a set of unlabeled document vectors U.

Initialization step:
- For each class C_j appearing in L, set D_{C_j} to the set of documents in L belonging to class C_j.
- For each class C_j, calculate the class's centroid $\overrightarrow{c_j}$, using one of the formulas (1) to (4).

Estimation step:
- For each class C_j appearing in L, set U_{C_j} to the empty set.
- For each document vector $\overrightarrow{d_i} \in U$:
  - Let C_k be the class to whose centroid $\overrightarrow{d_i}$ has the greatest cosine similarity, calculated using Equation 5.
  - Add $\overrightarrow{d_i}$ to the set of document vectors labeled as C_k, i.e., set U_{C_k} to $U_{C_k} \cup \{\overrightarrow{d_i}\}$.

Maximization step:
- For each class C_j, calculate $\overrightarrow{c_{j_{new}}}$, using $D_{C_k} \cup U_{C_k}$ as the set of documents labeled as C_k, for each class C_k.

Iterate:
- If, for some j, $\overrightarrow{c_j} \neq \overrightarrow{c_{j_{new}}}$, then set $\overrightarrow{c_j}$ to $\overrightarrow{c_{j_{new}}}$ and repeat from the "Estimation step" forward.

Outputs: For each class C_j, the centroid $\overrightarrow{c_j}$.

EM is a class of iterative algorithms for maximum likelihood estimation of hidden parameters in problems with incomplete data. In our case, we consider that the labels of the unlabeled documents are unknown and use EM to estimate these (unknown) labels. EM has been used to combine labeled and unlabeled data for classification in conjunction with several different methods: Shahshahani and Landgrebe [21] use a mixture of Gaussians; Miller and Uyar [16] use mixtures of experts; McCallum and Nigam [15] use pool-based active learning; and Nigam [17] uses Naive Bayes. EM has also been used with k-means [14], which can be considered as a centroid-based method, but for clustering rather than for classification, under the name of constrained k-means [1, 2].

We propose the combination of EM with a centroid-based method for TC, which works according to the following algorithm, after choosing one of the formulas (1) to (4) to calculate each class's centroid.

4. INCREMENTALLY UPDATING THE MODEL OF THE DATA

Online methods [20] build a classifier soon after examining the first training document, and incrementally refine it as they examine new ones. This may be an advantage in the applications where the training set is not available in its entirety from the start, or in which the meaning of the category may change in time. We can describe the incremental method as a very generic algorithm:

Given a classification model, M, and a set of documents to classify, D, repeat for each document $d \in D$:

- Classify d according to model M
- Update model M with the new document d classified in the previous step
---

The incremental approach is very suited for tasks that require continuous learning, because the available data changes over time. Centroid-based methods are particularly suitable for this kind of approach because they are very fast, both for training and for testing, and can be applied to very large domains like the web. Moreover, unlike the traditionally used perceptron-based model [19, 23], which needs to train different classifiers to consider more than two classes, centroid-based models trivially generalize to multiple classes. In the case of single-label TC, there is no need to fine tune a threshold for deciding when a document belongs to a class, because it will belong to the class represented by the most similar centroid. In terms of computational efficiency, centroid-based methods are very fast, because updating a centroid-based method can be easily achieved, provided that we keep some additional information in the model with each centroid.

The next paragraphs describe how the model is updated for each of the centroid-based methods presented in Section 2. In each case, we want to update the model with a new document, $\overrightarrow{d_{new}}$, classified as belonging to class C_j. The simplest case is for the *sum* method (formula (3)), where the model is updated by calculating a new value for centroid $\overrightarrow{c_j}$ using the following attribution:

$$\overrightarrow{c_j} \hookleftarrow \overrightarrow{c_j} + \overrightarrow{d_{new}} \qquad (6)$$

To simplify the incremental update of the *average* method (formula (2)), we maintain in the model also the number of

documents, n_j, which were used to calculate each centroid $\overrightarrow{c_j}$. With this information, the model is updated according to the following attributions:

$$\overrightarrow{c_j} \hookleftarrow \frac{(\overrightarrow{c_j}.n_j) + \overrightarrow{d_{new}}}{n_j + 1} \qquad \text{and then} \qquad n_j \hookleftarrow n_j + 1 \quad (7)$$

For the *normalized sum* method (formula (4)), we maintain, with each normalized centroid $\overrightarrow{c_j}$, the non-normalized centroid, $\overrightarrow{nnc_j}$, so that updating the model can be performed by the following attributions:

$$\overrightarrow{nnc_j} \hookleftarrow \overrightarrow{nnc_j} + \overrightarrow{d_{new}} \qquad \text{and then} \qquad \overrightarrow{c_j} \hookleftarrow \frac{\overrightarrow{nnc_j}}{\parallel \overrightarrow{nnc_j} \parallel} \quad (8)$$

Finally, the most complex case is for the *Rocchio* method (formula (1)), because each new document forces the update of every centroid in the model. In this case, we maintain, for each centroid, $\overrightarrow{c_j}$, two vectors: the sum of the positive examples, $\overrightarrow{pos_j}$, and the sum of the negative examples, $\overrightarrow{neg_j}$. Using these two vectors, formula (1) can be rewritten as

$$\overrightarrow{c_j} = \beta \cdot \overrightarrow{pos_j} - \gamma \cdot \overrightarrow{neg_j} \qquad (9)$$

Updating the model can be achieved by first updating the appropriate vectors:

$$\overrightarrow{pos_j} \hookleftarrow \overrightarrow{pos_j} + \overrightarrow{d_{new}} \text{ and, for each } i \neq j, \overrightarrow{neg_i} \hookleftarrow \overrightarrow{neg_i} + \overrightarrow{d_{new}}$$
$$(10)$$

and then, calculating all the new centroids according to formula (9).

In this paper we show how incrementally updating the model of the data (that is, the centroids) with the unlabeled documents during the training phase influences the accuracy of a centroid-based method.

5. EXPERIMENTAL SETUP

In this section we present the experimental setup that was used for this work, namely the datasets and the evaluation measures that were used. In order to show that using unlabeled data improves performance when the initial model of the data is sufficiently precise, while hurting performance if the initial model is not precise enough, we used one synthetic and three real-world datasets.

5.1 Synthetic Dataset

The synthetic dataset corresponds to four different mixtures of Gaussians, in one dimension. The data points belonging to each Gaussian distribution are randomly generated according to a Gaussian probability distribution function:

$$g(x) = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}} \qquad (11)$$

In each combination, each Gaussian distribution corresponds to a different class, and we used different ratios between parameters μ and σ to simulate problems with differing difficulties. Figure 1 depicts the Gaussian distributions that we used. Is is easy to see that, as the ratio $\frac{\mu}{\sigma}$ decreases, the problem of deciding which distribution originated a randomly generated point belongs is more difficult, because the overlap between the distributions increases. In particular, the limit for the accuracy of the optimal classifier can be obtained as the value of the cumulative distribution function of the Gaussian at the point of intersection.
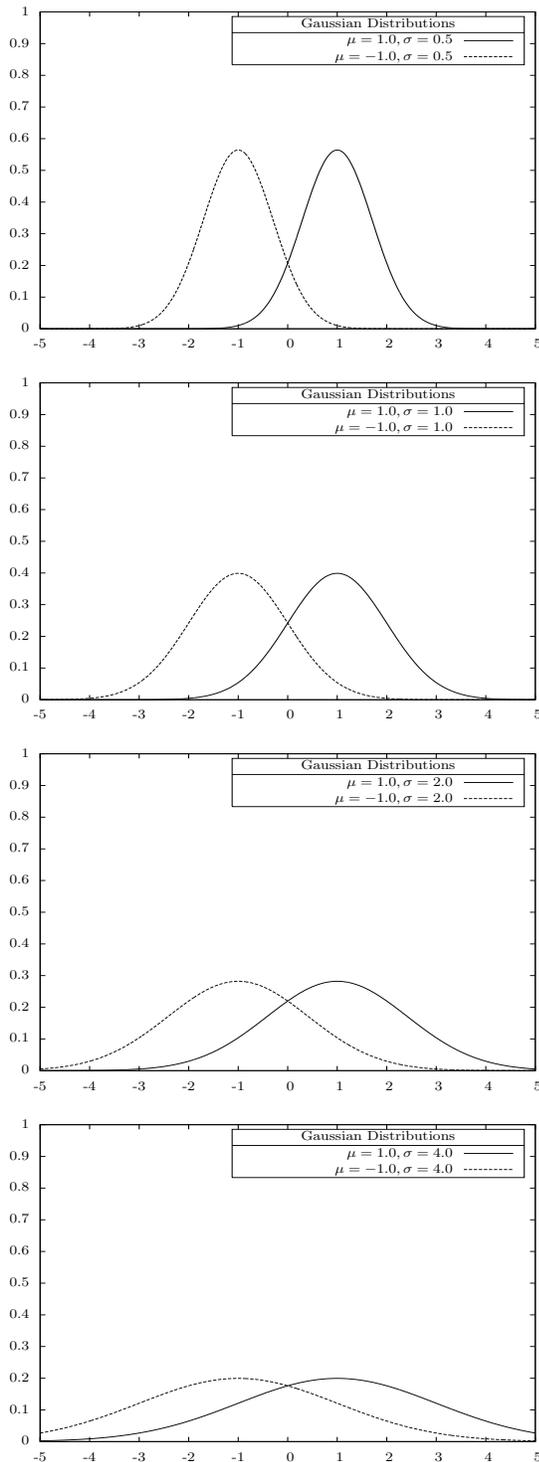
Figure 1: Combinations of Gaussians that were used.

5.2 Real-world Datasets

To allow the comparison of our work with previously published results, we used two standard TC benchmarks in our evaluation, downloaded from a publicly available repository of datasets for single-label text categorization.[1] In this website there is also a description of the datasets, their standard train/test splits, how they were processed to become single-labeled, and the pre-processing techniques that were applied to each dataset, namely character clean-up, removal of short words, removal of stopwords, and stemming. We also used a set of classified web pages extracted from the CADÊ Web Directory.[2]

20 Newsgroups — The **20ng** dataset is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. We used its standard "ByDate" split, where documents are ordered by date and the first two thirds are used for training and the remaining third for testing. For this dataset, we used the files `20ng-train-stemmed` and `20ng-test-stemmed`, available from that website.

Reuters 21578 — The documents in Reuters-21578 appeared on the Reuters newswire in 1987 and were manually classified by personnel from Reuters Ltd. We used the standard "modApté" train/test split. Due to the fact that the class distribution for these documents is very skewed, two sub-collections are usually considered for text categorization tasks [6]: R10, the set of the 10 classes with the highest number of positive training examples; and R90, the set of the 90 classes with at least one positive training and testing example. Because we are concerned with single-label TC, we used **r8**, which corresponds to the documents with a single topic and the classes which still have at least one training and one test example after removing documents with more than one topic from R10. For this dataset, we used the files `r8-train-stemmed` and `r8-test-stemmed`, available from that website.

CADE — The documents in the **Cade12** dataset correspond to web pages extracted from the CADÊ Web Directory, which points to Brazilian web pages classified by human experts. This dataset corresponds to a subset of the pages that are available through that directory. We randomly chose two thirds of the documents for training and the remaining third for testing.

Table 1 shows some information about these datasets, namely the number of classes and the numbers of documents in the train and test sets.

5.3 Evaluation Measure

TC methods are usually evaluated in terms of measures based on Precision and Recall, like F1 or PRBP [20]. However, to evaluate *single-label* TC tasks, these measures are not adequate, because Recall does not make sense in this setting. So, accuracy, which is the percentage of correctly classified test documents (or queries), is used to evaluate this kind of tasks. We will therefore use accuracy as the criterion for evaluating the performance of the algorithms.

$$Accuracy = \frac{\#\text{Correctly classified test documents}}{\#\text{Total test documents}} \quad (12)$$

---

[1] Available at `http://www.gia.ist.utl.pt/~acardoso/datasets/`
[2] Available at `http://www.cade.com.br`, in Brazilian Portuguese.

20ng (20 classes)	Train	Test	Total
alt.atheism	480	319	799
comp.graphics	584	389	973
comp.os.ms-windows.misc	572	394	966
comp.sys.ibm.pc.hardware	590	392	982
comp.sys.mac.hardware	578	385	963
comp.windows.x	593	392	985
misc.forsale	585	390	975
rec.autos	594	395	989
rec.motorcycles	598	398	996
rec.sport.baseball	597	397	994
rec.sport.hockey	600	399	999
sci.crypt	595	396	991
sci.electronics	591	393	984
sci.med	594	396	990
sci.space	593	394	987
soc.religion.christian	598	398	996
talk.politics.guns	545	364	909
talk.politics.mideast	564	376	940
talk.politics.misc	465	310	775
talk.religion.misc	377	251	628
Total	11293	7528	18821
r8 (8 classes)	Train	Test	Total
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Total	5485	2189	7674
Cade12 (12 classes)	Train	Test	Total
01–servicos	5627	2846	8473
02–sociedade	4935	2428	7363
03–lazer	3698	1892	5590
04–informatica	2983	1536	4519
05–saude	2118	1053	3171
06–educacao	1912	944	2856
07–internet	1585	796	2381
08–cultura	1494	643	2137
09–esportes	1277	630	1907
10–noticias	701	381	1082
11–ciencias	569	310	879
12–compras-online	423	202	625
Total	27322	13661	40983

Table 1: List of classes and number of documents for each dataset.

It can be shown that, in single-label classification tasks,

$$Accuracy = microaveraged\ F1 =$$
$$microaveraged\ Precision = microaveraged\ Recall \quad (13)$$

because each document can be correctly classified or not, so the number of false positives in contingency tables is the same as the number of false negatives.

5.4 Preliminary Testing

In order to be able to compare our work with some other TC methods, we have determined the accuracy that some of the most common methods achieve in our datasets. As usual, *tfidf* term weighting is used to represent document vectors, and they were normalized to unitary length. It has already been shown that, of the several centroid-based methods proposed in the literature, Centroid-NormalizedSum (the one that uses formula (4) to calculate the centroids) was the best performing one [3], and so we used it in our experiments.

For comparison purposes, Table 2 shows the accuracy obtained with some well known TC methods using our framework and **all the training documents** as labeled documents. The "dumb classifier" ignores the contents of the test document and always gives as the predicted class the most frequent class in the training set.

	r8	20ng	Cade12
Dumb classifier	0.4947	0.0530	0.2083
Vector	0.7889	0.7240	0.4142
k-NN (k = 10)	0.8524	0.7593	0.5120
Centroid-NormalizedSum	0.9543	0.7885	0.5147
SVM (linear kernel)	0.9698	0.8278	0.5283

Table 2: Accuracy achieved by some TC methods using our framework, considering all training documents as labeled.

Note that, because r8 is very skewed, the dumb classifier has a "reasonable" performance for this dataset. Also, it is worth noting that, while for r8 and 20ng we can find good classifiers, that is, classifiers that achieve a high accuracy, for Cade12 the best we can get does not reach 53% accuracy, even with one of the best classifiers available.

6. EXPERIMENTAL RESULTS

In this section we provide empirical evidence that, if the initial model of the data is sufficiently precise, using unlabeled data improves performance, and that, on the other hand, using unlabeled data degrades performance if the initial model is not precise enough. We do this, first by using one synthetic dataset, and then confirming the results obtained using three real-world datasets.

6.1 Using Unlabeled Data with the Synthetic Dataset

The synthetic dataset was created with several goals in mind. First, we wanted a dataset that was simple, and whose properties were well known. Additionally, we wanted to be able to generate as many "documents" as necessary for our experiments. Ultimately, our goal was to prove that the effect of using unlabeled data depends not only on the classification method that is used, but also on the quality of the dataset.

We randomly generated four different two-class datasets, each according to two different one-dimensional Gaussian distributions, so that each dataset posed a different difficulty level, known in advance. With each dataset, we used the same classification methods, and in the end we compared the results.

So that our experiments would not depend on one particular ordering of the dataset, we repeated the following steps 500 times for each dataset:

1. Randomly generate from 1 to 20 labeled training documents per class.
2. Based on the training documents alone, calculate each class's centroid, that is, the average of the numbers corresponding to the training documents.
3. Randomly generate 5000 test documents. These should be approximately half from each class.
4. Determine accuracy for the centroid-based method.
5. Randomly generate 5000 unlabeled documents.
6. Using EM / incremental, update each class's centroid.

7. Determine accuracy for EM / incremental, using the same test documents as for the centroid-based method alone.

This can be summed-up in the following algorithm:

```
For each dataset, repeat 500 times
 For i = 1 to 20
  Randomly generate i training docs per class
  Calculate each class's centroid
  Randomly generate 5000 test docs
  Determine accuracy for the centroid-based method
  Randomly generate 5000 unlabeled docs
  Using EM / incremental, update centroids
  Determine accuracy for EM / incremental
```

The mean accuracy values for the 500 tests for each dataset, as a function of the number of labeled documents per class that were used, are presented in figure 2.

We can observe that there are some features that are common, independently of the difficulty level of the dataset:

- As expected, more labeled training documents per class improve results, because the curves go up as the number of available labeled documents increases. However, this observed improvement decreases as the number of labeled documents increases.
- As the number of labeled training documents per class increases, the effect of using unlabeled documents decreases. This means that, having unlabeled data is always good, and it is better when we have less labeled data.
- Both methods (EM and incremental) of updating the centroids of the classes give the same results, because both lines are the same. In this setting, it is better to incrementally update the centroids, because the results are the same and this method is computationally more efficient.

As for the features that depend on the difficulty level of the dataset:

- As the difficulty level of the dataset increases, the accuracy that can be achieved decreases (note the different ranges in the Y axis).
- When only one labeled document per class is available, 5000 unlabeled documents allow us to improve accuracy from 0.9196 to 0.9771, for the easier dataset, while for the most difficult dataset improvement is only from 0.5252 to 0.5331.
- As a general rule, the effect of using 5000 unlabeled documents to update our model of the data decreases as the difficulty level of the dataset increases.

All these observations allow us to conclude that the effect of using unlabeled data depends not only on the classification method that is used, but also on the quality of the dataset that we are considering. If our initial model of the labeled data is able to achieve a high accuracy, using unlabeled data will help improve the results, and it will help more if the initial model is better.

6.2 Using Unlabeled Data with the Real World Datasets

To confirm the results obtained in the previous section, we used two standard TC benchmarks and a set of classified
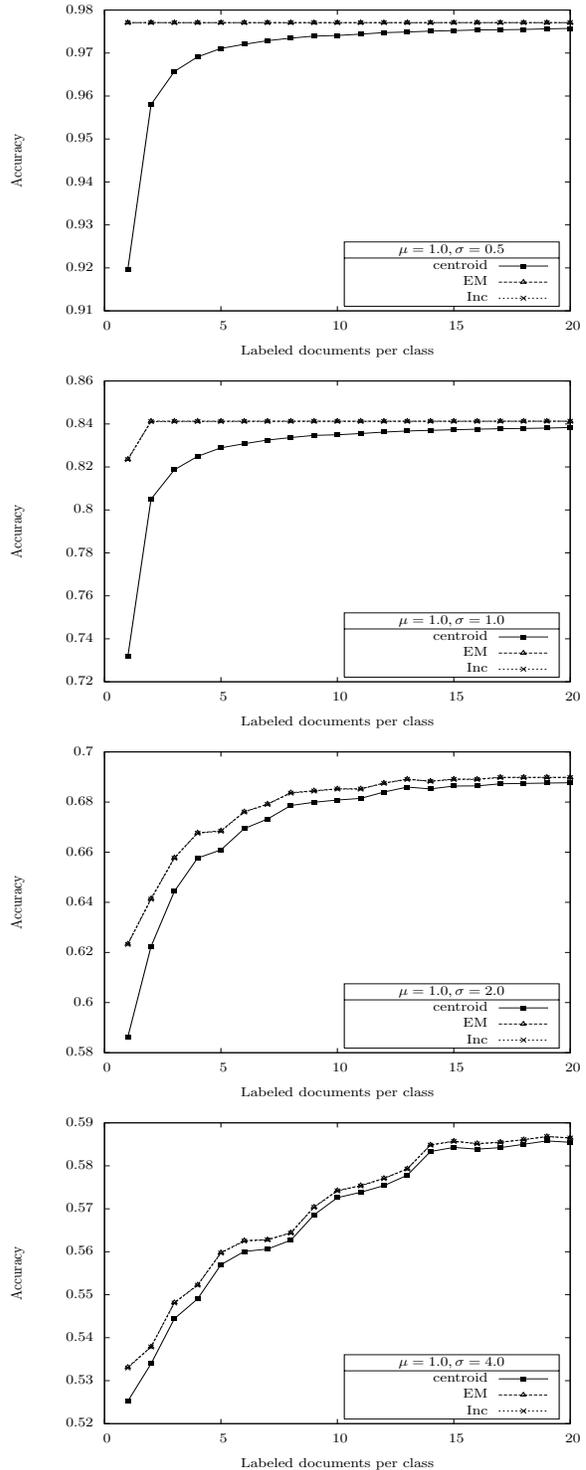

Figure 2: **Accuracy for the Gaussians dataset, as a function of the number of labeled documents per class that were used.**

web pages extracted from the CADÊ Web Directory. As we already saw in Section 5.4, we were able to find good classifiers for the first two datasets, but not for the third.

The steps we followed for testing these datasets were similar to the ones followed in the previous section, and can be summed-up in the following algorithm[3]:

```
For each dataset, consider its train/test split
For each dataset, repeat 5 times
 For i in {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15,
          20, 30, 40, 50, 60, 70}
  Randomly select i labeled training docs per class
  Calculate each class's centroid using Formula 4
  Determine accuracy for the centroid-based method
  Randomly select 5000 unlabeled docs from the
          remaining training docs
  Using EM / incremental, update centroids
  Determine accuracy for EM / incremental
```

The mean accuracy values for the 5 runs for each dataset are presented in Figure 3.

Once more, we can observe that, independently of the dataset that is used, more labeled training documents per class improve results, and that this improvement decreases as the number of labeled documents increases.

The rest of the observations depend on the dataset that is used:

- r8, for which it was possible to achieve a high accuracy using all 5485 training documents with SVM and Centroid-NormalizedSum (see Table 2), accuracy varies from 0.6492 with one labeled document per class to 0.9321 with 40 labeled documents per class. 20ng, for which it was also possible to achieve high accuracy values with Centroid-NormalizedSum using all 11293 training documents and SVM, accuracy varies from 0.2627 with one labeled document per class to 0.7424 with 70 labeled documents per class. For Cade12, which already had poor accuracy values using all 27322 training documents with Centroid-NormalizedSum and SVM, accuracy varies from 0.1212 with one labeled document per class to 0.3782 with 70 labeled documents per class (note the different ranges in the Y axis).
- For r8 and 20ng, using unlabeled data improves results, and the improvement is larger for smaller numbers of labeled documents per class. For Cade12, using unlabeled data worsens results. This observation allows us to experimentally confirm our initial intuition that it is only worth it to use unlabeled data if the initial model for the labeled data already provided good results.
- For r8 and 20ng, as the number of labeled training documents per class increases, the effect of using unlabeled documents decreases. As for the synthetic dataset, having unlabeled data is good, and it is better when we have less labeled data.
- For all datasets, the two methods (EM and incremental) of updating the centroids of the classes give different results, because now the way by which the centroids are updated is different. Moreover, the difference in the results decreases as the number of labeled

---

[3]Due to its reduced size, for the r8 dataset we could only select up to 40 labeled documents per class and 1000 unlabeled documents.
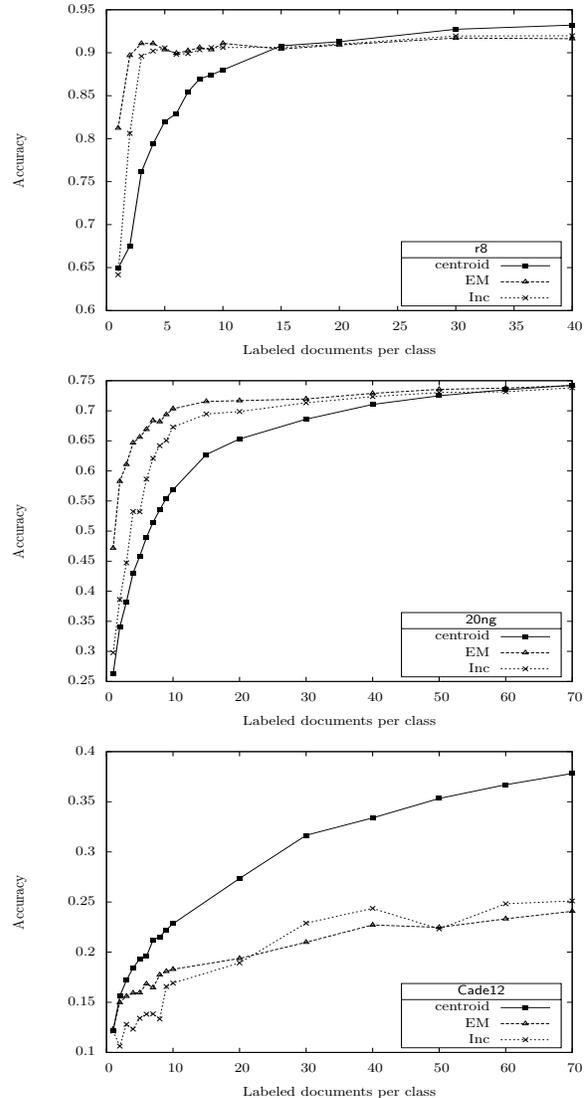

Figure 3: Accuracy for the three real world datasets, as a function of the number of labeled documents per class that were used.

documents increases. This happens because when more labeled documents are available, the initial model is better, and therefore the centroids are less moved by either one of the updating techniques. Generally, using EM to update the centroids yields better results.

All these observations allow us to confirm our previous conclusion that the effect of using unlabeled data depends not only on the classification method that is used, but also on the quality of the dataset that we are considering. For the datasets for which we could come up with good classification accuracy (r8 and 20ng), using unlabeled data helped improve results, while for the dataset for which the initial results were not so good (Cade12), using unlabeled data actually made our results even worse.

7. CONCLUSIONS AND FUTURE WORK

In this paper we are concerned with single-label text categorization. We proposed the combination of EM with a centroid-based classifier that uses information from small amounts of labeled documents together with information from larger amounts of unlabeled documents. We also showed how a centroid-based method can be used to incrementally update the model of the data, based on new evidence from the unlabeled data. Using one synthetic dataset and three real-world datasets, we provided empirical evidence that, if the initial model of the data is sufficiently precise, using unlabeled data improves performance. On the other hand, using unlabeled data degrades performance if the initial model is not precise enough. As future work, we plan to extend this approach to multi-label datasets.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. Mooney. Model-based overlapping clustering. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 532–537. ACM Press, 2005.

[2] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM Press, 2004.

[3] A. Cardoso-Cachopo and A. Oliveira. Empirical evaluation of centroid-based models for single-label text categorization. Technical Report 7/2006, INESC-ID, June 2006.

[4] W. Chuang, A. Tiyyagura, J. Yang, and G. Giuffrida. A fast algorithm for hierarchical text classification. In *Proceedings of DaWaK-00, 2nd International Conference on Data Warehousing and Knowledge Discovery*, pages 409–418. Springer Verlag, 2000.

[5] W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999.

[6] F. Debole and F. Sebastiani. An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and Technology*, 56(6):584–596, 2004.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[8] E.-H. Han and G. Karypis. Proceedings of the 4th european conference on centroid-based document classification: Analysis and experimental results. In *Principles of Data Mining and Knowledge Discovery*, pages 424–431, 2000.

[9] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 282–289. Springer Verlag, 1994.

[10] D. Ittner, D. Lewis, and D. Ahn. Text categorization of low quality images. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 301–315, 1995.

[11] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151. Morgan Kaufmann Publishers, 1997.

[12] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142. Springer Verlag, 1998.

[13] V. Lertnattee and T. Theeramunkong. Effect of term distributions on centroid-based text categorization. *Information Sciences*, 158(1):89–115, 2004.

[14] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[15] A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of ICML-98, 15th International Conference on Machine Learning*, pages 350–358. Morgan Kaufmann Publishers, 1998.

[16] D. Miller and H. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in Neural Information Processing Systems*, volume 9, pages 571–577. MIT Press, 1997.

[17] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[18] G. Salton. *Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[19] H. Schütze, D. Hull, and J. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 229–237, 1995.

[20] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[21] B. Shahshahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes Phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, 1994.

[22] S. Shankar and G. Karypis. Weight adjustment schemes for a centroid based classifier, 2000. Computer Science Technical Report TR00-035, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota.

[23] E. Wiener, J. Pedersen, and A. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, 1995.