Term-Centric Active Learning for Naïve Bayes Document Classification

Sunghwan Sohn*[,1], Donald C. Comeau[2], Won Kim[3] and W. John Wilbur[4]

[1]*Biomedical Statistics and Informatics, Health Sciences Research, Mayo Clinic, Rochester, MN 55905, USA;* [2,3,4]*National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA*

Abstract: In real world document classification, a subset of documents often needs to be chosen for labeling as a training set for a machine learner. Random sampling is generally not the most effective approach for choosing documents to be labeled. Active learning selects useful examples for labeling to improve the efficiency of learning. We consider two factors in order to measure the usefulness of a document for labeling. Such a document should be 1) largely unknown to the current learner 2) influential by being close to many other documents. These factors are stated from a document-centric viewpoint. A similar analysis can be made from a term-centric viewpoint. It is the purpose of this paper to present this term-centric approach to active learning using a naïve Bayes classifier. We study both document-centric and our new term-centric active learning methods. We find good performance of the term-centric methods on numerous data sets with different characteristics. In addition, a genetic algorithm is employed to compare our results with estimated optimal performance at fixed training set size and our results are between 84% and 99% of the estimated optimum.

Keywords: Active learning, genetic algorithm, naïve Bayes classifier, pool adjacent violators algorithm, uncertainty sampling.

1. INTRODUCTION

Machine learning algorithms for document classification require training documents with known class labels. Traditionally, machine learners take as many labeled documents as can be obtained and passively learn from them. However, in many cases where documents require classification, available documents to be labeled are rapidly increasing but the labeling cost by human experts is high. Imagine a situation where one has a large collection of documents and the need to classify them each into one of two classes, but none of the documents are labeled as to their class. In this situation, one could randomly sample documents one after another and ask a human to label them. However, it is possible that as labels are given to documents the information being gained could be used to guide the process of choosing the next documents to be labeled. Active learning is the name given to that class of strategies that attempts to use current knowledge to predict the best choice of unknown documents to label next in an attempt to improve the efficiency of learning. Active learning starts with a certain number of labeled documents, usually small, as the initial training set. It then repeatedly cycles through learning from the training set, predicting the most informative documents to be labeled, and adding newly labeled documents to the training set.

The goal of active learning is to obtain the best possible classification performance from the fewest possible labeled documents. In order to obtain this efficiency, the most useful documents must be chosen for the next round of labeling. In order to measure the usefulness of a document for labeling there are two factors that need to be considered. First, such a document should be one about which we are largely ignorant. If we can already predict the label with high accuracy, the work of labeling will be the same, but we will learn little from the effort to label the document. Thus we need to be able to measure something we term the level of ignorance (LIG) of a document's label. Second, a good candidate document for labeling will be one which has influence in the sense it will improve knowledge of other documents, i.e., we want to label a document which has many other documents close to it. Thus we need to measure something we term the level of influence (LIF) of a document. Much prior work on active learning for document classification focuses on this document-centric view, but here we present the term-centric view. Many machine learning methods instantiate learning as a set of weights for the terms or features, e.g., naïve Bayes, support vector machines, and maximum entropy methods. The varying weights assigned to terms show that terms are not all of the same influence. Also the considerable research on the problem of feature selection attests to the importance of distinguishing the influence of different terms, see e.g., [1-3]. Not only do different terms have different levels of influence, but we will have different levels of ignorance about terms depending on their occurrence within the labeled set of documents at any particular stage in active learning. We implement term based methods by assigning terms a rating that measures usefulness as a combination estimate based on a terms influence and ignorance. We then score documents by summing these values for terms in each document and the document(s) with the highest score(s) are chosen for the next round of labeling. In our study of term-centric active learning methods in document classification, we use the naïve Bayes as a learning method. There exist more sophisticated methods that generally outperform naïve Bayes. However, the naive Bayes classifier continues to be widely used in text classification because of its simplicity and efficiency [4-9]. In spite of the popularity of the naïve Bayes classifier its optimal active learning approach is not

*Address correspondence to this author at the Biomedical Statistics and Informatics, Health Sciences Research, Mayo Clinic, Rochester, MN 55905, USA; E-mail: sohn.sunghwan@mayo.edu

1874-1339/09

known. These facts motivate us to investigate active learning methods for the naïve Bayes classifier. However, the active learning methods we developed may not in general apply to other learning methods because of the simplicity of naïve Bayes.

In our experiments, we use the naïve Bayes Binary Independence Model (BIM) [9]. In the BIM a document is represented by a vector of binary attributes indicating presence or absence of terms in the document (for details see APPENDIX I). We made this choice because the BIM can be trained very rapidly and yet has good performance [10]. The performance of our methods is compared with other known methods—uncertainty sampling [11, 12] and error reduction sampling [13]—as well as the estimate of the best performance obtained using a genetic algorithm (GA). Section 2 reviews related work in active learning. Section 3 describes how we estimate the class probability of unlabeled documents in our active learning methods. Section 4 explains the active learning methods we tested. Section 5 describes the data sets used in this paper. Methods of evaluation are explained in Section 6. Section 7 explains the GA used to estimate optimal performance. Section 8 provides experimental results. Section 9 concludes the paper with discussion.

2. RELATED WORK

A number of approaches have been tried in active learning to select the most informative examples. One of them is uncertainty sampling. Lewis and Gale [12] apply uncertainty sampling to choose the document most uncertain by the current classifier. Such a document is close to the current decision boundary and may be the most informative to update the decision boundary. They applied logistic regression to the likelihood ratio of positive and negative examples to estimate class probability for unlabeled examples and select uncertain examples based on this probability. The same approach but using a decision tree classifier has also been tried [11]. Some people use uncertainty sampling with support vector machines (SVM). Campbell *et al*. [14] introduce active learning with SVM. They select examples decreasing the margin between two classes. Tong and Koller [15] also study SVM and select examples that bisect version space and so reduce classifier uncertainty. Luo *et al.* [16] apply active learning with SVM to multi-class problems by choosing examples with the smallest difference in probabilities between the largest and the runner-up.

A related approach is query by committee. Seung *et al*. [17] create a committee of classifiers and choose the next example based on maximal disagreement. Freund *et al.* [18] provide a theoretical proof for the result of Seung *et al*.'s query by committee. They show that if query by committee yields high information gain, then the prediction error decreases rapidly with the number of labeled examples. Dagan and Engelson [19] also analyze committee-based sampling with probabilistic models and apply it to Hidden Markov Models used for part-of-speech tagging.

Some researchers select examples that reduce the estimated error. Roy and McCallum [13] select examples that directly reduce the estimated error rate instead of reducing version space. They estimate the entropy error of each unlabeled example using its posterior class probability from a naïve Bayes classifier and then select an example with the largest predicted decrease in error rate. They also utilize bagging [20] in order to improve Bayes' posterior probability estimates. A challenge of their approach is computational tractability. They use fast naïve Bayes updates and work with a small portion of the unlabeled documents to solve this problem. Saar-Tsechansky and Provost [21] also choose examples that reduce the class probability estimation error. They apply the bootstrap algorithm [22] to estimate the variance of class probability for unlabeled examples and try to reduce this estimated variance in order to reduce the class probability estimation error.

Baram *et al.* [23] observe that there is no single active learner to consistently outperform others on multiple data sets. Instead of using a single active learner they dynamically combine multiple active learners by the multi-armed bandit algorithm [24]. In the multi-armed bandit problem, a gambler must choose which of K non-identical slot machines to play in a sequence of trials to yield the maximum reward. Baram *et al*. consider an unlabeled training example as a slot machine that an active learner chooses at each step. They calculate the reward based on the entropy change in the unlabeled set with or without adding a given example in the training set. Then, they try to maximize this reward over all steps by updating a selection probability distribution and choosing an example with highest probability in each step. Similar to Baram *et al.* Settles and Craven [25] experiment with a number of active learning methods for conditional random fields and conclude that no one method works best in all cases.

As related to our term-centric approach, we also mention the work of Druck *et al.* [26] where human annotators label features instead of documents to facilitate learning and the most informative features for labeling are sought. They use latent Dirichlet allocation (LDA) to analyze features into topics and choose the features most closely associated with topics to show to the human annotators. However, some of our data sets are quite large and LDA is not efficient to compute for such sets and we have not experimented with the method. Previous work on active learning for document classification focuses on the document-centric view. In this paper we present term-centric active learning methods for naive Bayes document classification.

3. ESTIMATION OF CLASS PROBABILITY

Some of our active learning methods require estimating the class probability of unlabeled documents as accurately as possible. This class probability should be estimated *via* labeled training documents. Although the naïve Bayes classifier produces a class probability it is known that this probability is not accurate. First, the independence assumption is violated and second, in our setting we do not have accurate estimates of the prior class probabilities. We need a more accurate estimation of the class probability.

In this paper, we are interested in binary classification—whether a document belongs to C_1 ($y = 1$) or C_{-1} ($y = -1$). Given a document d with its class y, Bayes' theorem says

$$P(y=1|d)=\frac{P(d|y=1)P(y=1)}{P(d|y=1)P(y=1)+P(d|y=-1)P(y=-1)}. \quad (1)$$

Thus, $P(y=1|d)$ is a monotonically increasing function of

$$score=\ln\left(P(d|y=1)/P(d|y=-1)\right). \quad (2)$$

Equation (2) defines a document *d*'s score in naïve Bayes (see APPENDIX I for more details). This score is expected to order documents in the same way probability would, i.e., we expect that score satisfies

$$score(d_i)\le score(d_j)\Rightarrow P(y=1|d_i)\le P(y=1|d_j). \quad (3)$$

Define the set of numbers $\{\delta_i\}_{i=1}^{N}$

$$\delta_i=\begin{cases}1, & d_i\in C_1\\ 0, & d_i\in C_{-1}\end{cases}. \quad (4)$$

Consider the set of pairs $\{(score(d_i),\delta_i)\}_{i=1}^{N}$. The condition (3) suggests that we apply the Pool Adjacent Violators (PAV) Algorithm [27-29] to this set of pairs to obtain the maximum likelihood non-decreasing probability function $pr(score)$. This probability function of the *score* makes the observed set of data points most probable subject to condition (3). It has the interpretation

$$P(y=1|score(d_i))=pr(score(d_i)). \quad (5)$$

The better the scoring function the more useful the result of applying (5). In fact (3) need not be true to apply the PAV Algorithm and obtain (5), but the closer it is to truth, the better the result.

4. ACTIVE LEARNING METHODS

We have tested multiple active learning methods. The basic routine, which is used commonly in all methods, is described in Fig. (**1**). The details of each method are explained in the next sections. Section 4.1 describes a general concept of how to rank documents for the next labeling request. Section 4.2 explains previously known uncertainty sampling. Section 4.3 summarizes Roy and McCallum's [13] error reduction sampling, and Sections 4.4 to 4.8 explain new methods we tested.

4.1. Ranking for Active Learning

To measure the usefulness of a document for labeling we consider two factors: level of ignorance (*LIG)* and level of influence (*LIF)*. Both *LIG* and *LIF* are important. In fact if we already have complete knowledge of a document (*LIG* = 0), then it is not useful to choose that document for labeling no matter how high its *LIF* is. Likewise if a document has no neighbors and hence no influence (*LIF*=0), then there is no value in learning how to label it no matter how high *LIG* is. Thus each of these factors should have veto power over the other and a reasonable way to combine them is as a product. We can then compute the ranking score for a document as

$$Ranking\ Score_d=LIG_d\cdot LIF_d. \quad (6)$$

The above discussion concerns ranking a document's desirability for labeling. We call this a document-centric view. Since it is only documents that we can select for labeling, the document-centric view is natural. However, one of the claims we put forward in this work is that it is also useful to take a term-centric view. We first consider the value of learning each term. We then select the document with the most valuable terms. For each term *t* we attempt to measure our level of ignorance (LIG_t) about it and its level of influence (LIF_t). Then we compute a rating, r_t, for that term

$$r_t=LIG_t\cdot LIF_t. \quad (7)$$

The ranking score for a document then becomes

$$Ranking\ Score_d=\sum_{t\in d}r_t. \quad (8)$$

Of course one need not restrict an approach to be strictly document-centric or strictly term-centric. Rather it is possible that good results may come from combining, in some manner, ranking scores from (6) and (8). Also, one can use only *LIG* or *LIF* for ranking scores. The first step in investigating the *LIG-LIF* approach to ranking documents for active learning is to have some methods of measuring *LIG* and *LIF*, both document-centric and term-centric. Some possible measures for each case are described.

Document-Centric View

LIG: There are several approaches to measuring the level of ignorance about a document.

1) Uncertainty: This has generally referred to an estimate of a documents classification as a probability that $y=1$, i.e., an estimate of $p(y=1|d)$. Such an estimate can be based on scores produced by the classifier trained on the already labeled documents. For naïve Bayes' this has been done by Lewis and Gale [12] using a logistic regression method and by Roy and McCallum [13] using bagging. For convenience the measure of uncertainty by any of these methods may be taken as $1-2|0.5-p(y=1|d)|$. For the most uncertain document ($p(y=1|d)=0.5$) this value is 1 and the most certain document ($p(y=1|d)=1$) this is 0.

2) Score Uncertainty: An alternative approach to uncertainty based on probability is to use the score produced by the classifier directly. Documents are typically classified based on a threshold value of the

Start with a small number of labeled C_1 and C_{-1} docs
Train an initial classifier with those docs
While willing to label more docs
  Apply the current classifier to unlabeled docs
  Rank unlabeled docs based on a given active learning method
  Select M* top ranked docs
  Label the selected M docs
  Train a new classifier with all labeled docs

*M: number of documents to be labeled for each step.

Fig. (1). Basic routine of active learning methods.

score, thr. The uncertainty can then be measured as $1-\lambda|thr-score|$ for some suitable choice of $\lambda>0$.

3) Query-by-Committee: Randomly sample a set or committee of classifiers from version space [30]. Then those documents which the committee members are equally divided in predicting either class 1 or class -1 are the most uncertain documents. This method is a form of probability estimate where the fraction of the committee that predicts class 1 becomes an estimate of $p(y=1|d)$ for that document. In this sense it allows an estimate of uncertainty as in 1). Variations on this theme have been studied by a number of investigators [15, 17, 18, 31].

LIF: The level of influence of a document is an estimate of how information about that document will carry over to other closely related documents.

1) Density: A density measure has been used by McCallum and Nigam [31] to estimate how many documents are close to a given document (In this study we used a modified density. The details are in APPENDIX II). They used this measure in conjunction with the naïve Bayes classifier. For SVMs, Boley and Cao [32] use clustering to speed the training and the motivation seems to be similar.

2) Delta Error Rate: If we use the currently labeled set to train the classifier and estimate the class probabilities for all the unlabeled documents, then we can estimate the entropy of a document

$$Ent(d)=-\sum_{y\in\{1,-1\}}p(y|d)\log(p(y|d)) \tag{9}$$

and the expected error

$$Err(d)=1-\max_{y\in\{1,-1\}}p(y|d). \tag{10}$$

By taking averages of these measures over the unlabeled document set we obtain global estimates of the error in classifying the unlabeled documents, say, AEnt and AErr. Now assume that we select unlabeled document d to be labeled and the label y is assigned. Then we can retrain the classifier with the additional information concerning d and recompute the numbers AEnt and AErr to obtain numbers we will denote by $AEnt(y,d)$ and $AErr(y,d)$. Then one can compute the expected global entropy and error associated with labeling d,

$$\begin{aligned}E_{Ent(d)}&=\sum_{y\in\{1,-1\}}p(y|d)AEnt(y,d)\\ E_{Err(d)}&=\sum_{y\in\{1,-1\}}p(y|d)AErr(y,d)\end{aligned}. \tag{11}$$

Roy and McCallum [13] used $E_{Ent(d)}$ and $E_{Err(d)}$ as measures by which to select documents to be labeled in active learning, however, they did not find $E_{Err(d)}$ useful for this purpose. We consider these to be measures of influence, since they measure the effect on the categorization of the unlabeled documents.

Term-Centric View

LIG: For Bayesian weighting, useful knowledge about a term is contained in the number of times it occurs in documents of class 1 and class -1.

1) Label Frequency: The number of documents containing the term that have already been labeled. Clearly the higher this number, the more we know about the term.

2) Weight Gradient: If we should see one more document containing the term t with a label y the weight wt of the term would change to $wt(y)$. Thus the absolute change in the weight would be

$$\Delta wt(y)=|wt-wt(y)|. \tag{12}$$

This is a measure of how much we know about the term t because if we have already labeled many documents that contain the term t, $\Delta wt(y)$ is typically small, but if we have only seen a few such documents it may be large. This measure has an advantage over 1) in that the ratio of the number of documents that we have seen with label 1 and with label -1 will influence the weight. When we apply this measure to a document d the result is

$$\Delta wt(d)=\sum_{y\in\{1,-1\}}p(y|d)\Delta wt(y). \tag{13}$$

3) Weight Uncertainty: This measure is the average uncertainty of the documents in the training set that contain the term t. We rate uncertainty as the entropy so the measure is

$$uwt=\sum_{d:t\in d}Ent(d)/f_t, \tag{14}$$

where f_t is the frequency of t in the whole training set and the sum is also over this set (for labeled set $Ent(d)=0$). Since the maximum value of $Ent(d)$ is log(2) and the minimum 0, uwt varies between 0 and log(2) and provides a measure of our knowledge of how t is distributed over the class 1 and class -1 documents.

LIF: The influence of a term is intimately connected with how widely it is distributed.

1) Term Frequency: This is the number of documents in the training set that contain the term, which we denote by f_t.

2) Uncertain Frequency: This is the number of documents in the unlabeled training set containing the term and satisfying some uncertainty condition, which we denote by f_t^u. Such a condition can be stated as among the N most uncertain documents or among those documents whose uncertainty measure is above some threshold. Presumably it is in the uncertain documents where the influence of a term is important.

3) Entropic Frequency: This is a smoothed version of 2). Instead of either counting or not counting an occurrence of a term in a document based on an uncertainty criterion applied to the document, we count the occurrence with a weight of $Ent(d)$. Thus we have

$$uft = \sum_{d:t \in d} Ent(d). \quad (15)$$

In our active learning methods, we rank unlabeled documents based on *LIG-LIF* concepts and select the top ranked document(s) for the next labeling request. The details of using the *LIG-LIF* measures in active learning are explained under each method.

4.2. Score Uncertainty (SU)

Most uncertainty methods utilize some techniques to convert classifier's scores into class probabilities such as logistic regression [12] and bagging [13]. However, if we are only concerned with binary classification we can directly determine uncertain documents without using this extra process of probability estimation. These are documents near the threshold for deciding whether documents are in C_1 or C_{-1}. Instead of estimating class probability we simply choose uncertain documents directly using classifier scores. This method uses Score Uncertainty, an LIG_d. A document d's ranking score is computed by using only LIG_d in (6).

For binary classification, the most uncertain document is the one with $P(y=1|d)=0.5$. From (1) this can be obtained by

$$\frac{P(y=-1)}{P(y=1)} \exp\left(-\ln\left(\frac{P(d|y=1)}{P(d|y=-1)}\right)\right)=1. \quad (16)$$

Rewriting (16) with the *score* of a document from (2) gives

$$thr = \ln\left(\frac{P(y=-1)}{P(y=1)}\right). \quad (17)$$

Thus, the most uncertain documents are those with their *score* (2) closest to the threshold thr. Here $P(y=1)$ and $P(y=-1)$ are obtained from the distribution of the training set in each active learning cycle. It is true that we do not have very accurate estimates of these prior probabilities, especially at the beginning of active learning when the training set is small. But any bias in these estimates tends to be self correcting and we actually find quite good performance for this method.

4.3. Error Reduction (ER)

Roy and McCallum's [13] error reduction method is used. We use 0/1 error ($E_{Err(d)}$ in (11)), because entropy error ($E_{Ent(d)}$ in (11)), which is used in Roy and McCallum's paper, did not perform as well for us. Also, we used PAV to estimate the class probability of unlabeled documents. Roy and McCallum used Bayes' posterior probability with bagging. We choose the document with the greatest expected decrease in the error rate, $E_{Err(d)}$ in (11). Thus, this method uses Delta Error Rate, an LIF_d. A document d's ranking score is computed by using only LIF_d in (6).

4.4. Term Uncertainty (TU)

Traditional uncertainty sampling simply estimates document uncertainty based on scores and can be considered an LIG_d. Instead the term uncertainty method considers both LIG_t and LIF_t. This method focuses on individual terms in a document. We obtain a rating for each term and then calculate each document ranking score by summing rating values of all terms occurring in that document.

Term t's rating is defined by

$$r_t = \begin{cases} uft, & t \text{ occurs in less than 2 labeled documents} \\ uwt, & \text{otherwise} \end{cases} \quad (18)$$

where uft is given by (15) and uwt is given by (14) in which the entropy is obtained using the PAV probability. This rating favors terms that are relatively unknown to a learner because uft is larger than uwt. Then, a document d's ranking score is given by (8).

4.5. Term Gradient-Frequency (TGF)

In the Bayes classifier, learning is the result of updating term weights during the training process. One possible way to measure the value of a useful term is the Weight Gradient, $\Delta wt(d)$, an LIG_t. Here we add Uncertain Frequency, f_t^u, an LIF_t, into the calculation. Uncertain frequency is a term frequency but only counted in the uncertain document set, A. Then we have

$$r_t = \Delta wt(d) \cdot f_t^u. \quad (19)$$

The uncertain document set, A, is the 1% of the documents most uncertain in the unlabeled set using the score uncertainty method. Equation (8) is applied for ranking the documents. This method is related to the expected gradient length method proposed by Settles and Craven [25] for active learning, but our approach evaluates the effect of adding a single labeled document on the trained weights while in their setting they can only evaluate the effect on the gradient before retraining begins. This may explain the more positive outcome in our setting.

4.6. Density using Score Uncertainty (DS)

McCallum and Nigam [31] defined a density measure (section 4.1) and employed it in active learning. In APPENDIX II we show how their density measure may be transformed into an LIF_t. In order to obtain the best performance we combine this with a score based uncertainty measure, an LIG_d. First, we find the uncertain document set, A, that is the N most uncertain documents based on the score uncertainty. Here, N is 10 times the sampling size in active learning. Then based on A we define

$$u(d) = \begin{cases} 1, & \text{if } d \in \mathsf{A} \\ 0, & \text{otherwise} \end{cases}. \quad (20)$$

This allows us to set

$$r_t(d) = u(d)\log\left(\frac{p(t \mid d) + p(t)}{p(t)}\right) f_t^*. \quad (21)$$

Here elements of (21) are from (35) and (36) contained in APPENDIX II. Then (8) modified for document dependent term weights ranks the documents.

4.7. Selected Term Frequency using Score Uncertainty (STFS)

This method is related to Section 4.6. This method ignores terms that already appear many times in the labeled documents (because if a term occurs many times—i.e., Labeled Frequency is high—its LIG_t should become low) and emphasizes low frequency terms when selecting labeled documents. Thus we define

$$v_t = \begin{cases} 1, & t \text{ appears in } \leq 100 \text{ labeled documents} \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

and set

$$r_t(d) = u(d) v_t f_t . \quad (23)$$

Here $u(d)$ is defined just as in 4.6 and f_t is the term frequency, an LIF_t, in the whole training set. Equation (8) modified for document dependent term weights ranks the documents.

4.8. Term Number (TN)

The foregoing term-based methods obtain document scores by summing the appropriate term ratings. This tends to prefer longer documents. Is this the only reason for their success? To answer this question, we tried selecting simply the longest documents—documents that have the highest number of unique terms. This naive approach performed very poorly, in some cases worse than random. In order to make it competitive we restricted selection to uncertain documents as in the previous methods. Thus we write

$$r_t(d) = u(d) . \quad (24)$$

Here $u(d)$ is defined just as in Section 4.6. Equation (8) modified for document dependent term weights ranks the documents.

5. DATA SETS

We used five sets of natural language documents: REBASE, Newsgroups, MED[heart], MDR and Reuters. From these we set up 24 binary classification tasks. One task came from REBASE, two from two pairs of Newsgroups, eight from MED[heart], three from MDR, and ten from Reuters.

All the natural language texts were preprocessed in the following manner. Stop words and punctuation are removed, but no stemming is performed. We used single words and two word phrases from title and body. (We used only single words in Reuters.)

5.1. REBASE

REBASE is a database of 3,048 documents from the research literature on restriction enzymes. These documents comprise titles, abstracts, and medical subject headings (MeSH®) [29, 33] and are all contained in MEDLINE®. We have extracted 100,000 documents from MEDLINE that lie outside of REBASE but are most likely to be confused with REBASE documents. The classification task is to distinguish between REBASE and this similar portion of MEDLINE. We randomly sampled two thirds of both REBASE and non-REBASE for a training set with the remaining one third in each case held out for testing.

5.2. Newsgroups

We used the same classification tasks as Roy and McCallum [13]. Four data sets were selected from the 20 Newsgroups data set, which is a collection of 20,000 newsgroup documents, partitioned evenly across 20 UseNet discussion groups. We performed two binary classification tasks. One is comp.graphics vs. comp.windows.x and the other is comp.sys.ibm.pc.hardware vs. comp.os.ms-windows.misc. Each pair has 2000 documents—1000 documents from each Newsgroup. In each pair we randomly sampled half of the documents from each Newsgroup for training and reserved the others for testing.

5.3. MED[Heart]

The Medical Subject Headings (MeSH) are a controlled vocabulary produced by the National Library of Medicine and used for indexing, cataloging, and searching for biomedical and health-related information and documents. Each MEDLINE reference is assigned a number of relevant MeSH terms. Our classification task was to predict which documents were assigned a particular MeSH term (class C_1) and which documents were not assigned that term (class C_{-1}). The query "Heart[MeSH]" had 286,225 hits in MEDLINE when we created this data set. Denoting those documents as MED, we looked at eight different MeSH terms that occurred in MED documents: "Human[MeSH]", "Animals[MeSH]", "Myocardium[MeSH]", "Female[MeSH]", "Dogs[MeSH]", "Myocardial Contraction[MeSH]", "Thrombosis[MeSH]" and "Bundle of His[MeSH]". These terms are selected based upon the frequency of the MeSH terms in the MEDLINE database. Two MeSH terms are selected for roughly each of 50%, 30%, 10% and 1% occurrences in MEDLINE. For each MeSH term *x*, we perform binary classification distinguishing between documents including term x (MED_x) and documents not including term x ($\text{MED}_{\bar{x}}$). The MeSH terms are only used to label documents. In the actual training and test process MeSH terms are not used as features. We randomly sampled two thirds of both MED_x and $\text{MED}_{\bar{x}}$ for a training set and reserved the remaining one third for a test set.

5.4. MDR

This dataset contains information from the Center for Devices and Radiological Health (CDRH) device experience reports on devices that may have malfunctioned, caused a death, or serious injury (www.fda.gov/cdrh/mdrfile.html). These reports were received under both the mandatory Medical Device Reporting Program (MDR) from 1984 -

1996, and the voluntary reports up to June 1993. The dataset contains 620,119 reports with three classes—Malfunction, Death, and Serious Injury. We use the field, "event description" as documents. For binary classification we set up three classification tasks. We try to distinguish each class against the other two. Two thirds of each class is randomly selected for the training set and the remaining one third is used for testing.

5.5. Reuters

This is the Reuters-21578 data set consisting of Reuters newswire articles with 135 overlapping topic labels (www.daviddlewis.com/resources/testcollections/reuters21578/). It is split into 9,603 documents for a training set and 3,299 documents for a test set by "ModApte" split. The ten most popular topics were selected: "earn", "acq", "money", "fx," "crude", "grain", "trade", "interest", "wheat", and "ship". We performed binary classification for each topic.

6. METHODS OF EVALUATION

We have tested previously known methods (uncertainty and error reduction) as well as our term-centric methods described in Section 4. Random sampling was also examined.

For each classification problem a portion of the data was set aside as a test set, while the remaining data was used as a training set. This division was kept fixed and the same for all methods. For typical machine learning evaluations test and training sets are varied randomly and the results for the different splits averaged. Because we are investigating active learning methods all based on the same machine learning method (naïve Bayes), we instead randomly vary the starting seed documents repeatedly and average the results for the different random starting seed documents. For each problem ten different random seeds sets were used and the same random seed sets were used as starting sets for all active learning methods.

For each active learning trial, we generated an initial classifier based on the small number of randomly selected documents—five C_1 and five C_{-1} documents for REBASE, MeSH, and MDR sets, and three C_1 and three C_{-1} documents for Newsgroups and Reuters sets. As mentioned above, all methods of active learning were given the same starting sets. In large data sets the sampling size for next labeling requests was adjusted to reduce the number of times the learner had to be trained. For the REBASE data set, initially 10 documents were selected and labeled for training at each step of active learning until there were 5,000 labeled documents. Then 100 documents were selected and labeled at each step as the labeled set grew from 5,000 to 30,000 documents. And if needed, 1,000 documents were selected and labeled at each step as the labeled set grew beyond 30,000 documents. For the MED[heart] and MDR data sets initially 10 documents were selected at each step until 10,000 documents were labeled. Then, 100 documents were selected at each step while the number of labeled documents was between 10,000 and 40,000 and 1,000 documents were selected at each step after there were more than 40,000 labeled documents. The other data sets used the fixed sampling size of 1.

In order to carry out our investigations of active learning it was necessary to employ a measure of classifier performance. In particular we needed to measure the performance based on full training (using the whole training set) and to compare this with performance from training on the generally much smaller sets labeled in the active learning process. For measurement of the classification performance we used average precision [34]. To calculate average precision, documents are ordered by score, precisions are calculated for each point where a C_1 document is found and then these precision values are averaged.

Active learning is the idea that the classifier can select informative documents for training and possibly reach a high performance using as few labeled documents as possible. As a result it is natural to rate active learning methods by the number of documents that must be labeled to achieve a given level of class prediction on held out test data. To measure the performance of active learning methods three different evaluations were used: 1) Performance-Level Counts (PN): counts the smallest number of labeled documents used by active learning methods which produce an average precision which is 85% (PN85), 90% (PN90), or 95% (PN95) of the average precision obtained by training on the "full training set." All counts are averages of ten trials. Each trial starts with different initially labeled training documents which are selected randomly. 2) Winner Votes (WV): counts how many times each method wins, i.e., how many times each method has the smallest average count over each of the three performance levels. 3) Ratio-To-Best (RB): we calculated values normalized by the smallest counts (best method). In each data set and at each performance level and for each of the ten trials, every count is divided by the smallest count among all methods. Then all values for a given method are averaged. The method with a result closest to 1 is the best.

7. COMPARISON WITH GENETIC ALGORITHM

The goal of active learning is to select the most informative examples from a training set and so possibly obtain the maximum performance from that number of labeled training examples. However, it is not known which selected examples are the best for each sampling stage. One could merely compare the performance of each sampling stage with the classifier trained with all labeled training examples. However, it is more valuable to compare this performance with the best possible performance given the same number of examples. The exhaustive search for the best performance is computationally impractical. For this reason, we apply a genetic algorithm (GA) [35] to estimate maximum performance on a given number of documents. Note that GA is not used as an alternative of active learning methods. It simply estimates the best performance that could be achieved by a certain number of labeled training documents.

GA is a well known optimization method based on natural selection. It uses the concept of solution pieces encoded by *genes* and a set of genes, an *individual*, identifying a possible solution state. Natural selection takes place in that these characteristics are selected through the "survival of the fittest" criterion. A set (*population*) of candidate solution states

(*individuals*) is generated. A fitness function is used to evaluate each individual in the population. The individuals encoding better solutions have a better chance of being selected for genetic operators such as *crossover* and *mutation*. These operators create a new population with better individuals than the previous generation. This process is repeated until an acceptable solution is found within specific constraints.

In our GA process, a gene represents a document and an individual represents a set of labeled documents from the training set. The fitness value of an individual is the average precision on the test set of the naïve Bayes classifier trained with that individual (set of documents). We use custom genetic operators in order to make the process suitable to our problem. At initialization, each gene in a given individual is unique—no duplicated documents are allowed in an individual. For crossover, we first set aside overlapped genes and then perform an ordinary crossover. Then, the overlapped part is attached to each child. This avoids duplicated genes. In mutation, a gene could be replaced by any gene not currently in that individual. We bias selection of a mutated value (document) by the degree of the uncertainty of the selected document relative to the rest of the documents. The more uncertain the document, the more likely it will be selected as the mutated gene. The uncertainty of a gene (document) is based on the PAV probability with the naïve Bayes classifier trained with the best individual (set of documents) of the previous generation.

After GA converges, we obtain the best individual (set of documents) and the corresponding fitness value (average precision). We performed 10 trials of GA and chose the best as the optimal performance. In a few cases, we use this result as a reference for our active learning achievement.

8. RESULTS

We have studied seven active learning methods. The pre-existing methods we tested are: Score Uncertainty (SU), and Error Reduction (ER). The new methods we investigated are: Term Uncertainty (TU), Term Gradient-Frequency (TGF), Density using Score Uncertainty (DS), Selected Term Frequency using Score Uncertainty (STFS), and Term Number (TN). Random (RAN) sampling was also used to compare the performance. ER was not used in MED[heart] and MDR sets because of impractical computational time.

Tables **1-5** show three performance evaluation measures (PN, RB, and WV) for each data set. In the Data column, the first number denotes the average precision when using all training documents, the second number denotes percentage of C_1 documents, and the number in the last row denotes the number of the total training documents. A bold number denotes the smallest Performance-Level Counts among the different methods tested. In some case Winner Votes has more than one vote in a given performance level because of a tie. For the MED[heart] and Reuters data sets, Performance-Level Counts are shown for only four tasks, but Ratio-To-Best and Winner Votes reflect all tasks. Performance-Level Counts have three levels: PN85, PN90, and PN95. For example, PN95 means that active learning obtained 95% of the average precision obtained by training on the full training set (e.g., in Table **1** (REBASE) 0.8144 is the average precision using full training set. Then, 95% level of this will be 0.7737 (=0.8114*0.95)).

In most cases, active learning methods required fewer documents than RAN. The difference between RAN and the best method becomes larger as the relative frequency of C_1 documents becomes smaller. RAN inevitably selects fewer C_1 documents in the data sets with low frequency of C_1 documents than the active learning methods. For example, "death" (2% C_1 documents) required 130,910 documents for RAN but 115 documents for TN at PN95 (Table **4**), and "bundle of his" (1% C_1 documents) required 17,907 documents for RAN but 116 documents for TGF at PN95 (Table **3**). However, the performance gain was not that large in the data sets with high frequency of C_1 documents. For example "human" (55.6% C_1 documents) required 128 documents for RAN and 95 documents for TU at PN95 (Table **3**) and surprisingly "injury" (56% C_1 documents) required 314 documents for RAN at PN95, which was the best (Table **4**).

For Winner Votes we counted the best method (the method with the smallest Performance-Level Counts) for each of the three performance levels. However Winner Votes simply recognizes the best but ignores the others and so provides no prediction of the expected amount of work in applying a method to an arbitrary data set. For this reason, we also used Ratio-To-Best, which is normalized by the best method, in order to compare each method with the best method for each classification task. Ratio-To-Best is the average over each task and performance level in a given data set. For example, in Table **2** (Newsgroups) for each method we calcu-

Table 1. Active Learning Performance Evaluation (REBASE)

Data	Eval.	RAN	SU	ER	TGF	TU	DS	STFS	TN
Rebase 0.8144 3%	PN_{85}	2973	448	427	220	**173**	211	177	407
	PN_{90}	5351	800	736	322	**305**	325	344	716
	PN_{95}	12720	1781	1372	683	**652**	837	881	1697
68664	RB	20.61	3.06	2.67	1.36	1.13	1.37	1.32	2.80
	WV					3			

In the data column, the first value is the average precision using the full training data, the second value is percentage of C_1 documents, and the value in the last row is the number of training examples. The evaluations are performance-level counts (PN_{level}), Ratio-to-Best (RB), and Winner Votes (WV). The other column labels are active learning methods.

Table 2. Active Learning Performance Evaluation (Newsgroups)

Data	Eval.	RAN	SU	ER	TGF	TU	DS	STFS	TN
News 1[a] 0.9679 50%	PN_{85}	101.7	59.9	96.7	24.1	**23.1**	53.0	45.7	77.4
	PN_{90}	157.2	90.3	158.1	45.6	**37.9**	91.6	72.7	115.8
	PN_{95}	265.1	207.7	276.4	**86.6**	90.8	186.2	137.7	227.4
News 2[b] 0.9389 50%	PN_{85}	65.0	59.9	68.6	52.8	**34.9**	41.5	47.3	72.4
	PN_{90}	125.1	127.1	116.2	92.4	**64.1**	96.1	88.1	134.8
	PN_{95}	275.1	295.7	247.7	202.8	**199.5**	257.4	220.4	333.8
1000	RB	3.16	2.41	3.13	1.41	1.16	2.09	1.79	2.88
	WV				1	5			

[a]comp.graphics vs. comp.windows.x, [b]comp.sys.ibm.pc.hardware vs. comp.os.ms-window.misc.

Table 3. Active Learning Performance Evaluation (MED[heart])

Data	Eval.	RAN	SU	TGF	TU	DS	STFS	TN
human 0.9263 55.6%	PN_{85}	43	40	42	**25**	58	53	55
	PN_{90}	59	56	53	**42**	71	66	76
	PN_{95}	128	114	144	**95**	130	118	123
myocardium 0.7966 36.5%	PN_{85}	182	136	168	**89**	159	172	141
	PN_{90}	382	273	310	**173**	413	423	282
	PN_{95}	1380	826	1135	**406**	3142	1819	754
dogs 0.7027 10%	PN_{85}	3619	113	167	250	**109**	120	134
	PN_{90}	6834	160	211	353	**139**	145	166
	PN_{95}	22740	225	379	578	**191**	205	216
bundle of his 0.2144 1%	PN_{85}	5705	127	**105**	121	128	143	121
	PN_{90}	9788	134	**111**	129	142	160	133
	PN_{95}	17907	177	**116**	145	175	184	147
190816	RB	71.66	2.46	1.66	1.56	3.55	2.44	2.35
	WV		1	3	15	3		2

Table 4. Active Learning Performance Evaluation (MDR)

Data	Eval.	RAN	SU	TGF	TU	DS	STFS	TN
injury 0.9728 56%	PN_{85}	54	85	109	**42**	108	137	151
	PN_{90}	**99**	210	584	109	195	276	269
	PN_{95}	**314**	1441	2811	395	1223	2756	918
malfunction 0.9401 42%	PN_{85}	**152**	325	733	192	250	313	260
	PN_{90}	**367**	936	1571	403	953	1536	635
	PN_{95}	1529	9300	7813	**1117**	13646	11122	9847
death 0.4070 2%	PN_{85}	18864	137	322	457	164	204	**107**
	PN_{90}	44140	155	352	536	180	216	**111**
	PN_{95}	130910	159	364	622	192	230	**115**
413412	RB	228.39	3.28	5.54	2.94	3.66	4.72	3.08
	WV	4			2			3

Table 5. Active Learning Performance Evaluation (Reuters)

Data	Eval.	RAN	SU	ER	TGF	TU	DS	STFS	TN
acq 0.9859 17.2%	PN_{85}	41.1	8.9	28.4	8.7	11.7	**7.3**	8.8	8.2
	PN_{90}	74.1	9.9	33.0	9.9	16.8	**9.0**	12.2	**9.0**
	PN_{95}	137.0	11.0	47.2	16.7	26.5	10.6	15.3	**10.3**
crude 0.6757 4.1%	PN_{85}	2103.2	16.1	132.4	14.7	17.3	24.6	15.2	**13.5**
	PN_{90}	2481	17.3	147.8	16.4	18.2	25.4	16.1	**13.9**
	PN_{95}	3339.1	20.1	180.2	19.5	22.2	28.4	20.7	**16.1**
interest 0.457 3.6%	PN_{85}	1019.4	**15.9**	111.0	17.3	32.2	28.5	22.6	17.8
	PN_{90}	1847.7	**20.4**	130.9	21.0	36.0	29.5	23.5	21.1
	PN_{95}	2817.3	**24.8**	148.1	31.6	52.7	30.5	28.0	25.3
ship 0.7864 2.1%	PN_{85}	2669.2	28.4	303.1	27.9	**20.8**	40.0	34.6	30.1
	PN_{90}	3370.7	31.8	322.1	41.5	**26.5**	54.5	41.0	34.0
	PN_{95}	4680.0	36.8	347.1	45.0	**33.9**	59.4	48.4	37.2
9603	RB	146.38	1.45	9.96	1.75	2.02	2.04	1.81	1.52
	WV		14			4	6		7

late Ratio-To-Best for each of six performance levels (three for both News 1 and News 2) and then average them. Winner Votes and Ratio-To-Best do not present the same picture for performance. For example, in Table **4** TU is the best based on the Ratio-To-Best measure but RAN has greater Winner Votes. We prefer Ratio-To-Best over Winner Votes as an overall performance measure.

Figs. (**2**) and (**3**) compare the active learning performance with the estimated best performance (GA results). We chose the best out of 10 GA trials as GA results. In active learning methods, all results are the average of 10 trials. Fig. (**2**) shows the performance on the Newsgroups data set comp.graphics vs. comp.windows.x. The previously proposed methods (SU & ER), RAN, and the best three methods (TU, TGF, & STFS) on this data are included. A GA was performed on 50, 100, and 200 documents to estimate the maximum average precision at those numbers of documents. Average precisions from the best GA on 50, 100, and 200 documents were 0.933, 0.950, and 0.956 respectively. The best method in this data set, TU, produced 0.880, 0.921, and 0.943 respectively. The difference between the best and GA was relatively large with only 50 documents but became much smaller for 100 and 200 documents. Fig. (**3**) shows the performance on the REBASE data. The previously proposed methods (SU & ER), RAN, and the best three methods (TU, STFS, & TGF) on this data are included. In this case, GA was performed on 100, 300, and 1000 documents to estimate the maximum average precision. The best average precisions from GA on 100, 300, and 1000 documents were 0.702, 0.764, and 0.800 respectively. The best method in this data set, TU, produced 0.592, 0.733, and 0.782 respectively. Like Fig. (**2**) the difference between the best and GA became smaller when using more documents. After a sufficient number of documents are labeled, TU produces very good performance compared with GA.

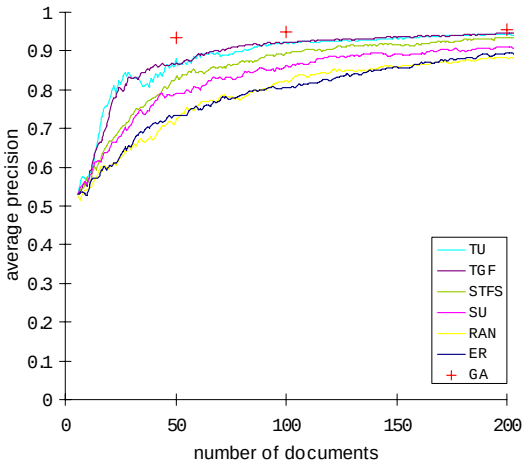

Fig. (2). Average precision of test set for comp.graphics vs. comp.windows.x. Note that the performance order seen in the order of the graphs is reflected by the order in the legend.

Compared to active learning, GA is a radically different approach to obtaining the best N documents for representing a data set. While GA offers no guarantees as to the quality of the optimum obtained, it often gives good solutions to difficult problems [36]. The fact that a genetic algorithm solution is only a modest improvement over the best active learning methods suggests these active learning methods are performing quite well.

9. DISCUSSION

We have introduced a number of term-centric active learning methods and examined their performance. Some methods were closely related but they were used to show how the different measures performed in different data sets. We also examined the performance of the uncertainty

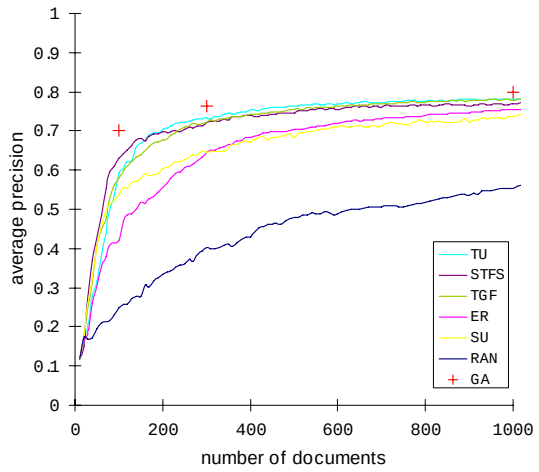

Fig. (3). Average precision of test set for REBASE. Note that the performance order seen in the order of the graphs is reflected by the order in the legend.

method [11, 12] and the error reduction methods introduced by Roy and McCallum [13]. The motivation for all of these methods can be understood based on the *LIG-LIF* concepts. Some methods use only *LIG* or *LIF* and some use both of them. Some are term-centric and some are document-centric and some combine the two approaches. Table **6** shows which *LIG* and *LIF* are used in each method. In our term-centric active learning we are trying to combine *LIG* and *LIF* measures to obtain improved performance. The term-centric methods using both *LIG* and *LIF* performed well in many cases.

Table 6. LIG and LIF used in Active Learning Methods

Method	LIG_d	LIF_d	LIG_t	LIF_t
SU	√			
TGF			√	√
TU			√	√
DS	√			√
STFS			√	√
TN	√			

Tables **1-5** show active learning performance for the methods we tested. As can be seen there was no one best method for all data sets even though we can obtain the overall best by averaging. Baram *et al.* [23] also observed that there was no single active learning method to consistently outperform others on multiple data sets. They noted that some data favored particular methods. In this experiment we performed 24 classification tasks from five different data sets. It is clear that different data sets have different characteristic and that different measures (*LIG & LIF)* differentially achieve best performance. However most of the best performances in active learning came from our term-centric methods.

Table **7** shows the performance of averaged ratio-to-best over all data sets. TU was the best and TGF was the second best. Both are term-centric methods. We further examined which active learning methods might be most effective on different data sets. This relates to the balance between the classes C_1 and C_{-1}. Call a problem balanced if at least 30% of the documents fall into the C_1 class (note: never more than 56% fall into C_1). Otherwise call the problem unbalanced. Among the 24 problems there are 15 in the unbalanced category and 9 in the balanced group. We computed performance as averaged ratio-to-best over the two classes separately for the different methods. The results are contained in Table **8**. For the unbalanced problems TGF gave the best results, TN was second best, and SU third best. We performed significance testing with the Bootstrap Shift Test [37] with a significance level 0.05. The result showed that TGF and TN are not significantly different, but are both significantly better than SU. In the balanced group TU was best and was significantly better than the second best, RAN.

Table 7. Ratio-to-Best Averaged Over all Data Sets

Method	All
RAN	114.55
SU	2.16
TGF	2.15
TU	**1.87**
DS	2.72
STFS	2.36
TN	2.16

Table 8. Ratio-to-Best Averaged Over Unbalanced (C_1 < 30%) and Balanced ($C_1 \geq$ 30%) Data Sets

Method	Unbalanced	Balanced
RAN	181.93	2.24
SU	2.04	2.36
TGF	**1.85**	2.65
TU	2.31	**1.15**
DS	2.56	2.99
STFS	2.06	2.86
TN	1.94	2.53

The PAV algorithm was used to estimate the class probability of unlabeled documents both for the previously known methods we tested as well as the new methods we introduced. The PAV algorithm is a technique for converting a raw score into a probability. It is especially appropriate when an increase in score generally means an increase in the probability (in this case the probability of class $y = 1$). The advantages of the PAV algorithm are its simplicity (no parameters) and its speed (time complexity of order $n\log(n)$ for n data points). One of the consequences of using the PAV algorithm is that our results may differ from the results obtained from other methods of estimating class probability. In our implementation the error reduction approach [13] did

not perform well. In seeking to understand this we note first that the error reduction approach consists of two methods which Roy and McCallum call log loss (entropy based) and 0/1 loss (error based). They found good results using log loss, but results no better than random using 0/1 loss. They had no explanation for this discrepancy. With our approach at estimating class probabilities with PAV we had the opposite experience in that log loss gave worse results than 0/1 loss. However, with our approach neither method worked well. Our hypothesis is that a greedy approach to error reduction can easily get trapped in a local optimum. We believe this may explain our poor results with this approach. Because of the difficulty of comparing results across different methods of data preparation and different algorithm implementations, we included the genetic algorithm results. They allow us to conclude that our best methods are close to optimal.

We have examined our methods on a relatively large number of data sets with different characteristics—24 classification tasks from medical areas, computer related newsgroups, Reuters, and device reports. Our term-centric active learning methods showed best performance in many cases. In both balanced and unbalanced and overall problem types, the best method was term-centric. However, significant work remains to be done in understanding the characteristics of different data sets and how these influence active learning. Theoretical work has produced interesting results on active learning, but there remains a large gap between theoretical limits and what is observed in practice [38]. Our study is largely empirical but based on good justification in terms of a general approach using the *LIG-LIF* concepts. It opens up some new directions for thought and investigations in an area which is not currently well understood.

APPENDIX I. DETAILS OF THE NAÏVE BAYES BINARY INDEPENDENCE MODEL (BIM)

In the BIM, a document is a binary vector over the space of terms. Given a vocabulary T, there is a term or (an attribute) t_k, $t_k \in T$ corresponding to each dimension of the vector space. A document d may be written in vector form $\vec{x}_d = \{x_{dk}\}_{k=1}^{|T|}$ where the value x_{dk} for the document d is either 1 or 0, indicating whether the term t_k occurs in the document or not. With such a document representation, we make the naïve Bayes assumption that the probability of each term occurring in a document is independent of the occurrence of other terms in the same document. Then, the probability of a document given its class, required in (2), is simply the product of the probability of the attribute values over all terms in vocabulary T:

$$P(d|y=1) = \prod_{k=1}^{|T|} \left(x_{dk} P(t_k|y=1) + (1-x_{dk})(1-P(t_k|y=1)) \right)$$
$$P(d|y=-1) = \prod_{k=1}^{|T|} \left(x_{dk} P(t_k|y=-1) + (1-x_{dk})(1-P(t_k|y=-1)) \right). \quad (25)$$

Let N be the total number of documents in the training set. We define several subsets of these N documents: n_k is the number of documents containing term t_k; n_s is the number of C_1 documents (documents with $y=1$); n_{sk} is the number of C_1 documents containing term t_k. Then we may estimate the probability of the term t_k in each class with:

$$p_k = P(t_k|y=1) = \frac{n_{sk}}{n_s}$$
$$q_k = P(t_k|y=-1) = \frac{n_k - n_{sk}}{N - n_s} \quad (26)$$

Then, (2) is given explicitly as the score of document *d*,

$$score = \ln\left(\frac{P(d|y=1)}{P(d|y=-1)} \right) = \sum_{t_k \in d} w_k + C \quad (27)$$

where the term weight w_k and the constant C are given by

$$w_k = \ln\left(\frac{p_k(1-q_k)}{q_k(1-p_k)} \right), \; C = \sum_{t_k \in T} \ln\left(\frac{1-p_k}{1-q_k} \right) \quad (28)$$

In (28) if p_k and q_k are 0 or 1, we cannot properly define the term weight. To avoid this problem we use the following scheme for obtaining p_k and q_k in (26). If n_{sk} is the same value as the minimum of n_s and n_k, then $p_k = 1$ or $q_k = 0$. To avoid this we subtract 1 from n_{sk} if $E(n_{sk})$ is less than $n_{sk} - 1$, otherwise we disregard the term t_k. If $n_{sk} = 0$ or $n_{sk} = n_k + n_s - N$, then $p_k = 0$ or $q_k = 1$. To avoid this we add 1 to n_{sk} if $E(n_{sk})$ is greater than $n_{sk} + 1$, otherwise we disregard the term t_k. Here, $E(n_{sk})$ is the expected value of n_{sk} and equals $n_k n_s / N$.

APPENDIX II. DOCUMENT DENSITY MEASURE

We first describe the density measure for a document d in a database of documents D as proposed by McCallum and Nigam [31]. For any term t let $p(t|d)$ be the fraction of the tokens in d that are t. Likewise let $p(t)$ be the marginal distribution over terms, i.e., $p(t)$ is the fraction of all the tokens in all the documents in D that are t. Following methods proposed in Pereira *et al.* [39], McCallum and Nigam [31] use the Kullback-Leibler (KL) divergence of two distributions to define the distance between individual documents

$$Y(d_i, d_h) = e^{-\beta D(p(t|d_h) \| \lambda p(t|d_i) + (1-\lambda) p(t))} \quad (29)$$

where the KL divergence required here is given by

$$D(p(t|d_h) \| \lambda p(t|d_i) + (1-\lambda) p(t)) = \sum_{t \in V} p(t|d_h) \log\left(\frac{p(t|d_h)}{\lambda p(t|d_i) + (1-\lambda) p(t)} \right) \quad (30)$$

and V denotes the vocabulary of the database. Here in (29) β is a positive constant which McCallum and Nigam take to be 3. Also λ is taken as 0.5 and is used to smooth the $p(t|d_i)$ distribution with $p(t)$ to avoid singularities in the

divergence formula. They go on to define the density at the document d_i by

$$Z(d_i) = e^{\frac{1}{|D|}\sum_{d_h \in D} \ln(Y(d_i, d_h))}. \quad (31)$$

Here we want to show how we modify formula (31) for our application. Since we only use the density to rank documents we begin by observing that the same ranking is produced by the exponent on the right in (31) and that a positive constant factor also has no influence on the ranking. Thus for ranking purposes we may use

$$Z_1(d_i) = \sum_{d_h \in D} \ln(Y(d_i, d_h)). \quad (32)$$

If we substitute from equation (29) into (32) the log and exponential functions cancel each other and the positive constant β may be dropped yielding

$$\begin{aligned} Z_2(d_i) &= -\sum_{d_h \in D} D(p(t \mid d_h) \| \lambda p(t \mid d_i) + (1-\lambda) p(t)) \\ &= -\sum_{d_h \in D} \sum_{t \in V} p(t \mid d_h) \log\left(\frac{p(t \mid d_h)}{\lambda p(t \mid d_i) + (1-\lambda) p(t)} \right) \end{aligned} \quad (33)$$

For efficiency in performing the calculation we observe that $Z_2(d_i)$ can be rewritten as

$$\begin{aligned} Z_2(d_i) = &-\sum_{d_h \in D} \sum_{t \in V} p(t \mid d_h) \log\left(\frac{p(t \mid d_h)}{(1-\lambda) p(t)} \right) \\ &+ \sum_{d_h \in D} \sum_{t \in d_i} p(t \mid d_h) \log\left(\frac{\lambda p(t \mid d_i) + (1-\lambda) p(t)}{(1-\lambda) p(t)} \right) \end{aligned} \quad (34)$$

Finally we see that the first term on the right side in equation (34) is a constant and cannot affect the ranking. Thus we may further simplify to

$$\begin{aligned} Z_3(d_i) &= \sum_{d_h \in D} \sum_{t \in d_i} p(t \mid d_h) \log\left(\frac{p(t \mid d_i) + p(t)}{p(t)} \right) \\ &= \sum_{t \in d_i} \log\left(\frac{p(t \mid d_i) + p(t)}{p(t)} \right) \sum_{d_h \in D} p(t \mid d_h) \\ &= \sum_{t \in d_i} \log\left(\frac{p(t \mid d_i) + p(t)}{p(t)} \right) f_t^*. \end{aligned} \quad (35)$$

We have dropped λ because it equals 0.5 and have introduced the symbol

$$f_t^* = \sum_{d_h \in D} p(t \mid d_h) \quad (36)$$

which may be understood as a modified term frequency in which each occurrence is weighted by the probability of the term in that document. For ranking purposes, $Z_3(d_i)$ must yield the same result as the original density measure $Z(d_i)$ of McCallum and Nigam. Moreover, $Z_3(d_i)$ can be seen as a score coming from a sum of ratings for each term in that document. The term ratings are each a product of a log term which is very closely related to the common inverse document frequency (IDF) weight used in the information retrieval literature [40, 41] and the modified term frequency given by (36). A simple sum of the modified frequencies would be closely related to the sum of frequencies that we use as an LIF_t measure, but the IDF weight works in the opposite direction giving more importance to less frequent terms. Thus $Z_3(d_i)$ is intermediate between our frequency rating of terms and the IDF measure of term influence. As such it makes an interesting comparison with the frequency based LIF_t.

ACKNOWLEDGMENT

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

REFERENCES

[1] A. L. Blum, and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, pp. 245-271, 1997.

[2] I. Guyon, and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

[3] Y. Yang, and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, 1997, pp. 412-420.

[4] S. Eyheramendy, D. Lewis, and D. Madigan, "On the naive bayes model for text categorization," in *Ninth International Workshop on Artificial Intelligence & Statistics* Key West, FL, 2003.

[5] K.-M. Schneider, "A comparison of event models for naive bayes anti-spam e-mail filtering," in *EACL'03*, 2003, pp. 307-314.

[6] K.-M. Schneider, "Techniques for improving the performance of naive Bayes for text classification " in *Computational Linguistics and Intelligent Text Processing, 6th International Conference*, Mexico City, 2005, pp. 682-693.

[7] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparision of a Naive Bayesian and a memory-based approach," in *Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000)*, Lyon, France, 2000, pp. 1-13.

[8] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, "Learning to construct knowledge bases from the World Wide Web," *Artificial Intelligence*, vol. 118, pp. 69-113, 2000.

[9] D. D. Lewis, "Naive (Bayes) at forty: the independence assumption in information retrieval," in *ECML*, 1998, pp. 4-15.

[10] W. J. Wilbur, and W. Kim, "The ineffectiveness of within-document term frequency in text classification," *Information Retrieval*, vol. in press.

[11] D. D. Lewis, and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceeding of the 11th International Conference on Machine Learning*, New Brunswick, USA, pp. 148-156, 1994.

[12] D. D. Lewis, and W. A. Gale, "A sequential algorithm for training text classifiers," in *International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 3-12.

[13] N. Roy and A. McCallum, "Toward Optimal Active Learning through Sampling Estimation of Error Reduction," in *18th International Conference on Machine Learning*, 2001, pp. 441-448.

[14] C. Campbell, N. Cristianini, and A. Smola, "Query learning with large margin classifiers," in *17th International Conf. on Machine Learning*, 2000, pp. 111-118.

[15] S. Tong, and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45-66, 2001.

[16] T. Luo, K. Kramer, D. B. Goldgof, L. O. Hall, S. Samson, A. Remsen, and T. Hopkins, "Active Learning to Recognize Multiple Types of Plankton," *Journal of Machine Learning Research*, vol. 6, pp. 589-613, 2005.

[17] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Fifth Annual ACM Workshop on Computational Learning Theory* 1992, pp. 287-294.

[18] Y. Freund, H. Seung, E. Shamir, and N. Tishby, "Selective sampling using the Query By Committee algorithm," *Machine Learning*, vol. 28, pp. 133-168, 1997.

[19] I. Dagan, and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *ICML*, 1995.

[20] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123-140, 1996.

[21] M. Saar-Tsechansky, and F. Provost, "Active Sampling for Class Probability Estimation and Ranking," *Machine Learning*, vol. 54, pp. 153-178, 2004.

[22] B. Efron, and R. Tibshirani, *An introduction to the Bootstrap*: Chapman and Hall, 1993.

[23] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 255-291, 2004.

[24] P. Auer, N. Cesa-Bianchi, Y. Freund, and E. Schapire, "The nonstochastic multiarmed bandit problem.," *SIAM Journal on Computing*, vol. 32, pp. 48-77, 2002.

[25] B. Settles, and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008, pp. 1069-1078.

[26] G. Druck, G. Mann, and A. McCallum, "Learning from Labeled Features using Generalized Expectation Criteria," in *Proceedings of ACM Special Interest Group on Information Retreival (SIGIR)*, Singapore, 2008, pp. 595-602.

[27] M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman, "An empirical distribution function for sampling with incomplete information," *Annals of Mathematical Statistics*, vol. 26, pp. 641-647, 1954.

[28] W. Hardle, *Smoothing techniques: with implementation in S*. New York: Springer-Verlag, 1991.

[29] W. J. Wilbur, L. Yeganova, and W. Kim, "The Synergy Between PAV and AdaBoost," *Machine Learning*, vol. 61, pp. 71-103, 2005.

[30] T. M. Mitchell, *Machine Learning*. Boston: WCB/McGraw-Hill, 1997.

[31] A. K. McCallum, and K. Nigam, "Employing EM and pool-based active learning for text classification," in *Fifteenth International conference on machine learning*, 1998, pp. 359-367.

[32] D. Boley, and D. Cao, "Training Support Vector Machines Using Adaptive Clustering.," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, USA, 2004, pp. 126-137.

[33] W. J. Wilbur, "Boosting Naive Bayesian Learning on a Large Subset of MEDLINE," in *American Medical Informatics 2000 Annual Symposium*, Los Angeles, CA, 2000, pp. 918-922.

[34] C. D. Manning, and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999.

[35] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.

[36] H. Braun, "On solving travelling salesman problems by genetic algorithms " in *Parallel Problem Solving from Nature*. vol. 496 Berlin / Heidelberg: Springer, 1991, pp. 129-133.

[37] E. W. Noreen, *Computer Intensive Methods for Testing Hypotheses*. New York: John Wiley & Sons, 1989.

[38] G. Stemp-Morlock, "Learning More About Active Learning," *Communications of the ACM*, vol. 52, pp. 11-13, April 2009.

[39] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of English words," in *31st annual meeting on Association for Computational Linguistics*, 1993, pp. 183-190

[40] R. Baeza-Yates, and B. Ribeiro-Neto, *Modern Information Retrieval*. Harlow, England: Addison-Wesley Longman Ltd., 1999.

[41] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes*, 2nd ed. San Francisco: Morgan-Kaufmann Publishers, Inc., 1999.

Received: April 21, 2009 Revised: May 22, 2009 Accepted: May 27, 2009