Active Online Classification via Information Maximization

Noam Slonim
IBM Haifa Research Lab
Haifa, Israel
noams@il.ibm.com

Elad Yom-Tov
IBM Haifa Research Lab
Haifa, Israel
eladyt@yahoo-inc.com*

Koby Crammer
Department of Electrical Engineering
The Technion
Haifa, Israel
koby@ee.technion.ac.il

Abstract

We propose an online classification approach for co-occurrence data which is based on a simple information theoretic principle. We further show how to properly estimate the uncertainty associated with each prediction of our scheme and demonstrate how to exploit these uncertainty estimates. First, in order to abstain highly uncertain predictions. And second, within an active learning framework, in order to preserve classification accuracy while substantially reducing training set size. Our method is highly efficient in terms of run-time and memory footprint requirements. Experimental results in the domain of text classification demonstrate that the classification accuracy of our method is superior or comparable to other state-of-the-art online classification algorithms.

1 Introduction

In the online classification paradigm, a classifier observes instances in a sequential manner. After each observation, the classifier predicts the class label of the observed instance and receives as feedback the correct class label. The online classifier may then update its prediction mechanism, presumably improving the accuracy of future predictions [4]. To motivate our derivation we consider a stream of instances of co-occurrence data, each labeled with one or more labels out of a set of N_l possible labels. Let N_f denote the number of distinct features in our data, and let C denote a co-occurrence matrix with N_f rows and N_l columns, such that $C(i,k)$ indicates the number of occurrences of the i-th feature in all instances associated with the k-th label. If the assigned labels indeed represent distinct classes, one may expect an approximated block structure in C, where each block consists of features representative of a particular class. For example, this expectation underlies most text classification schemes under the standard bag of words model.

A natural route to quantify the statistical signal in C is via the mutual information [3] embodied in a matrix P, which is the normalized form of C, denoted henceforth I_{fl}. This information quantifies the average number of bits revealed over the label identity while observing a particular feature in a given instance. If each label is characterized by a set of distinct features, we expect that C will have an approximated block structure, and correspondingly that I_{fl} will be relatively high. The current work exploits precisely this intuition. Specifically, given a training set of labeled instances, the goal of a classification scheme is to properly label the remaining test set instances. The above discussion implies that if one constructs C based on the *entire data* where the test set instances are properly labeled, a relatively clear structure will emerge, with an associated relatively high I_{fl} value. In contrast, if the test set instances are poorly labeled, the resulting C will have little structure, if any, as the wrong labels over the test set will smear the statistical signal arising from the training set. Correspondingly, a relatively low I_{fl} value will be observed. Thus, our starting point here is to cast supervised classification as labeling the test set instances so to maximize the mutual information, I_{fl}. Importantly, this information should be estimated via a co-occurrence matrix, C, constructed out of the labeled training instances along with the test set instances and their predicted labels. In Fig. 1 we present simulation results over real world data that support this proposed formulation.

Various algorithms can be derived to find labellings that aim to maximize I_{fl}. Here, we focus on a simple online learning strategy [4]. Specifically, we examine the matrix C which is constructed using the instances scanned thus far and their associated labels. Given a new instance, x, the algorithm performs N_l trials, simulating the addition of x to the construction of C where it is labeled with each of the N_l possible labels; the particular label resulting with an updated matrix C with an associated maximal information I_{fl} will be predicted as x's label.

In classical online learning, once a label is predicted the true label is revealed and the prediction mechanism is updated accordingly. This setup is not suitable if the cost of each true label is relatively high. In the active learning paradigm this issue is addressed via selective sampling techniques in which the label is queried only if the relevant prediction is relatively uncertain [11; 14; 2; 12]. Our framework can naturally embody this intuitive principle. Specifically, we examine the loss in I_{fl} due to assigning x with the *second* best label, and demonstrae how this loss estimates prediction uncertainty. We further use this information loss in two prac-

*Present affiliation: Yahoo Research, New York.

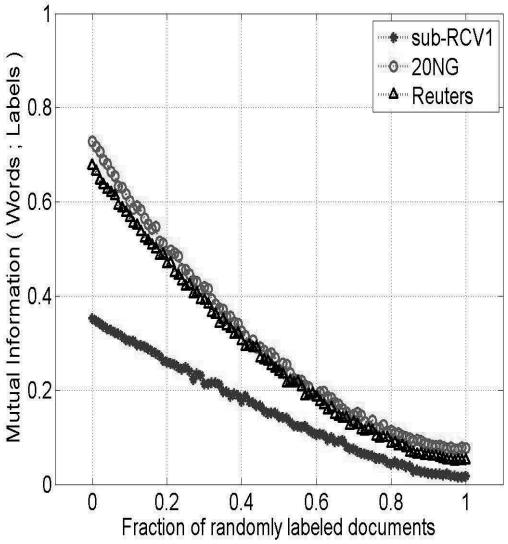

Figure 1: I_{fl} decreases as a function of the fraction of documents that are randomly labeled. The three curves were estimated for three real-world corpora, described in our experimental results.

tically important scenarios. First, in order to abstain highly uncertain predictions so to increase precision at the cost of recall reduction. Second, in asking for instances' labels only if the associated prediction is considered highly uncertain. We demonstrate the validity of our approach for the task of text classification over real world corpora. Comparisons with state-of-the-art online classification schemes [17; 10; 5; 9; 6; 7] suggest that in spite of its simplicity, our proposed method is comparable or superior to these alternative methods.

2 Classification via information maximization

Let F be a discrete random variable with N_f possible values that represent the set of distinct features in our data, $\{f_1....f_{N_f}\}$. Let X be a random variable with N_x possible values that represent the set of N_x instances in our data, $\{x_1...x_{N_x}\}$. Let $n(f_i, x_j)$ be the number of occurrences of the i-th feature in the j-th instance. Then, ignoring the order of the features observed in x_j, as done, e.g., in the standard bag of words model [18], we obtain that the empirical probability of observing f_i in x_j is

$$P(f_i|x_j) = n(f_i, x_j) / \sum_{i'=1}^{N_f} n(f_{i'}, x_j) \ . \tag{1}$$

Further, for brevity we assume a uniform prior, $P(x_j) = 1/N_x$, ending up with an estimation of the joint distribution, $P(f_i, x_j) = P(x_j)P(f_i|x_j)$. Next, we denote by L a random variable with N_l possible values that represent the distinct labels, $\{l_1...l_{N_l}\}$. The assignment of x_j with a label l_k can be represented via $P(l_k|x_j) = 1$ while $P(l_{k'}|x_j) = 0 \ \forall \ k' \neq k$.[1]

[1]Clearly, it is also possible to represent the assignment of x_j with more than one label. Nonetheless, for brevity purposes we limit our derivation to the case where each instance is solely assigned with one label.

Considering the joint distribution over all variables mentioned, we have

$$P(f_i, x_j, l_k) = P(f_i, x_j)P(l_k|x_j) \tag{2}$$

where we used $P(l_k|x_j, f_i) = P(l_k|x_j)$, namely that instance identity, x_j, solely determines its label. Given this formulation we have

$$P(f_i, l_k) = \sum_{j=1}^{N_x} P(f_i, x_j)P(l_k|x_j) = \sum_{j \in l_k} P(f_i, x_j) \ , \tag{3}$$

where $j \in l_k$ denotes $P(l_k|x_j) = 1$. It is further easy to verify that under this formulation

$$\begin{cases} P(l_k) = n(k)/N_x \\ P(f_i|l_k) = (1/P(l_k)) \sum_{j \in l_k} P(f_i|x_j)P(x_j) \end{cases} \tag{4}$$

where $n(k)$ denotes the number of instances assigned with the k-th label, l_k. Given $P(f_i, l_k)$, one can quantify the dependency between the features and the labels via the mutual information [3]

$$I_{fl} \equiv I(F; L) = \sum_{i,k} P(f_i, l_k) \log \frac{P(f_i|l_k)}{P(f_i)} \ . \tag{5}$$

Our motivating underlying assumption is that each label, or class, is characterized by a set of distinct features. Hence, we expect an approximated block-structure in $P(f_i, l_k)$; correspondingly, I_{fl} is expected to be relatively high. Importantly, I_{fl} can be estimated over the training set labeled instances, as well as test set instances for which the label is predicted. Accurate predictions are expected to even sharpen the statistical dependency observed between F and L over the training set, resulting with relatively high I_{fl} values. In contrast, poor predictions are expected to smear the statistical dependency between F and L, resulting with relatively low I_{fl} values. Here, we propose to turn this intuitive understanding over its head, and to predict the labels of the test set instances such that I_{fl} will be maximized.

3 An online learning setup

Various algorithms can be derived to predict labels aiming to maximize I_{fl}. Here, we focus on *online* classification. Under this paradigm, at any given time point, only one instance is examined and its label is predicted. In classical online learning, once a prediction is provided, the true label is revealed, and accordingly the learning model and the prediction loss are updated [4]. In our context, this corresponds to a situation where N_x instances were already scanned, and the label of each was predicted and then revealed. Next, a new instance, $x_{N_x^+}$, is encountered, where for conciseness of notation we used $N_x^+ = N_x + 1$. Let l_k be the tentative label predicted for $x_{N_x^+}$, and further denote $P_{N_x^+} \equiv P(f_i|x_{N_x^+})$. Observing $x_{N_x^+}$ requires to update the probabilistic model and we use the superscript $+$ to distinguish between components of the probabilistic model estimated before and after observing $x_{N_x^+}$. First, if before observing $x_{N_x^+}$ we had $P(x_j) = 1/N_x \ \forall \ j$, after observing $x_{N_x^+}$ we have $P^+(x_j) = 1/N_x^+$. Similarly,

$n^+(k') = n(k') \ \forall \ k' \neq k$, while $n^+(k) = n(k) + 1$. In addition, we have

$$\begin{cases} P^+(l_{k'}) = n^+(k')/N_x^+ = n(k')/N_x^+ \ \ \forall \ k' \neq k \\ P^+(l_k) = n^+(k)/N_x^+ = (n(k)+1)/N_x^+ \ . \end{cases} \tag{6}$$

Finally, using Eq. 4 it is easy to verify that $P^+(f_i|l_{k'}) = P(f_i|l_{k'}) \ \forall \ k' \neq k$; in contrast, after little algebra, for l_k we obtain

$$P^+(f_i|l_k) = \frac{n(k)P(f_i|l_k) + P_{N_x^+}}{n(k)+1} \ , \tag{7}$$

where we used $\sum_{j \in l_k} P(f_i|x_j) = n(k)P(f_i|l_k)$, as can be derived from Eq. 4. In particular, the online update rules in Eq. 6 and Eq. 7 imply that once $x_{N_x^+}$ is assigned with a particular label, updating $P(f,l)$ is straightforward and has linear complexity of $O(N_f)$.

3.1 Online assignment rule to maximize I_{fl}

In a straightforward approach one may simulate assigning $x_{N_x^+}$ with each possible label, find all the associated joint distributions, and eventually assign $x_{N_x^+}$ with the label for which I_{fl} is maximized. This process involves N_l simulation trials, each with a complexity of $O(N_f N_l)$ for estimating I_{fl}. Thus, the overall complexity will be $O(N_l^2 N_f)$. However, a more efficient approach arises from the analogy between the problem at hand and the problem of *unsupervised* clustering via sequential information maximization [21]. Specifically, in that earlier work, the goal is to cluster instances such that the information between the obtained clusters and the features is maximized. If we identify each cluster with a particular label in our setup, we see that the problems are in perfect analogy, except for two important differences. First, in [21] it is assumed that the entire data is available in advance; in contrast, here we assume an online learning setup that in particular requires updating $P(f,l)$ after encountering each new instance. Second, in [21] the focus is on unsupervised clustering, while here we expand this framework for supervised classification where labels are exploited during the classification process. Nonetheless, using the analogy between the two problems, we observe that assigning $x_{N_x^+}$ with a label such that I_{fl} will be maximized is equivalent to assigning in [21] a singleton cluster that consists solely of $x_{N_x^+}$ to one of the existing clusters so to maximize the information between the clusters and the features. Adapting the derivation in [21] to our needs, we conclude that the assignment that locally maximizes I_{fl} is given by

$$l_k = argmin_{l_{k'}} \ \delta(I_{fl}, k'), \tag{8}$$

where

$$\delta(I_{fl}, k') \equiv (P(l_{k'}) + P(x_{N_x^+}))JS(P(f|l_{k'}), P_{N_x^+}) \ , \tag{9}$$

where JS stands for the Jensen-Shannon divergence [16], defined by

$$JS(p_1, p_2) = \pi_1 KL(p_1 \| \bar{p}) + \pi_2 KL(p_2 \| \bar{p}) \ , \tag{10}$$

where in our case $\pi_1 = n(k')/(n(k')+1)$, $\pi_2 = 1/(n(k')+1)$, $p_1 = P(f|l_{k'})$, $p_2 = P_{N_x^+}$, $\bar{p} = \pi_1 p_1 + \pi_2 p_2$, and $KL(p(y)\|q(y)) = \sum_y p(y)\log(p(y)/q(y))$ is the KL divergence [3]. In short, to maximize I_{fl} one should assign $x_{N_x^+}$ with the label l_k for which the conditional distribution $P(f|l_k)$ is most similar to $P_{N_x^+}$ in terms of Eq. 8. Since the complexity of estimating the JS divergence is $O(N_f)$, the overall complexity of predicting the label of a new instance is $O(N_l N_f)$.

3.2 Estimating prediction uncertainty

Properly estimating the uncertainty associated with a prediction is practically useful in various scenarios. In order to address this issue we adopt a simple Best-versus-Second-Best approach (cf. [12]). Specifically, given an incoming instance, $x_{N_x^+}$, we denote by l_{k_1} the label for which Eq. 8 is minimized, namely the predicted label. We further denote by l_{k_2} the second best label, namely the label for which Eq. 8 is minimized over all $k \neq k_1$. Thus, the loss in I_{fl} due to assigning $x_{N_x^+}$ with the *second* best label is given by

$$g(x_{N_x^+}) \equiv \delta(I_{fl}, k_2) - \delta(I_{fl}, k_1) \geq 0 \ . \tag{11}$$

A relatively high $g(x_{N_x^+})$ value implies that l_{k_1} is clearly distinguished from all other labels for $x_{N_x^+}$, namely the prediction is relatively certain. Conversely, low $g(x_{N_x^+})$ implies that at least two labels are hard to distinguish as potential assignments for $x_{N_x^+}$, hence high uncertainty should be associated with the prediction. Finally, instead of considering $g(x_{N_x^+})$ directly, we consider a normalized form, reflecting our intuition that prediction uncertainty should gradually decrease as the prediction mechanism being exposed to more instances. Specifically, we define the uncertainty associated with the label predicted for $x_{N_x^+}$ via

$$u(x_{N_x^+}) \equiv g(x_{N_x^+})^{-1}/(N_x^+)^2 \ . \tag{12}$$

We found this simple definition to work well in practice. Nonetheless, other definitions could certainly be exploited.

3.3 Algorithms

We define three algorithms that rely on the derivation above. The first, denoted *oMaxI*, is using the classical online learning paradigm. Given a stream of instances, the label of each incoming instance, x_j, is predicted using Eq. 8; next, the true label is revealed and used to update $P(f,l)$ via Eq. 6 and Eq. 7. The second algorithm, denoted *oAbMaxI*, is exploiting the uncertainty score, Eq. 12, to abstain relatively uncertain predictions. Specifically, a spurious "abstain" class is defined, and x_j is classified to this class if and only if the associated $u(x_j)$ score is greater than some pre-specified threshold, denoted u^*. In this algorithm as well, after each prediction the true label is revealed and $P(f,l)$ is updated accordingly. The underlying motivation is that by abstaining relatively uncertain predictions one may increase classification precision at the cost of reducing the associated classification recall. The single input parameter, u^*, may be thought of as a knob to control the precision/recall trade-off. Finally, in the third algorithm, denoted *oAcMaxI*, we exploit $u(x_j)$ within the active-learning paradigm. Specifically, the true label of x_j is requested and revealed if and only if $u(x_j)$ is greater than some pre-specified threshold, denoted u^*, that now represents a knob to control training set size. High u^* value implies a stringent threshold, leading to a low rate of requesting the true label. Conversely, low u^* value will result with a high rate of

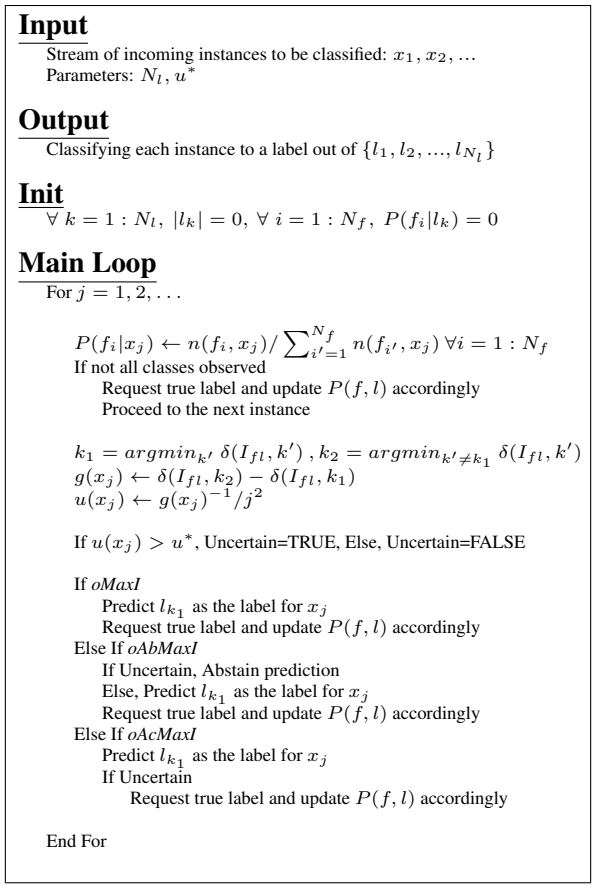

Figure 2: Pseudo-code for all three online classification algorithms proposed in this work. For the classical online setup, *oMaxI* is TRUE. For the algorithm with an abstain option, *oAbMaxI* is TRUE. For the active-learning algorithm, *oAcMaxI* is TRUE. The first "If" statement in the main "For" loop guarantees that the true label is always requested until at least one example was observed per class. For all algorithms, once the true label is requested, $P(l_k)$ is updated via Eq. 6, $\forall k = 1 : N_l$, and $P(f|l^*)$ is updated using Eq. 7 and $P(f|x_j)$, where l^* denotes the obtained true label. In our experiments we used $u^* = 0.001$ or $u^* = 0.0001$.

requesting the true label. In any event, due to the division by $(N_x^+)^2$ in Eq. 12 the obtained uncertainty scores are expected to gradually decrease, leading to a decreasing rate of asking for the true label, given that u^* is fixed. A Pseudo-code describing all three proposed algorithms is given in Fig. 2.

4 Experimental Design

4.1 Datasets and run details

We demonstrate the performance of our approach over the task of text classification. Our medium size datasets consisted of the *20NG* corpus [13], the Reuters-21578 corpus[2], and a subset of the RCV1 corpus [15], denoted as subRCV1. Following the pre-processing of these datasets reported in [21], we had for the *20NG* corpus $N_x = 16,323,\ N_f = 2,000,\ N_l = 20$; for the Reuters-21578 corpus $N_x = 8,796,\ N_f = 2,000,\ N_l = 10$; and for the subRCV1 corpus $N_x = 22,463,\ N_f = 2,000,\ N_l = 10$. In addition, we considered two large corpora. First, all documents in the entire RCV1 corpus [15] associated with the 50 most frequent topics; thus, for this corpus we had $N_x = 804,414,\ N_f = 5,000,\ N_l = 50$. And second, a large subset of the pages collected from the USA government ".gov" domain provided by the TREC conference [8], for which we had $N_x = 695,017,\ N_f = 5,000,\ N_l = 50$. In all five datasets the words were selected through feature selection by information gain, where the information examined is the information in the words-documents count matrix [21]. Importantly, this standard feature selection scheme is completely unsupervised and does not involve any usage of documents' labels. However, we note that this feature selection assumes access to the entire corpus of documents, which is not valid in real world online text classification. To somewhat address this concern, in the RCV1 corpus the 5000 words were selected based on the words-documents counts matrix constructed only from the first $20,000$ documents in the corpus, that are typically used as the training set for this corpus [15]. Finally, we note that the Reuters-21578 corpus and the RCV1 corpus are multi-labeled. In particular, in our RCV1 data, each document was assigned with ≈ 3 different labels. For the *oAbsMaxI* and the *oAcMaxI* algorithms we used $u^* = 0.001$ for the three medium size datasets, and

[2]Originally downloaded from www.daviddlewis.com/resources/testcollections/reuters21578/

$u^* = 0.0001$ for the two large datasets. For all datasets and algorithms, the reported results are averaged over 10 runs using 10 different random permutations of documents order.

4.2 Benchmark algorithms

We compare the performance of our algorithms in terms of macro-averaged F1 [24] to the performance of several state-of-the-art online learning algorithms. These include, the well-known Perceptron algorithm [17; 10], the Passive-Aggressive (PA) algorithm [5], and a multi-class version of the recently proposed AROW algorithm [6; 7]. For all three algorithms we reduced multi-class multi-label learning into a binary update by comparing the highest-scoring negative label with the lowest-scoring positive label. For multi-class problems for the Perceptron algorithm, this is often called Kesler's construction [19]. We denote these algorithms via *Per, PA*, and *AR*, respectively. Several recent works have demonstrated the high performance of these methods over various online text classification tasks [9; 7]. For each algorithm we tested two variants – one using the current linear model for the next prediction, and another one using the average of all previous models, which may help in averaging out noise [10]. This latter option is denoted with a suffix *Avg* in Section 5. In a pilot study we tried to normalize the counts vector representing each document under the $L2$-norm and found it has no significant impact on the quality of the results, hence we used the raw counts as input in the reported results. In addition, for the *PA* and the *AR* algorithms we first optimally tuned the algorithm's trade-off parameter to guarantee high performance of these algorithms. In the medium-size datasets this tuning was done using half of the documents along with their labels. In the large datasets it was done using 10,000 randomly selected documents along with their labels. Obviously, this tuning implies that these benchmark algorithms were using additional valuable information that may be hard to obtain in practice. Finally, to gain some perspective regarding the effectiveness of our active-learning variant, *oAcMaxI*, we implemented an uncertainty-sampling option [14] for each of the benchmark algorithms, in the spirit of [2], as explained below.

5 Experimental Results

5.1 Online learning results

F1	20NG	Reuters	subRCV1	gov	RCV1
oMaxI	**76**	68	**65**	65	**54**
Per	55	61	42	74	20
PerAvg	58	65	42	79	19
PA	72	68	38	81	13
PAAvg	73	70	35	83	19
AR	75	**71**	45	**88**	14
ARAvg	74	70	44	88	13
N_x	16,323	8,796	22,463	695,017	804,414

Table 1: Macro-averaged F1 results for all online algorithms. The last row indicates the total number of documents in each corpus.

In Table 1 we present the macro-averaged F1 results for all the examined online algorithms. Evidently, *oMaxI* was superior to the benchmark algorithms in three datasets, and inferior in two datasets – *Reuters* and *gov*, especially with respect to the *AR* algorithm. However, while for *oMaxI* we have relatively robust performance over all datasets examined, for the *AR* algorithm we have very low performance over the *RCV1* data, due to its low recall results, presumably since these data are highly multi-labeled. In contrast, *oMaxI* performs well even on these data, although by construction it classifies each document to a single class. Thus, we conclude that the proposed *oMaxI* algorithm may at the least be considered comparable to the benchmark state-of-the-art algorithms we examined. In addition, it is important to bear in mind that the performance of the *PA* and *AR* algorithms rely on tuning their trade-off parameter, using many labeled documents. In fact, preliminary results of these algorithms with no tuning were substantially inferior. In contrast, the *oMaxI* algorithm requires no tuning, which is a practically important advantage.

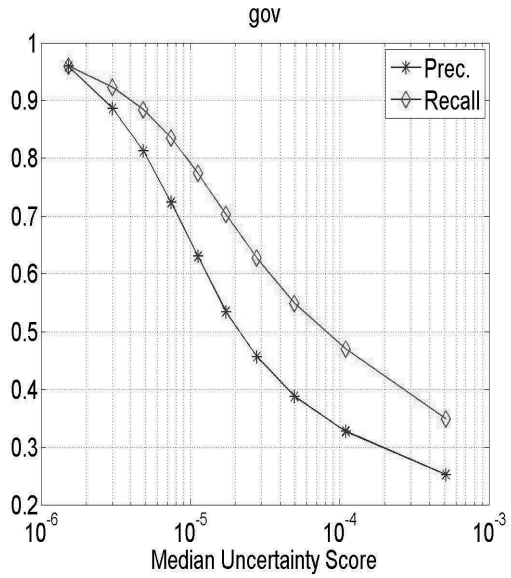

Figure 3: Macro-averaged precision and recall as a function of the uncertainty score, $u(x_j)$, for the *gov* dataset.

5.2 Prediction Accuracy vs. Prediction Uncertainty

We considered the relation between our uncertainty score definition and the associated classification accuracy. To that end, we estimated $u(x_j)$ per document using Eq. 12 during the runs of the *oMaxI* algorithm. Next, we divided all documents in each run into 10 equally populated groups according to their $u(x_j)$ scores, and estimated the macro-averaged precision and recall [24] obtained within each group. In Fig. 3 we present these results for the *gov* data. Similar results were obtained for the other datasets. As depicted in the figure, as the median uncertainty score found in each group is decreasing, the associated precision and recall are increasing, supporting the validity of Eq. 12 as a strategy to estimate prediction uncertainty.

Prec. / Recall	20NG	Reuters	subRCV1	gov	RCV1
oMaxI	76 / 76	72 / 65	82 / 54	58 / 74	79 / 41
oAbMaxI	83 / 56	80 / 45	80 / 42	63 / 68	79 / 37
oAbMaxI*	87 / 59	88 / 50	88 / 46	65 / 79	81 / 42
fracAbst	33%	25%	21%	16%	9%

Table 2: Final Macro-averaged **Precision** and **Recall** results for the *oAbMaxI* algorithm that abstains relatively uncertain predictions. For comparison, in the first row we repeat the results of the *oMaxI* algorithm. The results in the *oAbMaxI* row are estimated across all documents. The results in the *oAbMaxI** row are estimated only for documents that were not classified to the "abstain" class. The last row indicates the fraction of documents for which the algorithm abstained a prediction.

5.3 Results in Abstain mode

In the *oAbMaxI* algorithm a document is classified to a spurious "abstain" class if and only if $u(x_j) > u^*$ where u^* is a pre-specified threshold. We experimented with this algorithm over the three medium-size datasets with $u^* = 0.001$, and over the two large datasets with $u^* = 0.0001$. In Table 2 we present the obtained macro-averaged precision and recall [24]. As expected, the results of this algorithm are typically higher in terms of precision, while lower in terms of recall. If we estimate the precision and recall without considering the documents assigned to the "abstain" class, this tendency is even more dominant, as evident in the row entitled *oAbMaxI**. In addition, in light of the results depicted in Fig. 3 we conclude that the single input parameter, u^*, may be used as a knob to control precision/recall trade-off.

5.4 Results in Active Learning mode

MacAvg F1	20NG	Reuters	subRCV1	gov	RCV1
oMaxI	76	68	65	65	54
oAcMaxI	76 (25%)	69 (20%)	65 (17%)	68 (9.7%)	53 (3.7%)
oAcMaxI*	83 (25%)	78 (20%)	68 (17%)	71 (9.7%)	54 (3.7%)
AcPer	53 (23%)	59 (29%)	35 (20%)	60 (7%)	17 (17%)
AcPerAvg	56 (23%)	61 (29%)	35 (20%)	64 (7%)	17 (17%)
AcPA	62 (18%)	68 (40%)	33 (17%)	76 (17%)	15 (7%)
AcPAAvg	63 (18%)	68 (40%)	33 (17%)	77 (17%)	14 (7%)
AcAR	66 (21%)	67 (17%)	40 (15%)	74 (4%)	14 (9%)
AcARAvg	67 (21%)	66 (17%)	38 (15%)	74 (4%)	14 (9%)

Table 3: Final Macro-averaged F1 results for all algorithms in an active learning mode. The fraction of documents for which the true label was requested is indicated in parenthesis. The results in the *oAcMaxI** column are estimated only for documents for which the true label was not requested, i.e., the prediction was considered relatively certain.

In the *oAcMaxI* algorithm the true label is requested if and only if $u(x_j) > u^*$ where u^* is again a pre-specified threshold. For comparison, we implemented an uncertainty-sampling variant [14] for each of the benchmark algorithms, based on the technique proposed in [2]. Specifically, given an incoming document the algorithm requests the true label with probability $b/(b+\Delta)$, where Δ is the difference between the highest- and the second-highest prediction score and $b > 0$ is a pre-specified parameter controlling the number of requested labels. The learning model is then updated sequentially only with respect to documents for which the true label was obtained. For the *oAcMaxI* algorithm we used $u^* = 0.001$ for the three medium-size datasets, and $u^* = 0.0001$ for the two large datasets. For each of the benchmark algorithms we tried $b = 0.001, 0.01, 0.1, 1.0, 10.0, 50.0, 100.0$ in each dataset and we report the results for the b value for which the obtained training set size was roughly the same to that used by the *oAcMaxI* algorithm.

In Table 3 we present the obtained macro-averaged F1. In parenthesis we report the fraction of documents for which the true label was requested. For comparison, we repeat the results of the *oMaxI* algorithm that requests the true label after each prediction. First, we notice that the F1 results of the *oAcMaxI* algorithm are approximately the same as those obtained by the *oMaxI* algorithm that requests the true label after each prediction. The most extreme example is for the *RCV1* corpus in which selectively asking for $< 4\%$ of the labels is sufficient for the *oAcMaxI* algorithm to perform well. In contrast, for the benchmark algorithms we see a more significant reduction in F1 as selective sampling is employed; e.g., in the *subRCV1* data for the *ARAvg* algorithm – mainly due to reduction in precision; or in the *20NG* data for the *PA* and *AR* algorithms, mainly due to reduction in recall. To further examine the effectiveness of the *oAcMaxI* algorithm we made an additional run over the *RCV1* data while setting a more stringent uncertainty threshold of $u^* = 0.001$. Correspondingly, the algorithm requested the true label for $< 1\%$ of the documents, but nonetheless obtained macro-averaged precision and recall of 72 and 37, respectively (F1=49). Remarkabely, these results are still clearly superior to the results of all the benchmark online algorithms over these data, although these algorithms requested and exploited the true label after each and every prediction. Finally, from a practical perspective, one obviously can correct classification errors once the true label is revealed. Hence, it is meaningful to estimate the precision and recall only for documents for which the true label was not requested. These results are presented in the row entitled *oAcMaxI**. As expected, we see that F1 performance are improved in this mode. In other words, we obtain higher classification accuracy for documents for which our algorithm estimated its prediction as relatively certain.

6 Discussion

We presented a classification scheme which is based on a simple principle of assigning the test set with labels so to maximize the information in the obtained co-occurrence matrix of features vs. labels. We further described a simple heuristic to estimate the uncertainty associated with each prediction and outlined three concrete algorithms. A classical online classification algorithm; an algorithm that abstains highly uncertain predictions; and an active-learning algorithm that requests the true label if and only if the prediction is considered relatively uncertain. Our experimental results suggest that the proposed algorithms are comparable or superior to state-of-the-art online classification methods.

The current work is inspired by the Information Bottleneck (IB) method [23; 22] and in particular by the sequential IB algorithm [21]. Here, we expand this earlier work in three dimensions. We show that the notion of information maximization can be exploited for *supervised* classification. We

derive the probabilistic framework and update rules that allow utilizing these ideas in the context of online learning. And finally, we propose a simple strategy to estimate prediction uncertainty and demonstrate its utility in abstaining uncertain predictions and within the active-learning paradigm.

It seems worth pursuing the performance of our approach in batch mode, where the algorithm can continue to cycle over the instances while aiming to improve classification accuracy. In addition, the definition of our prediction uncertainty score, in Eq. 12, calls for a more rigorous understanding, as we intend to investigate in future research.

Acknowledgments

The authors thank Shai Fine for insightful discussions. This research was supported in part by the Israeli Science Foundation grant ISF-1567/10. KC is a Horev Fellow, supported by the Taub Foundations.

References

[1] Allan, J.: Topic Detection and Tracking: Event-based Information Organization. The Kluwer International Series On Information Retrieval (2002)

[2] Cesa-Bianchi, M., Gentile, C., Zaniboni, L.: Worst-Case Analysis of Selective Sampling for Linear Classification. The Journal of Machine Learning Research. 7, 1205–1230 (2006)

[3] Cover, T. M., Thomas, J. A.: Elements of Information Theory. Wiley (2006)

[4] Crammer, K.: Online Learning of Complex Categorical Problems. Doctoral dissertation, The Hebrew University of Jerusalem, Jerusalem, Israel (2004)

[5] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online Passive-Aggressive Algorithms. The Journal of Machine Learning Research. 7, 551–585 (2006)

[6] Crammer, K., Dredze, M., and Kulesza, A.: Adaptive Regularization Of Weight Vectors. In: Advances in Neural Information Processing Systems (NIPS) (2009)

[7] Crammer, K., Dredze, M., Kulesza, A.: Multi-Class Confidence Weighted Algorithms. In: Empirical Methods in Natural Language Processing (EMNLP) (2009)

[8] Craswell, N., Hawking, D.: Overview of the Trec-2002 Web track. In: Proc. of the 10th Text Retrieval Conference (TREC-11) (2002)

[9] Dredze, M., Crammer, K., Pereira, F.: Confidence-Weighted Linear Classification. In: The International Conference on Machine Learning (ICML) (2008)

[10] Freund, Y., Schapire, R.: Large Margin Classification Using the Perceptron Algorithm. Machine Learning. 277–296 (1998)

[11] Freund, Y., Seung, H. S., Shamir, E., Tishby, N.: Selective Sampling Using the Query by Committee Algorithm. Machine Learning. 28(2), 133 (1997)

[12] Joshi, A. J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2372–2379 (2009)

[13] Lang, K.: Learning to filter netnews. In: Proc. of the 12th Int. Conf. on Machine Learning (ICML) (1995)

[14] Lewis, D. D., Gale, W. A.: A sequential algorithm for training text classifiers. In: Proc. of the 17th Ann. Int. ACM SIGIR conference. 3–12 (1994)

[15] Lewis, D. D., Yang, Y., Rose, T. G., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. The Journal of Machine Learning Research. 5, 361 (2004)

[16] Lin, J.: Divergence Measures Based on the Shannon Entropy. IEEE Transactions on Information Theory. 37(1), 145–151 (1991)

[17] Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review. 65, 386 (1958)

[18] Salton, G.: Developments in Automatic Text Retrieval. Science. 253, 974–980 (1990)

[19] Duda, R.O., Hart, P.E., and Stork, D.G.: Pattern Classification, 2nd edition, Wiley press (2001)

[20] Slonim, N., Tishby, N.: The power of word clusters for text classification. In: 23rd European Colloquium on Information Retrieval Research (ECIR) (2001)

[21] Slonim, N., Friedman, N., Tishby, N.: Unsupervised document classification using sequential information maximization. In: Proc. of the 25th Ann. Int. ACM SIGIR conference (2002)

[22] Slonim, N.: The Information Bottleneck: Theory and applications. Doctoral dissertation, The Hebrew University of Jerusalem, Jerusalem, Israel (2002)

[23] Tishby, N., Pereira, F., Bialek, W.: The Information Bottleneck method. In: Proc. 37th Allerton Conf. on Communication and Computation (1999)

[24] van Rijsbergen, C. J.: Information Retrieval. London: Butterworths (1979)

[25] Yom-Tov, E., Fine, S., Carmel, D., Darlow, A., Amitay, E.: Improving Document Retrieval According to Prediction of Query Difficulty. In: Proc. of the 13th Text REtrieval Conference (TREC2004) (2004)

[26] Zhang, L., Zhu, J., Yao, T.: An evaluation of statistical spam filtering techniques. In: ACM Trans. on Asian Language Information Processing (TALIP). 3 (4), 243–269 (2004)