Optimistic Active Learning using Mutual Information

Yuhong Guo and Russ Greiner
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
{ yuhong | greiner }@cs.ualberta.ca

Abstract

An "active learning system" will sequentially decide which unlabeled instance to label, with the goal of efficiently gathering the information necessary to produce a good classifier. Some such systems greedily select the next instance based only on properties of that instance and the few currently labeled points — e.g., selecting the one closest to the current classification boundary. Unfortunately, these approaches ignore the valuable information contained in the other unlabeled instances, which can help identify a good classifier much faster. For the previous approaches that do exploit this unlabeled data, this information is mostly used in a conservative way. One common property of the approaches in the literature is that the active learner sticks to one single query selection criterion in the whole process. We propose a system, MM+M, that selects the query instance that is able to provide the maximum conditional mutual information about the labels of the unlabeled instances, given the labeled data, in an optimistic way. This approach implicitly exploits the discriminative partition information contained in the unlabeled data. Instead of using one selection criterion, MM+M also employs a simple on-line method that changes its selection rule when it encounters an "unexpected label". Our empirical results demonstrate that this new approach works effectively.

1 Introduction

There are many situations where *unlabeled* instances are plentiful and cheap, but it is expensive to label these instances. For example, consider the challenge of learning a classifier that can determine which webpages contain job ads. One can easily grab literally billions of webpages at essentially no cost. However, to produce an effective training set, we first need *labels* for a sufficient number of these pages; unfortunately, this typically requires paying a person to produce each such label. It is therefore useful to find a small set of pages that (when labeled) will produce a high quality classifier. An "active learning" system will sequentially select the most informative pages to label from a pool of unlabeled pages, then produce a classifier from these labeled pages. An active learner is good if it produces the best classifier from a small number of labeled pages.

This paper presents an effective active learner, "MM+M", based on two ideas: (1) In general, select the query instance that provides the *maximum conditional mutual information* about the labels of the unlabeled instances, given the labeled data. There is a subtlety here that we resolve by using an "optimistic guess" about the query's label. (2) It can be helpful to allow the selection criterion to depend on the outcomes of previous selections. In our case, if this optimistic guess is wrong for the selected query, MM+M uses a different strategy to identify the next query, then returns to use the maximum information strategy for the next selection.[1]

Section 2 presents related works, to help position, and motivate, our approach. Section 3 then introduces our specific MM+M approach, and Section 4 presents experimental results based on our implementation of these ideas. The webpage http://www.cs.ualberta.ca/~greiner/RESEARCH/OptimisticActiveLearning provides more information about our approach, and the experiments.

2 Related Work

Many previous researchers have addressed this "active learning" task in various different ways. Some of their systems use a very simple heuristic to determine which instance to label next: select the most uncertain instance, based on the classifier produced using the current set of labeled instances. Freund *et al.* [1997] employed a committee of classifiers and choose the instance on which the committee members disagree. Lewis and Gale [1994] used a probabilistic classifier, over binary classes; here this most-uncertain approach would select the instance whose conditional probability $P(\, y_i = 1 \,|\, \mathbf{x}_i \,)$ is closest to 0.5. The same principle is also used in active learning with support vector machines [Tong and Koller, 2000; Schohn and Cohn, 2000; Campbell *et al.*, 2000], where it suggests choosing the instance closest to the classification boundary. Tong and Koller [2000] analyzed this active learning as a version space reduction process, while Schohn and Cohn [2000] used a

[1] The name "MM+M" is short for "MCMI[min]+MU". The first point corresponds to "MCMI", the subtlety to "[min]" and the second point to "+MU".

heuristic search view. Our MM+M algorithm incorporates this approach as one of its components. While this "most uncertain" approach often works well, we provide empirical results that the complete MM+M system typically works better.

These "most uncertain" approaches decide which instance to select based only on how that instance relates to the current classifier(s), which in turn is based on only a few labeled instances; notice in particular this selection does not depend on the remaining unlabeled instances. As the goal is producing a classifier that has a good generalization performance, it makes sense to use (at least) the marginal distribution $P(\mathbf{x})$ over this unlabeled data. This motivates a second class of approaches, which use this unlabeled data. Cohn *et al.* [1996] and Zhang and Chen [2002] employed the unlabeled data by using the prior density $P(\mathbf{x})$ as weights. Roy and McCallum [2001] selected instances to optimize expected generalization error over the unlabeled data. Others also used the clustering distribution of the unlabeled instances: Xu *et al.* [2003] proposed a representative sampling approach that selected the cluster centers of the instances lying within the margin of the support vector machine. Nguyen and Smeulders [2004] presented a mathematical model that explicitly combines clustering and active learning together. McCallum and Nigam [1998] used an EM approach to integrate the information from unlabeled data, while Muslea *et al.* [2002] combined active learning with semi-supervised learning.

Our MM+M uses an "optimistic" information theoretic way to use the unlabeled instances: seek the instance whose *optimistic* label (i.e., the "best" of its possible labels) leads to the maximum mutual information about the labels of the remaining unlabeled instances. This approach implicitly exploits the clustering information contained in the unlabeled data, in an optimistic way.

While this optimistic heuristic typically works, it is occasionally misled. When this happens, MM+M employs a different rule to select the next instance to label. This means our selection process is not as myopic as the other approaches, as it is based on the outcome of the previous instance. Our empirical results show that this component is critical to MM+M's success.

3 The MM+M Active Learner

We let $\mathbf{x}$ denote the input features of an instance, $y \in \{1, ..., K\}$ denote the class label, $L = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ denote the set of labeled instances and U denote the index set for the unlabeled data $\{\mathbf{x}_1, ..., \mathbf{x}_m\}$. Here, $\mathbf{X}_U$ refers to the set of unlabeled instances.

Our approach uses probabilistic classifiers that compute the posterior distribution of the class label y, conditioned on the input $\mathbf{x}$; our MM+M implementation uses logistic regression.

3.1 Using Mutual Information

As the goal of active learning is to learn the best classifier with the least number of labeled instances, it may make sense to select the instance whose label provides the most information; i.e., the one we currently know least about:

$$\operatorname*{argmax}_{i \in U} H(\, Y_i \,|\, \mathbf{x}_i, L\,) \,. \tag{1}$$

where

$$H(\, Y_i \,|\, \mathbf{x}_i, L\,) \;\; = \;\; -\sum_{y_i} P(\, y_i \,|\, \mathbf{x}_i, L\,) \log P(\, y_i \,|\, \mathbf{x}_i, L\,)$$

represents the conditional entropy of the unknown label Y_i wrt the instance $\mathbf{x}_i$ given the labeled data L. For binary classes, this measure prefers instances whose conditional distribution $P(\, y = 1 \,|\, \mathbf{x}, L\,)$ is closest to 0.5. We refer to this as the "MU-instance" (for "most uncertain").

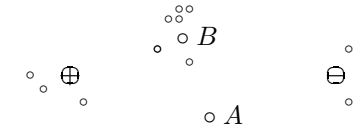

Figure 1: Problem with Optimizing (1)

Unfortunately, this MU approach is limited in that its assessment of an instance involves only the small set of currently-labeled instances (that produce the classifier used in this step) but not the distribution of the other unlabeled instances. To see why this is problematic, consider Figure 1. The MU-instance here will be the one nearest the bisector — "A". Notice, however, this label does not provide as much information about the remaining unlabeled instances as B: As B is in the midst of a cluster of points, its value will significantly increase our confidence in the labels of the neighbors.

Based on this observation, we propose using a *mutual information criterion* for instance selection: select the instance whose label will provide maximum mutual information about the labels of the remaining unlabeled instances, given the labeled data:

$$\operatorname*{argmax}_{i \in U} \{ H(\, Y_U \,|\, \mathbf{X}_U, L\,) \;-\; H(\, Y_U \,|\, \mathbf{X}_U, L, (\mathbf{x}_i, y_i)\,) \} \tag{2}$$

As the first term in (2) does not depend on the instance i selected, we can rewrite (2) as:

$$\operatorname*{argmin}_{i \in U} H(\, Y_U \,|\, \mathbf{X}_U, L, (\mathbf{x}_i, y_i)\,). \tag{3}$$

Assuming we use a parametric probabilistic conditional model $P(\, y \,|\, \mathbf{x}, \boldsymbol{\theta}\,)$ for the classification task, and use maximum likelihood for parameter $\boldsymbol{\theta}$ estimation, then the labeled data $L + (\mathbf{x}_i, y_i)$ will produce a classifier parameterized by $\boldsymbol{\theta}_{L+(\mathbf{x}_i, y_i)}$. This allows us to approximate criterion (3) as:

$$\operatorname*{argmin}_{i \in U} \sum_u H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}_{L+(\mathbf{x}_i, y_i)}\,), \tag{4}$$

since here $H(\, Y_U \,|\, \mathbf{X}_U, \boldsymbol{\theta}\,) = \sum_{u \in U} H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}\,)$. Therefore, maximizing conditional mutual information corresponds minimizing the classification uncertainty, i.e., minimizing entropy, on unlabeled data set. Minimizing entropy on the unlabeled data has been showed useful in semi-supervised learning [Grandvalet and Bengio, 2005]. A smaller classification uncertainty usually indicates a larger classification

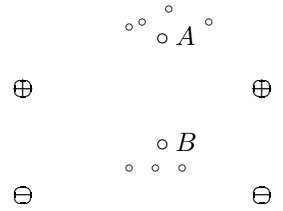

Figure 2: Why should (6) use the Optimistic Assignment?

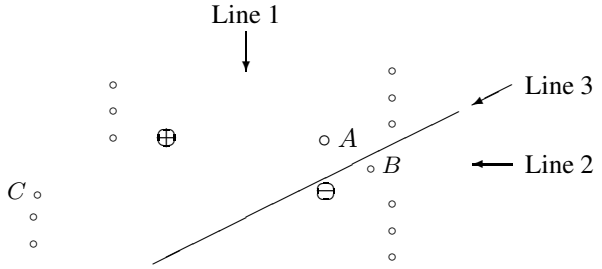

Figure 3: The MCMI[min] criterion can guess wrong. (Lines 1 and 2 are not shown, to avoid cluttering the image.)

margin, therefore leading to a better classification performance. As this means our criterion is directly related to optimizing the classification performance, we can view (4) as a discriminative information criterion for selecting query instances.

Unfortunately, (4) can not be used directly since y_i (the true value of x_i) is unknown. One obvious approach to resolve this problem is to use the apparently most-likely value of y_i based on the conditional model $\boldsymbol{\theta}_L$ trained on the labeled data L — i.e., use

$$y_i^* \quad = \quad \operatorname*{argmax}_y P(\, y \,|\, \mathbf{x}_i, \boldsymbol{\theta}_L \,) \,.$$

Alternatively, Roy and McCallum [2001] chose to take the expectation wrt Y_i

$$\operatorname*{argmin}_i \sum_y P(\, y \,|\, \mathbf{x}_i, \boldsymbol{\theta}_L \,)\, H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}_{L+(\mathbf{x}_i,y)} \,) \,. \qquad (5)$$

(We will later refer to this as the "MCMI[avg]-instance".) This corresponds to the "SELF-CONF" approach in [Baram *et al.*, 2004]. Notice both of these approaches use the $P(\, y \,|\, \mathbf{x}_i, \boldsymbol{\theta}_L \,)$ information determined by the labeled data; unfortunately, in the typical active learning situations where there are very few labeled instances L, the labeled data might lead to a bad classifier, therefore the $P(\, y \,|\, \mathbf{x}_i, \boldsymbol{\theta}_L \,)$ will not be helpful. This concern is verified in our empirical study.

We propose instead using an "optimistic" strategy, which takes the y value that minimizes the entropy term. Using this optimistic strategy, our MCMI[min] approach would compute (4) as

$$\operatorname*{argmin}_{i \in U} f(\, i \,)$$

where

$$f(\, i \,) = \min_y \sum_u H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}_{L+(\mathbf{x}_i,y)} \,). \qquad (6)$$

Notice this measure will give each candidate $\mathbf{x}_i$ instance its best possible score, over the set of y labels. For instance, if the classifier is seeking a linear separator, and if there is a setting y for which the classifier based on $L + (\mathbf{x}_i, y)$ nicely separates the unlabeled data with a wide margin, then (6) will give the score associated with this large margin. We will refer to this optimal $i^* = \operatorname{argmin}_{i \in U} f(\, i \,)$ as the "MCMI[min] instance".

This differs from Roy and McCallum [2001], which chooses y in a supervised way, as our approach can be viewed as choosing y in an unsupervised way. This approach is motivated by the "clustering assumption" [Sindhwani *et al.*, 2005] used in many unsupervised and semi-supervised learning algorithms: instances separated by a wide margin usually belong to different classes. This means minimizing classification uncertainty is usually consistent with the underlying clustering partitions.

Example: To make this concrete, consider Figure 2, where each $\ominus$ and $\oplus$ is a labeled instance, and the $\circ$'s are unlabeled. Assigning A the label $\ominus$ would mean the data is not linearly separable, which would add a great deal of uncertainty to the remaining unlabeled instances; hence the value of $H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}_{L+(A,\ominus)} \,)$ for each unlabeled instance $\mathbf{x}_u$ would be high. By contrast, the $\oplus$ label would not change the support vectors, meaning $H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}_{L+(A,\oplus)} \,)$ would be essentially unchanged; i.e., $H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}_{L+(A,\oplus)} \,) \approx H(\, Y_u \,|\, \mathbf{x}_u, \boldsymbol{\theta}_L \,)$. Hence MCMI[min] would use the "more informative" $\oplus$ label, if it decides to use A. Now consider B, and observe both $\oplus$ and $\ominus$ are consistent with some linear separator. Notice, however, that while the $\ominus$ label will well specify the label of the points below immediately below it, the $\oplus$ label will not. This is why MCMI[min] will use the $\ominus$ label for B.

This assignment clearly depends on the *other* unlabeled instances; e.g., if those 3 instances were just *above* B, then *mutatis mutandis*, MCMI[min] would use the $\oplus$ label. Hence, the $\min$ part of MCMI[min] is exploiting relevant nuances of the $P(\, \mathbf{x} \,)$ density over the instances themselves, as it seeks the labels with the largest separation margin.

Given the choice between A versus B, our MCMI[min] criterion will prefer B, as neither $(A, \oplus)$ nor $(A, \ominus)$ will significantly increase the information about the labels for the instances (beyond the information from the other current labels), while B could be very informative — especially if its label is $\ominus$.

3.2 On-Line Adjustment

Potential Problem: There is a potential problem with this approach: at the beginning, when we only have a few labeled data, there may be many consistent parameter settings (read "classifiers"); here, any method, including ours, may well "guess" wrong. To illustrate this, consider seeking a linear separator within the data shown in Figure 3. Which point should be selected next?

Recall that MCMI[min] will seek the instance that could lead to the most certainty over the remaining unlabeled instances. If A is labeled $\ominus$, then the resulting set of 3 labeled

MM+M(U: indices of unlabeled instances; L: labeled instances)
  Repeat
    For each $i \in U$, compute
      $y(i) := \text{argmin}_y \sum_u H(Y_u \mid \mathbf{x}_u, \boldsymbol{\theta}_{L+(\mathbf{x}_i,y)})$
      $f(i) := \sum_u H(Y_u \mid \mathbf{x}_u, \boldsymbol{\theta}_{L+(\mathbf{x}_i,y(i))})$
      % ie, score based on this minimum $y(i)$ value
    Let $i^* := \text{argmin}_i f(i)$
      % instance with optimal MCMI*[min] score*
    Purchase true label w_{i^*} for $\mathbf{x}_{i^*}$
    Remove i^* from U; add $(\mathbf{x}_{i^*}, w_{i^*})$ to L.
    If $w_{i^*} \neq y(i^*)$, then
      Let $i^* := \text{argmax}_{i \in U} H(Y_i \mid \mathbf{x}_i, L)$
      Purchase true label w_{i^*} for $\mathbf{x}_{i^*}$
      Remove i^* from U; add $(\mathbf{x}_{i^*}, w_{i^*})$ to L.
  until bored.
End MM+M

Figure 4: The MM+M Algorithm

instances would suggest a vertical dividing line ("Line 1"); in fact, it is easy to see that this leads to the largest margin, which means the highest MCMI[min] score. Our algorithm will therefore select instance A, anticipating this $\ominus$ label. Now imagine the (unknown) true class dividing line is Line 3, which means our MCMI[min] approach was misled — that is, the optimistic prediction of A's label was different from its true label.

At this point, given these 3 labeled instances, MCMI[min] would next select C, anticipating that it would be labeled $\ominus$, consistent with a horizontal separating line (Line 2), Unfortunately, this too is wrong: C's label is $\oplus$, meaning MCMI[min] again guessed wrong. This example suggests it might be helpful to consider a different selection strategy after MCMI[min] has been misled.

Approach: Fortunately, our algorithm can easily detect this "guessed wrong" situation in the immediate next step, by simply comparing the actual label for A with its optimistically predicted label. When they disagree, we propose switching from MCMI[min] to another criterion; here we choose MU. This is done in our MM+M algorithm, shown in Figure 4.[2] If MM+M guesses correctly (i.e., if the label returned for the selected instance is the one producing the minimum entropy), then MM+M continues to use the MCMI[min] approach — i.e., if it is "converging" to the correct separator, it should keep going. Otherwise, it will select a single MU-instance, to help it locate a more appropriate "split" in the space, before returning to the MCMI[min] criterion.

For our example, once MM+M has seen that it was wrong about A (i.e., A was labeled $\oplus$ rather than the anticipated $\ominus$), MM+M will then select a MU-instance — i.e., an instance near the current boundary (Line 2) — which means it will select B. When B is labeled $\ominus$, we can come closer to the true separator, Line 3. Notice this means we will be confident that C's label is $\oplus$, which means we will no longer need to query this instance. In this case, the information produced by the MU-instance is more relevant than the MCMI[min]-instance.

[2]This requires $O(|U|^2)$ time to make each selection: first iterating over each $i \in U$, then for each such i, iterating over each $u \in U$ when computing the $f(i)$ score. For larger datasets, would could use sampling, for both loops.

Section 4 provides empirical evidence that this approach works well.

3.3 Approximating Logistic Regression

Recall our system uses logistic regression, which in general takes a set of labeled instances $L = \{(\mathbf{x}_i, y_i)\}$, and seeking the parameters $\boldsymbol{\Theta}$ that best fit the logistic function[3]

$$P(y \mid \mathbf{x}) = \sigma(-y\,\mathbf{x}^T\boldsymbol{\Theta}) \triangleq \frac{1}{1+\exp(-y\,\mathbf{x}^T\boldsymbol{\Theta})}\,.$$

(We use the obvious variant for multiclass classification [Schein and Ungar, 2005].) We use a regularized version, seeking the $\boldsymbol{\Theta}$ that minimizes[4]

$$\ell(\boldsymbol{\Theta}) = \sum_i \log(1+\exp(-y\mathbf{x}^T\boldsymbol{\Theta})) + \frac{\lambda}{2}\|\boldsymbol{\Theta}\|^2\,. \qquad (7)$$

There are a number of iterative algorithms for this computation [Minka, 2003]; we use Newton's method: $\boldsymbol{\Theta}^{new} := \boldsymbol{\Theta}^{old} - \mathbf{H}_L^{-1}\mathbf{g}_L$ where

$$\mathbf{g}_L = \sum_{(x,y)\in L} (1-\sigma(-y\mathbf{x}^T\boldsymbol{\Theta}))\,y\,\mathbf{x} + \lambda\boldsymbol{\Theta}$$

is the gradient based on the labeled dataset L and

$$\mathbf{H}_L = -\sum_{(\mathbf{x},y)\in L} \sigma(\boldsymbol{\Theta}_L^T\mathbf{x})\,(1-\sigma(\boldsymbol{\Theta}_L^T\mathbf{x}))\,\mathbf{x}\,\mathbf{x}^T - \lambda\mathbf{I}$$

is the Hessian. If there are n instances of d dimensions, this requires $O(n\,d^2)$ time per iteration. To compute each MCMI[min] instance, our MM+M algorithm (Figure 4) will have to compute parameters $|U| \times K$ times, as this requires computing $\boldsymbol{\Theta}_{L+(\mathbf{x}_i,y)}$ for each $i \in U$ and each of the K classes Y_i.

In order to save computational cost, we use a simple way to approximate these parameters: Assuming we have a good estimate of $\boldsymbol{\Theta}_L$, based on the $\mathbf{g}_L$ and $\mathbf{H}_L$, we can then approximate the values of $\boldsymbol{\Theta}_{L+(\mathbf{x}_i,y_i)}$, by starting with that $\boldsymbol{\Theta}_L$ value and performing just one update iteration, based on the new values of $\mathbf{g}_{L+(\mathbf{x}_i,y_i)}$ and $\mathbf{H}_{L+(\mathbf{x}_i,y_i)}$:

$$\boldsymbol{\Theta}_{L+(\mathbf{x}_i,y_i)} := \boldsymbol{\Theta}_L - \mathbf{H}^{-1}_{L+(\mathbf{x}_i,y_i)}\mathbf{g}_{L+(\mathbf{x}_i,y_i)}$$

Moreover, there are easy ways to approximate $\mathbf{H}^{-1}_{L+(\mathbf{x}_i,y_i)}\mathbf{g}_{L+(\mathbf{x}_i,y_i)}$ based on $\mathbf{H}_L^{-1}\mathbf{g}_L$.

4 Experiments

To investigate the empirical performance of our MM+M algorithm, we conducted a set of experiments on many UCI datasets [UCI, 2006], comparing MM+M with several other active learning algorithms. The four primary algorithms we considered are:

1. MCMI[min]+MU: current MM+M algorithm
2. MCMI[min]: always use the MCMI[min]-instance

[3]Of course, we include $x_0 = 1$ within each $\mathbf{x}_i$ to avoid the need to explicitly separate out the constant bias term.

[4]In our experiments, we used $\lambda = 0.01$ in binary classes, and $\lambda = 1$ for others.

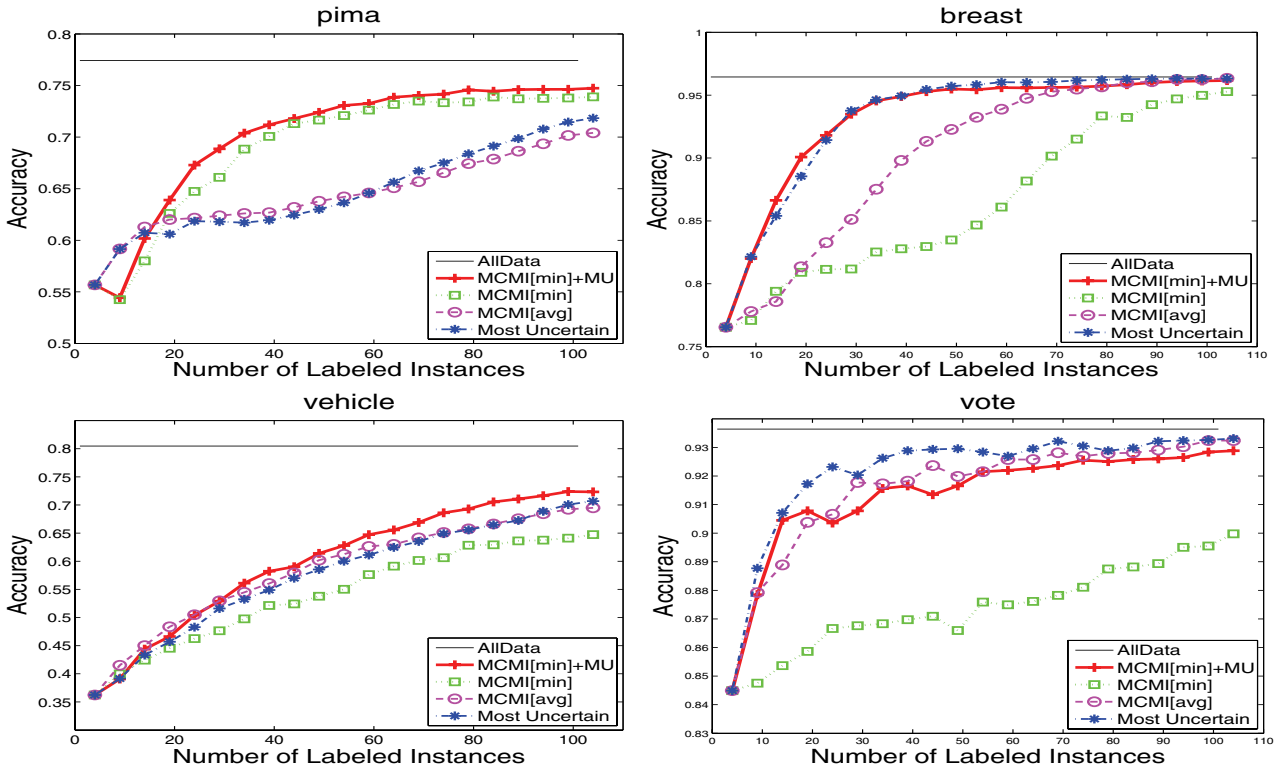

Figure 5: Comparing active learners on various UCI datasets

3. MCMI[avg]: differs from MCMI[min] by averaging over the y values (5) rather than taking the minimum (6) (note, this is same as the SELF-CONF in [Baram *et al.*, 2004])
4. MU: "most uncertain"; based on (1)

We consider the following 17 UCI datasets (we show the name, followed by its number of classes, number of instances and the number of attributes): AUSTRALIAN(2;690;14), BREAST(2;286;9); CLEVE(2;303;13); CRX(2;690;15), DIABETES(2;768;8), FLARE(2;1389;13), GERMAN(2;1000;20), GLASS2(2;163;9), HEART(2;270;13), HEPATITIS(2;155;20), MOFN(2;10;300), PIMA(2;768;8), VOTE(2;435;15),[5] IRIS(3;150;4), LYMPHOGRAPHY(4;148;18) and VEHICLE(4;846;18). Note each of the last 3 datasets has strictly more than 2 classes.

For each dataset, we ran 30 trials. For each trial, we first randomly selected $1/3$ of the instances from each class to serve as the test data. From the remaining $2/3$, we randomly picked k labeled instances from each class to form the initial labeled set L, where $k = 2$ for binary-class databases, and $k = 1$ for multiclass databases,[6] and left the remaining instances as the unlabeled pool. Each of the active learners then sequentially selects 100 instances[7] from the unlabeled pool to add to the labeled set. Every time a new instance is labeled, that active learner then retrains a new classifier on the increased labeled set and evaluates its performance on the test set. In general, let $acc_i(\, D,\, A[m]\,)$ be the accuracy of the classifier learned after the A active learner has added m labels to the D database, on the ith run, and let $acc(\, D,\, A[m]\,)$ be the average of this accuracy values, over the 30 runs, $i = 1..30$.

Our MM+M ("MCMI[min]+MU") wins on most datasets. Figure 5 shows these averaged $acc(\, D,\, A[m]\,)$ results for 4 datasets. The PIMA graph shows that MM+M can do well even when MU does relatively poorly. The fact that MCMI[min], which did not include the "MU-correction", does comparably to MM+M shows that MM+M did not need this correction here. Now consider BREAST, and notice that MCMI[min] was the worst performer here, while MM+M was able to match MU's performance — i.e., here the MU-correction was essential. (We observed here that MM+M used this MU-correction as often as possible — i.e., for essentially every other query.) That is, when MCMI[min] is working well, MM+M can capitalize on it; but when MCMI[min] is misled, MM+M can then fall back on the alternative MU approach. In the third database, VEHICLE, we see that MM+M in fact does better than both MU and MCMI[min]. (This dataset also shows that MM+M system can perform well even

[5] As suggested in [Holte, 1993], we remove the"Physician-free-freeze" variable from the VOTE data, as it alone leads to 99% accuracy.

[6] Active learning is harder for small starting Ls, and our starting size is smaller than many other projects; e.g., Schein and Ungar [2005] began with at least 20 instances, and Nguyen and Smeulders [2004] with 10 instances.

[7] We used fewer for the 3 smallest databases.

Table 1: Comparing MM+M to other Active Learners. (See text for descriptions of notation.)

database	vs MU	vs MCMI[min]	vs MCMI[avg]	vs RANDOM	vs MU-SVM
AUSTRALIAN*	**58**/ 0 (+)	1 / 0 (0)	**24**/ 0 (+)	11 / 14 (−)	**16**/ 0 (+)
BREAST*	0 / 0 (−)	**94**/ 0 (+)	**56**/ 0 (+)	**88**/ 0 (+)	0 /53 (−)
CLEVE*	**56**/ 0 (+)	**10**/ 0 (+)	**32**/ 0 (+)	**26**/ 0 (+)	**77**/ 0 (+)
CORRAL	5 / 14 (0)	**44**/ 2 (+)	1 / 0 (0)	**13**/ 0 (+)	0 /23 (−)
CRX	**80**/ 0 (+)	0 / 5 (−)	**26**/ 0 (+)	2 / 8 (−)	**13**/ 0 (+)
DIABETES*	**87**/ 0 (+)	0 / 1 (0)	**87**/ 1 (+)	11 / 12 (+)	**73**/ 8 (+)
FLARE*	**53**/ 0 (+)	**41**/ 0 (+)	**55**/ 0 (+)	**27**/ 2 (+)	**59**/ 1 (+)
GERMAN*	**42**/ 0 (+)	**18**/ 0 (+)	**77**/ 0 (+)	0 / 0 (+)	**91**/ 0 (+)
GLASS2*	**38**/ 0 (+)	4 / 0 (+)	**24**/ 0 (+)	**8**/ 0 (+)	**17**/ 0 (+)
HEART*	**30**/ 0 (+)	**19**/ 0 (+)	**20**/ 0 (+)	0 / 0 (+)	**29**/ 0 (+)
HEPATITIS*	**7**/ 0 (+)	0 / 0 (0)	**6**/ 0 (+)	0 / 0 (0)	**50**/ 0 (+)
MOFN	0 / 33 (−)	**93**/ 0 (+)	**67**/ 0 (+)	**31**/ 6 (0)	0 /37 (0)
PIMA*	**85**/ 2 (+)	2 / 0 (+)	**84**/ 2 (+)	0 / 0 (0)	**36**/13 (+)
VOTE	0 / 27 (−)	**97**/ 0 (+)	0 / 0 (−)	2 / 0 (+)	0 /16 (−)
IRIS	**68**/ 1 (+)	**17**/ 0 (+)	13 / 23 (0)	**38**/ 1 (+)	– / – ()
VEHICLE*	**56**/ 0 (+)	**84**/ 0 (+)	**38**/ 0 (+)	0 / 19 (−)	– / – ()
LYMPHOGRAPHY	0 / 4 (0)	0 / 1 (−)	0 / 30 (−)	0 / 2 (−)	– / – ()
TOTAL W/L/T	**12** / 3 / 2	**10** / 1 / 6	**13** / 2 / 2	**7** / 2 / 8	**10** / 4 / 0
Signed Rank Test	**12** / 3 / 2	**12** / 2 / 3	**13** / 1 / 3	**10** / 3 / 4	**9** / 4 / 1

when there are more than 2 classes.) In all of these examples (and many others), we see that MM+M — a composition of MCMI[min] and MU — typically does at least as well as either of these two approaches individually, and often better.

Even though the MCMI[avg] approach (of *averaging* over the possible y values) is intuitive, the PIMA graph shows that it can work worse than the (at first counter-intuitive) MCMI[min] system. But not always: MCMI[avg] is better than MCMI[min] for BREAST.

Our MM+M did not always win: The final graph shows that MU is best for the VOTE database. We see, however, that MM+M is still close.

Of course, these 4 graphs are only anecdotal. Moreover, even here we find that different active-learners are best for some, but not all, of the 100 evaluation points, where each corresponds to a specific number of additional labeled instances. We present two evaluations to help quantify this comparison. First, we quantify the quality of the different learners by counting the number of times that one algorithm was significantly better than another over the 100 evaluation points, at the $p < 0.05$ level based on a 2-sided paired t-test. (We did not not bother with a Bonferroni correction, as this is just to quickly determine which differences, at each specific evaluation point, appears significant.) We used this to compare MM+M with each other active learner $A \in \{$ MU, MCMI[min], MCMI[avg] $\}$, over all 17 databases. The results are shown in Table 1, whose entries are of the form "**#**MM+M**-wins / #**A**-wins** (s)", where A is the algorithm associated with the column. (We will explain the parenthesized "(s)" value below.) An entry is **bold** if MM+M was statistically better at least 5 more times than it was statistically worse. Each entry in the "**TOTAL** W/L/T" row shows the number of datasets where MM+M "dominated" by this "> 5 measure", when it lost by this same quantity, versus tied. We also note that, wrt these 4 active learners, our MM+M was the best active learner for 11 of the 17 databases (in that it was at least as good as the other 3 approaches, and strictly better than at least one); we marked each such database with a "*".

To describe our second set of evaluations, note that the 100 numbers $\{acc(\, D,\, A[m]\,)\}_{m=1}^{100}$ characterize the performance of A on D. We therefore ran a Wilcoxon signed rank test to determine whether these scores were significantly better for MM+M, or for the "challenger" associated with the column. This produced the parenthesized value "(s)" in each entry in Table 1, which is "+" if this signed rank test suggests that MM+M is statistically better than the alternative algorithm A at the $p < 0.05$ level, is "−" if this test suggests that A is statistically better at $p < 0.05$, and is "0" otherwise. (Hence, this test suggests that MM+M was statistically better than MU on the AUSTRALIAN database, but it was statistically worse on the BREAST database, and is comparable for CORRAL.)

The numbers at the very bottom of Table 1 are the totals for this signed rank test; they are in a similar form as the first evaluation, i.e., the number of datasets where MM+M was better/worse/tied. Hence, the Wilcoxon signed rank test shows that MM+M is statistically better than MU for 12 of the 17 datasets, worse on 3, and tied in the remaining 2. Notice the results of this Ranked Sign test is very similar to our other (ad hoc) scoring measure. We also computed 2-sided t-tests over this data, and found that it also produced very similar results; see `http://www.cs.ualberta.ca/~greiner/RESEARCH/OptimisticActiveLearning`.

We also considered two other active learners. The "RANDOM" learner simply selected instances at random. The final learner, "MU-SVM", uses a (linear kernel) SVM, and selects

the instance closest to the classification boundary — i.e., this is a variant of the "most uncertain" selection criterion. (Note we only consider the 14 binary databases here.) Using the evaluation measures shown above, the results of our experiments appear in the final two columns of Table 1.

These results confirm that our MM+M algorithm works effectively — and appears better than the other algorithms considered here. We note, in particular, that under either test considered, our MM+M is better than any of the other active learners, in at least twice as many databases.

5 Conclusion

Future Work: Section 3 motivated why our MM+M algorithm should work, and Section 4 provided evidence that it does. It would be useful to augment this with a more formal theoretical analysis, one that could precisely characterize those domains where MM+M is *guaranteed* to work effectively, or better, optimally. This may depend on both the instance distribution $P(\mathbf{x})$ as well as the nature of the conditional distribution $P(y \mid \mathbf{x})$, (e.g., linearly separable or not), and also perhaps on the initially provided labeled instances. Investigating better on-line adjustment strategies is another direction.

Contributions: This paper explores the insight that an active learner should identify instances whose label will help determine the labels of the other unlabeled instances. This leads to a novel approach: select the instance that provides the maximum conditional mutual information about the labels of the unlabeled instances, given the labeled data, based on an optimistic assumption about the label for each instance — the "min" in (6). When this optimistic assumption fails for an instance (i.e., when the real label does not match the predicted optimistic one), our MM+M selects one MU-instance before reverting back to MCMI-instances.

Our empirical results demonstrate first that our MM+M works effectively, and second that its effectiveness depends on its two major design decisions — the counter-intuitive "min" in (6) and the on-line "MU-correction".

For more details about the experiments, as well as other insights about the MM+M algorithm, see `http://www.cs.ualberta.ca/~greiner/RESEARCH/OptimisticActiveLearning`.

Acknowledgments

We gratefully acknowledge the insightful comments of the four anonymous reviewers, and of our colleagues, W Bishof and D Schuurmans. This work was partially funded by NSERC and the Alberta Ingenuity Centre for Machine Learning.

References

[Baram *et al.*, 2004] Y. Baram, R. EI-Yaniv, and K. Luz. Online choice of active learning algorithms. *Machine Learning*, 5, 2004.

[Campbell *et al.*, 2000] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *ICML*, 2000.

[Cohn *et al.*, 1996] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4, 1996.

[Freund *et al.*, 1997] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28, 1997.

[Grandvalet and Bengio, 2005] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[Holte, 1993] Robert C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[Lewis and Gale, 1994] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *ACM-SIGIR*, 1994.

[McCallum and Nigam, 1998] A. McCallum and K. Nigam. Employing em in pool-based active learning for text classification. In *ICML*, 1998.

[Minka, 2003] Thomas P. Minka. A comparison of numerical optimizers for logistic regression. Technical report, 2003. http://research.microsoft.com/ minka/papers/logreg/.

[Muslea *et al.*, 2002] I. Muslea, S. Minton, and C. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *ICML*, 2002.

[Nguyen and Smeulders, 2004] H. Nguyen and A. Smeulders. Active learning using pre-clustering. In *ICML*, 2004.

[Roy and McCallum, 2001] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.

[Schein and Ungar, 2005] Andrew I. Schein and Lyle H. Ungar. Active learning for multi-class logistic regression. In *LEARNING 2005*, 2005.

[Schohn and Cohn, 2000] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *ICML*, 2000.

[Sindhwani *et al.*, 2005] V. Sindhwani, M. Belkin, and P. Niyogi. The geometric basis of semi-supervised learning. In *Semi-supervised Learning*. MIT Press, 2005.

[Tong and Koller, 2000] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *ICML*, 2000.

[UCI, 2006] UCI, 2006. `http://www.ics.uci.edu/~mlearn/MLSummary.html`.

[Xu *et al.*, 2003] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *European Conference on Information Retrieval*, 2003.

[Zhang and Chen, 2002] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Trans on Multimedia*, 4:260–258, 2002.