Active Algorithm Selection

Feilong Chen and **Rong Jin**
Department of Computer Science
Michigan State University
East Lansing, MI 48823
chenfeil,rongjin@cse.msu.edu

Abstract

Most previous studies on active learning focused on the problem of model selection, i.e., how to identify the optimal classification model from a family of predefined models using a small, carefully selected training set. In this paper, we address the problem of **active algorithm selection**. The goal of this problem is to efficiently identify the optimal learning algorithm for a given dataset from a set of algorithms using a small training set. In this study, we present a general framework for active algorithm selection by extending the idea of the Hedge algorithm. It employs the worst case analysis to identify the example that can effectively increase the weighted loss function defined in the Hedge algorithm. We further extend the framework by incorporating the correlation information among unlabeled examples to accurately estimate the change in the weighted loss function, and Maximum Entropy Discrimination to automatically determine the combination weights used by the Hedge algorithm. Our empirical study with the datasets of WCCI 2006 performance prediction challenge shows promising performance of the proposed framework for active algorithm selection.

Most of the previous studies on active learning are focused on the problem of model selection. Given a dataset, the goal of active model selection is to select the most informative examples for labeling such that the optimal model from a predefined model family can be identified using a small number of labeled examples (i.e., the performance of the learning algorithms is maximized). The key idea behind most active learning algorithms is to select the examples that are the most uncertain to classify by the learned classification model. Thus, one of the key issues with active learning is how to estimate the classification uncertainty for each unlabeled example. A number of approaches have been developed for measuring classification uncertainty, including query by committee (Abe & Mamitsuka 1998; Seung, Opper, & Sompolinsky 1992), Fisher information (MacKay 1992), and classification margin (Tong & Koller 2000).

In this paper, we will investigate another type of active learning problem, named "**active algorithm selection**". Given a dataset and a set of machine learning algorithms for binary classification, the goal of active algorithm selection is to efficiently identify the optimal learning algorithm using a small number of labeled examples. To be general, we assume each learning algorithm in the ensemble to be a black box that can be assessed only through two interfaces, i.e., the *training interface* that builds a classification model from given labeled examples, and the *testing interface* that classifies test examples.

A concern one may have is that the algorithm identified as optimal based on a small training set might not be the optimal given a large training set. A similar situation rose in the study of active learning. However, it has been proved both theoretically and empirically by a number of studies (Freund *et al.* 1997; Lewis & Gale 1994; Schohn & Cohn 2000) that by carefully selecting a small number of examples, one may very well tell the overall performance of a classifier given a large training set. This is further confirmed by our empirical study that the optimal learning algorithm can be identified efficiently with a small number of examples.

A straightforward approach toward active algorithm selection is to extend the existing active learning algorithms. For instance, following the idea of query by committee(QBC) algorithm, we can first create a model ensemble by applying the given machine learning algorithms to build a set of classification models from the labeled examples. We can then measure the classification uncertainty of an unlabeled example by the disagreement in the predictions by the ensemble. One problem with this simple approach is that each learning algorithm in the ensemble is treated with equal importance in measuring the uncertainty of unlabeled examples. Note that this treatment is contradictory to our goal, i.e., identifying the best learning algorithm from the ensemble. Thus, one key question in extending the idea of QBC to active algorithm selection is how to determine the appropriate weights for the given learning algorithms such that we can reliably estimate the classification uncertainty of unlabeled examples. To this end, we propose a general framework for active algorithm selection based on the Hedge algorithm (Freund & Schapire 1997). The key idea of the proposed framework is to identify the unlabeled example that can effectively increase the weighted loss function defined in the Hedge algorithm, even in the worst case. We will show that this is equivalent to selecting

examples with the largest classification uncertainty. Furthermore, we incorporate the correlation information among the unlabeled examples to provide a more accurate estimation of the weighted loss function, and Maximum Entropy Discrimination (Jaakkola, Meila, & Jebara 1999) to automatically determine the weights used by the Hedge algorithm.

In the remaining of this paper, we first review the previous work on active learning. We also review the Hedge algorithm and Maximum Entropy Discrimination, which are closely related to this study. Then, we define the problem of active algorithm selection, and present the general framework for active algorithm selection and the related machine learning algorithms. Furthermore, we present our empirical study using the datasets of the WCCI 2006 performance prediction challenge. Finally we conclude our work.

Related Work

In this section, we will first review the previous work on active learning, followed by the overview of the Hedge algorithm and the Maximum Entropy Discrimination algorithm.

Active Learning Active learning is a widely studied problem in machine learning. Most active learning algorithms are conducted iteratively. In each iteration, the algorithm selects the most informative example, i.e., the example with the highest classification uncertainty, for manual labeling; the acquired labeled example is then added to the training data to retrain the classification model. The step of model training and the step of example selection will alter iteratively until the satisfying classification accuracy is achieved. One of the key issues in active learning is how to measure the classification uncertainty of unlabeled examples. Three major approaches have been developed for this purpose. In the first approach (Seung, Opper, & Sompolinsky 1992; Abe & Mamitsuka 1998), an ensemble of classification models that are consistent with the labeled examples is first created. The classification uncertainty of an unlabeled example is then measured by the disagreement in the classes predicted by all the classification models in the ensemble for the example. The second approach (Tong & Koller 2000; Campbell, Cristianini, & Smola 2000; Roy & McCallum 2001) measures the classification uncertainty of an example by its distance to the decision boundary of the learned classification model. The third approach (MacKay 1992; Zhang & Oles 2000) represents the uncertainty of a classification model by its Fisher information matrix, and measures the classification uncertainty of an unlabeled example by its projection onto the Fisher information matrix. In the past, active learning algorithms have been applied to a number of real-world problems, such as text categorization (Tong & Koller 2000; Liere & Tadepalli 1997), information ranking (Saar-Tsechansky & Provost 2004), natural language processing (Thompson, Califf, & Mooney 1999).

Hedge Algorithm The Hedge algorithm (Freund & Schapire 1997) is designed to combine multiple classification models in order to achieve optimal classification accuracy. Since our work is motivated by the Hedge algorithm, we will review it in slightly more detail.

Let us denote by $\mathcal{M} = (\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_S)$ the ensemble of S binary classification models. Let $l(y, \hat{y})$ denote the loss function between the true class label y and the predicted label $\hat{y}$, which outputs one when $y \neq \hat{y}$ and zero if $y = \hat{y}$. Let's denote by $\mathcal{D} = \{(\mathbf{x}_1, y_l), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{N_l}, y_{N_l})\}$ the set of labeled examples, where $\mathbf{x}_t \in \mathbf{R}^d$ is a vector of d dimension and $y_t \in \{-1, +1\}$ is a binary class label. For each classification model $\mathbf{m}_i$ in the ensemble $\mathcal{M}$, we define its loss function for the labeled examples $\mathcal{D}$ as follows:

$$L_i(\mathcal{D}) \quad = \quad \sum_{t=1}^{N_l} l\left(y_t, \mathbf{m}_i(\mathbf{x}_t)\right)$$

where $\mathbf{m}_i(\mathbf{x})$ represents the class label of example $\mathbf{x}$ predicted by the classification model $\mathbf{m}_i$.

To achieve good classification accuracy, the Hedge algorithm linearly combines all the classification models in the ensemble $\mathcal{M}$ by assigning a w_i to each model $\mathbf{m}_i$. These weights are initialized to be an uniform distribution at the beginning, and are updated by each labeled example $(\mathbf{x}_t, y_t)$:

$$w_i^{t+1} \quad \leftarrow \quad \frac{w_i^t \beta^{l(y_t, \mathbf{m}_i(\mathbf{x}_t))}}{\sum_{j=1}^{S} w_j^t \beta^{l(y_t, \mathbf{m}_i(\mathbf{x}_t))}} \tag{1}$$

where parameter $\beta \in (0, 1)$, referred to as the *punishment factor*, is used to decrease the weights for the classification models that make incorrect predictions for example $(\mathbf{x}_t, y_t)$. One of the most important properties regarding the Hedge algorithm is the mistake bound when we linearly combine the classification models in the ensemble $\mathcal{M}$ using the weights w_i^t, i.e.,

$$\begin{aligned} L_\beta(\mathcal{D}) \quad &= \quad \sum_{i=1}^{S} \sum_{t=1}^{N_l} w_i^t l\left(y_t, \mathbf{m}_i(\mathbf{x}_t)\right) \\ &\leq \quad \frac{\log(1/\beta)}{1-\beta} \min_i L_i(\mathcal{D}) + \frac{1}{1-\beta} \log S \end{aligned} \tag{2}$$

where $L_\beta(\mathcal{D})$, referred to as the weighted loss function, is the cumulative classification error for the combined classification model. Eqn. (2) indicates that, by using appropriate combination weights, the combined classification model can achieve a classification error that is close to the one by the best classification model in the ensemble $\mathcal{M}$.

Maximum Entropy Discrimination Finally, we extended the work of Maximum Entropy Discrimination (Jaakkola, Meila, & Jebara 1999) in this study to estimate the weights for combining different learning algorithms. The essential idea of Maximum Entropy Discrimination is to estimate the posterior distribution for the given classification models that satisfies the following objectives simultaneously: On one hand, the posterior distribution should maximize its entropy. In other words, the posterior distribution should be close to an uniform distribution; On the other hand, the linearly combined classification model weighted by the posterior distribution should minimize the classification error of the labeled examples.

Active Algorithm Selection

In this section, we will first define the problem of active algorithm selection, and then present a general framework of active algorithm selection and the related machine learning algorithms.

Problem Definition

Let $\mathcal{A} = (\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_S)$ denote the set of learning algorithms for binary classification. For a given dataset, our goal is to identify the best learning algorithm among $\mathcal{A}$ using the minimum number of labeled examples. To generalize our problem, we assume that each learning algorithm can be assessed only through two application interfaces, i.e., the training interface that builds a binary classification model from given labeled examples and the testing interface that classifies every unlabeled example.

One approach toward active algorithm selection is to extend the algorithm of QBC (Seung, Opper, & Sompolinsky 1992). The major problem with this approach is that each learning algorithm is weighted equally in determining the classification uncertainty, which is insufficient given that some learning algorithms can be significantly better than others. In the following subsections, we will first present a general framework for active algorithm selection based on the Hedge algorithm, followed by the algorithm to explore example correlation information and the Maximum Entropy Discrimination algorithm that automatically determines the combination weights.

A MaxMin Framework for Active Algorithm Selection

Given a limited number of labeled examples $\mathcal{D}$ and the algorithm ensemble $\mathcal{A}$, we will first acquire a set of classification models $\mathcal{M} = (\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_S)$ by running each algorithm in $\mathcal{A}$ over $\mathcal{D}$. To identify the most informative example $\mathbf{x}$ from the pool of unlabeled examples, we follow the idea of the Hedge algorithm. In particular, since the classification error of the weighted model ensemble is upper bounded by the classification error of the best classification model within the ensemble, we will choose the unlabeled example that maximizes the weighted loss function L_β. To this end, we employ the MaxMin approach by selecting the unlabeled example $\mathbf{x}$ that can greatly increase L_β no matter which class label y is assigned to $\mathbf{x}$. This can be formulated as the following optimization problem:

$$\begin{aligned}\mathbf{x}^* &= \arg\max_{\mathbf{x}} \min_{y\in\{-1,+1\}} L_\beta(\mathcal{D}\cup(\mathbf{x},y)) \\ &\approx \arg\max_{\mathbf{x}} \min_{y\in\{-1,+1\}} \sum_{i=1}^{S} w_i^t l(y,\mathbf{m}_i(\mathbf{x})) \quad (3)\end{aligned}$$

If we view each weight w_i^t as the probability of choosing classification model $\mathbf{m}_i$, we can interpret $\sum_{i=1}^{S} w_i^t l(y,\mathbf{m}_i(\mathbf{x}))$ as $1 - p(y|\mathbf{x},\mathcal{M})$. Here, $p(y|\mathbf{x},\mathcal{M})$ is defined as $\sum_{i=1}^{S} w_i^t(1 - l(y,\mathbf{m}_i(\mathbf{x})))$, and can be interpreted as the probability of classifying the example $\mathbf{x}$ into class y by the ensemble $\mathcal{M}$. Thus, the criterion in (3) can also be written as:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \max_{y\in\{-1,+1\}} p(y|\mathbf{x},\mathcal{M}) \quad (4)$$

It is interesting to see that the above criterion of example selection is consistent with the idea of selecting the example with the largest classification uncertainty. This is because we can rewrite $\max_{y\in\{-1,+1\}} p(y|\mathbf{x},\mathcal{M})$ as

$$\begin{aligned}\max_{y\in\{-1,+1\}} p(y|\mathbf{x},\mathcal{M}) &= \max\left(\frac{1}{2}+\Delta(\mathbf{x}), \frac{1}{2}-\Delta(\mathbf{x})\right) \\ &= \frac{1}{2}+|\Delta(\mathbf{x})|\end{aligned}$$

where $\Delta(\mathbf{x}) = 1/2 - p(+1|\mathbf{x},\mathcal{M})$. Hence, the optimization problem in (4) is equivalent to the following problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left|\frac{1}{2} - p(+1|\mathbf{x},\mathcal{M})\right| \quad (5)$$

In other words, the MaxMin approach chooses the example with classification probability $p(y|\mathbf{x},\mathcal{M})$ closest to $1/2$, which is the most uncertain example to classify by the ensemble $\mathcal{M}$.

Incorporating Example Correlation Information

One problem with the above formalism is that each example is treated independently, and as a result, the impact of each example $\mathbf{x}$ to the weighted loss function L_β is limited to the example itself. As pointed out by (Roy & McCallum 2001), it is important to explore correlation information among unlabeled examples in active learning, particularly when the goal is to find the model that achieves the best classification accuracy for a given dataset. Let $\mathcal{U} = (\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_{N_u})$ denote the set of unlabeled examples. To explore the correlation among examples, we introduce the matrix of pairwise correlation $P = [P_{i,j}]_{N_u\times N_u}$, where each element $P_{i,j}$ in the matrix is defined as the probability for both $\mathbf{x}'_i$ and $\mathbf{x}'_j$ to be assigned to the same class. Given the pairwise correlation information, the change to the weighted loss function made by the additional example $(\mathbf{x}_i, y)$, denoted by $L_\beta(\mathbf{x}_i, y)$, can be calculated as follows:

$$\begin{aligned}L_\beta(\mathbf{x}_i,y) &= L_\beta(\mathcal{D}\cup(\mathbf{x}_i,y)) - L_\beta(\mathcal{D}) \\ &= n - p(y|\mathbf{x}_i,\mathcal{M}) - \sum_{j\neq i}[P_{i,j}p(y|\mathbf{x}_j,\mathcal{M}) \\ &\quad +(1-P_{i,j})p(-y|\mathbf{x}_j,\mathcal{M})]\end{aligned}$$

Thus, we can rewrite the optimization problem in (3) as follows:

$$\begin{aligned}\mathbf{x}^* = \arg\min_{\mathbf{x}_i\in\mathcal{U}} \max_{y\in(-1,+1)} & p(y|\mathbf{x}_i,\mathcal{M}) + \\ & \sum_{j\neq i}[P_{i,j}p(y|\mathbf{x}_i,\mathcal{M}) + (1-P_{i,j})p(-y|\mathbf{x}_j,\mathcal{M})] \quad (6)\end{aligned}$$

It is not difficult to see that the above optimization problem becomes the problem in (4) when the correlation $P_{i,j} = 1/2, \forall j \neq i$. Note that the expression $P_{i,j}p(y|\mathbf{x}_i,\mathcal{M}) + (1-P_{i,j})p(-y|\mathbf{x}_j,\mathcal{M})$ can be interpreted as the consistency between the correlation information $P_{i,j}$ and the

classification probability $p(y|\mathbf{x}_j, \mathcal{M})$. Thus, by minimizing the second term in the objective function in (6), i.e., $\sum_{j\neq i}(P_{i,j}p(y|\mathbf{x}_i, M) + (1 - P_{i,j})p(-y|\mathbf{x}_j, M))$, we find the example with class label being inconsistent with the example correlation. Hence, according to (6), the chosen example $\mathbf{x}_i$ on one hand should be uncertain to classify, and on the other hand should be informative to the prediction of other examples.

Finally, the correlation matrix can be calculated using the prediction made by the models in the ensemble $\mathcal{M}$. We assume that two examples are strongly correlated when most classification models in $\mathcal{M}$ make the same prediction for them. Thus, we can compute the correlation between two examples based on the overlapping in their class labels that are predicted by models in $\mathcal{M}$. Specifically, if $\mathbf{t}_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,S}) \in \{-1, +1\}^{N_u}$ denotes the class predictions for the ith unlabeled example by all the classification models in $\mathcal{M}$, we can calculate $P_{i,j}$ as

$$P_{i,j} = \frac{\mathbf{t}_i^T \mathbf{t}_j + S + 2C}{2S + 2C}$$

where C is a smoothing constant determined empirically.

Determining Combination Weights w

The key parameter to the general framework described above is the combination weights $\mathbf{w} = (w_1, w_2, \ldots, w_S)$ for different classification models. One way to determine $\mathbf{w}$ is to use the Hedge algorithm directly. In particular, we can use Eqn. (1) to update the weights. However, the problem with using Eqn. (1) is that the same punishment factor β is used for every labeled example. Intuitively, the weights should be adjusted significantly (small β) if the example is very informative, and adjusted slightly (large β) if the example is not so informative. To this end, we follow the idea of Maximum Entropy Discrimination (Jaakkola, Meila, & Jebara 1999), and present an algorithm that determines the punishment factor β for an example $\mathbf{x}$ according to the "difficulty" in classifying $\mathbf{x}$. In particular, given the set of labeled examples $\{(\mathbf{x}_t, y_t)\}_{t=1}^T$, we determine $\mathbf{w}$ by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbf{R}^S} \quad & \sum_{i=1}^S w_i \log w_i + \gamma \sum_{t=1}^T \epsilon_t \\ \text{s. t.} \quad & y_t \left(\sum_{i=1}^S w_i \mathbf{m}_i(\mathbf{x}_t) \right) \geq 1 - \epsilon_t,\ \epsilon_t \geq 0, \\ & \qquad t = 1, 2, \ldots, T \\ & \sum_{i=1}^S w_i = 1,\ w_i \geq 0,\ , i = 1, 2, \ldots, S \end{aligned} \tag{7}$$

where parameter γ weights the importance of the classification error ϵs against the uniformity of weights $\mathbf{w}$. Above we enforce the sum of weights to be one so that the weights can be interpreted as probabilities of using individual models for classifying examples. According to the above formalism, on one hand, by maximizing the entropy of the weights, the algorithm will choose weights that are close to the uniform distribution; on the other hand, by satisfying the constraints, the above algorithm tends to assign larger weights to classification models that make smaller numbers of errors.

We can further convert the above problem into its dual form, which is expressed as:

$$\begin{aligned} \min_{\lambda\in\mathbf{R}^T} \quad & \log \left(\sum_{i=1}^S \exp \left(\sum_{t=1}^T \lambda_t y_t \mathbf{m}_i(\mathbf{x}_t) \right) \right) - \sum_{t=1}^T \lambda_t \\ \text{s. t.} \quad & 0 \leq \lambda_t \leq \gamma,\ t = 1, 2, \ldots, T. \end{aligned} \tag{8}$$

Given the dual variables λ, the combination weights $\mathbf{w}$ can be calculated as follows:

$$w_i = \frac{1}{Z} \exp \left(\sum_{t=1}^T \lambda_t y_t \mathbf{m}_i(\mathbf{x}_t) \right),\ i = 1, 2, \ldots, S \tag{9}$$

where $Z = \sum_{i=1}^S \exp\left(\sum_{t=1}^T \lambda_t y_t \mathbf{m}_i(\mathbf{x}_t)\right)$ is a normalization factor. The dual problem (8) is a convex programming problem and can be solved efficiently using Newton's method. It is interesting to see the relationship between expression of w_i in (9) and that in (1). First we can rewrite the above expression in an inductive way, i.e.,

$$\begin{aligned} w_i^{t+1} &\leftarrow \frac{w_i^t \exp\left(\lambda_t y_t \mathbf{m}_i(\mathbf{x}_t)\right)}{\sum_{j=1}^S w_j^t \exp\left(\lambda_t y_t \mathbf{m}_j(\mathbf{x}_t)\right)} \\ &= \frac{w_i^t \beta_t^{2l(y_t, \mathbf{m}_i(\mathbf{x}_t))-1}}{\sum_{j=1}^S w_j^t \beta_t^{2l(y_t, \mathbf{m}_j(\mathbf{x}_t))-1}} \end{aligned} \tag{10}$$

Here we define $\beta_t = \exp(-\lambda_t)$ and use the relation $y_t \mathbf{m}_i(\mathbf{x}_t) = 1 - 2l(y_t, \mathbf{m}_i(\mathbf{x}_t))$. Comparing Eqn. (10) to (1), the key difference is that, instead of using the same punishment factor for every example, a different punishment factor β_t is used for each example. Note that the modified algorithm still keeps the great property of the Hedge algorithm, that the weighted cumulative loss of the ensemble is bounded by a function of the best algorithm's loss.

According to the above algorithm, λ_t will take a smaller value when $\mathbf{x}_t$ is an easier example, i.e., most of the candidates in the ensemble make correct predictions on x_t; hence $\beta_t = \exp(-\lambda_t)$ will be larger. On the contrary, a more difficult example x_t yields a larger λ_t and a smaller β_t. From Eqn. (10), an algorithm's weight will be discounted more significantly (multiplied by a smaller β_t) when it makes incorrect prediction on a more difficult (and thus more informative) example than on an easier one (larger β_t). The proposed active algorithm selection method always selects the most informative example for labeling. Hence the earlier selected examples are more informative, and the weights are adjusted more significantly; the later examples are less informative, and smaller adjustment are applied to the weights. As the result, on one hand, the weights are adjusted substantially during the early learning stage; and on the other hand, undesirable fluctuations of the weights are avoided after sufficient information has been learned.

Experiments and Results

In this section, we report our empirical study of active algorithm selection in the application to three different domains: drug discovery, marketing and digit recognition.

Table 1: Summary of the datasets

Dataset	ADA	HIVA	GINA
# of labeled instance	4147	3845	3153
# of unlabeled instance	41471	38449	31532
# of features	48	1617	970

Table 2: Classification errors of algorithm candidates(%)

Algorithm	ADA	HIVA	GINA
A1	21.6 ± 0.8	3.8 ± 0.1	17.7 ± 1.5
A2	24.6 ± 3.1	10.7 ± 1.6	31.6 ± 2.4
A3	24.5 ± 1.7	5.7 ± 0.5	11.1 ± 0.3
A4	25.7 ± 2.0	13.2 ± 1.4	19.3 ± 2.4
A5	17.6 ± 1.0	5.4 ± 0.7	23.5 ± 2.4
A6	21.5 ± 1.2	13.4 ± 0.3	19.9 ± 1.3
A7	22.6 ± 2.4	14.3 ± 0.7	20.6 ± 1.2
A8	17.5 ± 0.6	14.3 ± 0.5	22.5 ± 2
A9	22.9 ± 1.1	12.4 ± 0.4	20.6 ± 2.4
A10	14.3 ± 0.3	10.1 ± 0.4	25.1 ± 1.8

Testbeds

The testbeds used in our study come from the WCCI 2006 Performance Prediction Challenge[1] (WCCI PCP), which provides five large datasets. In this study, we selected datasets HIVA (drug discovery), ADA (marketing) and GINA (digit recognition) as our testbeds, which are summarized in Table 1.

In reality, given a dataset and an ensemble of machine learning algorithms, we may run into two different conditions: there is only one optimal candidate or there are multiple optimal candidates.In this study, we focus on the simple scenario where there is only one optimal algorithm that outperforms the others. For each dataset, we created an ensemble of 10 learning algorithms to be selected. We use the WEKA[2] implementation of the algorithms. The algorithms were carefully selected such that for every dataset, the optimal algorithm outperformed the others significantly. It is important to note that the optimal algorithm may differ from one dataset to another. It is important to note that the optimal algorithm may differ from one dataset to another. Table 2 summarizes the classification errors of all ten learning algorithms that are computed using five-fold cross validations. It is clear that the optimal algorithms for the three datasets are A10, A1, and A3, respectively.

Empirical Evaluation

To evaluate the proposed active algorithm selection methods, we first randomly selected 20 labeled examples as the initial training set, on which we built 10 models. Then for each iteration, one more example was selected, labeled, and added to the training pool. The total number of iterations was 300 for all the experiments. Three variants of the proposed algorithm selection methods were examined in this study. "**Hedge**" refers to the Hedge algorithm in Eqn. (1) with fixed $\beta = 0.7$ (chosen according to our experience) without utilizing example correlation. "**Hedge+Correlation**" refers to the proposed method using both the Hedge algorithm ($\beta = 0.7$) and example correlation. "**MED+Correlation**" refers to the proposed method automatically determining weights and exploring example correlation. In addition, two baseline methods for algorithm selection were used. One randomly selects one example for labeling ("**Random Selection**"), and the other treats each learning algorithm equally and selects the example that holds the largest disagreement in predictions made by the ensemble ("**Majority Vote**").

For performance measurement, we adopt two evaluation metrics. The first is the number of examples required for the optimal algorithm to win over the other algorithms. To this end, we employ the Hedge algorithm with $\beta = 0.7$ to compute the weight assigned to each algorithm given the number of misclassified examples. We then compute the minimum number of examples that is required for the weight of the optimal algorithm to reach a predefined threshold. We refer to the minimum number of examples as "**critical number**". Evidently, the smaller the critical number, the more efficient the algorithm selection method. The threshold is set to be 0.9. We chose a large threshold because it would guarantee that the selected algorithm was significantly better than others, thus reliably identifying the optimal. The second evaluation metric is classification accuracy of the optimal algorithm identified by the selection method. We record during each iteration the classification accuracy of each model. The assumption is that a good algorithm selection method will yield higher classification accuracy than a bad one, given the same number of labeled examples. All experiments were repeated 10 times and averaged results were reported.

Results

Table 3 summarizes the critical numbers of the five methods for algorithm selection. Table 4 shows the accuracy of the best algorithm selected after 300 iterations. First, we observe that the three algorithms (i.e., Random Selection, Majority Vote and Hedge) that do not explore the example correlation perform significantly worse than the other two (i.e., Hedge+Corr and MED+Corr) that do so. This is illustrated both by the critical numbers and by the accuracy. Second, the MED+Corr algorithm, which allows the combination weights to be determined by "classification difficulty" of each example, shows further improvement according to both the critical number and accuracy.

To further understand why MED+Corr performed better than Hedge+Corr, we carefully examined the punishment factors β_t used in Med+Corr. We observed that the punishment factor β_t changed over iteration. In particular, comparing to the the average β_t for the first 200 iterations, the average β_t for the last 100 iterations was 7 percent larger, and the average β_t for the last 50 iterations was 15 percent larger. The observation is consistent with the discussion in the previous section, and shows that MED+Corr achieves a better performance by adjusting β_t according to the "classification difficulty" of selected examples, while Hedge+Corre uses a constant punishment factor.

Some may question why we choose the winner-take-all strategy instead of using the readily available ensemble. One

[1] http://www.modelselect.inf.ethz.ch

[2] http://www.cs.waikato.ac.nz/ml/weka

Table 3: The critical number for the five selection methods. The smaller the critical number the better the performance

Algorithm	ADA	HIVA	GINA
Random Sel.	338 ± 108	262 ± 93	207 ± 81
Majority Vote	259 ± 127	199 ± 57	143 ± 45
Hedge	289 ± 31	253 ± 76	146 ± 32
Hedge+Corr	181 ± 65	169 ± 51	120 ± 26
MED+Corr	117 ± 46	129 ± 37	103 ± 35

Table 4: The classification accuracy of the selected algorithm by the five algorithm selection methods(%)

Algorithm	ADA	HIVA	GINA
Random Sel.	82.4 ± 1.2	84.6 ± 1.7	85.2 ± 2.4
Majority Vote	83.4 ± 1.1	89.8 ± 1.3	86.5 ± 2.1
Hedge	83.6 ± 1.4	88.8 ± 1.1	86.2 ± 1.6
Hedge+Corr	84.7 ± 1.2	91.2 ± 1.3	88.2 ± 1.2
MED+Corr	86.5 ± 1.1	94.6 ± 0.6	89.7 ± 0.8

reason of prefering the optimal algorithm to an ensemble is that it can significantly reduce the training cost. This is particularly important when the number of training examples is very large. Furthermore, we have evaluated the performance of the ensemble as a whole comparing to the optimal algorithm selected by the proposed Hedge+Corr and MED+Corr methods. We have evaluated both the the ensemble weighted by $\mathbf{w}$ (referred to as $E1$) and the equally weighted ensemble (referred to as $E2$). Figure 1 shows a sample plot of the classification accuracy, produced by the MED+Corr method on ADA dataset. It is not surprising to observe that with a very small number of labeled examples, the ensemble approaches outperformed the optimal algorithm selected by the proposed method. However, with a modest number of labeled examples (only 300 examples), the optimal algorithm achieved similar performance as both ensemble approaches. This indicates that the winner-take-all approach is able to reach similar performance as the ensemble approach.

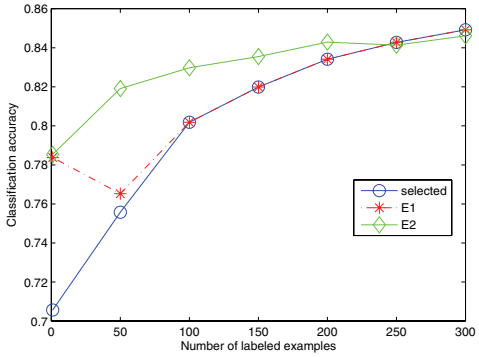

Figure 1: Classification accuracy: the optimal algorithm vs. ensemble approaches. $E1$ is the ensemble weighted by $\mathbf{w}$, while $E2$ is the equally weighted ensemble. Produced by MED+Corr on ADA dataset.

Conclusion

In this paper, we investigate the problem of active algorithm selection. Given a dataset and a set of learning algorithms, the goal of active algorithm selection is to identify the best learning algorithm among the given ones using the least number of labeled examples. We present a general framework for active algorithm selection based on the Hedge algorithm and the MaxMin principle. We further extend the framework by incorporating the example correlation information and by automatically determining the combination weights. Our empirical studies with the datasets of WCCI 2006 performance prediction challenge have shown promising performance of the proposed algorithm in efficiently identifying the best learning algorithm for given datasets.

In the future, we will evaluate how the proposed active algorithm selection methods will generalize to the case that the difference in classification performance between the best and second best classifier is varied, and the case that there are more than one optimal candidate in the ensemble.

References

Abe, N., and Mamitsuka, H. 1998. Query learning strategies using boosting and bagging. In *ICML*, 1–9.

Campbell, C.; Cristianini, N.; and Smola, A. 2000. Query learning with large margin classifiers. In *ICML*, 111–118.

Freund, Y., and Schapire, R. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences* 55:119–139.

Freund, Y.; H.Seung; Shamir, E.; and Tishby, N. 1997. Selective sampling using the query by committee algorithm. *Mach. Learn.* 28(2-3):133–168.

Jaakkola, T.; Meila, M.; and Jebara, T. 1999. Maximum entropy discrimination. Technical Report AITR-1668, MIT, Artificial Intelligence Laboratory.

Lewis, D., and Gale, W. 1994. A sequential algorithm for training text classifiers. In *SIGIR*, 3–12.

Liere, R., and Tadepalli, P. 1997. Active learning with committees for text categorization. In *AAAI*, 591–596.

MacKay, D. 1992. Information-based objective functions for active data selection. *Neural Comp.* 4(4):590–604.

Roy, N., and McCallum, A. 2001. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 441–448.

Saar-Tsechansky, M., and Provost, F. 2004. Active sampling for class probability estimation and ranking. *Mach. Learn.* 54(2):153–178.

Schohn, G., and Cohn, D. 2000. Less is more: Active learning with support vector machines. In *ICML*, 839–846.

Seung, H.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *Com. Learning Theory*, 287–294.

Thompson, C. A.; Califf, M. E.; and Mooney, R. J. 1999. Active learning for natural language parsing and information extraction. In *ICML*.

Tong, S., and Koller, D. 2000. Support vector machine active learning with applications to text classification. In *ICML*, 999–1006.

Zhang, T., and Oles, F. 2000. A probability analysis on the value of unlabeled data for classification problems. In *ICML*.