# Improved Algorithms for Confidence-Rated Prediction with Error Guarantees

**Kamalika Chaudhuri** **Chicheng Zhang**
Department of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, 92093-0404
{kamalika, chz038}@cs.ucsd.edu

## Abstract

We study confidence-rated prediction in a binary classification setting, where the goal is to design a predictor that can choose to abstain from prediction on test examples. Such predictors can be used to determine which data points are *easy* to classify. The performance of a confidence-rated predictor is measured by its error, or misclassification rate, and its coverage, or the fraction of examples on which it does not abstain. Typically, there is a tradeoff between these two metrics, and the goal is to design predictors that have good error-coverage tradeoffs. We provide an algorithm in the transductive setting that gives a predictor with guaranteed upper bound on the error. Our algorithm has optimal coverage in the realizable case, and can be extended to the agnostic setting. While our algorithm is computationally inefficient in general, we show how to implement an approximate version, and evaluate its performance on several real datasets.

## 1 Introduction

We study confidence-rated prediction in a binary classification setting. In this problem, we are given training examples labelled $-1$ or $1$, and our goal is to design a classifier, which, given a test example, can either choose to predict a label in $\{-1, 1\}$, or to abstain from prediction by outputting $0$. Such predictors can be used to determine which data points are *easy* to classify, and are useful in applications such as medical diagnosis and credit card fraud detection where classification mistakes are costly.

The performance of a confidence-rated predictor is measured by two parameters – the error, or the fraction of examples on which the predictor outputs the wrong label, and the coverage, or the fraction of examples on which the predictor does not abstain. As the error of a predictor typically grows with growing coverage, there is a tradeoff between the error and the coverage, and the goal in confidence-rated prediction is to develop predictors that have improved error-coverage tradeoffs.

In this paper, we address the task of designing confidence-rated predictors which provide a guaranteed upper bound on the error. In the realizable case, it is possible to provide error guarantees with respect to the true labels based on training data. In the non-realizable case, errors may arise due to inherent label noise, and it is impossible to provide strong error guarantees with respect to the true labels without strong assumptions. Following [1], we therefore consider a different kind of error guarantee – error with respect to the best hypothesis in a hypothesis class $\mathcal{H}$.

While there are several existing models of confidence in prediction [2, 3, 4], we consider the recent learning-theoretic formalization due to [5]. The state-of-the-art in this framework is due to [5] and [1]. [5] provides a predictor which achieves zero error in the realizable case by abstaining in the disagreement region of the version space; to guarantee an error $\delta > 0$, it predicts with an arbitrary classifier in the version space with some probability, and abstains otherwise. [1] extends the results

of [5] to the non-realizable case by providing an algorithm which has guaranteed zero error with respect to the best hypothesis in the hypothesis class. It can be shown that the algorithm of [5] has suboptimal coverage for a number of classification problems, and a natural question is whether one can achieve higher coverage while still ensuring a guaranteed upper bound on the error of the classifier, and what kind of algorithms will result in such high coverage.

In this paper, we provide an algorithm in the transductive setting, which given a set of labelled and unlabelled samples drawn from a data distribution, finds a confidence-rated predictor with guaranteed error $\delta$ on the unlabelled samples. We show that in the realizable case, our algorithm is *optimal*, in the sense that any other algorithm that guarantees error $\delta$ given the input labelled samples will necessarily have equal or lower coverage. We then show how to apply our algorithm in the agnostic setting, when its error with respect to the best hypothesis in the class $\mathcal{H}$ is at most $\delta$.

While our algorithm is computationally inefficient in general, we show how to implement an approximate version of our algorithm, and one of its variants efficiently through bootstrap sampling from the version space. The approximate version has error guarantees with respect to the bootstrapped subsample. We evaluate these algorithms through two tasks – comparing the coverage as a function of the error guarantee, and measuring the actual risk (or error to coverage ratio) with respect to the test labels. We show that our algorithm outperforms the algorithm in [5], and achieves error-coverage tradeoff competitive with that of the algorithms in [1] and [6, 7], which do not have error guarantees.

## 2 Algorithms

We study binary classification in the *transductive setting*. We are given a set $S$ of labelled examples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where each $x_i \in \mathcal{X}$ and each $y_i \in \{-1, 1\}$. We are also given a set $U = \{x_{n+1}, \ldots, x_{n+m}\}$ of $m$ unlabelled examples.

**Confidence-Rated Predictors and Selective Classifiers.** A confidence-rated predictor $P$ is a mapping from $U$ to a set of $m$ distributions over $\{-1, 0, 1\}$. If the $j$-th distribution is $[p_{-1}, p_0, p_1]$, then $P(x_{n+j}) = -1$ wp $p_{-1}$, 0 wp $p_0$ and 1 wp $p_1$. A selective classifier $C$ is a tuple $(h, (\gamma_1, \ldots, \gamma_m))$, where $h$ lies in a hypothesis class $\mathcal{H}$, and $0 \leq \gamma_i \leq 1$ for all $i = 1, \ldots, m$. For any $x_{n+j} \in U$, $C(x_{n+j}) = h(x_{n+j})$ wp $\gamma_j$ and 0 wp $1 - \gamma_j$.

**Coverage.** The *coverage* $\mathbf{cov}(P)$ of a confidence-rated predictor $P$ is the probability that $P$ predicts a label (that is, does not predict 0) wrt the uniform distribution over $U$.

**Error.** The *error* $\mathbf{err}(P)$ *of a confidence-rated predictor* $P$ *wrt the true labels* is the probability that $P$ predicts 1 when the true label is $-1$ and vice versa, wrt the uniform distribution over $U$. Let $\mathcal{H}$ be a hypothesis class, and let $h^*$ be the true error minimizer in $\mathcal{H}$ wrt data distribution $D$. Then, the *error* $\mathbf{err}_{\mathcal{H}}(P)$ *of a confidence-rated predictor* $P$ *wrt the best hypothesis in* $\mathcal{H}$ is the probability that $P$ predicts 1 when $h^*(x) = -1$ and vice versa. We also define the *risk* of a confidence-rated predictor as the ratio $\mathbf{err}(P)/\mathbf{cov}(P)$ (in both the realizable and the non-realizable case).

**Version Space.** In the realizable setting, given a set of labelled examples $S$, and a hypothesis class $\mathcal{H}$, the version space of $S$ is the subset of classifiers in $\mathcal{H}$ that are consistent with $S$.

**Disagreement Region.** Let $H \subseteq \mathcal{H}$ be a set of hypotheses. The disagreement region of $H$, denoted by $\mathbb{DIS}(H)$, is the set of all examples $x \in \mathcal{X}$ for which there exist two hypotheses $h_1, h_2 \in H$ such that $h_1(x) \neq h_2(x)$. More formally, $\mathbb{DIS}(H) = \{x \in \mathcal{X} : \exists h_1, h_2 \in H \text{ such that } h_1(x) \neq h_2(x)\}$.

We now present two algorithms for confidence-rated prediction in the transductive setting – a confidence-rated predictor in Algorithm 1, and a selective classifier in Algorithm 2. We state both algorithms for the realizable case, and then discuss how to translate them to the non-realizable setting.

Given a training set $S$ and an unlabelled dataset $U$, Algorithm 1 first constructs the version space $V$ of $S$ with respect to the hypothesis class $\mathcal{H}$. Our key observation is that once this version space has been constructed, finding the optimal coverage confidence-rated predictor which has guaranteed error $\leq \delta$ can be expressed as a *linear program*. A similar observation can be used to construct a selective classifier; we present this construction in Algorithm 2.

**Algorithm 1** Confidence-rated Predictor

1: **Inputs:** labelled data $S$, unlabelled data $U$, error bound $\delta$.
2: Compute version space $V$ with respect to $S$.
3: Solve the linear program:

$$\max \sum_{i=1}^{m}(\alpha_i + \beta_i)$$

subject to:

$$\forall i,\ \alpha_i + \beta_i \leq 1 \quad (1)$$

$$\forall h \in V,\ \sum_{i:h(x_{n+i})=1} \beta_i + \sum_{i:h(x_{n+i})=-1} \alpha_i \leq \delta m \quad (2)$$

$$\forall i,\ \alpha_i, \beta_i \geq 0 \quad (3)$$

4: Output the confidence-rated predictor: $\{[\beta_i, 1-\alpha_i-\beta_i, \alpha_i], i = 1, \ldots, m\}$.

**Algorithm 2** Selective Classifier

1: **Inputs:** labelled data $S$, unlabelled data $U$, error bound $\delta$.
2: Compute version space $V$ with respect to $S$. Pick an arbitrary $h_0 \in V$.
3: Solve the linear program:

$$\max \sum_{i=1}^{m}\gamma_i$$

subject to:

$$\forall i,\ \gamma_i \leq 1 \quad (4)$$

$$\forall h \in V,\ \sum_{i:h(x_{n+i})\neq h_0(x_{n+i})} \gamma_i \leq \delta m \quad (5)$$

$$\forall i,\ \gamma_i \geq 0 \quad (6)$$

4: Output the selective classifier: $(h_0, (\gamma_1, \ldots, \gamma_m))$.

**Performance Guarantees and the Non-Realizable Case** Algorithms 1 and 2 have the following performance guarantees.

**Theorem 1** *Let $P$ be the confidence-rated predictor output by Algorithm 1 on inputs $S$, $U$ and $\delta$ in the realizable setting. Then, $\mathbf{err}(P) \leq \delta$. Moreover, if $P'$ is any other confidence-rated predictor that guarantees $\mathbf{err}(P') \leq \delta$ given $S$ and $U$, then $\mathbf{cov}(P') \leq \mathbf{cov}(P)$.*

**Theorem 2** *Let $C$ be the selective classifier output by Algorithm 2 on inputs $S$, $U$ and $\delta$ in the realizable case where $h_0$ is arbitrarily chosen in $V$. Then, $\mathbf{err}(C) \leq \delta$. Moreover, $\mathbf{cov}(C) \geq \mathbf{cov}(P) - \delta$, where $P$ is the predictor output by Algorithm 1 on input $S$, $U$ and $\delta$.*

In the non-realizable case, to ensure guaranteed error, we use instead a subset of $\mathcal{H}$ that is very likely to include the true error minimizer $h^*$.

Given a sample set $S$, a set $C(S) \subseteq \mathcal{H}$ is called a *level* $1-\delta_0$*-confidence set* if for all data distributions $D$, $\Pr_{S\sim D^n}[h^*(D) \in C(S)] \geq 1-\delta_0$, where $h^*(D)$ is a hypothesis in $\mathcal{H}$ that minimizes the expected classification error according to $D$. If we replace $V$ in Algorithms 1 and 2 by a level $1-\delta_0$-confidence set $C(S)$, then we can show that the resulting predictor provides an error guarantee with probability $\geq 1-\delta_0$.

If $\hat{h}$ is the hypothesis that minimizes the empirical error on the training data, then, the following set $\hat{\mathcal{V}}(\hat{h})$, used by [1], is a $1-\delta_0$-level confidence set: $\hat{\mathcal{V}}(\hat{h}) = \{h \in \mathcal{H} | \hat{\mathbf{err}}(h) \leq \hat{\mathbf{err}}(\hat{h}) + 2\sigma(n, \delta_0)\}$. Here $\hat{\mathbf{err}}(h)$ is the empirical error of the hypothesis $h$ on the training set $S$. $\sigma(n, \delta_0)$ is a function of the training set size $n$, the hypothesis class $\mathcal{H}$, a parameter $\delta_0$, which ensures that with probability $\geq 1-\delta_0$, for all $h \in \mathcal{H}$, $|\hat{\mathbf{err}}(h) - \mathbf{err}(h)| \leq \sigma(n, \delta_0)$. The expressions for $\sigma(n, \delta_0)$ can be obtained from standard generalization bounds for classification.

Depending on the hypothesis class, $\hat{\mathcal{V}}(\hat{h})$ may have complex structure and may even be disconnected. To address this, active learning literature [8] uses instead the set $\hat{B}(\hat{h}, 2\hat{\mathbf{err}}(\hat{h}) + 2\sigma(n, \delta_0))$, the empirical disagreement ball around $\hat{h}$ of radius $2\hat{\mathbf{err}}(\hat{h}) + 2\sigma(n, \delta_0)$. Since $\hat{\mathcal{V}}(\hat{h}) \subseteq \hat{B}(\hat{h}, 2\hat{\mathbf{err}}(\hat{h}) + 2\sigma(n, \delta_0))$, this process preserves the error guarantees, and results in a smaller coverage.

## 3 Implementation and Experiments

**Implementation.** The LPs in Algorithms 1 and 2 have a constraint for each hypothesis in the version space; to implement them, we draw samples to approximate the version space by a finite hypothesis set $H$, and use constraints corresponding to the hypotheses in $H$. In the realizable case, we sample from the convex version space $V$ using the Hit and Run Markov Chain [9, 10]. In non-realizable case, we set $\hat{h}$ as the SVM solution and sample from the star-shaped body $\hat{B}(\hat{h}, 2\hat{\mathbf{err}}(\hat{h}) + C\sigma(n, \delta_0)), C = 0.2$ using a ball walk[11]. In each case, we run the Markov Chain until $t = 100000$, and randomly select 1000 classifiers from the trajectory.

The linear programs in Algorithms 1 and 2 tend to have multiple optimal solutions for the same value of $\delta$; we break ties among these solutions by selecting the one which has the best *alignment* with the SVM solution. To do this, first we solve the original LP for a given $\delta$ to get an optimal coverage value $C(\delta)$. Next, we add an additional linear constraint to the original LP to enforce that the coverage is equal to $C(\delta)$ and select the solution that maximizes, under these constraints, the quantity $\sum_{i=1}^{m}(\alpha_i - \beta_i)\langle w_0, x_i\rangle$ for Algorithm 1 and the quantity $\sum_{i=1}^{m}\gamma_i|\langle w_0, x_i\rangle|$ for Algorithm 2, where $w_0$ is the SVM solution vector.

**Risk-Coverage Tradeoffs.** We evaluate the actual risk-coverage tradeoffs achieved by Algorithms 1 and 2 on real data. For comparison, we choose the algorithm in [5] (EYW10), the Agnostic Selective Classification (ASC) algorithm [1] and thresholding based on the distance from the decision boundary (DDB) of the SVM classifier. ASC sorts unlabelled examples based on a *disbelief index* and abstains whenever this index is below a threshold. DDB abstains from prediction when the distance from the decision boundary of an SVM classifer is below a threshold. Each algorithm has a parameter which can be varied to control the error-coverage tradeoff; we run several iterations of each algorithm with different values of these parameters, and plot the corresponding risk (as measured with respect to the actual test labels) as a function of the coverage. We observe that DDB does not offer any error guarantees, and ASC only has theoretical guarantees for zero error.

Figures 1 to 6 show the results; the datasets used are KDDCup99 (normal vs. malicious connections), MNIST, and Breast Cancer from the UCI repository. Each plotted point is an average over 20 rounds of random selection of training and test sets with error bars at one standard deviation. EYW10 performs the worst, as it treats all points in the disagreement region as equivalent. The performance of Algorithms 1 and 2 are competitive with ASC, which usually performs the same as or better than DDB. This is to be expected as Algorithms 1 and 2 are more conservative. Interestingly, Algorithm 2 usually performs better than Algorithm 1 in practice, even though it is worse in theory. This may be because Algorithm 1 treats all hypotheses in the version space the same way, and generates the predicted labels by solving an LP; while the labels predicted by Algorithm 2 always agree with the SVM solution, and as seen from the results on DDB, these predictions work quite well in practice.

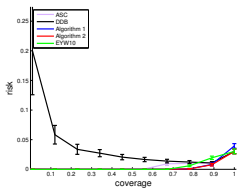


Figure 1: kddcup

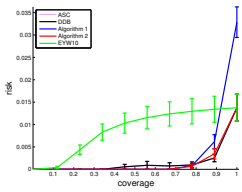


Figure 2: mnist 0v1

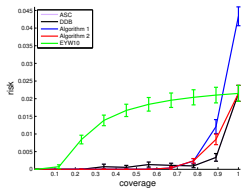


Figure 3: mnist 6v9

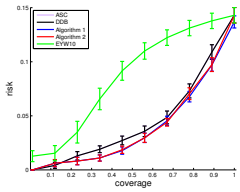


Figure 4: mnist 3v5

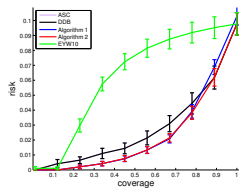


Figure 5: mnist 2v3

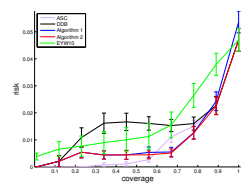


Figure 6: breast

## References


[1] R. El-Yaniv and Y. Wiener. Agnostic selective classification. In *NIPS*, 2011.

[2] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

[3] Y. Freund, Y. Mansour, and R. E. Schapire. Generalization bounds for averaged classifiers. *The Ann. of Stat.*, 32, 2004.

[4] L. Li, M. L. Littman, and T. J. Walsh. Knows what it knows: a framework for self-aware learning. In *ICML*, 2008.

[5] R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *JMLR*, 11, 2010.

[6] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3), 2002.

[7] S. Mukherjee. Chapter 9. classifying microarray data using support vector machines. In *of scient. from the Univ. Penn. Sch. of Med. and the Sch. of EAS*. Kluwer Academic Publishers, 2003.

[8] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *NIPS*, 2007.

[9] R. Gilad-Bachrach, A. Navot, and N. Tishby. Query by committee made real. In *NIPS*, 2005.

[10] L. Lovász and S. Vempala. Hit-and-run from a corner. *SIAM J. Comput.*, 35(4), 2006.

[11] K. Chandrasekaran, D. Dadush, and S. Vempala. Thin partitions: Isoperimetric inequalities and a sampling algorithm for star shaped bodies. In *SODA*, 2010.