Conference on Artificial Intelligence, AAAI-2002 (Edmonton, Canada) pp. 140–145.

Progressive Rademacher Sampling

Tapio Elomaa
Department of Computer Science
P. O. Box 26 (Teollisuuskatu 23)
FIN-00014 Univ. of Helsinki, Finland
elomaa@cs.helsinki.fi

Matti Kääriäinen
Department of Computer Science
P. O. Box 26 (Teollisuuskatu 23)
FIN-00014 Univ. of Helsinki, Finland
matti.kaariainen@cs.helsinki.fi

Abstract

Sampling can enhance processing of large training example databases, but without knowing all of the data, or the example producing process, it is impossible to know in advance what size of a sample to choose in order to guarantee good performance. Progressive sampling has been suggested to circumvent this problem. The idea in it is to increase the sample size according to some schedule until accuracy close to that which would be obtained using all of the data is reached. How to determine this stopping time efficiently and accurately is a central difficulty in progressive sampling.

We study stopping time determination by approximating the generalization error of the hypothesis rather than by assuming the often observed shape for the learning curve and trying to detect whether the final plateau has been reached in the curve. We use data dependent generalization error bounds. Instead of using the common cross validation approach, we use the recently introduced Rademacher penalties, which have been observed to give good results on simple concept classes.

We experiment with two-level decision trees built by the learning algorithm T2. It finds a hypothesis with the minimal error with respect to the sample. The theoretically well motivated stopping time determination based on Rademacher penalties gives results that are much closer to those attained using heuristics based on assumptions on learning curve shape than distribution independent estimates based on VC dimension do.

Introduction

Sampling can be a powerful technique for inductive algorithms to avoid unnecessary processing of the whole available data. It helps to circumvent memory limitations, gain efficiency, and can even result in increased accuracy (Fürnkranz 1998). Sampling is particularly useful in the data mining context, where the sheer abundance of the data may impair even the fastest algorithms (Kivinen and Mannila 1994, Toivonen 1996, Scheffer and Wrobel 2000). However, it is hard to know how large a sample to choose. Drawing a too small sample will not yield a good-enough performance and, on the other hand, choosing a too large sample will unavoidably mean wasting computational effort. An apparent solution is to use *progressive (or dynamic) sampling*; taking gradually increasing portions of the available data as the sample (John and Langley 1996, Provost, Jensen, and Oates 1999). With a suitable sampling *schedule* — sample size selection scheme — and an efficient stopping time determination method progressive sampling is asymptotically as efficient as knowing the right sample size in advance (Provost, Jensen, and Oates 1999). Sampling schedules are theoretically well understood, but stopping time determination is not. Premature stopping will yield suboptimal results and too late stopping will mean unbeneficial wasting of resources. The obvious goal is to stop as soon as growing the sample will not yield any more advantage. Unfortunately, determining this stopping time is extremely difficult.

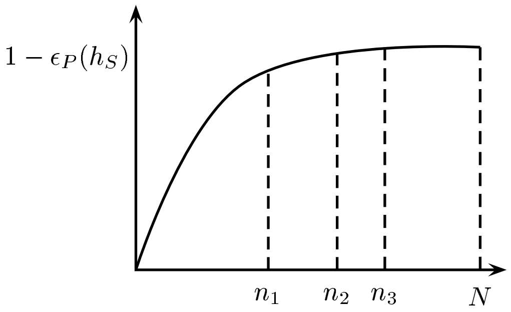

Figure 1: A common shape for a learning curve.

Previous studies on progressive sampling have relied on the empirically often observed common shape of the learning curves (Cortes et al. 1994, Oates and Jensen 1997, Frey and Fisher 1999): The classification accuracy of a learning algorithm first typically slopes steeply from the initial accuracy with the first few examples (see Fig. 1). After a while the accuracy growth diminishes before reaching a plateau, during which practically no accuracy gain is obtained. The optimal sample size to draw with this behavior would be the first one (the smallest one) giving an accuracy on the plateau.

It is hard to determine which sample sizes give accuracies belonging to the plateau and which do not. For example, which of the indicated subsample sizes, if any, in Fig. 1 is the first one giving an accuracy that belongs to the plateau? Furthermore, all domains do not have a plateau in their learning curve at all (Catlett 1991). One can approximate the steepness of the learning curve in the local neighborhood of a sample size in order to decide whether the expected plateau has been reached or not (Provost, Jensen, and Oates 1999), or one can try to approximate whether the accuracy with the sample size is with high probability close enough to that which will be reached eventually if all of the data is used (John and Langley 1996). In practice, though, learning curves are not always as well-behaving as the one depicted in Fig. 1 (Haussler at al. 1996, Provost, Jensen, and Oates 1999). Moreover, stopping time determination using the above-mentioned methods can be computationally costly.

Our approach, rather, is to rely on a data dependent bound on the generalization error of the learning algorithm in question. The idea is to stop sampling as soon as the generalization error of the hypothesis chosen by the learning algorithm can be guaranteed to be close to its error on the training set with high probability. Thus, we try to choose the smallest sample size that enables us to prove good results on the generalization capability of the hypothesis chosen by the learning algorithm. Instead of using distribution independent generalization error bounds based on, e.g., the Vapnik-Chervonenkis (VC) dimension (Vapnik 1998) we use the recently introduced approach based on *Rademacher penalization* (Koltchinskii 2001, Bartlett and Mendelson 2001). A related approach based on sequential statistical tests has been introduced by Schuurmans and Greiner (1995).

In the remainder of this paper we first review approaches to sampling large databases. Then data dependent bounds on the generalization error of a learning algorithm are recapitulated. In particular, Rademacher penalization is reviewed. Thereafter, progressive Rademacher sampling is introduced. We prove that it is possible to combine geometric sampling schedules efficiently with the Rademacher penalization approximation of the generalization error. Following that two-level decision trees and learning them (Auer, Holte, and Maass 1995) are briefly recapitulated. Our empirical experiments chart the utility and feasibility of progressive Rademacher sampling using two-level decision trees. We contrast the results to both theoretical and practical alternatives. Finally, some conclusions on the study are presented.

Static and Dynamic Sampling of Large Databases

Let $S = \{ (x_i, y_i) \mid i = 1, \ldots, N \}$ be an example set consisting of N independent examples $(x_i, y_i) \in X \times \{ 0, 1 \}$ each of which is drawn according to some unknown probability measure P on $X \times \{ 0, 1 \}$. We assume that N is finite, but so large that it cannot be exhausted in practice. For example, it may be impossible to keep N examples in main memory at one time. In such a situation the theoretical time complexities of learning algorithms do not necessarily hold.

Kivinen and Mannila (1994) have derived sample size bounds for approximate verification of the truth of first-order logical formulas represented in tuple relational calculus for a given database relation by considering only a random sample of the relation. The work of Toivonen (1996) was motivated by the need to reduce the number of expensive passes through the database in searching for the frequent association rules. One pass through the whole database can usually be avoided by inferring a (super)set of candidate rules with a lowered frequency threshold from a random sample of the database. Only one pass through the complete database is required to validate which of the candidates actually are frequent enough in the whole database. Scheffer and Wrobel's (2000) sequential sampling algorithm's main contributions are to return k best hypotheses instead of only one, work with many different utility functions, and to rank the candidate hypotheses already at an early stage.

The static sampling approaches require a lot of information in advance if results of guaranteed quality are desired. Dynamic approaches to sampling have also been proposed. An early dynamic sampling technique is Quinlan's (1983) *windowing*, in which a consistent decision tree is first grown on a random sample, falsely classified examples are then augmented to the data, and the process is repeated until convergence. Windowing can be beneficial in noise-free domains, but cannot cope well with noise (Fürnkranz 1998).

Successively increasing the sample size until it gives good enough results is the idea behind progressive sampling. The method of determining what size samples to choose next is called a *schedule*. John and Langley (1996) used an *arithmetic* schedule, in which the size of the sample is increased by a constant portion, n_Δ, at each step. Let n_0 be the number of examples in the initial sample S_0. Then the size of the ith sample S_i will be $|S_i| = n_i = n_0 + (i \cdot n_\Delta)$. The problem with this schedule is that one may need to iterate too many times before reaching the required accuracy.

An alternative to the arithmetic schedule is to use a *geometric* schedule, in which the initial sample size is multiplied according to a geometric sequence. In this case $n_i = a^i n_0$, where n_0 is the initial sample size and $a > 1$ is a constant. Provost, Jensen, and Oates (1999) showed that geometric schedule combined with an efficient stopping time detection is asymptotically optimal for superlinear learning algorithms in the sense that it gives the same asymptotic time complexity as knowing the optimal sample size in advance.

They also observed that it is in principle possible to compute the optimal schedule as well. However, that requires knowing the execution time $t(n)$ of the learning algorithm on n instances and the probability $\Phi(n)$ that convergence requires more than n instances. Then the expected cost of schedule $\Sigma = \{ n_1, \ldots, n_k \}$ can be computed as

$$C(\Sigma) = \sum_{i=1}^{k} \Phi(n_{i-1}) t(n_i),$$

where $n_0 = 0$ and $\Phi(0) = 1$, because convergence definitely requires more than 0 examples. Let $c[i, j]$ denote the cost of the minimum expected cost schedule, which includes samples of i and j instances, of all samples in the size range $[i, j]$. The cost of the optimal schedule $c[0, N]$ can be computed by the recurrence

$$c[i, j] = \min \left\{ \Phi(i)t(j), \min_{i<k<j} c[i, k] + c[k, j] \right\}.$$

Dynamic programming can be used to solve it in $O(N^3)$ time.

Provost, Jensen, and Oates (1999) based the approximation of the stopping time on learning curve analysis, where the underlying assumption is that machine learning algorithms perform on all domains with increasing sample sizes roughly as depicted in Fig. 1. However, as they also discuss, this well-behavedness assumption is not always true, even though many empirical studies have supported this view. If the learning curve does not behave well, there is no ground in trying to determine the stopping time by examining the learning curve's local slope. In John and Langley's (1996) work stopping time determination was, rather, based on trying to approximate the difference between the accuracy of the hypothesis chosen after seeing n examples and that of the one chosen after seeing all N examples. Approximating the accuracy on all of the data requires extrapolating the learning curve, and in this task an explicit power law assumption about the shape of the learning curve was also made.

Data Dependent Bounds on the Generalization Error of a Hypothesis

We now give up assumptions on the shape of the learning curve. However, if nothing is assumed about the learning algorithm or the hypotheses it may choose, it is impossible to prove any bounds on the generalization capability of a hypothesis. Therefore, we assume — as one usually does in the PAC and statistical learning settings — that the learning algorithm chooses its hypothesis from some fixed hypothesis class $\mathcal{H}$. Under this assumption generalization error analysis provides theoretical results bounding the generalization error of hypotheses $h \in \mathcal{H}$ that are based on the sample and the properties of the hypothesis class. We review next some results of generalization error analysis that will be useful in stopping time detection.

Given a hypothesis h, its *generalization error* is the probability that a randomly drawn example (x, y) is misclassified:

$$\epsilon_P(h) = P(h(x) \neq y).$$

The general goal of learning, of course, is to find a hypothesis with a small generalization error. However, since the generalization error of a hypothesis depends on the unknown probability distribution P, it cannot be computed directly based on the sample alone. We can try to approximate generalization error of the hypothesis h by its *training error* on n examples:

$$\hat{\epsilon}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i),$$

where L is the $0/1$ loss function

$$L(y, y') = \begin{cases} 1, & \text{if } y \neq y'; \\ 0, & \text{otherwise.} \end{cases}$$

Empirical Risk Minimization (ERM) is a principle that suggest choosing the hypothesis $h \in \mathcal{H}$ whose training error is minimal. In relatively small and simple hypothesis classes finding the minimum training error hypothesis is computationally feasible. To guarantee that ERM yields hypotheses with small generalization error, one can try to bound $\sup_{h\in\mathcal{H}} |\epsilon_P(h) - \hat{\epsilon}_n(h)|$. Under the assumption that the examples are independent and identically distributed (i.i.d), whenever the hypothesis class $\mathcal{H}$ is not too complex, the difference of the training error of the hypothesis h on n examples and its true generalization error converge to 0 in probability as n tends to infinity. We take advantage of this asymptotic behavior, and base sampling stopping time determination on a data-dependent upper bound of the difference between generalization and training error.

The most common approach to deriving generalization error bounds for hypotheses is based on taking the VC dimension of the hypothesis class into account. The problem with this approach is that it provides optimal results only in the worst case — when the underlying probability distribution is as bad as can be. Thus, the generalization error bounds based on VC dimension tend to be overly pessimistic. Data dependent generalization error bounds, on the other hand, are provably almost optimal for any given domain (Koltchinskii 2001). In the following we review the foundations of a recent promising approach to bounding the generalization error.

A *Rademacher random variable* (Koltchinskii 2001) takes values $+1$ and -1 with probability 1/2 each. Let $r_1, r_2, \ldots, r_n$ be a sequence of i.i.d. Rademacher random variables independent of the data $(x_1, y_1), \ldots, (x_n, y_n)$. The *Rademacher penalty* of the hypothesis class $\mathcal{H}$ is defined as:

$$R_n(\mathcal{H}) = \sup_{h\in\mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n} r_i L(h(x_i), y_i) \right|.$$

By a symmetrization inequality of the theory of empirical processes (Van der Vaart and Wellner 2000)

$$\mathbf{E}\left\{ \sup_{h\in\mathcal{H}} |\epsilon_P(h) - \hat{\epsilon}_n(h)| \right\} \leq 2\mathbf{E}\left\{ R_n(\mathcal{H}) \right\}, \qquad (1)$$

where expectations are taken over the choice of examples on the left and over the choice of examples and Rademacher random variables on the right. Furthermore, the random variables $\sup_{h\in\mathcal{H}} |\epsilon_P(h) - \hat{\epsilon}_n(h)|$ and $R_n(\mathcal{H})$ are tightly concentrated around their expectations (Koltchinskii 2001). Thus, one can show using standard concentration inequalities that with probability at least $1 - \delta$

$$\epsilon_P(h) \leq \hat{\epsilon}_n(h) + 2R_n(\mathcal{H}) + \eta(\delta, n), \qquad (2)$$

where

$$\eta(\delta, n) = 5\sqrt{\frac{\ln(2/\delta)}{2n}}$$

is a small error term that takes care of the fluctuations of the analyzed random variables around their expectations.

The usefulness of inequality (2) stems from the fact that its right-hand side depends only on the training sample and not on P directly. Furthermore, Koltchinskii (2001) has shown that computation of the Rademacher penalty is equivalent to the minimization of the training error on relabeled training data. This means that $R_n(\mathcal{H})$ can be computed with the algorithm used for ERM. To get the effects of Rademacher random variables r_i we define a new set of labels z_i for the training data as the flipping of the original label with probability $1/2$.

Altogether, the computation of the Rademacher penalty entails the following steps.

- Flip the label of each example (x_i, y_i) with probability $1/2$ to obtain a new set of labels z_i.
- Find the functions $h_1, h_2 \in \mathcal{H}$ that minimize the empirical error with respect to the set of labels z_i and $-z_i$, respectively.
- Compute $|(1/n)\sum_{i=1}^{n} r_i L(h(x_i), y_i)|$ for $h = h_1, h_2$ and select the maximum out of these two values as the Rademacher penalty.

Progressive Rademacher Sampling

Koltchinskii et al. (2000) have applied Rademacher penalties to provide approximate solutions to difficult control problems. Dynamic sampling schedules with provable properties have been applied in this context as well. As sampling schedules the bootstrap approach as well as the geometric schedule $n_i = 2^i n_0$ were used. We now adapt the techniques introduced by Koltchinskii et al. (2000) to stopping time detection.

The least required sample size $n^P_{\min}(\varepsilon, \delta)$ over the class $\mathcal{H}$ with respect to P is the minimal number of examples needed to guarantee that the training error of the hypothesis h is within a distance ε from the generalization error of h for every $h \in \mathcal{H}$:

$$\arg\min_{n \geq 1} \left\{ \mathbf{Pr} \left\{ \sup_{h \in \mathcal{H}} |\epsilon_P(h) - \hat{\epsilon}_n(h)| \geq \varepsilon \right\} \leq \delta \right\}.$$

$n^P_{\min}(\varepsilon, \delta)$ can be thought of as an optimal sample size in the sense that a smaller sample size would not enable us to be confident that the training error of the hypothesis is a good approximation of its generalization error. However, $n^P_{\min}(\varepsilon, \delta)$ depends directly on P and, thus, cannot be computed. We show next how Rademacher penalties can be used to give computable approximations of $n^P_{\min}(\varepsilon, \delta)$.

Given $\varepsilon > 0$ and $\delta \in (0, 1)$, let $n_0(\varepsilon, \delta)$ denote the initial sample size of our learning algorithms. It is assumed to be a non-increasing function of both ε and δ. A random variable τ, taking positive integer values, is called a *stopping time* if, for all $n \geq 1$ the decision whether $\tau \leq n$, or not, depends only on the information available by time n; i.e., only on $(x_1, y_1), \ldots, (x_n, y_n)$. A stopping time τ is called *well-behaving* with parameters (ε, δ) if it is such that $\tau \geq n_0(\varepsilon, \delta)$ and

$$\mathbf{Pr} \left\{ \sup_{h \in \mathcal{H}} |\hat{\epsilon}_\tau(h) - \epsilon_P(h)| \geq \varepsilon \right\} \leq \delta.$$

An immediate consequence of this definition is that if τ is well-behaving with parameters (ε, δ) and $\hat{h}$ is a hypothesis that minimizes empirical risk based on the sample $\{ (x_i, y_i) \mid i = 1, \ldots, \tau \}$, then

$$\mathbf{Pr} \left\{ \epsilon_P(\hat{h}) \geq \inf_{h \in \mathcal{H}} \epsilon_P(h) + 2\varepsilon \right\} \leq \delta.$$

In other words, it is enough to draw τ examples in order to find, with high probability, a hypothesis in $\mathcal{H}$ that is almost as accurate as the most accurate one in $\mathcal{H}$.

The question, though, is how to construct the well-behaving stopping times on the basis of the available data only — without using the knowledge of P — and which of the stopping times from this set is the best used in the learning algorithms. Let us now define stopping times that are tied to a geometric sampling schedule and reduction of the Rademacher penalty.

Definition The *Rademacher stopping time* $\nu(\varepsilon, \delta)$ with parameters (ε, δ) for the hypothesis class $\mathcal{H}$ is

$$\nu(\varepsilon, \delta) = \min_{i \geq 1} \left\{ n_i = 2^i n_0(\varepsilon, \delta) \mid R_{n_i}(\mathcal{H}) < \varepsilon \right\}.$$

Koltchinskii et al. (2000) derived data dependent results that hold for any distribution that could have produced the sample S. Instead of considering the set of all probability distributions on S and its supremum upper bound (Koltchinskii at al. 2000), in the following results we examine the (unknown) true probability distribution P producing S.

Theorem 1 *Let*

$$n_0(\varepsilon, \delta) \geq \left\lfloor \frac{4}{\varepsilon^2} \log\left(\frac{4}{\delta}\right) \right\rfloor + 1.$$

Then, for all $\varepsilon > 0$ and $\delta \in (0, 1)$,

1. *$\nu(\varepsilon, \delta)$ is well-behaving with parameters $(5\varepsilon, \delta)$.*
2. *Moreover, if $n^P_{\min}(\varepsilon, \delta) \geq n_0(\varepsilon, \delta)$, then for all $\varepsilon > 0$ and $\delta \in (0, 1/2)$, the probability that $\nu(24\varepsilon, \delta) > n^P_{\min}(\varepsilon, \delta)$ is at most 3δ (for any class $\mathcal{H}$ of hypotheses and any distribution P).*

Theorem 2 *If*

$$n_0(\varepsilon, \delta) \geq \left\lfloor \frac{4}{\varepsilon^2} \log\left(\frac{4}{\delta}\right) \right\rfloor + 1$$

and $12/\varepsilon \leq n^P_{\min}(\varepsilon, \delta) \leq n_0(\varepsilon, \delta)$, then

$$\mathbf{Pr} \left\{ \nu(30\varepsilon, \delta) > 2n_0(\varepsilon, \delta) \right\} \leq \delta.$$

We omit the proofs of Theorems 1 and 2 because they are both simple modification of the corresponding proofs by Koltchinskii et al. (2000). The theorems show that, under certain mild conditions, $\nu(\varepsilon, \delta)$ is well-behaving and, furthermore, that it yields nearly as good sample sizes as knowing the unknown distribution-dependent sample complexity $n^P_{\min}(\varepsilon, \delta)$. This is in clear contrast with the stopping times that one could define based on the VC dimension of $\mathcal{H}$ which would be competitive with $n^P_{\min}(\varepsilon, \delta)$ only for worst-case P.

Learning Two-Level Decision Trees

Computation of the Rademacher penalty entails finding the hypothesis that minimizes the training error. Not many training error minimizing learning algorithms are known for hypothesis classes of reasonable size. Moreover, two executions of the learning algorithm are required to compute the Rademacher penalty of the underlying hypothesis class. Therefore, it is vital that the learning algorithm is efficient. Thus far the only practical experiments on using Rademacher penalties that we are aware of are those of Lozano (2000), who used real intervals as his concept class.

T2 (Auer, Holte, and Maass 1995) is an agnostic PAC-learning algorithm with guaranteed performance for any distribution of data. It learns two-level decision trees that minimize the training error within this class of hypotheses. Handling of numerical attributes is the main difficulty in learning concise representations; one cannot reiterate the splitting of a numerical value range like, e.g., C4.5 does (Quinlan 1993).

In the root of a decision tree produced by T2 a numerical value range can be split into two intervals using a threshold value. Missing attribute values, which are common in real-world data, are treated as an extra value in T2. Thus, if a numerical attribute is chosen to the root of a tree in T2, then the tree will have three subtrees rooted at the second level of the tree. At the second level the value range of a continuous attribute (even the one that was chosen to the root) can be split up to k intervals, where k is a prespecified parameter of the algorithm. A discrete attribute is handled, as usual, so that the node testing the value of such an discrete attribute will have as many subtrees as there are different values in the attribute's value range (plus one subtree for missing values).

The time complexity of T2 for n examples on m attributes is $O(k^2 m^2 n \log n)$. In other words, with respect to the sample size T2 only requires $O(n \log n)$ time. In experiments (Auer, Holte, and Maass 1995) the two-level decision trees produced by T2 have been observed to be highly competitive with the more complex decision tree hypotheses of C4.5 (Quinlan 1993).

Experiments

We have tested progressive Rademacher sampling combined with the T2 learning algorithm on some UCI (Blake and Merz 1998) domains. In the following results from the Adult (Census) domain, which was also used by Provost, Jensen, and Oates (1999), are reviewed.

Fig. 2 plots the penalties based on Rademacher penalization and VC dimension. For both penalties the confidence parameter δ equals 0.01. Rademacher penalty is the one given by (2). The VC dimension penalty is determined by the formula (Vapnik 1998):

$$2\sqrt{\frac{d(\ln(2n/d)+1)+\ln(9/\delta)}{n}},$$

where $d = 656$ is a lower bound of the VC dimension of the hypothesis class used by T2. The lower bound is determined by the maximal number of leaves in a single two-level tree for the Adult domain. The figure shows that the results obtained by Rademacher penalization are almost an order of magnitude smaller than those given by the VC method.

Fig. 3 plots the training and test accuracies of two-level decision trees on different sized samples. Observe that the x-axis in this figure is in logarithmic scale (with base 2). Thus, the points on the curves correspond to successive sample sizes of the geometric schedule that was used. The generalization error lower bound computed on the basis of Rademacher penalties is also displayed.

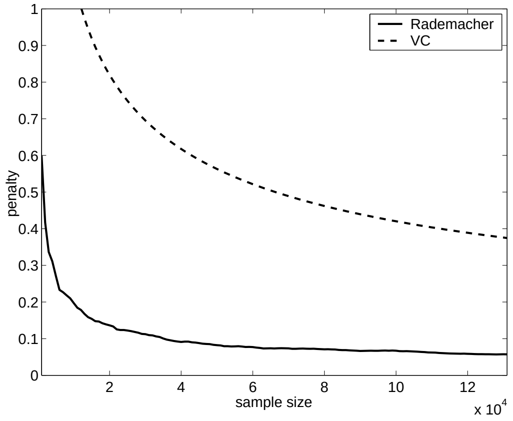

Figure 2: Development of the Rademacher penalty (solid line) and VC-based penalty (dashed line) with increasing sample size.

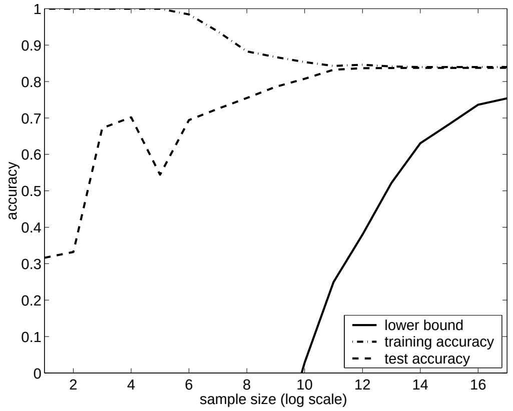

Figure 3: Training accuracy (dash-dotted line), test accuracy (dashed line), and the accuracy lower bound determined by Rademacher penalty (solid line) in progressive sampling.

We chose $\varepsilon = 0.1$ and $\delta = 0.01$. With these values the initial sample size $n_0(\varepsilon, \delta)$ would be approximately 60,000, but for illustration we have plotted the accuracies and the lower bound starting from $n_0 = 2$. With these choices the stopping time $\nu(\varepsilon/5, \delta)$ evaluated to 65,536. If the sample size had been determined by the bounds based on VC dimension, a total of approximately 2,623,000 examples would have been needed.

On the other hand, the optimal sample size as determined empirically by Provost, Jensen, and Oates (1999) is approximately 8,000. Thus, although the sample size obtained by the method proposed in this paper is larger than the one suggested by heuristic stopping time detection methods, it dramatically outperforms the one based on VC dimension.

Conclusions

Sampling, in principle, is a powerful method for enhancing space and time efficiency as well as, on occasion, classifier accuracy in inductive learning. However, in static sampling we have to know a lot in advance, if results of guaranteed quality are desired. Progressive sampling has been proposed to circumvent the problems associated with static sampling. Nevertheless, it too has its problems, the most serious of which is the problem of determining the stopping time efficiently and accurately. To compute the optimal sampling schedule requires a lot of information. Also then the stopping time approximation can be computationally expensive.

In this paper we have studied combining data dependent Rademacher generalization error bound approximation with straightforward geometric sampling schedules. Computation of Rademacher penalties requires executing the learning algorithm two times and finding the minimum training error hypothesis within the hypothesis class. These requirements limit the applicable hypothesis classes to relatively simple ones. However, the two-level decision trees that were used in this study have been observed, in practice, to be competitive in their prediction accuracy with the more complex decision trees produced by C4.5.

Our experiments indicated that using Rademacher penalization gives orders of magnitude more realistic required sample size estimates than the ones based on VC dimension. However, the level attainable using direct learning curve estimation is hard to reach at least with the relatively small domain sizes and whenever the learning curve behaves at all well.

References

Auer, P.; Holte, R. C.; and Maass, W. 1995. Theory and Application of Agnostic PAC-Learning with Small Decision Trees. In *Proceedings of the Twelfth International Conference on Machine Learning*, 21–29. San Francisco, Calif.: Morgan Kaufmann.

Bartlett, P. L., and Mendelson, S. 2001. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. In *Computational Learning Theory, Proceedings of the Fourteenth Annual Conference*, 224–240. Lecture Notes in Artificial Intelligence **2111**. Heidelberg: Springer.

Blake, C. L., and Merz, C. J. 1998. UCI Repository of Machine Learning Databases. Univ. of California, Irvine, Dept. of Information and Computer Science.

Catlett, J. 1991. Megainduction: A Test Flight. In *Proceedings of the Eighth International Workshop on Machine Learning*, 596–599. San Mateo, Calif.: Morgan Kaufmann.

Cortes, C.; Jackel, L. D.; Solla, S. A.; Vapnik, V.; and Denker J. S. 1994. Learning Curves: Asymptotic Values and Rate of Convergence. In *Advances in Neural Information Processing Systems 6*, 327–334. San Francisco, Calif.: Morgan Kaufmann.

Frey, L. J., and Fisher, D. H. 1999. Modeling Decision Tree Performance with the Power Law. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, 59–65. San Francisco, Calif.: Morgan Kaufmann.

Fürnkranz, J. 1998. Integrative Windowing. *Journal of Artificial Intelligence Research* **8**: 129–164.

Haussler, D.; Kearns, M.; Seung, H. S.; and Tishby, N. 1996. Rigorous Learning Curve Bounds from Statistical Mechanics. *Machine Learning* **25**(2–3): 195–236.

John, G., and Langley, P. 1996. Static versus Dynamic Sampling for Data Mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 367–370. Menlo Park, Calif.: AAAI Press.

Kivinen, J., and Mannila, H. 1994. The Power of Sampling in Knowledge Discovery. In *Proceedings of the Thirteenth ACM Symposium on Principles of Database Systems*, 77–85. New York, NY: ACM Press.

Koltchinskii, V. 2001. Rademacher Penalties and Structural Risk Minimization. *IEEE Transactions on Information Theory* **47**(5): 1902–1914.

Koltchinskii, V.; Abdallah, C. T.; Ariola, M.; Dorato, P.; and Panchenko, D. 2000. Improved Sample Complexity Estimates for Statistical Learning Control of Uncertain Systems. *IEEE Transactions on Automatic Control* **45**(12): 2383–2388.

Lozano, F. 2000. Model Selection Using Rademacher Penalization. In *Proceedings of the Second ICSC Symposium on Neural Networks.* Berlin, Germany: ICSC Academic.

Oates, T., and Jensen, D. 1997. The Effects of Training Set Size on Decision Tree Complexity. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 254–261. San Francisco, Calif.: Morgan Kaufmann.

Provost, F.; Jensen, D.; and Oates, T. 1999. Efficient Progressive Sampling. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 23–32. New York, NY: ACM Press.

Quinlan, J. R. 1983. Learning Efficient Classification Procedures and Their Application to Chess End Games. In Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M., eds., *Machine Learning: An Artificial Intelligence Approach*, 463–482. Palo Alto, Calif.: Tioga.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. San Francisco, Calif.: Morgan Kaufmann.

Scheffer, T., and Wrobel, S. 2000. A Sequential Sampling Algorithm for a General Class of Utility Criteria. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 330–334. New York, NY: ACM Press.

Schuurmans, D., and Greiner, R. 1995. Practical PAC Learning. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1169–1175. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Toivonen, H. 1996. Sampling Large Databases for Association Rules. In *Proceedings of the Twenty-Second International Conference on Very Large Databases*, 134–145. San Francisco, Calif.: Morgan Kaufmann.

Van der Vaart, A. W., and Wellner, J. A. 2000. *Weak Convergence and Empirical Processes*. Corrected second printing. New York, NY: Springer-Verlag.

Vapnik, V. N. 1998. *Statistical Learning Theory*. New York, NY: Wiley.