Finding the Most Interesting Patterns in a Database Quickly by Using Sequential Sampling

Tobias Scheffer and Stefan Wrobel
Otto von Guericke University, FIN/IWS, Universitaetsplatz 2
39106 Magdeburg, Germany
{scheffer, wrobel}@iws.cs.uni-magdeburg.de

Technical Report
University of Magdeburg, School of Computer Science

January 22, 2001

Abstract. Many discovery problems, *e.g.,* subgroup or association rule discovery, can naturally be cast as n-best hypotheses problems where the goal is to find the n hypotheses from a given hypothesis space that score best according to a certain utility function. We present a sampling algorithm that solves this problem by issuing a small number of database queries while guaranteeing precise bounds on confidence and quality of solutions. Known sampling approaches have treated single hypothesis selection problems, assuming that the utility be the average (over the examples) of some function — which is not the case for many frequently used utility functions. We show that our algorithm works for all utilities that can be estimated with bounded error. We provide these error bounds and resulting worst-case sample bounds for some of the most frequently used utilities, and prove that there is no sampling algorithm for a popular class of utility functions that cannot be estimated with bounded error. The algorithm is sequential in the sense that it starts to return (or discard) hypotheses that already seem to be particularly good (or bad) after a few examples. Thus, the algorithm is almost always faster than its worst-case bounds.

1 Introduction

The general task of knowledge discovery in databases (KDD) is the "automatic extraction of novel, useful, and valid knowledge from large sets of data" [7]. An important aspect of this task is *scalability*, *i.e.,* the ability to successfully perform discovery in ever-growing datasets. Unfortunately, even with discovery algorithms optimized for very large datasets, for many application problems it is infeasible to process all of the given data. Whenever more data is available than can be processed in reasonable time, an obvious strategy is to use only a randomly drawn *sample* of the data. Clearly, if parts of the data are not looked at, it is impossible in general to guarantee that the results produced by the discovery algorithm will be identical to the results returned on the complete dataset. If the use of sampled datasets is to be more than a practitioner's "hack", sampling must be combined with discovery algorithms in a fashion that allows us to give the user *guarantees* about how far the results obtained using sampling differ from the optimal (non-sampling based) results. The goal of a sampling discovery algorithm then is to guarantee this quality using the minimum amount of examples [25].

Known algorithms that do give rigorous guarantees on the quality of the returned solutions for all possible problems usually require an impractically large amount of data. One approach to finding practical algorithms is to process a fixed amount of data but determine the possible strength of the quality guarantee dynamically, based on characteristics of the data; this is the idea of self-bounding learning algorithms [8] and shell decomposition bounds [13, 19]. Another approach (which we pursue) is to demand a certain fixed quality and determine the required sample size dynamically based on characteristics of the data that have already been seen; this idea has originally been referred to as sequential analysis [4, 28, 9].

In the machine learning context, the idea of sequential sampling has been developed into the Hoeffding race algorithm [20] which processes examples incrementally, updates the empirical utility values simultaneously, and starts to output (or discard) hypotheses as soon as it becomes very unlikely that some hypothesis is not near-optimal (or very poor, respectively). The incremental greedy learning algorithm PALO [11] has been reported to require many times fewer examples than the worst-case bounds suggest. In the context of knowledge discovery in databases, too, sequential sampling algorithms can reduces the required amount of data significantly [12, 6].

These existing sampling algorithms address discovery problems where the goal is to select from a space of possible hypotheses H one of the elements with maximal value of an *instance-averaging* quality function f, or all elements with an f-value above a user-given threshold (*e.g.,* all association rules with sufficient support). With instance-averaging quality functions, the quality of a hypothesis h is the average across all instances in a dataset D of an instance quality function f_{inst}.

Many discovery problems, however, cannot easily be cast in this framework. Firstly, it is often more natural for a user to ask for the n best solutions instead of the single best or all hypotheses above a threshold – see, *e.g.,* [30]. Secondly, many popular quality measures cannot be expressed as an averaging quality function. This is the case *e.g.,* for all

functions that combine generality and distributional properties of a hypothesis; generally, both generality and distributional properties (such as accuracy) have to be considered for association rule and subgroup discovery problems. The task of subgroup discovery [17] is to find maximally general subsets of database transactions within which the distribution of a focused feature differs maximally from the default probability of that feature in the whole database. As an example, consider the problem of finding groups of customers who are particularly likely (or unlikely) to buy a certain product.

In this paper, we present a general sampling algorithm for the n-best hypotheses problem that works for any utility functions that can be estimated with bounded error at all. To this end, in Section 2, we first define the n-best hypotheses problem more precisely and identify appropriate quality guarantees. Section 3 then presents the generic sequential sampling algorithm. In Section 4, we prove that many of the popular utility functions that have been used in the area of knowledge discovery in databases indeed can be estimated with bounded error, giving detailed bounds. In order to motivate the instantiations of our sampling algorithm and put it into context, we first define some relevant knowledge discovery tasks in Section 4. For one popular class of functions that cannot be used by our algorithm, we prove that there cannot be a sampling algorithm at all. Our results thus also give an indication as to which of the large numbers of popular utility functions are preferable with respect to sampling. In Section 6, we evaluate our results and discuss their relation to previous work.

2 Approximating n-Best Hypotheses Problems

In many cases, it is more natural for a user to ask for the n best solutions instead of the single best or all hypotheses above a threshold. Such n-best hypotheses problems can be stated more precisely as follows – adapted from [30], where this formulation is used for subgroup discovery:

Definition 1 (n-best hypotheses problem) *Let D be a database of instances, H a set of possible hypotheses, $f : H \times D \to \mathbb{R}^{\geq 0}$ a quality or utility function on H, and n, $1 \leq n < |H|$, the number of desired solutions. The* n-best hypotheses problem *is to find a set $G \subseteq H$ of size n such that*

$$\textit{there is no } h' \in H\textit{: } h' \notin G \textit{ and } f(h', D) > f_{min}\textit{, where } f_{min} := min_{h \in G} f(h, D).$$

Whenever we use sampling, the above optimality property cannot be guaranteed, so we must find appropriate alternative guarantees. Since for n-best problems, the exact quality and rank of hypotheses is often not central to the user, it is sufficient to guarantee that G indeed "approximately" contains the n best hypotheses. We can operationalize this by guaranteeing that there will never be a non-returned hypothesis that is "significantly" better than the worst hypothesis in our solution. More precisely, we will use the following problem formulated along the lines of PAC (probably approximately correct) learning:

Definition 2 (Approximate n-best hypotheses problem) *Let D, H, f and n as in the preceding definition. Then let δ, $0 < \delta \leq 1$, be a user-specified confidence, and $\varepsilon \in \mathbb{R}^+$ a user-specified maximal error. The* approximate n-best hypotheses problem *is to find a set $G \subseteq H$ of size n such that*

> *with confidence $1 - \delta$, there is no $h' \in H$: $h' \notin G$ and $f(h', D) > f_{min} + \varepsilon$, where $f_{min} := min_{h \in G} f(h, D)$.*

In other words, we want to find a set of n hypotheses such that, with high confidence, no other hypothesis outperforms any one of them by more than ε, where f is an arbitrary performance measure.

In order to design an algorithm for this problem, we need to make certain assumptions about the quality function f. Ideally, an algorithm should be capable of working (at least) with the kinds of quality functions that have already proven themselves useful in practical applications. If the problem is to *classify* database items (*i.e.,* to find a total function mapping database items to class labels), *accuracy* is often used as utility criterion. For the discovery of *association rules*, by contrast, one usually relies on *generality* as primary utility criterion [1]. Finally, for subgroup discovery, it is commonplace to combine both generality and *distributional unusualness*, resulting in relatively complex evaluation functions (see, *e.g.,* [18] for an overview).

In light of the large range of existing and possible future utility functions and in order to avoid unduly restricting our algorithm, we will not make syntactic assumptions about f. In particular, unlike [6], we will not assume that f is a single probability nor that it is (a function of) an average of instance properties. Instead, we only assume that it is possible to determine a *confidence interval* f that bounds the possible difference between true utility (on the whole database) and estimated utility (on the sample) with a certain confidence. We expect the confidence interval to narrow as the sample size grows. As will show in Section 4 below, finding such confidence intervals is straightforward for classification accuracy, and is also possible for all but one of the popular utility functions from association rule and subgroup discovery. More precisely, we define an confidence interval for f as follows.

Definition 3 (Utility confidence interval) *Let f be a utility function, let $h \in H$ be a hypothesis. Let $f(h, D)$ denote the true quality of h on the entire dataset, $\hat{f}(h, Q_m)$ its estimated quality computed based on a sample $Q_m \subseteq D$ of size m. Then $E : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ is a* utility confidence bound *for f iff for any δ, $0 < \delta \leq 1$,*

$$Pr_S[|\hat{f}(h, Q_m) - f(h, D)| \leq E(m, \delta)] \geq 1 - \delta \tag{1}$$

Equation 1 says that E provides a two-sided confidence interval on $\hat{f}(h, Q_m)$ with confidence δ. In other words, the probability of drawing a sample Q_m (when drawing m transactions independently and identically distributed from D), such that the difference between true and estimated utility of any hypothesis disagree by ε or more (in either direction) lies below δ. If, in addition, for any $\delta, 0 < \delta \leq 1$ and any ε there is a number m such that $E(m, \delta) \leq \varepsilon$ we say that the confidence interval *vanishes*. In this case, we

can shrink the confidence interval (at any confidence level δ) to arbitrarily low nonzero values by using a sufficiently large sample. We sometimes write the confidence interval for a specific hypothesis h as $E_h(m, \delta)$. Thus, we allow the confidence interval to depend on characteristics of h, such as the variance of one or more random variables that the utility of h depends on.

We will discuss confidence intervals for different functions of interest in Section 4; here, as a simple example, let us only note that if f is simply a probability over the examples, then we can use the Chernoff inequality to derive a confidence interval; when f is the average (over the examples) of some function with bounded range, then the Hoeffding inequality implies a confidence interval. Of course, we should also note that the trivial function $E(m, \delta) := \Lambda$ is an error probability bound function for any f with lower bound of zero and upper bound of Λ, but we will see that we can only guarantee termination when the confidence interval vanishes as the sample size grows.

3 Sampling Algorithm

The general approach to designing a sampling algorithm is to use an appropriate error probability bound to determine the required number of examples for a desired level of confidence and accuracy. When estimating a single probability, Chernoff bounds [3] that are used in PAC theory [15, 29, 27] and many other areas of statistics and computer science can be used to determine appropriate sample bounds [25]. When such algorithms are implemented, the Chernoff bounds can be replaced by tighter normal or t distribution tables.

Unfortunately, the straightforward extension of such approaches to selection or comparison problems like the n-best hypotheses problem leads to unreasonably large bounds: to avoid errors in the worst case, we have to take very large samples to recognize small differences in utility, even if the actual differences between hypotheses to be compared are very large. This problem is addressed by *sequential* sampling methods [4, 28] (that have also been referred to as *adaptive* sampling methods [5]). The idea of sequential sampling is that when a difference between two frequencies is very large after only a few examples, then we can conclude that one of the probabilities is greater than the other with high confidence; we need not wait for the sample size specified by the Chernoff bound, which we have to when the frequencies are similar. Sequential sampling methods have been reported to reduce the required sample size by several orders of magnitude (*e.g.,* [10]).

In our algorithm (Table 1), we combine sequential sampling with the popular "loop reversal" technique found in many KDD algorithms. Instead of processing hypotheses one after another, and obtaining enough examples for each hypothesis to evaluate it sufficiently precisely, we keep obtaining examples (step 3b) and apply these to all remaining hypotheses simultaneously (step 3c). This strategy allows the algorithm to be easily implemented on top of database systems (assuming they are capable of drawing samples), *and* enables us to reach tighter bounds. After the statistics of each remaining hypothesis have been updated, the algorithm checks all remaining hypotheses and (step 3(e)i) outputs those

where it can be sufficiently certain that the number of better hypotheses is no larger than the number of hypotheses still to be found (so they can all become solutions), or (Step 3(e)ii) discards those hypotheses where it can be sufficiently certain that the number of better other hypotheses is at least the number of hypotheses still to be found (so it can be sure the current hypothesis does not need to be in the solutions). When the algorithm has gathered enough information to distinguish the good hypotheses that remain to be found from the bad ones with sufficient probability, it exits in step 3.

Indeed it can be shown that this strategy leads to a total error probability less than δ as required.

Table 1: Sequential sampling algorithm for the n-best hypotheses problem

---

Algorithm Generic Sequential Sampling. Input: n (number of desired hypotheses), ε and δ (approximation and confidence parameters). **Output:** n approximately best hypotheses (with confidence $1-\delta$).

1. **Let** $n_1 = n$ (the number of hypotheses that we still need to find) and **Let** $H_1 = H$ (the set of hypotheses that have, so far, neither been discarded nor accepted). **Let** $Q_0 = \emptyset$ (no sample drawn yet). **Let** $i = 1$ (loop counter).
2. **Let** M be the smallest number such that $E(M, \frac{\delta}{2|H|}) \leq \frac{\varepsilon}{2}$.
3. **Repeat until** $n_i = 0$ **Or** $|H_{i+1}| = n_i$ **Or** $E(i, \frac{\delta}{2|H_i|}) \leq \frac{\varepsilon}{2}$
   (a) **Let** $H_{i+1} = H_i$.
   (b) Query a random item of the database q_i. **Let** $Q_i = Q_{i-1} \cup \{q_i\}$.
   (c) Update the empirical utility $\hat{f}$ of the hypotheses in H_i.
   (d) **Let** H_i^* be the n_i hypotheses from H_i which maximize the empirical utility $\hat{f}$.
   (e) **For** $h \in H_i$ **While** $n_i > 0$ **And** $|H_i| > n_i$
      i. **If** $\hat{f}(h, Q_i) \geq E_h(i, \frac{\delta}{2M|H_i|}) + \max_{h_k \in H_i \setminus H_i^*} \left\{ \hat{f}(h_k, Q_i) + E_{h_k}(i, \frac{\delta}{2M|H_i|}) \right\} - \varepsilon$ **And** $h \in H_i^*$ (h appears good) **Then Output** hypothesis h and then **Delete** h from H_{i+1} and **let** $n_{i+1} = n_i - 1$. **Let** H_i^* be the new set of empirically best hypotheses.
      ii. **Else If** $\hat{f}(h, Q_i) \leq \min_{h_k \in H_i^*} \left\{ \hat{f}(h_k, Q_i) - E_{h_k}(i, \frac{\delta}{2M|H_i|}) \right\} - E_h(i, \frac{\delta}{2M|H_i|})$ (h appears poor) **Then Delete** h from H_{i+1}. **Let** H_i^* be the new set of empirically best hypotheses.
   (f) **Increment** i.
4. **Output** the n_i hypotheses from H_i which have the highest empirical utility.

Theorem 1 *The algorithm will output a group G of exactly n hypotheses such that, with confidence $1-\delta$, no other hypothesis in H has a utility which is more than ε higher than the utility of any hypothesis that has been returned:*

$$Pr[\exists h \in H \setminus G : f(h) > f_{min} + \varepsilon] \leq \delta \tag{2}$$

where $f_{min} = \min_{h' \in G}\{f(h')\}$; assuming that $|H| \geq n$.

The proof of Theorem 1 can be found in Appendix A

Theorem 2 (Termination) *If for any δ ($0 < \delta \leq 1$) and $\varepsilon > 0$ there is a number m such that $E(m, \delta) \leq \varepsilon$, then the algorithm can be guaranteed to terminate.*

Correctness of Theorem 2 follows immediately from Step 3e of the algorithm. Theorem 2 says that we can guarantee termination if the confidence interval vanishes for large numbers of examples. This is a rather weak assumption that is satisfied by most utility functions, as we will see in the next section.

4 Instantiations

In order to implement the algorithm for a given utility function we have to find a utility confidence interval $E(m, \delta)$ that satisfies Equation 1 for that specific f. In this section, we will introduce some terminology, and present a list of confidence intervals for the utility functions that are most commonly used in knowledge discovery systems. Since the database is constant, we abbreviate $f(h, D)$ as $f(h)$ throughout this section.

Most of the known utility functions refer to *confidence*, *accuracy*, *"statistical unusualness"*, *support* or *generality* of hypotheses. Let us quickly put these terms into perspective. Association rules and classification rules are predictive; for *some* database transaction they predict the value of an attribute given the values of some other attributes. For instance, the rule "beer= 1 $\rightarrow$ chips= 1" predicts that a customer transaction with attribute beer= 1 will also likely have the attribute chips= 1. However, when a customer does not buy beer, then the rule does not make any prediction. In particular, the rule does not imply that a customer who does not buy beer does not buy chips either. The number of transactions in the database for which the rule makes a correct prediction (in our example, the number of transactions which include beer and chips) is called the *support*, or the *generality*.

Among those transitions for which the rule does make a prediction, some predictions may be erroneous. The *confidence* is the fraction of correct predictions among those transactions for which a prediction is made. The *accuracy*, too, quantifies the probability of a hypothesis conjecturing a correct attribute. However, the term accuracy is typically used in the context of classification and refers to the probability of a correct classification for a future transaction whereas the confidence refers to the database (*i.e.,* the training data). From a sampling point of view, confidence and accuracy can be treated equally. In both cases, a relative frequency is measured on a small sample; from this frequency we want

to derive claims on the underlying probability. It does not make a difference whether this probability is itself a frequency on a much larger instance space (confidence) or a "real" probability (accuracy), defined with respect to an underlying distribution on instances.

Subgroups are of a more descriptive character. They describe that the value of an attribute differs from the global mean value within a particular subgroup of transactions without actually conjecturing the value of that attribute for a new transaction. The *generality* of a subgroup is the fraction of all transactions in the database that belong to that subgroup. The term *statistical unusualness* refers to the difference between the probability p_0 of an attribute in the whole database and the probability p of that attribute within the subgroup. Usually, subgroups are desired to be both general (large g) and statistically unusual (large $|p_0 - p|$). There are many possible utility functions for subgroup discovery which trade generality against unusualness [18]. Unfortunately, none of these functions can be expressed as the average (over all transactions) of an instance utility function. But, in Sections 4.2 through 4.4 we will show how instantiations of the GSS algorithm can solve sampling problems for these functions.

We would like to conclude this subsection with a remark on whether a sample should be drawn with or without replacement. When the utility function is defined with respect to a finite database, it is, in principle, possible to draw the sample without replacement. When the sample size reaches the database size, we can be certain to have solved the real, not just the approximate, n best hypothesis problem. So it should be possible to give a tighter utility confidence bound when the sample is drawn without replacement. Consider the simple case when the utility is a probability. When the sample is drawn with replacement, the relative frequency corresponding to the target probability is governed by the binomial distribution whereas, when the sample is drawn without replacement, it is governed by the hyper-geometrical distribution and we can specify a tighter bound. However, for sample sizes in the order of magnitude that we envision, the only feasible way of calculating both the hyper-geometrical distribution and the binomial distribution is to use a normal approximation. But the normal approximation of both distributions are equal and so we cannot realize the small advantage that drawing without replacement seems to promise. The same situation arises with other utility functions.

4.1 Instance-Averaging Functions

This simplest form of a utility function is the average, over all example instances, of some instance utility function $f_{inst}(h, q_i)$ where $q_i \in D$. The utility is then defined as $f(h) = \frac{1}{|D|} \sum_{i=1}^{|} D| f_{inst}(h, q_i)$ (the average over the whole database) and the estimated utility is $\hat{f}(h, Q_m) = \frac{1}{m} \sum_{q_i \in Q_m} f_{inst}(h, q_i)$ (average over the example queries). An easy example of an instance-averaging utility is classification accuracy (where $f_{inst}(h, q_i)$ is 0 or 1). Besides being useful by itself, this class of utility functions serves as an introductory example of how confidence intervals can be derived. We assume that the possible range of utility values lies between 0 and Λ. In the case of classification accuracy, Λ equals one.

We can use the *Hoeffding* inequality [14] to bound the chance that an arbitrary

(bounded) random variable X takes a value which is far away from its expected value $E(X)$ (Equation 3). When X is a relative frequency and $E(X)$ the corresponding probability, then we know that $\Lambda = 1$. This special case of the Hoeffding inequality is called *Chernoff's* inequality.

$$Pr[|X - E(X)| \leq \varepsilon] \geq 1 - 2\exp\left\{-2m\frac{\varepsilon^2}{\Lambda^2}\right\} \tag{3}$$

We now need to define a confidence interval that satisfies Equation 1, where the Hoeffding inequality serves as a tool to prove Equation 1. We can easily see that Equation 4 satisfies this condition.

$$E(m,\delta) = \sqrt{\frac{\Lambda^2}{2m}\log\frac{2}{\delta}} \tag{4}$$

In Equation 5 we insert Equation 4 into Equation 1. We apply the Hoeffding inequality (Equation 3) in Equation 6 and obtain the desired result in Equation 7.

$$\begin{aligned}
& Pr\left[|\hat{f}(h,Q_m) - f(h)| > E(m,\delta)\right] \\
&= Pr\left[|\hat{f}(h,Q_m) - f(h)| > \sqrt{\frac{\Lambda^2}{2m}\log\frac{2}{\delta}}\right] && (5)\\
&\leq 2\exp\left\{-2m\frac{\left(\sqrt{\frac{\Lambda^2}{2m}\log\frac{2}{\delta}}\right)^2}{\Lambda^2}\right\} && (6)\\
&\leq 2\exp\left\{-\log\frac{2}{\delta}\right\} = \delta && (7)
\end{aligned}$$

For implementation purposes, the Hoeffding inequality is less suited since it it not very tight. For large m, we can replace the Hoeffding inequality by the normal distribution, referring to the central limit theorem. $\hat{f}(h,Q_m) - f(h)$ is a random variable with mean value 0; we further know that $\hat{f}(h,Q_m)$ is bounded between zero and Λ. In order to calculate the normal distribution, we need to refer to the true variance of our random variable. In step 3, the variance is not known since we do not refer to any particular hypothesis. We can only bound the variance from above and thus obtain a confidence interval E which is tighter than Hoeffding's/Chernoff's inequality and still satisfies Equation 1. $\hat{f}(h,Q_m)$ is the average of m values, namely $\frac{1}{m}\sum_{i=1}^{m}\hat{f}_{inst}(h,q_i)$. The empirical variance $s_{\hat{f}(h,Q_m)-f(h)} = \frac{1}{m}\sqrt{\sum_{i=1}^{m}(\hat{f}_{inst}(h,q_i) - \hat{f}(h,Q_m))^2}$ is maximized when $\hat{f}(h,Q_m) = \frac{\Lambda}{2}$ and the individual $\hat{f}_{inst}(h,q_i)$ are zero for half the instances q_i and Λ for the other half of all instances. In this case, $s \leq \frac{\Lambda}{2\sqrt{m}}$. Consequently, $\frac{2\sqrt{m}(\hat{f}(h,Q_m)-f(h))}{\Lambda}$ is governed by the standard normal distribution which implies that Equation 8 satisfies Equation 1. z is the inverse standard normal distribution that can be looked up in a table.

$$E(m,\delta) \;=\; z_{1-\frac{\delta}{2}} \cdot \frac{\Lambda}{2\sqrt{m}} \tag{8}$$

In Steps 3(e)i and 3(e)ii, we refer to specific hypotheses h and can therefore determine the empirical variance of $\hat{f}(h, Q_m)$. We can define $E_h(m,\delta)$ as in Equation 10.

$$E(m,\delta) \;=\; z_{1-\frac{\delta}{2}} \cdot s_h \tag{9}$$

$$\;=\; z_{1-\frac{\delta}{2}} \frac{1}{m} \sqrt{\sum_{i=1}^{m} (f_{inst}(h,q_i) - \hat{f}(h,Q_i))^2} \tag{10}$$

Note that we have simplified the situation a little. We have confused the true variance σ (the average squared distance from the true mean $f(h)$) and the empirical variance s_h in Equation 10. The empirical variance possesses one degree of freedom less than the true variance and, to be quite accurate, we would have to refer to Student's t distribution rather than the normal distribution. Empirically, we observed that the algorithm does not start to output or discard any hypotheses until the sample size has reached the order of a hundred. In this region, Student's distribution can well be approximated by the normal distribution and we can keep this treatment (and the implementation) simple.

Let us now determine a worst-case bound on m (the number of queries that our sampling algorithm issues). The algorithm exits the for loop (at the latest) when $E\left(m, \frac{\delta}{2|H|}\right) \leq \frac{\varepsilon}{2}$. We can show that this is the case with certainty when $m \geq \frac{2\Lambda^2}{\varepsilon^2} \log \frac{|H|}{2\delta}$. In Equation 11, we expand our definition of E. The Λ and log-terms cancel out in Equation 12; we can bound the confidence interval to $\frac{\varepsilon}{2}$ in Equation 13 as required for the algorithm to exit in step 3e.

$$E\left(\frac{2\Lambda^2}{\varepsilon^2} \log \frac{4|H|}{\delta}, \frac{\delta}{2|H|}\right) \;=\; \sqrt{\frac{\Lambda^2}{2\left(\frac{2\Lambda^2}{\varepsilon^2} \log \frac{4|H|}{\delta}\right)} \log \frac{2}{\left(\frac{\delta}{2|H|}\right)}} \tag{11}$$

$$\;=\; \sqrt{\frac{\varepsilon^2}{4} \frac{\log \frac{4|H|}{\delta}}{\log \frac{4|H|}{\delta}}} \tag{12}$$

$$\;=\; \frac{\varepsilon}{2} \tag{13}$$

But note that our algorithm will generally terminate much earlier; firstly, because we use the normal distribution (for large m) rather than the Hoeffding approximation and, secondly, our sequential sampling approach will terminate much earlier when the n best hypotheses differ considerably from many of the "bad" hypotheses. The worst case occurs only when all hypotheses in the hypothesis space are equally good which makes it much more difficult to identify the n best ones.

4.2 Functions that are Linear in g and $(p - p_0)$

The first class of nontrivial utility functions that we study weight the generality g of a subgroup and the deviation of the probability of a certain feature p from the default probability p_0 equally [22]. Hence, these functions multiply generality and distributional unusualness of subgroups. Alternatively, we can use the absolute distance $|p - p_0|$ between probability p and default probability p_0. The multi-class version of this function is $g\frac{1}{c}\sum_c |p_i - p_{0_i}|$ where p_{0_i} is the default probability for class i.

Theorem 3 *Let*

1. $f(h) = g(p - p_0)$ *and* $\hat{f}(h, Q) = \hat{g}(\hat{p} - p_0)$ or
2. $f(h) = g|p - p_0|$ *and* $\hat{f}(h, Q) = \hat{g}|\hat{p} - p_0|$ or
3. $f(h) = g\frac{1}{c}\sum_{i=1}^{c} |p_i - p_{0_i}|$ *and* $\hat{f}(h, Q) = \hat{g}\frac{1}{c}\sum_{i=1}^{c} |\hat{p}_i - p_{0_i}|$.

Then $Pr[|\hat{f}(h, Q_m) - f(h)| \leq E(m, \delta)] \geq 1 - \delta$ *when*

$$\text{small } m: \quad E(m, \delta) = 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} \tag{14}$$

$$\text{large } m: \quad E(m, \delta) = \frac{z_{1-\frac{\delta}{4}}}{\sqrt{m}} + \frac{(z_{1-\frac{\delta}{4}})^2}{4m} \tag{15}$$

$$E_h(m, \delta) = z_{1-\frac{\delta}{4}}(s_g + s_p + z_{1-\frac{\delta}{4}} s_g s_p) \tag{16}$$

Proof. (3.1) In Equation 17, we insert Equation 14 into Equation 1. We refer to the union bound in Equation 18. Then, we exploit that $\varepsilon^2 \leq \varepsilon$ for $\varepsilon \leq 1$ in Equation 19. The simple observation that $g \leq 1$ and $(p - p_0) \leq 1$ leads to Equation 20. Equations 21 and 22 are based on elementary transformations. In Equation 23, we refer to the union bound again. The key observation here is that ab cannot be greater than $(c + \varepsilon)(d + \varepsilon)$ unless at least $a > c + \varepsilon$ or $b > d + \varepsilon$. The Chernoff inequality (which is a special case of the Hoeffding inequality 3 for $\Lambda = 1$) takes us to Equation 24.

$$\begin{aligned}
&Pr[|\hat{f}(h, Q_m) - f(h)| > E(m, \delta)] \\
&= Pr\left[|\hat{g}(\hat{p} - p_0) - g(p - p_0)| > 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] && (17)\\
&\leq 2Pr\left[\hat{g}(\hat{p} - p_0) - g(p - p_0) > 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] && (18)\\
&\leq 2Pr\left[\hat{g}(\hat{p} - p_0) - g(p - p_0) > 2\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)^2\right] && (19)
\end{aligned}$$

$$
\begin{aligned}
&\leq 2Pr\Bigg[\hat{g}(\hat{p}-p_0)-g(p-p_0) \\
&\qquad > g\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}+(p-p_0)\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}+\left(\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)^2\Bigg] && (20)\\
&\leq 2Pr\Bigg[\hat{g}(\hat{p}-p_0)-g(p-p_0) \\
&\qquad > \left(g+\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\left(p-p_0+\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)-g(p-p_0)\Bigg] && (21)\\
&\leq 2Pr\left[\hat{g}(\hat{p}-p_0) > \left(g+\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\left(p-p_0+\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\right] && (22)\\
&\leq 4Pr\left[\hat{q} > \left(q+\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\right] = 4Pr\left[\hat{q}-q > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] && (23)\\
&\leq 4\exp\left\{-2m\frac{1}{2m}\log\frac{4}{\delta}\right\}=\delta && (24)
\end{aligned}
$$

Let us now prove the normal approximation of the above confidence bound. We start in Equation 25 by inserting Equation 16 into Equation 1. s_g and s_p denote the variances of g and p, respectively. We also cover Equation 15 in this proof. The variances can be bounded from above: $s_g, s_p \leq \frac{1}{2\sqrt{m}}$. Hence, $z_{1-\frac{\delta}{4}}(s_g+s_p+z_{1-\frac{\delta}{4}}s_g s_p) \leq 2z_{1-\frac{\delta}{4}}\frac{1}{2\sqrt{m}}+\left(z_{1-\frac{\delta}{4}}\frac{1}{2\sqrt{m}}\right)^2 \leq \frac{3z_{1-\delta/4}}{2\sqrt{m}}$. We expand the definition of f and apply the union bound in Equation 26. Equation 27 follows from $g \leq 1$ and $p-p_0 \leq 1$ and Equation 28 is just a factorization of $g+z_{1-\frac{\delta}{4}}s_g$. Again, note that ab cannot be greater than $(c+\varepsilon)(d+\varepsilon)$ unless $a > c+\varepsilon$ or $b > d+\varepsilon$. Applying the union bound in 29 proves the claim.

$$
\begin{aligned}
&Pr[|\hat{f}(h,Q_m)-f(h)| > z_{1-\frac{\delta}{4}}(s_g+s_p+z_{1-\frac{\delta}{4}}s_g s_p)] \\
&\quad \leq 2Pr\left[\hat{g}(\hat{p}-p_0)-g(p-p_0) > +z_{1-\frac{\delta}{4}}(s_g+s_p+z_{1-\frac{\delta}{4}}s_g s_p)\right] && (25)\\
&\quad \leq 2Pr\left[\hat{g}(\hat{p}-p_0)-g(p-p_0) > gz_{1-\frac{\delta}{4}}s_g+(p-p_0)z_{1-\frac{\delta}{4}}s_p+\left(z_{1-\frac{\delta}{4}}\right)^2 s_g s_p\right] && (26)\\
&\quad \leq 2Pr\left[\hat{g}(\hat{p}-p_0) > \left(g+z_{1-\frac{\delta}{4}}s_g\right)\left(p-p_0+z_{1-\frac{\delta}{4}}s_p\right)\right] && (27)\\
&\quad \leq 2\left(Pr\left[\hat{g} > g+z_{1-\frac{\delta}{4}}s_g\right]+Pr\left[\hat{p} > p+z_{1-\frac{\delta}{4}}s_p\right]\right) && (28)\\
&\quad \leq 2\left(\frac{\delta}{4}+\frac{\delta}{4}\right)=\delta && (29)
\end{aligned}
$$

This completes the proof for Theorem (3.1).

(3.2) Instead of having to estimate p, we need to estimate the random variable $|p-p_0|$. We define s_p to be the empirical variance of $|p-p_0|$. Since this value is bounded between zero

and one, all the arguments which we used in the last part of this proof apply analogously.

(3.3) Here, the random variable is $\frac{1}{c}\sum_{i=1}^{c}|p_i - p_{0_i}|$. This variable is also bounded between zero and one and so the proof is analogous to case (3.1). This completes the proof ∎

Theorem 4 *For all functions $f(h)$ covered by Theorem 3, the sampling algorithm will terminate after at most*

$$m = \frac{18}{\varepsilon^2}\log\frac{8|H_i|}{\delta} \tag{30}$$

database queries (but usually much earlier).

Proof. The algorithm terminates in step 3 when $E\left(i, \frac{\delta}{2|H|}\right) \leq \frac{\varepsilon}{2}$. We will show that this is always the case when $i \geq m = \frac{16}{\varepsilon^2}\log\frac{\sqrt{6}|H_i|}{\sqrt{\delta}}$. We insert the sample bound (Equation 30) into the definition of E for linear functions (Equation 14); after the log-terms rule out each other in Equation 31 we obtain the desired bound of $\frac{\varepsilon}{2}$.

$$E\left(\frac{18}{\varepsilon^2}\log\frac{8|H|}{\delta}, \frac{\delta}{2|H|}\right) = 3\sqrt{\frac{1}{2\left(\frac{18}{\varepsilon^2}\log\frac{8|H|}{\delta}\right)}\log\frac{4}{\left(\frac{\delta}{2|H|}\right)}} \tag{31}$$

$$\leq 3\sqrt{\frac{\varepsilon^2}{36}} = \frac{\varepsilon}{2} \tag{32}$$

This completes the proof. ∎

4.3 Functions with squared terms

Squared terms [30] are introduced to put more emphasis on the the difference between p and the default probability.

Theorem 5 *Let*

1. $f(h) = g^2(p - p_0)$ *and* $\hat{f}(h, Q) = \hat{g}^2(\hat{p} - p_0)$ or
2. $f(h) = g^2|p - p_0|$ *and* $\hat{f}(h, Q) = \hat{g}^2|\hat{p} - p_0|$ or
3. $f(h) = g^2\frac{1}{c}\sum_{i=1}^{c}|p_i - p_{0_i}|$ *and* $\hat{f}(h, Q) = \hat{g}^2\frac{1}{c}\sum_{i=1}^{c}|\hat{p}_i - p_{0_i}|$ or

Then $Pr[\hat{f}(h, Q_m) - f(h, Q) \leq E(m, \delta)] \geq 1 - \delta$ *when*

$$\text{small } m: \quad E(m,\delta) = \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{2}} + 3\left(\frac{1}{2m}\log\frac{4}{\delta}\right) + 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} \tag{33}$$

$$\text{large } m: \quad E(m,\delta) = \frac{3}{2\sqrt{m}}z_{1-\frac{\delta}{2}} + \frac{m+\sqrt{m}}{4m\sqrt{m}}(z_{1-\frac{\delta}{2}})^2 + \frac{1}{8m\sqrt{m}}(z_{1-\frac{\delta}{2}})^3 \tag{34}$$

$$E_h(m,\delta) = 2s_g z_{1-\frac{\delta}{2}} + s_g^2(z_{1-\frac{\delta}{2}})^2 + s_p z_{1-\frac{\delta}{2}} + 2s_g s_p(z_{1-\frac{\delta}{2}})^2 + s_p s_g^2(+z_{1-\frac{\delta}{2}}) \tag{35}$$

Proof. (5.1) $f(h) = g^2(p - p_0)$. As usual, we start in Equation 37 by combining the definition of E (Equation 33) with Equation 1 which specifies the property of E that we would like to prove. In Equation 37 we exploit that $g \leq 1$ and $p - p_0 \leq 1$. In Equation 38 we add $g^2(p - p_0)$ to both sides of the inequality an start factorizing. In Equation 39 we have identified three factors. The observation that $a^2 b$ cannot be greater than $(c+\varepsilon)(c+\varepsilon)(d+\varepsilon)$ unless at least $a > c+\varepsilon$ or $b > d+\varepsilon$ and the union bound lead to Equation 40; the Chernoff inequality completes this part of the proof.

$$
\begin{aligned}
& Pr[|\hat{f}(h, Q_m) - f(h)| > E(m, \delta)] \\
&= Pr\left[|\hat{g}^2(\hat{p} - p_0) - g^2(p - p_0)| > \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{2}} + 3\left(\frac{1}{2m}\log\frac{4}{\delta}\right) + 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] && (36) \\
&\leq 2Pr\left[\hat{g}^2(\hat{p} - p_0) - g^2(p - p_0) > g^2\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} + 2g(p - p_0)\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} + \right. \\
&\qquad \left. 2g\left(\frac{1}{2m}\log\frac{4}{\delta}\right) + (p - p_0)\left(\frac{1}{2m}\log\frac{4}{\delta}\right) + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{2}}\right] && (37) \\
&\leq 2Pr\left[\hat{g}^2(\hat{p} - p_0) > \left(g^2 + 2g\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)\right)\left(p - p_0 + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\right] && (38) \\
&\leq 2Pr\left[\hat{g}^2(\hat{p} - p_0) > \left(g + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\left(g + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\left(p - p_0 + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\right] && (39) \\
&\leq 2\left(Pr\left[\hat{g} - g > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] + Pr\left[\hat{p} - p > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right]\right) && (40) \\
&\leq 4\exp\left\{-2m\left(\frac{1}{2m}\log\frac{4}{\delta}\right)\right\} = \delta && (41)
\end{aligned}
$$

Let us now look at the normal approximation. First, we will make sure that Equation 34 is a special case of Equation 35 (variance bounded from above). The variance of both g and p is at most $\frac{1}{2\sqrt{m}}$. This takes us from Equation 35 to Equation 42. Equation 43 equals Equation 34.

$$
\begin{aligned}
& 2s_g z_{1-\frac{\delta}{2}} + s_g^2 (z_{1-\frac{\delta}{2}})^2 + s_p z_{1-\frac{\delta}{2}} + 2s_g s_p (z_{1-\frac{\delta}{2}})^2 + s_p s_g^2 (z_{1-\frac{\delta}{2}})^3 \\
&\leq \frac{1}{\sqrt{m}} z_{1-\frac{\delta}{2}} + \frac{1}{4m}(z_{1-\frac{\delta}{2}})^2 + \frac{1}{2\sqrt{m}} z_{1-\frac{\delta}{2}} + \frac{1}{\sqrt{m}}(z_{1-\frac{\delta}{2}})^2 + \frac{1}{8m\sqrt{m}}(z_{1-\frac{\delta}{2}})^3 && (42) \\
&= \frac{3}{2\sqrt{m}} z_{1-\frac{\delta}{2}} + \frac{m + \sqrt{m}}{4m\sqrt{m}}(z_{1-\frac{\delta}{2}})^2 + \frac{1}{8m\sqrt{m}}(z_{1-\frac{\delta}{2}})^3 && (43)
\end{aligned}
$$

In Equation 44 we want to see if the normal approximation (Equation 35) satisfies the requirement of Equation 1. We add $g^2(p - p_0)$ to both sides of the equation and start

factorizing the right hand side of the inequality in Equations 45 and 46. The union bound takes us to Equation 47; Equation 48 proves the claim.

$$
\begin{aligned}
& Pr[|\hat{f}(h, Q_m) - f(h)| > E(m, \delta)] && (44)\\
&= Pr[|\hat{g}^2(\hat{p} - p_0) - g^2(p - p_0)| > 2s_g z_{1-\frac{\delta}{2}} + s_g^2 (z_{1-\frac{\delta}{2}})^2 \\
&\quad + s_p z_{1-\frac{\delta}{2}} + 2s_g s_p (z_{1-\frac{\delta}{2}})^2 + s_p s_g^2 (z_{1-\frac{\delta}{2}})^3] \\
&\leq 2Pr\left[\hat{g}^2(\hat{p} - p_0) > \left(g^2 + 2g s_g z_{1-\frac{\delta}{2}} + s_g^2 (z_{1-\frac{\delta}{2}})^2\right)\left(p + s_p z_{1-\frac{\delta}{2}}\right)\right] && (45)\\
&\leq 2Pr\left[\hat{g}^2(\hat{p} - p_0) > (g + s_g z_{1-\frac{\delta}{2}})(g + s_g z_{1-\frac{\delta}{2}})(p + s_p z_{1-\frac{\delta}{2}})\right] && (46)\\
&\leq 2\left(Pr\left[\hat{g} - g > s_g z_{1-\frac{\delta}{2}}\right] + Pr\left[\hat{p} - p > s_p z_{1-\frac{\delta}{2}}\right]\right) && (47)\\
&\leq 2\left(\frac{\delta}{4} + \frac{\delta}{4}\right) = \delta && (48)
\end{aligned}
$$

This proves case (5.1). For cases (5.2) and (5.3), note that the random variables $|p - p_0|$ and $\frac{1}{c}\sum_{i=1} c(p - p_0)$ (both bounded between zero and one) play the role of p and the proof is analogous to the first case (5.1). ∎

Theorem 6 *For all functions $f(h)$ covered by Theorem 5, the sampling algorithm will terminate after at most*

$$m = \frac{98}{\varepsilon^2} \log \frac{8|H_i|}{\delta} \tag{49}$$

database queries (but usually much earlier).

Proof. The algorithm terminates in step 3 when $E\left(i, \frac{\delta}{2|H|}\right) \leq \frac{\varepsilon}{2}$. The utility functions of Theorem 5 are bounded between zero and one. Hence, we can assume that $\varepsilon \leq 1$ since otherwise the algorithm might just return n arbitrarily poor hypotheses and still meet the requirements of Theorem 1. This means that the algorithm cannot exit until $E(m, \frac{\delta}{2|H_i|}) \leq \frac{1}{2}$ (or n hypotheses have been returned). For $E(m, \frac{\delta}{2|H_i|})$ to be $\frac{1}{2}$ or less, each of the three terms in Equation 33 has to be below 1. Note that if $\varepsilon < 1$ then $\varepsilon^2 < \varepsilon$. We can therefore bound E as in Equation 51.

$$
\begin{aligned}
E(m, \delta) &= \left(\frac{1}{2m} \log \frac{4}{\delta}\right)^{\frac{3}{2}} + 3\left(\frac{1}{2m} \log \frac{4}{\delta}\right) + 3\sqrt{\frac{1}{2m} \log \frac{4}{\delta}} && (50)\\
&< 7\sqrt{\frac{1}{2m} \log \frac{4}{\delta}} && (51)
\end{aligned}
$$

Now we will show that E lies below $\frac{\varepsilon}{2}$ when m reaches the bound described in Equation 49. We insert the sample bound into the exit criterion in Equation 52. The log-terms rule out each other and the result is $\frac{\varepsilon}{2}$ as desired.

$$E\left(\frac{98}{\varepsilon^2}\log\frac{8|H|}{\delta}, \frac{\delta}{2|H|}\right) < 7\sqrt{\frac{1}{2\left(\frac{98}{\varepsilon^2}\log\frac{8|H|}{\delta}\right)}\log\frac{4}{\left(\frac{\delta}{2|H|}\right)}} \quad (52)$$

$$\leq 7\sqrt{\frac{\varepsilon^2}{4\cdot 49}} = \frac{\varepsilon}{2} \quad (53)$$

This completes the proof. ∎

4.4 Functions Based on the Binomial Test

The Binomial test heuristic [16] is based on elementary considerations. Suppose that the probability p is really equal to p_0 (*i.e.,* the corresponding subgroup is really uninteresting). How likely is it, that the subgroup with generality g displays a frequency of $\hat{p}$ on the sample Q with a greater difference $|\hat{p} - p_0|$? For large $|Q| \times g$, $(\hat{p} - p_0)$ is governed by the normal distribution with mean value of zero and variance at most $\frac{1}{2\sqrt{m}}$. The probability density function of the normal distribution is monotonic, and so the resulting confidence is order-equivalent to $\sqrt{m}(p - p_0)$ (m being the support) which is factor equivalent to $\sqrt{g}(p - p_0)$. Several variants of this utility function have been used.

Theorem 7 *Let*

1. $f(h) = \sqrt{g}(p - p_0)$ *and* $\hat{f}(h, Q) = \sqrt{\hat{g}}(\hat{p} - p_0)$ or
2. $f(h) = \sqrt{g}|p - p_0|$ *and* $\hat{f}(h, Q) = \sqrt{\hat{g}}|\hat{p} - p_0|$ or
3. $f(h) = \sqrt{g}\frac{1}{c}\sum_{i=1}^{c}|p_i - p_{0_i}|$ *and* $\hat{f}(h, Q) = \sqrt{\hat{g}}\frac{1}{c}\sum_{i=1}^{c}|\hat{p}_i - p_{0_i}|$.

Then $Pr[|\hat{f}(h, Q_m) - f(h)| \leq E(m, \delta)] \geq 1 - \delta$ *when*

$$\text{small } m: \quad E(m,\delta) = \sqrt[2]{\frac{1}{2m}\log\frac{4}{\delta}} + \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{4}} \quad (54)$$

$$\text{large } m: \quad E(m,\delta) = \sqrt{\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}} + \frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}} + \left(\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}\right)^{3/2} \quad (55)$$

$$E_h(m,\delta) = \sqrt{s_g z_{1-\frac{\delta}{4}}} + s_p z_{1-\frac{\delta}{4}} + \sqrt{s_g z_{1-\frac{\delta}{4}}} s_p z_{1-\frac{\delta}{4}} \quad (56)$$

Proof. (7.1) In Equation 57, we insert Equation 54 into Equation 1 (the definition of E). We refer to the union bound in Equation 58 and exploit that $\sqrt{g} \leq 1$ and $p - p_0 \leq 1$. As usual, we factor the right hand side of the inequality in Equation 59 and use the union bound in Equation 60. Now in Equation 61 we weaken the inequality a little. Note that $\sqrt[4]{x} \geq \sqrt{\sqrt{x} - y}$ when $y > 0$. Hence, subtracting the lengthy term in Equation 61 decreases the probability of the inequality (which we want to bound from above). The reason why we

subtract this term is that we want to apply the binomial equation and factor $\sqrt{g+\varepsilon}-\sqrt{g}$. We do this in the following steps 62 and 63 which are perhaps a little hard to check without a computer algebra system. Adding $\sqrt{g}$ and taking both sides of the inequality to the square leads to Equation 64, the Chernoff inequality leads to the desired result of δ.

$$
\begin{aligned}
& Pr[|\hat{f}(h,Q_m) - f(h)| > E(m,\delta)] \\
&= 2Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) - \sqrt{g}(p-p_0) > \sqrt[2]{\frac{1}{2m}\log\frac{4}{\delta}} + \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{4}}\right] && (57)\\
&\leq 2Pr\Bigg[\sqrt{\hat{g}}(\hat{p}-p_0) - \sqrt{g}(p-p_0) \\
&\qquad > \sqrt{g}\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} + (p-p_0)\sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{4}}\Bigg] && (58)\\
&\leq 2Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) > \left(\sqrt{g} + \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}}\right)\left(p - p_0 + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\right] && (59)\\
&\leq 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}}\right] + 2Pr\left[\hat{p} - p > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] && (60)\\
&\leq 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} - 2\left(\sqrt{\sqrt{g^2 + g\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}} - \sqrt{g^2}}\right)}\right] \\
&\qquad + 2\exp\left\{-2m\frac{1}{2m}\log\frac{4}{\delta}\right\} && (61)\\
&= 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{2g + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}} - 2\sqrt{g\left(g + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)}}\right] + \frac{\delta}{2} && (62)\\
&= 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{g + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}} - \sqrt{g}\right] + \frac{\delta}{2} && (63)\\
&= 2Pr\left[\hat{g} - g > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] + \frac{\delta}{2} = \delta && (64)
\end{aligned}
$$

Now we still need to prove the normal approximations (Equations 55 and 56. As usual, we would like Equation 55 to be a special case of Equation 55 with the variances bounded from above. Equation 65 confirms that this is the case since $s_{p,g} \leq \frac{1}{2\sqrt{m}}$.

$$
\sqrt{s_g z_{1-\frac{\delta}{4}}} + s_p z_{1-\frac{\delta}{4}} + \sqrt{s_g z_{1-\frac{\delta}{4}}} s_p z_{1-\frac{\delta}{4}} \leq \sqrt{\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}} - \frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}} + \sqrt{\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}}\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}} \qquad (65)
$$

This derivation is quite analogous to the previous one. We multiply the terms on the right hand side by factor which are less or equal to one (Equation 67) and then factor the right hand side (Equation 68). We subtract a small number from $s_g z_{1-\frac{\delta}{4}}$ in Equation 69 and factor $\sqrt{\hat{g}} - \sqrt{g}$ in Equation 70 and Equation 71. Basic manipulations and the Chernoff inequality complete the proof in Equation 73.

$$
\begin{aligned}
& Pr[|\hat{f}(h, Q_m) - f(h)| > E(m, \delta)] \\
& \leq 2Pr\left[\sqrt{\hat{g}}(\hat{p} - p_0) - \sqrt{g}(p - p_0) > \sqrt{s_g z_{1-\frac{\delta}{4}}} + s_p z_{1-\frac{\delta}{4}} + \sqrt{s_g z_{1-\frac{\delta}{4}}} s_p z_{1-\frac{\delta}{4}}\right] && (66) \\
& \leq 2Pr\left[\sqrt{\hat{g}}(\hat{p} - p_0) - \sqrt{g}(p - p_0) > \sqrt{g s_g z_{1-\frac{\delta}{4}}} - (p - p_0) s_p z_{1-\frac{\delta}{4}} + s_g z_{1-\frac{\delta}{4}} \sqrt{s_p z_{1-\frac{\delta}{4}}}\right] && (67) \\
& \leq 2Pr\left[\sqrt{\hat{g}}(\hat{p} - p_0) > \left(\sqrt{g} + \sqrt{s_g z_{1-\frac{\delta}{4}}}\right)\left(p - p_0 + s_p z_{1-\frac{\delta}{4}} s_p\right)\right] && (68) \\
& \leq 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{s_g z_{1-\frac{\delta}{4}}}\right] + 2Pr\left[\hat{p} - p > s_p z_{1-\frac{\delta}{4}}\right] && (69) \\
& \leq 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{s + g z_{1-\frac{\delta}{4}} - 2\left(\sqrt{g^2 + g s_g z_{1-\frac{\delta}{4}}} + \sqrt{g^2}\right)}\right] + \frac{\delta}{2} && (70) \\
& \leq 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{2g + s_g z_{1-\frac{\delta}{4}} - 2\left(g(g + s_g z_{1-\frac{\delta}{4}})\right)}\right] + \frac{\delta}{2} && (71) \\
& \leq 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{g + s_g z_{1-\frac{\delta}{4}}}\right] + \frac{\delta}{2} && (72) \\
& \leq 2Pr\left[\hat{g} - g > s_g z_{1-\frac{\delta}{4}}\right] + \frac{\delta}{2} = \delta && (73)
\end{aligned}
$$

This completes the proof for Theorem (7.1). The proofs of cases (7.2) and (7.2) are analogous; instead of p we need to estimate $|p - p_0|$ and $\frac{1}{c}\sum_{i=1}(p_i - p_{0_i})$, respectively. Both random variables are bounded between zero and one and so all our previous arguments apply. This completes the proof of Theorem 7. ∎

Theorem 8 *For all functions $f(h)$ covered by Theorem 7, the sampling algorithm will terminate after at most*

$$m = \frac{648}{\varepsilon^2} \log \frac{8|H_i|}{\delta} \tag{74}$$

database queries (but usually much earlier).

Proof. Proving the last theorem was a little tiring, so let us first see how we can find a simpler bound for E (Equation 54) which will help us to prove a sample size bound without having to think too hard. The middle term of Equation 55 dominates the expression since, for $\varepsilon \leq 1$ it is true that $\sqrt[4]{\varepsilon} \geq \sqrt{\varepsilon} \geq \varepsilon^{\frac{3}{4}}$. Hence, Equation 75 provides us with an easier bound.

$$\sqrt[2]{\frac{1}{2m}\log\frac{4}{\delta}} + \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{4}} \leq 3\sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} \tag{75}$$

The algorithm terminates in step 3 when $E(m, \frac{\delta}{2|H|}) \leq \frac{\varepsilon}{2}$. Considering the sample bound in Equation 74, Equation 76 proves that this is the case with guarantee. Note that, since we bounded the confidence interval quite sloppily, we expect the algorithm to terminate considerably earlier.

$$E\left(\frac{648}{\varepsilon^2}\log\frac{8|H|}{\delta}, \frac{\delta}{2|H|}\right) < 3\sqrt[4]{\frac{1}{2\left(\frac{648}{\varepsilon^2}\log\frac{8|H|}{\delta}\right)}\log\frac{4}{\left(\frac{\delta}{2|H|}\right)}} \tag{76}$$

$$\leq 3\sqrt[4]{\frac{\varepsilon^2}{16\cdot 81}} = \frac{\varepsilon}{2} \tag{77}$$

This completes the proof. ∎

4.5 Negative Results

Several independent impurity criteria have led to utility functions which are factor-equivalent to $f(h) = \frac{g}{1-g}(p-p_0)^2$; *e.g.,* Gini diversity index and twoing criterion [2], and the chi-square test [22]. Note that it is also order-equivalent to the utility measure used in Inferrule [26]. Unfortunately, this utility function is not bounded and a few examples that have not been included in the sample can impose dramatic changes on the values of this function. This motivates our negative result.

Theorem 9 *There is no algorithm that satisfies Theorem 1 when* $f(h) = \frac{g}{1-g}(p-p_0)^2$.

Proof. We need to show that $\hat{f}(h, Q_m) - f(h)$ is unbounded for any finite m. This is easy since $\frac{g+\varepsilon}{1-(g+\varepsilon)} - \frac{g}{1-g}$ goes to infinity when g approaches 1 or $1-\varepsilon$ (Equation 78).

$$\frac{g+\varepsilon}{1-(g+\varepsilon)} - \frac{g}{1-g} = \frac{\varepsilon}{(g+\varepsilon-1)(g-1)} \tag{78}$$

This implies that, even after an arbitrarily large sample has been observed (that is smaller than the whole database), the utility of a hypothesis with respect to the sample can be arbitrarily far from the true utility. But one may argue that demanding $\hat{f}(h, Q)$ to be within an additive constant ε is overly restricted. However, the picture does not change when we require $\hat{f}(h, Q)$ only to be within a multiplicative constant, since $\frac{g+\varepsilon}{1-(g+\varepsilon)} / \frac{g}{1-g}$ goes to infinity when $g+\varepsilon$ approach 1 or g approaches zero (Equation 79).

$$\frac{g+\varepsilon}{1-(g+\varepsilon)} / \frac{g}{1-g} = \frac{(g+\varepsilon)(1-g)}{g(1-g-\varepsilon)} \tag{79}$$

This means that no sample suffices to bound $\hat{f}(h, Q_m) - f(h)$ with high confidence when a particular $\hat{f}(h, Q_m)$ is measured. When a sampling algorithm uses all but very few database transactions as sample, then the few remaining examples may still impose huge changes on $\hat{f}(h, Q_m)$ which renders the use of sampling algorithms prohibitive. This completes the proof. ∎

5 Experiments

In our experiments, we want to study the order of magnitude of examples which are required by our algorithm for realistic tasks. Furthermore, we want to measure how much of an improvement our sequential sampling algorithm achieves over a static sampling algorithm that determines the sample size with worst-case bounds.

We implemented a simple subgroup discovery algorithm. Hypotheses consist of conjunctions of up to k attribute value tests. For discrete attributes, we allow tests for any of the possible values (*e.g.,* "color=green"); we discretize all continuous attributes and allow for testing whether the value of such attributes lies in an interval (*e.g.,* "size$\in [2.3, 5.8]$").

We also implemented a non-sequential sampling algorithm in order to quantify the relative benefit of sequential sampling. The non-sequential algorithm determines a sample size M like our algorithm does in step 2, but using the full available error probability δ rather than only $\frac{\delta}{2}$. Hence, the non-sequential sampling algorithm has a lower worst-case sample size than the sequential one but never exits or returns any hypothesis before that worst-case sample bound has been reached. Sequential and non-sequential sampling algorithm use the same normal approximation and come with identical guarantees on the quality of the returned solution.

For the first set of experiments, we used a database of 14,000 fruit juice purchase transactions. Each transaction is described by 29 attributes which specify properties of the purchased juice as well as attributes of the customer (*e.g.,* age and job). The task is to identify subgroups of customers that differ from the overall average with respect to their preference for cans, recyclable bottles, or non-recyclable bottles. For this problem, we studied hypothesis spaces of size 288 ($k = 1$, hypotheses test one attribute for a particular value), 37,717 ($k = 2$, conjunctions of two tests), and 3,013,794 ($k = 3$, conjunctions of three tests). Since δ has only a minor (logarithmic) influence on the resulting sample size, all results presented in Figure 1 were obtained with $\delta = 0.1$. We varied the utility function; the target attribute has three possible values, so we used the utility functions $f_1 = g\frac{1}{3}\sum_{i=1}^{3}|p_i - p_{0_i}|$, $f_2 = g^2\frac{1}{3}\sum_{i=1}^{3}|p_i - p_{0_i}|$, and $f_3 = \sqrt{g}\frac{1}{3}\sum_{i=1}^{3}|p_i - p_{0_i}|$.

Figure 1 shows the sample size of the non-sequential algorithm as well as the sample size required before the sequential algorithm returned the first (out of ten) hypothesis and the sample size that the sequential algorithm required to return the last (tenth) hypothesis and terminate. In every single experiment that we run, the sequential sampling algorithm terminated significantly earlier than the non-sequential one, even though the latter possesses a lower worst-case sample bound. As ε becomes small, the relative benefit of sequential

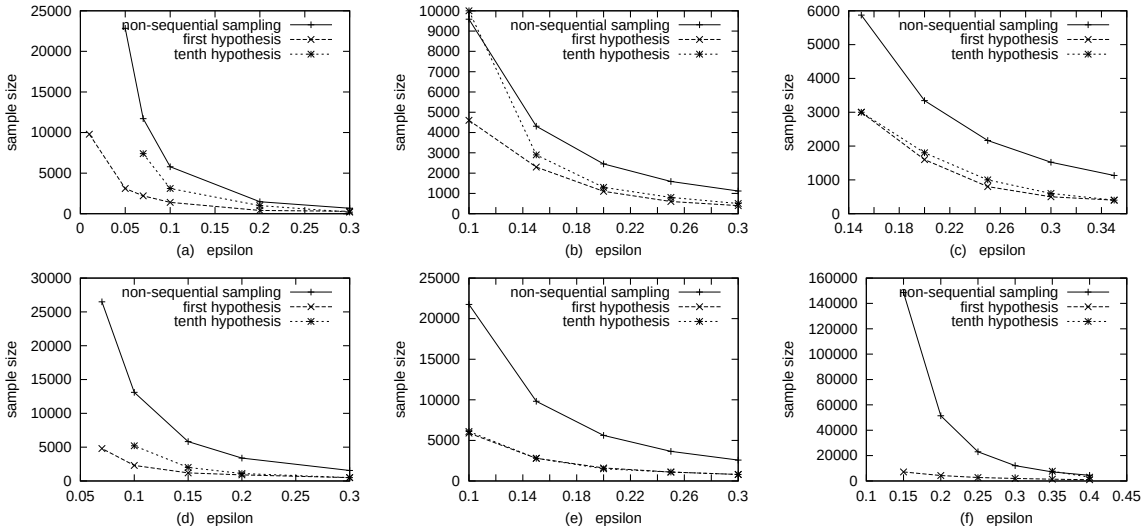

Figure 1: Sample sizes for the juice purchases database. (a) $f = g|p - p_0|$, $k = 1$, $\delta = .1$; (b) $k = 2$; (c) $k = 3$; (d) $f = g^2|p - p_0|$, $k = 1$, $\delta = .1$; (e) $k = 2$; (f) $f = \sqrt{g}|p - p_0|$, $k = 1$, $\delta = .1$

sampling can reach orders of magnitude. Consider, for instance, the linear utility function and $k = 1, \epsilon = .1, \delta = .1$. We can return the first hypothesis after 9,800 examples whereas the non-sequential algorithm returns the solution only after 565,290 examples. The sample size of the sequential algorithm is still reasonable for $k = 3$ and we expect it not to grow too fast for larger values as the worst-case bound is logarithmic in $|H|$ – *i.e.,* linear in k.

For the second set of experiments, we used the data provided for the KDD cup 1998. The data contains 95,412 records that describe mailings by a veterans organization. Each record contains 481 attributes describing one recipient of a previous mailing. The target fields note whether the person responded and how high his donation to the organization was. Our task was to find large subgroups of recipients that were particularly likely (or unlikely) to respond (we used the attribute "Target_B" as target and deleted "Target_D"). We discretized all numeric attributes (using five discrete values); our hypothesis space consists of all 4492 attribute value tests.

Figure 2 displays the sample sizes that our sequential sampling algorithm, as well as the non-sequential sampling algorithm that comes with exactly the same guarantee regarding the quality of the solutions required. Note that we use a logarithmic ($\log_{10}$) scale on the y axis. Although it is fair to say that this is a large-scale problem, the sample sizes used by the sequential sampling algorithm are in a reasonable range for all three studied utility functions. Less than 10,000 examples are required when ε is as small as 0.002 for $f = g|p - p_0|$ and $f = g^2|p - p_0|$ and when ϵ is 0.05 for $f = \sqrt{g}|p - p_0|$.

The relative benefit of sequential over non-sequential sampling is quite significant. For instance, in Figure 2a ($\varepsilon = 0.002$) the non-sequential algorithm requires over 10^7 examples

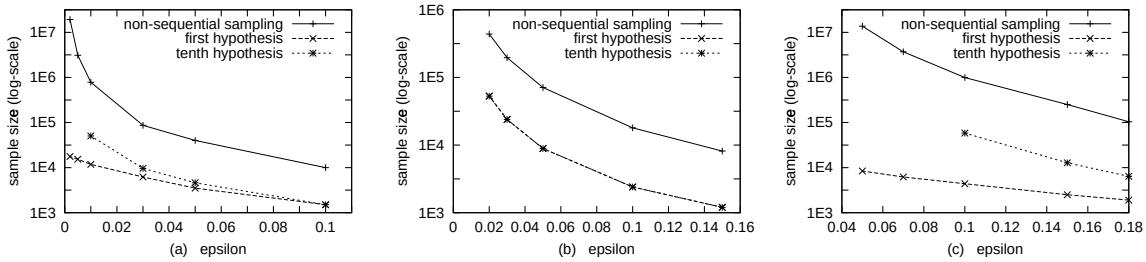

Figure 2: Required sample sizes (log-scale) for the KDD cup data of 1998. (a) $f = g|p-p_0|$, $k = 1$, $\delta = .1$; (b) $f = g^2|p - p_0|$, (c) $f = \sqrt{g}|p - p_0|$.

(of course, much more than are available) whereas the sequential one needs still less than 10^4.

6 Discussion and Related Results

Learning algorithms that require a number of examples which can be guaranteed to suffice for finding a nearly optimal hypothesis even in the worst case have early on been criticized as being impractical. Sequential learning techniques have been known in statistics for some time [4, 28, 9]. Maron, Moore, & Lee [20, 21] have introduced sequential sampling techniques into the machine learning context by proposing the "Hoeffding Race" algorithm that combines loop-reversal with adaptive Hoeffding bounds. A general scheme for sequential local search with instance-averaging utility functions has been proposed by Greiner [11].

Sampling techniques are particularly needed in the context of knowledge discovery in databases where often much more data are available than can be processed. A non-sequential sampling algorithm for KDD has been presented by Toivonen [25]; a sequential algorithm by Domingo *et al.* [5, 6]. A preliminary version of the algorithm presented in this paper has been discussed in [24]. This preliminary algorithm, however, did not use utility confidence bounds and its empirical behavior was less favorable than the behavior of the algorithm presented here. Our algorithm was inspired by the local searching algorithm of Greiner [11] but differs from it in a number of ways. The most important difference is that we refer to utility confidence bounds which makes it possible to handle all utility functions that can be estimated with bounded error even though they may not be an average across all instances.

In classification learning, error probabilities are clearly the dominating utility criterion. This is probably the reason why all sampling algorithms that have been studied so far are restricted to instance averaging utility functions. In many areas of machine learning and knowledge discovery (such as association rule and subgroup discovery), instance averaging utility functions are clearly inappropriate. The sampling algorithm of Domingo *et al.* [6] allows for utility criteria which are a function (with bounded derivative) of an average over

the instances. This, too, does not cover popular utility functions (such as $g|p - p_0|$) which depend on two averages (g and $|p - p_0|$) across the instances. Our algorithm is more general and works for all utility criteria for which a confidence interval can be found. We presented a list of instantiations for the most popular utility function for knowledge discovery tasks and showed that there is no solution for one function. Another minor difference between our algorithm and the one of [6] is that (when the utility confidence bound vanishes) our algorithm can be guaranteed to terminate *with certainty* (not just high probability) when it has reached a worst-case sample size bound.

So far, learning and discovery algorithms return the best hypothesis or all hypotheses over a certain utility threshold. Often, in particular in the context of knowledge discovery tasks, a user is interested in being provided with a number of the best hypotheses. Our algorithm returns the n approximately best hypotheses.

The approach that we pursue differs from the (PAC-style) worst-case approach by requiring smaller samples in all cases that are distinct from the worst case (in which all hypotheses are equally good). Instead of operating with smaller samples, it is also possible to work with a fixed-size sample but guarantee a higher quality of the solution if the observed situation differs from this worst case. This is the general idea of shell decomposition bounds [13, 19] and self-bounding learning algorithms [8].

Although we have discussed our algorithm only in the context of knowledge discovery tasks, it should be noted that the problem which we address is relevant in a much wider context. A learning agent that actively collects data and searches for a hypothesis (perhaps a control policy) which maximizes its utility function has to decide at which point no further improvement can be achieved by collecting more data. The utility function of an intelligent agent will generally be more complicated than an average over the observations. Our sequential sampling algorithm provides a framework for solving such problems.

As it is stated currently, our algorithm represents all considered hypotheses explicitly. It can therefore only be applied practically when the hypothesis space is relatively small. This is the case for most knowledge discovery tasks. The space of all association rules or subgroups over a certain number of attributes (which grows singly exponential in the number of monomials allowed) is much smaller than, for instance, the space of all decision trees (growing doubly exponential). However, most hypothesis spaces possess a symmetric structure which renders it unnecessary to represent all hypotheses explicitly. Although there are 2^n decision trees with n fixed leaf nodes it is trivial to assign optimal class labels in $O(n)$ steps without representing all 2^n alternatives. Similarly, the histogram of error rates of a set of decision trees or rule sets can be determined in time logarithmic in the number of hypotheses [23]. We are confident that our sampling algorithm can be applied analogously for complex and structured hypothesis spaces without explicit representation of all hypotheses.

By giving worst-case bounds on the sample size (and proving that there is no sampling algorithm for some utility functions) our results also give an indication as to which of the many utility functions appear preferable from a sampling point of view.

Table 2: Symbols used in the proof of theorem 1

n_i	value of n before iteration i
H	hypothesis space to be searched
H_i	hypotheses under consideration in iteration i of 3
H_i^*	the n_i best-looking hypotheses in iteration i of 3
$\overline{H_i^*}$	the remainder $(H_i \backslash H^*)$
G	solutions returned by the algorithm
G_i	solutions before iteration i
R_i	hypotheses removed iteration i
h	in arithmetic expressions: short for $f(h, D)$
$\hat{h}^{(i)}$	in arithmetic expressions: short for $\hat{f}(h, Q_i)$
i_{max}	value of i when leaving loop 3
i_{fin}	index used to denote sets after step 4

A Appendix: Proof of Theorem 1

Proof of Theorem 1.

In order to simplify writing the proof, we first introduce a few symbolic abbreviations and remind you of the meaning of the few others that we have already introduced (Table 2). Note that due to our notation, we have $G = G_{i_{fin}}$.

Further note that since $H_0 = H$, and due to the fact that hypotheses only ever get moved from H_i to G_{i+1} ("output") or to R_{i+1} ("delete"), we have

$$\forall i \in \{1, \ldots, i_{fin}\} : H = G_i \cup H_i \cup R_i \tag{80}$$

The proof is carried out by inductively showing that the following three loop invariants hold for all $i \in \{1, \ldots, i_{fin}\}$:

(I1) $\forall g \in G_i \forall r \in R_i : g \geq r - \varepsilon$

(I2) $\forall r \in R_i \exists^* h'_1, \ldots, h'_{n_i} \in H_i : h'_j \geq r \forall j \in \{1, \ldots, n_i\}$

(I3) $\forall g \in G_i \neg \exists^* h'_1, \ldots, h'_{n_i+1} \in H_i : h'_j > g + \varepsilon \forall j \in \{1, \ldots, n_i + 1\}$
which is equivalent to
$\forall g \in G_i \forall h'_1, \ldots, h'_{n_i+1} \in H_i \exists j \in \{1, \ldots, n_i + 1\} : g \geq h'_j - \varepsilon.$

In the invariant conditions, $\exists^*$ was used to denote "there exist distinct".

We will for now assume that in the course of executing loop 3, we never seriously misestimate any hypothesis (and will later bound the probability that this assumption is actually false):

(A1) $\forall h \in H \forall i \in \{1, \ldots, i_{max}\} :\mid \hat{h}^{(i)} - h \mid \leq E_h(i, \frac{\delta}{2M|H_i|})$

We will also assume that in selecting the final outputs in step 4, we will not err so as to violate our guarantee (and later quantify how likely this is to be the case):

(A2) If $E(i_{max}, \frac{\delta}{2|H_{i_{max}}|}) \leq \frac{\varepsilon}{2}$ then $\forall h \in H^*_{i_{max}} \forall h' \in \overline{H^*_{i_{max}}} : h \geq h' - \varepsilon$.

Note that for the purpose of the proof, we will consider only the case where during each loop iteration, only one of 3(e)i or 3(e)ii is executed for a single hypothesis. Since in the algorithm, H^* is redetermined whenever one of those steps is carried out, the version considered here is equivalent (but might consume more samples). We are now ready for our inductive proof.

Base case: $i = 1$

Since $G_1 = \emptyset$ and $R_1 = \emptyset$, I1, I2 and I3 are trivially true.

Inductive Step: $i \rightarrow i + 1$

We discuss this case by case depending on what happens during step i.

Case (1) A hypothesis h is output in step 3(e)i, i.e.,

$$G_{i+1} := G_i \cup \{h\}, H_{i+1} := H_i \backslash \{h\}, n_{i+1} := n_i - 1, R_{i+1} := R_i \tag{81}$$

From the algorithm, we know:

$$\hat{h}^{(i)} \geq E_h\left(i, \frac{\delta}{2M|H_i|}\right) + max_{h' \in \overline{H^*_i}} \left\{\hat{h}'^{(i)} + E_{h'}(i, \frac{\delta}{2M|H_i|})\right\} - \varepsilon \tag{82}$$

In the following, since we are always dealing with step i and sample Q_i, we will drop the superscript (i), and abbreviate n_i as n. We will further abbreviate $(i, \frac{\delta}{2M|H_i|})$ as $(\cdot)$, so Equation 82 becomes Equation 83.

$$\hat{h} \geq E_h(\cdot) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - \varepsilon \tag{83}$$

Now let $h'' \in \overline{H^*}$. We can rearrange Equation 83 by adding paired terms in Equation 84.

$$\begin{aligned} \hat{h} + h - h &\geq h'' - h'' + E_h(\cdot) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - \varepsilon & (84) \\ \Leftrightarrow h &\geq h'' + E_h(\cdot) - (\hat{h} - h) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - h'' - \varepsilon & (85) \end{aligned}$$

Now, since certainly $max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] \geq \hat{h}'' + E_{h''}(\cdot)$ for all $h'' \in \overline{H^*}$, we can derive Equation 86

$$\begin{aligned} h &\geq h'' + E_h(\cdot) - (\hat{h} - h) + \hat{h}'' + E_{h''}(\cdot) - h'' - \varepsilon & (86) \\ \Leftrightarrow h &\geq h'' + E_h(\cdot) - (\hat{h} - h) + E_{h''}(\cdot) - (h'' - \hat{h}'') - \varepsilon & (87) \end{aligned}$$

According to (A1), $\hat{h} - h \leq E_h(\cdot)$, and $h'' - \hat{h}'' \leq E_{h''}(\cdot)$, so Equation 88 must hold.

$$h \geq h'' - \varepsilon \text{ for all } h'' \in \overline{H^*} \tag{88}$$

We can now show the three invariant conditions.

Show (I1) By inductive assumption (I1), $\forall g \in G_i \forall r \in R_i : g \geq r - \varepsilon$. Now, since $G_{i+1} := G_i \cup \{h\}$, we have to show that Equation 89 holds.

$$\forall r \in R_{i+1} : h \geq r - \varepsilon \tag{89}$$

So consider any particular $r \in R_{i+1}$. Since $R_{i+1} = R_i$, $r \in R_i$, from inductive assumption (I2), we know that Equation 90 holds.

$$\exists^* h'_1, \ldots, h'_{n_i} \in H_i : h'_j \geq r \forall j \in \{1, \ldots, n_i\} \tag{90}$$

Note that h is from H^*, thus two cases can arise[1]. If all the h'_j are in H^*, since $|H^*| = n_i$, one of them, say h'_{j_1}, is equal to h, so we can show (89) immediately from (90), as in Equation 94.

$$h = h'_{j_1} \geq r \geq r - \varepsilon \tag{94}$$

Otherwise, one of the h'_j, say h'_{j_2}, must be from $\overline{H^*}$; in Equation 95, we use (88) and (90) to show Equation 89.

$$h \geq h'_{j_2} - \varepsilon \geq r - \varepsilon \tag{95}$$

N.B., it would not have been sufficient for (I2) (and step 3(e)ii) to guarantee $h'_j \geq r - \varepsilon$, since then in (95) we would arrive at $h \geq r - 2\varepsilon$, which would not maintain (I1).

Show (I2) Since (I2) holds by inductive assumption in step i, Equation 96 follows.

$$\forall r \in R_i \exists^* h'_1, \ldots, h'_{n_i} \in H_i : h'_j \geq r \forall j \in \{1, \ldots, n_i\} \tag{96}$$

Furthermore, as $R_{i+1} = R_i$ and we have removed only one hypothesis ($H_{i+1} := H_i \backslash \{h\}$), we know each r can have "lost" at most one of its h'_j, so Equation 97 must hold. Since $n_{i+1} = n_i - 1$, this implies Equation 98.

$$\forall r \in R_{i+1} \exists^* h'_1, \ldots, h'_{n_i - 1} \in H_{i+1} : h'_j \geq r \forall j \in \{1, \ldots, n_i - 1\} \tag{97}$$

$$\Rightarrow \quad \forall r \in R_{i+1} \exists^* h'_1, \ldots, h'_{n_{i+1}} \in H_{i+1} : h'_j \geq r \forall j \in \{1, \ldots, n_{i+1}\} \tag{98}$$

[1]Note (83) alone is not enough to guarantee that h is in H^*, since even if $h \in \overline{H^*}$ were true, we would have

$$\hat{h} > E_h(\cdot) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - \varepsilon \tag{91}$$

$$\Rightarrow \hat{h} > E_h(\cdot) + \hat{h} + E_h(\cdot) - \varepsilon \tag{92}$$

$$\Leftrightarrow E_h(\cdot) < \frac{\varepsilon}{2} \tag{93}$$

which cannot be excluded.

Show (I3) By inductive assumption, (I3) holds for all $g \in G_i$, and since $G_{i+1} := G_i \cup \{h\}$, we only have to show that Equation 99 holds.

$$\neg\exists^* h'_1, \ldots, h'_{n_{i+1}+1} \in H_{i+1} : h'_j > h + \varepsilon \forall j \in \{1, \ldots, n_{i+1}+1\} \tag{99}$$

From (88) we know that $h \geq h'' - \varepsilon \Leftrightarrow h'' \leq h + \varepsilon \forall h'' \in \overline{H^*}$, so any h'_j to violate Equation 99 could only come from H^* which is of size $n_i = n_{i+1} + 1$. However, since also $h \in H^*$, there are only n_{i+1} candidates left, proving Equation 99.

Case (2) A hypothesis h is removed in step 3(e)ii, *i.e.,* the statements of Equation 100 are executed.

$$G_{i+1} := G_i, H_{i+1} := H_i \backslash \{h\}, R_{i+1} := R_i \cup \{h\}, n_{i+1} := n_i \tag{100}$$

From the algorithm, we know that Equation 101 must hold.

$$\hat{h}^{(i)} \leq min_{h' \in H_i^*} \left\{ \hat{h}'^{(i)} - E_{h'} \left(i, \frac{\delta}{2M|H_i|} \right) \right\} - E_h \left(i, \frac{\delta}{2M|H_i|} \right) \tag{101}$$

Analogously to the previous case, we will abbreviate Equation 101 as Equation 102.

$$\hat{h} \leq min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] - E_h(\cdot) \tag{102}$$

We now proceed in a similar fashion as in Equations 84 to 88. So let $h'' \in H^*$, and rearrange Equation 102 by adding paired terms in Equation 103.

$$\begin{aligned} \hat{h} + h - h &\leq h'' - h'' + min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] - E_h(\cdot) && (103) \\ \Leftrightarrow h &\leq h'' + (h - \hat{h}) - E_h(\cdot) + min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] - h'' && (104) \end{aligned}$$

Now, since certainly $min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] \leq \hat{h}'' - E_{h''}(\cdot)$ for all $h'' \in H^*$, Equations 105 and 106 are equivalent.

$$\begin{aligned} h &\leq h'' + (h - \hat{h}) - E_h(\cdot) + \hat{h}'' - E_{h''} - h'' && (105) \\ \Leftrightarrow h &\leq h'' + (h - \hat{h}) - E_h(\cdot) + (\hat{h}'' - h'') - E_{h''} && (106) \end{aligned}$$

If (A1) is assumed true, then $h - \hat{h} \leq E_h(\cdot)$, and $\hat{h}'' - h'' \leq E_{h''}$, so Equation 107 must be true.

$$h \leq h'' \text{ for all } h'' \in H^* \tag{107}$$

We can now show the three invariant conditions.

Show (I1) Since $G_{i+1} := G_i$, by inductive assumption (I1) we have Equation 108.

$$\forall g \in G_{i+1} \forall r \in R_i : g \geq r - \varepsilon \tag{108}$$

Since $R_{i+1} = R_i \cup \{h\}$, we only need to show that Equation 109 is true.

$$\forall g \in G_{i+1}\ g \geq h - \varepsilon \tag{109}$$

Note that $h \in \overline{H^*}$, since due to Equation 102, $h \in H^*$ would imply that $E_h(\cdot) \leq 0$, as demonstrated in Equations 110 to 113. However, $E_h(\cdot) > 0$ for all h and all arguments of $E_h(\cdot)$ which means that $h \in \overline{H^*}$ and thus $|H^* \cup \{h\}| = n_i + 1$.

$$\begin{aligned} \hat{h} &\leq min_{h' \in H^*}[\hat{h}' - E_{h'}(\cdot)] - E_h(\cdot) && (110)\\ \Rightarrow \hat{h} &\leq \hat{h} - E_h(\cdot) - E_h(\cdot) && (111)\\ \Leftrightarrow 0 &\leq -2E_h(\cdot) && (112)\\ \Leftrightarrow E_h(\cdot) &\leq 0 && (113) \end{aligned}$$

By inductive assumption (I3) we have Equation 114 for all $g \in G_{i+1}$.

$$\forall g \in G_{i+1} = G_i \forall h'_1, \ldots, h'_{n_i+1} \in H_i \exists j \in \{1, \ldots, n_i + 1\} : g \geq h'_j - \varepsilon \tag{114}$$

From Equation 114, we can conclude that for one hypothesis out of $H^* \cup \{h\}$, call it h', it must be true that $g \geq h' - \varepsilon$. If $h' = h$, this shows Equation 109 immediately; if not, then $h' \in H^*$, so from Equation 107 it follows that $g \geq h' - \varepsilon \geq h - \varepsilon$.

Show (I2) Since $R_{i+1} = R_i \cup \{h\}$, let us first show Equation 115.

$$\exists^* h'_1, \ldots, h'_{n_{i+1}} \in H_{i+1} : h'_j \geq h \forall j \in \{1, \ldots, n_i + 1\} \tag{115}$$

Now since $H_{i+1} = H_i \backslash \{h\}$ and $h \notin H^*$, we know that $H^* \subseteq H_{i+1}$. Since $|H^*| = n_i = n_{i+1}$, and due to (107), we can simply choose $h'_1, \ldots, h'_{n_{i+1}}$ to be H^*, thus showing (115). Then consider $r \in R_i$. By inductive assumption, since $n_i = n_{i+1}$, we have Equation 116.

$$\exists^* h'_1, \ldots, h'_{n_{i+1}} \in H_i : h'_j \geq r \forall j \in \{1, \ldots, n_{i+1}\} \tag{116}$$

If none of the h'_j is equal to h, we know they are all in H_{i+1} also. If one of them, say h_{j_1} is equal to h, we know from Equation 116 that $h_{j_1} \geq r \Leftrightarrow h \geq r$, so we can again use H^* in place of the h'_j, and (I2) is true for all $r \in R_i$ also, and thus for R_{i+1}, as required.

Show (I3) By inductive assumption (I3), and since $G_{i+1} = G_i, n_{i+1} = n_i$, Equation 117 must be satisfied.

$$\forall g \in G_{i+1} \neg \exists^* h'_1, \ldots, h'_{n_{i+1}+1} \in H_i : h'_j > g + \varepsilon \forall j \in \{1, \ldots, n_{i+1} + 1\} \tag{117}$$

Now, since $H_{i+1} = H_i \backslash \{h\}$, surely such offending sets of h'_j do not exist in H_{i+1} either. This shows that the statement of Equation 118 is true.

$$\text{(I1), (I2), and (I3) are true whenever we exit the loop.} \tag{118}$$

Let i_{max} be the index reached at this point. We now show that (I1), (I2), and (I3) continue to hold when the algorithm exits, *i.e.,* after step 4.

Case 3a We exit the loop because $n_{i_{max}} = 0$ is true. Then step 4 leaves everything unchanged, so (I1), (I2), and (I3) continue to hold.

Case 3b We exit the loop because $|H_{i_{max}}| = n_{i_{max}}$ is true. Then step 4 moves all of $H_{i_{max}}$ to $G_{i_{fin}}$, as described in Equation 119.

$$G_{i_{fin}} := G_{imax} \cup H_{i_{max}}, H_{i_{fin}} := \emptyset, R_{i_{fin}} := R_{i_{max}}, n_{i_{fin}} = 0 \tag{119}$$

Show I1 The argument is essentially identical to case 1/I1. Since (I1) is true when exiting the loop, and $R_{i_{fin}} = R_{i_{max}}$, we only have to show Equation 120.

$$\forall h \in H_{i_{max}} \forall r \in R_{i_{max}} : h \geq r - \varepsilon \tag{120}$$

So consider any particular $h \in H_{i_{max}}$, $r \in R_{i_{max}}$. From (I2), we know that Equation 121 holds.

$$\exists^* h'_1, \ldots, h'_{n_{i_{max}}} \in H_{i_{max}} : h'_j \geq r \forall j \in \{1, \ldots, n_{i_{max}}\} \tag{121}$$

But since $|H_{i_{max}}| = n_{i_{max}}$, h must be among the h_j, which shows Equation 120.

Show I2 Since $n_{i_{fin}} = 0$, (I2) is trivially true.

Show I3 Since $H_{i_{fin}} = \emptyset$, (I3) is trivially true.

Case 3c We exit the loop when $E(i, \frac{\delta}{2|H_{imax}|}) \leq \frac{\varepsilon}{2}$. Then after step 4, the conditions displayed in Equation 122 hold.

$$G_{i_{fin}} := G_{i_{max}} \cup H^*_{i_{max}}, H_{i_{fin}} := \overline{H^*_{i_{max}}}, R_{i_{fin}} := R_{i_{max}}, n_{i_{fin}} = 0 \tag{122}$$

So if we assume (A2) to be true, we can conclude that Equation 123 is true.

$$\forall h \in H^*_{i_{max}} \forall h' \in \overline{H^*_{i_{max}}} : h \geq h' - \varepsilon \tag{123}$$

Show I1 (I1) is true for all $g \in G_{i_{max}}$ by Equation 118, and true for $H^*_{i_{max}}$ according to Equation 123, so it is true for $G_{i_{fin}}$.

Show I2 Trivially true since $n_{i_{fin}} = 0$.

Show I3 Since $n_{i_{fin}} = 0$, we need to show the condition of Equation 124.

$$\forall g \in G_{i_{fin}} = G_{i_{max}} \cup H^*_{i_{max}}; \forall h \in H_{i_{fin}} = \overline{H^*_{i_{max}}} : g \geq h - \varepsilon \tag{124}$$

Equation 124 again is true due to Equation 123. Thus (I1), (I2), and (I3) are true when the algorithm exits.

We are now ready to show that indeed the guarantee of our theorem holds. From (I1), we know that Equation 125 holds.

$$\forall g \in G_{i_{fin}} \neg\exists r \in R_{i_{fin}} : r > g + \varepsilon \tag{125}$$

From (I3), we know that $\forall g \in G_{i_{fin}} \neg\exists h' \in H_{i_{fin}} : h' > g + \varepsilon$. Since according to Equation 80, $R_{i_{fin}} \cup H_{i_{fin}} = H \backslash G_{i_{fin}}$, this shows the guarantee of our theorem.

Now we are left with quantifying the probability that (A1) or (A2) are false, which together must be at most δ.

The risk of (A1) being false is quantified by the following lemma.

Lemma 1 *With confidence at least $1 - \frac{\delta}{2}$, there is no time step i ($1 \leq i \leq M$) and no hypothesis $h \in H$ such that $|\hat{f}(h, Q_i) - f(h)| E_h(i, \frac{\delta}{2M|H_i|})$.*

Proof. First note that the loop (step 3.) will be executed at most M times: Since for all i, $|H_i| \leq |H|$, thus $\frac{\delta}{2M|H_i|} \geq \frac{\delta}{2M|H|}$. From the definition of E, Equation 126 follows.

$$\forall 0 < \delta_1 \leq \delta_2 < 1, \forall m \in \mathbb{N} : E(m, \delta_1) \geq E(m, \delta_2) \tag{126}$$

Thus, Equation 127 shows that the algorithms stops when $i = M$ at the latest.

$$E_h\left(M, \frac{\delta}{2M|H|}\right) \leq \frac{\varepsilon}{2} \Rightarrow E_h\left(M, \frac{\delta}{2M|H_i|}\right) \leq \frac{\varepsilon}{2} \tag{127}$$

Using this, in Equations 128 and 129, we refer to the union bound. In Equation 130, we apply the definition of E. Equation 131 completes the proof.

$$\begin{aligned}
& Pr\left[\exists i, \exists h \in H_i : |\hat{f}(h, Q_i) - f(h)| > E_h\left(i, \frac{\delta}{2M|H_i|}\right)\right] \\
\leq & \sum_{i=1}^{M} Pr\left[\exists h \in H_i : |\hat{f}(h, Q_i) - f(h)| > E_h\left(i, \frac{\delta}{2M|H_i|}\right)\right] && (128) \\
\leq & \sum_{i=1}^{M} |H_i| Pr\left[|\hat{f}(h, Q_i) - f(h)| > E_h\left(i, \frac{\delta}{2M|H_i|}\right)\right] && (129) \\
< & \sum_{i=1}^{M} |H_i| \frac{\delta}{2M|H_i|} && (130) \\
= & \frac{\delta}{2} && (131)
\end{aligned}$$

Now consider (A2).

Lemma 2 *For any integer $i_{max} > 0$, if $E(i_{max}, \frac{\delta}{2|H_{i_{max}}|}) \leq \frac{\varepsilon}{2}$ then with probability $1 - \frac{\delta}{2}$, Equation 132 holds.*

$$\forall h \in H^*_{i_{max}} \forall h' \in \overline{H^*_{i_{max}}} : h \geq h' - \varepsilon \tag{132}$$

Proof. If $h \in H^*_{i_{max}}$ and $h' \in \overline{H^*_{i_{max}}}$, then $\hat{h} \geq \hat{h}'$. If nonetheless Equation 132 is true, we must have estimated at least one the two hypotheses wrong by more than $\frac{\varepsilon}{2}$. We can bound the probability of this as follows. Equation 133 follows from the definition of M in step 2 of the algorithm. We refer to the union bound in Equation 134 and to the definition of E (Equation 1) in Equation 135.

$$
\begin{aligned}
& Pr\left[\exists h \in H_{i_{max}} : \hat{h} - h > \frac{\varepsilon}{2}\right] \\
& \leq \quad Pr\left[\exists h \in H_{i_{max}} : \hat{h} - h > E\left(i_{max}, \frac{\delta}{2|H_{i_{max}}|}\right)\right] && (133) \\
& \leq \quad |H_{i_{max}}| Pr\left[\hat{h} - h > E\left(i_{max}, \frac{\delta}{2|H_m|}\right)\right] && (134) \\
& \leq \quad \frac{\delta}{2} && (135)
\end{aligned}
$$

This completes the proof of Lemma 2 and thereby the proof of Theorem 1. ∎

ACKNOWLEDGMENT

We wish to thank Frank Schulz for carefully proof-reading the paper and giving us helpful comments.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Conference on Management of Data*, pages 207–216, 1993.

[2] L. Breiman, J Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Pacific Grove, 1984.

[3] H. Chernoff. A measure of asymptotoc efficiency for tests of a hypothesis based on the sums of observations. *Annals of Mathematical Statistics*, 23:409–507, 1952.

[4] H. Dodge and H. Romig. A method of sampling inspection. *The Bell System Technical Journal*, 8:613–631, 1929.

[5] C. Domingo, R. Gavelda, and O. Watanabe. Practical algorithms for on-line selection. In *Proc. International Conference on Discovery Science*, pages 150–161, 1998.

[6] C. Domingo, R. Gavelda, and O. Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. Technical Report TR-C131, Dept. de LSI, Politecnica de Catalunya, 1999.

[7] U. Fayyad, G. Piatetski-Shapiro, and P. Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *KDD-96*, 1996.

[8] Y. Freund. Self-bounding learning algorithms. In *Proceedings of the International Workshop on Computational Learning Theory (COLT-98)*, 1998.

[9] K. Ghosh, , M. Mukhopadhyay, and P. Sen. *Sequential Estimation.* Wiley, 1997.

[10] R. Greiner and R. Isukapalli. Learning to select useful landmarks. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B:473–449, 1996.

[11] Russell Greiner. PALO: A probabilistic hill-climbing algorithm. *Artificial Intelligence*, 83(1–2), July 1996.

[12] P. Haas and A. Swami. Sequential sampling procedures for quesy size estimation. Research Report RJ 9101 (80915), IBM, 1992.

[13] D. Haussler, M. Kearns, S. Seung, and N. Tishby. Rigorous learning curve bounds from statistical mechanics. *Machine Learning*, 25, 1996.

[14] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[15] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[16] W. Klösgen. Problems in knowledge discovery in databases and their treatment in the statistics interpreter explora. *Journal of Intelligent Systems*, 7:649–673, 1992.

[17] W. Klösgen. Assistant for knowledge discovery in data. In P. Hoschka, editor, *Assisting Computer: A New Generation of Support Systems*, 1995.

[18] W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In Fayyad et al., editor, *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI, 1996.

[19] J. Langford and D. McAllester. Computable shell decomposition bounds. In *Proceedings of the International Conference on Computational Learning Theory*, 2000.

[20] O. Maron and A. Moore. Hoeffding races: Accelerating model selection search for classification and function approximating. In *Advances in Neural Information Processing Systems*, pages 59–66, 1994.

[21] A. Moore and M. Lee. Efficient algorithms for minimizing cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 190–198, 1994.

[22] G. Piatetski-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248, 1991.

[23] T. Scheffer and T. Joachims. Expected error analysis for model selection. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999.

[24] T. Scheffer and S. Wrobel. A sequential sampling algorithm for a general class of utility functions. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2000.

[25] H. Toivonen. Sampling large databases for association rules. In *Proc. VLDB Conference*, 1996.

[26] R. Uthurusamy, U. Fayyad, and S. Spangler. Learning useful rules from inconclusive data. In *Knowledge Discovery in Databases*, pages 141–158, 1991.

[27] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1996.

[28] A. Wald. *Sequential Analysis.* Wiley, 1947.

[29] D. H. Wolpert. The relationship between PAC, the statistical physics framework, the Bayesian framework, and the VC framework. In D. H. Wolpert, editor, *The Mathematics of Generalization*, The SFI Studies in the Sciences of Complexity, pages 117–214. Addison-Wesley, 1995.

[30] Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proc. First European Symposion on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pages 78–87, Berlin, 1997.