Journal of Machine Learning Research 3 (2002) 833-862 Submitted 12/01; Published 12/02

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

Tobias Scheffer SCHEFFER@IWS.CS.UNI-MAGDEBURG.DE
University of Magdeburg, FIN/IWS
P.O. Box 4120, 39016 Magdeburg, Germany

Stefan Wrobel WROBEL@AIS.FHG.DE
FhG AiS, Schloß Birlinghoven 53754 Sankt Augustin, Germany
University of Bonn, Informatik III, Römerstr. 164, 53117 Bonn, Germany

Editors: Carla E. Brodley and Andrea Danyluk

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 the confidence and quality of solutions. Known sampling approaches have treated single hypothesis selection problems, assuming that the utility is 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.

Keywords: Sampling, online learning, incremental learning, large databases.

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" (Fayyad et al., 1996). An important aspect of this task is the *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 some part of the data is 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 much 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 (Toivonen, 1996).

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 (Freund, 1998) and shell decomposition bounds (Haussler et al., 1996, Langford and McAllester, 2000). 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 (Dodge and Romig, 1929, Wald, 1947, Ghosh et al., 1997).

In the machine learning context, the idea of sequential sampling has been developed into the Hoeffding race algorithm (Maron and Moore, 1994) which processes examples incrementally, updates the empirical utility values simultaneously, and starts to output (or discard) hypotheses as soon as it becomes very likely that some hypothesis is near-optimal (or very poor, respectively). The incremental greedy learning algorithm PALO (Greiner, 1996) 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 (Haas and Swami, 1992, Domingo et al., 1999).

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 the 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.,* Wrobel (1997). 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 (Klösgen, 1995) 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. 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 Wrobel (1997), where that 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*

> *there is no $h' \in H$: $h' \notin G$ and $f(h', D) > f_{min}$, 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 are 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, with high probability, there will be no 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 the utility criterion. For the discovery of *association rules*, by contrast, one usually relies on *generality* as the primary utility criterion (Agrawal et al., 1993). Finally, for subgroup discovery, it is commonplace to combine both generality and *distributional unusualness*, resulting in relatively complex evaluation functions – see, *e.g.,* Klösgen (1996) 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 Domingo et al. (1999), 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 the 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 we 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 a 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 $E(m,\delta)$ 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 $E(m,\delta)$ 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 a 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. The GSS Sampling Algorithm

A 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 (Chernoff, 1952) that are used in PAC theory (Kearns and Vazirani, 1994, Wolpert, 1995, Vapnik, 1996) and many other areas of statistics and computer science can be used to determine appropriate sample bounds (Toivonen, 1996). 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 (Dodge and Romig, 1929, Wald, 1947) which have also been referred to as *adaptive* sampling methods (Domingo et al., 1998). The idea of sequential sampling is that when the 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 wait for when the frequencies are similar. Sequential sampling methods have been reported to reduce the required sample size by several orders of magnitude – *e.g.,* Greiner and Isukapalli (1996).

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 (most newer database systems are capable of drawing samples at random), *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 for which 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 for which 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 with $n < |H|$), 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** $H_{i+1} = H_i$, **Let** $n_{i+1} = n_i$.

   (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 **Decrement** 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 4 *When the algorithm terminates, it will output a group G of exactly n hypotheses such that, with confidence* $1-\delta$*, no other hypothesis in H has a utility that 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 4 can be found in Appendix A

Theorem 5 (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.*

The correctness of Theorem 5 follows immediately from Step 3e of the algorithm. Theorem 5 says that we can guarantee termination if the confidence interval vanishes for large numbers of examples. This is a benign 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 transactions they predict the value of an attribute given the values of some other attributes. For instance, the rule "beer= 1 → 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 that 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 used for training. 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 (Klösgen, 1996). 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 for which 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 (Hoeffding, 1963) to bound the chance that an arbitrary (bounded) random variable X takes a value that 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 is 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 $E(m,\delta)$ 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 standard deviation $s_{\hat{f}(h,Q_m)-f(h)}=\frac{1}{m}\sqrt{\sum_{i=1}^{m}(\hat{f}_{inst}(h,q_i)-\hat{f}(h,Q_m))^2}$ 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 (a normal distribution with mean value zero and standard deviation one) which implies that Equation 8 satisfies Equation 1. z is the inverse standard normal distribution that can be looked up in a table.

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

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

$$
\begin{aligned}
E(m,\delta) &= z_{1-\frac{\delta}{2}}\cdot s_h && (9)\\
&= z_{1-\frac{\delta}{2}}\frac{1}{m}\sqrt{\sum_{i=1}^{m}(f_{inst}(h,q_i)-\hat{f}(h,Q_i))^2} && (10)
\end{aligned}
$$

Note that we have simplified the situation a little. We have confused the true standard deviation σ (the average squared distance from the true mean $f(h)$) and the empirical standard deviation 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 $m \geq \frac{2\Lambda^2}{\varepsilon^2} \log \frac{4|H|}{\delta}$. In Equation 11, we expand our definition of $E(m,\delta)$. The Λ and log-terms cancel out in Equation 12; we can bound the confidence interval to $\frac{\varepsilon}{2}$ in Equation 12 as required for the algorithm to exit in step 3e.

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

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

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 (Piatetski-Shapiro, 1991). 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 6 *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*

$$\text{small } m: \quad E(m,\delta) = 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} \tag{13}$$

$$\text{large } m: \quad E(m,\delta) = \frac{z_{1-\frac{\delta}{4}}}{\sqrt{m}} + \frac{(z_{1-\frac{\delta}{4}})^2}{4m} \tag{14}$$

$$E_h(m,\delta) = z_{1-\frac{\delta}{4}}(s_g + s_p + z_{1-\frac{\delta}{4}} s_g s_p) \tag{15}$$

where z is the inverse normal distribution.

Proof. (6.1) In Equation 16, we insert Equation 13 into Equation 1 and, for better readability, introduce E as an abbreviation for $\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}$. We refer to the union bound in Equation 17. Then, we exploit that $\varepsilon^2 \leq \varepsilon$ and $2\varepsilon - \varepsilon^2 > 0$ for $\varepsilon \leq 1$ in Equation 18. The simple observation that $g \leq 1$ and $(p - p_0) \leq 1$ leads to Equation 19. Equations 20 and 21 are based on elementary transformations. In Equation 22, 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 23.

$$
\begin{aligned}
& Pr[|\hat{f}(h,Q_m) - f(h)| > E(m,\delta)] \\
= \;& Pr\left[|\hat{g}(\hat{p}-p_0) - g(p-p_0)| > 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} = 3E\right] && (16)\\
\leq \;& Pr[\hat{g}(\hat{p}-p_0) - g(p-p_0) > 3E] + Pr[\hat{g}(\hat{p}-p_0) - g(p-p_0) < -3E] && (17)\\
\leq \;& Pr\left[\hat{g}(\hat{p}-p_0) - g(p-p_0) > 2E + E^2\right] \\
& + Pr\left[\hat{g}(\hat{p}-p_0) - g(p-p_0) < -2E + E^2\right] && (18)\\
\leq \;& Pr\left[\hat{g}(\hat{p}-p_0) - g(p-p_0) > gE + (p-p_0)E + E^2\right] \\
& + Pr\left[\hat{g}(\hat{p}-p_0) - g(p-p_0) < -gE - (p-p_0)E + E^2\right] && (19)\\
\leq \;& Pr[\hat{g}(\hat{p}-p_0) - g(p-p_0) > (g+E)(p-p_0+E) - g(p-p_0)] \\
& + Pr[\hat{g}(\hat{p}-p_0) - g(p-p_0) < (g-E)(p-p_0-E) - g(p-p_0)] && (20)\\
\leq \;& Pr[\hat{g}(\hat{p}-p_0) > (g+E)(p-p_0+E)] + Pr[\hat{g}(\hat{p}-p_0) < (g-E)(p-p_0-E)] && (21)\\
\leq \;& 2Pr\left[\hat{q} > \left(q + \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\right] + 2Pr\left[\hat{q} < \left(q - \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right)\right] && (22)\\
= \;& 2Pr\left[|\hat{q} - q| > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] \leq 4\exp\left\{-2m\frac{1}{2m}\log\frac{4}{\delta}\right\} = \delta && (23)
\end{aligned}
$$

Let us now prove the normal approximation of the above confidence bound. We start in Equation 25 by inserting Equation 15 into Equation 1. s_g and s_p denote the standard deviations of g and p, respectively. We also cover Equation 14 in this proof. The standard deviations can be bounded from above: $s_g, s_p \leq \frac{1}{2\sqrt{m}}$. Hence, $z_{1-\frac{\delta}{4}}(s_g + s_p + z_{1-\frac{\delta}{4}} s_g s_p) \leq \frac{z_{1-\frac{\delta}{4}}}{\sqrt{m}} + \left(z_{1-\frac{\delta}{4}}\frac{1}{2\sqrt{m}}\right)^2 \leq \frac{z_{1-\frac{\delta}{4}}}{\sqrt{m}} + \frac{(z_{1-\frac{\delta}{4}})^2}{4m}$ We expand the definition of f in Equation 24 and apply the union bound in Equation 25. Equation 26 follows from $g \leq 1$ and $p - p_0 \leq 1$ and in Equation 27 we exploit that $P(X < x) \leq P(X < y+z)$. 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.

$$
\begin{aligned}
& Pr[|\hat{f}(h,Q_m) - f(h)| > z_{1-\frac{\delta}{4}}(s_g + s_p + z_{1-\frac{\delta}{4}} s_g s_p)] && (24)\\
\leq \;& Pr\left[\hat{g}(\hat{p}-p_0) - g(p-p_0) > z_{1-\frac{\delta}{4}}(s_g + s_p + z_{1-\frac{\delta}{4}} s_g s_p)\right] \\
& + Pr\left[\hat{g}(\hat{p}-p_0) - g(p-p_0) < -z_{1-\frac{\delta}{4}}(s_g + s_p + z_{1-\frac{\delta}{4}} s_g s_p)\right] && (25)\\
\leq \;& Pr\left[\hat{g}(\hat{p}-p_0) - g(p-p_0) > (p-p_0) z_{1-\frac{\delta}{4}} s_g + g z_{1-\frac{\delta}{4}} s_p + \left(z_{1-\frac{\delta}{4}}\right)^2 s_g s_p\right]
\end{aligned}
$$

$$+Pr\left[\hat{g}(\hat{p}-p_0)-g(p-p_0) < -(p-p_0)z_{1-\frac{\delta}{4}}s_g - gz_{1-\frac{\delta}{4}}s_p - \left(z_{1-\frac{\delta}{4}}\right)^2 s_g s_p\right] \quad (26)$$

$$\leq Pr\left[\hat{g}(\hat{p}-p_0)-g(p-p_0) > (p-p_0)z_{1-\frac{\delta}{4}}s_g + gz_{1-\frac{\delta}{4}}s_p + \left(z_{1-\frac{\delta}{4}}\right)^2 s_g s_p\right]$$

$$+Pr\left[\hat{g}(\hat{p}-p_0)-g(p-p_0) < -(p-p_0)z_{1-\frac{\delta}{4}}s_g - gz_{1-\frac{\delta}{4}}s_p + \left(z_{1-\frac{\delta}{4}}\right)^2 s_g s_p\right] \quad (27)$$

$$\leq Pr\left[\hat{g}(\hat{p}-p_0) > \left(g+z_{1-\frac{\delta}{4}}s_g\right)\left(p-p_0+z_{1-\frac{\delta}{4}}s_p\right)\right]$$

$$+Pr\left[\hat{g}(\hat{p}-p_0) < \left(g-z_{1-\frac{\delta}{4}}s_g\right)\left(p-p_0-z_{1-\frac{\delta}{4}}s_p\right)\right] \quad (28)$$

$$\leq Pr\left[|\hat{g}-g| > z_{1-\frac{\delta}{4}}s_g\right] + Pr\left[|\hat{p}-p| > p+z_{1-\frac{\delta}{4}}s_p\right] \leq 2\left(\frac{\delta}{4}+\frac{\delta}{4}\right) = \delta \quad (29)$$

This completes the proof for Theorem (6.1).

(6.2) Instead of having to estimate p, we need to estimate the random variable $|p-p_0|$. We define s_p to be the empirical standard deviation 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.

(6.3) 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 (6.1). This completes the proof ▮

Theorem 7 *For all functions $f(h)$ covered by Theorem 6, the sampling algorithm will terminate after at most*

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

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 $E(m,\delta)$ for linear functions (Equation 13); after the log-terms rule out each other in Equation 31 we obtain the desired bound of $\frac{\varepsilon}{2}$.

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

$$\leq 3\sqrt{\frac{\varepsilon^2}{36}} = \frac{\varepsilon}{2} \quad (32)$$

This completes the proof. ▮

4.3 Functions with squared terms

Squared terms (Wrobel, 1997) are introduced to put more emphasis on the the difference between p and the default probability.

Theorem 8 *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*

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

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

$$E_h(m,\delta) = 2s_g z_{1-\frac{\delta}{2}} + s_g^2(z_{1-\frac{\delta}{2}})^2 + s_p z_{1-\frac{\delta}{2}} + 2s_g s_p(z_{1-\frac{\delta}{2}})^2 + s_p s_g^2(+z_{1-\frac{\delta}{2}})^3 \tag{35}$$

Proof. (8.1) $f(h) = g^2(p-p_0)$. As usual, we start in Equation 36 by combining the definition of $E(m,\delta)$ (Equation 33) with Equation 1 which specifies the property of $E(m,\delta)$ that we would like to prove. Again, we introduce the abbreviation E for better readability in Equation 37. In Equation 38 we exploit that $g \leq 1$ and $p - p_0 \leq 1$. In Equation 39 we again use that fact that $P(X < y) \leq P(X < y+z)$. In Equation 40 we add $g^2(p-p_0)$ to both sides of the inequality an start factorizing. In Equation 41 we have identified three factors. The observation that a^2b 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 42; the Chernoff inequality completes this part of the proof.

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

$$\begin{aligned}
&\leq \; Pr\left[\hat{g}^2(\hat{p}-p_0) > \left(g+\sqrt{E}\right)\left(g+\sqrt{E}\right)\left(p-p_0+\sqrt{E}\right)\right] \\
&\quad +Pr\left[\hat{g}^2(\hat{p}-p_0) > \left(g-\sqrt{E}\right)\left(g-\sqrt{E}\right)\left(p-p_0-\sqrt{E}\right)\right] && (41)\\
&\leq \; Pr\left[|\hat{g}-g| > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] + Pr\left[|\hat{p}-p| > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right] && (42)\\
&\leq \; 4\exp\left\{-2m\left(\frac{1}{2m}\log\frac{4}{\delta}\right)\right\} = \delta && (43)
\end{aligned}$$

Let us now look at the normal approximation. First, we will make sure that Equation 34 is a special case of Equation 35 (standard deviation bounded from above). The standard deviation of both g and p is at most $\frac{1}{2\sqrt{m}}$. This takes us from Equation 35 to Equation 44. Equation 45 equals Equation 34.

$$\begin{aligned}
&2s_g z_{1-\frac{\delta}{2}} + s_g^2(z_{1-\frac{\delta}{2}})^2 + s_p z_{1-\frac{\delta}{2}} + 2s_g s_p(z_{1-\frac{\delta}{2}})^2 + s_p s_g^2(z_{1-\frac{\delta}{2}})^3 \\
&\quad \leq \; \frac{1}{\sqrt{m}} z_{1-\frac{\delta}{2}} + \frac{1}{4m}(z_{1-\frac{\delta}{2}})^2 + \frac{1}{2\sqrt{m}} z_{1-\frac{\delta}{2}} + \frac{1}{\sqrt{m}}(z_{1-\frac{\delta}{2}})^2 + \frac{1}{8m\sqrt{m}}(z_{1-\frac{\delta}{2}})^3 && (44)\\
&\quad = \; \frac{3}{2\sqrt{m}} z_{1-\frac{\delta}{2}} + \frac{m+\sqrt{m}}{4m\sqrt{m}}(z_{1-\frac{\delta}{2}})^2 + \frac{1}{8m\sqrt{m}}(z_{1-\frac{\delta}{2}})^3 && (45)
\end{aligned}$$

In Equation 46 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 47 and 48. The union bound takes us to Equation 49; Equation 50 proves the claim.

$$\begin{aligned}
&Pr[|\hat{f}(h,Q_m) - f(h)| > E(m,\delta)] && (46)\\
&\quad = \; Pr[\hat{g}^2(\hat{p}-p_0) - g^2(p-p_0) > 2s_g z_{1-\frac{\delta}{2}} + s_g^2(z_{1-\frac{\delta}{2}})^2 \\
&\qquad +s_p z_{1-\frac{\delta}{2}} + 2s_g s_p(z_{1-\frac{\delta}{2}})^2 + s_p s_g^2(z_{1-\frac{\delta}{2}})^3] \\
&\qquad +Pr[\hat{g}^2(\hat{p}-p_0) - g^2(p-p_0) < -2s_g z_{1-\frac{\delta}{2}} - s_g^2(z_{1-\frac{\delta}{2}})^2 \\
&\qquad -s_p z_{1-\frac{\delta}{2}} - 2s_g s_p(z_{1-\frac{\delta}{2}})^2 - s_p s_g^2(z_{1-\frac{\delta}{2}})^3] \\
&\quad \leq \; Pr\left[\hat{g}^2(\hat{p}-p_0) > \left(g^2 + 2g s_g z_{1-\frac{\delta}{2}} + s_g^2(z_{1-\frac{\delta}{2}})^2\right)\left(p-p_0+s_p z_{1-\frac{\delta}{2}}\right)\right] \\
&\qquad +Pr\left[\hat{g}^2(\hat{p}-p_0) < \left(g^2 - 2g s_g z_{1-\frac{\delta}{2}} + s_g^2(z_{1-\frac{\delta}{2}})^2\right)\left(p-p_0-s_p z_{1-\frac{\delta}{2}}\right)\right] && (47)\\
&\quad \leq \; Pr\left[\hat{g}^2(\hat{p}-p_0) > (g+s_g z_{1-\frac{\delta}{2}})(g+s_g z_{1-\frac{\delta}{2}})(p-p_0+s_p z_{1-\frac{\delta}{2}})\right] \\
&\qquad +Pr\left[\hat{g}^2(\hat{p}-p_0) < (g-s_g z_{1-\frac{\delta}{2}})(g-s_g z_{1-\frac{\delta}{2}})(p-p_0-s_p z_{1-\frac{\delta}{2}})\right] && (48)\\
&\quad \leq \; Pr\left[|\hat{g}-g| > s_g z_{1-\frac{\delta}{2}}\right] + Pr\left[|\hat{p}-p| > s_p z_{1-\frac{\delta}{2}}\right] && (49)\\
&\quad \leq \; 2\left(\frac{\delta}{4}+\frac{\delta}{4}\right) = \delta && (50)
\end{aligned}$$

This proves case (8.1). For cases (8.2) and (8.3), 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 (8.1). ∎

Theorem 9 *For all functions $f(h)$ covered by Theorem 8, the sampling algorithm will terminate after at most*

$$m = \frac{98}{\varepsilon^2} \log \frac{8|H_i|}{\delta} \tag{51}$$

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 8 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 4. 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 $E(m, \delta)$ as in Equation 53.

$$\begin{aligned} E(m,\delta) &= \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{2}} + 3\left(\frac{1}{2m}\log\frac{4}{\delta}\right) + 3\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} && (52)\\ &< 7\sqrt{\frac{1}{2m}\log\frac{4}{\delta}} && (53) \end{aligned}$$

Now we will show that $E(m, \delta)$ lies below $\frac{\varepsilon}{2}$ when m reaches the bound described in Equation 51. We insert the sample bound into the exit criterion in Equation 54. The log-terms rule out each other and the result is $\frac{\varepsilon}{2}$ as desired.

$$\begin{aligned} E\left(\frac{98}{\varepsilon^2}\log\frac{8|H|}{\delta}, \frac{\delta}{2|H|}\right) &< 7\sqrt{\frac{1}{2\left(\frac{98}{\varepsilon^2}\log\frac{8|H|}{\delta}\right)}\log\frac{4}{\left(\frac{\delta}{2|H|}\right)}} && (54)\\ &\leq 7\sqrt{\frac{\varepsilon^2}{4\cdot 49}} = \frac{\varepsilon}{2} && (55) \end{aligned}$$

This completes the proof. ∎

4.4 Functions Based on the Binomial Test

The Binomial test heuristic (Klösgen, 1992) 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 10 *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*

$$\text{small } m: \quad E(m,\delta) = \sqrt[2]{\frac{1}{2m}\log\frac{4}{\delta}} + \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{4}} \tag{56}$$

$$\text{large } m: \quad E(m,\delta) = \sqrt{\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}} + \frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}} + \left(\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}\right)^{3/2} \tag{57}$$

$$E_h(m,\delta) = \sqrt{s_g z_{1-\frac{\delta}{4}}} + s_p z_{1-\frac{\delta}{4}} + \sqrt{s_g z_{1-\frac{\delta}{4}} s_p z_{1-\frac{\delta}{4}}} \tag{58}$$

Proof. (10.1) In Equation 59, we insert Equation 56 into Equation 1 (the definition of $E(m,\delta)$). We refer to the union bound in Equation 61 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 62 and use the union bound in Equation 63. Now in Equation 64 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 64 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 65 and 66 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 67, the Chernoff inequality leads to the desired result of δ.

$$\begin{aligned}
& Pr[|\hat{f}(h,Q_m) - f(h)| > E(m,\delta)] \\
= \; & Pr\left[|\sqrt{\hat{g}}(\hat{p}-p_0) - \sqrt{g}(p-p_0)| > \sqrt[2]{\frac{1}{2m}\log\frac{4}{\delta}} + \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{4}}\right] && (59) \\
= \; & Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) - \sqrt{g}(p-p_0) > \sqrt[2]{E} + \sqrt[4]{E} + E^{\frac{3}{4}}\right] \\
& +Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) - \sqrt{g}(p-p_0) < -\sqrt[2]{E} - \sqrt[4]{E} - E^{\frac{3}{4}}\right] && (60) \\
\leq \; & Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) - \sqrt{g}(p-p_0) > \sqrt{g}\sqrt{E} + (p-p_0)\sqrt[4]{E} + E^{\frac{3}{4}}\right] \\
& +Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) - \sqrt{g}(p-p_0) < -\sqrt{g}\sqrt{E} - (p-p_0)\sqrt[4]{E} - E^{\frac{3}{4}}\right] && (61) \\
\leq \; & Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) > \left(\sqrt{g}+\sqrt[4]{E}\right)\left(p-p_0+\sqrt{E}\right)\right] \\
& +Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) < \left(\sqrt{g}-\sqrt[4]{E}\right)\left(p-p_0-\sqrt{E}\right)\right] && (62) \\
\leq \; & Pr\left[|\sqrt{\hat{g}} - \sqrt{g}| > \sqrt[4]{E}\right] + Pr\left[|\hat{p}-p| > \sqrt{E}\right] && (63) \\
\leq \; & Pr\left[|\sqrt{\hat{g}} - \sqrt{g}| > \sqrt{\sqrt{E} - 2\left(\sqrt{g^2 + g\sqrt{E}} - \sqrt{g^2}\right)}\right] + 2\exp\left\{-2m\frac{1}{2m}\log\frac{4}{\delta}\right\} && (64) \\
= \; & 2Pr\left[\sqrt{\hat{g}} - \sqrt{g} > \sqrt{2g + \sqrt{E} - 2\sqrt{g\left(g+\sqrt{E}\right)}}\right] + \frac{\delta}{2} && (65)
\end{aligned}$$

$$= \quad 2Pr\left[\sqrt{\hat{g}}-\sqrt{g} > \sqrt{g+\sqrt{\frac{1}{2m}\log\frac{4}{\delta}}}-\sqrt{g}\right]+\frac{\delta}{2} \tag{66}$$

$$= \quad 2Pr\left[\hat{g}-g > \sqrt{\frac{1}{2m}\log\frac{4}{\delta}}\right]+\frac{\delta}{2}=\delta \tag{67}$$

Now we still need to prove the normal approximations (Equations 57 and 58). As usual, we would like Equation 57 to be a special case of Equation 58 with the standard deviations bounded from above. Equation 68 confirms that this is the case since $s_{p,g} \leq \frac{1}{2\sqrt{m}}$.

$$\sqrt{s_g z_{1-\frac{\delta}{4}}}+s_p z_{1-\frac{\delta}{4}}+\sqrt{s_g z_{1-\frac{\delta}{4}}}s_p z_{1-\frac{\delta}{4}} \quad \leq \quad \sqrt{\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}}-\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}+\sqrt{\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}}}\frac{z_{1-\frac{\delta}{4}}}{2\sqrt{m}} \tag{68}$$

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 70) and then factor the right hand side (Equation 71). We subtract a small number from $s_g z_{1-\frac{\delta}{4}}$ in Equation 72 and factor $\sqrt{\hat{g}}-\sqrt{g}$ in Equation 73 and Equation 74. Basic manipulations and the Chernoff inequality complete the proof in Equation 76.

$$\begin{aligned}
&Pr[|\hat{f}(h,Q_m)-f(h)|>E(m,\delta)] \\
&\leq \quad Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0)-\sqrt{g}(p-p_0) > \sqrt{s_g z_{1-\frac{\delta}{4}}}+s_p z_{1-\frac{\delta}{4}}+\sqrt{s_g z_{1-\frac{\delta}{4}}}s_p z_{1-\frac{\delta}{4}}\right] \\
&\qquad +Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0)-\sqrt{g}(p-p_0) < -\sqrt{s_g z_{1-\frac{\delta}{4}}}-s_p z_{1-\frac{\delta}{4}}-\sqrt{s_g z_{1-\frac{\delta}{4}}}s_p z_{1-\frac{\delta}{4}}\right] &(69)\\
&\leq \quad Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0)-\sqrt{g}(p-p_0) > (p-p_0)\sqrt{s_g z_{1-\frac{\delta}{4}}}+\sqrt{g}s_p z_{1-\frac{\delta}{4}}+s_p z_{1-\frac{\delta}{4}}\sqrt{s_g z_{1-\frac{\delta}{4}}}\right] &(70)\\
&\qquad +Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0)-\sqrt{g}(p-p_0) < -(p-p_0)\sqrt{s_g z_{1-\frac{\delta}{4}}}-\sqrt{g}s_p z_{1-\frac{\delta}{4}}-s_p z_{1-\frac{\delta}{4}}\sqrt{s_g z_{1-\frac{\delta}{4}}}\right] \\
&\leq \quad Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) > \left(\sqrt{g}+\sqrt{s_g z_{1-\frac{\delta}{4}}}\right)\left(p-p_0+s_p z_{1-\frac{\delta}{4}}\right)\right] \\
&\qquad +Pr\left[\sqrt{\hat{g}}(\hat{p}-p_0) < \left(\sqrt{g}-\sqrt{s_g z_{1-\frac{\delta}{4}}}\right)\left(p-p_0-s_p z_{1-\frac{\delta}{4}}\right)\right] &(71)\\
&\leq \quad Pr\left[|\sqrt{\hat{g}}-\sqrt{g}| > \sqrt{s_g z_{1-\frac{\delta}{4}}}\right]+Pr\left[|\hat{p}-p| > s_p z_{1-\frac{\delta}{4}}\right] &(72)\\
&\leq \quad Pr\left[|\sqrt{\hat{g}}-\sqrt{g}| > \sqrt{s_g z_{1-\frac{\delta}{4}}-2\left(\sqrt{g^2+g s_g z_{1-\frac{\delta}{4}}}+\sqrt{g^2}\right)}\right]+\frac{\delta}{2} &(73)\\
&\leq \quad Pr\left[|\sqrt{\hat{g}}-\sqrt{g}| > \sqrt{2g+s_g z_{1-\frac{\delta}{4}}-2\sqrt{\left(g(g+s_g z_{1-\frac{\delta}{4}})\right)}}\right]+\frac{\delta}{2} &(74)
\end{aligned}$$

$$\leq \quad Pr\left[|\sqrt{\hat{g}} - \sqrt{g}| > \sqrt{g + s_g z_{1-\frac{\delta}{4}}} - \sqrt{g}\right] + \frac{\delta}{2} \tag{75}$$

$$\leq \quad Pr\left[|\hat{g} - g| > s_g z_{1-\frac{\delta}{4}}\right] + \frac{\delta}{2} = \delta \tag{76}$$

This completes the proof for Theorem (10.1). The proofs of cases (10.2) and (10.2) 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 10. ∎

Theorem 11 *For all functions $f(h)$ covered by Theorem 10, the sampling algorithm will terminate after at most*

$$m = \frac{648}{\varepsilon^2} \log \frac{8|H_i|}{\delta} \tag{77}$$

database queries (but usually much earlier).

Proof. The middle term of Equation 57 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 78 provides us with an easier bound.

$$\sqrt[2]{\frac{1}{2m}\log\frac{4}{\delta}} + \sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} + \left(\frac{1}{2m}\log\frac{4}{\delta}\right)^{\frac{3}{4}} \quad \leq \quad 3\sqrt[4]{\frac{1}{2m}\log\frac{4}{\delta}} \tag{78}$$

The algorithm terminates in step 3 when $E(m, \frac{\delta}{2|H|}) \leq \frac{\varepsilon}{2}$. Considering the sample bound in Equation 77, Equation 79 proves that this is the case with guarantee. Note that, since we bounded the confidence interval quite loosely, we expect the algorithm to terminate considerably earlier.

$$E\left(\frac{648}{\varepsilon^2}\log\frac{8|H|}{\delta}, \frac{\delta}{2|H|}\right) \quad < \quad 3\sqrt[4]{\frac{1}{2\left(\frac{648}{\varepsilon^2}\log\frac{8|H|}{\delta}\right)}\log\frac{4}{\left(\frac{\delta}{2|H|}\right)}} \tag{79}$$

$$\leq \quad 3\sqrt[4]{\frac{\varepsilon^2}{16 \cdot 81}} = \frac{\varepsilon}{2} \tag{80}$$

This completes the proof. ∎

4.5 Negative Results

Several independent impurity criteria have led to utility functions that are factor-equivalent to $f(h) = \frac{g}{1-g}(p - p_0)^2$; *e.g.,* Gini diversity index and twoing criterion (Breiman et al., 1984), and the chi-square test (Piatetski-Shapiro, 1991). Note that it is also order-equivalent to the utility measure used in Inferrule (Uthurusamy et al., 1991). 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 12 *There is no algorithm that satisfies Theorem 4 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 81).

$$\frac{g+\varepsilon}{1-(g+\varepsilon)} - \frac{g}{1-g} = \frac{\varepsilon}{(g+\varepsilon-1)(g-1)} \tag{81}$$

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 82).

$$\frac{g+\varepsilon}{1-(g+\varepsilon)} \Big/ \frac{g}{1-g} = \frac{(g+\varepsilon)(1-g)}{g(1-g-\varepsilon)} \tag{82}$$

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 sampling can reach orders of magnitude. Consider, for instance, the linear utility function and $k = 1, \varepsilon = .1, \delta = .1$. 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.

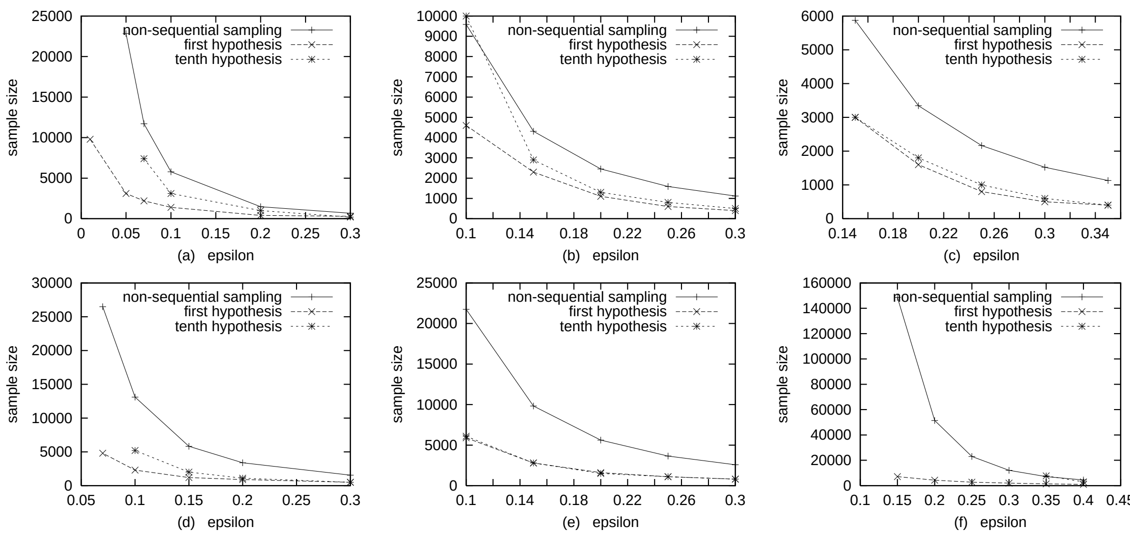

Figure 1: Sample sizes for the juice purchases database. (a) $f = g|p - p_0|$, $k = 1$, $\delta = .1$; (b) $k = 2$; (c) $k = 3$; (d) $f = g^2|p - p_0|$, $k = 1$, $\delta = .1$; (e) $k = 2$; (f) $f = \sqrt{g}|p - p_0|$, $k = 1$, $\delta = .1$

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

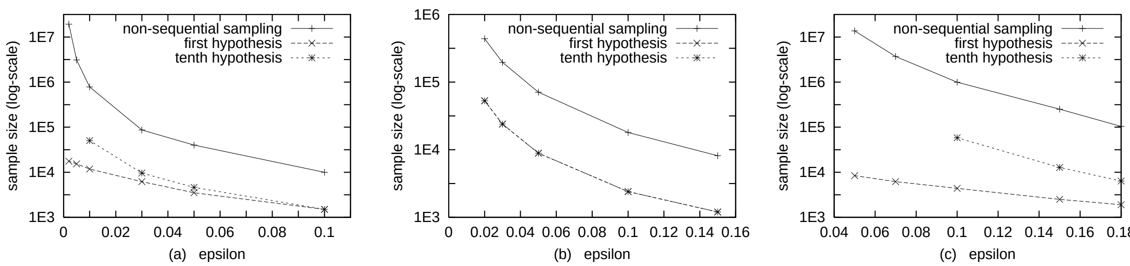

Figure 2: Required sample sizes (log-scale) for the KDD cup data of 1998. (a) $f = g|p - p_0|$, $k = 1$, $\delta = .1$; (b) $f = g^2|p - p_0|$, (c) $f = \sqrt{g}|p - p_0|$.

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 (Dodge and Romig, 1929, Wald, 1947, Ghosh et al., 1997). Maron, Moore, & Lee (Maron and Moore, 1994, Moore and Lee, 1994) 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 (1996).

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 (1996); a sequential algorithm by Domingo et al. (1998, 1999). For the special case of decision trees, the algorithm of Hulten and Domingos (2000) samples a database and finds a hypothesis that is similar to the one that C4.5 would have obtained.

A preliminary version of the algorithm presented in this paper has been discussed by Scheffer and Wrobel (2000). 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 (1996) 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. (1999) 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 functions for knowledge discovery tasks and showed that there is no solution for one function. Another minor difference between our algorithm and the one of Domingo et al. (1999) 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 (Haussler et al., 1996, Langford and McAllester, 2000) and self-bounding learning algorithms (Freund, 1998).

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 here, the GSS 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).

For large hypothesis spaces such as decision trees, it is possible to define a generating function that can enumerate all hypotheses although they do not fit into the main memory. Based on the GSS algorithm presented here, Scheffer and Wrobel (2002) have developed a constant memory algorithm that provides identical guarantees but uses only constant memory, independent of the size of the hypothesis space.

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.

Acknowledgements

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

The research reported here was partially supported by Grant "Information Fusion / Active Learning" of the German Research Council and was partially carried out when Stefan Wrobel was at the University of Magdeburg.

Appendix A. Appendix: Proof of Theorem 4

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 $G = G_{i_{fin}}$.

Table 2: Symbols used in the proof of Theorem 4

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

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

$$\forall i \in \{1,\ldots,i_{fin}\} : H = G_i \cup H_i \cup R_i \tag{83}$$

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

M is the sample-independent worst-case bound.

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: $i \to i+1$

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.,

$$G_{i+1} := G_i \cup \{h\}, H_{i+1} := H_i \backslash \{h\}, n_{i+1} := n_i - 1, R_{i+1} := R_i \tag{84}$$

From the algorithm, we know:

$$\hat{h}^{(i)} \geq E_h\left(i, \frac{\delta}{2M|H_i|}\right) + max_{h' \in \overline{H_i^*}} \left\{ \hat{h}'^{(i)} + E_{h'}(i, \frac{\delta}{2M|H_i|}) \right\} - \varepsilon \tag{85}$$

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 85 becomes Equation 86.

$$\hat{h} \geq E_h(\cdot) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - \varepsilon \tag{86}$$

Now let $h'' \in \overline{H^*}$. We can rearrange Equation 86 by adding paired terms in Equation 87.

$$\begin{aligned} \hat{h} + h - h &\geq h'' - h'' + E_h(\cdot) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - \varepsilon & (87) \\ \Leftrightarrow h &\geq h'' + E_h(\cdot) - (\hat{h} - h) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - h'' - \varepsilon & (88) \end{aligned}$$

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 89

$$\begin{aligned} h &\geq h'' + E_h(\cdot) - (\hat{h} - h) + \hat{h}'' + E_{h''}(\cdot) - h'' - \varepsilon & (89) \\ \Leftrightarrow h &\geq h'' + E_h(\cdot) - (\hat{h} - h) + E_{h''}(\cdot) - (h'' - \hat{h}'') - \varepsilon & (90) \end{aligned}$$

According to (A1), $\hat{h} - h \leq E_h(\cdot)$, and $h'' - \hat{h}'' \leq E_{h''}(\cdot)$, so Equation 91 must hold.

$$h \geq h'' - \varepsilon \text{ for all } h'' \in \overline{H^*} \tag{91}$$

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 92 holds.

$$\forall r \in R_{i+1} : h \geq r - \varepsilon \tag{92}$$

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 93 holds.

$$\exists^* h'_1, \ldots, h'_{n_i} \in H_i : h'_j \geq r \quad \forall j \in \{1, \ldots, n_i\} \tag{93}$$

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 (92) immediately from (93), as in Equation 97.

$$h = h'_{j_1} \geq r \geq r - \varepsilon \tag{97}$$

Otherwise, one of the h'_j, say h'_{j_2}, must be from $\overline{H^*}$; in Equation 98, we use (91) and (93) to show Equation 92.

$$h \geq h'_{j_2} - \varepsilon \geq r - \varepsilon \tag{98}$$

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 (98) 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 99 follows.

$$\forall r \in R_i \exists^* h'_1, \ldots, h'_{n_i} \in H_i : h'_j \geq r \quad \forall j \in \{1, \ldots, n_i\} \tag{99}$$

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 100 must hold. Since $n_{i+1} = n_i - 1$, this implies Equation 101.

$$\forall r \in R_{i+1} \exists^* h'_1, \ldots, h'_{n_i - 1} \in H_{i+1} : h'_j \geq r \quad \forall j \in \{1, \ldots, n_i - 1\} \tag{100}$$

$$\Rightarrow \quad \forall r \in R_{i+1} \exists^* h'_1, \ldots, h'_{n_{i+1}} \in H_{i+1} : h'_j \geq r \quad \forall j \in \{1, \ldots, n_{i+1}\} \tag{101}$$

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 102 holds.

$$\neg \exists^* h'_1, \ldots, h'_{n_{i+1}+1} \in H_{i+1} : h'_j > h + \varepsilon \, \forall j \in \{1, \ldots, n_{i+1} + 1\} \tag{102}$$

From (91) we know that $h \geq h'' - \varepsilon \Leftrightarrow h'' \leq h + \varepsilon \, \forall h'' \in \overline{H^*}$, so any h'_j to violate Equation 102 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 102.

1. Note (86) alone is not enough to guarantee that h is in H^*, since even if $h \in \overline{H^*}$ were true, we would have

$$\hat{h} \quad > \quad E_h(\cdot) + max_{h' \in \overline{H^*}}[\hat{h}' + E_{h'}(\cdot)] - \varepsilon \tag{94}$$

$$\Rightarrow \hat{h} \quad > \quad E_h(\cdot) + \hat{h} + E_h(\cdot) - \varepsilon \tag{95}$$

$$\Leftrightarrow E_h(\cdot) \quad < \quad \frac{\varepsilon}{2} \tag{96}$$

which cannot be excluded.

Case (2) A hypothesis h is removed in step 3(e)ii, *i.e.,* the statements of Equation 103 are executed.

$$G_{i+1} := G_i, H_{i+1} := H_i \backslash \{h\}, R_{i+1} := R_i \cup \{h\}, n_{i+1} := n_i \tag{103}$$

From the algorithm, we know that Equation 104 must hold.

$$\hat{h}^{(i)} \leq min_{h' \in H_i^*} \left\{ \hat{h}'^{(i)} - E_{h'}\left(i, \frac{\delta}{2M|H_i|}\right) \right\} - E_h\left(i, \frac{\delta}{2M|H_i|}\right) \tag{104}$$

Analogously to the previous case, we will abbreviate Equation 104 as Equation 105.

$$\hat{h} \leq min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] - E_h(\cdot) \tag{105}$$

We now proceed in a similar fashion as in Equations 87 to 91. So let $h'' \in H^*$, and rearrange Equation 105 by adding paired terms in Equation 106.

$$\begin{aligned} \hat{h} + h - h &\leq h'' - h'' + min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] - E_h(\cdot) && (106) \\ \Leftrightarrow h &\leq h'' + (h - \hat{h}) - E_h(\cdot) + min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] - h'' && (107) \end{aligned}$$

Now, since certainly $min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] \leq \hat{h}'' - E_{h''}(\cdot)$ for all $h'' \in H^*$, Equations 108 and 109 are equivalent.

$$\begin{aligned} h &\leq h'' + (h - \hat{h}) - E_h(\cdot) + \hat{h}'' - E_{h''} - h'' && (108) \\ \Leftrightarrow h &\leq h'' + (h - \hat{h}) - E_h(\cdot) + (\hat{h}'' - h'') - E_{h''} && (109) \end{aligned}$$

If (A1) is assumed true, then $h - \hat{h} \leq E_h(\cdot)$, and $\hat{h}'' - h'' \leq E_{h''}$, so Equation 110 must be true.

$$h \leq h'' \text{ for all } h'' \in H^* \tag{110}$$

We can now show the three invariant conditions.

Show (I1) Since $G_{i+1} := G_i$, by inductive assumption (I1) we have Equation 111.

$$\forall g \in G_{i+1} \, \forall r \in R_i : g \geq r - \varepsilon \tag{111}$$

Since $R_{i+1} = R_i \cup \{h\}$, we only need to show that Equation 112 is true.

$$\forall g \in G_{i+1} \; g \geq h - \varepsilon \tag{112}$$

Note that $h \in \overline{H^*}$, since due to Equation 105, $h \in H^*$ would imply that $E_h(\cdot) \leq 0$, as demonstrated in Equations 113 to 116. However, $E_h(\cdot) > 0$ for all h and all arguments of $E_h(\cdot)$ which means that $h \in \overline{H^*}$ and thus $|H^* \cup \{h\}| = n_i + 1$.

$$\begin{aligned} \hat{h} &\leq min_{h' \in H^*} [\hat{h}' - E_{h'}(\cdot)] - E_h(\cdot) && (113) \\ \Rightarrow \hat{h} &\leq \hat{h} - E_h(\cdot) - E_h(\cdot) && (114) \\ \Leftrightarrow 0 &\leq -2E_h(\cdot) && (115) \\ \Leftrightarrow E_h(\cdot) &\leq 0 && (116) \end{aligned}$$

By inductive assumption (I3) we have Equation 117 for all $g \in G_{i+1}$.

$$\forall g \in G_{i+1} = G_i \, \forall h'_1, \ldots, h'_{n_i+1} \in H_i \exists j \in \{1, \ldots, n_i + 1\} : g \geq h'_j - \varepsilon \tag{117}$$

From Equation 117, 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 112 immediately; if not, then $h' \in H^*$, so from Equation 110 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 118.

$$\exists^* h'_1, \ldots, h'_{n_{i+1}} \in H_{i+1} : h'_j \geq h \; \forall j \in \{1, \ldots, n_i + 1\} \tag{118}$$

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 (110), we can simply choose $h'_1, \ldots, h'_{n_{i+1}}$ to be H^*, thus showing (118). Then consider $r \in R_i$. By inductive assumption, since $n_i = n_{i+1}$, we have Equation 119.

$$\exists^* h'_1, \ldots, h'_{n_{i+1}} \in H_i : h'_j \geq r \; \forall j \in \{1, \ldots, n_{i+1}\} \tag{119}$$

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 119 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 120 must be satisfied.

$$\forall g \in G_{i+1} \neg \exists^* h'_1, \ldots, h'_{n_{i+1}+1} \in H_i : h'_j > g + \varepsilon \; \forall j \in \{1, \ldots, n_{i+1} + 1\} \tag{120}$$

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 121 is true.

$$\text{(I1), (I2), and (I3) are true whenever we exit the loop.} \tag{121}$$

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 122.

$$G_{i_{fin}} := G_{imax} \cup H_{i_{max}}, H_{i_{fin}} := \emptyset, R_{i_{fin}} := R_{i_{max}}, n_{i_{fin}} = 0 \tag{122}$$

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 123.

$$\forall h \in H_{i_{max}} \; \forall r \in R_{i_{max}} : h \geq r - \varepsilon \tag{123}$$

So consider any particular $h \in H_{i_{max}}$, $r \in R_{i_{max}}$. From (I2), we know that Equation 124 holds.

$$\exists^* h'_1, \ldots, h'_{n_{i_{max}}} \in H_{i_{max}} : h'_j \geq r \; \forall j \in \{1, \ldots, n_{i_{max}}\} \tag{124}$$

But since $|H_{i_{max}}| = n_{i_{max}}$, h must be among the h_j, which shows Equation 123.

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 $E(i, \frac{\delta}{2|H_{i_{max}}|}) \leq \frac{\varepsilon}{2}$. Then after step 4, the conditions displayed in Equation 125 hold.

$$G_{i_{fin}} := G_{i_{max}} \cup H^*_{i_{max}}, H_{i_{fin}} := \overline{H^*_{i_{max}}}, R_{i_{fin}} := R_{i_{max}}, n_{i_{fin}} = 0 \tag{125}$$

So if we assume (A2) to be true, we can conclude that Equation 126 is true.

$$\forall h \in H^*_{i_{max}} \ \forall h' \in \overline{H^*_{i_{max}}} : h \geq h' - \varepsilon \tag{126}$$

Show I1 (I1) is true for all $g \in G_{i_{max}}$ by Equation 121, and true for $H^*_{i_{max}}$ according to Equation 126, 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 127.

$$\forall g \in G_{i_{fin}} = G_{i_{max}} \cup H^*_{i_{max}};\ \forall h \in H_{i_{fin}} = \overline{H^*_{i_{max}}} : g \geq h - \varepsilon \tag{127}$$

Equation 127 again is true due to Equation 126. 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 128 holds.

$$\forall g \in G_{i_{fin}} \neg \exists r \in R_{i_{fin}} : r > g + \varepsilon \tag{128}$$

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 83, $R_{i_{fin}} \cup H_{i_{fin}} = H \backslash G_{i_{fin}}$, 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 13 *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 $E(m, \delta)$, Equation 129 follows.

$$\forall 0 < \delta_1 \leq \delta_2 < 1,\ \forall m \in \mathbb{N} : E(m, \delta_1) \geq E(m, \delta_2) \tag{129}$$

Thus, Equation 130 shows that the algorithms stops when $i = M$ at the latest.

$$E_h\left(M, \frac{\delta}{2M|H|}\right) \leq \frac{\varepsilon}{2} \Rightarrow E_h\left(M, \frac{\delta}{2M|H_i|}\right) \leq \frac{\varepsilon}{2} \tag{130}$$

Using this, in Equations 131 and 132, we refer to the union bound. In Equation 133, we apply the definition of $E(m, \delta)$. Equation 133 completes the proof.

$$
\begin{aligned}
& Pr\left[\exists i, \exists h \in H_i : |\hat{f}(h,Q_i) - f(h)| > E_h\left(i, \frac{\delta}{2M|H_i|}\right)\right] \\
\leq \quad & \sum_{i=1}^{M} Pr\left[\exists h \in H_i : |\hat{f}(h,Q_i) - f(h)| > E_h\left(i, \frac{\delta}{2M|H_i|}\right)\right] && (131) \\
\leq \quad & \sum_{i=1}^{M} |H_i| Pr\left[|\hat{f}(h,Q_i) - f(h)| > E_h\left(i, \frac{\delta}{2M|H_i|}\right)\right] && (132) \\
< \quad & \sum_{i=1}^{M} |H_i| \frac{\delta}{2M|H_i|} = \frac{\delta}{2} && (133)
\end{aligned}
$$

Now consider (A2).

Lemma 14 *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 134 holds.*

$$\forall h \in H^*_{i_{max}}\ \forall h' \in \overline{H^*_{i_{max}}} : h \geq h' - \varepsilon \qquad (134)$$

Proof. If $h \in H^*_{i_{max}}$ and $h' \in \overline{H^*_{i_{max}}}$, then $\hat{h} \geq \hat{h}'$. If nonetheless Equation 134 is true, we must have estimated at least one of the two hypotheses wrong by more than $\frac{\varepsilon}{2}$. We can bound the probability of this as follows. Equation 135 follows from the definition of M in step 2 of the algorithm. We refer to the union bound in Equation 136 and to the definition of $E(m,\delta)$ (Equation 1) in Equation 136.

$$
\begin{aligned}
& Pr\left[\exists h \in H_{i_{max}} : \hat{h} - h > \frac{\varepsilon}{2}\right] \\
\leq \quad & Pr\left[\exists h \in H_{i_{max}} : \hat{h} - h > E\left(i_{max}, \frac{\delta}{2|H_{i_{max}}|}\right)\right] && (135) \\
\leq \quad & |H_{i_{max}}| Pr\left[\hat{h} - h > E\left(i_{max}, \frac{\delta}{2|H_m|}\right)\right] \leq \frac{\delta}{2} && (136)
\end{aligned}
$$

This completes the proof of Lemma 14 and thereby the proof of Theorem 4. ∎

References

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.

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

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

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

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

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.

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

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

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

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

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

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

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

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

G. Hulten and P. Domingos. Mining high-speed data streams. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2000.

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

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

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

W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI, 1996.

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

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.

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.

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

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.

T. Scheffer and S. Wrobel. Hidden markov models for text classification and information extraction. Technical report, University of Magdeburg, 2002.

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

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

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

A. Wald. *Sequential Analysis*. Wiley, 1947.

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.

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.