Journal of Artificial Intelligence Research 10 (1999) 375-397 Submitted 2/99; published 6/99

Efficient Heuristic Hypothesis Ranking

Steve Chien STEVE.CHIEN@JPL.NASA.GOV
Andre Stechert ANDRE.STECHERT@JPL.NASA.GOV
Darren Mutz DARREN.MUTZ@JPL.NASA.GOV
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, M/S 126-347
Pasadena, CA 91109-8099

Abstract

This paper considers the problem of learning the ranking of a set of stochastic alternatives based upon incomplete information (i.e., a limited number of samples). We describe a system that, at each decision cycle, outputs either a complete ordering on the hypotheses or decides to gather additional information (i.e., observations) at some cost. The ranking problem is a generalization of the previously studied hypothesis selection problem—in selection, an algorithm must select the single best hypothesis, while in ranking, an algorithm must order all the hypotheses.

The central problem we address is achieving the desired ranking quality while minimizing the cost of acquiring additional samples. We describe two algorithms for hypothesis ranking and their application for the probably approximately correct (PAC) and expected loss (EL) learning criteria. Empirical results are provided to demonstrate the effectiveness of these ranking procedures on both synthetic and real-world datasets.

1. Introduction

In many applications, the cost of information can be quite high, imposing a requirement that learning algorithms glean as much usable information as possible with a minimum of data. For example:

- Data may be scarce, making learning the most possible from limited training data key.
- In speedup learning, minimizing processing time is critical. Here, reducing the number of necessary training examples is key since the expense of processing each example can be significant (Tadepalli, 1992).
- In decision tree learning, the cost of using all available training examples when evaluating potential attributes for partitioning can be computationally expensive (Musick, Catlett, & Russell, 1993).
- In evaluating medical treatment policies, acquiring additional training examples might imply that human subjects are exposed to an experimental treatment for a longer period than is necessary.

When one wishes some sort of guarantee on the quality of a solution, a statistical decision theoretic framework is useful. The framework answers the questions: How much information is enough? At what point do we have adequate information to rank the alternatives with some requested confidence?

This paper focuses on parametric ranking problems, a general class of statistical machine learning problems in which the goal is to rank a set of alternative hypotheses where the goodness of a hypothesis is a function of a set of parameters whose values are unknown (e.g., Chien, Stechert, & Mutz, 1998; Gratch, 1992; Greiner & Jurisica, 1992; Kaelbling, 1993; Moore & Lee, 1994; Musick et al., 1993). The learning system determines and refines estimates of these parameters by using training examples, with a secondary goal of minimizing learning cost.

The principal contributions of this paper are:

- We define two families of hypothesis ranking algorithms, based on recursive selection and adjacency, respectively. We provide specific details on how to apply them using the probably approximately correct (PAC) and expected loss (EL) decision criteria.
- We provide empirical results demonstrating the effectiveness of these algorithms at achieving the requested decision criteria on synthetic data.
- We provide empirical results showing that these algorithms significantly outperform existing statistical methods on real-world data from spacecraft design optimization and image compression applications.

The remainder of this paper is structured as follows. First, we describe the hypothesis ranking problem more formally, including definitions for the probably approximately correct (PAC) and expected loss (EL) decision criteria. We then define two algorithms for establishing these criteria for the hypothesis ranking problem—a recursive hypothesis selection algorithm and an adjacent comparison algorithm. Next, we describe empirical tests demonstrating the effectiveness of these algorithms as well as documenting their improved performance over a standard algorithm from the statistical ranking literature. Finally, we describe related work and future extensions to the algorithms.

2. Hypothesis Ranking Problems

Hypothesis ranking problems are an abstract class of learning problems where an algorithm is given a set of hypotheses to rank. The ranking desired is that which orders the hypotheses by their *expected utility*, which is determined by the hypothesis' underlying probability distribution. These expected utilities are unknown to the algorithm and must be estimated from the training data.

Hypothesis ranking problems are an extension of hypothesis selection problems (Chien, Gratch, & Burl, 1995), in which a learning system attempts to select the best alternative from a set of hypotheses. The distinction between hypothesis ranking and hypothesis selection is that in selection the learning algorithm is interested in a single best hypothesis, while in ranking the learning algorithm must determine the relative order of *all* of the hypotheses[1].

Hypothesis selection and ranking is an important aspect of many machine learning problems. For example, the utility problem in speedup learning can be viewed as a selection

[1] The algorithms and results described in this paper extend in a straightforward fashion to hybrid ranking-selection problems in which the system must select and rank the top M out of N hypotheses.

problem where a single problem-solving heuristic or strategy is chosen from a larger set of candidates. In this case, the expected utility is typically defined as the average time to solve a problem (Gratch, 1992; Greiner & Jurisica, 1992; Minton, 1988). The attribute selection problem in machine learning can also be viewed as a hypothesis selection problem in which one must select the best attribute split from a set of possible attribute splits and utility is often measured by information gain (Musick et al., 1993). In reinforcement learning, a system must learn the appropriate action for each context, where utility is interpreted as expected reward (Kaelbling, 1993).[2]

A key observation regarding each of these problems (and all learning problems, in general) is that each of them could be viewed as an optimization problem, where the utility is the function being optimized. Then, the application of traditional (or non-traditional) optimization methods will yield good results within the guarantees provided by the algorithm and depending on the features of the landscape being optimized. However, with the addition of a model of sampling cost, a new degree of freedom is added to the problem. Where the cost of samples is very high, traditional optimization algorithms will fare poorly.

Additionally, while in many of the mentioned applications the system chooses a single alternative and never revisits the decision, there are also many cases for which a system will want to investigate several prioritized options (either serially or in parallel), and hence a ranking is useful. Motivation is provided by the following scenarios:

- *Upper and lower bounds, span*: Minimax search algorithms can use metaknowledge (such as upper and lower bounds of a node) for pruning other parts of the tree. Also, there are times when knowing the span of the expected utilities of the candidate set is useful (e.g., when checking for convergence conditions in an adaptive algorithm such as a GA).

- *Augmenting external knowledge*: Another area in which hypothesis ranking may have important applications is hypothesis selection with human supervision. When the stochastic objective function (i.e., the hypothesis) represents only a part of the problem, the ranking can be used to augment external knowledge of the problem. For example, engineering simulations usually capture the physical properties of the candidate designs, but usually choose to forego the details of manufacturing, logistics, and economics.

- *The entire ranking*: In some cases, the entire ranking is significant. For instance, in evolutionary algorithms, the individuals to be propagated to future generations are often selected with likelihood that is proportionate to their rank in the current generation (Goldberg, 1989). Another example arises in the case of search algorithms that take advantage of node ordering heuristics, such as beam search or iterative broadening (Ginsberg & Harvey, 1992).

In any hypothesis evaluation problem, always achieving a correct ranking is impossible in practice, because the exact underlying probability distributions are unknown. Thus, there is always a (perhaps vanishingly) small chance that the algorithms will be unlucky

[2] Note that the analogous reinforcement learning problem is the one in which we are learning the appropriate action with immediate feedback rather than delayed feedback.

because only a finite number of samples can be taken. Consequently, rather than always requiring an algorithm to output a correct ranking, we impose probabilistic criteria on the rankings to be produced. While several families of such requirements exist, in this paper we examine two criteria: the *probably approximately correct* (PAC) model for selecting a hypothesis function that approximates well a target function (Valiant, 1984) and the *expected loss* (EL) requirement frequently used in decision theory and gaming problems (Russell & Wefald, 1992). Informally, to satisfy the PAC requirement, an algorithm must produce a result that with high probability is close to correct (e.g., incorrect orderings will be most likely to occur between hypotheses with similar expected utilities). The satisfy the EL requirement, on the other hand, a bound must be established on the expected loss of the result, where loss is the difference in utilities between two incorrectly ordered hypothese in an incorrect ranking.

The expected utility of a hypothesis can be estimated by observing its values over a finite set of training examples. However, to satisfy the decision criteria, an algorithm must also be able to reason about the potential difference between the estimated and true utilities of each hypotheses. Let U_i denote the true expected utility of hypothesis i and let $\hat{U}_i$ be the estimated expected utility of hypothesis i. Without loss of generality, let us presume that the proposed ranking of hypotheses is $U_1 > U_2 >, ..., > U_{k-1} > U_k$.

The PAC requirement states that, for some user-specified ϵ, with probability $1 - \delta$:

$$\bigwedge_{i=1}^{k-1} [(U_i + \epsilon) > MAX(U_{i+1}, ..., U_k)] \tag{1}$$

In the context of the PAC criterion, the number ϵ is called the *indifference interval* and δ is the *overall ranking error* or *total error rate.* [3]

The issue of how to allocate the overall ranking error among the many possible pairwise comparisons of hypotheses is discussed in the next section.

Correspondingly, when selecting a hypothesis H_1 to be the best from a set of k hypotheses $H_1, ..., H_k$, let the *selection loss* L be as follows.

$$L(H_1, \{H_1, ..., H_k\}) = MAX(0, MAX(U_2, ..., U_k) - U_1) \tag{2}$$

Then, the *ranking loss* RL of a ranking $H_1, ..., H_k$ would be:

$$RL(H_1, ..., H_k) = \sum_{i=1}^{k-1} L(H_i, \{H_{i+1}, ..., H_k\}) \tag{3}$$

[3] The distinction betwen the true means and the estimated means (for which we use the sample means) is a confusing one. When assessing the validity of a ranking produced by an algorithm, one would use the true means of the distributions (if available, as in test distributions) or the most accurate estimation possible (such as from an edxtremely large sampling of the distribution). However, a ranking algorithm uses the estimated parameters (including sample mean) to estimate the error. For estimation of a single mean the estimate of the mean is normally distributed around the true mean so that this usage is justified. However, we have not proven (and indeed are unsure) whether using the estimate in more complex ranking and selection contexts is guaranteed correct (see later section on the heuristic nture of our algorithms).

A hypothesis ranking algorithm which obeys the expected loss requirement must produce rankings that on average have less ranking loss than the requested expected loss bound. The policy for loss allocation is also discussed in the next section.

As an example, consider ranking the hypotheses with expected utilities: $U_1 = 1.0, U_2 = 0.95, U_3 = 0.86$. The ranking $U_2 > U_1 > U_3$ is a valid PAC ranking for the indifference interval $\epsilon = 0.06$ but not for $\epsilon = 0.01$ and the observed ranking loss is $0.05 + 0 = 0.05$.

However, while the confidence in a pairwise comparison between two hypotheses is well understood to be the complement of the probability of the comparison's result being in error, it is less clear how to define and ensure that a desired confidence is met in the set of comparisons required for a selection or the even more complex set of comparisons required for a ranking. Equation 4 defines the confidence that $U_i + \epsilon > U_j$, when the utilities are normally distributed with unknown and unequal variances.

$$\gamma = \phi\left((\hat{U}_{i-j} + \epsilon)\frac{\sqrt{n}}{\hat{S}_{i-j}}\right) \tag{4}$$

where ϕ represents the cumulative standard normal distribution function, and n, $\hat{U}_{i-j}$, and $\hat{S}_{i-j}$ are the size, sample mean, and sample standard deviation of the blocked differential distribution[4], respectively.

Likewise, computation of the expected loss for asserting an ordering between a pair of hypotheses is well understood, but the estimation of expected loss for an entire ranking is less clear. Equation 5 defines the expected loss for drawing the conclusion $U_i > U_j$, again under the assumption of normality (see Chien et al., 1995, for further details).

$$EL[U_i > U_j] = \frac{\hat{S}_{i-j} e^{-0.5n(\frac{\hat{U}_{i-j}}{\hat{S}_{i-j}})^2}}{\sqrt{2\pi n}} + \frac{\hat{U}_{i-j}}{\sqrt{2\pi}} \int_{-\frac{\hat{U}_{i-j}\sqrt{n}}{\hat{S}_{i-j}}}^{\infty} e^{-0.5z^2} dz \tag{5}$$

In the next two subsections, we describe two interpretations for estimating the likelihood that an overall ranking satisfies the PAC or EL requirements by estimating and combining pairwise PAC errors or EL estimates. Each of these interpretations lends itself directly to an algorithmic implementation as described below.

2.1 Ranking as Recursive Selection

One obvious way to determine a ranking $H_1, ..., H_k$ is to view ranking as recursive selection from the set of remaining candidate hypotheses. In this view, the overall ranking error, as specified by the desired confidence in PAC algorithms and the loss threshold in EL algorithms, is first distributed among $k-1$ *selection errors* which are then further subdivided into *pairwise comparison errors* (Figure 1). Data is then sampled until the estimates of the pairwise comparison error (as dictated by equation 4 or 5) satisfy the bounds set by the algorithm.

[4] Note that in our approach we *block*, or match, examples to further reduce sampling complexity. Blocking makes estimates by using the difference in utility between competing hypotheses on each observed example. Blocking can significantly reduce the variance in the data when the hypotheses are not independent. The differential distribution is formed by taking the differences of the blocked individual samples to form a new distribution. It is trivial to modify the formulas to address the cases in which it is not possible to block data (see Moore & Lee, 1994; Chien et al., 1995, for further details).

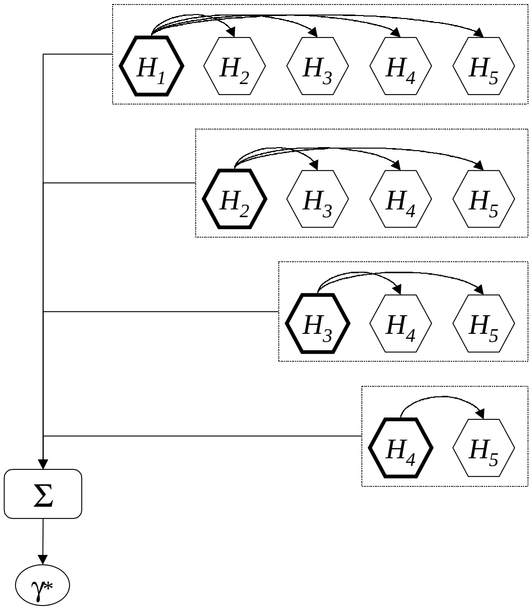

Figure 1: Computing the overall error of a recursive ranking. The per-comparison errors are summed at each level in the recursion, and the overall sum (across all levels) is compared with the specified total error, γ^*.

Thus, another degree of freedom in the design of recursive ranking algorithms is the method by which the overall ranking error is ultimately distributed among individual pairwise comparisons between hypotheses. Two factors influence the way in which we compute error distribution. First, our model of error combination determines how the error allocated for individual comparisons or selections combines into overall ranking error and therefore how many candidates are available for the distribution of error.

Using Bonferroni's inequality, which asserts that the probability of a union of events is no greater than the sum of the probabilities of the individual events[5], one would be inclined to combine the errors additively. However, following a more conservative approach, one can assert that because the predicted "best" hypothesis may change during sampling in the worst case, the conclusion might depend on all possible pairwise comparisons and that the error should be distributed among all $\binom{n}{2}$ pairs of hypotheses.[6]

Second, our policy with respect to allocation of error among the candidate comparisons or selections determines how samples will be distributed. For example, in some contexts, the consequences of early selections far outweigh those of later selections. For these scenarios, we have implemented ranking algorithms that divide overall ranking error unequally in

[5] Note that this is only the simplest of the Bonferonni inequalities, which fall into clean correspondence with the terms of the expansion of the probability of a union of events according to the principle of inclusion and exclusion in a natural way.

[6] For a discussion of this issue, see pp. 18-20 of (Gratch, 1993).

favor of earlier selections.[7] Also, it is possible to divide selection error into pairwise error unequally based on estimates of hypothesis parameters in order to reduce sampling cost (for example, Gratch, Chien, & DeJong, 1994, allocates error rationally).

Within the scope of this paper, we only consider algorithms that: (i) combine pairwise error into selection error additively, (ii) combine selection error into overall ranking error additively, and (iii) allocate error equally at each level.

One disadvantage of recursive selection is that once a hypothesis has been selected, it is removed from the pool of candidate hypotheses. This is an issue in rare cases when, while sampling to increase the confidence of some later selection, the estimate for a hypothesis' mean changes enough that some previously selected hypothesis no longer dominates it. However, it remains that the original hypotheses were shown to dominate the others *with a specified level of certainty*, γ^*.

These assumptions result in the following formulations (where $\delta(U_1 \rhd_\epsilon \{U_2, ..., U_k\})$ is used to denote the error due to the action of selecting hypothesis 1 under Equation 1 from the set $\{H_1, ..., H_k\}$ and $\delta(U_1 \rhd \{U_2, ..., U_k\})$ denotes the error due to selection loss in situations where Equation 2 applies):

$$\begin{aligned} \delta_{rec}(U_1 > U_2 > ... > U_k) = \quad & \delta_{rec}(U_2 > U_3 > ... > U_k) \\ & + \delta(U_1 \rhd_\epsilon \{U_2, ..., U_k\}) \end{aligned} \tag{6}$$

where $\delta_{rec}(U_k) = 0$ (the base case for the recursion) and the selection error is as defined in (Chien et al., 1995):

$$\delta(U_1 \rhd_\epsilon \{U_2, ..., U_k\}) = \sum_{i=2}^{k} \delta_{1,i} \tag{7}$$

using Equation 4 to compute pairwise confidence.

Algorithmically, we implement this with the following pseudo-code:

```
ensure there are $n_0$ samples per hypothesis
distribute the error to individual selections
while (stopping criteria has not been met)
    take more samples
    if (means are ordered differently than ranking)
        restart the algorithm
```

An analogous recursive selection algorithm based on expected loss is defined as follows

$$\begin{aligned} EL_{rec}(U_1 > U_2 > ... > U_k) = \quad & EL_{rec}(U_2 > U_3 > ... > U_k) \\ & + EL(U_1 \rhd \{U_2, ..., U_k\}) \end{aligned} \tag{8}$$

where $EL_{rec}(U_k) = 0$ and the selection EL is as defined in (Chien et al., 1995):

$$EL(U_1 \rhd \{U_2, ..., U_k\}) = \sum_{i=2}^{k} EL(U_1, U_i) \tag{9}$$

[7] Space constraints preclude their description here.

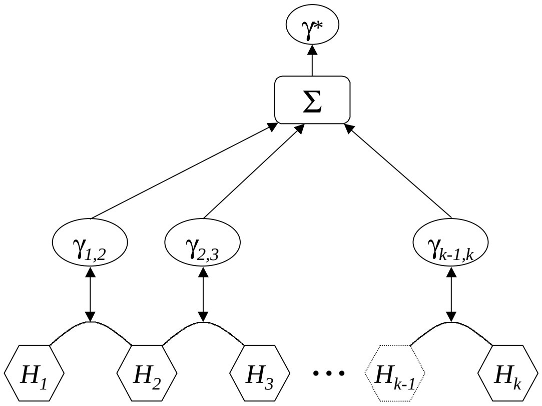

Figure 2: Computing the overall error in an adjacent ranking. Per-comparison errors between neighboring hypotheses in the proposed ranking are summed and compared with the required total error, γ^*.

2.2 Ranking by Adjacency Comparison

Another interpretation of ranking confidence (or loss) is that only adjacent elements in the ranking need be compared. In this case, the overall ranking error is divided directly into $k-1$ pairwise comparison errors (Figure 2). This leads to the following confidence equation for the PAC criteria:

$$\delta_{adj}(U_1 > U_2 > ... > U_k) = \sum_{i=1}^{k-1} \delta_{i,i+1} \tag{10}$$

And the following equation for the EL criteria.

$$EL_{adj}(U_1 > U_2 > ... > U_k) = \sum_{i=1}^{k-1} EL(U_i, U_{i+1}) \tag{11}$$

Because ranking by comparison of adjacent hypotheses does not establish dominance or loss bounds between non-adjacent hypotheses (where the hypotheses are ordered by observed mean utility), it has the advantage of requiring fewer comparisons than recursive selection (and thus may require fewer samples than recursive selection). However, for the same reason, adjacency algorithms may be less likely than the recursive selection algorithms to bound the probability of a correct ranking (or average loss) correctly. In the case of the PAC algorithms, this is because ϵ-dominance is not necessarily transitive. In the case of the EL algorithms, it is because expected loss is not necessarily additive when considering two hypothesis comparisons sharing a common hypothesis.[8]

[8] An example where ranking loss between non-adjacent hypotheses exceeds the desired loss bound for the ranking, even though the sum of the adjacent losses does not, occurs when the blocked differential distribution induced by two non-adjacent hypotheses has high variance relative to an hypothesis adjacent

2.3 The Heuristic Nature of the Algorithms

Both the recusrsive selection and adjacency algorithms are heuristic in the sense that they are not proven to statistically meet the specified decision criteria (i.e., for the PAC criteria select a ranking that satisfies equation (1) with probability $1 - \delta$ and similarly for the EL criteria average a ranking loss specified by equation (3) less than the requested bound. Indeed, several aspects of these algorithms make it extremely difficult to prove that they would (probabilistically) achieve the corresponding decision criteria. These aspects include:

- *Sharing of samples*: In order to have n_1 samples for a differential distribution (i.e. blocking) for H_1 and H_2, it takes n_1 samples of H_1 and n_1 samples on the the same problems for H_2. Our algorithms further reduce the sampling cost by reusing these samples in differential distributions comparing H_1 to other hypotheses and H_2 to other hypotheses. This makes the errors derived from these samples not independent. Hence we have traded accuracy and ease of analysis of the algorithms for heuristic efficiency. Particularly in the recursive selection approach, samples for the lowest ranking hypothesis would have been used in $k - 1$ differential comparisons.

- *Heuristic error combination*: Both the recursive selection and adjacency error combination models are heuristic means of combining pairwise errors. This is because the pairwise errors are not independent (see above). Empirically we have observed that the pairwise errors tend to be overestimated but the error combination function tends to under-combine. Overall empirically the combined error estimates tend to be reasonably accurate, as the remaining sections show.

- *Ignorance of lead switches and multiple comparison paths*: During the sampling process, the ordering of the hypotheses may change (e.g., the ordering of sample means may change). This means that implicitly, the decision depended on an additional pairwise comparison that may not be reflected in the final set of comparisons contributing a pairwise error. This complexity could be avoided by fixing the order of the hypotheses after n_0 samples. However, this would require more samples as is would involve showing ϵ-dominance of a hypothesis over a higher sample mean hypothesis (indeed, it may never converge). We choose to ignore this complexity and base the combined error used in the stopping condition on the final ordering.

- *Use on non-normal distributions*: In many of the applications described in the remainder of this article, the real-world data is distributed in a manner not very simlar to normal distributions (we further investigate this issue later in the article). The algorithms we describe are heuristic in that they presume that the data is normally distributed even though this is not the case.

to both (i.e., currently ranked between them). The variance of the differential distribution makes its maximum contribution when the sample set is small, so, e.g., with $\mu_{1-2} = 2$, $\sigma_{1-2} = 2$, $n_{1-2} = 2$, $\mu_{2-3} = 2$, $\sigma_{2-3} = 2$, and $n_{2-3} = 2$, there exists a configuration for which $\mu_{1-3} = 4$, $\sigma_{1-3} = 8$. The expected losses are $EL(H_1, H_2) = 2.05$, $EL(H_2, H_3) = 2.05$, but $EL(H_1, H_3) = 4.80 > 4.10$.

2.4 Other Relevant Approaches

Most standard statistical ranking/selection approaches make strong assumptions about the form of the problem (e.g., the variances associated with underlying utility distribution of the hypotheses might be assumed known and equal). Among these, the method of Turnbull and Weiss (Turnbull & Weiss, 1984) is most comparable to our PAC-based approach.[9]

Turnbull and Weiss' algorithm is a sequential interval-based procedure for selecting the member of a population with the largest mean. They treat hypotheses as normally distributed random variables of unknown mean that have unknown and possibly unequal variance. Their algorithm also carries the additional stipulation that the hypotheses be independent. The procedure consists of taking an initial sample of n_0 observations on each of the hypotheses and then taking samples sequentially according to their stopping criteria. When the stopping criteria has been satisfied, the hypothesis with the highest sample mean is chosen. The stopping criteria is that the inequality $\frac{S_i^2}{n_i} \leq \frac{1}{n^*}$ is satisfied, where S_i and n_i are the sample mean and the number of samples of the ith hypothesis and n^* is chosen according to the indifference interval ϵ and the confidence level γ^*. In particular, $n^* = \frac{d^2}{\epsilon^2}$ and d is chosen to satisfy $\int_{-\infty}^{\infty} (F(y+d))^{k-1} f(y) dy = \gamma^*$ where $F(y)$ and $f(y)$ are the cumulative distribution function and probability density function of the standard normal distribution.

While it is still reasonable to use this approach when the candidate hypotheses are not independent, excessive statistical error or unnecessarily large training set sizes may result. In the case that the hypotheses are truly independent, Turnbull and Weiss' technique should be able to exploit this knowledge and outperform our methods which do not adopt this assumption.

3. Empirical Performance Evaluation

We now turn to empirical evaluation of the hypothesis ranking techniques on both synthetic and real-world datasets. This evaluation serves three purposes. First, it demonstrates that the techniques perform as predicted (in terms of bounding the probability of incorrect selection or expected loss). Second, it validates the performance of the techniques as compared to standard algorithms from the statistical literature. Third, the evaluation demonstrates the robustness of the new approaches to real-world hypothesis ranking problems.

An experimental trial consists of solving a hypothesis ranking problem with a given technique and a given set of problem and control parameters. We measure performance by (1) how well the algorithms satisfy their respective criteria; and (2) the number of samples taken or, alternatively, the cost (in seconds) of executing the algorithm. Since the performance of these statistical algorithms on any single trial provides little information about its overall behavior, each trial is repeated multiple times and the results are averaged across trials. Synthetic experimental trials were repeated 500 times, while trials on the real-world data were repeated 100 times. Because the PAC and expected loss criteria are not directly comparable, the approaches are analyzed separately.

[9] PAC-based approaches have been investigated extensively in the statistical ranking and selection literature under the topic of *confidence interval based* algorithms (see Haseeb, 1985, for a review of the recent literature).

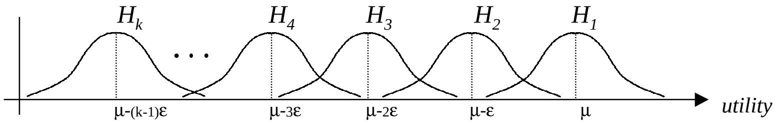

Figure 3: The stepped means hypothesis configuration.

3.1 Evaluation on Synthetic Datasets

Evaluation on synthetic data is used to show that: (1) the techniques correctly bound probability of incorrect ranking and expected loss as predicted when the underlying assumptions are valid even when the underlying utility distributions are inherently hard to rank [10], and (2) that the PAC techniques compare favorably to the algorithm of Turnbull and Weiss in a wide variety of circumstances.

For the synthetic datasets, the utility distributions of the hypotheses were modeled as random variables defined on some underlying parameterized distribution. Thus, characterizing a ranking problem consists of choosing some number of hypotheses to rank and then assigning values for parameters representing each utility distributions for these hypotheses. In our case, we model the utilities as independent normal random variables with some mean and standard deviation. Thus, if we let k be the number of hypotheses, then each hypothesis ranking problem is described by the $2k$ parameters specifying the expected utility and utility standard deviation for each hypothesis. In general, while several more parameters may be required to characterize a ranking problem fully[11], the number of hypotheses and the choices for the parameters of the utility distributions underlying these hypotheses characterize the overall difficulty of the ranking problem.

The statistical ranking and selection community uses a standard family of selection problems with known difficulty to analyze the performance of hypothesis selection strategies. The method, called the least favorable configuration (LFC) of the population means is that assignment of the parameters to distributions which is most likely to cause a technique to choose a wrong hypothesis and thus provides the most severe test of the technique's abilities. Under this configuration, all utilities are independent normally distributed variables of equal variance. $k-1$ of the hypotheses have utilities with equal expectation, μ, and the remaining hypothesis has expected utility $\mu + \epsilon$.

Because we are interested in hypothesis ranking problems rather than selection problems, we use a generalization of the LFC that we call stepped means. In this configuration, one of the hypotheses is assigned expected utility μ and successive hypotheses are assigned expected utility $\mu - i\epsilon$ for i from $1, ..., k-1$ (Figure 3).

In general, problems based on the least favorable configuration become more difficult (i.e., require more samples) when the number of hypotheses k increases, the common utility variance σ^2 increases, or the difference in the means of the utility distributions decreases. In the standard methodology, a technique is evaluated by its ability to achieve a confidence of

[10] Configurations that contain hypotheses with high variance relative to the separation between their means are more difficult to rank.

[11] For instance, when samples are allocated rationally in (Chien et al., 1995), it becomes necessary to assign parameters to a cost distribution as well, or if only a few of the candidate hypotheses were to be ranked, the number of hypotheses to rank would be another problem parameter.

correct selection γ^* using several settings for k and $\frac{\sigma}{\epsilon}$. This last ratio combines σ and ϵ into a single quantity which, as it increases, makes the problem more difficult. This methodology extends to stepped means directly.

The hypothesis ranking strategies themselves have *algorithm control parameters* that govern how they attack a problem. The PAC techniques have three control parameters: an initial sample size n_0, a desired confidence of correct ranking γ^* and an indifference setting ϵ[12]. The expected loss techniques have two control parameters: an initial sample size n_0 and a loss threshold H^*.

The observed number of samples required and achieved accuracy of the PAC techniques on the stepped means configuration are shown in Table 3.1. The results indicate that all systems are roughly comparable in the number of examples required to choose a hypotheses. As expected, the number of examples increases with k, γ^*, and $\frac{\sigma}{\epsilon}$. The PAC_{adj} algorithm required the least number of samples but was inconsistent in meeting the desired accuracy bound (as indicated by its failure to meet the prescribed error bound in several cases). It is interesting that the Turnbull and Weiss method did not significantly outperform the PAC techniques despite the fact that the algorithm assumes that the hypotheses are independent (as is the case in the stepped means configuration), while the PAC approaches do not make this assumption. In this comparison, the principal performance metric is the number of samples required to achieve the requested ranking, both methods were effective at achieving the requested accuracy.

In the expected loss experiments, we ran the expected loss hypothesis ranking algorithms on the same stepped means configurations described above with a range of expected loss bounds. Table 3.1 shows the results of this experiment, displaying the number of samples required to produce a ranking and the average observed loss for each configuration. These results show that the EL_{rec} algorithm correctly bounded the loss and that the EL_{adj} algorithm required less samples than the EL_{rec} algorithm, but did not correctly bound the expected loss (since the observed loss was greater than the loss bound H^*.[13]

3.2 Evaluation on Real Datasets

The test of real-world applicability is based on data drawn from several datasets relating to spacecraft design and the processing of science data gathered in the context of planetary exploration. The first two datasets we investigate relate to spacecraft design optimization problems in which the hypotheses we wish to rank are candidate solutions to the design problem. The third and last dataset we examine involves ranking various lossless image compression approaches based on their performance on a large set of terrestrial images collected by the spacecraft Galileo. Cost of evaluation is given in seconds for all empirical data

[12] Note that in our formulation of the stepped means test for the PAC approaches, ϵ is both the difference in the expected mean of successive hypotheses *and* the indifference interval of the algorithm. Thus, ϵ plays the roles of both problem parameter and control parameter here.

[13] One confusing point is that for identical hypothesis and ranking algorithm settings, one can observe a lower loss when ranking a larger number of hypotheses. This is because the algorithm first divides the loss over the number of pirwise comparisons. Thus, for the same overall error (or expected loss bound), with more hypotheses, the pairwise expected error (or loss) will be smaller if there are more hypotheses. The ranking loss is defined previously. Thus, it is possible for the observed loss to increase or decrease compared to the same settings with fewer hypotheses.

k	γ^*	$\frac{\sigma}{\epsilon}$	TURNBULL	PAC_{rec}	PAC_{adj}
3	0.75	2	62 (0.88)	55 (0.95)	38 (0.78)
3	0.75	3	117 (0.89)	101 (0.86)	49 (0.80)
3	0.90	2	97 (0.96)	86 (0.94)	58 (0.92)
3	0.90	3	183 (0.99)	152 (0.96)	96 (0.89)
3	0.95	2	130 (0.97)	122 (0.97)	89 (0.97)
3	0.95	3	231 (0.96)	204 (0.95)	146 (0.94)
5	0.75	2	177 (0.83)	165 (0.95)	105 (0.87)
5	0.75	3	321 (0.95)	314 (0.93)	161 (0.75)
5	0.90	2	245 (0.98)	245 (0.97)	163 (0.91)
5	0.90	3	445 (0.98)	409 (0.91)	290 (0.92)
5	0.95	2	299 (0.98)	294 (0.98)	216 (1.00)
5	0.95	3	541 (0.98)	538 (0.98)	377 (0.92)
10	0.75	2	558 (0.92)	624 (0.91)	345 (0.85)
10	0.75	3	1,015 (0.94)	1,042 (0.95)	635 (0.83)
10	0.90	2	700 (0.97)	742 (0.96)	523 (0.91)
10	0.90	3	1,254 (0.97)	1,359 (0.97)	883 (0.90)
10	0.95	2	821 (1.00)	877 (0.97)	661 (0.94)
10	0.95	3	1,462 (0.99)	1,569 (0.98)	1,164 (0.93)

Table 1: Estimated expected total number of observations by PAC algorithms in the stepped means configuration. Achieved probability of correct ranking is shown in parenthesis.

Parameters			EL_{rec}		EL_{adj}	
k	ϵ	H^*	Samples	Loss	Samples	Loss
3	2	1.0	96	0.6	43	1.2
3	2	0.75	102	0.5	56	1.0
3	2	0.5	139	0.2	73	0.6
3	2	0.25	235	0.1	139	0.4
5	2	1.0	320	0.7	140	1.3
5	2	0.75	343	0.4	169	1.2
5	2	0.5	464	0.4	247	0.7
5	2	0.25	575	0.2	350	0.5
10	2	1.0	1,136	0.5	572	1.4
10	2	0.75	1,325	0.5	668	1.1
10	2	0.5	1,533	0.3	872	0.7
10	2	0.25	1,856	0.1	1,153	0.4

Table 2: Estimated expected total number of observations of EL algorithms in stepped means configuration. Observed average loss of produced rankings.

because, unlike the synthetic problems, the cost of sampling a hypothesis is not constant in these domains. Table 3 gives a summary of the three ranking problems we considered.

Dataset	fixed parameters	random variables	optimization criteria
DS-2 Penetrator	penetrator diameter penetrator length	impact orientation impact velocity soil density	maximize penetration probability maximize penetration depth
DS-2 Aeroshell	fore body overlap nose cone angle bluntness ratio fillet radius outer diameter tail geometry	stagnation pressure coef.	minimize weight achieve target entry velocity
Lossless Image Comp.	compression method	randomly selected test image	maximize compression ratio

Table 3: Description of datasets used for algorithm evaluation.

3.2.1 DS-2 Penetrator

The goal of the New Millennium Deep Space Two (DS-2) mission is to deliver a pair of microprobes to the planet Mars for scientific study of the Martian soil. The probes will be released from orbit, travel through the Martian atmosphere, and embed themselves in the soil near the southern polar ice cap. The primary science objectives for the mission are (Balacuit., 1997):

- to determine if ice is present below the surface of Mars,
- to measure the local atmospheric pressure,
- and to characterize the thermal properties of the Martian subsurface soil.

The goal of this spacecraft design problem is to determine a good set of physical dimensions for the penetrator—a small, robust probe designed to impact the surface at extremely high velocity and to operate in the extreme cold. Specifically, we use design and simulation data from the DS-2 mission penetrator design.

For our casting of the design problem, we hold the shape of the penetrator constant and generate design candidates based on different values for the variables of penetrator diameter and length. For a specific design a sample is taken by acquiring impact orientation, impact velocity, and soil density from a parameterized multivariate distribution and then calling a complex physical simulation to determine if and to what depth the penetrator bored into the Martian surface. The goal of the penetrator design problem is to determine the physical dimensions of the penetrator that maximize the probability of penetration, and in cases of penetration, maximize penetration depth.

Tables 4 and 5 show the results of applying the PAC-based, Turnbull, and expected loss algorithms to a ranking problem in which the system is requested to rank 10 penetrator designs.[14] In this problem the utility function is the depth of penetration of the penetrator,

[14] "True" expected utility values were computed by performing 20,000 samples and using the sample mean for this large sample as ground truth. These expected utilities were then used to compute PAC ϵ-validity of rankings and observed loss using the provided definitions.

with those cases in which the penetrator does not penetrate being assigned zero utility. As shown in Table 4, both PAC algorithms significantly outperformed the Turnbull algorithm, which is to be expected because the hypotheses are somewhat correlated (via impact orientations and soil densities). Table 5 shows that the EL_{rec} expected loss algorithm effectively bounded actual loss but the EL_{adj} algorithm was inconsistent.

k	γ^*	$\frac{\sigma}{\epsilon}$	TURNBULL	PAC_{rec}	PAC_{adj}
10	0.75	2	534 (0.96)	144 (1.00)	92 (0.98)
10	0.90	2	667 (0.98)	160 (1.00)	98 (1.00)
10	0.95	2	793 (0.99)	177 (1.00)	103 (0.99)

Table 4: Estimated expected total number of observations to rank DS-2 spacecraft designs. Achieved probability of correct ranking is shown in parenthesis.

Parameters		EL_{rec}		EL_{adj}	
k	H^*	Samples	Loss	Samples	Loss
10	0.10	152	0.05	77	0.14
10	0.05	200	0.03	90	0.06
10	0.02	378	0.03	139	0.03

Table 5: Estimated expected total number of observations and expected loss of an incorrect ranking of DS-2 penetrator designs.

3.2.2 DS-2 Aeroshell Design Ranking

The objective of this problem is to design an aeroshell for the soil penetrator described in the previous section that gives the appropriate entry velocity with minimum weight. Design candidates are defined by six continuous variables that represent various geometric quantities: the extent to which the fore body overlaps the aftbody, nose cone angle, bluntness ratio, fillet radius, outer diameter, and the tail geometry. Candidate designs (hypotheses) are evaluated by running a simple physical simulation of the aeroshell's behavior. Such a sample is taken by running the simulation with the fixed design variables of the hypothesis and a value for the stagnation pressure coefficient taken from a normal distribution. The simulation computes values for the achieved entry velocity and the mass of the aeroshell; then the weighted sum of the reciprocals of these values is maximized.

We give the results of ranking three, five, and ten hypotheses using the Turnbull, PAC, and expected loss algorithms in Tables 6 and 7.[15]

As in the previous experiment, the PAC-based algorithms outperformed the Turnbull algorithm in all cases. While the PAC_{adj} algorithm represents a significant increase in

[15] Again, deep sampling (500 samples) was performed to obtain the "correct" ranking, against which these algorithms are compared.

performance here, we note that it did not achieve the desired level of confidence in all cases; both the Turnbull and PAC_{rec} algorithms did achieve the required confidence.

k	γ^*	$\frac{\sigma}{\epsilon}$	TURNBULL	PAC_{rec}	PAC_{adj}
3	0.75	2	8.9 (1.00)	8.4 (1.00)	3.5 (1.00)
3	0.75	3	22.9 (1.00)	11.3 (1.00)	3.8 (1.00)
3	0.90	2	17.1 (1.00)	14.0 (1.00)	7.1 (1.00)
3	0.90	3	38.2 (1.00)	18.6 (1.00)	7.2 (1.00)
3	0.95	2	22.6 (1.00)	21.6 (1.00)	7.1 (1.00)
3	0.95	3	52.0 (1.00)	32.1 (1.00)	7.3 (1.00)
5	0.75	2	29.1 (0.92)	20.1 (0.94)	11.8 (0.91)
5	0.75	3	69.0 (1.00)	33.9 (0.96)	11.7 (0.91)
5	0.90	2	42.4 (0.99)	30.0 (0.93)	11.7 (0.91)
5	0.90	3	94.7 (0.99)	54.8 (0.96)	11.8 (0.84)
5	0.95	2	51.9 (0.98)	43.6 (1.00)	11.7 (0.91)
5	0.95	3	117.9 (0.99)	81.5 (0.99)	11.5 (0.92)
10	0.75	2	84.0 (0.99)	42.0 (0.94)	22.1 (0.92)
10	0.75	3	196.6 (1.00)	57.9 (0.96)	22.1 (0.90)
10	0.90	2	112.0 (0.98)	53.8 (0.98)	22.6 (0.89)
10	0.90	3	252.9 (0.99)	85.5 (1.00)	21.6 (0.91)
10	0.95	2	129.1 (1.00)	61.3 (0.97)	20.6 (0.90)
10	0.95	3	315.7 (1.00)	125.7 (1.00)	20.4 (0.92)

Table 6: Estimated expected cost (in seconds) to rank aeroshell designs. Achieved probability of correct ranking is shown in parenthesis.

Parameters		EL_{rec}		EL_{adj}	
k	H^*	Execution Cost	Loss	Execution Cost	Loss
3	20	9.5	4.3	7.9	3.4
3	30	7.6	3.4	7.3	3.7
3	40	7.3	4.1	6.9	2.7
5	20	21.7	7.0	7.2	8.6
5	30	18.1	12.0	6.4	12.4
5	40	15.0	9.3	10.5	8.5
10	20	55.3	9.7	18.3	7.9
10	30	42.6	8.9	14.2	9.8
10	40	38.2	10.4	13.1	9.6

Table 7: Estimated expected cost (in seconds) and expected loss of an incorrect ranking of DS-2 aeroshell designs.

3.2.3 Lossless Image Compression on Galileo Image Data

This problem utilizes a large set of raw image data acquired by the Galileo spacecraft. Each of the images is 256 by 256 in size and is made up of greyscale pixels ranging from 0 to 255 in intensity. The goal is to select the lossless compression method[16] that performs best on this class of images. The performance of an image compression algorithm on a particular image could be measured in a number of ways. For example, execution time, compression ratio, and image quality (in the case where lossy compression methods are being considered) could define algorithm performance. In our tests we chose to consider only the compression ratio achieved by a given compression method as our utility function. To sample each method (hypothesis), an image is randomly selected, the method is applied to that image, and the achieved compression ratio is recorded.

Given below (Tables 8 and 9) are the results of ranking three, five, and seven hypotheses using the Turnbull, PAC, and expected loss algorithms. Ranking correctness was determined by comparison to a "correct" ranking established by sampling each compression method on a set of 1500 distinct images.

We again note the substantial performance improvement the PAC-based algorithms have over the Turnbull algorithm. Although both the Turnbull algorithm and the PAC algorithms (Table 8) achieved the desired confidence level, the adjacent version of the EL algorithm (Table 9) failed to bound the loss to the specified level in over half the cases.

It is interesting to consider the results presented in this section in light of the fact that each of the statistical techniques being used makes some form of normality assumption. In fact, all three of the problem domains we investigate have some number of hypotheses whose utility functions are *not* normally distributed. From past experience it is known that utility functions in the DS-2 Penetrator domain (Section 3.2.1) are highly non-normal; Figure 4 illustrates the difference between data that is normally distributed and data that is not.

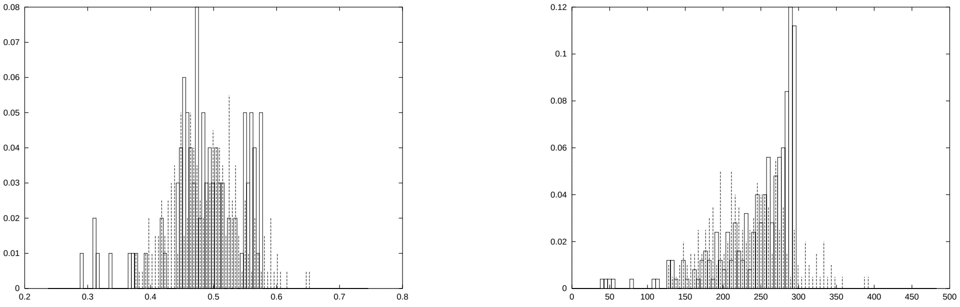

Figure 4: A comparison of (a) data that is normally distributed with high likelihood and (b) data that is very likely *not* normally distributed. In each case, the histogram of experimental data is shown in solid boxes; data drawn from a normal distribution with the same mean and standard deviation is shown with dashed lines.

To determine the extent to which the utilities of hypotheses in the remaining two domains are normally distributed we applied the Kolmogorov-Smirnov test (see Appendix A

[16] The seven compression methods we considered were: CALIC, lossless JPEG, GIF, TIFF, pack, gzip, and compress.

k	γ^*	$\frac{\sigma}{\epsilon}$	TURNBULL	PAC_{rec}	PAC_{adj}
3	0.90	2	62.8 (1.00)	30.1 (1.00)	14.8 (1.00)
3	0.90	3	150.8 (1.00)	30.5 (1.00)	14.8 (1.00)
3	0.95	2	84.6 (1.00)	28.6 (1.00)	15.0 (1.00)
3	0.95	3	206.8 (1.00)	29.0 (1.00)	20.5 (1.00)
3	0.99	2	142.0 (1.00)	30.1 (1.00)	23.3 (1.00)
3	0.99	3	359.4 (1.00)	30.6 (1.00)	23.2 (1.00)
5	0.90	2	134.7 (1.00)	39.5 (1.00)	29.9 (1.00)
5	0.90	3	329.9 (1.00)	39.9 (1.00)	30.0 (1.00)
5	0.95	2	176.1 (1.00)	39.3 (1.00)	29.8 (1.00)
5	0.95	3	399.8 (1.00)	39.3 (1.00)	29.6 (1.00)
5	0.99	2	249.3 (1.00)	39.2 (1.00)	29.9 (1.00)
5	0.99	3	598.1 (1.00)	39.2 (1.00)	30.7 (1.00)
7	0.90	2	210.8 (1.00)	35.6 (1.00)	37.2 (1.00)
7	0.90	3	499.3 (1.00)	35.7 (1.00)	34.5 (1.00)
7	0.95	2	250.3 (1.00)	37.4 (1.00)	35.6 (1.00)
7	0.95	3	608.7 (1.00)	36.0 (1.00)	35.0 (1.00)
7	0.99	2	339.6 (1.00)	36.5 (1.00)	34.5 (1.00)
7	0.99	3	813.7 (1.00)	37.2 (1.00)	35.3 (1.00)

Table 8: Estimated expected cost (in seconds) to rank lossless image compression approaches on Galileo image data. Achieved probability of correct ranking is shown in parenthesis.

Parameters		EL_{rec}		EL_{adj}	
k	H^*	Execution Cost	Loss	Execution Cost	Loss
3	10	31.7	0.0	24.9	0.0
3	5	32.5	0.0	24.9	0.0
3	1	33.7	0.0	24.9	0.0
5	10	80.6	0.0	32.7	17.4
5	5	83.5	0.0	33.7	69.4
5	1	101.0	0.0	32.3	49.4
7	10	99.5	0.0	42.3	17.4
7	5	105.7	0.0	33.3	34.7
7	1	119.8	0.0	30.4	86.8

Table 9: Estimated expected cost (in seconds) and expected loss of an incorrect ranking of DS-2 penetrator designs.

for details). The test determined that none of the ten hypotheses from the DS-2 Aeroshell domain (Section 3.2.2) had normally distributed utility. Additionally, only two of the seven hypotheses from the image compression domain (Section 3.2.3) were shown to have greater than 90% likelihood of having normally distributed utility functions[17]. For these reasons, evaluating the ranking strategies on these datasets provides a particularly strong test of the applicability of the techniques.

We draw the reader's attention to the particularly large disparity in performance between the Turnbull algorithm and the PAC-based algorithms in the image compression domain, especially apparent when the number of hypotheses, and the confidence level, are high. Additionally, this problem domain has two hypotheses with normally distributed utility and five that are non-normal. These observations suggest that the PAC-based algorithms perform better (in relative terms) when faced with a domain that violates the assumption of normality.

4. Discussion and Conclusions

There are a number of areas of related work. First, there has been considerable analysis of hypothesis selection problems. Selection problems have been formalized using a Bayesian framework (Moore & Lee, 1994; Rivest & Sloan, 1988) that does not require an initial sample, but uses a rigorous encoding of prior knowledge. Howard (Howard, 1970) also details a Bayesian framework for analyzing learning cost for selection problems. If one uses a hypothesis selection framework for ranking, allocation of pairwise errors can be performed rationally (Gratch et al., 1994). Reinforcement learning work (Kaelbling, 1993) with immediate feedback can also be viewed as a hypothesis selection problem.

The framework presented invites future work in a number of directions. Currently, the stopping criteria used are relaxations of the ranking requirement. Another approach that could be used is to bound the resources available for ranking. Limiting the number of samples where sample cost is high and limiting the time of computation (so that we have an anytime algorithm) are two straightforward application areas.

Another area for future work is discovery of composite strategies or hypotheses. Thus far we have examined ranking (and in other articles, selection) of a hypothesis with highest expected value over an entire distribution. For example, learning a scheduling control strategy that will do well over a distribution of problems. However, it is likely that for most distributions of problems, there exists a composite strategy which would outperform any single strategy. For example, a single strategy might be to apply method A to solve a problem. A composite strategy would be, test the problem for feature X, if X true apply method A, else apply method B. These composite strategies correspond to algorithm portfolios as named in Operations Research. Indeed the results of applying methods could also be viewed as strategies. One might have the composite strategy of trying method A for 10 CPU seconds, then if that fails trying method B. Of course, in all these composition and portfolio approaches, the difficulty isefficiently proposing and evaluating plausible compositions. For even a small set of base strategies the number of copositions is enormous.

[17] For reference, the data in Figure 4 (a) was normally distributed with 97.5% likelihood, according to the Kolmogorov-Smirnov test.

In summary, this paper has described the hypothesis ranking problem, an extension to the hypothesis selection problem. We defined the application of two decision criteria, *probably approximately correct* and *expected loss*, to this problem. We then defined two families of algorithms, recursive selection and adjacency, for solution of hypothesis ranking problems. Finally, we demonstrated the effectiveness of these algorithms on both synthetic and real-world datasets, documenting improved performance over existing statistical approaches.

Acknowledgments

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

Appendix A. Applying the K-S Test to Real Datasets

The Kolmogorov-Smirnov Test is a statistical means of accepting, with a certain level of confidence, the hypothesis that some sampleset fits a parametric distribution with a given set of parameters. The method compares the CDF generated by the empirical distribution to that of the corresponding parametric distribution (i.e., with estimated parameters). The K-S test gives a confidence based on the maximum, D, of the discrepancies between these two CDFs:

$$D = max|F_1(x) - F_2(x)|$$

For our purposes we wish to determine, for each hypothesis in a given domain, whether the values of the utility function are normally distributed or not. In each case, half of the utility samples taken are used to compute the mean and standard deviation of the normal; the remaining half are used to compute the CDF.

A.1 DS-2 Penetrator

20000 samples taken.

design number	$max\|F_1(x) - F_2(x)\|$	normally distributed?
1	0.1415	$\ll$ 90% likely
2	0.1202	$\ll$ 90% likely
3	0.1020	$\ll$ 90% likely
4	0.1261	$\ll$ 90% likely
5	0.1207	$\ll$ 90% likely
6	0.1261	$\ll$ 90% likely
7	0.1020	$\ll$ 90% likely
8	0.1493	$\ll$ 90% likely
9	0.1461	$\ll$ 90% likely
10	0.1261	$\ll$ 90% likely

A.2 DS-2 Aeroshell Design Ranking

500 samples taken.

design number	$max\|F_1(x) - F_2(x)\|$	normally distributed?
1	0.08	$<$ 90% likely
2	0.08	$<$ 90% likely
3	0.08	$<$ 90% likely
4	0.08	$<$ 90% likely
5	0.08	$<$ 90% likely
6	0.08	$<$ 90% likely
7	0.08	$<$ 90% likely
8	0.08	$<$ 90% likely
9	0.08	$<$ 90% likely
10	0.08	$<$ 90% likely

A.3 Lossless Image Compression on Galileo Image Data

200 samples taken.

compression method	$max\|F_1(x) - F_2(x)\|$	normally distributed?
gif	0.10	90% likely
compress	0.14	$<$ 90% likely
calic	0.19	$\ll$ 90% likely
gzip	0.09	97.5% likely
jpegls	0.18	$\ll$ 90% likely
pack	0.12	$<$ 90% likely
tiff	0.11	$<$ 90% likely

References

Balacuit., C. P. (1997). Deep Space 2 – Mars Microprobe Home Page (mission objectives statement). Tech. rep. http://nmp.jpl.nasa.gov/ds2, NASA/JPL.

Chien, S. A., Gratch, J. M., & Burl, M. C. (1995). On the Efficient Allocation of Resources for Hypothesis Evaluation: A Statistical Approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *17*(7), 652–665.

Chien, S. A., Stechert, A. D., & Mutz, D. H. (1998). Efficient Heuristic Ranking of Hypotheses. In *Advances in Neural Information Processing Systems 10 (Jordan, Kearns, and Solla eds.)*, pp. 444–450 Denver, Colorado. NIPS.

Ginsberg, M., & Harvey, W. (1992). Iterative Broadening. *Artificial Intelligence Journal*, *55*, 367–383.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

Gratch, J. (1992). COMPOSER: A Probabilistic Solution to the Utility Problem in Speed-up Learning. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 235–240 San Jose, CA. AAAI.

Gratch, J. (1993). COMPOSER: A Decision-theoretic Approach to Adaptive Problem Solving. Tech. rep. UIUCDCS-R-93-1806, Department of Computer Science, University of Illinois.

Gratch, J., Chien, S., & DeJong, G. (1994). Improving Learning Performance Through Rational Resource Allocation. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 576–582 Seattle, WA. AAAI.

Greiner, R., & Jurisica, I. (1992). A Statistical Approach to Solving the EBL Utility Problem. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 241–248 San Jose, CA. AAAI.

Haseeb, R. M. (1985). *Modern Statistical Selection.* American Sciences Press, Columbus, OH.

Howard, R. A. (1970). Decision Analysis: Perspectives on Inference, Decision, and Experimentation. *Proceedings of the IEEE*, *58*(5), 823–834.

Kaelbling, L. P. (1993). *Learning in Embedded Systems.* MIT Press, Cambridge, MA.

Minton, S. (1988). *Learning Search Control Knowledge: An Explanation-Based Approach.* Kluwer Academic Publishers, Norwell, MA.

Moore, A. W., & Lee, M. S. (1994). Efficient Algorithms for Minimizing Cross-Validation Error. In *Proceedings of the International Conference on Machine Learning* New Brunswick, MA.

Musick, R., Catlett, J., & Russell, S. (1993). Decision Theoretic Subsampling for Induction on Large Databases. In *Proceedings of the International Conference on Machine Learning*, pp. 212–219 Amherst, MA.

Rivest, R. L., & Sloan, R. (1988). A New Model for Inductive Inference. In *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge.*

Russell, S., & Wefald, E. (1992). *Do the Right Thing: Studies in Limited Rationality.* MIT Press, Cambridge, MA.

Tadepalli, P. (1992). A theory of unsupervised speedup learning. In *Proc. of the Tenth National Conference on Artificial Intelligence*, pp. 229–234 San Jose, CA. AAAI.

Turnbull, B. W., & Weiss, L. I. (1984). A Class of Sequential Procedures for k-sample Problems Concerning Normal Means with Unknown Equal Variances. In Santner, T. J., & Tamhane, A. C. (Eds.), *Design of Experiments: Ranking and Selection*, pp. 225–240. Marcel Dekker.

Valiant, L. G. (1984). A Theory of the Learnable. *Communications of the ACM*, *27*, 1134–1142.