Active Learning for Subsampling Functions Below a Specified Level-Set

Brent Bryan Jeff Schneider Chad M. Schafer

December 2007
CMU-ML-07-120

Carnegie Mellon

Active Learning for Subsampling Functions Below a Specified Level-Set

Brent Bryan[1] **Jeff Schneider**[2] **Chad M. Schafer**[3]

December 2007
CMU-ML-07-120

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present an efficient algorithm to actively select samples which accurately model a target function below a specified threshold. We describe several heuristics for choosing these samples, and show how these heuristics perform on synthetic and real target functions. We then show how these algorithms can be used to make a sophisticated statistical procedure for finding confidence regions two orders of magnitude more data efficient.

[1] Machine Learning Department, Carnegie Mellon University, Pittsburgh PA, USA
[2] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA , USA
[3]Department of Statistics, Carnegie Mellon University, Pittsburgh, PA USA

Keywords: Active Learning, Level-Sets, Statistical Inference, Cosmology

1 Introduction

In many scientific and statistical applications, one desires to learn a target function based on a (smallish) set of examples. However, in some cases one is not interested in the exact value of the function over the entire domain, but instead in finding those regions where the function is less (or greater) than some particular value. For instance, a geologist may be interested only in geographically modeling the concentration of some mineral or oil deposit where the deposit's concentration is above some financially viable threshold. In this work, we consider the problem of finding, given observed data, the subset of all candidate models for which a goodness-of-fit measure exceeds a cutoff value. This is a critical initial step to the utilization of the Minimax Expected-Size (MES) confidence procedure, described below, as it ensures that approximations made in the procedure remain valid moderate sample sizes.

The problem of determining which fixed number of samples to choose to best model a target function is NP-hard. However, actively choosing samples to learn a concept based on a model of the true function is known to significantly reduce the required number of samples, in some cases by exponential factors (e.g.[1]). Active strategies for selecting samples include choosing points where the model predicts poorly [8], points where the model has low confidence in its prediction [14], points where we expect the model to have the greatest change (e.g. [7]), and points which reduce the variance of the model (e.g.[9]). Recently, [5] showed that choosing samples in proportion to one's uncertainty about the target was $(1-\frac{1}{e})$ optimal when trying to predict the target function over the entire domain. While one could employ algorithms for detecting function level-sets [10, 2], and then choose samples with large variances within the detected level-sets, the authors are unaware of any heuristics which directly sample functions in regions above (or below) a specified threshold.

In this work we present a new heuristic based on the straddle heuristic of [2] for actively sampling a target function where one is interested in points above (or below) a specified threshold. We show that this technique is more efficient in predicting the target function's value above (or below) the specified threshold than global variance minimization, as well as other heuristics. Moreover, the heuristic easily scales to multiple dimensional and continuous valued input spaces. In Section 3, we demonstrate the utility of the heuristic on several real and synthetic data sets. Finally, in Section 4, we show how it can be used to speed up the computation of confidence regions for a cosmological problem using the MES confidence procedure by two orders of magnitude.

1.1 Motivating Astronomical Problem

One key prediction of the Big-Bang model for the origin of the universe is the presence of a $2.73K$ cosmic microwave background (CMB) afterglow emitted when electrons recoupled with protons and neutrons forming atoms. Since its emission shortly after the formation of the universe, this radiation has permeated throughout the universe, providing a unique probe of the primordial universe. An important summary of the CMB is its power spectrum, the shape of which is affected by at least seven cosmological parameters: optical depth (τ), dark energy mass fraction (Ω_Λ), total mass fraction (Ω_m), baryon density (ω_b), dark matter density (ω_dm), neutrino fraction (f_n), and spectral index (n_s). Constraining these parameters is the focus of much recent effort in the cosmological community as these parameters describe the composition, age and eventual fate of

the universe.

Several programs exist for computing theoretical models of the CMB power spectrum given the values of the seven cosmological parameters above [13]. than computing the full Boltzmann solution, these codes take roughly 5 minutes to perform a single model computation. As such, care must be taken to minimize the number of CMB models required to perform the desired statistical analysis.

1.2 Minimax Expected-Size Confidence Procedure

While there are a number approaches that one could use to derive confidence regions, the Minimax Expected-Size (MES) confidence procedure has many advantages. This frequenist procedure typically has both high power and correct $1-\alpha$ coverage [12]. Emperically, we have found that confidence regions generated by MES are significantly smaller than their counterparts generated with other confidence procedures. For example, while χ^2 tests are ubiquitously used in many domains, they are known to be conservative, resulting in regions that are larger than necessary [15]. While techniques have been developed to reduce the loss of power of χ^2 tests [2], they are generally complex and result in confidence regions larger than those obtained by MES.

Another popular technique to derive "confidence" regions is the use of Markov Chain Monte Carlo (MCMC). Instead of directly deriving the credible regions for the data, MCMC computes the approximate posterior of the likelihood and then highest posterior density regions can be found. However, there is no guarantee that credible regions derived from a posterior will contain the true value of the parameter in at least $1-\alpha$ fraction of the instances in which the technique is applied. This problem becomes particularly acute in high-dimensional and non-parametric models, where $1-\alpha$ credible intervals may trap the true value of the parameter close to zero percent of the time [15].

The MES confidence procedure is described in detail in [12, 3]. [3] shows how it can be formulated as a convex game where the row player (nature) tries to find the worst possible truth θ in some set parameter space Θ', while the column player (the statistician) develops a strategy which will minimize the size of the confidence regions subject to the $1-\alpha$ coverage constraint. A minimax solution to the convex game ensures the statistician that the derived confidence regions are guaranteed to be no larger than some fixed size (the value of the game). For any other strategy the statistician could have chosen, there exists some possible truth, $\theta \in \Theta'$, for which the confidence regions derived by the statistician will be larger than the value of the game.

While the convex game formulation and solutions given in [3] are computationally efficient, the MES method is not data efficient. In particular, when building the convex game, it is often desirable to limit Θ' to just those models within a confidence region computed by a broad χ^2 test, as this results in a better approximation by the discrete samples used in the convex game. While the MES procedure could be used without the χ^2 cut, the size of the convex game would have to be orders of magnitude larger to obtain the same accuracy in some cases. Let $\Theta \subseteq \mathbb{R}^d$ be the a priori parameter set, and let Θ' be those parameter vectors from Θ that satisfy the χ^2 test. While conservative, the χ^2 test will exclude from Θ those models that are very poor fits to the data. By selecting the value of the χ^2 cut large enough, say $1-\alpha/10$, and using a Bonferroni correction, we can then ensure that the combination of the χ^2 cut along with the MES procedure will result in

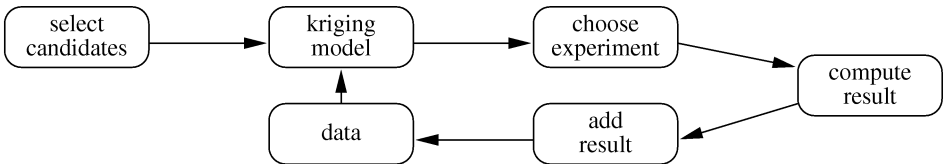

Figure 1: Outline of our sampling algorithm. Given an initial set of points (possibly empty), we randomly select a set of candidates and score them according to a kriging model. The best scoring point is chosen, and we evaluate the function at that point and add it to our data set.

confidence regions with the correct coverage, while increasing the procedure's chances of rejecting incorrect models. [12, 3] choose models randomly from Θ and perform the χ^2 test to create Θ'. However, for many problems, including the CMB problem mentioned above, only a fraction of the parameter vectors in Θ result in models within the χ^2 cutoff and therefore parameter vectors in Θ'. For computationally expensive models, such as the CMB, this data inefficiency makes the MES procedure intractable. We now propose an active learning algorithm for efficiently constructing Θ' given Θ.

2 Algorithm

Let us formalize the problem. Assume that we are given a bounded sample space $\Theta \subseteq \mathbb{R}^d$ and a scoring function: $f : \Theta \to \mathbb{R}$. Given a threshold t, we want to find the set $\Theta' \subseteq \Theta$ where f is equal to or less than the threshold: $\{\theta \in \Theta' | \theta \in \Theta, f(\theta) \leq t\}$. If f is invertible, then the solution is trivial. However, it is often the case that f is not trivially invertible, such as the CMB model mentioned in §1.1. In these cases, we can discover Θ' actively choosing samples from Θ, which help us efficiently refine our estimate of Θ'. In practice we will only be able to sample p points from Θ due to computational resources. We desire to choose these p points in order to minimize the estimated variance of f over all $\theta \in \Theta'$.

For cases where the cost to compute a model based on a parameter vector $\theta \in \Theta$ is significant, care should be taken when choosing the next sample, as picking optimum points can reduce the run time of the algorithm by orders of magnitude. Thus, it is preferable to analyze current knowledge about the underlying function and select samples which quickly refine the estimate of f, where $f(\theta) \leq t$. Since it is intractable to estimate the value of f over Θ (as Θ is continuous), we select 1000 random points from Θ to use as a candidate set. We then approximate the value of f at each of these candidate points and select one at which we compute f. This point and the resulting value of f is added to our data set and the process is repeated. The algorithm is illustrated in Figure 1.

There are several methods one could use to approximate f, notably some form of parametric regression. However, we chose to approximate f using Gaussian process regression, as other forms of regression may smooth the data, ignoring subtle features of the function that may become pronounced with more data. A Gaussian process is a non-parametric form of regression. Predictions for unobserved points are computed by using a weighted combination of the function values for those points which have already been observed, where a distance-based kernel function

is used to determine the relative weights. These distance-based kernels generally weight nearby points significantly more than distant points. Thus, assuming the underlying function is continuous, Gaussian processes will perfectly describe the function given an infinite set of unique data points.

Here, we use ordinary kriging, a form of Gaussian processes that assumes that the semi-variance, $\mathcal{K}(\cdot,\cdot)$, between two points is a linear function of their distance [6]; for any two points $s_i, s_j \in \mathcal{P}$,

$$\mathcal{K}(s_i, s_j) = \frac{k}{2}\mathsf{E}\left[\left(f(s_i) - f(s_j)\right)^2\right]$$

where k is a constant — known as the kriging parameter — which is an estimate of the maximum magnitude of the first derivative of the function. Therefore, the expected semi-variance between two points, $\theta_i, \theta_j \in \Theta$ is given by $\gamma(\theta_i, \theta_j) = \mathsf{E}(\mathcal{K}(\theta_i, \theta_j)) = k\mathcal{D}(\theta_i, \theta_j) + c$ where $\mathcal{D}(\cdot,\cdot)$ is a distance function defined on the parameter space Θ and c is the observed variance (e.g. experimental noise) when repeatedly sampling the function f at the same location. We have found that using a simple weighted Euclidean distance function where each dimension is linearly scaled such that the semi-variance along the axis is unity reasonably ensures that parameters are given equal consideration given their disparate values and derivatives. We conservatively set $k = 2$ and $c = 1 \times 10^{-5}$.

For the Gaussian process framework, sampled data are assumed to be Normally distributed with means equal to the true function and variance given by the sampling noise. Moreover, a combination of any subset of these points results in a Normal distribution. Thus, we can use the observed set of data, $\mathcal{A} \subset \Theta$, to predict the value of f for any $\theta_q \in \Theta$. This query point, θ_q, will be normally distributed, $(N(\mu_{\theta_q}, \sigma_{\theta_q}))$, with mean and variance given by

$$\begin{aligned} \mu_{\theta_q} &= \bar{f}_{\mathcal{A}} + \Sigma_{\mathcal{A}q}^{T}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}(f_{\mathcal{A}} - \bar{f}_{\mathcal{A}}) \\ \sigma_{\theta_q}^2 &= \Sigma_{\mathcal{A}q}^{T}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}\Sigma_{\mathcal{A}q} \end{aligned}$$

where the elements of the matrix $\Sigma_{\mathcal{A}\mathcal{A}}$ and arrays $\Sigma_{\mathcal{A}q}$ and $f_{\mathcal{A}} - \bar{f}_{\mathcal{A}}$ are given by

$$\begin{aligned} \Sigma_{\mathcal{A}\mathcal{A}}[i,j] &= \gamma(a_i, a_j) \\ \Sigma_{\mathcal{A}q}[i] &= \gamma(a_i, \theta_q) \\ (f_{\mathcal{A}} - \bar{f}_{\mathcal{A}})[i] &= f(s_i) - \bar{f}_{\mathcal{A}} \end{aligned} \qquad \bar{f}_{\mathcal{A}} = \frac{1}{|\mathcal{A}|}\sum_{i=1}^{|\mathcal{A}|} f(a_i)$$

and the a_i's and a_j's are the observed data used to make an inference: $a_i, a_j \in \mathcal{A}$, $0 \leq i, j \leq |\mathcal{A}|$.

As given, for a set of n observed points ($|\mathcal{A}| = n$), prediction with a Gaussian process requires $O(n^3)$ time, as an $n \times n$ linear system of equations must be solved. However, for many Gaussian process — and ordinary kriging in particular — the correlation between two points decreases as a function of distance. Thus, the full Gaussian process model can be approximated well by a local Gaussian process, where only the k nearest neighbors of the query point are used to compute the prediction value; this reduces the computation time to $O(k^3 + \log(n))$ per prediction, since $O(\log(n))$ time is required to find the k-nearest neighbors using spatial indexing structures (e.g. kd-trees).

2.1 Choosing Samples from Among Candidates

Given a set of random candidate points, the algorithm evaluates each one and chooses the point with the highest score according to some heuristic as the location for the next sample. Here we consider:

Random: One of the candidate points is chosen uniformly at random. This method serves as a baseline for comparison, as it selection technique employed by [12, 3].

Variance: Since variance is related to the distance to nearest neighbors, this strategy chooses points that are far from areas currently searched and ultimately evenly covers the space exhaustively [9].

Straddle & Variance: Instead of mapping f throughout Θ, a more practical solution is to concentrate on f where $f(\theta) \leq t$. One strategy is to first spend a fixed number of samples to locate the boundary of Θ' using the straddle heuristic of [2]: $\text{straddle}(\theta_q) = 1.96\sigma_{\theta_q} - |f(\theta_q) - t|$. The remaining samples are then selected to be those that are predicted to be within the boundary with the largest expected variance. If none of the points in the candidate set is predicted to be within the boundary region, we pick the point that most helps refine our boundary estimate: the point with the largest straddle value.

Entropy$\times$Variance & Variance: Similar to the last sampling heuristic, this heuristic first tries to estimate the location of the function's level-set and then sample within this level-set. However, instead of using the straddle heuristic, we use the product of misclassification entropy and variance to find the boundary [10]; here entropy is defined to be $-p\log_2(p) - (1-p)\log_2(1-p)$, where p is the probability that the candidate point is above the threshold, t.

Variance Above Threshold: Note that the previous two heuristics switch from exploration to exploitation after a fixed number of samples. If the estimate of the boundary does not correctly include all locations in Θ where $f \leq t$, then samples chosen by the heuristics will only be a subset of the desired Θ'. Intuitively, we desire our heuristic to actively be trading off exploitation for exploration throughout the selection process. We propose a modified version of the straddle heuristic which preforms this trade off:

$$\text{threshvar}(\theta_q) = 1.96\sigma_{\theta_q} - \max\left\{0, f(\theta_q) - t\right\}.$$

Note that when $f < t$, the threshvar heuristic is solely a function of variance, leveraging the theoretical results [5]. When $f > t$, then the threshvar heuristic reverts to the straddle heuristic of [2], selecting samples which are expected to be near the boundary, and far from other samples. We also tried a form of this heuristic derived from the entropy and variance product of [10]; however this heuristic consistently performed worse than the threshvar heuristic.

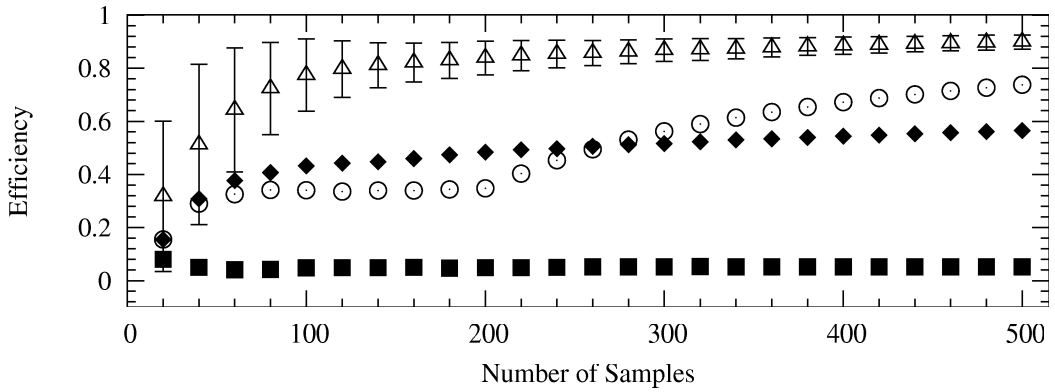

Figure 2: Efficiency of the straddlevar(0) (triangle), straddlevar(200) (open circle), threshvar (diamond) and variance (square) heuristics as a function of the number of samples chosen. Note the sharp increase in efficiency in the straddlevar(200) heuristic after 200 samples, when the heuristic switches from a exploration to an exploitation mode. The threshvar heuristic out-performs all other heuristics while they remain in their exploration mode.

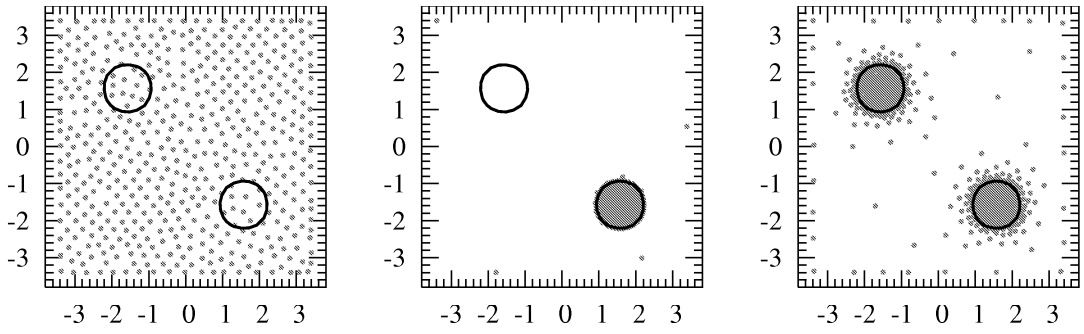

Figure 3: 500 samples chosen by the variance (left), straddlevar(0) (center) and threshvar (right) heuristics in a single trial. The variance heuristic focuses only on exploration, while the straddlevar(0) heuristic focuses only on exploitation, ignoring one of minima. The threshvar heuristic balances exploration with exploitation, both finding and frequently sampling the two minima.

	2D Peak	2D DeBoor	2D Sin.	4D Sin.	SNLS	WMAP YR1
random	0.06±0.01	0.04±0.01	0.55±0.02	0.13±0.01	0.06±0.01	0.001
variance	0.05±0.00	0.04±0.00	0.53±0.01	0.11±0.01	0.06±0.01	—
straddlevar(0)	0.89±0.03	0.89±0.03	0.90±0.02	0.62±0.01	0.71±0.04	—
straddlevar(200)	0.74±0.01	0.75±0.02	0.83±0.01	0.52±0.02	0.52±0.02	—
entvar-thresh(0)	0.89±0.02	0.88±0.04	0.90±0.01	0.62±0.01	0.71±0.03	—
threshvar	0.57±0.01	0.67±0.03	0.83±0.01	0.39±0.01	0.16±0.01	0.32

Table 1: Efficiency of various heuristics after picking 500 samples. The threshvar heuristic has greater efficiency than all heuristics that remain in an exploration mode throughout the trial (random, variance), and is competitive with heuristics that switch to an exploitation mode (straddlevar(·), entvar-thresh(0)).

3 Experiments

We now assess how the previously mentioned heuristics perform on several real and synthetic data sets using two metrics: efficiency and variance. We define efficiency to be the fraction of samples from Θ which are truly in Θ'. We also assess the heuristics based upon their maximum expected variance for points in Θ'. As the exact value of the maximum expected variance is computationally prohibitive — as it requires computing the variance at all $\theta \in \Theta'$ — we estimate the maximum expected variance using a randomly selected subset of points from Θ'. We desire heuristics which have high efficiency, and low variance. Heuristics which favor exploitation over exploration will have high efficiency, but also high variance, while those that favor exploration will have lower efficiency and variance.

Five target functions where chosen to assess the various heuristics. We chose to look at the 2D DeBoor function used in [10] (deboor2d), as well as the 2D and 4D sinusoidal functions used in [2] (sin2d and sin4d, respectively). We also developed a 2D sinusoidal function, sinpeak, with two distinct minima located away from the parameter space boundaries given by $f(x, y) = \sin(x)\sin(y)$. This function was particularly troublesome for the heuristics which only had a limited number of samples to detect the boundary before switching to an exploitation mode. Finally, we tested the heuristics on the task of computing 95% confidence regions for the Supernova Legacy Survey data set [11] using χ^2 tests.

Results for the various heuristics on these five data sets after performing 500 samples each in 20 trials are shown in Tables 1 and 2. We have denoted the Random, Variance, Straddle & Variance, Entropy×Variance & Variance, and Variance Above Threshold heuristics as random, variance, straddlevar, entvar-thresh, and threshvar, respectively. The number in parentheses after straddelvar and entvar-thresh indicates how many samples were chosen to define the boundary before switching to an exploitation mode.

As Tables 1 and 2 show, there is a strong correlation between efficiency and maximum expected variance among the points in Θ'. The most efficient heuristic on all of the experiments was straddlevar(0). However, this efficiency came at the cost of failing to locate all of the desired minima seen in Figure 3. Random sampling and variance-weighted sampling performed equally well

	2D Peak	2D DeBoor	2D Sin.	4D Sin.	SNLS
random	0.39±0.00	0.43±0.02	1.17±0.11	3.78±0.13	13052±1163
variance	0.32±0.00	0.35±0.01	0.69±0.01	2.76±0.02	9082±53
straddlevar(0)	1.02±0.37	0.16±0.00	2.09±1.35	6.60±0.66	36941±16408
straddlevar(200)	0.17±0.00	0.17±0.00	0.61±0.00	3.59±0.16	13377±3511
entvar-thresh(0)	1.00±0.39	0.16±0.00	2.38±1.07	6.61±0.66	32304±5481
threshvar	0.17±0.00	0.17±0.00	0.61±0.00	3.23±0.12	8316±264

Table 2: Maximum residual variance of various heuristics after picking 500 samples. The threshvar heuristic has the minimum variance of all heuristics on points where the function is less than the given threshold (except the 4D Sin. case).

in terms of efficiency. However, variance-weighted sampling resulted in lower expected variance. Unsurprisingly, the entvar-thresh(0) heuristic performs similarly to the straddlevar(0) heuristic.

As it generally takes only a few dozen to a hundred points to locate the minima in two dimensions, the straddlevar(200) heuristic appears to be the winner for these low dimensional tasks. This also is unsurprising, as the constant 200 was chosen to approximately maximize the performance of the straddlevar heuristic on the 2D problems. After quickly finding the minima, the straddlevar(200) heuristic was able to switch to an exploitation mode (seen in Figure 2), allowing it to obtain both high efficiency and low expected variance. However, a major drawback of this heuristic is choosing the number of samples after which to switch to an exploitation mode. Using too few samples results in failure to locate the desired minima (shown in Figure 3), while using too many samples results in worse efficiency than the threshvar heuristic (as seen in Figure 2). While using 200 samples for boundary predication for straddlevar results in a good trade off between discovery and exploitation for the 2D problems, it does not fare as well on the Sin4D or SNLS problems. In practice it is very difficult to choose this exploration constant without knowing the underlying function.

The threshvar heuristic performs well both in terms of efficiency and expected variance in Θ'. While it's efficiency was less than heuristics that focused on exploitation, it generally has the minimal expected variance among all the heuristics. This indicates that the heuristic is correctly identifying and sampling the correct regions of the function. Moreover, its performance is similar to that of the straddlevar(200) heuristic for the 2D tasks, without switch to an exclusively exploitation mode.

All of the heuristics perform poorly on the variance metric for the 4D sinusoidal function, as parts of parameter space have semi-variances far in excess of 10 times the mean semi-variance. Increasing the kriging parameter results in more similar variance estimates for the threshvar and variance heuristics, while the threshvar heuristic remains more efficient than the variance heuristic.

4 Statistical Analysis of the Cosmic Microwave Background

Let us now demonstrate the utility of active learning heuristics for sampling functions below a specified threshold by demonstrating how they can be employed to increase the data efficiency of MES by orders of magnitude. In particular, let us revisit the problem of computing minimax expected size confidence regions for the seven dimensional cosmological model discussed in §1.1. For this task, we will consider fitting WMAP Year 1 data [11] with test power spectra from CMBFast [13]. We fix Θ, the range of parameters to be searched to be equivalent to that used by [4].

The task of deriving confidence regions for the WMAP Year 1 data is interesting both from an scientific (astrophysical) perspective, as well as from an algorithmic (machine-learning) perspective. Scientifically, tight confidence regions on the cosmological parameters allow cosmologists to better understand the primordial universe, and consequently discover information about the relative density of different types of matter and energy in the universe. Algorithmically, the WMAP Year 1 data is exciting, as it provides a challenging real-world testbed. Not only is the parameter space relatively large — 7D — but [2, 4] show that there are at least two disjoint regions in parameter space which fit the data reasonably well. Moreover, simple algorithms, such as grid based approaches or randomly sampling are computationally intractable due to the size of the parameter space; a simple grid with 100 samples per dimension would take a nearly billion years of CPU time on current hardware.

Instead, let us look at an active learning approach for computing MES confidence regions. As discussed in §1.2, it is desirable to limit the parameter space Θ to just those models which fit the data well: Θ'. For this application, we chose to use a χ^2 cut with $\alpha = 3.6 \times 10^{-11}$. While the probability that this cut removes the true value of $\theta \in \Theta'$ is quite small (just $3.6 \times 10^{-13}\%$), this cut reduces the size of Θ' to be just under 0.2% of the size of Θ. For the MES procedure, we now desire an even sampling of points from Θ'. Since the procedure described in § 2 uses a fixed homogeneous semi-variance function, this desire amounts to picking points from Θ below the χ^2 cutoff with the largest expected variance. In the previous section, we demonstrated how the threshvar heuristic efficiently performs this task. Indeed, using a data set of roughly 800,000 samples sampled uniformly at random from Θ, we find that only 0.12% of the samples are in Θ'. However, sampling 74,000 samples using the threshvar heuristic results in roughly 33% of the samples within Θ', a 277 times increase of efficiency. More importantly, using the threshvar heuristic allowed us to sample 25,000 samples from Θ' in just 8.5 CPU months, as opposed to the 201 CPU years that would have been required for random sampling to complete this task.

5 Conclusions

We have described an algorithm for efficiently sampling a function above (or below) a specified threshold value, and have evaluated several heuristics for actively choosing samples. Our experiments indicate that the Variance Above Threshold heuristic, threshvar, outperforms other heuristics, and significantly out performs random sampling. We find that using the threshvar heuristic is two orders of magnitude more efficient than random sampling when computing the restricted parameter space based on a χ^2 cut for the CMB confidence region problem. Thus, the thresh-

var heuristic allows us to perform the MES confidence procedure in a data and computationally tractable manner.

References

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[2] B. Bryan, et al. Active learning for identifying function threshold boundaries. In *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2005.

[3] B. Bryan, et al. Efficiently computing minimax expected-size confidence regions. In *ICML '07: Proceedings of the 24th International Conference on Machine learning*, New York, NY, 2007. ACM Press.

[4] B. Bryan, J. Schneider, R. C. Nichol, C. J. Miller, C. R. Genovese, and L. Wasserman. Mapping the cosmological confidence ball surface. *Astrophysical Journal*, 665:25, August 2007.

[5] C. Guestrin, et al. Near-optimal sensor placements in gaussian processes. In *ICML '05: Proceedings of the 22nd International Conference on Machine learning*, page 265, New York, NY, 2005. ACM Press.

[6] N. A. C. Cressie. *Statistics for Spatial Data*. Wiley, New York, 1991.

[7] D. A. Cohn, et al. Improving generalization with active learning. *Machine Learning*, 15(2):201, 1994.

[8] A. Linden and F. Weber. Implementing inner drive by competence reflection. In *In Proceedings of the 2nd International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1993. MIT Press.

[9] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[10] N. Ramakrishnan, et al. Gaussian processes for active data mining of spatial aggregates. In *Proceedings of the SIAM International Conference on Data Mining*, 2005.

[11] P. Astier, et al. The Supernova Legacy Survey: measurement of Ω_M, Ω_Λ and w from the first year data set. *Astronomy and Astrophysics*, 447:31–48, February 2006.

[12] C. M. Schafer and P. B. Stark. Constructing Confidence Sets of Optimal Expected Size. Technical Report 836, Department of Statistics, Carnegie Mellon University, 2006.

[13] U. Seljak and M. Zaldarriaga. A Line-of-Sight Integration Approach to Cosmic Microwave Background Anisotropies. *Astrophyical Journal*, 469:437–+, October 1996.

[14] S. B. Thrun and K. Möller. Active exploration in dynamic environments. In *Advances in Neural Information Processing Systems*, volume 4, pages 531–538. Morgan Kaufmann Publishers, Inc., 1992.

[15] L. Wasserman. *All of Statistics*. Springer-Verlag, New York, 2004.

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon