Actively Learning Level-Sets of Composite Functions

Brent Bryan Jeff Schneider

December 2007
CMU-ML-07-121

Carnegie Mellon

Actively Learning Level-Sets of Composite Functions

Brent Bryan[1] **Jeff Schneider**[2]

December 2007
CMU-ML-07-121

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Scientists frequently have multiple types of experiments and data sets on which they can test the validity of their models and the plausible or optimal regions for the model parameters. Identifying these parameter regions reduces to finding a level set on a function defined as a composite of the evaluations of each experiment or data set for a parameter setting. An active learning algorithm for this problem must at each iteration select a parameter setting to be tested and decide which experiment type to use for the test. We propose an active learning algorithm for identifying level sets of composite functions. Empirical tests on synthetic functions and on real data for a 7D cosmological model show it significantly reduces the number of samples required to identify desired regions.

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

Keywords: Active Learning, Level-Sets

1 Introduction

Active learning of target functions using informed choices of future experiments has long been known to drastically decrease a problem's sample complexity [1]. Many sampling heuristics have been developed to learn either the entire target function (e.g. [9, 3]) or some feature of the target function, such as its level sets (e.g. [2, 10]). However, in some applications the target function is actually a composite of several component functions that are evaluated independently. One example is determining the spatial location of a disease outbreak using information derived from medical records (e.g. hospital admits), as well as sales of over the counter and prescription medications. Another example is cost/benefit analysis of resource extraction where one must estimate the value of all resources to be obtained along with the extraction costs in terms of infrastructure and human resources required. In this work, we focus on a third application: simultaneous statistical analysis of multiple related data sets for finding valid parameter ranges in scientific models.

Scientists frequently have multiple types of experiments and data sets on which they can test the validity of their models and the plausible or optimal regions for the model parameters. They would like to find the region of parameter space that is statistically plausible for all types of experiments. A test on a single data set may be sufficient to reject a particular model or parameter setting without testing other data sets. Traditionally, this has been achieved in the sciences in a somewhat ad-hoc fashion where one scientist publishes plausible parameters derived from one type of experiment and another uses that information to guide the selection of parameters in future experiments. In Bayesian analysis, results from one experiment might form the priors for the next. A more rigorous and efficient approach is to consider multiple experimental sources of evaluation simultaneously and choose evaluation samples in light of their contribution to the combined evaluation function. In our empirical efforts, we consider two independent data sets for evaluating a 7D cosmological model.

Joint Statistical Analysis Joint analyses tend to take one of two forms. In the first we create statistical model which simultaneously considers all data sets. For instance, when performing an analysis on two data sets using χ^2 tests, we will have one χ^2 test for data set A and a second for data set B. However, since the χ^2 test assumes that each of the data points have dependencies given by the covariance matrix, we can combine the two tests into a single χ^2 test of the form

$$\left[(\vec{x}_A - \vec{\mu}_A)^T, (\vec{x}_B - \vec{\mu}_B)^T\right] \begin{bmatrix} \Sigma_A & \Sigma_{AB} \\ \Sigma_{AB} & \Sigma_B \end{bmatrix}^{-1} \begin{bmatrix} \vec{x}_A - \vec{\mu}_A \\ \vec{x}_B - \vec{\mu}_B \end{bmatrix} \sim \chi^2_{(a+b)}$$

where $x_\star$, $\mu_\star$ and $\Sigma_\star$ are the associated test model, observed data and observed covariance of model $\star$ given some vector from the parameter space, a and b are the degrees of freedom of the tests associated with data sets A and B respectively, and Σ_{AB} is the covariance of the data points between data sets A and B. If data sets A and B are independent, then all elements of Σ_{AB} are zero, and we can write the above expression as:

$$(\vec{x}_A - \vec{\mu}_A)^T \Sigma_A^{-1} (\vec{x}_A - \vec{\mu}_A) + (\vec{x}_B - \vec{\mu}_B)^T \Sigma_B^{-1} (\vec{x}_B - \vec{\mu}_B) \sim \chi^2_{(a+b)}.$$

That is, the target function is merely the sum of the two observable functions: the variance weighted sum of squares for both data sets.

Another approach to performing simultaneous joint analyses is to combine the models' p-values. There are many ways to combine test procedures, including using Bonferroni corrections, the inverse normal method, and inverse logit methods [8]. However, the most common method to combine p-values is Fisher's method [7]. Fisher noted that since a p-value, p_i, has a Uniform distribution, then $-2\log(p_i)$ will have a $\chi^2_{(2)}$ distribution. Again, using the fact that the sum of independent χ^2 random variables has a χ^2 distribution, the test becomes: reject H_0 if and only if:

$$-2\sum_{i=1}^{k}\log(p_i) \geq C$$

where C is the critical value of a $\chi^2_{(2k)}$ distribution for some particular level α. Again, we see that the target function is the sum of two observable functions.

Active Learning Composite Functions As we have seen, the target function that we are interested in learning for combining statistical evidence is some composite of readily available observable functions. In particular, the previous two techniques rely on the sum of observable functions. It is clearly possible to sample all observable functions at each query point and then directly compute the value of the target function, effectively reducing the problem into a standard active learning problem. However, such an approach disregards any strong evidence provided by a single statistical test, and hence may result in extraneous sampling of the remaining statistical models.

Instead, we are interested in active learning algorithms which use information about each observable function to learn some composite target function. In this work, we propose a heuristic for actively learning level sets of composite functions of sums for continuous valued input spaces, without arbitrarily restricting the input space (e.g. imposing a grid). In Section 3, we show that this heuristic performs the level-set discovery task more efficiently than both random and sequential sampling of the constituent functions using state of the art heuristics. Finally, in Section 4, we demonstrate the utility of our algorithm by computing confidence regions for seven cosmological parameters using two independent data sets.

2 Active Learning Algorithm

Suppose we are given some sample space $\Theta \subseteq \mathbb{R}^d$ and a set of observable functions $f_i : \Theta \to \mathbb{R}$ $(i = 1, 2, \ldots, m)$, such that $\sum_{i=1}^{M} f_i(x) = f(x)$, where f is the target function we are interested in learning. Given a threshold t, we want to find the set of points Θ' where f is equal or less than the threshold: $\{s \in \Theta' | s \in \Theta, f(s) \leq t\}$. In general, computing the value of each $f_i(x)$'s may not incur the same cost. However, in this work we will assume that the costs are similar, and hence try to minimize the total number of samples of all observable functions, f_i, required to accurately estimate Θ'. Moreover, we assume that f cannot be directly sampled, and that neither f nor any of the f_i's is invertible. That is, the only way to estimate the level-sets of f is to sample points from the f_i's and infer f. This formulation accurately mimics combining p-values using Fisher's method, as the method for finding the individual p-values may be entirely unknown.

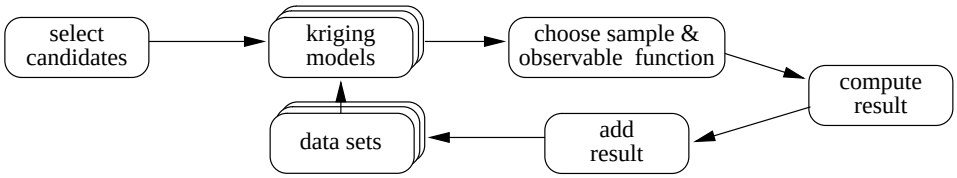

Figure 1: Outline of our sampling algorithm. Given an initial set of points (possibly empty), we randomly select a set of candidates and score them using a set of kriging models. The best scoring point and observable function pair is chosen, and we evaluate the selected observable function at the given point. This data is added to the corresponding data set.

We must now determine how best to choose samples both among and within the f_i's. Ideally, we want to sample the observable function f_i at the point which best increases our prediction accuracy (e.g. whether a point is above or below the threshold) over f. Since the parameter space is continuous and multi-dimensional, we cannot afford to test all possible points and observable functions to find the best. Instead, we model each of the observable functions given the current samples taken from that function. Then, for each experiment, we randomly select a small subset of the parameter space (usually 1000 points drawn uniformly at random) and choose the best point and observable function pair upon which to experiment from among these candidates. We compute the value of the observable function at the desired point and add it to the data set used to model that function, and the process is repeated. The algorithm is illustrated in Figure 1.

There are several methods one could use to model each of the f_i's, notably some form of parametric regression. However, we chose to approximate the $f_i(s)$ 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.

In particular, we use ordinary kriging, a form of Gaussian processes that assumes the semivariance, $\mathcal{K}(\cdot,\cdot)$, between two points is a linear function of their distance [4]; for any two points $s, s' \in \Theta$,

$$\mathcal{K}(s,s') = \frac{k}{2}\mathsf{E}\left[\Big(f_i(s) - f_i(s')\Big)^2\right]$$

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, $s, s' \in \Theta$ is given by $\gamma(s,s') = E(\mathcal{K}(s,s')) = k\mathcal{D}(s,s') + c$ 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_i at the same location. We have found that using a simple weighted Euclidean distance function where each dimension is scaled linearly reasonably ensures that parameters are given equal consideration given their disparate values and derivatives. For our

analysis, we adjusted the weights of each dimension in the distance function to ensure that the mean semi-variance along each axis was approximately unity during the sampling process. Additionally, we conservatively set $k = 2$ and $c = 1 \times 10^{-5}$ (to account for round-off errors).

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_i for any $s_q \in \Theta$. This query point, s_q, will be Normally distributed, $(N(\mu_{i,s_q}, \sigma_{i,s_q}))$, with mean and variance given by

$$\mu_{i,s_q} = \bar{f}_i(\mathcal{A}) + \Sigma_{\mathcal{A}q}^T \Sigma_{\mathcal{A}\mathcal{A}}^{-1} (f_i(\mathcal{A}) - \bar{f}_i(\mathcal{A})) \quad (1)$$

$$\sigma_{i,s_q}^2 = \Sigma_{\mathcal{A}q}^T \Sigma_{\mathcal{A}\mathcal{A}}^{-1} \Sigma_{\mathcal{A}q} \quad (2)$$

where the elements of the matrix $\Sigma_{\mathcal{A}\mathcal{A}}$ and arrays $\Sigma_{\mathcal{A}q}$ and $f_i(\mathcal{A}) - \bar{f}_i(\mathcal{A})$ are given by

$$\begin{aligned} \Sigma_{\mathcal{A}\mathcal{A}}[j,k] &= \gamma(a_j, a_k) \\ \Sigma_{\mathcal{A}q}[j] &= \gamma(a_j, s_q) \\ (f_i(\mathcal{A}) - \bar{f}_i(\mathcal{A}))[j] &= f_i(s_j) - \bar{f}_i(\mathcal{A}) \end{aligned} \qquad \bar{f}_i(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{A}|} f_i(a_j)$$

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

As given, for a set of n_i observed points ($|\mathcal{A}| = n_i$), prediction with a Gaussian process requires $O(n_i^3)$ time, as an $n_i \times n_i$ 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 $O(k^3 + \log(n_i))$ per prediction, since $O(\log(n_i))$ time is required to find the k-nearest neighbors using spatial indexing structures.

2.1 Choosing Experiments

Given this active learning framework, we must now decide how to choose sample / observable function pairs. We consider the following heuristics:

Random One of the candidate points and the associated observable function, f_i, is chosen uniformly at random. This method serves as a baseline for comparison of the other heuristics.

Variance We choose the point which has the largest variance on any of the observable functions; our choice then, is the point with the maximum variance, and the observable function on which that variance was found. Using model variance to pick the next experiment is common for active learning methods whose goal is to map out the target function over a parameter space [9, 3]. In particular, [3] showed that greedily picking experiments based upon model variance performs nearly as well as using a mutual information heuristic when learning the target over the entire parameter space; this is significant, as the mutual information heuristic can be shown to be $(1-1/e)$ optimal [3]. Since variance is closely related to distance when using kriging, this heuristic samples

points which are distant from their nearest neighbors. However, when searching for level-sets, we are less interested in the function away from the level-set boundary, and instead want to focus our sampling resources near this predicted boundary. In particular, sampling based solely on variance results in substantially worse performance than heuristics than concentrate on the function level-set [2].

Information Gain Information gain is a common myopic metric used in active learning. Computing the information gain over the whole state space for each observable function provides an optimal 1-step experiment choice. In some discrete or linear problems this can be done, but is intractable for continuous non-linear spaces. As such we do not consider a traditional information gain heuristic, but rely on efficient point estimates which act as proxies for global information gain.

Sequential-Straddle As noted in the introduction, the problem can be simplified to a standard active learning problem if one sequentially samples each of the observable functions in order to directly compute f. [2] showed that in a setting where experiments yield the (approximately) true values of the target function, a good heuristic for level set identification is the straddle heuristic: $\text{straddle}(s_q) = 1.96\sigma^2_{s_q} - |\mu_{s_q} - t|$. This heuristic balances the need to explore uncertain parts of parameter space, with the desire to refine the model's estimate around those regions already known to be close to the level-set boundary. This heuristic leverages the straddle heuristic by choosing the candidate point with the highest combined straddle score,

$$\text{combined–straddle}(s_q) = 1.96 \sum_{i=1}^{m} \sigma^2_{i,s_q} - \left| \sum_{i=1}^{m} \mu_{i,s_q} - t \right|, \tag{3}$$

and then sequentially sampling all m observable functions at this point.

Variance-Straddle While [2] showed that the straddle heuristic works well when directly sampling the target function, we can hope to do better by considering the output from each observable function individually. For instance, if a sample point results in a very large value for one of the observable functions, it may be unlikely that the results of the other f_i's will be such that the resulting value of f is near the level-set. In particular, when dealing with the χ^2 models mentioned in the introduction, we know that $f_i(x) \geq 0$ for all i. Thus, if a single f_i is greater than the level-set boundary, the target function will also be greater than the level-set boundary, and hence it may be more efficient to sample elsewhere. This heuristic simply computes the combined-straddle score as in Equation 3, and then chooses the candidate point and observable function with the largest variance.

Variance-MaxVarStraddle Finally, we consider a variant of the straddle heuristic. This heuristic tries to mimic the information gain of choosing a particular point and observable function pair. Note that after observing a point, the variance of the kriging model is effectively zero at that point (since we have set c to be a very small positive value). The original straddle heuristic balances

the expected gain in the model fit (σ_{s_q}) with the expected distance of the point to the level-set boundary.

However, with the multiple model formulation, we do not expect the model variance to decrease by $\sigma_{s_q} = \sum_{i=1}^{m} \sigma^2_{i,s_q}$, but rather by σ_{i,s_q} where f_i is the observable function we pick. Thus, a more accurate estimate of the information gain of a candidate point and observable function pair is:

$$\text{Variance–MaxVarStraddle}(s_q) = \max_i \left\{1.96\sigma^2_{i,s_q}\right\} - \left|\sum_{i=1}^{m} \mu_{i,s_q} - t\right|.$$

We choose the candidate point that maximizes this heuristic and the corresponding f_i.

3 Experiments

We now assess the accuracy with which our active learning model reproduces known target functions for the sampling heuristics just described. This is done by computing the fraction of test points in which the predictive model (the sum of the kriging model associated with each observable function) agrees with the true target function about which side of the threshold the test points are on after some fixed number of experiments; for the four following functions, accuracies were assessed after 200 experiments. This process was repeated 20 times to account for variations due to the random nature of the candidate generation process. The first three target functions considered were sums of two observable functions, while the fourth was a sum of four observable functions. The considered functions are:

Gaussian This problem consisted of determining the 95% acceptance region of two axis aligned perpendicular two dimensional Gaussian distributions centered at the origin. Both Gaussians had diagonal covariance matrices with on diagonal elements of 1 and 16. Since working in probability space results in many near-zero values, the problem was considered in log-space. As such, the target function was a 2 dimensional symmetric quadratic function, and the level-set was a circle centered at the origin. The range of the parameter space ($x, y \in [-3.4, 3.4]$)

Sin2D The second problem consists of finding where the two 2D sinusoidal observable functions sum to zero where $x, y \in [0, 2]$. These observable functions were chosen because 1) the target threshold winds through the plot giving ample length to test the accuracy of the approximating model, 2) the boundary is discontinuous with several small pieces, 3) there is an ambiguous region around $(0.9, 1)$, where the true function is approximately equal to the threshold, and the gradient is small and 4) there are areas in the domain where the function is far from the threshold and hence we can see whether algorithms refrain from oversampling in these regions.

SimpleSin2D This problem is a simplified version of the previous problem, where the sinusoidal observable functions were chosen to reduce the problem's semi-variances (again $x, y \in [0 : 2]$). Since problems with large semi-variances result in large model variance estimates in the kriging

	Gaussian	SimpleSin2D	Sin2D	4-Sin2D
Random	> 1000	> 1000	> 1000	> 1000
Variance	95.0±11.0	> 500	105.0±11.5	188.6±32.2
Variance-Straddle	89.5±5.0	157.9±12.3	90.4±9.0	72.5±12.0
Sequential-Straddle	76.2±3.5	150.3±6.5	87.0±7.3	98.1±14.0
Variance-MaxVarStraddle	71.7±3.3	127.3±6.8	82.9±10.2	54.9±16.9

Table 1: Number of samples required to achieve a 99% accuracy on the Gaussian and SimpleSin2D tests, and a 90% accuracy on the Sin2D and 4-Sin2D tests based on 20 trials. The Variance-MaxVarStraddle heuristic consistently performs better than competitors.

models, such problems require extensive sampling to correctly identify function level-sets. Performance on this function highlights an algorithm's ability to quickly rule out portions of the function.

4-Sin2D This task consisted of finding where four 2D sinusoids sum to -2. The sinusoids chosen for this problem were similar to those of the SimpleSin2D problem. The resulting target function contains both regions of high slope, as well as regions with low derivatives near the specified threshold.

Classification accuracy results for the four tests are given in Table 1. Variance-MaxVarStraddle out performs all of the other heuristics on each of the target functions. Not surprisingly, the straddle-based heuristics beat out the random and variance-weighted heuristics, as both the random and variance-weighted heuristics sample the observable functions over the entire parameter space, while the straddle-based heuristics focus on the level-set of interest. Moreover, Variance-MaxVarStraddle beats out the Sequential-Straddle heuristic; this validates our supposition that treating each of the observable functions individually allows for additional learning opportunities.

One surprising result of our experimentation is that the Sequential-Straddle performs as well as the Variance-Straddle heuristic on all test functions the Gaussian, SimpleSin2D and Sin2D tasks. We believe that this result illustrates the fact that the Variance-Straddle heuristic is over estimating the importance of the variance component of the candidate points to the information gain of a point. The Variance-Straddle heuristic will be as likely to choose a candidate point where one of the two kriging models for the observable functions is large, and the other is zero as it is to choose a point where both estimated variances are equal (given the sum of the variances and estimated values for f_i are similar). However, the first candidate has much more information than the second, as selecting the first candidate will give us the (approximately) exact value of the target function, while selecting the second will only reduce the overall variance by a moderate amount. On the 4-Sin2D task the Variance-Straddle heuristic is able to make use of the individual observable functions, but still does not do as well as the Variance-MaxVarStraddle heuristic.

To illustrate the differences in sampling patterns between these heuristics, we plot the samples chosen for the observable functions (with squares, circles, triangles and x's, respectively) with the true (dashed) and predicted (solid) function level-sets for the 4-Sin2D task in Figure 2. The Variance-MaxVarStraddle heuristic is much better at picking points than the other two heuristics.

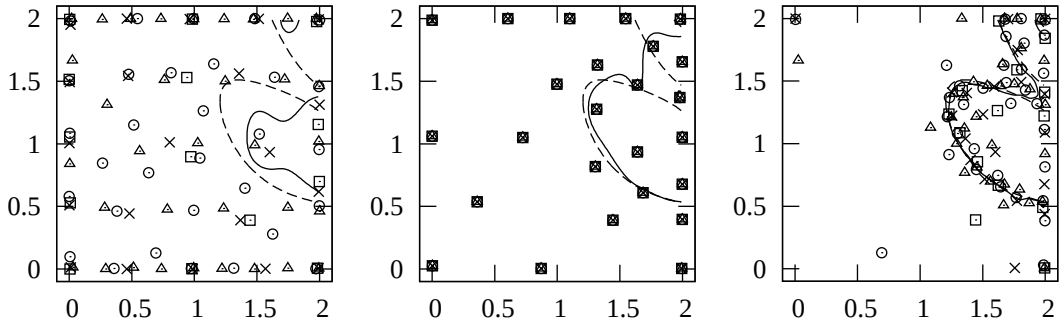

Figure 2: Predicted level-set (solid), true level-set (dashed) and experiments (squares, circle, triangles and x's) for the 4-Sin2D function after sampling 100 points using the Variance heuristic (left), the Sequential-Straddle heuristic (center), and C) the Variance-MaxVarStraddle heuristic (right).

Note that the Variance-MaxVarStraddle heuristic is able to learn that some regions of the space are poor without having to sample all observable functions; as such, its samples lie much closer to the target level-set. This reinforces our hypothesis that modeling the observable functions separately results in additional learning opportunities.

4 Joint Analysis of Cosmological Data Sets

Let us now look at a concrete application of this work: a statistical analysis of 7 cosmological parameters that affect the formation and evolution of our universe using two data sets. The first data set, the Cosmic Microwave Background (CMB) power spectra observed by the Wilkinson Microwave Anisotropy Project (WMAP), depicts temperature variations over the sky. The size and spatial proximity of these temperature fluctuations depict the types and rates of particle interactions in the early universe and hence characterize the formation of large scale structure (galaxies, clusters, walls and voids) in the current observable universe. It is conjectured that this radiation permeated through the universe unchanged since its formation 15 billion years ago. The sizes and angular separations of these CMB fluctuations give an unique picture of the universe immediately after the Big Bang and have a large implication on our understanding of primordial cosmology. It is well known that the shape of this curve 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). For instance, the height of first peak is determined by the total energy density of the universe, while the third peak is related to the amount of dark matter. Thus, by fitting models of the CMB power spectrum for given values of the seven parameters, we can determine how the parameters influence the shape of the model spectrum. Examining those models that fit the data, we can then establish the ranges of the parameters that result in models which fit the data.

Additionally, we use a supernovae survey from [6]. The data set of [6] contains observations of type Ia supernovae, recording both the distance modulus (the observed luminosity minus the intrinsic luminosity), μ and redshift, z, for each supernova. The processes — and hence the in-

trinsic luminosities — governing type Ia supernovae are well known. (e.g. [5]). Assuming a homogeneous, isotropic and flat universe, the Robertson-Walker metric [11] predicts

$$\mu = 5 \log_{10} \left(\frac{c(1+z)\sqrt{\Omega_{\mathrm{M}}}}{100\sqrt{\omega_{\mathrm{DM}} + \omega_{\mathrm{B}}}} \int_0^z \frac{dt}{\sqrt{\Omega_{\mathrm{M}}(1+t)^3 + \Omega_\Lambda}} \right) + 25, \tag{4}$$

where c is the speed of light. Comparing the predicted distance moduli to the supernovae data of [6], we can make inferences about the true values of the associated cosmological parameters. Constraining all seven of these cosmological parameters is the focus of much recent effort in the astronomical community as these parameters describe the composition, age and eventual fate of the universe.

In this work, we use Fisher's method of combining p-values to compute 95% confidence regions for our astronomical problem. Computing expected observations given parameter vectors is quite fast for the supernovae data of [6] (using Equation 4), and hence we can quickly compute the p-values associated with the supernovae using χ^2 tests. However, computing the expected observations for the CMB data set is much more time consuming. Typically one employs a numerical solver, such as CMBFast to approximate the Boltzmann equation and yield the expected power spectrum. For this work, we use the database of 1.3 million p-values derived by [2] using confidence balls, a statistical procedure akin to χ^2 tests wherein the expected model is compared with a nonparametric fit of the observed data, rather than the data themselves, resulting in tighter confidence regions. However, since database contains only a million models, we had to limit the choice of candidates presented to the algorithm. To reduce the bias resulting from the non-uniform sampling within the CMB p-value database, we chose points uniformly at random from within the parameter space, and the snapped each point to the nearest model in the database.

Using these precomputed p-values reduced the sampling time for the CMB data set from several minutes to seconds, making the computational cost of sampling p-values from both the CMB and supernovae statistical tests nearly equivalent. Instead of using the kriging approximator to model the p-values directly, we modeled -2 times the log p-values, as many of the p-values were nearly zero, similar to the Gaussian case discussed earlier. Moreover, by approximating the values of -2 times the log p-values, the target function is exactly the sum of the two approximations.

In Figure 3 we depict confidence regions derived using only a single data set (those from [2] and from the supernovae data set of [6] using χ^2 tests), along with the regions obtained by including both data sets. Observe that while the improved confidence region is smaller than that previously reported, it is not a strict subset. Fisher's method combines the evidence from both tests, accounting for the effect of multiple hypothesis testing. A point rejected as being just outside the 95% confidence bound on a single test may be included when two tests are considered since the probability under the null distribution of exceeding the bound on one of two tests is greater than on one alone, explaining the increase of acceptable value of ω_{B} for small ω_{DM}. Similarly, points with p-values for both models are just above 0.05 will be rejected by Fisher's method, resulting in points accepted by both the CMB and supernovae tests individually being rejected by the combined test. The supernovae results strongly reject all models with high ω_{DM}, eliminating the second peak from the right-hand plot of Figure 3.

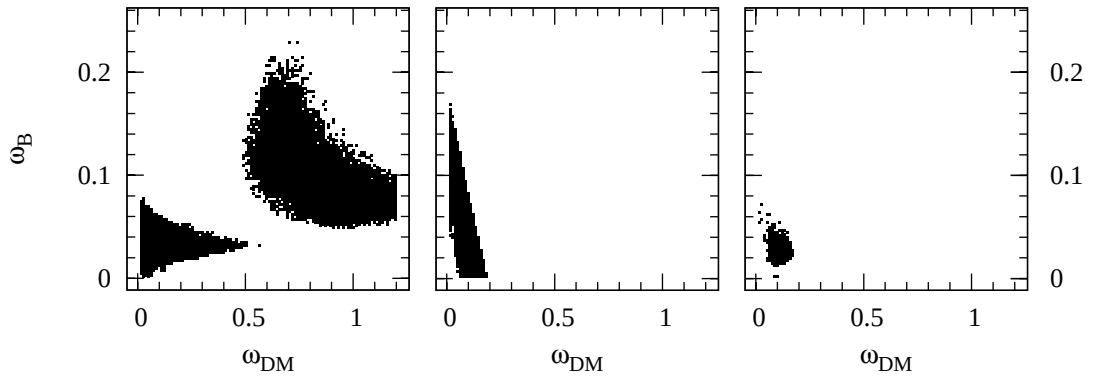

Figure 3: 95% confidence regions for ω_{B} as a function of ω_{DM} using just the CMB confidence ball p-values from [2] (left), the p-values from χ^2 tests on just the supernovae data of [6] (center) and combining the confidence ball and supernovae p-values after 30,000 samples drawn using the Variance-MaxVarStraddle heuristic (right).

5 Conclusions

We have described the problem of learning a hidden target function based on a set of related observable functions. We have developed an algorithm for locating the level set of this hidden target function while minimizing the number of experiments necessary. We described and showed how several different heuristics for choosing experiments from a set of candidates perform on synthetic target functions. Our experiments indicate that Variance-MaxVarStraddle outperforms both random and variance-weighted heuristics typically applied to active learning problem. Moreover, Variance-MaxVarStraddle is better than both the Sequential- and Variance-Straddle heuristics, as it appears to better approximate the information gain of a candidate point. We applied this algorithm to a seven dimensional joint analysis of cosmological parameters using two independent astronomical data sets. Using the method detailed in this work, we are able to efficiently compute the 95% confidence regions for the seven parameters constrained by both astronomical data sets. Using both the supernovae from [6] and WMAP data results in much different confidence regions than those reported in [2]; the supernovae data strongly rejects the second peak (at high ω_{DM}), while suggesting additional fits for low ω_{DM} and high ω_{B}. We are currently looking at applying this work to problems where the observable functions have different associated costs. This would have been the case, for instance, if we would have computed the CMB power spectra using CMBFast, rather than looking them up in a database.

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] 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.

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

[5] D. Morrison, et al. *Abell's Exploration of the Universe*. Saunders College Publishing, 7th edition, 1995.

[6] T. M. Davis, E. Mörtsell, J. Sollerman, A. C. Becker, S. Blondin, P. Challis, A. Clocchiatti, A. V. Filippenko, R. J. Foley, P. M. Garnavich, S. Jha, K. Krisciunas, R. P. Kirshner, B. Leibundgut, W. Li, T. Matheson, G. Miknaitis, G. Pignata, A. Rest, A. G. Riess, B. P. Schmidt, R. C. Smith, J. Spyromilio, C. W. Stubbs, N. B. Suntzeff, J. L. Tonry, W. M. Wood-Vasey, and A. Zenteno. Scrutinizing Exotic Cosmological Models Using ESSENCE Supernova Data Combined with Other Cosmological Probes. *Astrophysical Journal*, 666:716–725, September 2007.

[7] R.A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, London, 4 edition, 1932.

[8] L. V. Hedges. *Statistical Methods for Meta-Analysis*. Academic Press, 1985.

[9] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590, 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] H.P. Robertson. An interpretation of page's "new relativity". *Phyiscal Review*, 49(10):755, May 1936.

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon