Efficiently Computing Minimax Expected-Size Confidence Regions

Brent Bryan BRYANBA@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 USA

H. Brendan McMahan MCMAHAN@GOOGLE.COM
Google Pittsburgh, 4720 Forbes Avenue, Pittsburgh, PA 15213 USA

Chad M. Schafer CSCHAFER@STAT.CMU.EDU
Department of Statistics, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 USA

Jeff Schneider SCHNEIDE@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 USA

Abstract

Given observed data and a collection of parameterized candidate models, a $1-\alpha$ confidence region in parameter space provides useful insight as to those models which are a good fit to the data, all while keeping the probability of incorrect exclusion below α. With complex models, optimally precise procedures (those with small expected size) are, in practice, difficult to derive; one solution is the Minimax Expected-Size (MES) confidence procedure. The key computational problem of MES is computing a minimax equilibria to a certain zero-sum game. We show that this game is convex with bilinear payoffs, allowing us to apply any convex game solver, including linear programming. Exploiting the sparsity of the matrix, along with using fast linear programming software, allows us to compute approximate minimax expected-size confidence regions orders of magnitude faster than previously published methods. We test these approaches by estimating parameters for a cosmological model.

1. Introduction

Learning often requires the ability to compare hypothetical models with observed data to assess the models' validity. When the model is a function of several unknown parameters, we are interested in computing combinations of parameter values which fit the observed data well. While we could compute maximum likelihood estimates for the parameters, here we consider computing confidence intervals (or regions, when we take the parameters as an ensemble), as they provide more than just a point estimate for the parameter; confidence regions give us a range of parameter values which yield acceptable model fits to the data. They are especially useful for statistical inference in fields such as astronomy, biology, and geophyiscs, as multi-parameter models are common in these disciplines.

While there are many methods for computing $1-\alpha$ confidence regions, it is natural to prefer methods that produce small confidence regions. Conceptually, frequentist confidence procedures are defined before seeing any data. Hence, a frequentist procedure for computing the minimally-sized confidence region must minimize the size of the derived region for any possible realization of the data and any possible parameter setting. In general, it is impossible to find a minimally-sized region over all possible realizations of the data for a given parameter θ; however, the expected size of a region can be ascertained for any θ. Evans et al. (2005) show that the $1-\alpha$ confidence procedure which minimizes the maximum (worst-case) expected size of the confidence region is the inversion of a family of level α tests of simple nulls versus a single simple alternative; in this paper we will assume that "size" refers to Euclidean volume. Schafer and Stark (2006) describe an algorithm for approximately deriving these confidence regions using Monte Carlo sampling.

In this work we extend the work of Schafer and Stark (2003; 2006) in three ways. First we show

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

that the Monte Carlo method for computing minimax expected-size (MES) confidence regions can be compactly represented as a convex game. Second, we note that the associated game matrix has many values very close to zero. Exploiting this approximate sparsity results in both memory and computational savings. Finally, the convex game representation lends itself to a variety of solution techniques, notably fictitious play and linear program solvers. By utilizing these solvers, we are able to speed up the computation of MES confidence regions by at least two orders of magnitude, as demonstrated by our application of the MES confidence procedure to an astronomical data set. There are two ways in which this speedup is critical. First, it allows for the solution of larger games, allowing for more accurate approximations using additional Monte Carlo samples. Second, it allows for the testing of a larger collection of models, which leads to better resolution of the confidence region boundary.

1.1. A Motivating Problem from Astronomy

Throughout this paper we consider the task of computing the MES confidence region for cosmological parameters, using the Supernova Legacy Survey (SNLS) data set (Astier, P., et al., 2006). The SNLS data set 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 governing type Ia supernovae are well known, and hence so are their intrinsic luminosities (e.g. Morrison et al. (1995)). Assuming a homogeneous, isotropic and flat universe, the Robertson-Walker metric (Robertson, 1936) predicts

$$d_L = \frac{c(1+z)}{H_0} \int_0^z \frac{dt}{\sqrt{\Omega_{\mathrm{M}}(1+t)^3 + \Omega_\Lambda}}, \qquad (1)$$

where the luminosity distance d_L is given by

$$\mu = 5 \log_{10}(d_L) + 25, \qquad (2)$$

where c is the speed of light, H_0 is the Hubble constant (the recession speed of local galaxies), and Ω_{M} and Ω_Λ are the fraction of matter and dark energy in the universe, respectively. Comparing the distance moduli models predicted by combining Equations 1 and 2 to the SNLS data allows us to make inferences about the true values of the parameters $\theta = (H_0, \Omega_{\mathrm{M}}, \Omega_\Lambda)$. Constraining 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.

While we focus on the SNLS data set throughout this work, we stress that the ideas and techniques developed here can be applied to computing MES confidence regions in many situations. Before introducing the MES procedure, we discuss necessary background from game theory as well as the advantages of the MES procedure over other confidence procedures.

1.2. Matrix Games and Convex Games

A zero-sum matrix (normal-form) game is played by two players, player row with strategies $R = \{1, \ldots, m\}$ and player column with strategies $C = \{1, \ldots, n\}$. A $m \times n$ matrix $\mathbf{A}$ specifies the payoffs. If row plays strategy $i \in R$ and column plays $j \in C$, the payment from column to row is the $(i,j)th$ entry of $\mathbf{A}$, denoted a_{ij}. The players select their strategies simultaneously, without knowledge of the other player's choice.

We use $\Delta(\cdot)$ to denote the probability simplex over a finite set; for example $\Delta(R) = \{\mathbf{y} \in R^m \mid \sum_{i=1}^m y_i = 1 \text{ and } y_i \geq 0\}$. A mixed strategy is an element $\mathbf{y} \in \Delta(R)$ for the row player or $\mathbf{z} \in \Delta(C)$ for the column player, corresponding to a distribution over the rows or columns, respectively. If the players select mixed strategies $\mathbf{y}$ and $\mathbf{z}$, the expected payoff $V(\mathbf{y}, \mathbf{z})$ from column to row is given by the bilinear form $\mathbf{y}^T \mathbf{A} \mathbf{z}$. A solution to the game is a minimax equilibrium $(\mathbf{y}^*, \mathbf{z}^*)$, a pair of strategies such that neither player has an incentive to play differently given that the other player selects their strategy from the pair. The minimax theorem states that if the players are allowed to select mixed strategies, there is no advantage to playing second:

$$\max_{\mathbf{y} \in \Delta(R)} \min_{\mathbf{z} \in \Delta(C)} \mathbf{y}^T \mathbf{A} \mathbf{z} = \min_{\mathbf{z} \in \Delta(C)} \max_{\mathbf{y} \in \Delta(R)} \mathbf{y}^T \mathbf{A} \mathbf{z}. \qquad (3)$$

Thus, solving either the min max or max min optimization problem from (3) results in a minimax equilibrium for the game. This problem can easily be converted to a linear program and solved via standard techniques.

An ϵ-approximate minimax equilibrium for a matrix game is a pair of strategies $(\mathbf{y}', \mathbf{z}')$ where neither player can gain more than ϵ value by switching to some other strategy. That is,

$$V(\mathbf{y}', \mathbf{z}') \geq \max_{\mathbf{y} \in \Delta(R)} V(\mathbf{y}, \mathbf{z}') - \epsilon \qquad (4)$$

$$V(\mathbf{y}', \mathbf{z}') \leq \min_{\mathbf{z} \in \Delta(C)} V(\mathbf{y}', \mathbf{z}) + \epsilon. \qquad (5)$$

If $\epsilon = 0$, we have an exact minimax equilibrium.

Convex Games Two-player zero-sum bilinear-payoff convex games (simply "convex games" for the sequel) are a natural generalization of matrix games. [1]

[1] Our convex games are non-cooperative, and are unrelated to the super-modular coalitional games often called

Convex games allow arbitrary convex sets Y and Z in place of the probability simplexes $\Delta(R)$ and $\Delta(C)$ for matrix games. A convex game is specified by a tuple $(Y, Z, \mathbf{A})$ where $Y \subseteq \mathbb{R}^m$ and $Z \subseteq \mathbb{R}^n$ are the strategy sets for the two players, and $\mathbf{A}$ is a $m \times n$ payoff matrix. The first player (who we will call y) selects an action $\mathbf{y} \in Y$, the second player (called z) simultaneously chooses $\mathbf{z} \in Z$, and the payoff from player y to player z is given by $V(\mathbf{y}, \mathbf{z}) = \mathbf{y}^T \mathbf{A} \mathbf{z}$. The concepts of equilibria and ϵ-approximate equilibria naturally generalize to convex games, and it can be shown that the minimax theorem still holds (McMahan, 2006).[2]

While convex games are a simple generalization of matrix games, the ability to represent arbitrary convex strategy sets lets us take advantage of structure in many types of games, often yielding exponentially smaller representations. Notable examples include cost-paired Markov decision process games, extensive-form games (including poker), and the problem of computing an optimal oblivious routing (McMahan, 2006). Moreover, the key computational problem for our statistical approach can be formulated as a convex game.

Fictitious play (FP) is a classic algorithm for solving zero-sum matrix games, and generalizes to convex games. The FP algorithm simulates play of the game; on each iteration both players select the action which is a best response to the average of the opponent's past actions. Standard, or synchronous, fictitious play (SFP) executes the updates independently in parallel for each player. Asynchronous fictitious play (AFP) does updates first for (say) y, and then for z, using the new average strategy computed for y. As demonstrated in Section 4, AFP can be significantly faster.

A polyhedron is a convex set defined by a finite number of linear equality and inequality constraints. We say a convex game is polyhedral if Y and Z are polyhedra. Polyhedral convex games can be solved in polynomial time via linear programming, as shown by (Koller et al., 1994) in the context of extensive-form games. In this work, we use the CPLEX 10.0 commercial optimization package to solve the linear program, which proves to be very effective. However, for very high dimensional and extremely sparse convex games like poker, the single or double oracle algorithms of (McMahan & Gordon, 2007) can be orders of magnitude faster than standard linear programming approaches. We ran preliminary experiments with these algorithms; for generating approximate solutions, we were sometimes able to significantly outperform FP. However, the run times for a reasonable approximation using these algorithms were such that directly solving the linear program with CPLEX was preferable. For this reason, we do not report further on these results.

convex games in the cooperative game theory literature.

[2]Some mild technical assumptions are required.

1.3. Confidence Regions

A confidence procedure maps the observed data into a subset of the set of all possible model parameter settings. We say that a confidence procedure has coverage probability $1-\alpha$ if, regardless of the truth $\theta^\star$, the probability that confidence region includes $\theta^\star$ is at least $1-\alpha$. Optimal confidence procedures are generally only known for simple parametric models. With the complex models used in most scientific applications, general procedures for forming confidence regions are desirable. Such procedures include χ^2 tests (e.g. Wasserman (2004)), confidence balls (Genovese, C. et al., 2004; Bryan, B., et al., 2005), and Markov Chain Monte Carlo (MCMC) (e.g. Wasserman (2004)). However, such procedures often have low power, and hence low precision.

Often, χ^2 tests are used to compute $1-\alpha$ confidence regions, due to their simplicity both in implementation and interpretation. In their simplest form, one defines parameter ranges to be searched and then exhaustively iterates over this space, computing test models, assessing the fit of the data to the model, typically in the form of the sums of squared difference between the observed and expected data values. This is then compared to the $\chi^2_{(n)}$ distribution to determine the significance level of the proposed parameter vectors. However, it is well known that such an approach is conservative (Wasserman, 2004).

Alternatively, Genovese, C. et al. (2004) propose the idea of confidence balls, constructed by first fitting the observed data non-parametrically and then comparing a proposed model to this non-parametric fit. Intuitively, confidence balls correct for the power loss of χ^2 tests by mimicking the underlying function with the non-parametric fit, thereby reducing the noise inherent in the data. However, computing the radius of the confidence ball is non-trivial, as it relies on both the fit to the data as well as the observational error.

Bayesian approaches are also possible. One can use MCMC (using the Metropolis Hasting algorithm or Gibbs Sampling) to approximate the posterior and use this posterior to derive $1-\alpha$ credible regions for the parameters. 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 in-

tervals may trap the true value of the parameter close to zero percent of the time (Wasserman, 2004).

The MES confidence procedure allows us to efficiently find confidence regions that have both optimal precision and correct coverage. Here, we compute MES confidence regions for the SNLS data; Schafer and Stark (2003; 2006) show examples of synthetic and real data sets where MES out-performs alternative techniques.

2. MES Confidence Regions

Let $\Theta \subseteq \mathbb{R}^p$ denote the set of possible model parameter settings, and let θ and $\tilde{\theta}$ be arbitrary values of Θ. For each $\theta \in \Theta$, there is a distribution P_θ on the space of possible observations $\mathcal{X} \subseteq \mathbb{R}^m$. Let X be a random variable and x be a generic observation of X. Assume each distribution P_θ has a density $f(x|\theta)$ relative to Lebesgue measure. We are interested in constructing a confidence region for the true value of the parameter, denoted $\theta^\star$, based on the observation that $X = x$ and the *a priori* constraint that $\theta^\star \in \Theta$.

Consider testing the hypothesis that $\theta^\star = \tilde{\theta}$ at level α for some arbitrary $\tilde{\theta} \in \Theta$. The associated acceptance region for the test, $A(\tilde{\theta}) \subset \mathcal{X}$, is the set of values for which the test will not reject the hypothesis $\theta^\star = \tilde{\theta}$. Since we are interested in tests with significance level α, we require $P_{\tilde{\theta}}(X \in A(\tilde{\theta})) \geq 1 - \alpha$.

The power of the test is the probability that the test rejects the hypothesis that $\theta^\star = \tilde{\theta}$. Define the power function as

$$\beta(\theta, \tilde{\theta}) \equiv 1 - P_\theta(X \in A(\tilde{\theta})).$$

The test has significance level α, so $\beta(\tilde{\theta}, \tilde{\theta}) \leq \alpha$. We are interested in small (i.e., precise) confidence regions, and we see below that this leads us to choose $A(\tilde{\theta})$ to maximize $\beta(\theta, \tilde{\theta})$ over all θ subject to $\beta(\tilde{\theta}, \tilde{\theta}) \leq \alpha$.

The above can be repeated for all $\tilde{\theta} \in \Theta$, and the result is a family of acceptance regions $A(\tilde{\theta})$. These can be inverted into a confidence procedure by defining a rule

$$d(\tilde{\theta}, x) = \begin{cases} 1, & \text{if } x \in A(\tilde{\theta}) \\ 0, & \text{if } x \notin A(\tilde{\theta}) \end{cases} .$$

Thus, we can either discuss the choice of $A(\tilde{\theta})$ for all $\tilde{\theta}$, or the choice of d; in what follows it will be more natural to think of selecting d. We can use the rule d to construct a $1-\alpha$ confidence region, $C_d(x)$, for $\theta^\star$ based on the observed data x:

$$C_d(x) \equiv \{\tilde{\theta} \in \Theta : d(\tilde{\theta}, x) = 1\}.$$

Ideally, $C_d(x)$ would be small, since a small confidence set implies high precision in the estimate. Define $\nu(C_d(x))$ to be the size of $C_d(x)$ using a measure ν. In this paper we will assume ν is a Euclidean measure over the parameter space, although other choices could be justified. Choosing d to minimize $\nu(C_d(x))$ for fixed data x is trivial: simply define $d(\tilde{\theta}, x) = 0$ for all $\tilde{\theta}$. This is equivalent to "data snooping" and is not statistically valid. Instead, we seek to choose d to make the expected size of the confidence region ($\mathbf{E}_\theta[\nu(C_d(X))]$) small for all possible truths $\theta \in \Theta$.

Unfortunately, it is not usually true that a single d simultaneously minimizes $\mathbf{E}_\theta[\nu(C_d(X))]$ over all θ. Instead, we consider minimizing the weighted average

$$\mathcal{S}(\pi, d) \equiv \int_\Theta \mathbf{E}_\theta[\nu(C_d(X))]\pi(d\theta), \tag{6}$$

with the weighting provided by a probability measure π defined on Θ. Pratt's theorem (Pratt, 1961) gives

$$\mathbf{E}_\theta[\nu(C_d(X))] = \int_\Theta (1 - \beta(\theta, \tilde{\theta}))\, \nu(d\tilde{\theta}).$$

This link between expected size and power allows us to apply the classic Neyman-Pearson lemma (Neyman & Pearson, 1933) to find the d that minimizes Equation 6: set $d(\tilde{\theta}, x) = 1$ if and only if $T_\pi(\tilde{\theta}, x) \leq c_{\tilde{\theta}}$ where

$$T_\pi(\tilde{\theta}, x) \equiv \frac{\int_\Theta f(x|\theta)\pi(d\theta)}{f(x|\tilde{\theta})} \tag{7}$$

and $c_{\tilde{\theta}}$ is a cutoff chosen large enough to ensure d has $1-\alpha$ coverage. Call this confidence procedure d_π. We now address the question of how to choose π.

2.1. The Convex Game

The selection of the measure π is subjective, but there is a particular choice which can be justified using statistical decision theory. Let π_0 be the π that maximizes $\mathcal{S}(\pi, d_\pi)$. The key result (Evans et al., 2005) is that d_{π_0} minimizes the maximum expected size over $\theta \in \Theta$. In other words, for any possible truth $\theta \in \Theta$,

$$\mathbf{E}_\theta[\nu(C_{d_{\pi_0}}(X))] \leq \mathcal{S}(\pi_0, d_{\pi_0}),$$

while for all π' (and hence $d_{\pi'}$), there is some $\theta' \in \Theta$ such that

$$\mathbf{E}_{\theta'}[\nu(C_{d_{\pi'}}(X))] \geq \mathcal{S}(\pi_0, d_{\pi_0})$$

This is analogous to the minimax equilibrium of the convex game constructed next.

In general, calculating π_0 is intractable. However, we can approximate it using finite sampling. Note that

$$\begin{aligned} &\int_\Theta \mathbf{E}_\theta[\nu(C_d(X))]\pi(d\theta) \\ &\quad = \int_\Theta \int_X \nu(C_d(x)) f(x|\theta)\, dx\, \pi(d\theta) \\ &\quad = \int_\Theta \int_X \int_\Theta d(\tilde{\theta}, x) f(x|\theta)\, \nu(d\tilde{\theta})\, dx\, \pi(d\theta). \end{aligned}$$

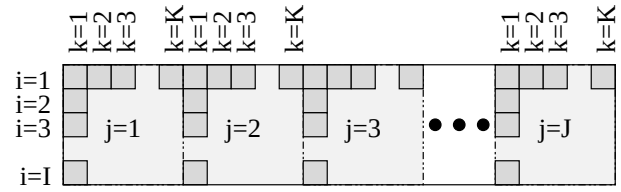

Figure 1. The payoff matrix A is a concatenation of J sub-matrices, where each sub-matrix corresponds to a single sample point. Rows of the sub-matrices correspond to the same I prior points, while the columns correspond to K different simulations of the sub-matrix sample j.

We will approximate the integrals in the previous equation with finite samples. For instance,

$$\int_\Theta \mathbf{E}_\theta[\nu(C_d(X))]\pi(d\theta) \approx \sum_{i=1}^{I} \mathbf{E}_{\theta_i}[\nu(C_d(X))]\pi(d\theta_i)$$

where $\theta_1, \theta_2, \ldots, \theta_I$ are chosen uniformly from Θ. Next, we approximate the integral $\nu(d\tilde{\theta})$ as the sum over values $\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_J$ sampled uniformly from Θ. Finally, for each $\tilde{\theta}_j$, a sample of K data values is simulated from distribution $P_{\tilde{\theta}_j}$ and labeled $x_{j1}, x_{j2}, \ldots, x_{jK}$; these are used in a Monte Carlo approximation to the integral against $f(x|\tilde{\theta}_j)$. Thus,

$$\begin{aligned}\int_\Theta \mathbf{E}_\theta[\nu(C_d(X))]\pi(d\theta) \qquad (8)\\ \approx \frac{1}{JK}\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i=1}^{I} d(\tilde{\theta}_j, x_{jk})\left(\frac{f(x_{jk}|\theta_i)}{f(x_{jk}|\tilde{\theta}_j)}\right)\pi(\theta_i),\end{aligned}$$

where $x_{j1}, x_{j2}, \ldots, x_{jK} \sim P_{\tilde{\theta}_j}$. Finally, Equation 8 can be written compactly as

$$\int_\Theta \mathbf{E}_\theta[\nu(C_d(X))] \approx \frac{1}{JK}\pi^T \mathbf{A}\mathbf{d}, \qquad (9)$$

where

$$\begin{aligned}\mathbf{d}^T &= [\mathbf{d}_1^T, \mathbf{d}_2^T, \ldots, \mathbf{d}_J^T]\\ \mathbf{d}_j^T &= [d(\tilde{\theta}_j, x_{j1}), d(\tilde{\theta}_j, x_{j2}), \ldots, d(\tilde{\theta}_j, x_{jK})]\\ \pi^T &= [\pi(\theta_1), \pi(\theta_2), \ldots, \pi(\theta_I)]\end{aligned}$$

and the matrix $\mathbf{A}$ has elements given by

$$a_{i\ell} = \frac{f(x_{jk}|\theta_i)}{f(x_{jk}|\tilde{\theta}_j)} \qquad (10)$$

where $\ell = (j-1)K + k$. $\mathbf{A}$ can be viewed as the concatenation of J $I \times K$ matrices, as shown in Figure 1.

Equation 9 denotes a convex game. The first player, y, has strategy set

$$Y = \Delta(\{\theta_i \mid 1 \le i \le I\}),$$

that is, a vector $\pi = \mathbf{y} \in Y$ is simply a probability distribution over the finite set of samples θ_i. The second player, z, has more complex constraints. Each subvector of $\mathbf{d}$, $\mathbf{d}_j$, must define a decision rule which gives (approximate) probability 1–α to accepting $\tilde{\theta}_j$ into the confidence region when $\theta^\star = \tilde{\theta}_j$. This is guaranteed by requiring that the entries of $\mathbf{d}_j$ sum to $K(1-\alpha)$. We can fully represent the set of allowed $\mathbf{d}_j$ as the polyhedra $\mathbf{D}_j$ using the linear constraints

$$\begin{aligned}\mathbf{1}\mathbf{d}_j &= K(1-\alpha)\\ (\forall k) \quad 0 &\le d_{jk} \le 1\end{aligned}$$

where $\mathbf{1}$ is the vector of all 1s. Also accounting for the normalizing $1/JK$, we define the convex strategy set

$$Z = \left\{ \frac{1}{JK(1-\alpha)}\langle \mathbf{d}_1, \ldots, \mathbf{d}_J\rangle \in \mathbb{R}^{JK} \mid \mathbf{d}_j \in \mathbf{D}_j \right\}.$$

Thus, we have the convex game $\mathcal{G} = (Y, Z, A)$. The "nature" player, y, chooses a vector $\mathbf{y}$ corresponding to π, and the "statistician" player, z, chooses $\mathbf{z}$ corresponding to $\mathbf{d}$. For fixed $\pi(= \mathbf{y})$, the statistician knows the ideal strategy formed by finding d_π and using it to find the entries of $\mathbf{d}$ (and hence $\mathbf{z}$). Moreover, the statistician assumes that nature plays a best response to $\mathbf{d}$; that is, nature maximizes Equation 9 with respect to $\mathbf{d}$. This is equivalent to assuming nature chooses π_0. The statistician's minimax strategy is hence $\mathbf{d}_{\pi_0}$.

Bounds on the value of $\mathcal{G}$ are bounds on the maximum expected size of the confidence region. An approximate equilibria is useful, as it gives a bound on the value of the game. The result is a strategy for nature π which may not be exactly minimax optimal, but still will greatly reduce the maximum expected size of the confidence region relative to standard approaches. This value of π is then used in the underlying hypothesis tests (Equation 7). As each test has level α, the derived confidence regions will have $1-\alpha$ coverage probability, even with the discrete sampling approximations.

2.2. Constructing the Estimate

Given a solution to $\mathcal{G}$, we wish to approximate the confidence region $C_{d_{\pi_0}}(x)$ utilizing our observed data x. While more detail is provided in Schafer and Stark (2006), the result of solving the above matrix game yields a collection of parameter values $\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_J$ for which the cutoffs $c_{\tilde{\theta}_j}$ are well-approximated. It is trivial to calculate $d_{\pi_0}(\tilde{\theta}_j, x)$ for each j. The accepted parameter values form the confidence region. Provided J is sufficiently large, the accepted parameter values will resolve the confidence region boundary.

In the SNLS application, X has the multivariate normal distribution with mean $\mathbf{m}(\theta^\star)$ and known variance, where $\mathbf{m} = \{m_1(\theta^\star), m_2(\theta^\star), \ldots, m_D(\theta^\star)\}$ is the vector of predicted distance moduli of the D supernova using model m with the parameter $\theta^\star$ (Equations 1 & 2). Thus we have, for instance,

$$f(x|\theta_i) = \prod_{d=1}^{D} \frac{1}{\sigma_d\sqrt{2\pi}} \exp\left\{-\frac{(x_d - m_d(\theta_i))^2}{2\sigma_d^2}\right\}$$

$f(\cdot|\tilde{\theta}_j)$ has a similar form. Moreover, $x_{jkd} = m_d(\tilde{\theta}_j) + \sigma_d\varepsilon_d$ where $\varepsilon_d \sim N(0,1)$ for all D supernova. Hence,

$$\frac{f(x|\theta_i)}{f(x|\tilde{\theta}_j)} = \exp\left\{\frac{1}{2}\sum_{d=1}^{D}\left[\varepsilon_d^2 - \left(\frac{x_d - m_d(\theta_i)}{\sigma_d}\right)^2\right]\right\}. \tag{11}$$

3. Solving the Convex MES Game

In this section, we describe how to exploit approximate sparsity and other properties of the game $\mathcal{G}$ in order to speed up the equilibria computation.

All matrix entries (defined by Equation 10) will be greater than zero, but many values will be very close to zero because the parameter space Θ results in the majority of the model pairs being largely dis-similar. We can exploit this property to approximately solve $\mathcal{G}$ much more quickly. The size of the linear program we solve is dominated by the non-zero entries in $\mathbf{A}$. For fictitious play, computation is dominated by computing the products $\mathbf{y}^T A$ and $A\mathbf{z}$. Once these products have been computed, simple algorithms implement the best response oracles with running times approximately linear in the number of non-zeros of the resulting vectors. Thus, using a sparser matrix significantly helps both algorithms. We construct a new matrix $\tilde{\mathbf{A}}$ by taking all entries $a_{i\ell} \leq \epsilon_{\mathrm{t}}$ and setting them to zero in $\tilde{\mathbf{A}}$. Approximately solving the approximated game $\tilde{\mathcal{G}} = (Y, Z, \tilde{\mathbf{A}})$ gives an approximate minimax equilibria for the original game $\mathcal{G}$:

Theorem 1. *Let $\mathcal{G} = (Y, Z, \mathbf{A})$ be a confidence region game. Let $\tilde{\mathbf{A}}$ be a matrix such that $0 \leq a_{i\ell} - \tilde{a}_{i\ell} \leq \epsilon_t$ for all entries (i, l), and let $(\tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ be an ϵ_a-approximate minimax equilibria to the convex game $(Y, Z, \tilde{\mathbf{A}})$. Then, $(\tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ is an $(\epsilon_t + \epsilon_a)$-approximate equilibria for the original game $(Y, Z, \mathbf{A})$.*

Sketch of proof: The key is $(\forall \mathbf{y} \in Y, \forall \mathbf{z} \in Z)$, $|\mathbf{y}|_1 = |\mathbf{z}|_1 = 1$, and so $0 \leq \mathbf{y}^T\mathbf{A}\mathbf{z} - \mathbf{y}^T\tilde{\mathbf{A}}\mathbf{z} \leq \epsilon_{\mathrm{t}}$. The theorem then follows from an application of the definition of approximate minimax equilibrium in $\tilde{\mathbf{A}}$, and then manipulation of the resulting inequalities. □

When we approximately solve $\tilde{\mathcal{G}}$ we introduce approximation error in two ways: both by finding an ϵ_{a}-approximate solution and by solving an ϵ_{t}-approximate game. Depending on the problem and the algorithm at hand, we can trade off these two sources of error for a fixed total error $\epsilon = \epsilon_{\mathrm{t}} + \epsilon_{\mathrm{a}}$. We fixed $\epsilon_{\mathrm{t}} = 1 \times 10^{-4}$ for the experiments we report, which results in a matrix that is 96% sparse; but even using $\epsilon_{\mathrm{t}} = 1 \times 10^{-32}$ (on the order of machine precision) results in a sparsity of 85%. [3]

Efficiently building the game matrix While, we could use Equation 11 to compute the values of the game matrix, recall that $x_{jkd} = m_d(\tilde{\theta}_j) + \varepsilon_d\sigma_d$, where $\varepsilon_d \sim N(0,1)$. Thus, we can rewrite Equation 11 as

$$a_{i\ell} \quad = \quad \exp\left\{\sum_{d=1}^{D} v\varepsilon_{dk} - \sum_{d=1}^{D}\frac{v^2}{2}\right\},$$

where $l = (j-1)K + k$ and v is a vector with elements $(m_d(\theta_i) - m_d(\tilde{\theta}_j))/\sigma_d)$. The vector v dependents only on i and j, the index of the row, and the sub-matrix. We only need to compute this term for each $\tilde{\theta}_j$ in each sub-matrix once. Moreover, ε_{dk} is the same for all entries of a particular column. Thus, by computing the max elements of ε_{dk} and v, we can bound the maximum magnitude of $a_{i\ell}$ for all i. Comparing this bound with the log of the zero cutoff value defined in Section 3, we can determine whether we can set $a_{i\ell}$ to zero without further computation. We find that by pruning those entries with maximal values less than the log of the zero cutoff value speeds up the matrix build process in proportion to the sparsity; for the SNLS problem we observe a factor of 5 speedup.

Generalized dominance Player $\mathbf{z}$'s strategy set is highly constrained, and this makes it possible to prove that many of player $\mathbf{y}$'s pure strategies (rows of $\mathbf{A}$) cannot appear in a minimax solution. Let $\mathbf{y}(i) \in Y$ be the pure strategy that always plays row $i \in \{1, \ldots, I\}$. If there exists a row $j \neq i$ such that

$$(\forall \mathbf{z} \in Z) \qquad \mathbf{y}(j)^T\mathbf{A}\mathbf{z} \;\geq \mathbf{y}(i)^T\mathbf{A}\mathbf{z}, \tag{12}$$

then there exists a minimax solution that never plays row i. We say j dominates i if Equation 12 holds. If j dominates i, for any strategy $\mathbf{y}$ that sometimes plays i, the strategy $\mathbf{y}_{i \leftarrow j}$ that plays j every time $\mathbf{y}$ plays i must do at least as well against all opponent strategies. Checking Equation 12 over all pairs of strategies would be prohibitive. Instead, we find

$$\mathrm{lb} = \max_{1 \leq i \leq I} \min_{\mathbf{z} \in Z} \quad \mathbf{y}(i)^T\mathbf{A}\mathbf{z},$$

[3] Some convex games have much greater sparsity. For example, a payoff matrix for the poker game Rhode Island Hold'em is 99.994% sparse.

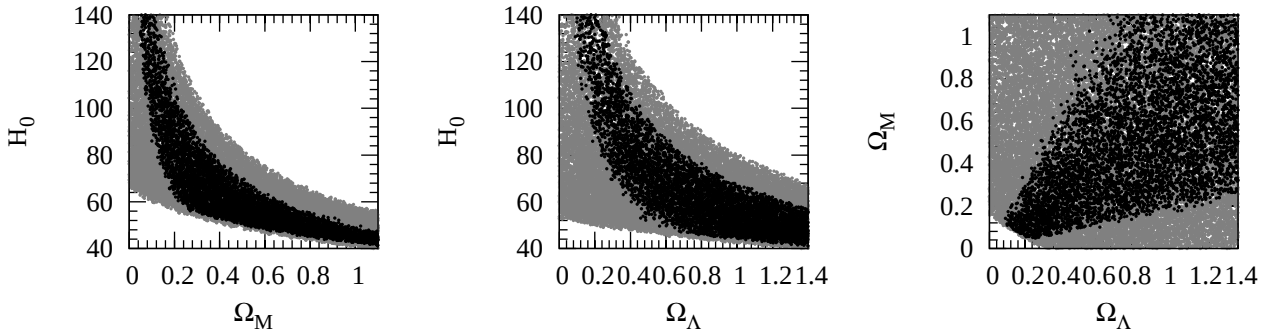

Figure 2. 2D projections of the derived MES confidence region for three cosmological parameters (H_0, Ω_{M}, and Ω_Λ) based on the the Supernova Legacy Survey data set. Points in black denote models that could not be rejected by the MES confidence procedure at the 95% confidence level, while points in grey depict those that cannot be excluded by a χ^2 test at the 95% confidence level. The figure shows that MES has significantly more power than the χ^2 test.

which can be computed by preforming I best-response calculations. Then, for each i we check whether

$$\max_{\mathbf{z}\in Z} \quad \mathbf{y}(i)^T \mathbf{A}\mathbf{z} \quad < \quad \mathrm{lb} \tag{13}$$

We can evaluate this expression by performing I "worst-response" calculations, which takes time approximately linear in the number of non-zeros in $\mathbf{A}$. If Equation 13 holds for some row i, then that row is dominated and can be removed. Running on different instances of the our example domain, we were able to eliminate from 20% to 60% of the rows in $\mathbf{A}$ in this manner. However, the overhead of computing this dominance approximately canceled out the speedup in the solution to the linear program. Nevertheless, this technique can be used as a preprocessing step for any convex game algorithm and we expect that on some domains improvements could be substantial. Further, more thorough direct checking of Equation 12 with respect to a small, diverse set of "good" strategies—perhaps the bundle maintained by the single oracle algorithm of (McMahan et al., 2003)—could have substantial benefit.

4. Experimental Results

Using the procedure detailed in Section 2, we have computed MES confidence regions for the SNLS data set, restricting Θ to the set of parameters with $40 \leq H_0 \leq 140$, $0 \leq \Omega_{\mathrm{M}} \leq 1.1$ and $0 \leq \Omega_\Lambda \leq 1.4$ that result in models with χ^2 values less than 200. The χ^2 cut introduces a small (2×10^{-14}) probability of eliminating the true value $\theta^\star$; we can correct for this cut using a Bonferroni correction ensuring that our derived confidence region maintains the desired $1{-}\alpha$ coverage. This restriction is natural as it limits the MES procedure to considering parameter vectors that are at least minimally supported by the data. While we could perform the MES procedure without the χ^2 restriction, the MES procedure would be forced to compute the minimax expected size over even those parameter values which are extremely unlikely given the data, resulting in an extremely conservative estimate. Figure 2 shows MES and χ^2 confidence regions; the MES procedure has significantly more power.

The experiments reported in Schafer and Stark (2006) used SFP on the dense game matrix $\mathbf{A}$; we use this approach as a baseline against which to compare our methods. We observed that the initial strategies chosen for FP are somewhat important; we initialize both players to the uniform strategy.[4] Since FP is an iterative algorithm, we can imagine stopping the algorithm as soon as a specified error ratio has been met. For this work we consider relative error, which we define to be the error of the MES confidence region (the difference between the upper and lower bounds on the value of the convex game) relative to the lower bound on the optimal size of the MES region. Figure 3 displays the average speedups, derived over 10 trials, obtained for 3 algorithms over SFP when solving for MES confidence regions. The exact speedup depends on the random samples drawn. We used a fixed $\epsilon_{\mathrm{t}} = 1 \times 10^{-4}$ for all experiments (except for the baseline, which used the dense matrix). CPLEX took 61 seconds to solve this game. For CPLEX, $\epsilon_{\mathrm{a}} = 0$ and so the total error introduced was due to ϵ_{t}. This absolute additive error of 0.0001 resulted in a solution with a relative error of 0.4%; for the FP implementations, we ran four experiments with relative total error stopping criteria of 20%, 10%, 5%, and 1%. Since the error due to ϵ_{t} was

[4]When the column player is initialized to the best response to a uniform strategy for the row player, the initial bounds on the game value are good, but it takes many iterations before the bounds improve. Starting with the uniform distribution produces worse bounds initially, but better improvements quickly overcome the initial advantage of the best response initialization.

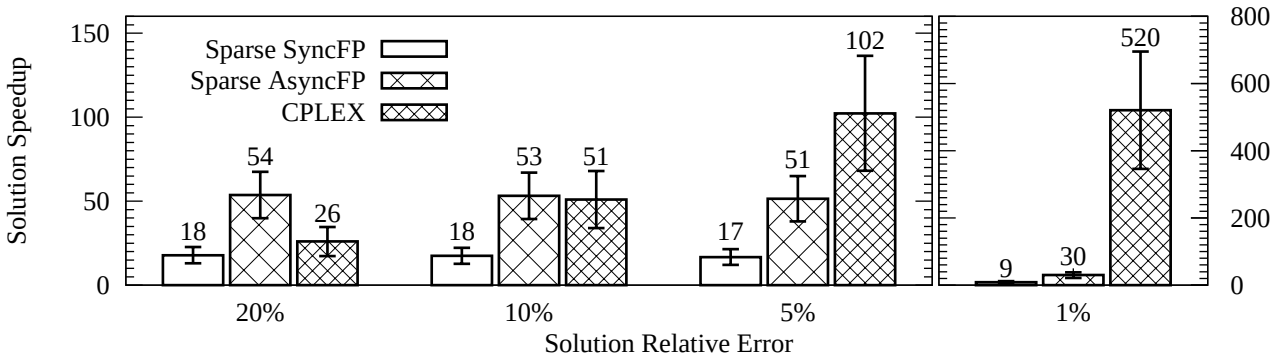

Figure 3. Convex game solution speedups obtained by the respective algorithms over synchronous fictitious play (SFP) using a dense matrix representation. The game matrix was composed of $I = 500$ rows and $JK = 1000 \times 200$ columns. Fictitious play (FP) is an iterative algorithm and can be terminated as soon as a certain relative error threshold has been met, while CPLEX solves the convex game exactly, up to the additive error ϵ_t induced by the sparse representation. Thus, the relative speedup of CPLEX over FP algorithms increases as the relative error decreases, as CPLEX's solve time is fixed while FP requires additional time to achieve lower relative errors. All of the bars for CPLEX correspond to a fixed run time of 61 seconds; SFP took 8.8 hours to solve the same game to a relative error of 1%.

fixed for all of these runs, smaller total relative errors are achieved by running more iterations of FP. Thus, for all of the sets of columns in the figure, CPLEX is producing a higher-quality solution than FP. For example, the right hand set of results shows CPLEX generating a solution with relative error 0.4% over 500 times faster than FP generates a solution with relative error 1%; using CPLEX to solve the linear program becomes advantageous if we desire to find solutions with relative error rates less than $\sim 10\%$.

5. Conclusions

We have shown that MES confidence region methods can be formulated as a compact convex game with a sparse game matrix, allowing it to be solved by a variety of convex game and linear program solvers. Using this representation, we can solve the convex game for a real-world astronomy problem over 500 times faster than previously proposed methods. We stress, however, that the algorithms presented here can be applied to any suitable data set.

References

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

Bryan, B., et al. (2005). Active learning for identifying function threshold boundaries. In *Advances in neural information processing systems 18*. Cambridge, MA: MIT Press.

Evans, S. N., Hansen, B. B., & Stark, P. B. (2005). Minimax expected measure confidence sets for restricted location parameters. *Bernoulli*, *11*, 571–590.

Genovese, C. et al. (2004). Nonparametric inference for the cosmic microwave background. *Statistical Science*, *19*, 308–321.

Koller, D., Megiddo, N., & von Stengel, B. (1994). Fast algorithms for finding randomized strategies in game trees. *STOC*.

McMahan, H. B. (2006). *Robust planning in domains with stochastic outcomes, adversaries, and partial observability*. Doctoral dissertation, Carnegie Mellon University.

McMahan, H. B., & Gordon, G. J. (2007). A fast bundle-based anytime algorithm for poker and other convex games. *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

McMahan, H. B., Gordon, G. J., & Blum, A. (2003). Planning in the presence of cost functions controlled by an adversary. *ICML 2003*.

Morrison, D., Wolff, S., & Fraknoi, A. (1995). *Abell's exploration of the universe*. Saunders College Publishing. 7th edition.

Neyman, J., & Pearson, K. (1933). On the problem of the most efficient test of statistical hypotheses. *Phil. Trans. of Royal Soc. of London*, *231*, 289–337.

Pratt, J. W. (1961). Length of confidence intervals. *Journal of the American Statistical Association*, *56*, 549–567.

Robertson, H. (1936). An interpretation of page's "new relativity". *Phyiscal Review*, *49*, 755–760.

Schafer, C., & Stark, P. (2006). *Constructing Confidence Sets of Optimal Expected Size* (Technical Report 836). Department of Statistics, Carnegie Mellon University.

Schafer, C. M., & Stark, P. B. (2003). Using what we know: Inference with physical constraints. *PHYSTAT 2003: Statiscal Problems in Particle Physics, Astrophysics, and Cosmology*.

Wasserman, L. (2004). *All of statistics*. New York: Springer-Verlag.