Computing the Joint Range of a Set of Expectations

Charles J. Geyer
School of Statistics
University of Minnesota

Radu C. Lazar
School of Statistics
University of Minnesota

Glen D. Meeden
School of Statistics
University of Minnesota

Abstract

In the theory of imprecise probability it is often of interest to find the range of the expectation of some function over a convex family of probability measures. Here we show how to find the joint range of the expectations of a finite set of functions when the underlying space is finite and the family of probability distributions is defined by finitely many linear constraints.

Keywords. linear constraints, probability assessment, convex family of priors, polytope

1 Introduction

The theory of imprecise probability arises when subjective Bayesians are unable to select a single probability distribution that reflects their prior knowledge and beliefs about the unknown state of nature. In such cases a Bayesian often selects a convex family of prior distributions to represent their prior knowledge. Cozman (1999) noted that two of the three most common ways of specifying such families are either by extreme points or collections of linear constraints. The first is usually more convenient to deal with while the second is often a more natural way to incorporate prior information.

When the family of possible states of nature is finite the Minkowski-Weyl theorem states that these two approaches must be equivalent (every convex polyhedron can be represented either as the finite intersection of closed half spaces or as the convex hull of a finite set of points and directions). In the case of interest to us, where the polyhedron is bounded, hence a *polytope*, the two representations are a finite intersection of half spaces (H-representation) or the convex hull of a finite set of *vertices* (extreme points) (V-representation).

Recent advances in computational geometry have produced practicable algorithms for moving back and forth between the two representations. Fukuda (2004a) has produced a library (`cddlib`, version 093d) of C functions for this. We have written a package (`rcdd`) for the R statistical computing environment (R Development Core Team, 2004), which provides an interface to some of the functionality of `cddlib`, in particular the conversion between H- and V-representations. This makes `cddlib` much easier to use (for anyone familiar with R).

Given a family of distributions, say $\mathcal{P}$, one is often interested in finding the range of expectations of some specified real valued function over the family. Here we consider the problem of finding the joint range of expectations for a finite set of such functions. Since expectation is a linear operation, the extreme points of the range set must be contained in the the images of the extreme points of $\mathcal{P}$. We show (Theorem 1 below) that the family of posterior distributions given data is also a convex polytope whose extreme points are among the images (under the mapping induced by Bayes' theorem) of extreme points of the family of prior distributions. Therefore being able to find the extreme points of $\mathcal{P}$ is a powerful tool.

When the possible states of nature are finite and the family of possible distributions is defined by linear constraints, Dickey (2003) has developed an interactive computing environment which finds the minimum and maximum of the expectation of a specified function over the family. This can be helpful to a Bayesian who must sequentially incorporate prior information in a coherent manner. Lazar and Meeden (2003) argued that in such settings considering the joint range of possible expectations for a finite set of functions can be more informative than separately considering ranges of different functions. Here we revisit this problem and present more convenient methods for finding the solution.

In Section 2 we formally state our problem and show that in a statistical setting being able to solve the problem for prior expectations yields an easy solution for the problem with posterior expectations after the data have been observed. In Section 3 we show how to use our R library to find solutions when there are finitely many states of nature. In Section 4 we show how our approach can be used to find approximate solutions when the states of nature belong to a bounded interval of real numbers. In particular, solutions based on a finite subset of values belonging to an interval provide inner bounds for the actual solution for the interval.

2 The Finite-Dimensional Problem

Consider the case where the there are only a finite number of states of nature: the *parameter space* Θ is a finite set. A prior distribution p is a probability function on Θ, but we identify it with a vector in $\mathbb{R}^k$ where k is the number of points in Θ. This says no more than finite dimensional vector spaces of the same dimension are isomorphic and it makes no mathematical difference whether we consider p an element of $\mathbb{R}^\Theta$ or of $\mathbb{R}^k$. In one case we write $p(\theta)$ and in the other p_i, but the distinction is merely notational. In either case the index (θ or i) takes values in a finite set with k elements. When thinking probabilistically, the functional notation $p(\theta)$ is more natural. When thinking computationally, the vector notation p_i is more natural.

The expectation of a scalar function a with respect to a probability vector p is written

$$\sum_{\theta \in \Theta} a(\theta) p(\theta) \tag{1}$$

in functional notation. In vector notation we interpret a as a vector in $\mathbb{R}^k$ and write the expectation

$$a^T p = \sum_{i=1}^{k} a_i p_i \tag{2}$$

but in either case we have a sum with k terms, so (1) and (2) are the same thing in different notation.

The expectation of a vector function can be written as a matrix multiplication Ap, each row of the matrix A corresponding the transpose of a vector representing a scalar function. Now the vector notation is much simpler. If A has elements a_{ij}, then the i-th element of Ap is

$$(Ap)_i = \sum_{j=1}^{k} a_{ij} p_j$$

and translating the sum to functional notation we obtain

$$\sum_{\theta \in \Theta} a_i(\theta) p(\theta)$$

which gives the expectation of the i-th component $a_i(\theta)$ of the random vector under discussion. For the rest of the article, we will pass back and forth between vector and functional notations in silence.

The requirement that p represent a probability distribution can be written

$$p \geq 0 \tag{3a}$$
$$u^T p = 1 \tag{3b}$$

where here and throughout the article inequalities involving vectors are interpreted coordinate-wise, so (3a) means $p_i \geq 0$ for all i, and u is the vector with all coordinates equal to one, so (3b) means $\sum_i p_i = 1$.

Specifying equality and inequality restrictions on a finite set of scalar functions can be written in matrix notation as

$$A_1 p = b_1 \tag{3c}$$
$$A_2 p \leq b_2 \tag{3d}$$

(The dimensions are such that the equations make sense: A_1 and A_2 have column dimension k and the row dimension of A_i is the same as that of b_i, which is a column vector). The set of p satisfying (3a), (3b), (3c), and (3d) is a convex polytope in $\mathbb{R}^k$, which we denote $\mathcal{P}$, and represents an imprecise prior probability specification.

Now let ψ be a scalar function on Θ (or the vector in $\mathbb{R}^k$ representing it) and more generally let Ψ be a matrix, each row of which represents a scalar function on Θ, so Ψp is the vector of expectations of these scalar functions. The image of $\mathcal{P}$ under Ψ

$$\mathcal{R}(\Psi) = \{\, \Psi p \mid p \in \mathcal{P} \,\} \tag{4}$$

is the *joint range* of these expectations as our prior probabilities range over $\mathcal{P}$. Since the image of a convex polytope under a linear map is another convex polytope, and since the extreme points of the image must be images of extreme points, $\mathcal{R}(\Psi)$ is a convex polytope and its extreme points are among the images of the extreme points of $\mathcal{P}$.

In the usual statistical setting, after determining $\mathcal{P}$ and observing data according to a family of possible probability models indexed by θ, statisticians using the likelihood function update their prior information via Bayes' theorem producing the posterior. In our setup, the likelihood is represented by a diagonal matrix Λ_x, whose diagonal elements represent the probability of the observed data given the parameter, $f(x|\theta)$ in conventional notation. Then Bayes rule maps a prior p to a posterior

$$\frac{\Lambda_x p}{u^T \Lambda_x p} \tag{5}$$

(assuming the denominator is nonzero, which happens whenever the observed data is not impossible under the prior p). When we have a family of priors $\mathcal{P}$ we are interested in what the Bayes rule does to each one of them. Let B_x denote the function (the *Bayes map*) that maps a prior p to (5), and let $\mathcal{P}_x$ denote the image of $\mathcal{P}$ under the Bayes map. So $\mathcal{P}$ is our family of priors and $\mathcal{P}_x$ the corresponding family of posteriors (for observed data x).

Now let us return to the family of scalar functions on the parameter space represented by the matrix Ψ. We are interested not only in the joint range of prior expectations (4), but also in the joint range of posterior expectations

$$\mathcal{R}_x(\Psi) = \{\, \Psi p \mid p \in \mathcal{P}_x \,\} \tag{6}$$

The diagram below shows the relationships between these sets

$$\begin{array}{ccc} \mathcal{P} & \xrightarrow{\Psi} & \mathcal{R}(\Psi) \\ {\scriptstyle B_x}\big\downarrow & & \\ \mathcal{P}_x & \xrightarrow{\Psi} & \mathcal{R}_x(\Psi) \end{array}$$

The theorem below gives important properties of these sets. A convex subset F of a convex set C is a *face* of C if whenever $x \in F$ and $y, z \in C$ with $x = ty + (1-t)z$ and $0 < t < 1$ we actually have $y, z \in F$ (Rockafellar, 1970, p. 163). For a convex polytope, every face is the convex hull of the vertices it contains. A vertex is a face that consists of a single point.

Theorem 1. *Assume that each member in the diagonal of Λ_x is greater than zero, Then the Bayes map takes convex sets to convex sets and convex polytopes to convex polytopes, and pre-images of faces of the image are faces of the domain.*

Proof. The "numerator" of the Bayes map $p \mapsto \Lambda_x p$ is linear, and maps convex sets to convex sets. Let $\mathcal{L}_x$ denote the image of $\mathcal{P}$ under this map. Let $\operatorname{pos}\mathcal{L}_x$ denote the "positive hull" of this set (the set of all non-negative combinations of points in the set, which is the polyhedral convex cone generated by it). The intersection of $\operatorname{pos}\mathcal{L}_x$ with the hyperplane

$$\mathcal{H}_1 = \{\, p \in \mathbb{R}^k \mid u^T p = 1 \,\}$$

(where u is as in (3b)) is the image of $\mathcal{P}$ under the Bayes map, or, to be more precise, the image of those elements of $\mathcal{P}$ that do not map to zero under $p \mapsto \Lambda_x p$. Call this intersection $\mathcal{P}_x$.

As the intersection of a convex cone and a hyperplane is a convex set, so is $\mathcal{P}_x$. Since $\mathcal{P}_x$ is a subset of the unit simplex, it is bounded. If $\mathcal{P}$ is polyhedral, so are $\mathcal{L}_x$, $\operatorname{pos}\mathcal{L}_x$, and $\mathcal{P}_x$, and $\mathcal{P}_x$ is a convex polytope.

Let $\mathcal{F}_x$ be a face of $\mathcal{P}_x$ and let $\mathcal{F}$ be the pre-image of $\mathcal{F}_x$ under the Bayes map. Consider points $p \in \mathcal{F}$ and r and s in $\mathcal{P}$ such that $p = tr + (1-t)s$ with $0 < t < 1$. Define $p^* = \Lambda_x p$ and similarly for r^* and s^*. Then define $\tilde{p} = p^*/\|p^*\|$ and similarly for $\tilde{r}$ and $\tilde{s}$, where $\|p\| = \sum_i |p_i| = \sum_i p_i$. Then

$$\tilde{p} = \tilde{t}\tilde{r} + (1-\tilde{t})\tilde{s}$$

with $\tilde{t} = t\|r^*\|/\|p^*\|$. Since $\mathcal{F}_x$ is a face, this implies $\tilde{r}$ and $\tilde{s}$ are elements of $\mathcal{F}_x$, hence that r and s are elements of $\mathcal{F}$, and that implies $\mathcal{F}$ is a face of $\mathcal{P}$. $\square$

We were unaware of this fact when Lazar and Meeden (2003) was written. In retrospect it seems like it should be known but we have been unable to find a reference for it. We note there is nothing finite-dimensional about our proof except the assertion that $\mathcal{P}_x$ is a convex polytope whenever $\mathcal{P}$ is. The rest of the proof remains true when $\mathcal{P}$ is a family of probability measures on Θ, when the "numerator map" takes p to p^* defined by

$$p^*(B) = \int_B f(x|\theta)p(d\theta)$$

and when the hyperplane $\mathcal{H}_1$ is defined by

$$\mathcal{H}_1 = \left\{ q \in \mathfrak{M}(\Theta) : \int q(d\theta) = 1 \right\}$$

where $\mathfrak{M}(\Theta)$ is the set of all finite signed measures on Θ.

3 Using the RCDD Package in R

The key operation in all of this is finding the extreme points (vertices) of a convex polytope. Let us see how this is done in R. For this example $k = 4$. In addition to the usual constraints given in equations (3a) and (3b) we will add the equality constraint $\sum_i i \cdot p_i = 2.5$ and the inequality constraint $p_1 + p_2 \leq 0.4$. All this constraint information will be put into one matrix called `qux` below.

```
> library(rcdd)
> qux <- makeH(-diag(4), rep(0,4),
+     rep(1,4), 1)
> qux <- addHeq(c(1,2,3,4), 2.5, qux)
> qux <- addHin(c(1,1,0,0), 0.4, qux)
> print(qux)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1  1.0   -1   -1   -1   -1
[2,]    0  0.0    1    0    0    0
[3,]    0  0.0    0    1    0    0
[4,]    0  0.0    0    0    1    0
[5,]    0  0.0    0    0    0    1
```

```
[6,]     1  2.5   -1   -2   -3   -4
[7,]     0  0.4   -1   -1    0    0
attr(,"representation")
[1] "H"
```

(The > and + in the first column are prompts, the latter being the continuation prompt for a line continuing an incomplete statement. The symbol <- is the R assignment operator. The whole block makes a 7×6 matrix `qux`.)

The first command makes the `rcdd` library of functions available. The next command makes a matrix that contains the constraints given in in equations (3a) and (3b). The next command `addHeq` adds a row to the matrix specifying the additional equality constraint and the command `addHin` adds another row specifying the additional inequality constraint.

The first two columns of this matrix are special. In the first column 1 indicates an equality constraint and 0 an inequality constraint. The second column contains the elements of the right hand side vectors, the b_i in (3c), and (3d), and the zeros and 1 in (3a) and (3b). The rest of the columns are -1 times the left hand side matrices, the A_i in (3c), and (3d), and the negative of u^T in (3b), and the implied basis vectors in (3a). So row 1 of `qux` represents (3b), rows 2 through 5 represent (3a), row 6 represents the equality constraint $\sum_i i \cdot p_i = 2.5$, and row 7 represents the inequality constraint $p_1 + p_2 \leq 0.4$. Because the first two rows are special, the column dimension of `qux` is two more than the column dimension of the A_i's which is 4 in this case.

Having created the H-representation of $\mathcal{P}$, the matrix `qux`, we now can find the V-representation using one command.

```
> out <- scdd(qux)
> print(out)
$output
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    1 0.25  0.0 0.75  0.0
[2,]    0    1 0.40  0.0 0.30  0.3
[3,]    0    1 0.10  0.3 0.60  0.0
attr(,"representation")
[1] "V"
```

Again the first two columns of the matrix (`out$output`) are special. For a polytope, they are always 0 in the first column and 1 in the second column and can be ignored (they are only interesting for unbounded polyhedra). Each row of the remaining matrix (columns 3 through 6) is a vertex of $\mathcal{P}$. The first row says $(\frac{1}{4}, 0, \frac{3}{4}, 0)$ is a vertex. Another call to the `scdd` function would go back from V-representation to H-representation (but that is not of interest here).

Example 1. Here we consider a slightly more complicated toy example. We let $k = 10$ and imposed two equality and two inequality constraints. The equalities were $p_5 = p_6$ and $\sum_i i \cdot p_i = 5.5$. The inequalities were $p_1 \leq p_2$ and $p_1 + p_2 + p_3 + p_4 \leq 0.5$. We had two linear functions of interest. We let ψ_1 be the variance function defined by $\psi_1(i) = (i - 5.5)^2$ and let ψ_2 be the indicator function of the set $\{2, 3, 4, 5\}$. In order to do the posterior calculations we needed to specify probabilities of seeing the observed data, x, under the 10 possible parameter values. These are the diagonal elements of Λ_x and were taken to be 0.1, 0.15, 0.09, 0.2, 0.3, 0.2, 0.1, 0.05, 0.07 and 0.02.

The R code to create the H-representation is

```
> d <- 10
> qux <- makeH(-diag(d), rep(0,d),
+     rep(1,d), 1)
> qux <- addHeq(c(0,0,0,0,1,-1,0,0,0,0),
+     0, qux)
> qux <- addHeq(1:d, 5.5, qux)
> qux <- addHin(c(1,-1,0,0,0,0,0,0,0,0),
+     0, qux)
> qux <- addHin(c(1,1,1,1,0,0,0,0,0,0),
+     0.5, qux)
```

and to create the V-representation is

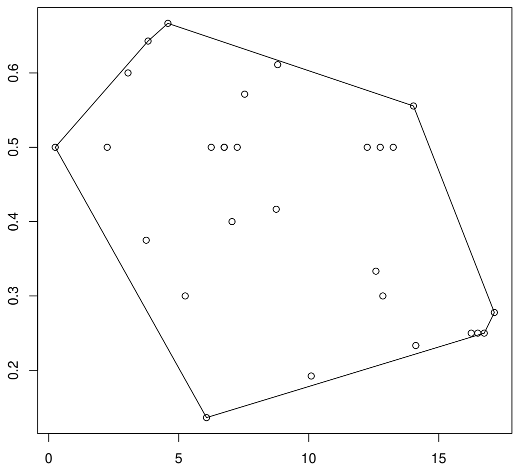
```
> out <- scdd(qux)
> vert <- out$output[ , -(1:2)]
> dim(vert)
[1] 28 10
```

The second line in the above throws away the first two columns of `out$output` to create the matrix vert whose rows contain all the vertices of $\mathcal{P}$ for this example. The function `dim` finds the dimensions of vert and we see that the V-representation vert contains 28 vertices.

Next we find $\mathcal{R}(\Psi)$. The code to do this is given just below.

```
> Psi <- rbind((1:d - 5.5)^2,
+     c(0,1,1,1,1,0,0,0,0,0))
> rang <- vert %*% t(Psi)
> plot(rang, xlab="", ylab = "")
> fred <- chull(rang)
> polygon(rang[fred, ])
> length(fred)
[1] 7
```

We begin by creating the matrix Ψ. This is done with the `rbind` command which binds row vectors together to form a matrix. The next line finds $\mathcal{R}(\Psi)$ which here we call `rang` (`%*%` denotes matrix multiplication in R and `t` is the transpose function). The next three



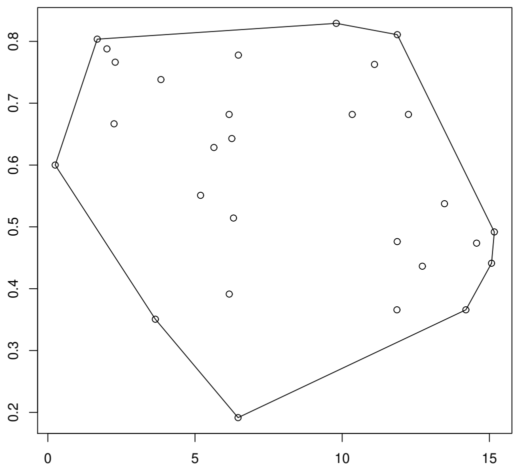


Figure 1: The plot of $\mathcal{R}(\Psi)$ for Example 1. The horizonal axis is $E_p(\psi_1)$ and the vertical axis is $E_p(\psi_2)$.

Figure 2: The plot of $\mathcal{R}_x(\Psi)$ for Example 1. The horizonal axis is $E_p(\psi_1|x)$ and the vertical axis is $E_p(\psi_2|x)$.

lines create the plot shown in Figure 1. The `chull` (for convex hull) function finds the extreme points of `rang`. The last line tells us that the polygon $\mathcal{R}(\Psi)$ has 7 vertices.

Next we find $\mathcal{P}_x$ as follows

```
> Lambda <- diag(c(0.1, 0.15, 0.09, 0.2,
+     0.3, 0.2, 0.1, 0.05, 0.07, 0.02))
> post <- vert %*% Lambda
> norm <- apply(post, 1, sum)
> post <- sweep(post, 1, norm, "/")
```

where `Lambda` is the matrix Λ_x, the first assignment to `post` creates the image of $\mathcal{P}$ under $p \mapsto \Lambda_x p$, and the `apply` and `sweep` commands are the R way of normalizing the rows of the matrix to sum to one.

The calculation of $\mathcal{R}_x$ from $\mathcal{P}_x$ is just like the calculation of $\mathcal{R}$ from $\mathcal{P}$ and is not shown (just do to `post` what we did to `vert` above). The result is shown in Figure 2. It has 9 vertices.

A feature of the `rcdd` package that we have not illustrated but which is important in applications is its ability to use exact unlimited precision rational arithmetic. This is necessary in large problems where rounding error in conventional computer arithmetic may cause failure of the algorithm used by `cddlib`. Rationals are represented as simple character strings, for example, `"-13/15"`, and `rcdd` contains functions to convert between rational representations and conventional computer floating point numbers.

4 Approximate Solutions

In this section we consider the situation where Θ is a bounded interval of real numbers and the constraints and prior information are specified by equality and inequality constraints on integrals.

Kemperman (1968) considered the situation where the possible states of nature were an interval of real numbers and the family of probability measures was defined by equality constraints on a finite set of expectations. He showed that the set of possible expected values for a given function was a closed interval of real numbers. Moreover, the endpoints of this interval correspond to distributions concentrated on finite sets whose size is at most the number of constraints plus one. This allows for solutions to be found approximately using linear programming. Kemperman (1968, p. 96) briefly considered the more general problem of finding the joint range of the expectations of a set of functions $(\psi_1, \ldots, \psi_k)$. He noted that the range of this vector over the family defined by the constraints is a convex set in k-dimensional Euclidian space. The closure of this space is completely determined by all its supporting hyperplanes. These hyperplanes can be determined by finding the maximum and minimum values of $\sum_{i=1}^{k} a_i \psi_i$ for all possible choices of the a_i's.

This suggests that in such cases one can find $\mathcal{R}(\Psi)$ approximately by specifying a finite subset of values in Θ and solving the corresponding finite problem. We now show how this works in two simple examples. For both examples we assume that $\Theta = [-1, 1]$ but

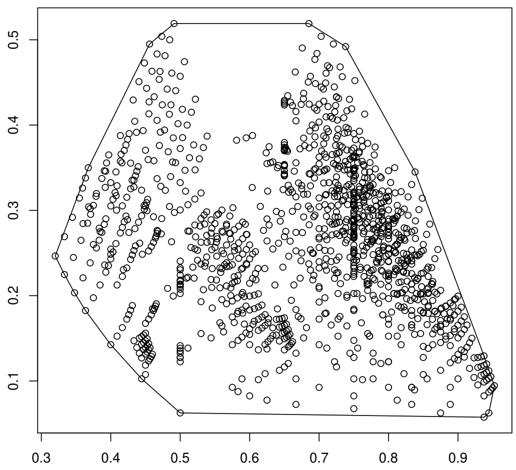


Figure 3: The plot of $\mathcal{R}(\Psi)$ for a grid of 21 points for Example 2. The horizonal axis is $E_p(\psi_1)$ and the vertical axis is $E_p(\psi_2)$.

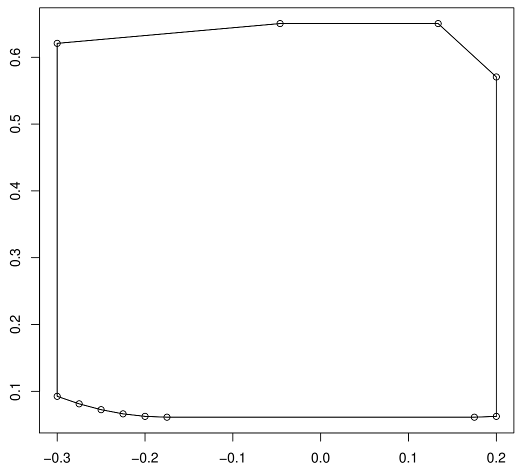


Figure 4: The plot of the boundary of $\mathcal{R}(\Psi)$ for a grid of 41 points for Example 3. The horizonal axis is $E_p(\psi_1)$ and the vertical axis is $E_p(\psi_2)$.

we will restrict ourselves to priors whose support is just a finite number of points.

Example 2. We assume that the prior information is defined by the constraints

$$
\begin{gathered}
P(\theta \le -0.6) \ge P(\theta \ge 0.6) \\
P(\theta < -0.9) \le P(-0.9 \le \theta < -0.8) \\
P(-0.9 \le \theta < -0.8) \le P(-0.8 \le \theta < -0.7) \\
P(-0.8 \le \theta < -0.7) \le P(-0.7 \le \theta < -0.6) \\
0.3 \le P(-0.3 \le \theta \le 0.3) \le 0.5 \\
P(0.6 \le \theta < 0.7) \ge P(0.7 \le \theta < 0.8) \\
P(0.7 \le \theta < 0.8) \ge P(0.8 \le \theta < 0.9) \\
P(0.8 \le \theta < 0.9) \ge P(\theta > 0.9) \\
E(\theta) = -0.15
\end{gathered}
$$

We selected as our grid the sequence of 21 equally spaced values running from -1.0 to 1.0 and constructed the matrix incorporating our constraints. We then ran `scdd` to find the vertices of the polytope of distributions which are defined on our grid and satisfy the constraints. This took just a couple of seconds on our PC and found 1,236 vertices. We let ψ_1 be the indicator function of the interval $[-1.0, 0.0]$ and $\psi_2(\theta) = (\theta + 0.15)^2$. Next we found that $\mathcal{R}(\Psi)$ had 17 extreme points. Its plot is given in Figure 3. We can see for any fixed value of $P(\theta \le 0)$ the approximate range of the variance of θ. Or for any fixed value of the variance of θ we can see the approximate range of $P(\theta \le 0)$.

Example 3. We assume that the prior information yields the constraints

$$
\begin{gathered}
P(\theta \le -0.6) \ge P(\theta \ge 0.6) \\
P(\theta \in [-1.0, -0.8]) \le P(\theta \in [-0.75, -0.50]) \\
0.3 \le P(-0.3 \le \theta \le 0.3) \le 0.5 \\
-0.3 \le E(\theta) \le 0.2
\end{gathered}
$$

We selected as our grid the sequence of 41 equally spaced values running from -1.0 to 1.0 and then constructed the matrix incorporating our constraints. We then ran `scdd` to find the vertices of the polytope of distributions which are defined on our grid and satisfy the constraints. This took two or three minutes on our PC and found 58,528 vertices. We set $\psi_1(\theta) = \theta$ and $\psi_2(\theta) = \theta^2$ and found that $\mathcal{R}(\Psi)$ had 12 extreme points and its plot is given in Figure 4.

In imprecise probability theory one is often interested in finding not only the range of the expected value of a function but the range of its variance as well. See for example (Walley, 1996). This range can be determined approximately just by studying our plot since for each point the variance is $E_p(\theta^2) - [E_p(\theta)]^2$. For example the maximum value for the variance is in the neighborhood of 0.65 and will arise from a distribution whose mean is close to zero. Furthermore as $E_p\psi_1$ moves away from zero the maximum value of the variance will behave roughly as $E_p\psi_2$ while the minimum value increases slightly. We believe that the consideration of such plots can prove helpful in the elicitation and assessment of prior information and

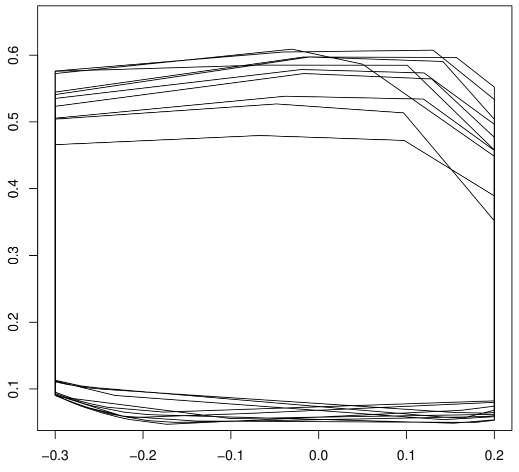


Figure 5: The plot of the boundary of $\mathcal{R}(\Psi)$ for 10 randomly selected grids of 30 points for Example 3. The horizonal axis is $E_p(\psi_1)$ and the vertical axis is $E_p(\psi_2)$.

beliefs.

Note that in the last two examples where Θ is an interval our plots only provide inner bounds for the true ranges. This suggests some natural questions. Can one find an outer bound? Failing that, can one find a bound on the error of our bound? How should one select the finite subset of points of Θ to compute our inner bound? Good questions all, but unfortunately we have no theoretical answers for them.

If in Example 3 we use grids with just 30 points, then the calculations take only a few seconds. To study how the choice of grid can affect our answer we selected 10 random samples of size 30 from the uniform distribution on the interval $[-1, 1]$ to use as grids. In Figure 5 we plotted the resulting 10 ranges of $(E_p\psi_1, E_p\psi_2)$. The convex set generated by these 10 ranges must be an inner bound as well and seems to be just about as large as the inner bound given in Figure 4. This suggests that $\mathcal{R}(\Psi)$ is probably not too much larger than the plot given in Figure 4. But of course we cannot know for sure without some theoretical results.

5 Discussion

Betrò and Gugliemi (2000) considered robust Bayesian analysis under moment constraints in a fairly abstract setting and concluded that none of the current algorithms were good enough to be adopted for routine use. We have argued here, for problems with finitely many states of nature, that modern computational geometry algorithms make specifying a family of possible prior distributions through a collection of linear constraints practical. This allows one to combine ease of specification with ease of computing for both prior and posterior expectations of not just one function of interest but any finite set of functions. Plotting the range of the prior expectations for different pairs of functions should be helpful in finding good approximations to one's prior beliefs and the corresponding posterior consequences.

A special case of conversion from H-representation to V-representation is called the *vertex enumeration* problem in the computational complexity literature (Fukuda, 2004b), and, as far as we know, its complexity is still an open question. However, the fact that the computational complexity of the simplex algorithm for linear programming was open for many years and finally resolved as worst-case exponential complexity in no way prevented the simplex algorithm from having a huge variety of important applications. The computational geometry code in `cddlib` may not have known computational complexity, but many scientists have found it useful in a wide variety of application areas. We claim it may be useful in imprecise probability.

We should say that, although we have presented our examples using these computational geometry algorithms, that some of the problems we address can be recast so as to require only the solution of multiple linear programming problems. Since linear programming is now known to have polynomial time complexity, such an algorithm would also have polynomial time complexity if the number of linear programs to be solved were also polynomial. Therefore, for very large problems that can be recast in this form, algorithms based on linear programming should be used instead of `rcdd`.

Although the algorithms we present may not have optimal worst-case computational complexity, their geometric nature makes their operation transparent so they are very easy to use and experiment with. If our approach turns out to have important large scale applications, then would be the time to switch to multiple linear programming algorithms.

An easy way to try out our approach in simple problems is to go to

`http://www.stat.umn.edu/geyer/imprecise/`

where our Example 1 is redone via Rweb (R on the web). One can modify the code in the example by simply editing the text in the web form and thus do small experiments with the technique.

For serious work you need to install `cddlib`, the GNU multiple precision (GMP) library that it requires, R, and our `rcdd` package. Instructions for doing this are at

`http://www.stat.umn.edu/geyer/rcdd/`

Acknowledgments

Charles Geyer is a Professor in the School of Statistics at the University of Minnesota, 313 Ford Hall, 224 Church St SE, Minneapolis, MN 55455-0493, USA. E-mail: charlie@stat.umn.edu.

Radu Lazar received his PhD from the School of Statistics in January 2005. E-mail: lazar@stat.umn.edu.

Glen Meeden is a Professor in the School of Statistics at the University of Minnesota, 313 Ford Hall, 224 Church St SE, Minneapolis, MN 55455-0493, USA. E-mail: glen@stat.umn.edu. His research was supported in part by NSF grant DMS-0406169.

References

B. Betrò and A. Gugliemi. Methods for Global Prior Robustness under Generalized Moment Conditions. *Robust Bayesian Analysis* (D. R. Insua and F. Ruggeri eds.) 273-293, Springer, 2000.

F. G. Cozman. Calculation of Posterior Bounds given Convex Sets of Prior Probability Measures and Likelihood Functions. *Journal of Computational and Graphical Statistics*, 8:824-838, 1999.

J. Dickey. Convenient Interactive Computing for Coherent Imprecise Prevision Assessments. *ISIPTA '03*, 218-230, Carleton Scientific, 2003.

K. Fukuda. `cdd` and `cddplus` Homepage. `http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html`, 2004a.

K. Fukuda. Frequently Asked Questions in Polyhedral Computation `http://www.ifor.math.ethz.ch/~fukuda/polyfaq/polyfaq.html`, 2004b.

J. H. B. Kemperman. The General Moment Problem, A Geometric Approach. *The Annals of Mathematical Statistics*, 39:93-122, 1968.

R. Lazar and G. Meeden. Exploring Imprecise Probability Assessments Based on Linear Constraints. *ISIPTA '03*, 361-371, Carleton Scientific, 2003.

R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing. `http://www.R-project.org`, 2004.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.

P. Walley. Inferences from Multinomial Data: Learning about a Bag of Marbles. *Journal of the Royal Statistical Society B* 58:3-57 1996.