4th International Symposium on Imprecise Probabilities and Their Applications, Pittsburgh, Pennsylvania, 2005

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) says that two of the three most common ways of specifying such families were 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 (2004) 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 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. For problems with finitely many states of nature we use `cddlib` to find the extreme points of a family of prior distributions that is a convex polytope We then 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 family of prior distributions, and hence the joint range of a finite set of posterior expectations must be contained in the images of the extreme points of the family of prior distributions. Where the set of states of nature is infinite, our solutions to finite-dimensional problems provide inner bounds for the infinite-dimensional problem.

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 two we formally state our problem. We then 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 three we demonstrate how to use our R library to find solutions when there are finitely many states of nature. In section four 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 section five we conclude with a brief discussion.

2 The Finite-Dimensional Problem

We 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. The requirement that p represent a probability distribution can be written

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

where here and throughout the paper inequalities involving vectors are interpreted coordinate-wise, so (1a) means $p_i \geq 0$ for all i, and u is the vector with all coordinates equal to one, so (1b) means $\sum_i p_i = 1$. The set of all probability vectors, those satisfying (1a) and (1b) are denoted by $\mathcal{S}$ (for "unit simplex").

An expectation of a scalar function a with respect to a probability vector p can be written $\sum_\theta a(\theta)p(\theta)$, but we also interpret a as a vector in $\mathbb{R}^k$ and write the expectation $a^T p$. The expectation of a vector function can be written as a matrix multiplication Ap, each row of the matrix A corresponding to an a^T for a vector a representing a scalar function. Specifying equality and inequality restrictions on a finite set of scalar functions can be written in matrix notation as

$$A_1 p = b_1 \tag{1c}$$
$$A_2 p \leq b_2 \tag{1d}$$

(the dimensions are such as to make 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 (1a), (1b), (1c), and (1d) is a convex polytope in $\mathbb{R}^k$, which we denote $\mathcal{P}$, and is our 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

$$\mathcal{R}(\Psi) = \{\, \Psi p \mid p \in \mathcal{P} \,\} \tag{2}$$

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 $\mathcal{P}$ the statistician observes data, and updates 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

$$\frac{\Lambda_x p}{u^T \Lambda_x p} \tag{3}$$

(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 Bayes rule does to each one of them. Let B_x denote the function (the *Bayes map*) that maps a prior p to (3), 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 (2), but also in the joint range of posterior expectations

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

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.

Theorem 1. *The Bayes map, when defined, maps convex polytopes to convex polytopes, and extreme points of the image are images of extreme points of the domain. The same is true if points where the Bayes map is undefined are excluded.*

Proof. The "numerator" of the Bayes map $p \mapsto \Lambda_x p$ is linear, and maps convex polytopes to convex polytopes, and extreme points of the image are images of extreme points of the domain. 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 extreme rays of $\operatorname{pos} \mathcal{L}_x$ are generated by images of (some) extreme points of $\mathcal{P}$. 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 (1b)) 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 $\mathcal{P}_x$ (which is what it is when the Bayes map is always defined).

As the intersection of a polyhedral convex cone and a hyperplane is a polytope, so is $\mathcal{P}_x$. Since $\mathcal{P}_x$ is a subset of the unit simplex, it is bounded, hence a convex polytope. Since $\mathcal{H}_1$ imposes no inequality constraints, any extreme point of $\mathcal{P}_x$ must lie on an extreme ray of $\operatorname{pos} \mathcal{L}_x$ and hence must be the image under B_x of an extreme point of $\mathcal{P}$. □

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.

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. We first incorporate all the constraints (1a), (1b), (1c), and (1d) into one R matrix.

```
> 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)
> 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 <- are assignment operators. The whole block makes a 7×6 matrix `qux`.) 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 (1c), and (1d), and the zeros and 1 in (1a) and (1b). The rest of the columns are the left hand side matrices, the A_i in (1c), and (1d), the u^T in (1b), and the implied basis vectors in (1a). So row 1 of `qux` is (1b), rows 2 through 5 are (1a), row 6 is an equality constraint $\sum_i i \cdot p_i = 2.5$, and row 7 is an inequality constraint $p_1 + p_2 \leq 0.4$. Because the first two rows are special, the column dimension of the A_i matrices is 4, hence the dimension of p is 4 here.

Having created the H-representation of $\mathcal{P}$ in terms of equalities and inequalities. The V-representation in terms of vertices is done by 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`) 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. 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 $\{2, 3, 4, 5\}$. When doing the posterior calculations we assumed that the probabilities of seeing the observed data under the 10 possible parameter values were 0.1, 0.15, 0.09, 0.2, 0.3, 0.2, 0.1, 0.05, 0.07 and 0.02 (so these are the diagonal elements of Λ_x).

The R 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
```

As the `dim` function shows, the V-representation has 28 vertices. The preceding statement throws away the first two columns of the output matrix so the rows of `vert` are the vertices.

To find $\mathcal{R}(\Psi)$ we create the matrix Ψ and map the vertices under it

```
> Psi <- rbind((1:d - 5.5)^2,
+     c(0,1,1,1,1,0,0,0,0,0))
> rang <- vert %*% t(Psi)
```

and then we use the function `chull` (for convex hull) to find the vertices of the image and plot the convex hull.

```
> plot(rang, xlab="", ylab = "")
> fred <- chull(rang)
> polygon(rang[fred, ])
```

The result is shown in Figure 1. The polygon $\mathcal{R}(\Psi)$ has 7 vertices, as shown by

```
> length(fred)
[1] 7
```

Next we found $\mathcal{P}_x$ 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`

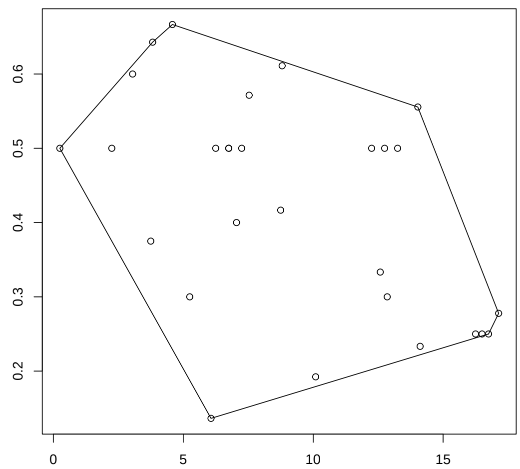

Figure 1: The plot of $\mathcal{R}(\Psi)$ for Example 1.

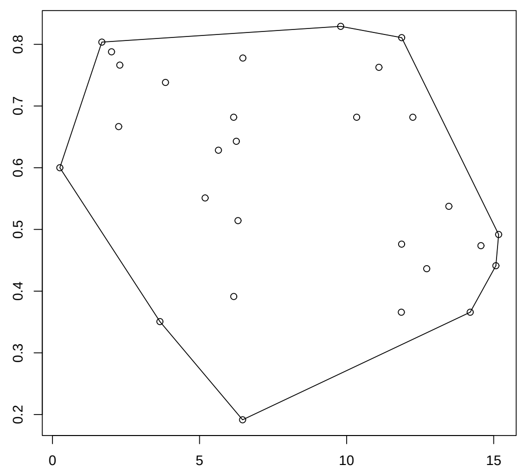

Figure 2: The plot of $\mathcal{R}_x(\Psi)$ for Example 1.

what we did to `vert` above). The result is shown in Figure 2. It has 9 vertices.

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 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

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

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 and its plot is given in Figure 3. We can see for any fixed value of $P(\theta \leq 0)$ 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 $P(\theta \leq 0)$.

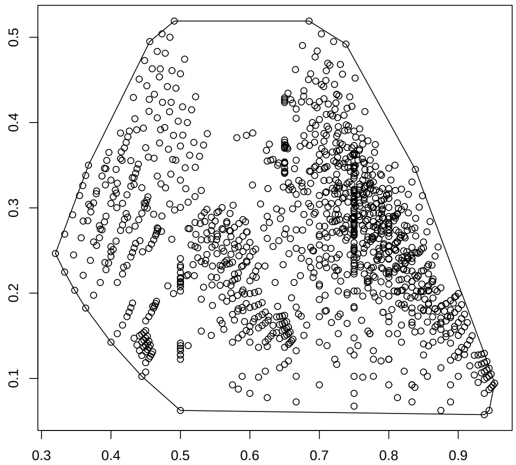

Figure 3: The plot of $\mathcal{R}(\Psi)$ for a grid of 21 points for Example 2.

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

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

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.

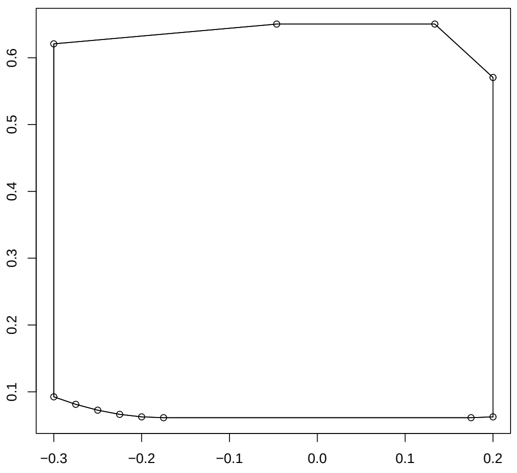

Figure 4: The plot of the boundary of $\mathcal{R}(\Psi)$ for a grid of 41 points for Example 3.

See for example (Walley, 1996). This range can be determined approximately just by studying our plot. 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 beliefs.

Note in the last two examples our plots only provide inner bounds for the true ranges corresponding to $\Theta = [-1, 1]$. One limitation of our approach is that it suffers from the curse of dimensionality. Once the possible states of nature gets two big it will not be able to find the set of vertices. Before that happens it can also become unstable because of floating point arithmetic and not find the correct answer. Our library has the option to do all the calculations in rational arithmetic. This takes longer but will find the correct answer if the problem is not two big.

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 that, for problems with finitely many states of nature, it is no longer just a theoretical fact that specifying a family of possible prior distributions through a collection of linear constraints is equivalent to knowing the extreme points but modern computational geometry algorithms make this useful in practice. 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.

An easy way to try out this 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`, 2004.

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.

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