Mathware & Soft Computing 8 (2001) 93-112

Marginalization Like a Projection

J.F. Verdegay-López and S. Moral
Depto. de Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingeniería Informática
Universidad de Granada. 18071 Granada, Spain
e-mail: jfvl@decsai.ugr.es

Abstract

This paper studies the problem of marginalizing convex polytopes of probabilities represented by a set of constraints. This marginalization is obtained as a special case of projection on a specific subspace. An algorithm that projects a convex polytope on any subspace has been built and the expression of the subspace, where the projection must be made for obtaining the marginalization, has been calculated.

Keywords: Approximate reasoning, Knowledge representation, Uncertainty, Capacities of order-2, Evidence Theory.

1. Introduction

Consider a variable, X, taking values over a finite universe $U = \{u_1, u_2, \ldots, u_n\}$. Information about this variable is given by a probability distribution, which is not completely known; we only know that it belongs to a set of probabilities P.

If $CH(P)$ denotes the convex hull of P [18, 17], the sets P and $CH(P)$ can be considered as equivalent[20]. When the set P is finite, $CH(P)$ is called convex polytope. From behavioral point of view, the convex polytopes of probabilities have been justified by Walley[22] as a form of representation of uncertain information. The convex polytopes of probabilities have been used in this way by a large groupe of authors[6, 3, 4, 15, 20, 22, 21].

Basic operators for convex polytopes of probabilities are combination and marginalization. There are several ways for combining and marginalizing[7, 23, 13, 2, 22]. In this paper, the problem of implementing these operators will be studied.

From our point of view, the combination should be carried out by means of the intersection of convex polytopes (A justification of this choice is to be formed by de Campos[5]). This operator can be implemented in a very simple way when the convex polytopes are given by a set of constraints: if

$$\mathcal{P} = \{x \in I\!R^n \ : \sum_{j=1}^{n} \alpha_{ij} x_j \leqslant b_i, i = 1, 2, \ldots, s\}$$

and

$$\mathcal{Q} = \{x \in I\!R^n \ : \sum_{j=1}^{n} \beta_{ij} x_j \leqslant b'_i, i = 1, 2, \ldots, r\}$$

are two convex polytopes of probabilities of $I\!R^n$, the combination could be the convex polytope given by the constraints:

$$\mathcal{P} \cap \mathcal{Q} = \{x \in I\!R^n \ : \sum_{j=1}^{n} \alpha_{ij} x_j \leqslant b_i, \ \sum_{j=1}^{n} \beta_{lj} x_j \leqslant b'_l, i = 1, 2, \ldots, s, \ l = 1, 2, \ldots, r\}$$

However, the problem of marginalizing is more complex. Traditionally, the marginalization of a convex polytope has been performed by marginalizing extreme points[3, 1, 8, 22]: If we have a convex polytope $\mathcal{P} = CH(\{p_1, p_2, \ldots, p_r\})$ representing information over the set of variables $(X_i)_{i \in I}$, each one defined on the universe $U_i = \{u_{i_1}, u_{i_2}, \cdots, u_{i_{r_i}}\}$, $i \in I$, the marginalization of $\mathcal{P}$ to the set of variables $(X_j)_{j \in J}$, $J \subseteq I$ is given by

$$\mathcal{P}^{\downarrow J} = CH(\{p_1^{\downarrow J}, p_2^{\downarrow J}, \ldots, p_r^{\downarrow J}\})$$

The aim of this paper is to carry out the marginalization when the convex polytope is represented by a set of constraints instead of the classical way using the extreme points. This marginalization will be done like a projection over a specific subspace. So, an algorithm for making the projection on any subspace will be constructed and, later, the specific subspace where the projection should be made for obtaining the marginalization will be determined.

Theoretical basis for projecting a convex polytope on any subspace is given in the second section. A special attention has been paid to typical problems which appear when we work with polytopes. At this section, projection algorithm is built. In the third one, the subspace, where we must project for obtaining the marginalization, is identified and its expression is calculated. Finally, in the last section, we will comment the obtained results.

2. Projection of a convex polytope

The projection of a convex polytope is a very powerful tool for solving problems automatically from their specification[11, 14]. For example, a typical problem in geometry is to find the convex hull of a given set of points. A point of coordinates $(x_1, x_2 \ldots, x_n)$ is in the convex hull of a set of points $\{p_1, p_2, \ldots, p_r\}$ if and only if there exist $\alpha_1, \alpha_2, \ldots, \alpha_r \geqslant 0$ such that the system

$$x = \sum_{i=1}^{r} \alpha_i p_i \quad \text{and} \quad \sum_{i=1}^{r} \alpha_i = 1$$

is satisfied. That is, if $\alpha_1, \alpha_2, \ldots, \alpha_r$ is a solution of the system

$$\begin{gathered} x_1 = \sum_{i=1}^{r} \alpha_i p_i^1 \\ \cdots\cdots \\ x_n = \sum_{i=1}^{r} \alpha_i p_i^n \\ \sum_{i=1}^{r} \alpha_i = 1 \\ \alpha_1, \alpha_2, \ldots, \alpha_r \geqslant 0 \end{gathered}$$

The representation of the convex hull is an equivalent set of relations solely between the x_i's. It is obtained by eliminating all the α_j's from system by projecting over the subspace related to the variables $\{x_1, \dots, x_n\}$.

This example illustrates the fact that projection provides a systematic way of characterizing interesting sets of constraints directly from a simple existential specification.

2.1. *Characterizing the projection*

Let $\mathcal{P}$ be a convex polytope of $\mathbb{R}^n$ represented by a set of constraints. Suppose $\mathbb{R}^n$ is given on a system of coordinates $\{x_1, x_2, \dots, x_n\}$. Theoretical basis for projecting a convex polytope $\mathcal{P}$ on the subspace $S = \{x_j,\ j \in J \subseteq \{1, 2, \dots, n\}\}$ is explained in this point. The projection on S will be denoted by μ_S.

In the next development, at this point, we shall suppose that the convex polytope is not degenerated, i.e., no degenerated extreme points[17, 9, 19] exist on the convex polytope and the dimension of the convex polytope is n. General case will be studied in another point.

Algorithm is based on, given an extreme point p of $\mathcal{P}$, determining the support hyperplanes of $\mathcal{P}$ which are orthogonal to S and which go through p. We shall prove that the set of these hyperplanes forms a convex cone whose extreme rays, when intersecting with the subspace S, establish the set of support hyperplanes of $\mu_S(\mathcal{P})$ which go through $\mu_S(p)$. That is, the boundary of $\mu_S(\mathcal{P})$ in the area round of $\mu_S(p)$.

This process is always performed starting from an extreme point p of $\mathcal{P}$, but not starting from any extreme point. In fact, the only extreme points which will be considered are those through which support hyperplanes of $\mathcal{P}$ orthogonal to S are going to pass. The following straightforward lemma characterizes these hyperplanes.

Lemma 1. *Let $\mathcal{P}$ be a convex polytope of $\mathbb{R}^n$ and $p = (p_1, p_2, \dots, p_n)$ an extreme point of $\mathcal{P}$. Then, the hyperplanes orthogonal to S going through p are in the following way:*

$$\sum_{j \in J} \alpha_j x_j = \sum_{j \in J} \alpha_j p_j, \qquad \forall\, \alpha_j \in \mathbb{R}$$

Well now, a convex polytope could be defined as the intersection of a set of half-spaces. Without any loss of generality, the hyperplanes orthogonal to S going through p could be forced to contain all the points of $\mathcal{P}$, that is,

$$\forall\, q \in \mathcal{P},\ \sum_{j \in J} \alpha_j q_j \leqslant \sum_{j \in J} \alpha_j p_j.$$

This set of hyperplanes will be denote by $H(p)$. So, $H(p)$ will be the set of support hyperplanes of $\mathcal{P}$ which are orthogonal to S going through p. Now, we shall verify that $H(p)$ forms a convex cone.

Lemma 2. *A bijective mapping could be defined between the hyperplanes of $H(p)$*

and the functions

$$\begin{aligned} \gamma_{\alpha_j, j\in J}: \quad & I\!\!R^n \longrightarrow I\!\!R \\ & f_{\alpha_j, j\in J}(x_1, x_2, \ldots, x_n) = \sum_{j\in J} \alpha_j x_j \end{aligned} \tag{1}$$

whose maximum over $\mathcal{P}$ is reached at point p. This set of functions is denoted by $\Gamma(p)$.

The proof of this lemma is straightforward.

If a function of $\Gamma(p)$ is maximized over $\mathcal{P}$ using the Simplex algorithm[19], this maximun is reached at p. At this moment, the cost vector of the olptimal tableau could be modified for representing any function of (1), i.e.,the new cost vector could be:

$$c_j = \begin{cases} \alpha_j & \text{if } j \in J \\ 0 & \text{otherwise} \end{cases} .$$

The functions of $\Gamma(p)$ will be those that maintain the maximum at p for this new cost vector. That is, those for which the optimality vector $z_k - c_k$ (depending on $\alpha = (\alpha_j)_{j\in J}$, $(z_k - c_k)(\alpha)$) continues being greater or equal than zero[19]. The set of α's values verifiyng this condition constitutes a convex cone of $I\!\!R^d$ ($d = card(J)$) that will be denoted by $\mathcal{C}(p)$. So, according to lemma , the set of hyperplanes $H(p)$ is equivalent $\mathcal{C}(p)$. This correspondence is linear as the next straightforward lemma proves.

Lemma 3. *There is a bijective linear mapping transforming the hyperplanes of $H(p)$ into points on the convex cone $\mathcal{C}(p)$.*

Therefore, $H(p)$ is a convex cone whose extreme directions correspond to extreme directions of $\mathcal{C}(p)$. Thus, $\mathcal{C}(p)$ could be used for calculating, by virtue of the previous lemma, the extreme directions of $H(p)$.

On the other hand, we are only interested in hyperplanes defining the boundary of $\mu_S(\mathcal{P})$ at $\mu_S(p)$. These hyperplanes are characterized in the following theorem:

Theorem 1. *Hyperplanes defining the boundary of $\mu_S(\mathcal{P})$ at $\mu_S(p)$ are given by the extreme directions of $H(p)$.*

Proof. $\mathcal{P}$ is a full dimension convex polytope, as we have previously assumed, and the dimension of S is d, then the boundary hyperplanes of $\mathcal{P}$ defining the projection intersect with the projected polytope on a face with dimension $d-1$. That is, there are d non-dependent points of $I\!\!R^n$, q_i, $i = 1, 2, \ldots, d$, belonging to the intersection of the boundary hyperplane with the projected polytope. Obviously, point $\mu_S(p)$ is one of them.

Let h be this boundary hyperplane. If h does not correspond to an extreme direction, there are two hyperplanes of $H(p)$,

$$h_1 : \sum_{j\in J} \beta_{1j} x_j = \sum_{j\in J} \beta_{1j} p_j$$

$$h_2 : \sum_{j\in J} \beta_{2j} x_j = \sum_{j\in J} \beta_{2j} p_j$$

such that h is a convex combination of h_1 and h_2.

$$h : \sum_{j\in J} (\alpha\beta_{1j} + (1-\alpha)\beta_{2j}) x_j = \sum_{j\in J} (\alpha\beta_{1j} + (1-\alpha)\beta_{2j}) p_j \qquad 0 \leqslant \alpha \leqslant 1$$

Since h goes through the points q_i, $i = 1, 2, \ldots, d$, convex combination goes through these points as well,

$$\sum_{j\in J} (\alpha\beta_{1j} + (1-\alpha)\beta_{2j}) q_{ij} = \sum_{j\in J} (\alpha\beta_{1j} + (1-\alpha)\beta_{2j}) p_j, \qquad \forall\, i = 1, 2, \ldots, d.$$

Hence

$$\sum_{j\in J} (\alpha\beta_{1j} + (1-\alpha)\beta_{2j})(q_{ij} - p_j) = 0, \qquad \forall\, i = 1, 2, \ldots, d$$

and

$$\alpha \sum_{j\in J} \beta_{1j}(q_{ij} - p_j) + (1-\alpha) \sum_{j\in J} \beta_{2j}(q_{ij} - p_j) = 0, \qquad \forall\, i = 1, 2, \ldots, d \tag{2}$$

On the other hand, h_1 and h_2 belong to $H(p)$, so,

$$\left.\begin{array}{l} \sum_{j\in J} \beta_{1j}(q_{ij} - p_j) \leqslant 0 \\ \sum_{j\in J} \beta_{2j}(q_{ij} - p_j) \leqslant 0 \end{array}\right. \qquad \forall\, i = 1, 2, \ldots, d$$

Therefore, if equation (2) is verified

$$\left.\begin{array}{l} \sum_{j\in J} \beta_{1j}(q_{ij} - p_j) = 0 \\ \sum_{j\in J} \beta_{2j}(q_{ij} - p_j) = 0 \end{array}\right. \qquad \forall\, i = 1, 2, \ldots, d$$

Thus, h_1 and h_2 go through the d non-dependent points q_i, hence, both are equal to h. Then, h is given by an extreme direction of $H(p)$

In this way, the support hyperplanes of $\mu_S(\mathcal{P})$ at $\mu_S(p)$ are given by the extreme directions of $H(p)$ which could be calculated from the extreme directions of $C(p)$.

Now, we have another problem, there are some cases in which the projection of an extreme point, $\mu_S(p)$, could not be an extreme point of $\mu_S(\mathcal{P})$. The problem appears when the number of support hyperplanes of $\mu_S(\mathcal{P})$ that intersect at $\mu_S(p)$ is smaller than the dimension of the subspace S where the projection is made. In order to avoid this problem, if the dimension of S is d, the set of extreme directions of $H(p)$ must contain d elements.

So, the algorithm will calculate the set of extreme directions of $C(p)$, $\mathcal{D}$, (which corresponds to the extreme directions of $H(p)$) and, then, it will verify that $card(\mathcal{D}) = d$. If this does not occur, the extreme point p must be rejected and the algorithm must move to an adjacent extreme point.

Let us see an example:

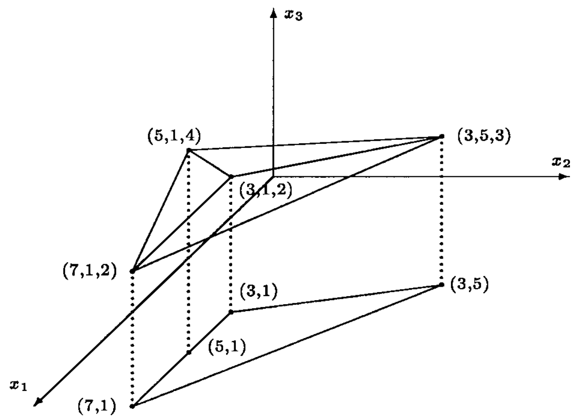

Fig. 1. Projection example

Example 1 *Consider the convex polytope drawn in Figure 1. We want to project this polytope over plane* $x_3 = 0$. *Hyperplane* $x_2 = 1$ *is a support hyperplane that is orthogonal to the half-space* $x_3 = 0$ *and it goes through extreme point* $(5, 1, 4)$. *Therefore, once the point* $(5, 1, 4)$ *has been selected, the set of extreme directions* $\mathcal{D} = \{(0, 1)\}$ *is determined. This means that, in the projection space, only one support hyperplane goes through projection of* p. *Then, the projection of point* p *cannot be an extreme point of the projection of the convex polytope.*

In the example, the point p *is projected on the point* $(5, 1)$ *belonging to the segment given by the extreme points* $(3, 1)$ *and* $(7, 1)$, *i.e., it is not an extreme point of the projection.*

Projection of $\mathcal{P}$ on S is based on determining the support hyperplanes of $\mathcal{P}$ orthogonal to S going through an extreme point p, i.e., the elements of $H(p)$. Thus, we could change the projection subspace without more than to change the form of the elements of $H(p)$, or what is the same thing, the general form of the elements of $\Gamma(p)$. Therefore, the projection algorithm could be generalized to an algorithm projecting $\mathcal{P}$ on any subspace S. In order to do so, firstly, the general definition of hyperplanes orthogonal to S going through an extreme point of $\mathcal{P}$ (elements of $H(p)$) must be obtained providing it for the algorithm in the functional form (elements of $\Gamma(p)$).

Example 2 *Consider the convex polytope* $\mathcal{P}$ *of* $\mathbb{R}^3$ *given by the set of constraints:*

$$\begin{aligned} &x_3 \geqslant 10 \\ &3x_1 - x_2 \geqslant 0 \\ &3x_1 - 5x_2 \leqslant 0 \\ &x_1 + x_2 + 2x_3 \leqslant 28 \end{aligned}$$

Suppose that we want to project it on the subspace S given by:

$$x_1 + x_2 = 10$$

The hyperplanes orthogonal to S going through an extreme point $p = (p_1, p_2, p_3)$ of the convex polytope are given by the following expression:

$$\alpha(x_1 - x_2) + \beta x_3 = \alpha(p_1 - p_2) + \beta x_3$$

and the functional form will be

$$\gamma_{\alpha,\beta}(x_1, x_2, x_3) = \alpha(x_1 - x_2) + \beta x_3$$

From this functional form, the algorithm will calculate the convex cone $C(p)$. So, for instance, the convex cone for the extreme point $(2, 6, 10)$ of $\mathcal{P}$ will be:

$$\begin{gathered} \alpha + \beta \leqslant 0 \\ \alpha \leqslant 0 \end{gathered}$$

whose extreme directions are:

$$\mathcal{D} = \{(-1, 1), (0, -1)\}.$$

Then, the extreme hyperplanes defining $\mu_S(\mathcal{P})$ at $\mu_S(2, 6, 10)$ will be given by

$$\begin{gathered} -x_1 + x_2 + x_3 = 14 \\ x_3 = 10 \end{gathered}$$

and $\mu_S(2, 6, 10) = (3, 7, 10)$.

Before looking at the algorithm implementation, we are going to study some typical problems that appear when we work with convex polytopes.

2.2. *Typical problems: full dimension, redundancy and degeneration*

2.2.1. *Full Dimension*

This problem appears when the dimension of the convex polytope $\mathcal{P}$ is not n, i.e., $\mathcal{P}$ is contained in a subspace given by the intersection of a set of hyperplanes. For our algorithm, this problem is important only if the hyperplanes are orthogonal to S because, otherwise, these are not considered.

Suppose we have a hyperplane orthogonal to S, h, containing to the convex polytope $\mathcal{P}$, then $h \cap S$ contains the projection of $\mathcal{P}$ to S, and therefore $\{h\} \in H(p)$ for any extreme point p of $\mathcal{P}$. This case can be detected in the following way:

Theorem 2. *For any $p \in Ext(\mathcal{P})$, if a linear subspace is contained in the convex cone $H(p)$, this subspace is constituted by the hyperplanes orthogonal to S containing $\mathcal{P}$.*

Proof. Let

$$h : \sum_{j \in J} \beta_j x_j = \sum_{j \in J} \beta_j p_j$$

be any hyperplane of the linear subspace contained in $H(p)$. It is verified

$$\sum_{j \in J} \beta_j q_j \leqslant \sum_{j \in J} \beta_j p_j, \text{ for all } q \in \mathcal{P}$$

because h belongs to $H(p)$.

If $\alpha \leqslant 0$, then

$$\alpha \sum_{j \in J} \beta_j q_j \geqslant \alpha \sum_{j \in J} \beta_j p_j, \text{ for all } q \in \mathcal{P} \tag{3}$$

Since h belongs to a linear subspace, for any $\alpha \in I\!R$, αh belongs as well. Therefore, $\alpha h \in H(p)$ and hence

$$\alpha \sum_{j \in J} \beta_j q_j \leqslant \alpha \sum_{j \in J} \beta_j p_j, \text{ for all } q \in \mathcal{P} \tag{4}$$

Thus, from 3 and 4,

$$\sum_{j \in J} \beta_j q_j = \sum_{j \in J} \beta_j p_j, \text{ for all } q \in \mathcal{P}.$$

That is, $\mathcal{P}$ is contained in h

Corollary 1 *The convex polytope $\mathcal{P}$ is contained in a set of hyperplanes orthogonal to S if and only if for any $p \in Ext(\mathcal{P})$, a linear subspace given by the linear combinations of the hyperplanes orthogonal to S containing $\mathcal{P}$ is included in $H(p)$.*

Proof. Let us see the direct implication. If K is the set of hyperplanes orthogonal to S containing $\mathcal{P}$, for any extreme point p of $\mathcal{P}$ and any hyperplane h of K, $h \in H(p)$ is verified. That is, $K \subseteq H(p)$ for all $p \in Ext(\mathcal{P})$. Obviously, for all $h_1, h_2 \in K$, $\alpha h_1 + \beta h_2 \in H(p)$ for all $p \in Ext(\mathcal{P})$ and $\alpha, \beta \in I\!R$. Then, a linear subspace given by the linear combinations of the hyperplanes orthogonal to S containing $\mathcal{P}$ is included in $H(p)$.

Inverse implication is straightforward from previous theorem

The last corollary is the key for detecting and solving the problem. For any extreme point p of $\mathcal{P}$ the convex cone $\mathcal{C}(p)$ is calculated. The maximum linear subspace contained in $\mathcal{C}(p)$ is detected by Galperin's algorithm[10]. If the result is the empty subset, problem does not appear, i.e., $\mathcal{P}$ is a full dimension convex polytope. Otherwise, Galperin's algorithm provides a base for the linear subspace. Each element of the base determines, according to the linear mapping established in lemma , a hyperplane orthogonal to S containing $\mathcal{P}$. If h is one of these hyperplanes, the projection of $\mathcal{P}$ is included in $h \cap S$, therefore, h can be eliminated and the dimension of the problem is reduced by one variable. Value of this variable is calculated from remaining variables and h. An algorithm for detecting and solving this problem is given bellow:

Firstly, a procedure for detecting the set of hyperplanes orthogonal to $S = \{x_j, j \in J\}$ containing the polytope $\mathcal{P}$ from convex cone $\mathcal{C}(p)$ for any $p \in Ext(\mathcal{P})$ is given. The following notations are used:

$\mathcal{C}$: Represents the convex cone $\mathcal{C}(p)$.

$\mathcal{S}$: Represents the set of variables on which the projection is performed, $S = \{x_j, j \in J\}$.

LH: List of hyperplanes orthogonal to S containing $\mathcal{P}$.

b: Vector of $I\!R^n$ representing the set of values assigned to elements of U by the right-hand side vector of a Simplex algorithm tableau.

γ_α : Represents the general functional form of the elements in $\Gamma(p)$.

Procedure Equality_Constraints $(\mathcal{C}, b, S, \gamma_\alpha, LH)$

1.- Calculate using Galperin's algorithm the maximum linear subspace, $\mathcal{L}$, included in $\mathcal{C}$.

2.- If $\mathcal{L} \neq \emptyset$, then

2.1.- $\mathcal{L}$ is given by a set of points of $I\!R^d$

$$\mathcal{L} = \{\alpha^i,\ i = 1, 2, \ldots, k\}$$

whose associated hyperplanes

$$\gamma_{\alpha^i}(u) = \gamma_{\alpha^i}(b),\ \ i = 1, 2, \ldots, k$$

are verified in equality for all the points of $\mathcal{P}$.

2.2.- For each α^i, keep the hyperplane

$$\sum_{v \in S} \alpha^i_v v = \gamma_{\alpha^i}(b)$$

in LH.

2.3.- Since $\mathcal{P}$ is contained into the intersection of all the hyperplanes of LH, it is possible to eliminate one variable of S for each hyperplane of LH. So, if the set of eliminated variables is noted by H_S and the values of the variables of $S - H_S$ are known, values of the variables of H_S may be calculated by solving the system of equations given by the hyperplanes of LH. Put $S = S - H_S$.

So, the algorithm for detecting the hyperplanes orthogonal to S containing $\mathcal{P}$ is the following:

Procedure Orthogonal_hyperplanes_in_equality$(\mathcal{P}, S, \gamma_\alpha, LH)$

1.- Maximize over $\mathcal{P}$, using the Simplex algorithm, the function:

$$f(u) = \sum_{u \in U} u$$

storing the optimal tableau.

2.- Define b as the vector representing the set of values assigned to elements of U by the right-hand side vector of a Simplex tableau.

3.- Change the objective function by $\gamma_\alpha(u)$, obtaining the new optimality vector $z_k - c_k$ given as a function of α.

4.- Construct the convex cone $\mathcal{C}$ with the constraints $z_k - c_k \geqslant 0$ for all k associated with the non-basic variables in the initial optimal tableau.

5.- Call the procedure **Equality_Constraints**$(\mathcal{C}, b, S, \gamma_\alpha, LH)$.

2.2.2. *Degeneration*

For each optimal tableau, an extreme point of $\mathcal{P}$ is obtained. This extreme point is given by the intersection of hyperplanes corresponding to slack and artificial non-basic variables[19]. From this point of view, an extreme point may be considered like a configuration of hyperplanes. So, the extreme points can be seen in two ways: like the proper points or like configurations of hyperplanes.

If the convex polytope is not degenerated, both concepts are the same. Each extreme point is given by only one configuration of hyperplanes. However, in general, this is not true because an extreme point can be generated by several configurations. To avoid this problem, in our algorithm, we do not work on the extreme points but on the configurations of hyperplanes. In this way, two configurations are independently handled even though the same extreme point is produced from both.

2.2.3. *Redundancy*

The redundant half-spaces[16] increase the complexity of the projection process since they give rise to non-necessary configurations of hyperplanes. Therefore, for an optimal running of the algorithm, they could be eliminated. However, the cost of the elimination process make its elimination prohibitive. So, the elimination should not be carried out excepting the case that a fast syntactic algorithm is used[12].

In a final remark, if we only consider the extreme hyperplanes in each projected point, redundant half-spaces are never obtained because redundant hyperplanes are never extreme hyperplanes.

2.3. *Projection Algorithm*

Suppose we have a convex polytope $\mathcal{P}$ of $I\!R^n$, represented on a set of coordinates $\{x_1, x_2, \ldots, x_n\}$, given by a set of linear inequalities

$$\mathcal{P} = \{x \in I\!R^n \;:\; \sum_{j=1}^{n} \lambda_{ij} x_j \leqslant b_i, i = 1, 2, \ldots, s\}$$

and its projection on $S = \{x_j, j \in J\}$, $J \subseteq \{1, 2, \ldots, n\}$ is wished.

Firstly, the trivial case is attached: The projection subspace is one-dimensional. Then, the procedure is:

Procedure Trivial_projection($\mathcal{P}, S, LS, \gamma_\alpha$).

1.- Maximize and minimize the function $\gamma_1(u)$*over the convex polytope $\mathcal{P}$.

2.- Let Max and Min be the maximum and minimum of $\gamma_1(u)$ over $\mathcal{P}$.

3.- Let v be the only element of S. Then, the half-spaces

$$v \leqslant Max$$

$$v \geqslant Min$$

are kept in LS.

So, the projection of $\mathcal{P}$ is a segment of S with Min and Max as extreme points.

Once trivial case has been studied, our algorithm is a procedure with the following parameters:

- The convex polytope $\mathcal{P}$ that we want to project.
- The space where the convex polytope is defined, denoted by U.
- The projection subspace, denoted by S.
- The general form of the functions of $\Gamma(p)$, denoted by γ_α.

The output of the algorithm is stored in the variables:

- LH: Orthogonal hyperplanes contaning $\mathcal{P}$.
- LS: List of half-spaces defining the projection of $\mathcal{P}$ to S.
- LV: List of extreme points of the projection of $\mathcal{P}$ to S.

Furthermore, the algorithm uses the working variables:

- LUT: List of examined tables.
- $LNUT$: List of tables that have not still been examined.

*$\gamma_1(u)$ represents the function $\gamma_\alpha(u)$, where α is a 1's vector, $\alpha = (1_v)_{v\in S}$

Procedure Projection$(U, S, \mathcal{P}, \gamma_\alpha)$**.**

1.- $\mathcal{T} = S$.

2.- $LH = \emptyset$, $LS = \emptyset$, $LV = \emptyset$, $LUT = \emptyset$, $LNUT = \emptyset$.

3.- Call **Orthogonal_hyperplanes_in_equality**$(\mathcal{P}, S, \gamma_\alpha, LH)$**.**

4.- If $card(S) = 1$, $S = \{v_j\}$, then

4.1.- Call **Trivial_projection**$(\mathcal{P}, S, LS, \gamma_\alpha)$.

4.2.- Add to LS the non-negativity conditions for the eliminated variables, i.e.,

$$LS = LS \cup \{v_k \geqslant 0, \ \forall\, v_k \in \mathcal{T} - S\}$$

4.3.- End.

5.- Maximize over $\mathcal{P}$ the function $\gamma_\delta(u)$, where δ is a vector of $\mathbb{R}^d$, defined by

$$\delta = (\delta_v)_{v \in \mathcal{T}} = \begin{cases} 1 & \text{for all } v \in S \\ 0 & \text{otherwise} \end{cases}$$

6.- Put BV as the set of basic variables.

7.- Build the new function γ_α changing α_v to 0 for all $v \in \mathcal{T} - S$.

8.- Put b as the vector of the values assigned to the elements of U by the right-hand side vector of the tableau.

9.- Values of α_v holding the optimality:

9.1.- Replace the objective function by $\gamma_\alpha(u)$ (given in point 6) obtaining the new optimality vector $z_k - c_k$ depending on $(\alpha_v)_{v \in S}$.

9.2.- Build the convex cone $\mathcal{C}$ given by the constraints $z_j - c_j \geqslant 0$ for all j corresponding to non-basic variables.

9.3.- Using Galperin's algorithm, calculate the set of extreme directions for $\mathcal{C}$. Let $\mathcal{D}$ be the set of points that are pointed by the vectors corresponding to these extreme directions.

9.4.- If $card(\mathcal{D}) < card(S)$ go to step 10.

10.- Obtaining vertices:

10.1.- For all $v_j \in S$,

10.1.1.- Calculate the vector of $\mathbb{R}^d$, δ, as

$$\delta = (\delta_v)_{v \in \mathcal{T}} = \begin{cases} 1 & \text{for all } v = v_j \\ 0 & \text{otherwise} \end{cases}$$

10.1.2.- Put $Ver(v_j) = \gamma_\delta(b)$.

10.2.- Complete the coordinates of Ver as a point of $\mathcal{T}$ by means of the LH hyperplanes.

10.3.- If Ver is not in the list of vertices LV, keep it.

11.- For each $\alpha \in \mathcal{D}$, calculate the associated half-space in the following way:

a) If $\alpha_v \leqslant 0$ for all $v \in S$, store the half-space

$$\sum_{v\in S} -\alpha_v v \geqslant \sum_{v\in S} -\alpha_v Ver(v)$$

in LS, if it does not appear.

b) Otherwise, store the half-space

$$\sum_{v\in S} \alpha_v v \leqslant \sum_{v\in S} \alpha_v Ver(v)$$

in LS, if it does not appear.

12.- If the lists of half-spaces and vertices have not been both updated go to step 13.

13.- For each $\alpha \in \mathcal{D}$,

13.1.- Obtain the new vector $z_k - c_k$, by substituting the values of α_v by the values of the point actually studied in the optimal tableau. If there is some non-basic column with $z_k - c_k = 0$ then:

13.2.- For each non-basic column with $z_k - c_k = 0$,

13.2.1.- Calculate I as the variable corresponding to the column studied.

13.2.2.- Calculate O as the variable satisfying feasibility criterion of the Simplex algorithm:

$$\underset{j\ :a_{jr}>0}{\text{Min}} \left[\frac{b_j}{a_{jr}}\right]$$

where r is the index corresponding to I, a_{jr} are the elements of the matrix of coefficients and b_j the elements of the right-hand side vector.

13.2.3.- Put $NBV = \Big(BV - \{O\}\Big) \cup \{I\}$.

13.2.4.- If NBV does not belong to $LUT \cup LNUT$, add it to $LNUT$.

14.- Add BV to LUT.

15.- If $LNUT$ is empty the alsorithm ends.

16.- Put BV as the value of the last element of $LNUT$.

17.- Delete this last element of $LNUT$.

18.- If BV does not belong to LUT:

18.1.- Obtain, by pivoting, the Simplex tableau associated to the set of basic variables given by BV.

18.2.- Go back to step 7 with the new tableau.

19.- If BV belongs to LUT, go to step 14.

3. Marginalization of a convex polytope of probabilities

As we know, final goal of this paper is to obtain the marginalization of a convex polytope of probabilities as a projection. In this point, the subspace where the projection must be performed for obtaining the marginalization is calculated. Moreover, we shall establish the general functional form, γ_α, needed by the projection algorithm.

Suppose $J \subseteq I \subseteq \{1, 2, \ldots, m\}$ and consider the subsets of variables $(X_i)_{i\in I}$ and $(X_j)_{j\in J}$ respectively defined over $U_I = \prod_{i\in I} U_i$ and $U_J = \prod_{j\in J} U_j$. Clearly, the marginalization of X_I to X_J is the function ($n = \prod_{i\in I} r_i$, $d = \prod_{j\in J} r_j$):

$$\begin{array}{ll} M: & \mathbb{R}^n \longrightarrow \mathbb{R}^d \\ & M(x) = (a_{vu})_{\substack{u\in U_I \\ v\in U_J}} x^t \end{array}$$

where x^t denotes the transposed vector of x and (a_{vu}) is the matrix whose elements are:

$$a_{vu} = \begin{cases} 1 & \text{if } u^{\downarrow I} = v \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

In short, we denote by U and V to U_I and U_J, respectively. Consider the d vectors $a_v = (a_{vu})_{u\in U}$ for all $v \in V$, of $\mathbb{R}^n$. These vectors are orthogonal and, hence, non-dependent. $n - d$ vectors b_j, $j = d+1, d+2, \ldots, n$ can always be found in order to complete the a_v vectors for constituting an orthogonal base of $\mathbb{R}^n$.

The next mapping is defined:

$$\begin{array}{ll} A^e: & \mathbb{R}^n \longrightarrow \mathbb{R}^n \\ & A^e(x) = \begin{pmatrix} (a_v)_{v\in V} \\ (b_j)_{j=d+1\cdots n} \end{pmatrix} x^t \end{array}$$

If the projection of $\mathbb{R}^n$ to $\mathbb{R}^d$ is denoted by $\mu_{\mathbb{R}^d}$,

$$M = \mu_{\mathbb{R}^d} \circ A^e$$

is verified.

So, the marginalization of $\mathcal{P}$ is

$$M(\mathcal{P}) = \mu_{\mathbb{R}^d}\big(A^e(\mathcal{P})\big)$$

The mapping A^e makes a change of base into $\mathbb{R}^n$. Then, $A^e(\mathcal{P})$ is the expression of $\mathcal{P}$ in the new base given by a_v, $v \in V$ and b_j, $j = d+1, d+2, \ldots, n$. That is, there are no changes in the convex polytope, but only in the coordinates of the points: the points and the faces are the same ones.

The next step for performing marginalization is the projection to the first d coordinates. That is, we have to make a projection over

$$\{x \in I\!R^n \ : x_{d+i} = 0,\ i = 1, 2, \ldots, n-d\}$$

Since there are no changes in the convex polytope, these steps can be resumed in one: the projection on the subspace

$$\begin{aligned} S &= \{x \in I\!R^n \ : \left(A^e(x)\right)_{d+i} = 0,\ i = 1, 2, \ldots, n-d\} = \\ &= \{x \in I\!R^n \ :< b_j, x >= 0,\ j = d+1, d+2, \ldots, n\} \end{aligned} \tag{6}$$

This projection gives us a convex polytope of $I\!R^n$ contained in the subspace S. On the other hand, the set of half-spaces defining this polytope and the one defined by marginalizing are the same one. So, we have expressed the marginalization as a projection on S.

At this moment, the general functional form γ_α of $\Gamma(p)$ can be fixed for the subspace S. Let $(\lambda_1, \lambda_2, \ldots \lambda_n)$ be the normal vector of any hyperplane of $I\!R^n$. If it is orthogonal to S, the vector $(\lambda_1, \lambda_2, \ldots \lambda_n)$ must be orthogonal to each vector b_j, $j = d+1, d+2, \ldots, n$.

Since the set $\left\{(a_i)_{i=1,2,\ldots,d}, (b_j)_{j=d+1,d+2,\ldots,n}\right\}$ is an orthogonal base of $I\!R^n$, the vector $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ must be a linear combination of the vectors a_i, $i = 1, 2, \ldots, d$. That is,

$$(\lambda_1, \lambda_2, \ldots \lambda_n) = \sum_{j \in J} \alpha_i a_i, \qquad \alpha_i \in I\!R$$

Thus, the functions that we have to maximize over $\mathcal{P}$ are

$$\gamma_\alpha(u) = \sum_{i=1}^{d} \alpha_i < a_i, u >$$

But the d vectors a_i are given as a set of vectors $a_v = (a_{vu})_{u \in U}$, where

$$a_{vu} = \begin{cases} 1 & \text{if } u^{\downarrow J} = v \\ 0 & \text{otherwise} \end{cases}$$

Therefore, if the general vector $\alpha = (\alpha_v)_{v \in V}$ of $I\!R^d$ is considered,

$$\gamma_\alpha(u) = \alpha (a_v)_{v \in V} u^t$$

is the function to be introduced into the algorithm.[†]

So, if we have the variable $(X_i)_{i \in I}$, over which the information is given by the convex polytope $\mathcal{P}$ defined in $I\!R^n$, in order to perform the marginalization of $\mathcal{P}$ to the variable $(X_j)_{j \in J}$, $J \subseteq I$ we have to call the projection procedure in the following way:

If $U = \prod_{i \in I} U_i$, $V = \prod_{j \in J} U_j$ and $\gamma_\alpha(u) = \alpha (a_v)_{v \in V} u^t$, then

$$\textbf{Projection}(U, V, \mathcal{P}, \gamma_\alpha).$$

[†] $\alpha (a_v)_{v \in V} u^t$ determines the matrix product of α by $(a_v)_{v \in V}$ and by the vector transposed of u

Example 3 *Suppose we have a two-dimensional variable $X \times Y$ defined over $\mathcal{X} \times \mathcal{Y} = \{x_1y_1, x_1y_2, x_1y_3, x_2y_1, x_2y_2, x_2y_3\}$. Assume that the information over $X \times Y$ is given by a convex polytope of probabilities of $I\!R^6$, $\mathcal{P}$, represented by the constraints:*

$$\begin{gathered} p(x_1y_2) \leqslant 0.8 \\ p(x_1y_3) \leqslant 0.4 \\ p(x_2y_1) \leqslant 0 \\ p(x_2y_2) = 0.1 \\ p(x_1y_2) + p(x_1y_3) + p(x_2y_3) \leqslant 0.9 \\ p(x_1y_1) + p(x_2y_1) + p(x_2y_2) + p(x_2y_3) \leqslant 0.3 \\ p(x_1y_1) + p(x_1y_2) + p(x_1y_3) + p(x_2y_1) + p(x_2y_2) + p(x_2y_3) = 1 \\ p(x_1y_1), p(x_1y_2), p(x_1y_3), p(x_2y_1), p(x_2y_2), p(x_2y_3) \geqslant 0 \end{gathered}$$

If we want to marginalize this information to the variable Y defined over $V = \{y_1, y_2, y_3\}$ and suppose $U = \mathcal{X} \times \mathcal{Y}$, i.e., $u = (x_1y_1, x_1y_2, x_1y_3, x_2y_1, x_2y_2, x_2y_3)$, the function to introduce is:

$$\begin{aligned} \gamma_\alpha(u) &= \gamma_\alpha(u) = \alpha(a_v)_{v \in V} u^t = (\alpha_{y_1}, \alpha_{y_2}, \alpha_{y_3}) \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1y_1 \\ x_1y_2 \\ x_1y_3 \\ x_2y_1 \\ x_2y_2 \\ x_2y_3 \end{pmatrix} = \\ &= \alpha_{y_1}\big(p(x_1y_1) + p(x_2y_1)\big) + \alpha_{y_2}\big(p(x_1y_2) + p(x_2y_2)\big) + \\ &+ \alpha_{y_3}\big(p(x_1y_3) + p(x_2y_3)\big) \end{aligned}$$

The convex polytope could be simplified before appliying the algorithm. So the new set of constraints defining the convex polytope will be as follows:

$$\begin{gathered} p(x_1y_2) \leqslant 0.8 \\ p(x_1y_3) \leqslant 0.4 \\ p(x_1y_2) + p(x_1y_3) + p(x_2y_3) \leqslant 0.9 \\ p(x_1y_1) + p(x_2y_3) \leqslant 0.2 \\ p(x_1y_1) + p(x_1y_2) + p(x_1y_3) + p(x_2y_3) = 0.9 \\ p(x_1y_1), p(x_1y_2), p(x_1y_3), p(x_2y_3) \geqslant 0. \end{gathered}$$

Since the values of the variables $p(x_2y_1)$ and $p(x_2y_2)$ are fixed, respectively 0 *and* 0.1*.*

The procedure **Orthogonal_hyperplanes_in_equality**$(\mathcal{P}, V, \gamma_\alpha, LH)$ *produces*

$$\begin{gathered} LH = \{y_1 + y_2 + y_3 = 1\} \\ V = \{y_1, y_2\} \end{gathered}$$

Then, we maximize the function $\gamma_\delta(u) = p(x_1y_1) + p(x_1y_2) + p(x_2y_1) + p(x_2y_2)$ by the Simplex algorithm. Once the optimal tableau has been obtained, we replace *the function* γ_δ *by the general projection function* $\gamma_\alpha(u) = \alpha_{y_1}\big(p(x_1y_1)+p(x_2y_1)\big) + \alpha_{y_2}\big(p(x_1y_2)+p(x_2y_2)\big)$. *So, the tableau in Table 1 is achieved. From this tableau,*

	$p(x_1y_3)$	$p(x_2y_3)$	h_4	
h_1	-1	0	1	0.1
h_2	1	0	0	0.4
h_3	0	1	1	0.2
α_{y_1} $p(x_1y_1)$	0	1	1	0.2
α_{y_2} $p(x_1y_2)$	1	0	-1	0.7
	α_{y_2}	α_{y_1}	$\alpha_{y_1}-\alpha_{y_2}$	

Table 1. Initial optimal tableau in the marginalization process

the convex cone $\mathcal{C}$

$$\begin{aligned} \alpha_{y_1} &\geqslant 0 \\ \alpha_{y_2} &\geqslant 0 \\ \alpha_{y_1}-\alpha_{y_2} &\geqslant 0, \end{aligned} \tag{7}$$

is obtained. The extreme directions are $\mathcal{D} = \{(1,1),(1,0)\}$ *and the projected extreme point is* $(0.2, 0.8, 0)$. *Furthermore, the marginalization half-spaces*

$$\begin{aligned} y_1 + y_2 &\leqslant 1 \\ y_2 &\leqslant 0.2 \end{aligned}$$

are calculated. If we change the values of α *by the values of points in* $\mathcal{D}$, *the new vectors* $z_k - c_k$ *drawn in Table 2 are calculated.*

	$p(x_1y_3)$	α_{y_1} $p(x_2y_1)$	$p(x_2y_3)$	h_5	
h_1	-1	0	0	1	0.1
h_2	1	0	0	0	0.4
h_3	0	1	0	0	0
h_4	0	0	1	1	0.2
α_{y_1} $p(x_1y_1)$	0	1	1	1	0.2
α_{y_2} $p(x_2y_2)$	0	0	0	0	0.1
α_{y_2} $p(x_1y_2)$	1	0	0	-1	0.7
	α_{y_2}	0	α_{y_1}	$\alpha_{y_1}-\alpha_{y_2}$	
	1	0	1	0	$(1,1)$
	0	0	1	1	$(1,0)$

Table 2. Initial tableau after substitution of α by the values of points in $\mathcal{D}$.

In this way, the set

$$\{h_1, h_2, h_3, h_4, p(x_1y_1), p(x_2y_2), p(x_1y_2)\}$$

is added to the list LUT *and the sets*

$$\{h_1, h_2, p(x_2y_1), h_4, p(x_1y_1), p(x_2, y_2), p(x_1y_2)\},$$
$$\{h_6, h_2, h_3, h_4, p(x_1y_1), p(x_2, y_2), p(x_1y_2)\},$$
$$\{h_1, p(x_1y_3), h_3, h_4, p(x_1y_1), p(x_2, y_2), p(x_1y_2)\}$$

to the list $LNUT$.

So, last element of $LNUT$

$$\{h_1, p(x_1y_3), h_3, h_4, p(x_1y_1), p(x_2, y_2), p(x_1y_2)\}$$

is chosen and eliminated. The Simplex tableau associated with it is shown in Table 3.

	h_2	α_{y_1} $p(x_2y_1)$	$p(x_2y_3)$	h_5	
h_1	1	0	0	1	0.5
0 $p(x_1y_3)$	1	0	0	0	0.4
h_3	0	1	0	0	0
h_4	0	0	1	1	0.2
α_{y_1} $p(x_1y_1)$	0	1	1	1	0.2
α_{y_2} $p(x_2y_2)$	0	0	0	0	0.1
α_{y_2} $p(x_1y_2)$	-1	0	0	-1	0.3
	$-\alpha_{y_2}$	0	α_{y_1}	$\alpha_{y_1} - \alpha_{y_2}$	

Table 3. Second tableau of the marginalization process

From this tableau, we apply the process again and so on. At the end of the algorithm

$$Vertices = \big\{(0.2, 0.8, 0), (0.2, 0.4, 0.4), (0, 0.4, 0.6), (0.1, 0.9, 0), (0, 0.9, 0.1)\big\}$$
$$LH = \big\{y_1 + y_2 + y_3 = 1\big\}$$
$$LS = \big\{y_1 + y_2 \leqslant 1, y_1 \leqslant 0.2, y_2 \geqslant 0.4, y_1 \geqslant 0, y_2 \geqslant 0.9\big\}$$

are obtained.

4. Final remarks

When a system works with sets of variables, two basic operations are needed: combination, for integrating pieces of information from several sources, and marginalization for focusing the information on the set of variables on which we are interested.

Once the operators have been selected, the problem appears when we must choose the way in which these operators must be performed. Combinations are very easy and computationally effective if the convex polytopes are represented by sets of constraints[20], However, this is not the natural way for marginalizing. The leading problem of this paper has consisted on performing the marginalization of a convex polytope of probabilities when it was represented by a set of constraints. This marginalization has been obtained like a projection on a specific subspace. Thus, we have generalized the problem constructing an algorithm for projecting a convex polytope on any subspace, and determining the projection subspace where the marginalization is obtained.

The way of performing the combination and the marginalization is very important because, normally, on any problem, we have a lot of operations. In general, the number of combinations is greater than the number of marginalizations. This is the reason on account of which we have chosen an easier way for performing the combination against the marginalization.

References

1. Jose Enrique Cano. *Propagación de probabilidades inferiores y superiores en grafos.* PhD thesis, Universidad de Granada, Granada, 1993.
2. Jose Enrique Cano, Serafin Moral, and Juan Francisco Verdegay-López. Combination of upper and lower probabilities. In B.D. Ambrosio, Ph. Smets, and P.P. Bonissone, editors, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, pages 61–68. Morgan & Kaufmann, 1991.
3. Jose Enrique Cano, Serafin Moral, and Juan Francisco Verdegay-López. Propagation of convex sets of probabilities in directed acyclic networks. In B. Bouchon-Meunier et al., editors, *Uncertainty in Intelligent Systems*, pages 15–26. Elsevier, 1993.
4. F. Cozman. Robust analysis of bayesian networks with finitely generated convex sets of distributions. Technical Report CMU-RI-TR96-41, Carnegie Mellon University, 1996.
5. Luis Miguel de Campos. *Caracterización y Estudio de Medidas e Integrales Difusas a partir de Probabilidades.* PhD thesis, Universidad de Granada, Granada, 1987.
6. Luis Miguel de Campos and Serafín Moral. Independence concepts for convex sets of probabilities. In Ph. Besnard and S. Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 108–115. Morgan & Kaufmann, 1995.
7. A.P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.
8. D. Dubois and H. Prade. A survey of belief revision and updating rules in various uncertainty models. *International Journal of Intelligent Systems*, 9:61–100, 1994.
9. Tomas Gal. *Postoptimal Analysis. Parametric Programming and Related Topics.* McGraw Hill, New York, 1979.
10. A. M. Galperin. The general solution of a finite system of linear inequalities. *Mathematics of Operations Research*, 1(2):185–196, 1976.
11. T. Huynh and Jean Louis Lassez. Practical issues on the projection of polyhedral sets. Research Report RC 15872, IBM T.J. Watson Research Center, 1990.

12. J. L. Imbert and P. van Hentenryck. Redundancy elimination with a lexicographic solved form. Technical report CS-95-02, Department of computer Science, Brown University, 1995.
13. J. Y. Jaffray. Bayesian updating and belief funcions. In *Extended abstract of the 3rd IPMU conference*, pages 449–451, Paris, 1990.
14. Catherine Lassez and Jean Louis Lassez. Quantifier elimination for conjunction of linear constraints by a convex hull algorithm. In B. R. Donald, D. Kapur, and J. L. Mundy, editors, *Symbolic and Numerical Computation for Artificial Intelligence*, Computational Mathematics and Applications, pages 103–119. Academic Press, London, 1992.
15. I. Levi. Imprecision and indeterminacy in probability judgement. *Philosophy of Science*, 52:390–409, 1985.
16. T. H. Mattheiss. An algorithm for determining irrelevant constraints and all vertices in systems of linear inequalities. *Operations Research*, 21:247–260, 1973.
17. Franco P. Preparata and Michael Ian Shamos. *Computational geometry.* Springer Verlag, New York, 1985.
18. R. Tyrrell Rockafellar. *Convex Analysis.* Princeton University Press, New Jersey, 1972.
19. M. Simonnard. *Programación lineal.* Paraninfo, Madrid, 1972.
20. Juan Francisco Verdegay-López. *Representación y combinación de la información con incertidumbre usando convexos de probabilidades.* PhD thesis, Departamento de Ciencias de la Computación e Inteligencia Artificial, ETSII, Universidad de Granada, Granada, 1998.
21. P. Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83:1–58, 1996.
22. Peter Walley. *Statistical reasoning with imprecise probabilities.* Chapman and Hall, London, 1991.
23. R.R. Yager. On the Dempster-Shafer framework and the new combinations rules. *Information and Science*, 41:93–137, 1987.