LE MATEMATICHE
Vol. LV (2000) – Fasc. I, pp. 91–101

ALGORITHMS FOR THE EXTENSION OF PRECISE AND IMPRECISE CONDITIONAL PROBABILITY ASSESSMENTS: AN IMPLEMENTATION WITH MAPLE V

VERONICA BIAZZO

In this paper, we illustrate an implementation with Maple V of some procedures which allow to exactly propagate precise and imprecise probability assessments. The extension of imprecise assessments is based on a suitable generalization of the concept of coherence of de Finetti. The procedures described are supported by some examples and relevant cases.

1. Introduction.

The analysis of many real problems, involving uncertainty, often requires some probabilistic assessments on a suitable family $\mathcal{K}$ of random quantities. Such family has not necessarily any particular algebraic structure, therefore the de Finetti's methodology is the most suitable one. Following this approach we examine some procedures by means of which some given conditional probability assessments can be propagated in a coherent way to a further conditional event. Based on the linear programming technique, the checking of the coherence and the extension of precise or imprecise assessments have been studies in many papers (see for example [5], [6], [7], [10], [12]). In [2] the fundamental theorem of de Finetti has been applied to conditional events and some theoretical results have been obtained. Moreover, in the quoted paper an

Entrato in Redazione il 25 ottobre 2000.

algorithm has been proposed to determine the interval $[p', p'']$. The consistency problem when an imprecise probability assessments.

$$\mathcal{A}_n : P(E_i|H_i \geq \alpha_i, \quad i = 1, \dots, n,$$

is defined on a family of n conditional events $\mathcal{F}_n = \{E_1|H_1, \dots, E_n|H_n\}$ can be examined by choosing some suitable definition of the concept of coherence (see [5], [7], [14], [15]). In particular, the definitions adopted in [5] and [7] are based on the coherence principle of de Finetti. In [14] and [15] some conditions of coherence, involving random quantities that can be interpreted as random gains, are introduced. Making a comparison ([12]) one has that the definitions adopted in [14] and [15] are stronger than that ones introduced in [5] and [7].

In [3] the extension of the results obtained in [2] to the case of imprecise assessments is examined. In particular, we have considered a suitable version of the fundamental theorem of de Finetti relative to the extension of an imprecise assessment $\mathcal{A}_n$ defined on $\mathcal{F}_n$ to a further conditional event $E_{n+1}|H_{n+1}$. In this paper we examine in detail some procedures by means of which the theoretical results obtained in [2] (and also in [3]) can be applied. A preliminary version of the implementation of these procedures, relative to the case of precise conditional probability assessments, has been given in [1].

2. Preliminaries.

Given a coherent probability assessment $\mathcal{P}_n = (p_1, \dots, p_n)$ on a family $\mathcal{F}_n = \{E_1|H_1, \dots, E_n|H_n\}$, let $E_{n+1}|H_{n+1}$ be a further conditional event. As well known, there exists an interval $[p', p''] \subseteq [0, 1]$ such that, for every $p_{n+1} \in [p', p'']$, the assessment $\mathcal{P}_{n+1} = (\mathcal{P}_n, p_{n+1}) = (p_1, \dots, p_n, p_{n+1})$ is a coherent extension of $\mathcal{P}_n$ to the family $\mathcal{F}_{n+1} = \mathcal{F}_n \cup \{E_{n+1}|H_{n+1}\} = \{E_1|H_1, \dots, E_n|H_n, E_{n+1}|H_{n+1}\}$.

Concerning the case of imprecise assessments we recall that, given a family $\mathcal{F}_n = \{E_1|H_1, \dots, E_n|H_n\}$ and a vector $\mathcal{A}_n = (\alpha_1, \dots, \alpha_n)$ of lower bounds $P(E_i|H_i) \geq \alpha_i$, with $i = 1, \dots, n$, the following definition of generalized coherence ($g - coherence$) has been adopted in [7].

Definition 1. The vector of lower bounds $\mathcal{A}_n$ on $\mathcal{F}_n$ is said g-coherent if and only if there exists a (precise) coherent assessment $\mathcal{P}_n = (p_1, \dots, p_n)$ on $\mathcal{F}_n$, with $p_i = P(E_i|H_i)$, which is consistent with $\mathcal{A}_n$, that is such that $p_i \geq \alpha_i$ for each i.

Then, based on the definition above, a necessary and sufficient condition of g-coherence for the extension of imprecise assessments will be given below. Let $\Pi_{\mathcal{A}_n}$ be the set of coherent precise assessments $\mathcal{P}_n$ on $\mathcal{F}_n$ which are consistent with $\mathcal{A}_n$. We observe that for each coherent assessment $\mathcal{P}_n$ on $\mathcal{F}_n$ there exists an interval $[p', p'']$ of the coherent extensions of $\mathcal{P}_n$ to $E_{n+1}|H_{n+1}$. In [3] it is proved that, defining $p^\circ = max_{P_n \in \Pi_{\mathcal{A}_n}} p''$, for each $\alpha \in [0, p^\circ]$ the following assessment on $\mathcal{F}_{n+1}$:

$$P(E_i|H_i \geq \alpha_i,\ i = 1, \ldots, n, \quad P(E_{n+1}|H_{n+1}) \geq \alpha$$

is coherent.

In the same way, given a g-coherent vector of upper bounds $\mathcal{B}_n = (\beta_1, \ldots, \beta_n)$ on a family $\mathcal{F}_n$ and a further conditional event $E_{n+1}|H_{n+1}$, the vector $\mathcal{B}_{n+1} = (\beta_1, \ldots, \beta_n, \beta)$ on $\mathcal{F}_{n+1}$ is g-coherent if $\beta \in [p_\circ, 1]$, where $p_\circ = min_{P_n \in \Pi_{\mathcal{B}_n}} p'$. a similar result has been obtained in [12] using the (stronger) definition of coherence given in [15].

Then, in [3] the following result has been obtained.

Theorem 1. *Given a g-coherent imprecise assessment $\mathcal{A}_n = ([\alpha_i, \beta_i],\ i \in J_n)$ on the family $\mathcal{F}_n = \{E_i|H_i, i \in J_n\}$, the extension $[\alpha_{n+1}, \beta_{n+1}]$ of $\mathcal{A}_n$ to a further conditional event $E_{n+1}|H_{n+1}$ is g-coherent if and only if the following condition is satisfied*

$$[\alpha_{n+1}, \beta_{n+1}] \cap [p_\circ, p^\circ] \neq \emptyset.$$

We briefly introduce some preliminary aspects. Let us consider a family $\mathcal{F}_{n+1} = \{E_1|H_1, \ldots, E_n|H_n, E_{n+1}|H_{n+1}\}$ and a coherent precise assessment $\mathcal{P}_n$ (in particular, a g-coherent vector $\mathcal{A}_n$ of lower bounds) on $\mathcal{F}_n$. We also enclose within square brackets the modifications relative to the case of imprecise assessments.

We consider the following assessment on $\mathcal{F}_{n+1}$

$$(\mathcal{P}_{n+1}) : P(E_i|H_i) = p_i, 1 = 1, \ldots, n, \quad P(E_{n+1}|H_{n+1} = p_{n+1},$$

$$[(\mathcal{A}_{n+1}) : P(E_i|H_i) \geq \alpha_i, 1 = 1, \ldots, n, \quad P(E_{n+1}|H_{n+1} = p_{n+1}]$$

where p_{n+1} is not fixed.

Let us denote by Π the partition of Ω obtained by expanding the expression

$$(1) \qquad (E_1 H_1 \vee E_1^c H_1 \vee H_1^c) \wedge \ldots \wedge (E_{n+1} H_{n+1} \vee E_{n+1}^c \wedge H_{n+1}^c)$$

and by $C_1, \ldots, C_m$ the atoms or constituents of Π contained in $H_0 = H_1 \vee \ldots \vee H_{n+1}$. Moreover, for each given i, we denote respectively by Γ_i and F_i

the sets of subscripts r such that $C_r \subseteq H_i$ and $C_r \subseteq E_i H_i$. We associate with pair $(\mathcal{F}_{n+1}, \mathcal{P}_{n+1}[(\mathcal{F}_{n+1}, \mathcal{A}_{n+1})]$ the following system in the unknowns $\lambda_1, \ldots, \lambda_m, p_{n+1}$

$$(2) \qquad \begin{cases} \sum_{r \in \mathcal{F}_{n+1}} \lambda_r = p_{n+1} \sum_{e \in \Gamma_{n+1}} \lambda_r \\ \sum_{r \in F_i} \lambda_r = p_i \sum_{r \in \Gamma_i} \lambda_r, & i = 1, \ldots, n; \\ \sum_r \lambda_r = 1; & \lambda_r \geq 0. \end{cases}$$

[replace the equalities in the second row by $\sum_{r \in F_i} \lambda_r \geq \alpha_i \sum_{r \in \Gamma_i}, \lambda_r \quad i = 1, \ldots, n$].

In what follows we assume that p_{n+1} has a fixed value. Then, we denote respectively by Λ and $\mathcal{S}$ the vector of unknowns and the set of solution of the system (2). Moreover, for each j we define the linear function $\phi_j(\Lambda) = \sum_{r \in \Gamma_j} \lambda_r$, and we denote by I_0 the strict subset of $J_0 = \{1, \ldots, n+1\}$ defined as

$$(3) \qquad I_0 = \{j \in J_0 : Max_{\Lambda \in \mathcal{S}} \Phi_j(\Lambda) = 0\}.$$

The following algorithm (see [2], [3]) allows to compute p' (respectively p'') [$p_\circ$ (respectively p°)], where the modifications needed to compute $p_\circ$ and p° are enclosed within square brackets. The case of precise conditional probability assessments has been also studied in [4], [5], [6], [13].

Algorithm 1. Let be given the pair $(\mathcal{F}_n, \mathcal{P}_n)[(\mathcal{F}_n, \mathcal{B}_n)$ (respectively $(\mathcal{F}_n, \mathcal{A}_n))]$ and the conditional event $E_{n+1}|H_{n+1}$.

- *Step 0.* By expanding the expression

$$\bigwedge_{i \in I} (E_i H_i \vee E_i^c H_i \vee H_i^c,$$

where $I = \{1, \ldots, n+1\}$, determine the constituents C_r contained in H_0. Then, construct the system (2).

- *Step 1.* Check the compatibility of the system (2) under the condition $p_{n+1} = 0$ (respectively $p_{n+1} = 1$). If the system (2) is not compatible goto *Step 2*, otherwise go to *Step 3*;
  - *Step 2.* Solve the linear programming problem:

$$\text{Compute} \qquad \gamma' = \max \sum_{r \in F_{n+1}} \lambda_r$$

$$\text{and/or} \quad \gamma'' = \min \sum_{r \in F_{n+1}} \lambda_r$$

$$(4) \qquad \text{subject to:} \begin{cases} \sum\limits_{r \in F_i} \lambda_r = p_i \sum\limits_{r \in \Gamma_i} \lambda_r, & i = 1, \dots, n; \\ \sum\limits_{r \in \Gamma_{n+1}} \lambda_r = 1; & \lambda_r \geq 0. \end{cases}$$

The minimum γ' (respectively the maximum γ'') of the *objective function* coincides with p' (respectively with p''), and the procedure stops; [by replacing the equalities in the first row of the constraints by $\sum_{r \in F_i} \lambda_r = \alpha_i \sum_{r \in \Gamma_i} \lambda_r, i = 1, \dots, n$, and applying the algorithm with $p_{n+1} = 1$ (respectively $p_{n+1} = 0$ when the pair $(\mathcal{F}_n, \mathcal{B}_n)$ is considered) we obtain $\gamma'' = p^\circ$ (respectively $\gamma' = p_\circ$)]

- *Step 3*. For each subscript j, compute the maximum M_j of the function Φ_j, subject to the constraints given by the system (2) with $p_{n+1} = 0$ (respectively $p_{n+1} = 1$). We have the following three cases:

  1. $M_{n+1} > 0$;
  2. $M_{n+1} = 0$, $M_j > 0$ for every $j \neq n+1$;
  3. $M_j = 0$ for $j \in I_0 = J \cup \{n+1\}$, with $J \neq \emptyset$.

  In the first two cases it is $p' = 0$ (respectively $p'' = 1$) and the procedure stops.
  In the third case replace the pair $(\mathcal{F}_n, \mathcal{P}_n$ by $(\mathcal{F}_J, \mathcal{P}_J)$ and I by I_0. then, go to *Step 0*.

The algorithm ends in a finite number of runs by computing the minimum p' (respectively the maximum p'')[$p_\circ$ (respectively p°)] of the values $p_{n+1}[\beta_{n+1}$ (respectively α_{n+1})] which are coherent (g-coherent) extensions of $p_1, \dots, p_n)[(\beta_1, \dots, \beta_n)$ (respectively $(\alpha_1, \dots, \alpha_n)$)].

3. Some procedures and examples.

In this section we examine the implementation of some procedures which allow to expand expression (1) and to construct the system (2), that is to execute the instructions at step 0 of the algorithm (1). The procedures concerning step 1 and step 2 exploit the *simplex package* of Maple V, therefore are not reported in this paper. Finally, some examples are considered.

In order to compute the constituents the logic operators *and, or* and the procedure *distrib,* which expands a boolean expression, have been used. In order to introduce in a simpler way the logical relations among the constituents, the *not* operator has been replaced by the following equivalent procedure, termed c, which allows to define easily partitions of the certain event.

Procedure c

```
c := proc() local a,i,n,m,k; a := [args];
if type(op(a), function and op(0, op(1,a)) = c then RETURN (op(1,op(1,a))) fi;
if has(a,'& or') then
n :=[seq(op(i, op(a)), i = 1 .. nops(op(a)))]; m := `&and'(seq(c(n[k]),k = 1 ..
nops(n)));
RETURN(m) fi;
if has (a,'&and') then
n := [seq(op(i, op(a)), i = 1 .. nops(op(a)))]; m := `&or'(seq(c(n[k]), k = 1 ..
nops(n)));
RETURN(m) fi; 'c'(op(a)) end.
```

Given a partition $\{E_1, \ldots, E_n\}$ of the certain event, the procedure c is exploited in the procedure *partition Om* described below to obtain, for each event E_i, its contrary

$$E_i^c = E_1 \vee \cdots \vee E_{i-1} \vee E_{i+1} \vee \cdots \vee E_n;$$

Procedure partition Om

```
partition Om :=proc() local k;
if nargs = 2 then c(args[1]) := args[2]; c(args[2]) := args[1]
else for i to nargs do if i = 1
then c(args[i] := '&or'(seq(args[k], k = 2 .. nargs))
else c(args[i]) := seq(args[k], k = 1 .. i-1) &or seq(args[k], k = i+1 .. nargs)
fi od fi end.
```

To take into account logical relations, such as implication and so on, the following procedure *int-imp* drops from expression (1) the impossible terms by labelling them with *false.* To detect them the procedure changes each term to a list m. Then the term is labelled *false* if there exists an element of m such that its contrary contains some other element of m.

Example. Let us consider a partition $\{A, B, C\}$. In this case

$$A^c = B \vee C, \quad B^c = A \vee C, \quad C^c = A \vee B,$$

so that the event ABC, for example, is impossible. In fact, the procedure changes ABC to the list $[A, B, C]$ and since $B \subset A^e$ the procedure returns *false*.

procedure int-imp

```
int-imp := proc(x)local i,m,j,v,r,k,s; contr := x; r := x;
if has(r, '&and') then m := convert(r, list); v := false;
for j to nops(m) while v = false do
if has(c(m[j]), '&or') then
for k to nops(c(m[j])) do
if member(op(k, c(m[j])), m) = true then
contr := false; RETURN(false) else v := false fi od else
if member(c(m[j]), m) then contr := false; RETURN (false)
else v := false fi fi od fi; contr end.
```

Concerning the relation of implication, if for example there are events $A_1, A_2; B_1, B_2, B_3; \ldots$ such that $A_1 \Rightarrow A_2$; $B_1 \Rightarrow B_2 \Rightarrow B_3; \ldots$, each relation is represented by an ordered list and then the set of relation is represented by the following list $[[A_1, A_2], [B_1, B_2, B_3], \ldots]$, named *logic-list*.

Then, to determine the constituents, using a general procedure (named *Constituents*) the presence of $[A_1, A_2]$ and $[B_1, B_2, B_3]$ in *logic-list* is taken into account by executing the logical conjunction between expression (1) and, respectively, the events $A_1^c \vee A_2$, $B_1^c \vee B_2, \vee B_3$, and so on.

Obviously, unconditional events are represented as conditional ones, with the conditioning event being the sure event which is denoted by the symbol OM.

The following procedures allow to construct the system (2). within the procedure *Chk-ind* (described below) it is exploited the procedure *Bln*, by means of which it is checked if the conjunction between two given events (denoted with x and y) coincides with the impossible event. The number of partitions of the sure event, given in input, is assigned as the value of a variable named *n-part*.

procedure Bln

```
Bln := proc(x,y) local conj;
c(true) := false; OM := true; conj := distrib(x and y);
if 0 < n-part then if has(conj, '&or') then
conj := distrib(map(int-imp, conj))
else conj := distrib(int-imp(conj))
fi fi; if evalb(conj = false) = false then answ 2 := true
else answ 2 := false fi; OM := evaln(OM); answ 2 end.
```

For each conditional event $E_i|H_i$ in the given family it is possible to determine $\sum_{r \in F_i} \lambda_r$ and $\sum_{r \in \Gamma_i} \lambda_r$ using the procedure $chkind$. Concerning the first sum the inputs must be the event $E_i H_i$ and the list of constituents, respectively denoted with x and y. To determine the second sum the inputs must be the event H_i and the list of constituents. For each constituent C_i, a variable λ_i is associated with the conjunction $\times \wedge C_i$ if the condition $\times \wedge C_i \neq \emptyset$ is satisfied. Finally, the expression representing the sum (named $Inds$) is defined.

procedure Chk-ind

```
Chk-ind := proc(x,y) local j,k,q,r,s,o;
s := [seq(k,k = 1 .. nops(y))]; j := 0;
for r to nops(y) do
if Bln(x,op(r,y)) = true then
j := j+1; q[j] := s[r]; o := [seq(q[k],k = 1 .. j)]
else j := j fi od;
Inds := sum('L.(op(k,o))','k' = 1 .. j); Inds end.
```

Many other procedures concerning geometrical aspects (also studied in [8] and [9]) have been implemented. In particular they allow to determine the convex hull $\mathcal{I}$ associated with a precise conditional probability assessment $\mathcal{P}$, defined on a finite family $\mathcal{F}$ of conditional events, and check the condition $\mathcal{P} \in \mathcal{I}$. Due to the lack of space, these procedures here are not described.

In the following we examine two examples, by showing the outputs produced by the algorithm (1) implemented with Maple V: in the first one we compute the minimum p' and the maximum p''; in the second one we compute the values $P_\circ$ and p°. The second example has been also studied in [11].

Example 1. Given the family $\mathcal{F} = \{B|A, BC|A\}$ and the precise probability assessment $\mathcal{P} = (1/', 1/3)$ on $\mathcal{F}$, we consider the extension of $\mathcal{P}$ to the conditional event $C|AB$. In our case the constituents, $C_0, C_1, \ldots, C_4$, are respectively

$$A^c,\ AB^c,\ ABC,\ ABC^c,\ AB^cC^c.$$

Then, we have

$$\Phi_A = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4, \quad \Phi_{AB} = \lambda_2 + \lambda_3.$$

The associated system

$$(5) \qquad \begin{cases} \lambda_2 = p(\lambda_2 + \lambda_3) \\ \lambda_2 = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\ \lambda_2 + \lambda_3 = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, \lambda_r \geq 0 \end{cases}$$

with the position $p = 0$ is infeasible.

Moreover, $Max\Phi_{AB}$ is positive and, based on the Algorithm 1, the following linear programming problem must be solved.

Compute

$$\min \lambda_2$$

Subject to the constraints:

$$(6)\qquad \begin{cases} \lambda_2 = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\ \lambda_2 + \lambda_3 = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\ \lambda_2 + \lambda_3 = 1, \lambda_r \geq 0 \end{cases}$$

The minimum p' is 1. Concerning the computation of p'', we observe that the system (5) with the position $p = 1$ is feasible and the maximum of the function Φ_{AB} is positive, then the assessment $(\mathcal{P}, 1)$ on $\{B|A, BC|A, C|AB\}$ is coherent, therefore $p'' = 1$.

Remark 2. In [11] the restrictive assumption that the probabilities of the *conditioning* events are *positive* is made. In the next example, since we don't make this assumption, a result different from that given in [11] is obtained, confirming that the de Finetti's approach is more general.

Example 2. Given the family $\mathcal{F} = \{B|A, A|B, C|B, B|C\}$ and the imprecise probability assessment $\mathcal{A} = ([1/3, 1/2], [0, 1/2], [1, 1], [1/3, 1/2])$ on $\mathcal{F}$, we consider the extension of $\mathcal{A}$ to the conditional event $C|A$. in this case the constituents, $C_0, C_1, \ldots, C_7$, are respectively

$$A^cB^cC^c,\ AB^cC,\ ABC,\ ABC^c,\ A^cBC,\ A^cBC^c,\ AB^cC^c,\ A^cB^cC.$$

Moreover it is

$$\Phi_A = \lambda_1 + \lambda_2 = \lambda_3 + \lambda_6, \quad \Phi_B = \Lambda_2\lambda_3 + \lambda_4 + \lambda_5$$
$$\Phi_C = \lambda_1 + \lambda_2 + \lambda_4 + \lambda_7.$$

The associated system

$$(7)\qquad \begin{cases} \lambda_1 + \lambda_2 = p(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6), \\ \lambda_2 + \lambda_3 \leq \frac{1}{2}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6), \\ \lambda_2 + \lambda_3 \geq \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6), \\ \lambda_2 + \lambda_3 \leq \frac{1}{2}(\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5), \\ \lambda_2 + \lambda_4 = \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5, \\ \lambda_2 + \lambda_4 \geq \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_4 + \lambda_7), \\ \lambda_2 + \lambda_4 \leq \frac{1}{2}(\lambda_1 + \lambda_2 + \lambda_4 + \lambda_7), \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 = 1, \quad \lambda_r \geq 0 \end{cases}$$

with the position $p = 0$ is feasible, so that we check the coherence of the assessment $(\mathcal{P}, 0)$ on the family $\{B|A, A|B, C|B, B|C, C|A\}$. The set $I_0 = J \cup \{5\}$ is $\{1, 5\}$. Then we consider the assessment ([1/3, 1/2]) on the subfamily $\mathcal{F}_J = \{B|A\}$ and the conditional event $C|A$. The constituents, $C_0, C_1, \dots, C_4$, generated by $B|A, C|A$, are respectively

$$A^c,\ AB^cC,\ ABC,\ ABC^c,\ AB^cC^c.$$

then we have

$$\Phi_A = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4.$$

The associated system

$$\begin{cases} \lambda_1 + \lambda_2 = p(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4), \\ \lambda_2 + \lambda_3 \geq \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4), \\ \lambda_2 + \lambda_3 \leq \frac{1}{2}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4), \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + = 1, \quad \lambda_r \geq 0 \end{cases} \tag{8}$$

with the position $p = 0$ is feasible. Obviously the maximum of the function Φ_A is positive, so $p_{*^\circ} = 0$. concerning the computation of p°, the system (7), with the position $p = 1$, is feasible and the maximum of the function Φ_A is positive, so $p^\circ = 1$.

Remark 3. We observe that, concerning the example above, in [11] the following values have been obtained: $p_\circ = \frac{1}{3}$, $p^\circ = 1$.

REFERENCES

[1] V. Biazzo, *Algoritmi probabilistici per il trattamento dell'incertezza: implementazione con Maple V.,* Degree Thesis, Dipartimento di Matematica, Università di Catania, Italy (1997).

[2] V. Biazzo - A. Gilio, *The fundamental theorem of de Finetti for conditional events: a linear algorithm to determine the interval* $[p', p'']$, Working paper (1998a).

[3] V. Biazzo - A. Gilio, *A generalization of the fundamental theorem of de Finetti for imprecise conditional probability assessments,* International Journal of Reasoning, 24 (2000), pp. 251–272.

[4] G. Coletti, *Numerical and qualitative judgements in probabilistic expert systems,* in *Probabilistic Methods in Expert Systems*, R. Scozzafava (ed.), S.I.S., Roma (1993) pp. 37–55.

[5] G. Coletti, *Coherent numerical and ordinal probabilistic assessments,* IEEE Trans on Systems, Man, and Cybernetics, 24 - 12 (1994), pp. 1747–1754.

[6] G. Coletti - R. Scozzafava, *Characterization of coherent conditional probabilities as a tool for their assessment and extension,* Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 4 - 2 (1996), pp. 103–127.

[7] A. Gilio, *Probabilistic consistency of conditional probability bounds,* in *Advances in Intelligent Computing*, Lecture Notes in Computer Science 945, Bouchon-Meunier, B. Yager, R.R. Zadeh, L. A. (eds.), Springer-Verlag, Berlin Heidelberg, (1995), pp. 200–209.

[8] A. Gilio, *Algorithms for precise and imprecise conditional probability assessments,* in *Mathematical Models for Handling Partial Knowledge in Artificial Intelligence*, Coletti G., Dubois D., Scozzafava R. (eds.), New York: Plenum Press (1995) pp. 231–254.

[9] A. Gilio - S. Ingrassia, *Geometrical aspects in checking coherence of probability assessments,* Proc. of the Sixth Int. Conf. on "Information Processing and Management of Uncertainty in Knowledge-Based Systems" (IPMU '96), Granada, July 1-5, vol. 1, (1996) pp. 55–59.

[10] F.R. Lad - J.M. Dickey - M.A. Rahman, *The fundamental theorem of prevision,* Statistica anno L, 1 (1990), pp. 19–38.

[11] H. Thone - U. Guntzer - W. Kiebling, *Toward precision of probabilistic bounds propagation,* in *Uncertainty in Artificial Intelligence*, (Eds. D. Dubois, M.P. Wellman, B. D'Ambrosio, P. Smets), Morgan Kaufmann Publishers, San Mateo, California, (1992) pp. 315–322.

[12] P. Vicig, *An algorithm for imprecise conditional probability assessments in expert systems,* Proceedings of the Sixth International Conference on *Information Processing and Management of Uncertainty in Knowledge-Based Systems*(IPMU '96), Granada, July 1-5, (1996) pp. 61–66.

[13] P. Vicig, *Upper and lower bound for coherent extensions of conditional probabilities given on finite sets,* Quaderni del Dip. di Mat. Appl. alle Sc. Econ. Stat. e Att. "B. de Finetti", Univ. di Trieste, n. 7/1997.

[14] P. Walley, *Statistical reasoning with imprecise probability,* Chapman and Hall, London.

[15] P. M. Williams, *Note on conditional previsions,* Technical Report, School of Mathematical and Physical Sciences, University of Sussex.

Dipartimento di Matematica e Informatica,
Viale A. Doria 6,
95125 Catania (ITALY)