
Imprecise Probabilities in Elvira

Andrés Cano and Manuel Gómez

Departamento de Ciencias de la Computación
Universidad de Granada

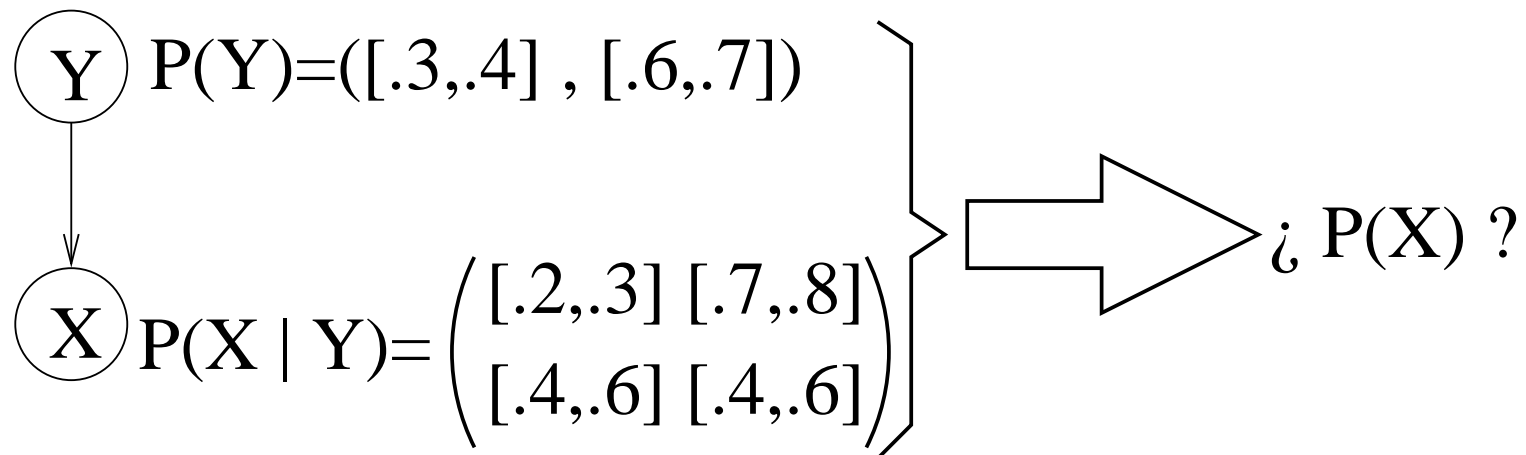
Outline

- Introduction: Main objective.
- Imprecise probabilities in Elvira format.
 - Relations with Intervals of probabilities.
 - Relations with Credal sets.
- New classes to represents imprecise probabilities.
- Variable elimination for credal networks.
- Generating intervals of probability in a Bnet.
- Future work.

Introduction

Main objective

- To use **intervals of probabilities** in the **dependence network** (credal network).
- To develop **propagation algorithms** to obtain a posteriori information from credal networks.



Imprecise probabilities in Elvira Format

Relations with intervals of probabilities

Each interval is enclosed between parenthesis

- Using **tables**: Use **table-interval**
Example

```
relation Tumor Cancer {
  values=table-interval (
    [0,presente]=(0.2,0.3)
    [0,ausente]=(0.05,0.09)
    [1,presente]=(0.7,0.8),
    [1,ausente]=(0.91,0.95)); }
```

Another Example

```
relation Tumor Cancer {
  values= table-interval (
    (0.05, 0.09) (0.2, 0.3) (0.91, 0.95) (0.7, 0.8) ); }
```

Imprecise probabilities in Elvira Format

Relations with intervals of probabilities

- Using **trees** (not now): Use **tree-interval**
Example

```
relation Tumor Cancer {
  values= tree-interval (
    case Tumor {
      0 = case Cancer {
        presente = (0.2,0.3);
        ausente = (0.05,0.09);
      }
      1 = case Cancer {
        presente = (0.7,0.8);
        ausente = (0.91,0.95);
      }
    }
  );
}
```

Imprecise probabilities in Elvira Format

Relations with credal sets

- Using **tables**: Use **convex-set** and enumerate the extreme points using any kind of **table**.

Example

```
relation Cancer {
  values=convex-set(
    table (0.2 0.8)
    table (
      [ausente]=0.3
      [presente]=0.7
    )
  );
}
```

Imprecise probabilities in Elvira Format

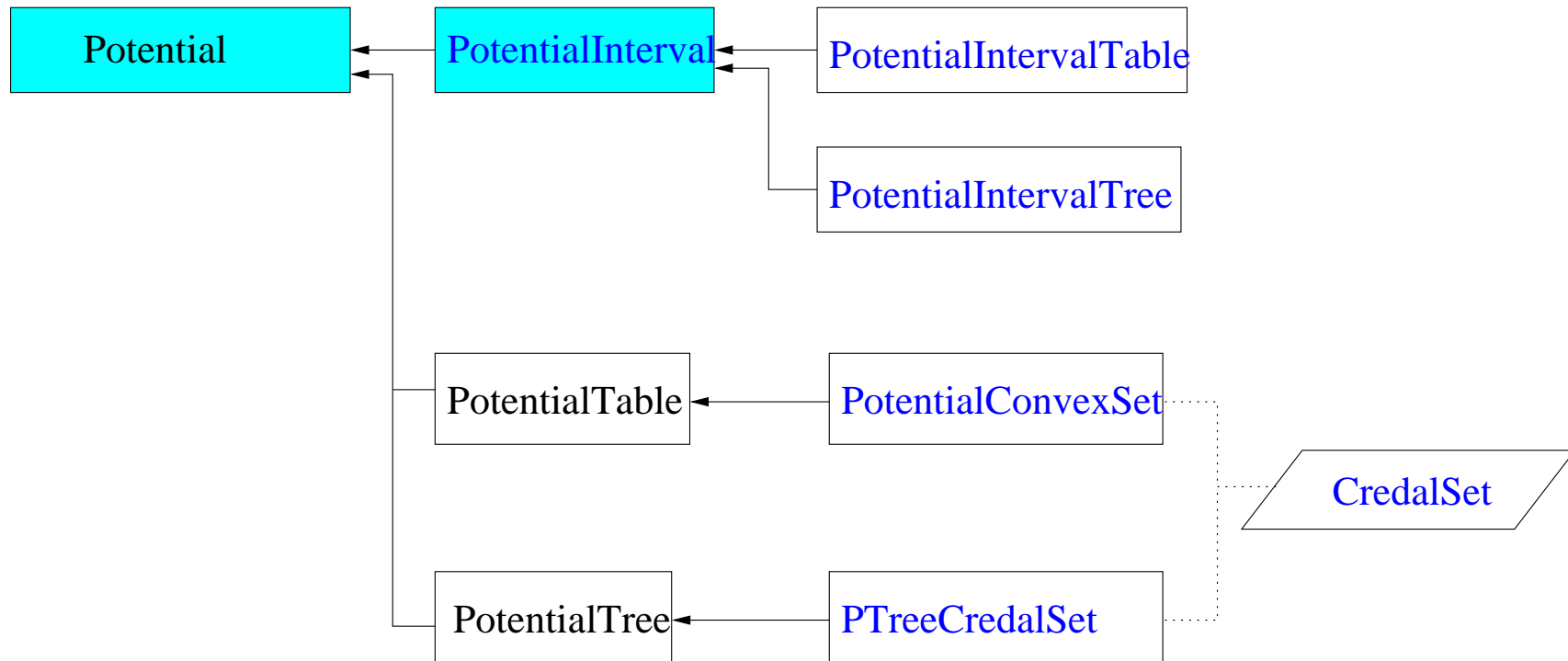
Relations with credal sets

Another Example

```
relation Tumor Cancer {  
  values= convex-set (  
    table (0.09 0.91 0.3 0.7)  
    table (0.09 0.91 0.2 0.8)  
    table (0.05 0.95 0.3 0.7)  
    table (0.05 0.95 0.2 0.8)  
  );  
}
```

New Java classes

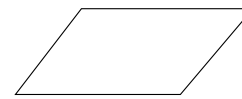
Classes to represent imprecise probabilities



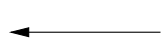
Class



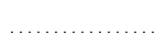
Abstract class



Interface



Extends class



Implements interface

New Java classes

Class PotentialInterval

Abstract superclass for Potentials that represents intervals of probability

Methods:

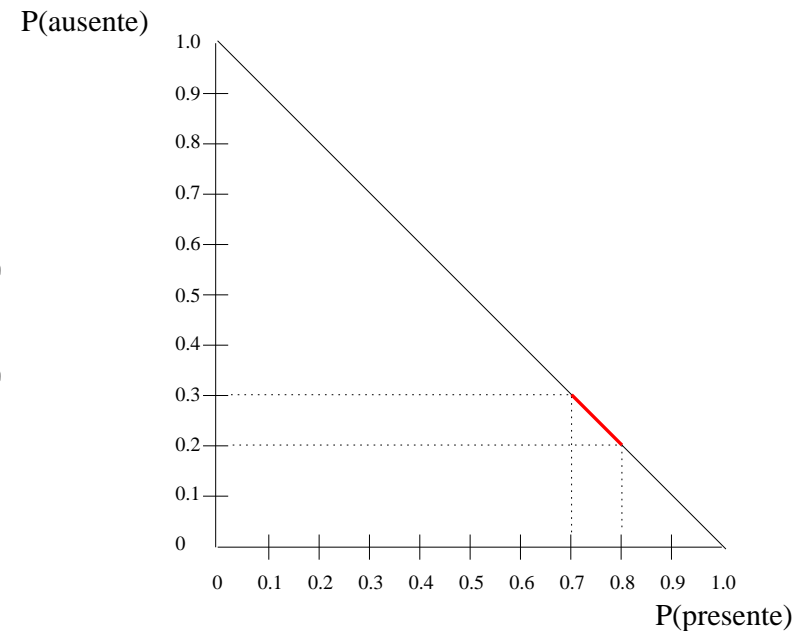
- `abstract double[] getArrayCopyMinValues()`
- `abstract double[] getArrayCopyMaxValues()`: These methods must be defined in subclasses. They are used in method `Vector getListExtrems()`.
- `Vector getListExtrems()`: Gets the extreme points from the set of intervals (De Campos, Huete and Moral, 1994).

New Java classes

Example of extreme points associated to intervals

$$P(\text{Cancer}) = \begin{cases} [0.2, 0.3] & \text{if Cancer = ausente} \\ [0.7, 0.8] & \text{if Cancer = presente} \end{cases}$$

$$\text{Extremes}(P(\text{Cancer})) = \begin{cases} (0.2, 0.8) \\ (0.3, 0.7) \end{cases}$$

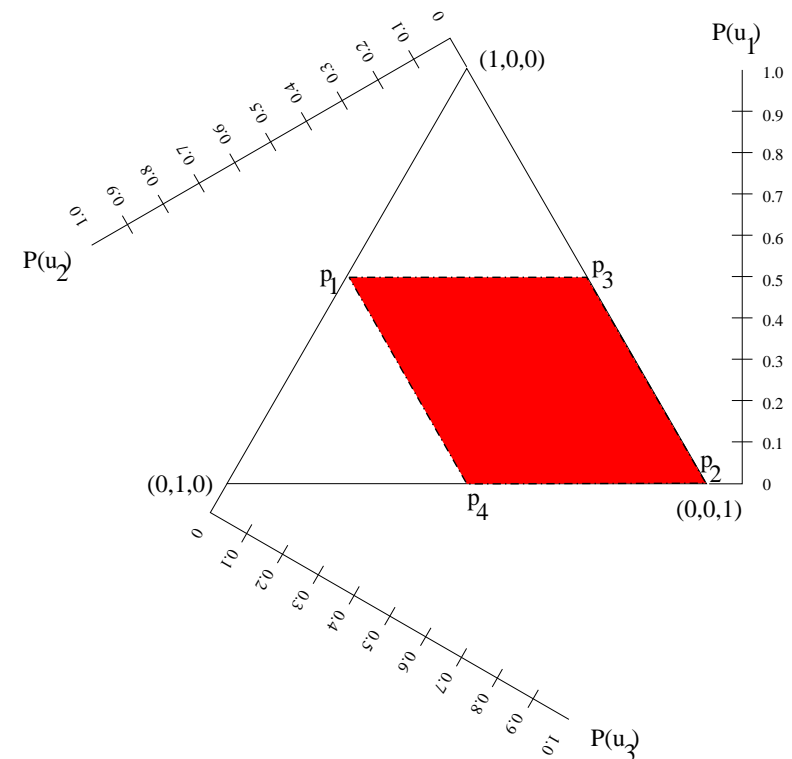


New Java classes

Another example

$$P(X) = \begin{cases} [0.0, 0.5] & \text{if } X = u_1 \\ [0.0, 0.5] & \text{if } X = u_2 \\ [0.0, 1.0] & \text{if } X = u_3 \end{cases}$$

$$\text{Extremes}(P(X)) = \begin{cases} p_1 = (0.5, 0.5, 0.0) \\ p_2 = (0.0, 0.0, 1.0) \\ p_3 = (0.5, 0.0, 0.5) \\ p_4 = (0.0, 0.5, 0.5) \end{cases}$$



New Java classes

Class `PotentialIntervalTable`

Subclass of `PontentialInterval`, it uses two arrays (tables) to represent a probability distribution defined by intervals.

Constructor:

- `PotentialIntervalTable(PotentialTable min, PotentialTable max)`: Used by the Elvira parser.
- `PotentialIntervalTable(CredalSet pot)`: Convert a `CredalSet` into a `PontentialIntervalTable`.

New Java classes

Interface CredalSet

- It specifies the methods that must be defined in Potentials for credal sets.
- It allows to implement **general propagation methods** for any kind of CredalSet.
- CredalSet potentials use one (or more) **transparent variable** with as many states as the number of extreme points.
- Transparent variables are available in the CredalSet potentials but not in the relations.

New Java classes

Methods in interface CredalSet

- `Vector getListNonTransparents()`
- `Vector getListTransparents()`
- `double getValue(Configuration conf)`
- `double getMinimum(Configuration subconf)`
- `double getMaximum(Configuration subconf)`

New Java classes

Class PotentialConvexSet

- Subclass of PotentialTable, implements CredalSet, it uses a table to represent a credal set.
- It contains one (or more) *transparent* variable with as many states as the number of extreme points.

Constructors:

- `PotentialConvexSet(CredalSet pot)`: Used to transform any CredalSet into a PotentialConvexSet.
- `PotentialConvexSet(NodeList vars, Vector C)`: It is used by the Elvira parser to build a PotentialConvexSet from a Vector of extreme points.

New Java classes

Example of PotentialConvexSet

$$\text{Extremes}(P(X)) = \begin{cases} p_1=(0.5,0.5,0.0) \\ p_2=(0.0,0.0,1.0) \\ p_3=(0.5,0.0,0.5) \\ p_4=(0.0,0.5,0.5) \end{cases}$$

T / X	u_1	u_2	u_3
1	0.5	0.5	0.0
2	0.0	0.0	1.0
3	0.5	0.0	0.5
4	0.0	0.5	0.5

New Java classes

Class PTreeCredalSet

- Subclass of PotentialTree, implements CredalSet, it uses a tree to represent a credal set.
- It contains one (or more) transparent variable with as many states as the number of extreme points.

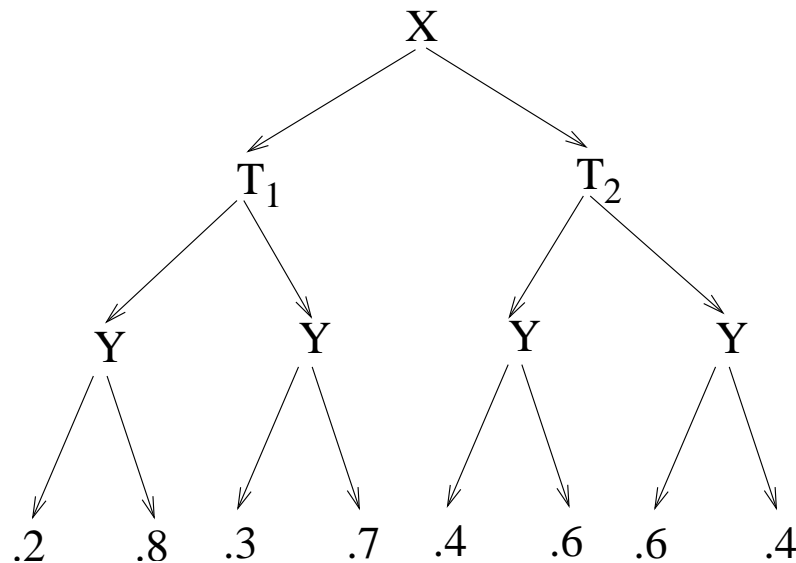
Constructor:

- `PTreeCredalSet(PotentialInterval intervalSet):`
Used to transform any PotentialInterval into a PTreeCredalSet.

New Java classes

Example of PTreeCredalSet

$$P(Y|X) = \begin{cases} & \begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & [0.2,0.3] & [0.4,0.6] \\ y_2 & [0.7,0.8] & [0.4,0.6] \end{array} \end{cases}$$



	(x_1, y_1)	(x_1, y_2)	(x_2, y_1)	(x_2, y_2)
p_1	0.2	0.8	0.4	0.6
p_2	0.2	0.8	0.6	0.4
p_3	0.3	0.7	0.4	0.6
p_4	0.3	0.7	0.6	0.4

Variable elimination for credal networks

Class `VEWithPTreeCredalSet`

Propagation method for **probability intervals** and **credal sets**.

- This class extends `VariableElimination`.
 - Variable elimination is based on tree operations over Potentials: Combination, marginalization and restriction (projection).
- Intervals are converted into credal sets (`PTreeCredalSet`) before propagating.
- Final results are converted again into probability intervals.

How to use it?

```
java elvira.inference.elimination.VEWithPTreeCredalSet  
bnet.elv results.out [evidence.evi]
```

Variable elimination for credal networks

Class `VEWithPTreeCredalSet` **overrides** the following methods:

- `Relation transformInitialRelation(Relation r)`: It converts the probability distributions given by intervals into `PTreeCredalSets`.
- `void normalizeResults()`: The result of propagation is a `PTreeCredalSet` for each variable. This method normalizes each one of the extreme points and it converts the credal set into a `PotentialIntervalTable`.

Generating intervals of probability in a Bnet

Imprecise Dirichlet model

(Walley, 1996)

- It is normally used to learn intervals from a database.
 - Sample size: N
 - The s parameter: Walley interprets it as the number of *hidden observations*. Normally taken as 1 or 2.
 - n_j : Number of observations of x_j
 - $n = \{n_1, n_2, \dots, n_k\}$
- Intervals for A_j :

$$\bar{P}(A_j|n) = \frac{n_j + s}{N + s}$$

$$\underline{P}(A_j|n) = \frac{n_j}{N + s}$$

Generating intervals of probability in a Bnet

Getting credal networks from Bnets

- We can apply the *Imprecise Dirichlet model* to obtain a credal network from a Bayesian network using

$$\bar{P}(A_j|n) = \frac{P(A_j) + s}{N + s}$$

$$\underline{P}(A_j|n) = \frac{P(A_j)}{N + s}$$

- Class `Bnet` contains a constructor to do that:

```
Bnet(String nameOfBnet, int N, int s)
```

Future Work

- Implements the Imprecise Dirichlet model to learn a credal network (parameters) from a database.
- Apply optimization combinatorial algorithms to propagate intervals.

References

- L.M. de Campos, J.F. Huete and S. Moral. *Probability Intervals: a Tool for Uncertain Reasoning*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2:167-196, 1994.
- P. Walley. *Inferences from multinomial data: learning about a bag of marbles (with discussion)*. Journal of the Royal Statistical Society, Series B, 58:3-57, 1996.