How to implement the belief functions?

Arnaud Martin
Arnaud.Martin@univ-rennes1.fr
Université de Rennes 1 - IRISA, Lannion, France
Autrans, April, 5 2011
Plan

- Natural order
- Smets codes
- General framework
- How to obtain bbas?
  - Random bbas
  - Distance based model
  - Probabilistic based model
Plan

- Natural order
- Smets codes
- General framework
- How to obtain bbas?
  - Random bbas
  - Distance based model
  - probabilistic based model
Discernment frame: $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$

Power set: all the disjunctions of $\Theta$:

$2^\Theta = \{\emptyset, \{\theta_1\}, \{\theta_2\}, \{\theta_1 \cup \theta_2\}, \ldots, \Theta\}$

Natural order:

$$2^\Theta = \{\emptyset, \{\theta_1\}, \{\theta_2\}, \{\theta_1 \cup \theta_2\},$$

$$\{\theta_3\}, \{\theta_1 \cup \theta_3\}, \{\theta_2 \cup \theta_3\}, \{\theta_1 \cup \theta_2 \cup \theta_3\},$$

$$\{\theta_4\}, \ldots, \Theta\}$$
### Natural order:

<table>
<thead>
<tr>
<th></th>
<th>$\emptyset$</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_1 \cup \theta_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>$3 = 2^2 - 1$</td>
<td></td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>$\theta_1 \cup \theta_3$</td>
<td>$\theta_2 \cup \theta_3$</td>
<td>$\theta_1 \cup \theta_2 \cup \theta_3$</td>
<td></td>
</tr>
<tr>
<td>$4 = 2^{3-1} + 1$</td>
<td>5</td>
<td>6</td>
<td>$7 = 2^3 - 1$</td>
<td></td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$8 = 2^{4-1} + 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\theta_i$</td>
<td>...</td>
<td>...</td>
<td>$\Theta$</td>
<td></td>
</tr>
<tr>
<td>$2^{i-1} + 1$</td>
<td></td>
<td></td>
<td>$2^n$</td>
<td></td>
</tr>
</tbody>
</table>
Bba in Matlab:
Example: $m_1(\theta_1) = 0.5$, $m_1(\theta_3) = 0.4$, $m_1(\theta_1 \cup \theta_2 \cup \theta_3) = 0.1$
$m_2(\theta_3) = 0.4$, $m_2(\theta_1 \cup \theta_3) = 0.4$

F1=[1 4 7]’;
F2=[4 5]’;
M1=[0.5 0.4 0.1]’;
M2=[0.4 0.6]’;
Combination

\[ m_{\text{Conj}}(X) = \sum_{Y_1 \cap Y_2 = X} m_1(Y_1)m_2(Y_2) \] (1)

\[ \theta_1 \cap (\theta_1 \cup \theta_3): 1 \cap 5 \]

In binary on base 3: 1 = 100 and 5 = 101 = 100 | 001

100 \& 101 = 100
In Matlab:
sizeDS=3;
F1=[1 4 7]’;
F2=[4 5]’;
M1=[0.5 0.4 0.1]’;
M2=[0.4 0.6]’;
Fres=[];
Mres=[];
for i=1:size(F1)
    for j=1:size(F2)
        Fres=[Fres bi2de(de2bi(F1(i),sizeDS)&de2bi(F2(j),sizeDS))];
        Mres=[Mres M1(i)*M2(j)];
    end
end
Natural order
Smets codes
General framework
How to obtain bbas?
  Random bbas
  Distance based model
  probabilistic based model
Plan

- Natural order
- Smets codes
- General framework
- How to obtain bbas?
  - Random bbas
  - Distance based model
  - Probabilistic based model
Smets codes for DST

Smets gives the codes of the Mobius transform (see Only_Mobius_Transf) for conversions:

- bba and belief: mtobel, beltom
- bba and plausibility: mtopl, pltom
- bba and communality: mtoq, qtom
- bba and implicability: mtob, btom
- bba to pignistic probability: mtobetp
- etc...

e.g. in Matlab:

```
m1=[0 0.4 0.1 0.2 0 0 0.1]’;
mtobel(m1)
gives: 0 0.4000 0.1000 0.7000 0.2000 0.6000 0.3000 1.0000
```
Conjunctive combination

\[ m_{\text{Conj}}(X) = \sum_{Y_1 \cap \ldots \cap Y_m = X} \prod_{j=1}^{m} m_j(Y_j) \]

The practical way:

\[ q(X) = \prod_{j=1}^{m} q_j(X) \]

Disjunctive combination

\[ m_{\text{Dis}}(X) = \sum_{Y_1 \cup \ldots \cup Y_m = X} \prod_{j=1}^{m} m_j(Y_j) \]

The practical way:

\[ b(X) = \prod_{j=1}^{m} b_j(X) \]
In Matlab

For the conjunctive rule of combination:

\[ m1 = [0 \ 0.4 \ 0.1 \ 0.2 \ 0.2 \ 0 \ 0 \ 0.1]' ; \]
\[ m2 = [0 \ 0.2 \ 0.3 \ 0.1 \ 0.1 \ 0 \ 0.2 \ 0.1]' ; \]
\[ q1 = mtoq(m1) ; \]
\[ q2 = mtoq(m2) ; \]
\[ qConj = q1 .* q2 ; \]
\[ mConj = qtom(qConj) \]
\[ mConj = \begin{bmatrix} 0.4100 & 0.2200 & 0.2000 & 0.0500 & 0.0900 & 0 & 0.0200 & 0.0100 \end{bmatrix} \]
In Matlab
For the disjunctive rule of combination:
\[
m1 = [0 \ 0.4 \ 0.1 \ 0.2 \ 0.2 \ 0 \ 0 \ 0.1]';
m2 = [0 \ 0.2 \ 0.3 \ 0.1 \ 0.1 \ 0 \ 0.2 \ 0.1]';
b1 = mtob(m1);
b2 = mtob(m2);
bDis = b1.*b2;
mDis = btom(bDis)
\]
mDis =
\[
0 \ 0.0800 \ 0.0300 \ 0.3100 \ 0.0200 \ 0.0800 \ 0.1300 \ 0.3500
\]
Once bbas are combined, to decide just use the functions mtobel, mtopl or mtobetp, etc.

**In Matlab**

```matlab
mtopl(mConj)
0 0.2800 0.2800 0.5000 0.1200 0.3900 0.3700 0.5900
mtobetp(mConj)
0.4209 0.4040 0.1751

mtopl(mDis)
0 0.8200 0.8200 0.9800 0.5800 0.9700 0.9200 1.0000
mtobetp(mDis)
0.3917 0.3667 0.2417
```
DST code for the combination:

- criteria=1 Smets criteria
- criteria=2 Dempster-Shafer criteria (normalized)
- criteria=3 Yager criteria
- criteria=4 disjunctive combination criteria
- criteria=5 Dubois criteria (normalized and disjunctive combination)
- criteria=6 Dubois and Prade criteria (mixt combination)
- criteria=7 Florea criteria
- criteria=8 PCR6
- criteria=9 Cautious Denoeux Min for non-dogmatics functions
- criteria=10 Cautious Denoeux Max for separable functions
- criteria=11 Hard Denoeux for functions subnormals
- criteria=12 Mean of the bbas
Yager rule:

\[ m_Y(X) = m_{\text{Conj}}(X) , \forall X \in 2^\Theta \setminus \{ \emptyset, \Theta \} \]
\[ m_Y(\Theta) = m_{\text{Conj}}(\Theta) + m_{\text{Conj}}(\emptyset) \]
\[ m_Y(\emptyset) = 0. \]

Florea rule \( \forall X \in 2^\Theta, X \neq \emptyset : \)

\[ m_{\text{Flo}}(X) = \beta_1(k)m_{\text{Dis}}(X) + \beta_2(k)m_{\text{Conj}}(X), \]

with:

\[ \beta_1(k) = \frac{k}{1 - k + k^2}, \]
\[ \beta_2(k) = \frac{1 - k}{1 - k + k^2}. \]
Dubois and Prade rule:

\[ m_{DP}(X) = \sum_{A \cap B = X} m_1(A)m_2(B) \]
\[ + \sum_{A \cup B = X} m_1(A)m_2(B). \]
\[ A \cap B = \emptyset \]

PCR6 transfers the partial conflict on focal elements given this conflict proportionally to the masses.

\[ m_{PCR6}(X) = m_{Conj}(X) + \sum_{Y \in 2^\Theta, X \cap Y = \emptyset} \left( \frac{m_1(X)^2m_2(Y)}{m_1(X) + m_2(Y)} + \frac{m_2(X)^2m_1(Y)}{m_2(X) + m_1(Y)} \right) \]
DST code

decision DST code for the decision:

- criteria=1 maximum of the plausibility
- criteria=2 maximum of the credibility
- criteria=3 maximum of the credibility with rejection

\[
\begin{aligned}
\text{bel}(\theta_d) &= \max_{1 \leq i \leq n} \text{bel}(\theta_i) \\
\text{bel}(\theta_d) &\geq \text{bel}(\theta_c^d)
\end{aligned}
\]

- criteria=4 maximum of the pignistic probability
- criteria=5 Appriou criteria (decision onto \(2^\Theta\)):

\[
A = \arg\max_{X \in 2^\Theta} (m_d(X)f_d(X)),
\]

where

\[
m_d(X) = \left( \frac{K_d \lambda_X}{|X|^r} \right), \quad r \in [0, 1]
\]
test.m:

m1=[0 0.4 0.1 0.2 0.2 0 0 0.1]’;
m2=[0 0.2 0.3 0.1 0.1 0 0.2 0.1]’;
m3=[0.1 0.2 0 0.4 0.1 0.1 0 0.1]’;

m3d=discounting(m3,0.95);

M_comb_Smets=DST([m1 m2 m3d],1);
M_comb_PCR6=DST([m1 m2],8);

class_fusion=decisionDST(M_comb_Smets’,1)
class_fusion=decisionDST(M_comb_PCR6’,1)
class_fusion=decisionDST(M_comb_Smets’,5,0.5)
Plan

- Natural order
- Smets codes
- General framework
- How to obtain bbas?
  - Random bbas
  - Distance based model
  - Probabilistic based model
Plan

- Natural order
- Smets codes
- General framework
- How to obtain bbas?
  - Random bbas
  - Distance based model
  - Probabilistic based model
Main problem of the DST code: all elements must be coded (not only the focal elements)

Only usable for belief functions defined on power set ($2^\Theta$)

General belief functions framework works for power set and hyper power set ($D^\Theta$)
DSmT introduced by Dezert, 2002.

- $D^\Theta$ closed set by union and intersection operators
- $D^\Theta$ is not closed by complementary, $A \in D^\Theta \nRightarrow \overline{A} \in D^\Theta$
- if $|\Theta| = n$: $2^n = |2^\Theta| << |D^\Theta| << |2^{2^\Theta}| = 2^{2^n}$
- $D^\Theta_r$: reduced set considering some constraints ($\theta_2 \cap \theta_3 \equiv \emptyset$)

$$\text{GPT}(X) = \sum_{Y \in D^\Theta_r, Y \neq \emptyset} \frac{C_M(X \cap Y)}{C_M(Y)}m(Y)$$

where $C_M(X)$ is the cardinality of $X$ in $D^\Theta_r$
**A simple codification** (Martin, 2009)

Affect an integer of \([1; 2^n - 1]\) to each distinct part of Venn diagram \((n = |\Theta|)\)

\[\Theta = \{[1 2 3 5], [1 2 4 6], [1 3 4 7]\}\]
Adding a constraint: if \( \Theta = \{[1 \ 2 \ 3 \ 5], [1 \ 2 \ 4 \ 6], [1 \ 3 \ 4 \ 7]\} \) and we know \( \theta_2 \cap \theta_3 \equiv \emptyset \) (i.e. \( \theta_2 \cap \theta_3 \notin D_r^\Theta \))

The parts 1 and 4 of Venn diagram do not exist:
\( \Theta_r = \{[2 \ 3 \ 5], [2 \ 6], [3 \ 7]\} \)

Operations on focal elements

\[
\begin{align*}
\theta_1 \cap \theta_3 &= [3] \\
\theta_1 \cup \theta_3 &= [2 \ 3 \ 5 \ 7] \\
(\theta_1 \cap \theta_3) \cup \theta_2 &= [2 \ 3 \ 6]
\end{align*}
\]

The cardinality \( C_M(X) \): the number of integers in the codification of \( X \)

Gives an easy Matlab programation of the combination rules and the decision functions
Decoding: to present the decision or a result to the human - *The codification is not understandable*

If the decision set is given, we just have to sweep the corresponding part of $D_r^\Theta$

Without any knowledge of the element to decode:

1. We can use the Smarandache condification more lisible but less practical in Matlab

2. We sweep all $D_r^\Theta$ (first considering $2^\Theta$). There is a combinatorial risk.
Description of the problem
CardTheta=3; % cardinality of Theta
% list of experts with focal elements and associated bba
  expert(1).focal={'1' '1u3' '3' '1u2u3'};
  expert(1).bba=[0.5 0.3 0.1 0.1];

  expert(2).focal={'1' '2' '1u3' '1u2u3'};
  expert(2).bba=[0.5 0.6 0.1 0.1];

  expert(3).focal={'1' '3n2' '(1n2)u3'};
  expert(3).bba=[0.2 0.7 0.1];

constraint={''}; % set of empty elements e.g. '1n2'
In test.m

**Description of the problem**

elemDec=’A’; % set of decision elements:

- list of elements on which we can decide,
- A for all,
- S for singletons only,
- F for focal elements only,
- SF for singleton plus focal elements,
- Cm for given specificity, e.g. elemDec=’Cm’ ’1’ ’4’; minimum of cardinality 1, maximum=4,
- 2T for only $2^\Theta$ (DST case)
Parameters

Combination criterium

criteriumComb = is the combination criterium

- criteriumComb=1 Smets criterium
- criteriumComb=2 Dempster-Shafer criterium (normalized)
- criteriumComb=3 Yager criterium
- criteriumComb=4 disjunctive combination criterium
- criteriumComb=5 Florea criterium
- criteriumComb=6 PCR6
- criteriumComb=7 Mean of the bbas
Parameters

Combination criterium

criteriumComb = is the combination criterium

- criteriumComb=8 Dubois criterium (normalized and disjunctive combination)
- criteriumComb=9 Dubois and Prade criterium (mixt combination)
- criteriumComb=10 Mixt Combination (Martin and Osswald criterium)
- criteriumComb=11 DPCR (Martin and Osswald criterium)
- criteriumComb=12 MDPCR (Martin and Osswald criterium)
- criteriumComb=13 Zhang’s rule
Parameters

Decision criterion

criteriumDec = is the combination criterion

- criteriumDec=0 maximum of the bba
- criteriumDec=1 maximum of the pignistic probability
- criteriumDec=2 maximum of the credibility
- criteriumDec=3 maximum of the credibility with reject
- criteriumDec=4 maximum of the plausibility
- criteriumDec=5 Appriou criterium
- criteriumDec=6 DSmp criterium
Parameters

Mode of fusion
mode=’static’; % or ’dynamic’

Display

display = kind of display

▶ display = 0 for no display,
▶ display = 1 for combination display,
▶ display = 2 for decision display,
▶ display = 3 for both displays
Fusion
fuse(expert,constraint,CardTheta,criteriumComb,criteriumDec,mode,
elemDec,display)
Called functions:

▶ Coding: call coding, addConstraint, codingExpert
▶ Combination: call combination
▶ Decision: call decision
▶ Display: call decodingExpert, decodingFocal
Plan

- Natural order
- Smets codes
- General framework
- How to obtain bbas?
  - Random bbas
  - Distance based model
  - Probabilistic based model
Plan

- Natural order
- Smets codes
- General framework
- How to obtain bbas?
  - Random bbas
  - Distance based model
  - Probabilistic based model
In Matlab:

1. ThetaSize = 3;
2. nbFocalElement = 4;
3. ind = randperm(2^ThetaSize);
4. indFocalElement = ind(1:nbFocalElement);
5. randMass = diff([0; sort(rand(nbFocalElement-1,1)); 1]);
   
   *We take the difference between 3 ordered random number in [0,1], e.g. diff([0; [0.3; 0.9]; 1]) gives 0.3 0.6 0.1*

6. MasseOut(indFocalElement,i) = randMass;
Random bbas

- focal elements can be everywhere:
  
  \[
  \text{ind} = \text{randperm}(2^{\Theta \text{Size}});
  \]

- focal elements not on the emptyset:
  
  \[
  \text{ind} = 1 + \text{randperm}(2^{\Theta \text{Size}} - 1);
  \]

- no dogmatic mass: one focal element is on Theta (ignorance):
  
  \[
  \text{ind} = [2^{\Theta \text{Size}} \text{ randperm}(2^{\Theta \text{Size}} - 1)];
  \]

- no dogmatic mass: one focal element is on Theta (ignorance) and focal elements are not on the emptyset
  
  \[
  \text{ind} = [2^{\Theta \text{Size}} (1 + \text{randperm}(2^{\Theta \text{Size}} - 2))] \\
  \text{(nbFocalElement} = \Theta \text{Size}));
  \]

- all the focal elements are the singletons:
  
  \[
  \text{ind} = [ ];
  \]

  \[
  \text{for } i = 1: \Theta \text{Size} \\
  \quad \text{ind} = [\text{ind}; 1 + 2^{(i-1)}];
  \]

end
Distance based model (Denœux 1995)

Only $\theta_i$ and $\Theta$ are focal elements, $n \times m$ sources (experts)

- Prototypes case ($x_i$ center of $\theta_i$). For the observation $x$

\[
m^i_j(\theta_i) = \alpha_{ij} \exp[-\gamma_{ij}d^2(x, x_i)]
\]
\[
m^i_j(\Theta) = 1 - \alpha_{ij} \exp[-\gamma_{ij}d^2(x, x_i)]
\]

- $0 \leq \alpha_{ij} \leq 1$: discounting coefficient and $\gamma_{ij} > 0$, are parameters to play on the quantity of ignorance and on the form of the mass functions

- The distance allows to give a mass to $x$ higher according to the proximity to $\theta_i$

- belief $k$-nn: we consider the $k$-nearest neighbors instead of $x_i$

- Then we combine the bbas
In Matlab (Denœux codes)

See ExampleIris.m

load iris
ind = randperm(150);
xapp = x(ind(1:100),:);
Sapp = S(ind(1:100));
xtst = x(ind(101:150),:);
Stst = S(ind(101:150));
[gamm, alpha] = knndsinit(xapp, Sapp); % initialization
[gamm, alpha, err] = knndsfit(xapp, Sapp, 5, gamm, 0); % parameter optimization
[m, L] = knndsval(xapp, Sapp, 5, gamm, alpha, 0, xtst); % test
[value, Sfind] = max(m);
[mat_conf, vect_prob_classif, vect_prob_error] = build_conf_matrix(Sfind, Stst)
Probabilistic based model (1/6)

- Need to estimate $p(S_j | \theta_i)$
- 2 models proposed by Appriou according to both axioms:
  1. the $n \times m$ couples $[M_{ij}, \alpha_{ij}]$ are distinct information sources where focal elements are: $\theta_i$, $\theta^c_i$ and $\Theta$
  2. If $M_{ij} = 0$ and the information is valid ($\alpha_{ij} = 1$) then it is certain that $\theta_i$ is not true.
Probabilistic based model (2/6)

Model 1: \[ m^i_j(\theta_i) = M^j_i \]
\[ m^i_j(\theta^c_i) = 1 - M^j_i \]

Model 2: \[ m^i_j(\Theta) = M^j_i \]
\[ m^i_j(\theta^c_i) = 1 - M^j_i \]

Adding the reliability \( \alpha_{ij} \) with the discounting:

Model 1:
\[ m^i_j(\theta_i) = \alpha_{ij} M^j_i \]
\[ m^i_j(\theta^c_i) = \alpha_{ij} (1 - M^j_i) \]
\[ m^i_j(\Theta) = 1 - \alpha_{ij} \]

Model 2:
\[ m^i_j(\theta_i) = 0 \]
\[ m^i_j(\theta^c_i) = \alpha_{ij} (1 - M^j_i) \]
\[ m^i_j(\Theta) = 1 - \alpha_{ij} (1 - M^j_i) \]
How to find $M^j_i$?

3th axiom:

3 Conformity to the Bayesian approach (case where $p(S_j|\theta_j)$ is exactly the reality ($\alpha_{ij} = 1$) for all $i, j$) and all the a priori probabilities $p(\theta_i)$ are known)
Probabilistic based model (4/6)

Model 1: \( M_j^i = \frac{R_j p(S_j | \theta_j)}{1 + R_j p(S_j | \theta_j)} \)

\begin{align*}
    m_j^i(\theta_i) &= \frac{\alpha_{ij} R_j p(S_j | \theta_j)}{1 + R_j p(S_j | \theta_j)} \\
    m_j^i(\theta^c_i) &= \frac{\alpha_{ij}}{1 + R_j p(S_j | \theta_j)} \\
    m_j^i(\Theta) &= 1 - \alpha_{ij}
\end{align*}

with \( R_j \geq 0 \) a normalization factor.
Model 2: \( M^j_i = R_j p(S_j|\theta_j) \)

\[
\begin{align*}
    m^i_j(\theta_i) &= 0 \\
    m^i_j(\theta^c_i) &= \alpha_{ij}(1 - R_j p(S_j|\theta_j)) \\
    m^i_j(\Theta) &= 1 - \alpha_{ij}(1 - R_j p(S_j|\theta_j))
\end{align*}
\]

with \( R_j \in [0, (\max_{S_{ij}} p(S_j|\theta_j))^{-1}] \)

In practical:

- \( \alpha_{ij} \): discounting coefficient fixed near 1 and \( p(S_j|\theta_j) \) can be given by the confusion matrix
- Adapted to the cases where we learn one class against all the others
In Malab:

- take the previous confusion matrix or mat_conf=[68 12 22 ; 9 42 5 ; 8 2 87]
- mat_masse= bbaType(mat_conf,alpha,model): gives all the possible bbas (i.e. number of classes) for the given confusion matrix, alpha (a constant such as 0.95) and the model (1 or 2)
- bbas=buildBbas(Stst,mat_conf,alpha,model): gives the bbas resulting of the founded classes given in Stst
Probabilistic vs Distance

Difficulties:

▶ Appriou: learning the probabilities $p(S_j|\theta_j)$
▶ Denœux: choice of the distance $d(x, x_i)$

Easiness:

▶ $p(S_j|\theta_j)$ easier to estimate on decisions with the confusion matrix of the classifiers
▶ $d(x, x_i)$ easier to choose on the numeric outputs of classifiers (ex.: Euclidian distance)
Toolboxes:

a lot of papers on:
On the presented codes:


Other way to code belief functions: