Chapter 3 : Multi-valued Data Fusion

John Klein

Université de Lille1 - CRIStAL UMR CNRS 9189

Chapter organization

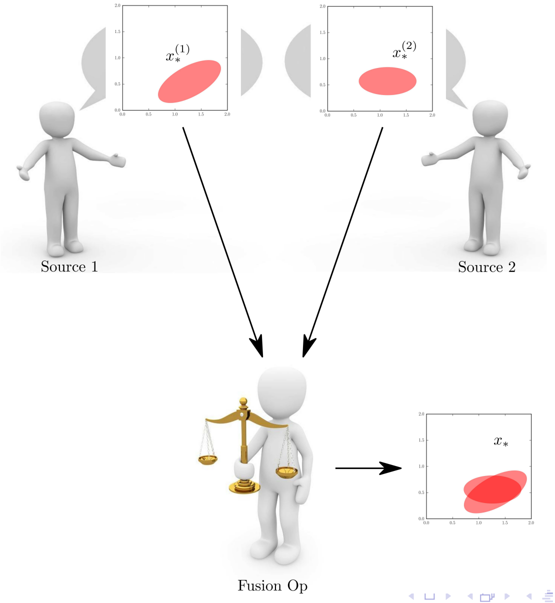
- In this chapter, information sources are allowed to deliver more imprecise advovacies, *i.e.* multi-valued advovacies.

- In this chapter, information sources are allowed to deliver more imprecise advovacies, *i.e.* multi-valued advovacies.

- This setting is more general than point data fusion.

- In this chapter, information sources are allowed to deliver more imprecise advovacies, *i.e.* multi-valued advovacies.
- This setting is more general than point data fusion.
- But the advocacy space $\mathbb{X}$ is more structured.

Multi-valued data fusion setting :



Let us give a definition for **multi-valued** **fusion problems** :

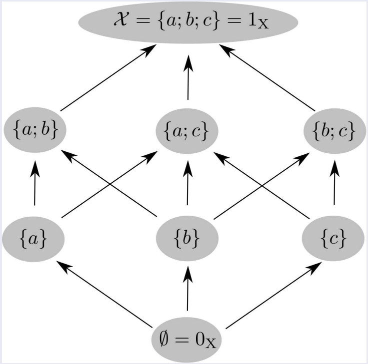
Definition

Partially ordered data fusion is a subclass of data fusion where advovacies live in $\mathbb{X} = 2^{\mathcal{X}}$.

- In this case, $\mathbb{X}$ has a distributive complemented lattice structure.

Example

$\mathbb{X} = 2^{\mathcal{X}}$ is the power set of a given set $\mathcal{X} = \{a; b; c\}$, then $\{2^{\mathcal{X}}, \subseteq, \cap, \cup, .^c\}$ is a complemented lattice. Here is the corresponding Hasse diagram :



The arrow stands for the inclusion partial order.

Chapter organization

Set-valued data fusion : operators

- In this context, each information source delivers a datum like $x_*^{(i)} = A \quad \Leftrightarrow \quad x \in A$.
- Two natural fusion operators arising from the binary operations endowing the lattice.
- Unless explicit comment, we will always consider that the lattice we are interested in is $(2^{\mathcal{X}}, \subseteq, \cap, \cup)$.

Set-valued data fusion : operators

- In this context, each information source delivers a datum like $x_*^{(i)} = A \quad \Leftrightarrow \quad x \in A$.
- Two natural fusion operators arising from the binary operations endowing the lattice.
- Unless explicit comment, we will always consider that the lattice we are interested in is $\left(2^{\mathcal{X}}, \subseteq, \cap, \cup\right)$.

Set-valued data fusion : operators

- In this context, each information source delivers a datum like $x_*^{(i)} = A \quad \Leftrightarrow \quad x \in A$.
- Two natural fusion operators arising from the binary operations endowing the lattice.
- Unless explicit comment, we will always consider that the lattice we are interested in is $\left(2^{\mathcal{X}}, \subseteq, \cap, \cup\right)$.

Set-valued data fusion : operators

Definition

Let us consider a data fusion problem as defined in 1. The (canonical) **conjunctive operator** $\hat{\hat{x}}_\cap$ is defined as follows :

$$\hat{\hat{x}}_\cap : \left(x_*^{(1)}, .., x_*^{(N_s)}\right) \to \bigcap_{i=1}^{N_s} x_*^{(i)}. \quad (1)$$

Definition

Let us consider a data fusion problem as defined in 1. The (canonical) **disjunctive operator** $\hat{\hat{x}}_\cup$ is defined as follows :

$$\hat{\hat{x}}_\cup : \left(x_*^{(1)}, .., x_*^{(N_s)}\right) \to \bigcup_{i=1}^{N_s} x_*^{(i)}. \quad (2)$$

Set-valued data fusion : operator nature

- Roughly speaking, conjunctive operators are efficient because they converge to the smallest possible set of solutions but they are risky if at least one source is not fully truthful. In this latter case, $\hat{\hat{x}}_{\cap}$ may give $\emptyset$ as output, which is a typical case of conflict between sources.
- On the opposite, disjunctive operators are regarded as cautious operators because they do not discard any possible value supported by any source. However, disjunctive operators are poorly efficient. For instance, $\hat{\hat{x}}_{\cup}$ may deliver $\mathcal{X}$ as output making it totally impossible to decide on the value of x.
- An operator can be neither conjunctive nor disjunctive.

Set-valued data fusion : operator nature

- Roughly speaking, conjunctive operators are efficient because they converge to the smallest possible set of solutions but they are risky if at least one source is not fully truthful. In this latter case, $\hat{\hat{x}}_{\cap}$ may give $\emptyset$ as output, which is a typical case of conflict between sources.

- On the opposite, disjunctive operators are regarded as cautious operators because they do not discard any possible value supported by any source. However, disjunctive operators are poorly efficient. For instance, $\hat{\hat{x}}_{\cup}$ may deliver $\mathcal{X}$ as output making it totally impossible to decide on the value of x.

- An operator can be neither conjunctive nor disjunctive.

Set-valued data fusion : operator nature

- Roughly speaking, conjunctive operators are efficient because they converge to the smallest possible set of solutions but they are risky if at least one source is not fully truthful. In this latter case, $\hat{\hat{x}}_{\cap}$ may give $\emptyset$ as output, which is a typical case of conflict between sources.

- On the opposite, disjunctive operators are regarded as cautious operators because they do not discard any possible value supported by any source. However, disjunctive operators are poorly efficient. For instance, $\hat{\hat{x}}_{\cup}$ may deliver $\mathcal{X}$ as output making it totally impossible to decide on the value of x.

- An operator can be neither conjunctive nor disjunctive.

Set-valued data fusion : operators

- Let alone standard set operations like intersection and union, there is also a possibility to map sets with objects with which calculus is possible.
- These objects are indicator functions. Let A be a subset of the solution space : $A \subseteq \mathcal{X}$ or equivalently $A \in \mathbb{X} = 2^{\mathcal{X}}$. The **indicator function** 1_A of set A is a mapping such that :

$$\begin{aligned} 1_A : \mathcal{X} &\rightarrow \mathbb{R}, \\ x_0 &\rightarrow \begin{cases} 1 \text{ if } x_0 \in A \\ 0 \text{ otherwise} \end{cases}. \end{aligned} \tag{3}$$

- Indicator functions and sets are clearly in bijective correspondance.
- Indicator functions belong to a vector space, which allows us to introduce a generalization of histograms.

Set-valued data fusion : operators

- Let alone standard set operations like intersection and union, there is also a possibility to map sets with objects with which calculus is possible.

- These objects are indicator functions. Let A be a subset of the solution space : $A \subseteq \mathcal{X}$ or equivalently $A \in \mathbb{X} = 2^{\mathcal{X}}$. The **indicator function** 1_A of set A is a mapping such that :

$$\begin{aligned} 1_A : \mathcal{X} &\rightarrow \mathbb{R}, \\ x_0 &\rightarrow \begin{cases} 1 \text{ if } x_0 \in A \\ 0 \text{ otherwise} \end{cases} . \end{aligned} \tag{3}$$

- Indicator functions and sets are clearly in bijective correspondance.

- Indicator functions belong to a vector space, which allows us to introduce a generalization of histograms.

Set-valued data fusion : operators

- Let alone standard set operations like intersection and union, there is also a possibility to map sets with objects with which calculus is possible.

- These objects are indicator functions. Let A be a subset of the solution space : $A \subseteq \mathcal{X}$ or equivalently $A \in \mathbb{X} = 2^{\mathcal{X}}$. The **indicator function** 1_A of set A is a mapping such that :

$$
\begin{aligned}
1_A : \mathcal{X} &\rightarrow \mathbb{R}, \\
x_0 &\rightarrow \begin{cases} 1 \text{ if } x_0 \in A \\ 0 \text{ otherwise} \end{cases} .
\end{aligned}
\tag{3}
$$

- Indicator functions and sets are clearly in bijective correspondance.

- Indicator functions belong to a vector space, which allows us to introduce a generalization of histograms.

Set-valued data fusion : operators

- Let alone standard set operations like intersection and union, there is also a possibility to map sets with objects with which calculus is possible.

- These objects are indicator functions. Let A be a subset of the solution space : $A \subseteq \mathcal{X}$ or equivalently $A \in \mathbb{X} = 2^{\mathcal{X}}$. The **indicator function** 1_A of set A is a mapping such that :

$$\begin{aligned} 1_A : \mathcal{X} &\rightarrow \mathbb{R}, \\ x_0 &\rightarrow \begin{cases} 1 \text{ if } x_0 \in A \\ 0 \text{ otherwise} \end{cases} . \end{aligned} \tag{3}$$

- Indicator functions and sets are clearly in bijective correspondance.

- Indicator functions belong to a vector space, which allows us to introduce a generalization of histograms.

Set-valued data fusion : set-histograms

Definition

Let $\left\{x_*^{(i)}\right\}_{i=1}^{N_s}$ denote a set-valued dataset such that $\forall i$, $x_*^{(i)} \in 2^{\mathcal{X}}$. The **set-histogram** h of the dataset $\left\{x_*^{(i)}\right\}_{i=1}^{N_s}$ is the a mapping such that :

$$\begin{aligned} h : \mathcal{X} &\longrightarrow \mathbb{N}, \\ x &\longrightarrow \sum_{i=1}^{N_s} 1_{x_*^{(i)}}(x). \end{aligned} \tag{4}$$

In other words, $h(x)$ is the number of sets in the dataset containing the value x.

Set-valued data fusion : operators from set-histograms

- Using this set-histogram all definitions of voting operators from the previous chapter immediatly apply : $\hat{\hat{x}}_{maj}$, $\hat{\hat{x}}_{smj}$, $\hat{\hat{x}}_{amj}$ and $\hat{\hat{x}}_{mod}$.
- This is tantamount to voting with approval ballots.

Example

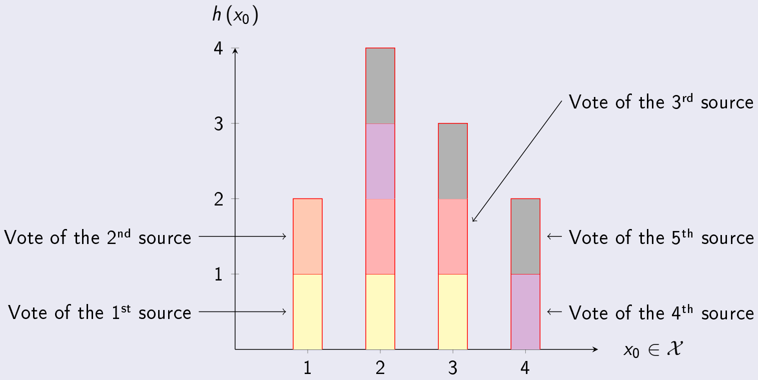
Suppose we have $\mathcal{X} = \{1; 2; 3; 4\}$ and the sources delivered the following advocacies :

- $x_*^{(1)} = \{1; 2; 3\}$,
- $x_*^{(2)} = \{1\}$,
- $x_*^{(3)} = \{2; 3\}$,
- $x_*^{(4)} = \{2; 4\}$,
- $x_*^{(5)} = \{2; 3; 4\}$.

The corresponding set histogram is : $\mathbf{h} = [2\ 4\ 3\ 2]$.

Set-valued data fusion : operators from set-histograms

Example



The contributions of the sources are in yellow for 1st source, in orange for the 2nd source, in red for the 3rd source, in violet for the 4th source and in grey for the 5th source. .

We have $\hat{\hat{x}}_{maj} = \hat{\hat{x}}_{mod} = 2$ while $\hat{\hat{x}}_{\cap} = \emptyset$ and $\hat{\hat{x}}_{\cup} = \mathcal{X}$.

Set-valued data fusion : example in Machine Learning

- Suppose you want to use SVM to solve a classification task with 3 classes.

- But SVM cannot be generalized to multi-class problems !

- Solution : train 2 SVM in a 1-versus-all fashion.

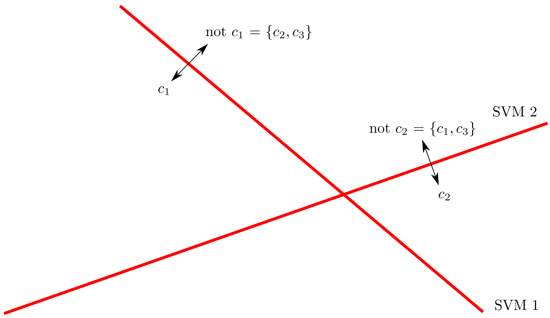
Set-valued data fusion : example in Machine Learning

- Suppose you want to use SVM to solve a classification task with 3 classes.
- But SVM cannot be generalized to multi-class problems !
- Solution : train 2 SVM in a 1-versus-all fashion.

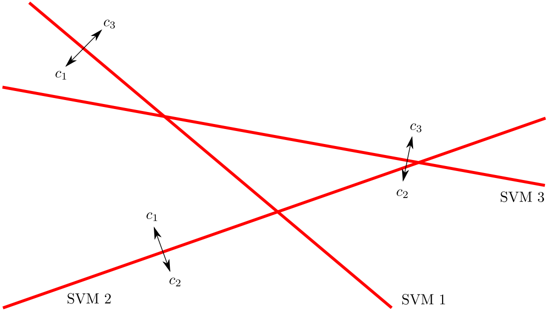
Set-valued data fusion : example in Machine Learning

- Suppose you want to use SVM to solve a classification task with 3 classes.
- But SVM cannot be generalized to multi-class problems !
- Solution : train 2 SVM in a 1-versus-all fashion.

SVM and multi-class problems : a solution using multi-valued data fusion



SVM and multi-class problems : another possiblility with point data fusion



Chapter organization

Posets :

Definition

Suppose $\mathbb{X}$ is an abstract space with $\leq$ a binary relation that is

(i) reflexive, *i.e.* $a \leq a$, $\forall a \in \mathbb{X}$,

(ii) antisymmetric, , *i.e.* $a \leq b$ and $b \leq a$ implies $a = b$, $\forall a, b \in \mathbb{X}$,

(iii) and transitive, *i.e.* $a \leq b$ and $b \leq c$ implies $a \leq c$, $\forall a, b, c \in \mathbb{X}$.

$\leq$ is a **partial order** on $\mathbb{X}$ and $(\mathbb{X}, \leq)$ is called a partially ordered set, or **poset** for short.

Bounds :

Definition

Let (a, b) be a pair of element in $\mathbb{X}$. An **upper bound** of $\{a; b\}$ is an element $u \in \mathbb{X}$ such that $a \leq u$ and $b \leq u$. The **least upper bound** (lub) of $\{a; b\}$ is the unique element $v \in \mathbb{X}$ (if it exists) such that v is an upper bound of $\{a; b\}$ and for any upper bound u of $\{a; b\}$ we have $v \leq u$.

Definition

Conversely, a **lower bound** of $\{a; b\}$ is an element $l \in \mathbb{X}$ such that $l \leq a$ and $l \leq b$. The **greatest lower bound** (glb) of $\{a; b\}$ is the unique element $w \in \mathbb{X}$ (if it exists) such that w is a lower bound of $\{a; b\}$ and for any lower bound l of $\{a; b\}$ we have $l \leq w$.

Bounds : example

Example

Let us suppose that $\mathbb{X}$ is made of subsets of a given set $\mathcal{X}$ and the partial order we are interested in is the inclusion $\subseteq$.

- If $\mathbb{X} = \{\{a\};\{b\};\{a;c\};\{b;d\}\}$ with a, b, c and $d \in \mathcal{X}$, then $\{\{a\};\{b\}\}$ has no upper bound because $\{b\} \not\subseteq \{a;c\}$ and $\{a\} \not\subseteq \{b;d\}$.
- If $\mathbb{X} = \{\{a\};\{b\};\{a;b;c\};\{a;b;c;d\}\}$ with a, b, c and $d \in \mathcal{X}$, then $\{a;b;c;d\}$ is an upper bound of $\{\{a\};\{b\}\}$ and $\{a;b;c\}$ is the least upper bound of $\{\{a\};\{b\}\}$.

Lattice :

Definition

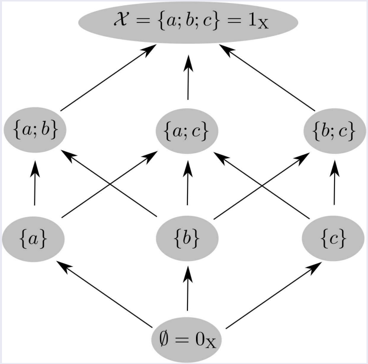
If any $\{a; b\} \subseteq \mathbb{X}$ has a lub denoted by $a \vee b$ and a gld denoted by $a \wedge b$, then $(\mathbb{X}, \leq, \wedge, \vee)$ is called a **lattice**.

- Note that the above definition is given for a pair of elements in $\mathbb{X}$. This definition is formally equivalent to a definition with a finite set of elements in $\mathbb{X}$.
- However, this is not true for infinite sets of elements in $\mathbb{X}$. If this latter property also holds, then the lattice is said to be **complete**.

Lattice : example

Example

Suppose $\mathbb{X}$ is the power set of a given set $\mathcal{X} = \{a; b; c\}$, then $\{2^{\mathcal{X}}, \subseteq, \cap, \cup\}$ is an example of complete lattice. Here is the corresponding Hasse diagram :



The arrow stands for the inclusion partial order.

Lattice : example

Example

As for an example of incomplete lattice, let us suppose that $\mathbb{X} =]0;1[$ endowed with the classical total order on reals $\leq$. For any $a \in \mathbb{X}$, let $b = a + \frac{1-a}{2}$. We have $b \in \mathbb{X}$ and $a \leq b$, hence a is not a lub of $\mathbb{X}$. Consequently, one cannot find a greatest element in $\mathbb{X}$.

Lattice : property

Property

Suppose $(\mathbb{X}, \leq, \wedge, \vee)$ is a lattice, then $\vee$ and $\wedge$ are two **monotone binary operations** with respect to $\leq$. For any a_1, a_2, b_2 and $b_2 \in \mathbb{X}$ such that $a_1 \leq a_2$ and $b_1 \leq b_2$:

$$
\begin{aligned}
a_1 \vee b_1 &\leq a_2 \vee b_2, && (5)\\
a_1 \wedge b_1 &\leq a_2 \wedge b_2. && (6)
\end{aligned}
$$

Lattice : property

Definition

A **bounded lattice** is a lattice $(\mathbb{X}, \leq, \wedge, \vee)$ with a greatest $1_\mathbb{X}$ and a least element $0_\mathbb{X}$, *i.e.* for any $a \in \mathbb{X}$, we have $0_\mathbb{X} \leq a \leq 1_\mathbb{X}$.

Definition

A **distributive lattice** is a lattice $(\mathbb{X}, \leq, \wedge, \vee)$ such that $\wedge$ is distributive over $\vee$, *i.e.* for any a, b and $c \in \mathbb{X}$, we have :

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c). \tag{7}$$

Lattice : property

Definition

A **complemented lattice** is a bounded lattice $(\mathbb{X}, \leq, \wedge, \vee)$ such that any $a \in \mathbb{X}$ has a complement $\neg a$, *i.e.* an element $b \in \mathbb{X}$ with :

$$a \wedge b = 0_{\mathbb{X}}, \quad (8)$$

$$a \vee b = 1_{\mathbb{X}}. \quad (9)$$

Some fusion problems are also often defined on Boolean algebras which are complemented distributive lattices. For instance $\left(2^{\mathcal{X}}, \subseteq, \cap, \cup, .^{c}\right)$ is a Boolean algebra. Propositional logic and positional calculus are intimately connected to Boolean algebras. This formalism is appealing for modeling deductive arguments in natural language. Boolean searches are notably implemented by Google search engine.