Interval computing applied to parameter estimation

Anthony WELTE
Class of 2017
Major in Robotics

Supervisor: Vladik Kreinovich
Tutor: Luc Jaulin
Address: UTEP, 500 W University Ave, El Paso, TX 79902, United States

October 10, 2016

Abstract

Parameter estimation is an essential part of robotics. In order to make decisions a robot has to know its environment. It must be able to locate itself or to locate others. A lot of work has been done these last decade to bring new and more efficient methods of dealing with those problems. In particular, Interval Analysis has been used for several years to solve parameter estimation problems.

Interval Analysis has brought a new way to take into consideration the uncertainty of the sensors. It has also shown its robustness in dealing with outliers (measurement failed, inaccurate). However, there are still a few problems left.

Among those are Fake Boundaries, sets that classic algorithms using intervals get stuck on and cannot decide if they are solutions or not. We can remove those Fake Boundaries by reformulating the problem. Indeed, using the Disjunctive Normal Form of the problem the unclassified sets can be classified.

Another problem of interval analysis is outliers. GOMNE algorithm has proven robust to deal with outlier but it has an issue. To remove ouliers it test every possible number of outlier until it manages to find a solution to the problem. Also it requires to have an idea of the distribution of the error in order to bound this error efficiently. Using the Maximum likelihood estimator we have a criteria that can rate parameters without needing any information about the error.

Résumé

Estimer des paramètres est une partie importante de la robotique. Pour pouvoir prendre des décisions, un robot doit connaitre son environnement, il doit être capable de se localiser ou de localiser d'autres robots. Ces dernières années de grandes avancées ont été faites dans le domaine de l'estimation de paramètre. En particulier, le calcul par intervalle est utilisé depuis quelques années pour résoudre se type de problèmes.

Le calcul par intervalle apporte une nouvelle manière de traiter les imprécisions des capteurs. De plus cette methode est robuste aux *outliers* (mesure fausse). Cependant tous les problèmes n'ont pas été résolus.

Un de ces problèmes est les *Fake Boundaries*, des ensembles de paramètres pour lesquels l'algorithme n'arrive pas à identifier s'ils sont solutions du problème ou non. En reformulant le problème en sa Forme Normale Disjunctive, on peut supprimer ses ensembles et résoudre le problême.

Un autre problème du calcul par intervalle est les *outliers*. L'algorithme GOMNE est capable d'estimer des paramètres mêmes en la présence d'*outliers*. Pour cela il test tous les nombres d'*outliers* possible jusqu'à obtenir un problème résoluble. Il requière une certaine connaissance de l'erreur sur les mesures. En utilisant l'estimateur maximum de vraisemblance on a un critère qui nous permet de choisir les paramètres idéaux au problème sans avoir besoin de connaitre la distribution de l'erreur.

Keywords

Interval analysis, GOMNE, Fake Boundaries, Parameter estimation, Maximum Likelihood

Thanks

I would like to thank everyone who supported me during my internship. Professor Luc JAULIN from ENSTA Bretagne for teaching me about Interval Analysis and introducing me to Professor Vladik KREINOVICH and Professor Martine CEBRIO.

I also thank M. KREINOVICH and Ms. CEBRIO for welcoming me in their laboratory in the University of Texas at El Paso (UTEP) and for showing me the city of El Paso.

And finally, I thank Leobardo VALERA and Afshin Gholamy for support and friendship.

Contents

Introduction

In robotics, estimating parameters such as positions and distances can be essential to the mission of a robot. But of course in practical situations we are not able to determine precisely those things. To obtain those parameters we rely on sensors. Such devices are not perfect, so the data they provide is inaccurate, they have a non negligible **uncertainty**. They can also give false data when something passes in front of the sensor for instance. Those values are called **outliers**.

As a result parameter estimation is challenging. In order to achieve our goal many methods exist. Probabilistic methods have been used for decades. However, they usually only work in specific cases and require the problem to have some characteristics. For instance the Kalman filter is a widely used estimator that has proved its efficiency, nonetheless it was only designed to be used for linear problems with Gaussian uncertainty. New methods have been developed to cope with these disadvantages. Interval analysis has brought a new way to deal with such problems but work still needs to be done. Interval analysis solves some issues from probabilistic methods, but it also introduces new ones.

During my internship, I mainly worked on two problems.

First, I worked on eliminating fake boundaries when working with contractors. Contractors make it possible to speed up computation by efficiently shrinking the interval sets. However, in some particular situations they are unable to identify if a set is solutions or not, those sets are unclassified. A human would be able to tell instantly if the sets are solutions or not because those unclassified sets appear on boundaries of sets, they are therefore called fake boundaries. Fortunately, some ideas have emerged on how to deal with such problems. My goal was to combine two methods that eliminate those Fake boundaries in order to obtain a simpler method.

I also took part in the development of a new way to use interval analysis for parameter estimation. This new method tries to deal with the two main problems when estimating something, the uncertainty of the sensor that causes an error on the measurement and outliers that are far from the real values and are caused by sensor failures or environmental events (something getting in front of the sensor, etc.).

Chapter 1

The University of Texas at El Paso (UTEP)

1.1 Presentation

The University of Texas at El Paso is a public research university. It was founded in 1914 and was primarily focused on mines and metallurgy. It host over 23.000 students and 2.800 staff members (1300 academic staff, 1500 administrative staff). Diana Natalicio became the first female president of UTEP in 1988 and has been holding this position since then. She has pushed for a better integration of the Hispanic community in the university to better reflect the demographic of El Paso (80% of Hispanic students). The university is composed of six colleges that cover various disciplines, Business, Education, Engineering, Health Sciences, Liberal Arts and Science. It also host a School of nursing and a School of Pharmacy.

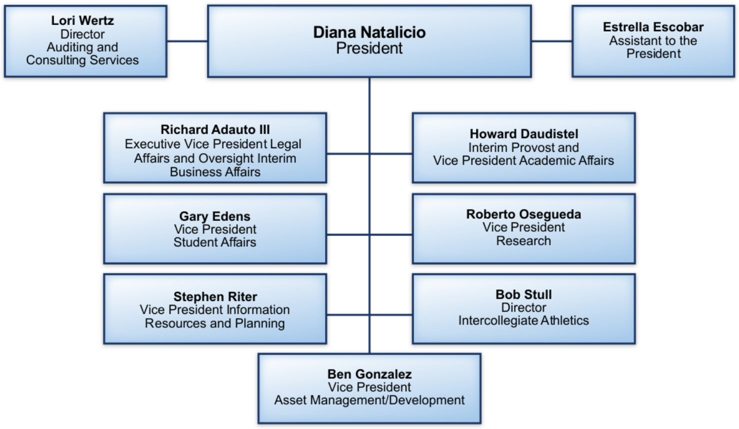


Figure 1.1: President and Executive officers

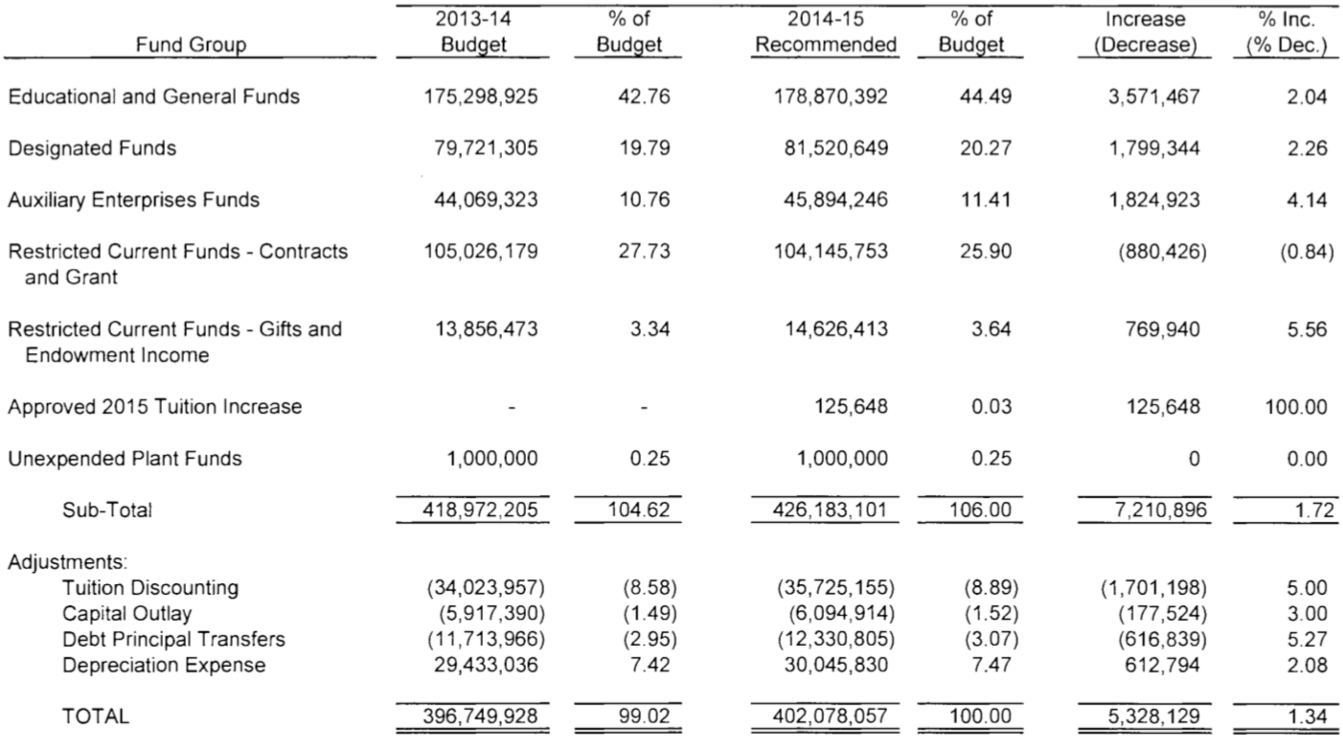
The university spends a lot on research and has been responsible for the several discoveries in Health Science (vaccine against Chagas disease, ...). The mechanical engineering department is supported by big companies such as Lockheed Martin Corporation. NASA Aerospace also helped finance a laboratory in the university.

1.2 The Computer Science Department

During my internship I worked in the Computer Science Department. The center I joined works on Theoretical Research and Applications in Computer Science. It focuses on noisy data processing, computer security and privacy and reasoning under uncertainty. The center is divided in three research groups: ELMOS (Electronic Money and Security), IF (Interval and Fuzzy, for processing of uncertainty) and GEOINFORMATICS (algorithm for processing geospatial data). Despite this three distinct groups, researchers can participate in several projects.

1.3 Economic analysis

The university has a major economic impact in El Paso. It has an operating budget of over $400 million and its economic impact in El Paso County is estimated to be around $1.4 billion. With almost 3.000 staff members it is 5th largest employer in El Paso.



Fund Group	2013-14 Budget	% of Budget	2014-15 Recommended	% of Budget	Increase (Decrease)	% Inc. (% Dec.)
Educational and General Funds	175,298,925	42.76	178,870,392	44.49	3,571,467	2.04
Designated Funds	79,721,305	19.79	81,520,649	20.27	1,799,344	2.26
Auxiliary Enterprises Funds	44,069,323	10.76	45,894,246	11.41	1,824,923	4.14
Restricted Current Funds - Contracts and Grant	105,026,179	27.73	104,145,753	25.90	(880,426)	(0.84)
Restricted Current Funds - Gifts and Endowment Income	13,856,473	3.34	14,626,413	3.64	769,940	5.56
Approved 2015 Tuition Increase	-	-	125,648	0.03	125,648	100.00
Unexpended Plant Funds	1,000,000	0.25	1,000,000	0.25	0	0.00
Sub-Total	418,972,205	104.62	426,183,101	106.00	7,210,896	1.72
Adjustments:						
Tuition Discounting	(34,023,957)	(8.58)	(35,725,155)	(8.89)	(1,701,198)	5.00
Capital Outlay	(5,917,390)	(1.49)	(6,094,914)	(1.52)	(177,524)	3.00
Debt Principal Transfers	(11,713,966)	(2.95)	(12,330,805)	(3.07)	(616,839)	5.27
Depreciation Expense	29,433,036	7.42	30,045,830	7.47	612,794	2.08
TOTAL	396,749,928	99.02	402,078,057	100.00	5,328,129	1.34

Figure 1.2: The budget of UTEP for 2013-14

1.3.1 Method of financing

UTEP gets its funding from different sources. Of course tuitions and student fees account for a big part of the university's budget (25%). However the efforts of the direction to make education affordable has led UTEP to have the lowest net price for a research university in the United

States. Therefore other income sources have to compensate for that. The university makes around $40 million with its assets (parking lots, housing, food services contract, Intercollegiate Athletics contract). Over a $100 million comes from contracts and grant. Most of that coming from financial aid programs and federal research. For instance the research done by the team I joined was funded by the National Foundation grants CAREER 0953339. Other smaller fundings complete the budget.

1.3.2 Spendings

Two categories represent most of the spendings.

Education is a big chunk of the spendings. The staff salaries account for a quarter of the total spendings. Also the university wants to provide accessible tuition, so it provides students with scholarships ($20 million).

Research is the second major source of expenditure. All of the money from contracts and grant is directly invested in the research activities of the university, it pays for the researchers' salaries and for the equipments.

Chapter 2

Introduction to interval analysis [1]

Intervals

An interval is a closed and connected subset of $\mathbb{R}$. For example $[-5;6]\,, \{7\}\,,]-\infty;3]$ are intervals.
An interval can be defined by an lower bound x^- and an upper bound x^+ such as:

$$[x] = \left[x^-; x^+\right] = \left\{x \in \mathbb{R}, x^- \leq x \leq x^+\right\}$$

Just like for real numbers basic operators $+, -, *, /$ exist. There is also the intersection operator $\cap$ and union operator $\cup$.
For $\diamond = \{+, -, *, /, \min, \max\}$:

$$[x] \diamond [y] = \{x \diamond y | x \in [x]\,, y \in [y]\}$$

Function can also be defined for intervals. For instance,

$$f\left([x]\right) = \{f(x)|\; x \in [x]\}$$

Boxes

A box $[x]$ is an interval of $\overline{\mathbb{R}}^n$. It is a vector of intervals $[x_i]$:

$$[X] = \left[x_1^-; x_1^+\right] * \left[x_2^-; x_2^+\right] * \ldots * \left[x_n^-; x_n^+\right] = [x_1] * [x_2] * \ldots * [x_n]$$

Contractor

A common way to work with intervals is to bisect the intervals into two and two compute the two part separately. Therefore, as we bisect we get a more and more accurate solution of our problem. However this has a major disadvantage. Indeed the complexity of those algorithms is exponential relative to the accuracy we want. Contractors help to deal with this issue, they don't change the computational complexity of the algorithm but they make it possible to shrink the intervals faster by removing parts of the intervals that are not in the range of solution as seen on figure 2.1.

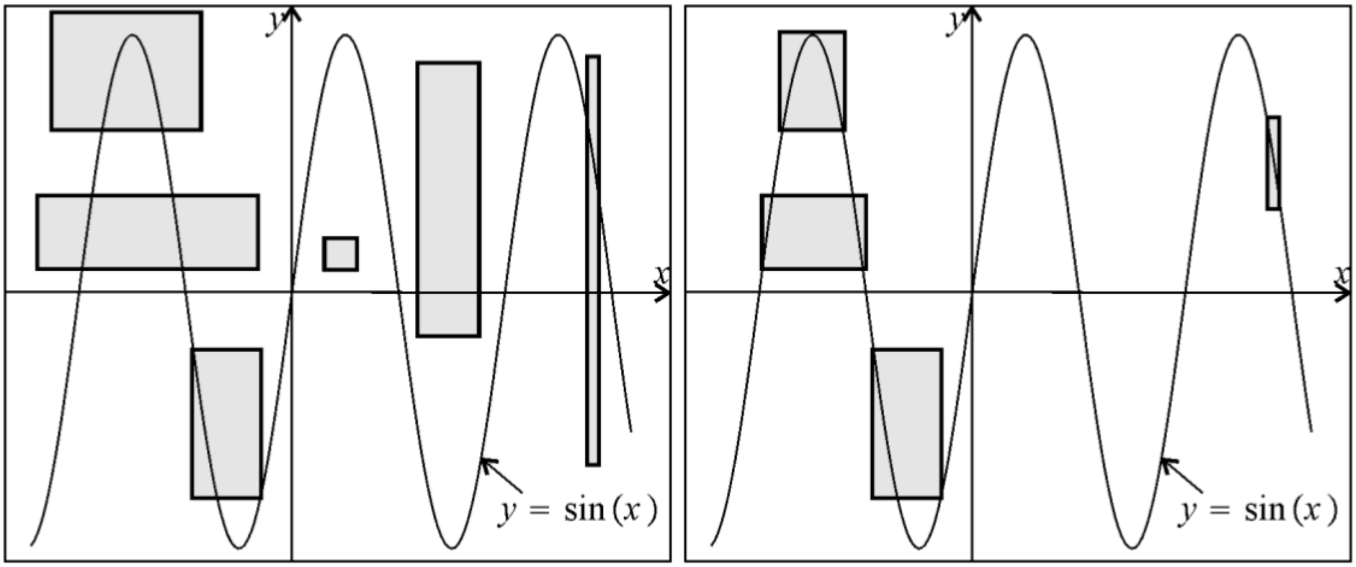


Figure 2.1: Effect of contractor on sample sets

Definition 1 *The operator* $\mathrm{L} : \mathbb{IR}^n \to \mathbb{IR}^n$ *is a contractor if*

- $\forall [x] \in \mathbb{IR}^n, \mathrm{L}([x]) \subset [x]$ *(contractance)*
- $(x \in [x], \mathrm{L}(\{x]\}) \implies x \in \mathrm{L}([x])$ *(consistence)*
- $\mathrm{L}(\{x\}) = \emptyset \implies (\exists \epsilon > 0, \forall [x] \subset B(x, \epsilon), \mathrm{L}([x]) = \emptyset)$ *(weak continuity)*

Chapter 3

Fake boundaries

The problem

When solving problems using a separator programming approach, we can sometime obtain unwanted boundaries. Boundaries that are clearly in the solution set or out of the solution set. Those boundaries have been called fake boundaries because the algorithm used to solve the problems see them as unclassified sets but a human is able to tell instantly if it belongs or not to the solution. These boundaries appear when working with intersections, unions and complement of sets.

Indeed if we look at four sets $\mathbb{A}, \mathbb{B}$ and $\mathbb{C}$ defined such that,

$$\begin{aligned}\mathbb{A} &= \left\{(x,y) \in \mathbb{R}^2 | (x+2)^2 + (x-2)^2 \leq 3^2\right\}\\ \mathbb{B} &= \left\{(x,y) \in \mathbb{R}^2 | (x+2)^2 + (x+2)^2 \leq 3^2\right\}\\ \mathbb{C} &= \left\{(x,y) \in \mathbb{R}^2 | (x-2)^2 + (x+2)^2 \leq 3^2\right\}\end{aligned}$$

And if we are searching for $\mathbb{X}$ so that,

$$\mathbb{X} = (\overline{\mathbb{A}} \cup \overline{\mathbb{B}} \cup \mathbb{C}) \cap (\overline{\mathbb{A}} \cup \mathbb{B} \cup \overline{\mathbb{C}}) \cap (\mathbb{A} \cup \overline{\mathbb{B}} \cup \overline{\mathbb{C}}) \cap (\mathbb{A} \cup \overline{\mathbb{B}}) \cap (\mathbb{A} \cup \overline{\mathbb{B}} \cup \mathbb{C}) \cap (\mathbb{A} \cup \mathbb{C})$$

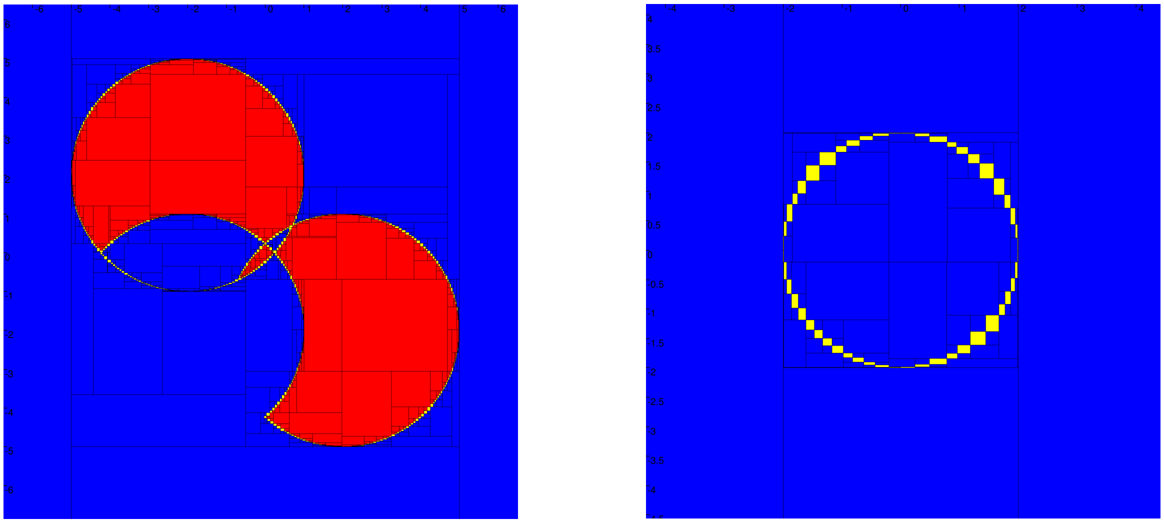
We obtain the result shown on figure 3.1a.

The cause

Several papers [2], [4] have shown that the existence of fake boundaries is directly linked to the way the expression is written. Indeed those boundaries appear when the edge of a set is hidden in the expression. For instance if we have the expression $\mathbb{Y} = \mathbb{A} \cap \overline{\mathbb{A}}$, it is obvious that the set $\mathbb{Y}$ is empty. However if we try to compute it using a contractor based approach, the boundary of the set $\mathbb{A}$ will appear unclassified as shown on figure 3.1b. This is because contractor based approach don't check if a given set is solution, instead they progressively shrink the sets around the edges.

Using the full DNF form

In order to remove fake boundaries several methods exist. It has been shown that using the full-DNF (Disjunctive normal form) of an expression, we can remove the unwanted boundaries. Any expression has a unique full-DNF defined in Definition 2.



(a) Representation of $\mathbb{X}$

(b) Representation of $\mathbb{Y} = \mathbb{A} \cap \overline{\mathbb{A}}$

Figure 3.1: Fake boundaries

Definition 2 *The grammar of a DNF is:*

- *disjunction* $\leftarrow$ (*conjunction* $\cup$ *disjunction*)
- *disjunction* $\leftarrow$ *conjunction*
- *conjunction* $\leftarrow$ (*literal* $\cap$ *conjunction*)
- *conjunction* $\leftarrow$ *literal*
- *literal* $\leftarrow \mathbb{A}_i$
- *literal* $\leftarrow \overline{\mathbb{A}_i}$

A DNF is called full if each of its variables appears exactly once in every clause.

This method is not perfect, to obtain the full DNF of an expression, we need to test every tuple $(\mathbb{A}_0, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ to $\left(\overline{\mathbb{A}_0}, \overline{\mathbb{A}_1}, \ldots, \overline{\mathbb{A}_n}\right)$. Therefore the computational complexity of the method is high, we have 2^n combination to test.

Using a Karnaugh map

Another method is to use Karnaugh map [2]. This method also has disadvantages. Building Karnaugh map of an expression is not simple and in some cases we will also have to test every combination. This method has however the advantage of producing simpler expression to process. Thus, it will slightly reduce the computing time of the algorithms used to solve the problem such as SIVIA (Set Inversion via Interval Analysis).

A simpler method

The last method proves that we don't need a full DNF to remove fake boundaries. The idea of this method is to simply use the DNF form of the expression. Finding the DNF of an expression doesn't necessarily means testing every combination of inputs. It can be obtained using the distributivity of the union and intersection operators and equivalence rules to modify the expression. Therefore, we can obtain an expression that doesn't produce fake boundaries without having to test every combination. By developing (using the distributivity) and simplifying the expression

$$\mathbb{X} = (\overline{\mathbb{A}} \cup \overline{\mathbb{B}} \cup \mathbb{C}) \cap (\overline{\mathbb{A}} \cup \mathbb{B} \cup \overline{\mathbb{C}}) \cap (\mathbb{A} \cup \overline{\mathbb{B}} \cup \overline{\mathbb{C}}) \cap (\mathbb{A} \cup \overline{\mathbb{B}}) \cap (\mathbb{A} \cup \overline{\mathbb{B}} \cup \mathbb{C}) \cap (\mathbb{A} \cup \mathbb{C})$$

we can obtain the expression

$$\mathbb{X} = (\mathbb{A} \cap \overline{\mathbb{B}} \cap \overline{\mathbb{C}}) \cup (\overline{\mathbb{A}} \cap \overline{\mathbb{B}} \cap \mathbb{C}) \cup (\mathbb{A} \cap \mathbb{B} \cap \mathbb{C})$$

using this expression we obtain the result shown on figure 3.2.

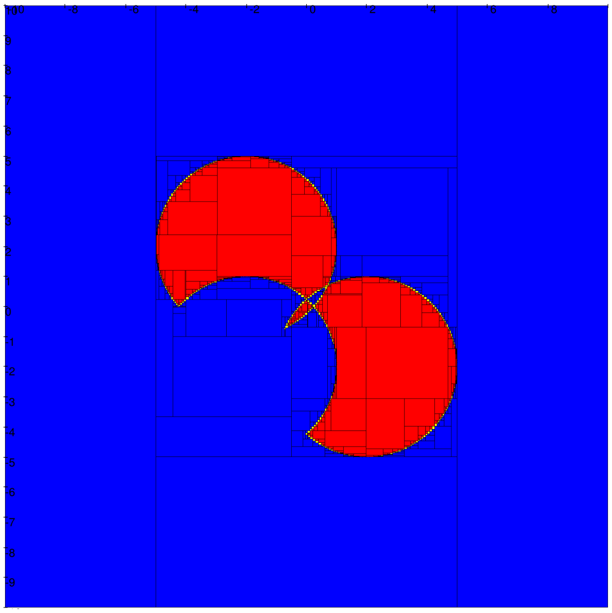


Figure 3.2: Representation of $\mathbb{X}$ (using the new method).

Chapter 4

Parameter estimation

4.1 GOMNE Algorithm

In parameter estimation problems, we usually have three things:

- A series of inputs $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$
- A series of measurement $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ obtained using sensors
- An idea of the relation f between the inputs $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and the parameter $p_0, p_1, \ldots, p_l$ that results in the measurements.

So given the inputs $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and the measurements $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ we want to solve the system of equations:

$$\begin{cases} f(\boldsymbol{x}_0, p_0, p_1, \ldots, p_l) = \boldsymbol{y}_0, \\ f(\boldsymbol{x}_1, p_0, p_1, \ldots, p_l) = \boldsymbol{y}_1, \\ \qquad\qquad \vdots \\ f(\boldsymbol{x}_n, p_0, p_1, \ldots, p_l) = \boldsymbol{y}_n \end{cases} \tag{4.1}$$

The Guaranteed Outlier Minimal Number Estimator (GOMNE) algorithm offers a way to find the parameters $p_0, p_1, \ldots, p_l$ that work for this equation. Moreover if no parameters fit the equation it can find the parameters that fit the maximum number of equations.

4.1.1 SIVIA Algorithm

The GOMNE algorithm uses Set Inversion via Interval Analysis (SIVIA) to obtain the parameters that solve the system of equations. It works by testing boxes P to see if the box is in the set of solution, out of the set of solution or somewhere in between (unclassified). In the last case it split the box into two and starts over. This algorithm stops when the width of the unclassified boxes is smaller than a precision ϵ.

Algorithm 1 SIVIA Algorithm

```
1:  $L \leftarrow \{[\boldsymbol{P}](0)\}$;
2:  $pull([\boldsymbol{x}], L)$;
3:  if $[f]([\boldsymbol{x}]) \subset [\boldsymbol{y}]$ then
4:      $Pin \leftarrow [\boldsymbol{x}]$
5:  else if $[f]([\boldsymbol{x}]) \cap [\boldsymbol{y}] = \emptyset$ then
6:      $Pout \leftarrow [\boldsymbol{x}]$
7:  else if $width([\boldsymbol{x}]) < \epsilon$ then
8:      $Pbound \leftarrow [\boldsymbol{x}]$
9:  else
10:     bisect $[\boldsymbol{x}]$ into $[\boldsymbol{x}](1)$ and $[\boldsymbol{x}](2)$;
11:     $L \leftarrow [\boldsymbol{x}](1)$;
12:     $L \leftarrow [\boldsymbol{x}](2)$;
13: if $L \neq \emptyset$ then
14:     go to 2
```

Of course this algorithm cannot give the exact set of solutions P. It returns the inclusion

$$P^{-} \subset P \subset P^{+}$$

where $P^{-} = Pin$
and $P^{+} = \overline{Pout} = Pbound \cup Pin$.

4.1.2 q-relaxed intersection

Just like many parameter estimation methods. Interval analysis method work better when we have a lot of measurements. However the more data we have, the more likely we are to obtain an empty set of parameters. This is the Demidenko paradox, with more data, we increase the chance to obtain outliers, values that don't match the other.

To cope with this issue, relaxed intersection can be used. In essence, we allow the parameters to satisfy only $n - q$ measurement.

4.1.3 The algorithm

Algorithm 2 GOMNE Algorithm

```
procedure TEST(P)
    $b \leftarrow [0;0]$
    for i in [1,..., size($\boldsymbol{y}$)] do
        $F_i = f(P, x_i)$
        if subset($F_i, y_i$) then
            $b[0] = b[0] + 1$
        if not_disjoint($F_i, y_i$) then
            $b[1] = b[1] + 1$
    return b
procedure RECURSIVE_SIVIA(P, Q)
    $\boldsymbol{b} = test(P)$
    if $b[1] < size(\boldsymbol{y}) - q$ then
        $Pout \leftarrow P$
    else if $b[0] >= size(\boldsymbol{y}) - q$ then
        $Pin \leftarrow P$
    else if $width\,(P) < \epsilon$ then
        $Pbound \leftarrow P$
    else
        bisect $P$ into $P_1$ and $P_2$
        $Recursive\_SIVIA(P_1, q)$
        $Recursive\_SIVIA(P_2, q)$
procedure GOMNE(P)
    $q \leftarrow 0$
    while $Pin = \emptyset$ do
        Recursive_SIVIA(P, q)
        $q \leftarrow q + 1$
```

4.2 Maximum compatibility method

To deal with uncertainty, we have seen that the GOMNE Algorithm tries to remove outliers. Another method would be to increase the size of the $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ boxes. Instead of considering that some measure are outliers and should not be considered we consider that every measure is reliable and that it is the accuracy of the measure that we overestimated. Therefore, by increasing the width of the intervals we eventually obtain a system that can be solved.

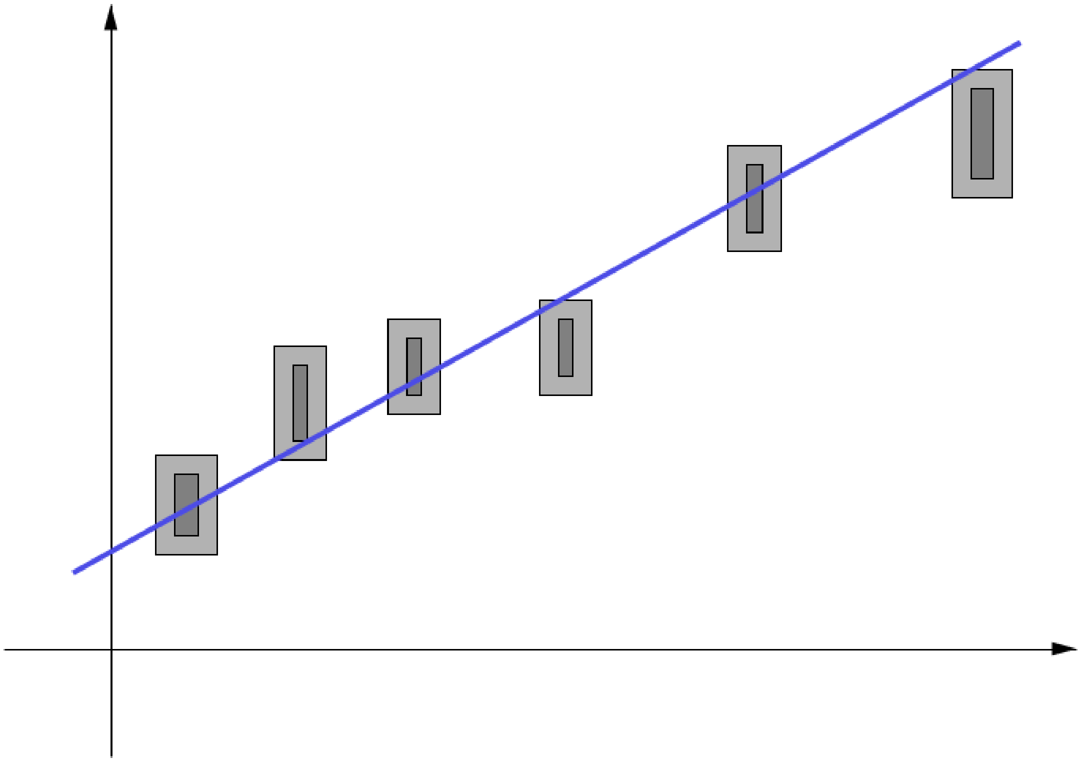


Figure 4.1: Inflating the uncertainty boxes inevitably results in a solvable system [6]

For parameters $p_0, p_1, \ldots, p_l$ the error between the measured y_i and the theoretical value $f(x_i, p_0, p_1, \ldots, p_l)$ is

$$\Delta_i = |y_i - f(x_i, p_0, p_1, \ldots, p_l)|$$

The goal is to minimize this error for all measurement, therefore we want to find $p_0, p_1, \ldots, p_l$ so that

$$\min_{p_0, p_1, \ldots, p_l} \max_{1 \leq i \leq m} |y_i - f(x_i, p_0, p_1, \ldots, p_l)|$$

We obtain a MINIMAX estimation problem which can be resolved using interval methods.

Chapter 5

The Maximum likelihood approach

5.1 The Idea

The objective of this new method is to deal with uncertainty and outliers at the same time. We are trying to solve the following problem,

$$\boldsymbol{y} = f(\boldsymbol{p}) + \boldsymbol{e}$$

$\boldsymbol{e}$ is the error vector of the measurements. To solve the problem we suppose that each element of this vector is independent and that they are distributed with the following density shown on figure 5.1.

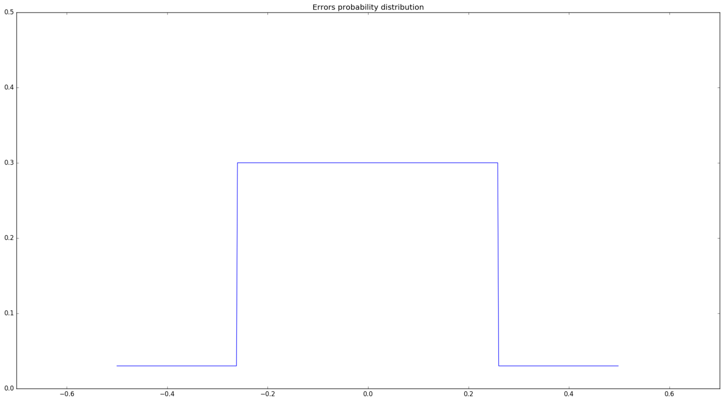


Figure 5.1: Density function

The main part, in the middle represent the data that we consider reliable. The part with a lower probability represent the outliers.

We can note that if we set the value of Δ we fall in the case of the GOMNE estimation method where the only variable left if the probability of outliers. And in the GOMNE algorithm we essentially try to find the smallest probability for which the system is solvable.

If now we set the value of the probability of outliers to zero, we are in the case of the Maximum Compatibility method. We then have to find Δ so that the error is included in $[-\Delta; \Delta]$ and solve the problem using this value.

To be able to set the value of Δ or of the probability of outlier we need some information about the distribution of the actual error. Even though we often make an hypothesis on this distribution, a common hypothesis in robotics is to suppose the distribution is Gaussian, we can't always make such assumption. The method proposed here doesn't need to have Δ of p_o constant. We will use the maximum likelihood estimator to "rate" the estimation.

The maximum likelihood estimator is defined such that

$$L \overset{def}{=} \prod_{i=1}^{N} \rho(y_i - f(p, x_i))$$

So for the aforementioned density distribution we have to maximize

$$L(p, \Delta) = \frac{1}{(2\Delta)^{N-q(p,\Delta)}} \times \frac{1}{(2W)^{q(p,\Delta)}}$$

where q is the number of outlier.

Using this formula or its logarithm we can judge whether or not a set of parameter is a good estimate of the solution.

For each p, we need to find the value

$$\max_{\Delta} \left(\frac{1}{(2\Delta)^{N-q(p,\Delta)}} \times \frac{1}{(2W)^{q(p,\Delta)}} \right)$$

We can then select the parameters that have the maximum likelihood.

5.2 The Result

In order to determine if the method has satisfying results, it was tested on a simple linear example. Several error distributions were tried to see if the method was robust to fat tailed distributions such as Cauchy distributions. A wide range of parameter was tried for each distribution. These tests had the purpose to show that the Maximum likelihood method ended up with results that were considering enough values (Δ high enough) and at the same time were not considering to many values in order to remove outliers (Δ high enough).

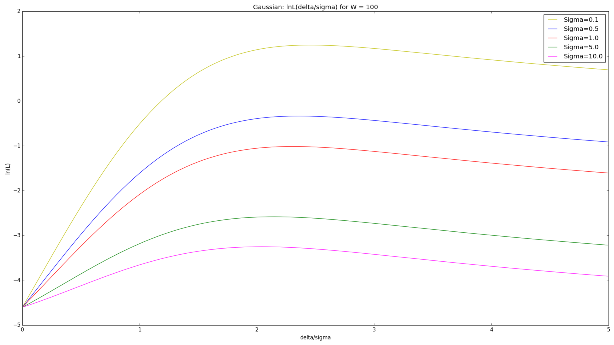


Figure 5.2: Maximum likelihood estimator (Gaussian)

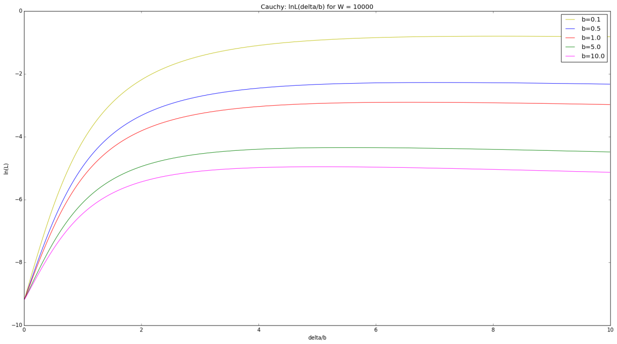


Figure 5.3: Maximum likelihood estimator (Cauchy)

On figure 5.2 we see that for a Gaussian distribution the maximum value of L is obtained for $\Delta \in [2\sigma; 2.5\sigma]$. This makes sense because 95% of the values are in this interval and its is an already commonly used confidence interval for this distribution.

Also if we look at the Cauchy distribution on figure 5.3 we see that the maximum of the function is harder to identify. Cauchy distributions are fat tailed, they have a lot of outliers, therefore finding an appropriate Δ is much harder. However if we take a look at the values for which the function reaches its maximum, we see that every times, less than 20% of the values are considered outliers. This shows that even for distributions with a lot of outliers this method can filter out those unwanted values.

Chapter 6

Practical example

6.1 Description of the problem

To show that the aforementioned method works, we try to use it to solve a classic robotics problem. Let's imagine a robot in a 2-dimensional environment with known landmarks. The location of those landmarks is known and the robot is able to measure the distances between it and the landmark. The measurements have an error that for the simulation will be either Normally distributed or will follow a Cauchy distribution. The Cauchy distribution has been chosen because it is a fat tailed distribution, it has a higher chance to produce outliers than a Normal distribution.

The system we are trying to solve can be expressed by this equation:

$$y_i = f(\boldsymbol{p}, \boldsymbol{x_i}) \quad i \in [1, N] \text{ where N is the number of measurements}$$

The y_i are the measured distances, $\boldsymbol{x_i}$ are a two dimensional vectors containing the coordinate of the landmark. Finally p is a two dimensional vector containing the position of the robot we are trying to estimate.

6.2 The Program

The program has two parts, an interface in which the user can select the setup of the simulation, and a window in which the user can see the result of the experiment.

6.2.1 Qt interface

The user interface is designed using Qt, a cross platform C++ library that provides tools to create interactive user interfaces. Our interface is in several parts.

First, the user can select the parameters of the simulation, such as the number of measurements, the uncertainty of those measurements, etc ... The user can also enter a number that will be used to generate random positions for the landmarks, that way we can compare the results for

The user can also select the method to estimate the position of the robot. Two methods are available. The GOMNE algorithm can be used in which case we need to enter the maximum error we are accepting on the measurement before treating them as outliers. If the Maximum likelihood estimation is selected no other information is needed since in this configuration the estimator doesn't do any supposition on the error.

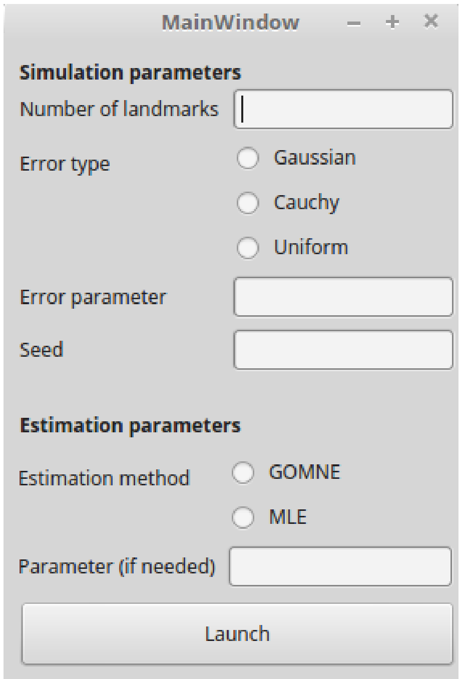


Figure 6.1: Qt interface

6.2.2 VIBES

To display the result of the parameter estimation problem VIBES is used. VIBES provides a way to easily display boxes so it is ideal to graphically represent a two dimensional problems. Results of VIBES graphical representation have already been shown in the chapter on Fake boundaries.

6.3 The Results

After some testing, we can conclude that the GOMNE algorithm does work well when we are able to set an appropriate Δ. Moreover, for common error distributions such as Gaussian distribution with a well adjusted Δ a solution is often found without having to consider any outliers. However, if the value Δ is chosen poorly we start to increase the number of oultlier (if Δ is too low) therefore the computing time. Also, if Δ is too high the size of the solution set increases which can also be a problem if we need a precise estimation.

Using the maximum likelihood method we manage to obtain an estimation of the best parameters without the need to set a value Δ. The method seems to successfully find parameters using Δs that remove outliers but keep enough values in the reliable range.

However, in its current form the last method is much slower than GOMNE. As a result it doesn't make sense to use this method if we have an idea of the type of distribution of the measurements error. Unlike GOMNE this method finds the number of outliers that maximize the likelihood of having accurate parameters, therefore it essentially add an new dimension to the problem which comes with an increase in computing time.

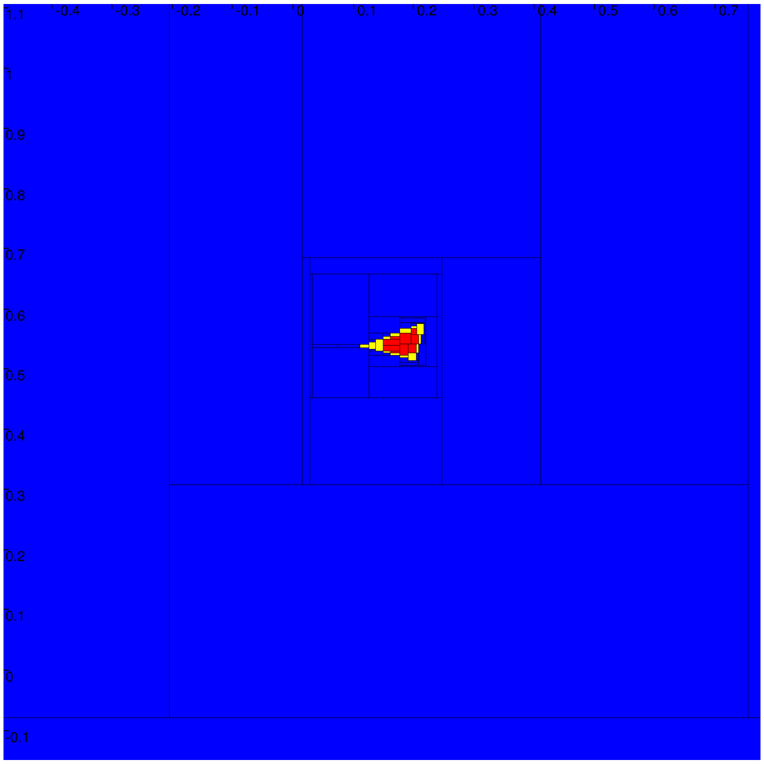


Figure 6.2: Localization problem using GOMNE (successful with two outliers)

Conclusion

This internship was successful for several reasons.

First it provided me with an international experience that cannot be obtained by other means. I was able to meet and work with people from many different origins (United-States, Venezuela, Iran, France, Mexico). This made me realize that working in an international teams has its challenges, different point of view, different ways to work. But it also showed me that the diversity in a team can be a strength and that teaming up with people from other culture can bring a new perspective to a project.

This internship was also very educational. Despite the fact that I had previously worked on interval analysis at ENSTA Bretagne, I first struggled a bit because the work done by professor Kreinovich is much more theoretical and focuses on the mathematics involved. However after some time I started to understand better the mathematical aspects of interval analysis. It gave me a much more complete understanding of this field.

This also enable me to see and experience the way a research team works. I realized that doing research does not mean working on a single project. The researchers I have worked with were involved in many different project simultaneously. This comforted me since I have a hard time focusing on one specific project for a long time. Of course working on multiple things at once has it disadvantages especially after having spent some time on another project it is hard to get back into another but it also help to take a step back from something to get a new perspective.

Bibliography

[1] L. Jaulin, M. Kiefer, O. Dicrit, and E. Walter, *Applied Interval Analysis*, Springer, London, 2001.

[2] G. S. Franco and L. Jaulin, “Avoiding fake boundaries in interval analysis”, *Abstracts of the Summer Workshop on Interval Methods SWIM’2016*, Lyon, France, June 19–22, 2016.

[3] J. Sliwka, L. Jaulin, M. Ceberio, and V. Kreinovich, “Processing Interval Sensor Data in the Presence of Outliers, with Potential Applications to Localizing Underwater Robots", *Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics SMC’2011*, Anchorage, Alaska, October 9–12, 2011, pp. 2330–2337.

[4] J. T. Yao, Y. Y. Yao, V. Kreinovich, P. Pinheiro da Silva, S. A. Starks, G. Xiang, and H. T. Nguyen, “Towards more adequate representation of uncertainty: from intervals to set intervals, with the possible addition of probabilities and certainty degrees”, *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE’2008*, Hong Kong, China, June 1–6, 2008, pp. 983–990.

[5] L. Jaulin, “Reliable minimax parameter estimation”, *Reliable Computing*, 2001, Vol. 7, No. 3, pp. 231–246.

[6] V. Kreinovich and S. Shary, “Interval Methods for Data Fitting under Uncertainty: A Probabilistic Treatment”, *Reliable Computing*, 2016, Vol. 23, pp. 105–141.

[7] J. Nicola and L. Jaulin, “OMNE is a Maximum Likelihood estimator”, *Abstracts of the 7th International Workshop on Reliable Engineering Computing REC’2016*, Bochum, Germany, June 15–17, 2016.

Appendix A

Appendices

Avoiding Fake Boundaries in Set Interval Computing

Anthony Welte and Luc Jaulin
Lab STICC
École Nationale Supérieure
de Techniques Avancées Bretagne
(ENSTA Bretagne)
2 rue François Verny
29806 Brest, France
Emails: tony.welte@gmail.com,
lucjaulin@gmail.com

Martine Ceberio and Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, Texas 79968, USA
Emails: mceberio@utep.edu,
vladik@utep.edu

***Abstract*—Set intervals techniques are an efficient way of dealing with uncertainty in spatial localization problems. In these techniques, the desired set (e.g., set of possible locations) is represented by an expression that uses intersection, union, and complement of input sets – which are usually only known with interval uncertainty. To find the desired set, we can, in principle, perform the corresponding set-interval computations one-by-one. However, the estimates obtained by such straightforward computations often contain extra elements – e.g., fake boundaries. In this paper, we show that we can eliminate these fake boundaries (and other extra elements) if we first transform the original set expression into an appropriate DNF/CNF form.**

I. Formulation of the Problem

Location and mapping problems are important. In many practical situations, we are interested in location and mapping: we want to find the location of an object, and we want to find the exact boundaries of a region.

In many cases, we can use the GPS signals to get reasonably accurate locations of different objects. This possibility is based on the fact that in most situations, electromagnetic propagate in the atmosphere with a known (and practically constant) speed and along straight lines.

For underwater objects, however, determining the exact location is not easy: radio-signals like the GPS signals do not penetrate in water. In principle, we can use sound signals to "ping" the object and thus, determine its location. However, due to inhomogeneity of water and to the presence of many potential obstacles, the direction and speed of a sound signal may change as the signal propagates.

Set computation: a useful tool for solving location and mapping problems. To locate an underwater object, we usually perform several measurements. Based on each measurement, we can find the set S of all the locations x which are consistent with the measurement results. In this case, if we perform n measurements, and find the corresponding sets $S_1, \ldots, S_n$, then we can conclude that the actual location x belongs to all these sets. In this case, the set S of all possible locations x is the intersection of the n sets corresponding to n measurement results: $S = S_1 \cap \ldots \cap S_n$.

We also know that the underwater object is in the water, so it cannot be inside the 3-D areas that were already identified as underwater rocks or peers. If we denote the corresponding "impossible-to-be" sets by $I_1, \ldots, I_m$, then we can get a better description of the set of possible locations S as the difference

$$S = (S_1 \cap \ldots \cap S_n) - (I_1 \cup \ldots \cup I_m),$$

where $A - B$ denotes the set difference.

This is an idealized situation, when we are sure that all the sensor recordings describe a signal reflected by the object. In practice, we may have outliers – recordings which are caused by some external noise or by a reflection from a nearby object. For such outlier reflections, the corresponding set S_i describes the location of a different object; as a result, the intersection of this set with the others sets S_j (that describe reflections from the object of interest) may be empty.

One of the techniques that helps to locate an object in such situations is based on knowing the maximum possible number of outliers q. In this case, instead of taking the intersection of all n sets S_i – i.e., instead of considering the set of all the elements that belong to all n sets, we consider the set of all the elements that belong to at least $n - q$ different sets S_i. Such a "q-relaxed intersection" can also be described in terms of union and intersection: namely, it can be described as

$$S = \bigcup_{A} \bigcap_{i \in A} S_i,$$

where we consider all possible subsets $A \subseteq \{1, \ldots, n\}$ with at least $n - q$ elements.

This idea implicitly assumes that all the sensors are equally reliable. In practice, different sensors may have different reliability. This reliability can be gauged by the probability p_i that the signal coming out of the i-th sensor actually reflects the location of the desired object. Then, the probability that the signal from the i-th sensor is an outlier is equal to $1 - p_i$.

Different sensors are usually independent. So, if a location x appears as possible based on the data provided by two sensors i and j, then the probability that this location is not real – i.e., the probability that both sensors malfunctioned – is equal to the product $(1-p_i)\cdot(1-p_j)$. In general, for each location x, if $A \subseteq \{1,\ldots,n\}$ is the set of all the sensors i for which x the possible location (i.e., for which $x \in S_i$), then the probably that this is a wrong location – i.e., that all the sensors from this set malfunctioned – is equal to the product $\prod\limits_{i\in A}(1-p_i)$. It is reasonable to conclude that the location x is possible if the probability of a mistake is sufficiently small: smaller than some threshold p_0:

$$\prod_{i\in A}(1-p_i) \le p_0.$$

For computational purposes, it is convenient to replace the product with the sum by taking minus logarithm of both sides; then, this condition takes the form

$$\sum_{i\in A} w_i \ge w_0,$$

where we denoted $w_i \stackrel{\text{def}}{=} -\ln(1-p_i)$ and $w_0 \stackrel{\text{def}}{=} -\ln(p_0)$. The resulting set of possible locations then takes the following form:

$$S = \bigcup_{A:\,\sum\limits_{i\in A} w_i \ge w_0} \left(\bigcap_{i\in A} S_i\right).$$

When all the sensors are equally reliable, i.e., when all the probabilities $p_1 = \ldots = p_n$ are equal, and thus,

$$w_1 = \ldots = w_n,$$

the condition $\sum\limits_{i\in A} w_i \ge w_0$ simply means that the set A contains at least $\dfrac{w_0}{w_1}$ elements. So, for

$$q = n - \frac{w_0}{w_1},$$

this means that we consider all the subsets with at least $n-q$ elements.

The above description includes the cases when the location appears as possible based on the signals from all the sensors. In some cases, however, when we know that a certain percentage of sensors is bound to malfunction, we may want to dismiss locations that appear on too many sensors – that would probably mean that the signal is too strong and is, thus, not a reflection from the object of interest. This leads to more complex schemes.

We can consider more sophisticated combination schemes. In all these cases, the desired set A is described as an expression that combines the input sets $A_1,\ldots,A_n,\ldots$ by using the three basic set operations: union, intersection, and complement. Thus, for localization problems, it is important to be able, given sets $A_1,\ldots$, to compute the set A described by such an expression. Such computation is known as *set computation*.

Need for set intervals. In many practical situations, we know the inputs sets only approximately. For example, in practice, we only have an approximate information about the 3-D location I_1 of an underwater rock – one of the locations where an underwater object cannot be. In the ideal case, when we know the exact 3-D map of this rock, for each spatial location x, we know whether x belongs to this set or not. In practice:

- for some locations x, we know that $x \in I_1$;
- for some other locations x, we know that $x \notin I_1$; however,
- for some locations x, we do not know whether $x \in I_1$ or $x \notin I_1$.

Such a situation can be naturally described by listing two sets:

- the set $\underline{I}_1$ of all the locations x that we know are inside I_1, and
- the set $\overline{I}_1$ of all the locations that *can be* inside I_1, i.e., locations x for which we either know that $x \in I_1$, or we do not know whether $x \in I_1$ or not.

In this case, the only information that we have about the actual (unknown) set I_1 is that this set is in between $\underline{I}_1$ and $\overline{I}_1$:

$$\underline{I}_1 \subseteq I_1 \subseteq \overline{I}_1.$$

Interval and set-interval computations are indeed very useful in location and mapping problems, especially for underwater objects; see, e.g., [1], [2], [3], [4], [5], [6], [7], [8].

Resulting computational problem. The need to consider set intervals in set computations leads to the following computational problem.

We have a set-theoretic expression $A = f(A_1,\ldots,A_N)$ that expressed the desired set A as a result of a sequence of basic set-theoretic operations (union, intersection, and complement) applied to the original sets $A_1,\ldots,A_N$.

In general, we do not know the sets A_i. Instead, for each i, we know the lower set $\underline{A}_i$ and the upper set $\overline{A}_i$ for which $\underline{A}_i \subseteq \overline{A}_i$. The only information that we have about the unknown set A_i is that $\underline{A}_i \subseteq A_i \subseteq A_i \subseteq \overline{A}_i$. In other words, for each i, we know the corresponding *set interval*

$$\mathbf{A}_i = \left[\underline{A}_i, \overline{A}_i\right] \stackrel{\text{def}}{=} \{A_i : \underline{A}_i \subseteq \overline{A}_i\}.$$

For different sets $A_i \in \mathbf{A}_i$, in general, we get different sets $A = f(A_1,\ldots,A_N)$. Our objective is to find the class of all such sets A:

$$\mathcal{A} = \{f(A_1,\ldots,A_N) : A_1 \in \mathbf{A}_1,\ldots,A_N \in \mathbf{A}_N\}.$$

How this problem is solved now. It is known (see, e.g., [10]) how to compute the range for the case when the set operation $f(A_1,\ldots)$ is simply one of the three basic set operations. In this case, we have explicit formulas for the corresponding range $\mathcal{A}$:

- for the union $f(A_1,A_2) = A_1 \cup A_2$, we have
$$\mathcal{A} = \left[\underline{A}_1 \cup \underline{A}_2, \overline{A}_1 \cup \overline{A}_2\right];$$
- for the intersection $f(A_1,A_2) = A_1 \cap A_2$, we have
$$\mathcal{A} = \left[\underline{A}_1 \cap \underline{A}_2, \overline{A}_1 \cap \overline{A}_2\right];$$

- for the complement $f(A_1, A_2) = A_1 - A_2$, we have

$$\mathcal{A} = \left[\underline{A}_1 - \overline{A}_2, \overline{A}_1 - \underline{A}_2\right].$$

In general, we can:

- *parse* the expression $f(A_1, \ldots, A_n)$, i.e., represent the formula $f(A_1, \ldots, A_n)$ as a sequence of elementary set-theoretic operations, and then
- perform computations step-by-step, replacing each elementary set operation with the corresponding operation with set intervals.

One can prove, by induction, that as a result, we always get an *enclosure* $\mathbf{A}' \supseteq \mathcal{A}$ for the desired range; see, e.g., [9].

Problem: fake boundaries. It is known that while the above procedure always leads to an enclosure for the desired class $\mathcal{A}$, the resulting class $\mathbf{A}'$ is often larger than the desired class $\mathcal{A}$, i.e., contains many unneeded sets.

This can be easily illustrated on the following toy example. Let U be the universal set, and let us assume that we know nothing about the set $A_1 \subseteq U$. In this case, the range $\mathbf{A}_1$ of possible sets A_1 is simply the class of all the subsets of the universal set: $\mathbf{A}_1 = [\emptyset, U]$.

Suppose now that we want to compute the range of the function $f(A_1) = A_1 \cup (U - A_1)$. Of course, for every set A_1, the resulting set $f(A_1)$ is simply equal to the universal set, so the actual range is $\mathcal{A} = \{U\} = [U, U]$. Let us see, however, what we get if we apply the above procedure. According to the above procedure, we first represent the expression $f(A_1)$ as a sequence of elementary set-theoretical operations:

- first, we compute $A_2 \overset{\text{def}}{=} U - A_1$;
- then, we compute the union $A = A_1 \cup A_2$.

According to the above procedure, we perform these two operations with set intervals:

- first, we compute

$$\mathbf{A}_2 = U - \mathbf{A}_1 = [U, U] - \left[\underline{A}_1, \overline{A}_1\right] =$$

$$[U, U] - [\emptyset, U] = [U - U, U - \emptyset] = [\emptyset, U];$$

- then, we compute

$$\mathbf{A}' = \mathbf{A}_1 \cup \mathbf{A}_2 =$$

$$[\emptyset, U] \cup [\emptyset, U] = [\emptyset \cup \emptyset, U \cup U] = [\emptyset, U].$$

Thus, instead of the single set U, we get the class of all possible subsets of U.

We can give more realistic examples where the resulting class has unnecessary sets. For example, let us assume that we have three sets A, B, and C, and we are computing the expression

$$X = (A \cup B \cup C) \cap (A \cup B \cup (U - C)).$$

One can easily check that this expression is equivalent to $X = A \cup B$, so the actual range is equal to

$$\mathbf{X} = \left[\underline{X}, \overline{X}\,\right] = \left[\underline{A} \cup \underline{B}, \overline{A} \cup \overline{B}\,\right].$$

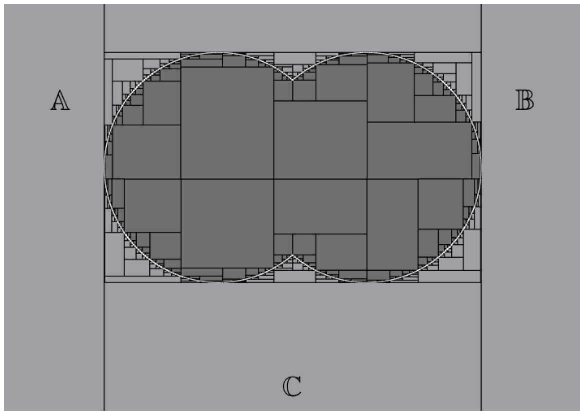


Fig. 1. Actual range $\mathbf{X}$

On the example, when all three sets are disks with uncertain boundary, the desired class $\mathbf{X}$ is shown in Fig. 1.

What will happen, however, if we apply the above algorithm to the original expression? To compute this expression:

- first, we compute the union $A_1 \overset{\text{def}}{=} A \cup B \cup C$;
- then, we compute the difference $A_2 \overset{\text{def}}{=} U - C$;
- after that, we compute the union $A_3 \overset{\text{def}}{=} A \cup B \cup A_2$; and
- finally, we compute the intersection $A = A_1 \cap A_3$.

In this case, the above algorithm leads to the following result:

- first, we compute the range of $A_1 = A \cup B \cup C$ as

$$\mathbf{A}_1 = \left[\underline{A} \cup \underline{B} \cup \underline{C}, \overline{A} \cup \overline{B} \cup \overline{C}\,\right];$$

- then, we compute the range of $A_2 = U - C$, as

$$\mathbf{A}_2 = [U, U] - \left[\underline{C}, \overline{C}\,\right] = \left[U - \overline{C}, U - \underline{C}\right];$$

- after that, we use the ranges for A, B, and A_2, to estimate the range of $A_3 = A \cup B \cup A_2$, as

$$\mathbf{A}_3 = \left[\underline{A} \cup \underline{B} \cup (U - \overline{C}), \overline{A} \cup \overline{B} \cup (U - \underline{C})\right];$$

- finally, we estimate the range of the intersection A as $\mathbf{A}' = \left[\underline{A}', \overline{A'}\right]$, where

$$\underline{A}' = (\underline{A} \cup \underline{B} \cup \underline{C}) \cap (\underline{A} \cup \underline{B} \cup (U - \overline{C})) \text{ and}$$

$$\overline{A'} = (\overline{A} \cup \overline{B} \cup \overline{C}) \cap \overline{A} \cup \overline{B} \cup (U - \underline{C})).$$

We can see that the upper bound $\overline{A'}$, in addition to the desired values $\overline{A} \cup \overline{B}$, also contains all the values from the "boundary" $\overline{C} - \underline{C}$ of the set interval C; see Fig. 2.

These fake boundaries is what we need to eliminate.

It is, in principle, possible to eliminate fake boundaries. In [9], we have proven:

- that the range $\mathcal{A}$ always has the form of a set interval $\mathcal{A} = \left[\underline{A}, \overline{A}\,\right]$ for appropriate sets $\underline{A}$ and $\overline{A}$, and
- that it is, in principle, possible to compute both sets $\underline{A}$ and $\overline{A}$.

Specifically:

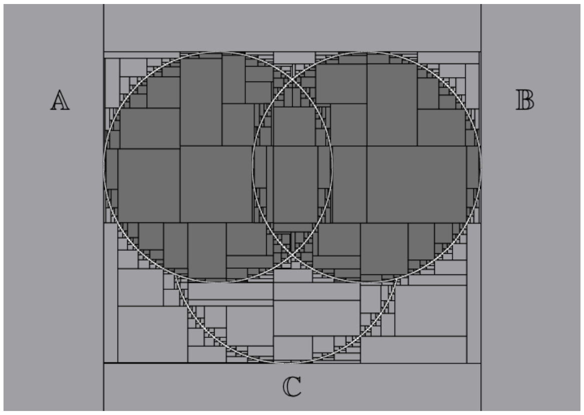


Fig. 2. An estimate $\mathbf{X}'$ with a fake boundary

- we get exactly the upper set $\overline{A}$ if we apply the above step-by-step algorithm to the equivalent canonical DNF form of the expression $f(A_1, \ldots, A_N)$, and
- we get exactly the lower set $\underline{A}$ if we apply the above step-by-step algorithm to the canonical CNF form of the expression $f(A_1, \ldots, A_N)$.

The notions of the canonical DNF and CNF forms come from propositional logic – which makes perfect sense since there is a 1-1 correspondence between set operations and propositional formulas:

- the condition $x \in A_1 \cup A_2$ means that
$$(x \in A_1) \vee (x \in A_2);$$
- the condition $x \in A_1 \cup A_2$ means that
$$(x \in A_1) \,\&\, (x \in A_2);\ \text{and}$$
- the condition that $x \in A_1 - A_2$ means that
$$(x \in A_1) \,\&\, \neg(x \in A_2).$$

By replacing union with "or", intersection with "and", etc., we can thus assign, to each set operation $f(A_1, \ldots, A_N)$, a propositional formula $F(a_1, \ldots, a_N)$ for which a point x belongs to the set $f(A_1, \ldots, A_N)$ if and only the formula $F(a_1, \ldots, a_N)$ is true for the variables a_i describing whether x belongs to A_i or not:

$$x \in f(A_1, \ldots, A_N) \Leftrightarrow F(x \in A_1, \ldots, x \in A_N).$$

For each propositional formula, we can build a canonical DNF form by enumerating all the combinations of truth variables $(a_1, \ldots, a_N)$ for which this formula is true.

For example, the above set operation

$$(A \cup B \cup C) \cap (A \cup B \cup (U - C))$$

corresponds to the propositional formula

$$(a \vee b \vee c) \,\&\, (a \vee b \vee \neg c).$$

By enumerating all possible tuples (a, b, c) for which this propositional formula is true, we can form the canonical DNF form. Specifically:

- this propositional formula is true when $a = b = c =$"true"; this leads to the term $a \,\&\, b \,\&\, c$;
- this formula is also true when $a = b =$"true" and $c =$"false"; this leads to $a \,\&\, b \,\&\, \neg c$;
- it is true when $a =$"true", $b =$"false", and $c =$"true"; this leads to $a \,\&\, \neg b \,\&\, c$;
- it is true when $a =$"true" and $b = c =$"false"; this leads to $a \,\&\, \neg b \,\&\, \neg c$;
- it is true when $a =$"false" and $b = c =$"true"; this leads to $\neg a \,\&\, b \,\&\, c$;
- finally, it is true when $a =$"false", $b =$"true", and $c =$"false"; this leads to $\neg a \,\&\, b \,\&\, \neg c$.

The formula is true if one of these cases is true. So, our formula $F(a, b, c)$ has the following canonical DNF form:

$$(a \,\&\, b \,\&\, c) \vee (a \,\&\, b \,\&\, \neg c) \vee (a \,\&\, \neg b \,\&\, c) \vee$$

$$(a \,\&\, \neg b \,\&\, \neg c) \vee (a \,\&\, b \,\&\, c) \vee (\neg a \,\&\, b \,\&\, \neg c).$$

This propositional formula corresponds to the following set function:

$$(A \cap B \cap C) \cup (A \cap B \cap (U - C)) \cup$$

$$(A \cap (U - B) \cap C) \cup (A \cap (U - B) \cap (U - C)) \cup$$

$$((U - A) \cap B \cap C) \cup ((U - A) \cap B \cap (U - C)).$$

One can easily check that if we compute $\overline{X}$ by applying the above step-by-step procedure to this formula, we get exactly the desired upper bound $\overline{X} = \overline{A} \cup \overline{B}$.

To compute the canonical CNF form, we, vice versa, list all the tuples $(a_1, \ldots, a_N)$ for which the original propositional formula is *false*. Then, we say that the propositional formula is true if the tuple is different from each of these false-producing tuples.

Let us show how this procedure works on the example of the same $(A \cup B \cup C) \cap (A \cup B \cup (U - C))$ and propositional formula $(a \vee b \vee c) \,\&\, (a \vee b \vee \neg c)$.

- This formula is false when $a = b =$"false" and $c =$"true". To avoid this tuple, we need to make sure that either a is true, or n is true, or c is false. The corresponding term is $a \vee b \vee \neg c$.
- This formula is also false when $a = b = c =$"false". To avoid this tuple, we need to make sure that either a is true, or n is true, or c is true. The corresponding term is $a \vee b \vee c$.

The formula is true for some tuple if this tuple is different from both false-inducing tuples, i.e., if

$$(a \vee b \vee \neg c) \,\&\, (a \vee b \vee c);$$

this is the canonical CNF form of the original formula. This propositional formula corresponds to the following set operation:

$$(A \cup B \cup (U - C)) \cap (A \cup B \cup C).$$

One can easily check that if we compute the lower set $\underline{X}$ by applying the above step-by-step algorithm to this set operation, we will get the exact lower set

$$\underline{X} = \underline{A} \cup \underline{B}.$$

Problem: using canonical DNF and CNF forms requires too much computation time. The main problem with the above idea is that often, it requires too many operations. For example, in the above example, the canonical DNF form requires computing 6 intersections of 3 set intervals each – and then computing the union of the resulting set intervals.

In general, when we have N sets, we can have 2^N different true-false tuples $(a_1, \ldots, a_N)$, For each of these tuples, the corresponding propositional formula is either true or false. If for a tuple, the given formula is true, then this tuple leads to a term in the canonical DNF form. If for this tuple, the given formula is false, we get a term in the canonical CNF form. To perform the above computations, we need to use both the DNF form (to compute $\overline{A}$) and the CNF form (to compute $\underline{A}$). Thus, overall, we need to compute the set interval values of 2^N terms.

In some practical situations, when we have many sensors, the number N can be huge; in this case, 2^N can be astronomically, unrealistically huge. So, a natural question is: how can we perform set interval computations faster and still eliminate all fake boundaries?

What we do in this paper. In this paper, we show how computations-without-fake-boundaries can be performed much faster.

II. How to Avoid Fake Boundaries Faster: Main Idea

Main idea. Our main idea is to use *general* DNF and CNF forms instead of the *canonical* ones.

For a propositional formula, a DNF form is a *disjunction* ("or"-combination) of *conjunctions*, i.e., "and"-combinations of propositional variables and their negations. In set-theoretic terms, a DNF form is thus a union of intersections of sets and their complements.

For example, for the above formula, $a \vee b$ is a DNF form, in which each of the conjunctions a and b consists of only one term. In this case, $A \cup B$ is the corresponding set expression.

Alternatively, we could use the following DNF expression:

$$(a \,\&\, b) \vee (a \,\&\, \neg b) \vee (\neg a \,\&\, b),$$

which corresponds to the set operation

$$(A \cap B) \cup (A \cap -B) \cup (-A \cap B),$$

where we denoted $-A \stackrel{\text{def}}{=} U - A$.

Similarly, for a propositional formula, a CNF form is a *conjunction* ("and"-combination) of *disjunctions*, i.e., "or"-combinations of propositional variables and their negations. In set-theoretic terms, a DNF form is thus an intersection of unions of sets and their complements.

For example, for the above formula, $a \vee b$ is a CNF form, with only one disjunction $a \vee b$. In this case, $A \cup B$ is the corresponding set expression.

Let us show that by using the general DNF and CNF forms, we can indeed get the same exact bounds $\underline{A}$ and $\overline{A}$ as by using the canonical DNF and CNF forms.

Proposition 1. *Let $f(A_1, \ldots, A_N)$ be a set operation in the DNF form, and let $\mathbf{A}_1, \ldots, \mathbf{A}_N$ be set intervals. Then, if we apply the above step-by-step algorithm to this data*

$$(f(A_1, \ldots, A_n), \mathbf{A}_1, \ldots, \mathbf{A}_N),$$

the resulting upper set $\overline{A'}$ will be equal to the upper set $\overline{A}$ of the corresponding range

$$\mathcal{A} = \left[\underline{A}, \overline{A}\right] = f(\mathbf{A}_1, \ldots, \mathbf{A}_N).$$

Proof.

1°. Due to the above-mentioned result from [9], the range $\mathcal{A}$ has the form of a set interval $\mathcal{A} = \left[\underline{A}, \overline{A}\right]$. Thus, the upper set $\overline{A}$ is equal to the union of all the possible sets from this range – i.e., to the union of all the sets $A = f(A_1, \ldots, A_N)$ corresponding to different combinations of sets $A_i \in \mathbf{A}_i$.

So, to prove that the set $\overline{A'}$ is equal to the desired set $\overline{A}$, it is sufficient to prove that the set $\overline{A'}$ is equal to the union of all such sets $A = f(A_1, \ldots, A_N)$.

2°. Let us first prove that the union $\overline{A}$ of all the sets $A = f(A_1, \ldots, A_N)$ is indeed contained in the resulting set $\overline{A'}$.

To prove this, we will prove that for each tuple $(A_1, \ldots, A_N)$ with $A_i \in \mathbf{A}_i$ for all i, the set $A = f(A_1, \ldots, A_N)$ is a subset of $\overline{A'}$. Indeed, the set operation $f(A_1, \ldots, A_N)$ has a DNF form

$$f(A_1, \ldots, A_N) = (A_i \cap -A_j \cap \ldots \cap A_k) \cup (\ldots) \cup \ldots$$

When we apply the above step-by-step algorithm to this form, when computing $\overline{A'}$, we replace A_i with

$$\overline{A}_i \supseteq A_i$$

and $-A_j$ with

$$-\underline{A}_j \supseteq -A_j.$$

For each term, this replacement makes it larger (or the same), so each conjunction is contained in the result of the corresponding replacement:

$$(A_i \cap -A_j \cap \ldots \cap A_k) \subseteq \left(\overline{A}_i \cap -\underline{A}_j \cap \ldots \cap \overline{A}_k\right).$$

Since this inclusion holds for each conjunction, it holds for their union as well:

$$(A_i \cap -A_j \cap \ldots \cap A_k) \cup (\ldots) \cup \ldots \subseteq$$

$$\left(\overline{A}_i \cap -\underline{A}_j \cap \ldots \cap \overline{A}_k\right) \cup (\ldots) \cup \ldots,$$

i.e., indeed, $A \subseteq \overline{A'}$.

3°. To complete the proof, we need to show that the set $\overline{A'}$ produced by the algorithm is contained in the union $\overline{A}$ of all the sets $A = f(A_1, \ldots, A_N)$ corresponding to $A_i \in \mathbf{A}_i$.

To prove this, we will show that every element $x \in \overline{A'}$ belongs to a set $A = f(A_1, \ldots, A_N)$ for appropriately chosen sets $A_i \in \mathbf{A}_i$. Indeed, let $x \in \overline{A'}$. Since the set $\overline{A'}$ is defined as a union of several conjunctions (intersections), the fact that the element x belongs to this union means that it belongs to one of these intersections, e.g., to a set of the type $\overline{A}_i \cap -\underline{A}_j \cap \ldots \cap \overline{A}_k$. This means that for the appropriate choice of the sets $A_1, \ldots$, namely, for $A_i = \overline{A}_i$, $A_j = \underline{A}_j$, …, the element x belongs to the corresponding intersection from the DNF expression $f(A_1, \ldots, A_N)$.

Since x belongs to this intersection, and the set $f(A_1, \ldots, A_N)$ is a union of several such intersections, we thus conclude that the element x belongs to the set $A = f(A_1, \ldots, A_N)$ – and hence, that x belongs to the union $\overline{A}$ of all such sets.

The proposition is proven.

Proposition 2. *Let $f(A_1, \ldots, A_N)$ be a set operation in the CNF form, and let $\mathbf{A}_1, \ldots, \mathbf{A}_N$ be set intervals. Then, if we apply the above step-by-step algorithm to this data*

$$(f(A_1, \ldots, A_n), \mathbf{A}_1, \ldots, \mathbf{A}_N),$$

the resulting lower set $\underline{A}'$ will be equal to the lower set $\underline{A}$ of the corresponding range

$$\mathcal{A} = \left[\underline{A}, \overline{A}\right] = f(\mathbf{A}_1, \ldots, \mathbf{A}_N).$$

Proof.

1°. Due to the above-mentioned result from [9], the range $\mathcal{A}$ has the form of a set interval $\mathcal{A} = \left[\underline{A}, \overline{A}\right]$. Thus, the lower set $\underline{A}$ is equal to the intersection of all the possible sets from this range – i.e., to the intersection of all the sets $A = f(A_1, \ldots, A_N)$ corresponding to different combinations of sets $A_i \in \mathbf{A}_i$.

So, to prove that the set $\underline{A}'$ is equal to the desired set $\underline{A}$, it is sufficient to prove that the set $\underline{A}'$ is equal to the intersection of all such sets $A = f(A_1, \ldots, A_N)$.

2°. Let us first prove that the intersection $\underline{A}$ of all the sets $A = f(A_1, \ldots, A_N)$ indeed contains the resulting set $\underline{A}'$.

To prove this, we will prove that for each tuple $(A_1, \ldots, A_N)$ with $A_i \in \mathbf{A}_i$ for all i, the set $A = f(A_1, \ldots, A_N)$ is a superset of $\underline{A}'$. Indeed, the set operation has a CNF form

$$f(A_1, \ldots, A_N) = (A_i \cup -A_j \cup \ldots \cup A_k) \cap (\ldots) \cap \ldots$$

When we apply the above step-by-step algorithm to this form, when computing $\underline{A}'$, we replace A_i with

$$\underline{A}_i \subseteq A_i$$

and $-A_j$ with

$$-\overline{A_j} \subseteq -A_j.$$

For each term, this replacement makes it smaller (or the same), so each disjunction contains the result of the corresponding replacement:

$$(A_i \cup -A_j \cap \ldots \cup A_k) \supseteq \left(\underline{A}_i \cup -\overline{A}_j \cup \ldots \cup \overline{A}_k\right).$$

Since this inclusion holds for each disjunction, it holds for their intersection as well:

$$(A_i \cup -A_j \cup \ldots \cup A_k) \cap (\ldots) \cap \ldots \supseteq$$
$$\left(\underline{A}_i \cup -\overline{A}_j \cup \ldots \cap \underline{A}_k\right) \cap (\ldots) \cap \ldots,$$

i.e., indeed, $A \supseteq \underline{A}'$.

3°. To complete the proof, we need to show that the set $\underline{A}'$ produced by the algorithm contains the intersection $\underline{A}$ of all the sets $A = f(A_1, \ldots, A_N)$ corresponding to $A_i \in \mathbf{A}_i$.

We will prove this by contradiction. Let us assume that for some $x \in \underline{A}$, we have $x \notin \underline{A}'$. Since the set $\underline{A}'$ is defined as an intersection of several disjunctions (unions), the fact that the element x does not belongs to this intersection means that it does not belong to one of these intersecting unions, e.g., to a set of the type $\underline{A}_i \cup -\overline{A}_j \cup \ldots \cup \underline{A}_k$. This means that for the appropriate choice of the sets $A_1, \ldots$, namely, for $A_i = \underline{A}_i$, $A_j = \overline{A}_j$, …, the element x does not belong to the corresponding union from the CNF expression $f(A_1, \ldots, A_N)$.

Since x does not belong to this union, and the set $f(A_1, \ldots, A_N)$ is an intersection of several such unions, we thus conclude that the element x does not belongs to the set $A = f(A_1, \ldots, A_N)$ – and hence, that x does not belongs to the intersection $\underline{A}$ of all such sets A. This contradicts to our assumption that x belongs to this intersection. Thus, indeed, every element $x \in \underline{A}$ belongs to $\underline{A}'$, i.e., $\underline{A} \subseteq \underline{A}'$.

The two inclusions, from Parts 2 and 3 of this proof, imply that $\underline{A} \subseteq \underline{A}'$ and $\underline{A}' \subseteq \underline{A}$. thus, $\underline{A} = \underline{A}'$. The proposition is proven.

Conclusion. Thus, to perform set interval computations and avoid fake boundaries, it is not necessary to transform the original expression into canonical DNF and CNF forms – any DNF and CNF forms will do.

The above example shows that CNF and DNF forms can indeed be much shorter than the canonical ones, so we can indeed speed up computations – without introducing fake boundaries.

III. How Do We Get Shorter DNF and CNF Forms

Main idea. How can we get shorter DNF and CNF forms? To get the canonical DNF forms, we start with all the tuples for which the corresponding propositional formula is true. For each tuple, we can then write down the corresponding conjunction.

If we have two conjunctions that differ only by one variable, i.e., which have the form $F \,\&\, v$ and $F \,\&\, \neg v$, then, we can easily see, we can replace the part $(F \,\&\, v) \vee (F \,\&\, \neg v)$ of the original DNF formula with the equivalent simpler term F.

Similarly, to get a CNF form, we start with all the tuples for which the corresponding propositional formula is false. For each tuple, we can then write down the corresponding disjunction.

If we have two disjunctions that differ only by one variable, i.e., which have the form $G \vee v$ and $G \vee \neg v$, then, we can easily see, we can replace the part $(G \vee v) \,\&\, (G \vee \neg v)$ of the original CNF formula with the equivalent simpler term G.

By applying this procedure again and again, we can get shorter and shorter expressions.

DNF example. Let us show how this idea can work on the above example. We start with the canonical DNF form

$$(a \,\&\, b \,\&\, c) \vee (a \,\&\, b \,\&\, \neg c) \vee (a \,\&\, \neg b \,\&\, c) \vee$$
$$(a \,\&\, \neg b \,\&\, \neg c) \vee (a \,\&\, b \,\&\, c) \vee (\neg a \,\&\, b \,\&\, \neg c)$$

that describes all the tuples for which the original formula $(a \vee b \vee c) \,\&\, (a \vee b \vee \neg c)$ is true.

By looking at the above formula, we immediately see the pairs of conjunctions that differ by only one variable and be thus combined together:

- the conjunctions $a \,\&\, b \,\&\, c$ and $a \,\&\, b \,\&\, \neg c$ can be combined into a single conjunction $a \,\&\, b$;
- the conjunctions $a \,\&\, \neg b \,\&\, c$ and $a \,\&\, \neg b \,\&\, \neg c$ can be combined into a single conjunction $a \,\&\, \neg b$, and
- the conjunctions $\neg a \,\&\, b \,\&\, c$ and $\neg a \,\&\, b \,\&\, \neg c$ can be combined into a single conjunction $\neg a \,\&\, b$.

After these replacements, the original DNF formula is simplified into the following form:

$$(a \,\&\, b) \vee (a \,\&\, \neg b) \vee (\neg a \,\&\, b).$$

This form can be further simplified:

- by combining $a \,\&\, b$ and $a \,\&\, \neg b$, we get a, and
- by combining $a \,\&\, b$ and $\neg a \,\&\, b$, we get b.

Thus, we get the simplified DNF form $a \vee b$.

We could reach this form differently. We could:

- combine the conjunctions $a \,\&\, b \,\&\, c$ and $a \,\&\, \neg b \,\&\, c$ into a single conjunction $a \,\&\, c$;
- combine the conjunctions $a \,\&\, b \,\&\, c$ and $a \,\&\, \neg b \,\&\, c$ into a single conjunction $b \,\&\, c$; combine the conjunctions $a \,\&\, b \,\&\, \neg c$ and $a \,\&\, \neg b \,\&\, \neg c$ into a single conjunction $a \,\&\, \neg c$;
- combine the conjunctions $a \,\&\, b \,\&\, \neg c$ and $\neg a \,\&\, b \,\&\, c$ into a single conjunction $b \,\&\, \neg c$.

Then, we would get the new DNF form

$$(a \,\&\, c) \vee (b \,\&\, c) \vee (a \,\&\, \neg c) \vee (b \,\&\, \neg c).$$

Then, new combinations are possible; we could:

- combine $a \,\&\, c$ and $a \,\&\, \neg c$ into a single conjunction a, and
- combine $b \,\&\, c$ and $b \,\&\, \neg c$ into a single conjunction b.

Thus, we will get the same short DNF form $a \vee b$.

CNF example. In the CNF case, we start with the canonical CNf form

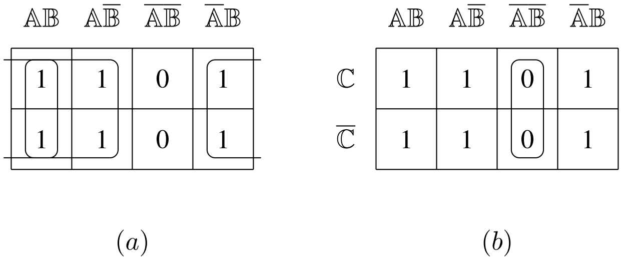
$$(a \vee b \vee c) \,\&\, (a \vee b \vee \neg c)$$

that describes all the tuples (a, b, c) for which the original propositional formula is false.

For this formula, there is only one possible combination: we can combine the disjunctions $a \vee b \vee c$ and $a \vee b \vee \neg c$ into a single disjunction $a \vee b$.

	$\mathbb{A}\mathbb{B}$	$\mathbb{A}\overline{\mathbb{B}}$	$\overline{\mathbb{A}}\overline{\mathbb{B}}$	$\overline{\mathbb{A}}\mathbb{B}$
$\mathbb{C}$	1	1	0	1
$\overline{\mathbb{C}}$	1	1	0	1

Fig. 3. Karnaugh map of the original formula



	$\mathbb{A}\mathbb{B}$	$\mathbb{A}\overline{\mathbb{B}}$	$\overline{\mathbb{A}}\overline{\mathbb{B}}$	$\overline{\mathbb{A}}\mathbb{B}$
$\mathbb{C}$	1	1	0	1
$\overline{\mathbb{C}}$	1	1	0	1

(a)

	$\mathbb{A}\mathbb{B}$	$\mathbb{A}\overline{\mathbb{B}}$	$\overline{\mathbb{A}}\overline{\mathbb{B}}$	$\overline{\mathbb{A}}\mathbb{B}$
$\mathbb{C}$	1	1	0	1
$\overline{\mathbb{C}}$	1	1	0	1

(b)

Fig. 4. Karnaugh map illustrating simplification of DNF (a) and CNF (b) forms

Karnaugh maps: a graphical representation of this idea. The above idea can be graphically represented by a *Karnaugh map*, where:

- cells corresponds to tuples,
- 1 (= "true") 0 (= "false") in a cell indicates whether the original formula is true or false for the corresponding tuple, and
- tuples differing by only variable are neighbor.

The Karnaugh-map representation of the original propositional formula is given on Fig. 3, and the above DNF and CNF reductions are illustrated on parts (a) and (b) of Fig. 4.

Acknowledgments

This work was supported in part by the National Science Foundation grants CAREER 0953339, HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation. This research was performed during Anthony Welte's visit to the University of Texas at El Paso.

The authors are thankful to all the participants of the Summer Workshop on Interval Methods SWIM'2016 (Lyon, France, June 19–22, 2016) for valuable discussions.

References

[1] Q. Brefort, L. Jaulin, M. Ceberio, and V. Kreinovich, "If we take into account that constraints are soft, then processing constraints becomes

algorithmically solvable", *Proceedings of the IEEE Symposium on Computational Intelligence for Engineering Solutions CIES'2014*, Orlando, Florida, December 9–12, 2014, pp. 1–10.

[2] Q. Brefort, L. Jaulin, M. Ceberio, and V. Kreinovich, "Towards fast and reliable localization of an underwater object: an interval approach", *Journal of Uncertain Systems*, 2015, Vol. 9, No. 2, pp. 95–102.

[3] B. Desrochers and L. Jaulin, "Computing a guaranteed approximation for the zone explored by a robot", *IEEE Transaction on Automatic Control*, 2016.

[4] B. Desrochers and L. Jaulin, "Relaxed intersection of thick sets", *Abstracts of the 17th International Symposium on Scientific Computing, Computer Arithmetic, and Verified Numerical Computation SCAN'2016*, Uppsala, Sweden, September 26–29, 2016.

[5] G. S. Franco and L. Jaulin, "Avoiding fake boundaries in interval analysis", *Abstracts of the Summer Workshop on Interval Methods SWIM'2016*, Lyon, France, June 19–22, 2016.

[6] L. Jaulin, "Range-only SLAM with occupancy maps: a set-membership approach", *IEEE Transactions on Robotics*, 2011, Vol. 27, No. 5, pp. 1004–1010.

[7] L. Jaulin, E. Walter, and O. Didrit, "Guaranteed robust nonlinear parameter bounding", *Proceedings of IMACS/IEEE-SMC CESA'96 Multi-Conference on Computational Engineering in Systems Applications CESA'96, Symposium on Modelling, Analysis and Simulation*, Lille, France, July 9–12, 1996, pp. 1156–1162.

[8] J. Sliwka, L. Jaulin, M. Ceberio, and V. Kreinovich, "Processing Interval Sensor Data in the Presence of Outliers, with Potential Applications to Localizing Underwater Robots", *Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics SMC'2011*, Anchorage, Alaska, October 9–12, 2011, pp. 2330–2337.

[9] J. T. Yao, Y. Y. Yao, V. Kreinovich, P. Pinheiro da Silva, S. A. Starks, G. Xiang, and H. T. Nguyen, "Towards more adequate representation of uncertainty: from intervals to set intervals, with the possible addition of probabilities and certainty degrees", *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'2008*, Hong Kong, China, June 1–6, 2008, pp. 983–990.

[10] Y. Y. Yao and X. Li, "Comparison of rough-set and interval-set models for uncertainty reasoning", *Fundamenta Informaticae*, 1996, Vol. 27, pp. 289–298.

Robust Data Processing in the Presence of Uncertainty and Outliers: Case of Localization Problems

Anthony Welte and Luc Jaulin
Lab STICC
École Nationale Supérieure
de Techniques Avancées Bretagne
(ENSTA Bretagne)
2 rue François Verny
29806 Brest, France
Emails: tony.welte@gmail.com,
lucjaulin@gmail.com

Martine Ceberio and Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, Texas 79968, USA
Emails: mceberio@utep.edu,
vladik@utep.edu

***Abstract*—To properly process data, we need to take into account both the measurement errors and the fact that some of the observations may be outliers. This is especially important in radar-based localization problems, where some signals may reflect not from the analyzed object, but from some nearby object. There are known methods for dealing with both measurement errors and outliers in situations in which we have full information about the corresponding probability distributions. There are also known statistics-based methods for dealing with measurement errors in situations when we only have partial information about the corresponding probabilities. In this paper, we show how these methods can be extended to situations in which we also have partial inf0ormation about the outliers (and even to situations when we have no information about the outliers). In some situations in which efficient semi-heuristic methods are known, our methodology leads to a justification of these efficient heuristics – which makes us confident that our new methods will be efficient in other situations as well.**

I. Formulation of the Problem

Need for data processing. In many practical situations, we are interested in the values of the quantities $p_1, \ldots, p_m$ which are difficult to measure directly.

For example, when solving a localization problem – whether it is a problem of locating a robot (see, e.g., [3]) or of locating a satellite (see, e.g., [4]) – we are interested in the coordinates $p_1, \ldots$ of this object. It is possible to directly measure physical quantities such as distance, velocity, density, etc. However, coordinates are an artificial construction that does not directly correspond to any physical quantity. As a result, it is not possible to directly measure coordinates of an object.

The quantities of interest do affect results of some measurements; namely, the value of the corresponding easier-to-measure quantity y depends, in a known way, on the values $p_1, \ldots, p_m$ – and on some auxiliary quantities $x_1, \ldots, x_n$ that describe the measurement's setting:

$$y = f(p_1, \ldots, p_m, x_1, \ldots, x_n).$$

For example, to determine 3-D coordinates (p_1, p_2, p_3) of an object, we can measure the distance

$$y = \sqrt{\sum_{i=1}^{3} (p_i - x_i)^2}$$

between the object of interest and another object with known coordinates (x_1, x_2, x_3).

So, to find the values of p_i, we measure the value y_k of the corresponding quantity y under different settings $(x_{k1}, \ldots, x_{kn})$, and then reconstruct the desired values p_i from the condition that

$$y_k = f(p_1, \ldots, p_m, x_{k1}, \ldots, x_{kn}) \quad (1)$$

for all the measurements $k = 1, \ldots, K$.

For example, to locate an object, we measure the distance between this object and several objects with known coordinates. This is how, e.g., radar-based systems determine the coordinates of an airplane.

Such reconstruction is an important case of *data processing*.

Need to take into account measurement uncertainty and outliers. Measurement are never absolutely accurate; see, e.g., [18]. As a result, there is always a non-zero difference between the measurement result y_k and the actual (unknown) value $f(p_1, \ldots, p_m, x_{k1}, \ldots, x_{kn})$ of the corresponding quantity:

$$\Delta y_k \overset{\text{def}}{=} y_k - f(p_1, \ldots, p_m, x_{k1}, \ldots, x_{kn}) \neq 0. \quad (2)$$

It is important to take into account this measurement uncertainty when processing data.

Measurement errors are usually reasonably small. Hence, the measured value y_k is usually close to the actual value

$f(p_1,\dots,p_m,x_{k1},\dots,x_{kn})$. However, the measuring instrument is not always 100% reliable. Sometimes, the measuring instrument malfunctions, and we get *outliers*, values which are very different from the actual values of the corresponding quantity. In processing data, we also need to take into account the existence of outliers.

This is especially important in localization problems, where the radar-type signal, instead of reflecting from the desired object, reflects from some other objects. In this case, the corresponding measurement result describes the distance to a different object – i.e., from the viewpoint of our problem, is an outlier.

What is known, what are the remaining problems, and what we do in this paper. There are many efficient techniques for taking into account measurement uncertainty. There are also techniques for taking into account outliers, and there are techniques for taking into account *both* measurement uncertainty and outliers.

Such methods work well if we have a complete knowledge about the probabilities of different values of the measurement error and the probabilities of different outliers. In practice, however, we often only have a *partial* information about these probabilities – all the way to the case when we have no information about such probabilities at all; see, e.g., [18]. In such extreme situations, there are methods that take into account either measurement uncertainty or outliers – but not both. In this paper, we briefly overview and analyze the existing techniques of taking into account measurement uncertainty and outliers, and then use this analysis to develop a natural new technique for taking into account both measurement uncertainty and outliers.

The structure of this paper is as follows. In Section 2, we describe the methods of dealing with uncertainty – beware, however that we will describe them in such a way so as to prepare us for the new technique. In Section 3, we use our analysis to show how outliers can also be taken into account.

Most of our results are new. In some cases, as a particular case of our general approach, we get a well-known effective outlier-processing technique; the fact that in some cases, we get well-known well-established efficient techniques makes us confident that our method will be efficient in other situations as well.

II. How Measurement Uncertainty Is Usually Taken Into Account

Case when we know the exact probability distribution of the measurement error. Let us first consider a situation in which we have a complete information about the probability density function $\rho(\Delta y)$ that describes the probability distribution of the measurement error. In this case, once we have the measurement results y_k ($1\le k\le K$) corresponding to settings $x_k=(x_{k1},\dots,x_{kn})$, then for each parameter tuple $p=(p_1,\dots,p_m)$ and for each k, the probability to observe y_k is proportional to $\rho(\Delta y_k)=\rho(y_k-f(p,x_k))$.

Measurement errors corresponding to different measurements are usually independent. Thus, the probability of observing all the observed values $y_1,\dots,y_K$ is equal to the product of the probabilities of observing each value y_k. Thus, this probability is proportional to the product $\prod\limits_{k=1}^K \rho(y_k-f(p,x_k))$.

In this case, we usually have different parameter tuples which are consistent with the given observations. If we need to select a single "best estimate", it is reasonable to select the parameter tuple which is the most probable, i.e., for which the product $L\stackrel{\rm def}{=}\prod\limits_{k=1}^K \rho(y_k-f(p,x_k))$ takes the largest possible value. This idea is known as the *Maximum Likelihood Method*; see, e.g., [14]. Under reasonable conditions, this method indeed leads to estimates which are optimal in some reasonable senses; see, e.g., [14], [19].

Example. Let us consider a simple example, in which the measurement error is normally distributed with 0 mean and a known standard deviation σ. In this case, the probability density function has the form

$$\rho(\Delta y)=\frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot\exp\left(-\frac{(\Delta y)^2}{2\sigma^2}\right).$$

Minimizing the corresponding product

$$L=\prod_{k=1}^K\frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot\exp\left(-\frac{(\Delta y_k)^2}{2\sigma^2}\right) \tag{2}$$

is equivalent to minimizing minus logarithm of this product

$$\psi\stackrel{\rm def}{=}-\ln(L)=K\cdot\ln(\sqrt{2\pi}\cdot\sigma)+\frac{1}{2\sigma^2}\cdot\sum_{k=1}^K(\Delta y_k)^2. \tag{3}$$

One can easily see that this minimization is equivalent to minimizing the sum

$$\sum_{k=1}^K(\Delta y_k)^2=\sum_{k=1}^K(y_k-f(p,x_k))^2.$$

This minimization – known as the *Least Squares Method* – is one of the most widely used data processing techniques.

What is we only have partial information about the probabilities: case of a finite-parametric family. In some cases, we do not know the exact probability distribution of the measurement errors, but we are aware that it belongs to a known finite-parametric family of probability distributions $\rho(\Delta y,\theta)$ depending on the parameter tuple $\theta=(\theta_1,\dots,\theta_\ell)$.

In this case, the corresponding "likelihood function" L takes the form $L=\prod\limits_{k=1}^K\rho(\Delta y_k,\theta)$. Now, instead of selecting only only the parameters p of the model, we also need to select the parameters θ of the corresponding probability distribution. In this case, it is reasonable to select the most probable pair (p,θ), i.e., the pair for which the product

$$L=\prod_{k=1}^K\rho(y_k-f(p,x_k),\theta)$$

takes the largest possible value.

Example. Let us assume that the measurement error is normally distributed with 0 mean, but this time, the standard deviation σ is unknown. In this case, we have $\ell = 1$ and $\theta_1 = \sigma$. So, we need to maximize the expression (2) – or, equivalently, minimize the expression (3) – with respect to both p and σ.

Minimizing the expression (3) with respect to parameters p leads to the same Least Squares estimate as before. Once we find p, we can differentiate the expression (3) with respect to σ, equate the derivative to 0, and get the desired expression

$$\sigma = \sqrt{\frac{1}{K} \cdot \sum_{k=1}^{K} (y_k - f(p, x_k)^2}.$$

What if we only have partial information about the probabilities: non-parametric case. In many practical situations, we do not know the finite-parametric family containing the actual distribution. For example, often, all we know is the upper bound Δ on the measurement error; see, e.g., [18]. In this case, the only information that we have about the actual probability distribution $\rho(\Delta y)$ is that this distribution is located somewhere on the interval $[-\Delta, \Delta]$.

There are many such probability distributions. To apply the Maximum Likelihood principle in this case, we need to select a single "most reasonable" distribution from all these possible distributions. Each of these distributions $\rho(\Delta y)$ can be characterized by its uncertainty (entropy)

$$S = -\int \rho(\Delta y) \cdot \ln(\rho(\Delta y))\, d\Delta y$$

that describes how many binary ("yes"-"no") questions we need to ask to uniquely determine the corresponding value Δy; see, e.g., [2], [8], [16].

Out of all possible distributions $\rho(\Delta y)$ we have distributions located on a single value v. For these distributions, we do not need any questions, we already know the value v. However, selecting such a distribution would be cheating – in actuality, we do not know the value Δy, so we would like to select the distribution that to the largest extent reflects this uncertainty. In other words, it is reasonable to select a distribution for which the entropy is the largest possible.

For all the distributions $\rho(\Delta y)$ located on the interval $[-\Delta, \Delta]$, maximum of entropy under the constraint $\int_{-\Delta}^{\Delta} \rho(\Delta y)\, d\Delta y = 1$ can be obtained by using the Lagrange multiplier method, that reduces the corresponding constraint optimization problem to the unconstrained optimization problem

$$-\int_{-\Delta}^{\Delta} \rho(\Delta y) \cdot \ln(\rho(\Delta y))\, d\Delta y +$$
$$\lambda \cdot \left(\int_{-\Delta}^{\Delta} \rho(\Delta y)\, d\Delta y - 1 \right) \to \max_{\rho(\Delta y)}$$

for an appropriate Lagrange multiplier λ. Differentiating this expression with respect to $\rho(\Delta y)$ and equating the derivative to 0, we conclude that $\rho(\Delta y) = \text{const}$, i.e., that we have a uniform distribution on the interval $[-\Delta, \Delta]$, with the probability density

$$\rho(\Delta y) = \frac{1}{2\Delta}.$$

This selection makes perfect sense: since we have no reason to believe that some values from the interval $[-\Delta, \Delta]$ are more probable than others, it is therefore reasonable to conclude that all the values from this interval are equally probable. This argument goes back to Laplace and is thus known as *Laplace Indeterminacy Principle*.

Now that we have selected a probability distribution, we can use the Maximum Likelihood method to find the corresponding parameter values p. In this case, each probability density $\rho(\Delta y_k)$ is equal to 0 if Δy_k is outside the interval $[-\Delta, \Delta]$ and to a constant (equal to $1/(2\Delta)$) when Δy_k inside this interval. Thus, the product L of the corresponding probabilities is equal to 0 if one of the values Δy_k is inside the interval, and to the same constant $\dfrac{1}{(2\Delta)^K}$ when $|\Delta y_k| \le \Delta$ for all k. So, instead of a *single* tuple p, we know need to describe *all* the tuples p for which $|y_k - f(p, x_k)| \le \Delta$ for all $k = 1, \ldots, k$.

The problem of finding the range of such tuples under *interval uncertainty* ($\Delta y_k \in [-\Delta, \Delta]$) is a particular case of *interval computations*; see, e.g., [5], [15]. In interval computations, there are many efficient techniques for solving this problem [5], [15].

What if we have no information whatsoever about the probabilities of measurement errors. In some practical situations, we have no information at all about the probability distribution $\rho(\Delta y)$ of the corresponding measurement error. This situation is somewhat similar to the previous one – with the only difference that now, we do not know the bound Δ.

How can we find a good estimate for this value Δ? A reasonable idea is to use the Maximum Likelihood method and select the value Δ for which the corresponding likelihood $L = \dfrac{1}{(2\Delta)^K}$ is the largest possible. Once can easily see that the smaller Δ, the larger this likelihood L. Thus, selecting the largest possible L is equivalent to selecting the smallest possible Δ.

The only constraints on Δ is that we should have $\Delta \ge |\Delta y_k|$ for all k. This is equivalent to having $\Delta \ge \max\limits_k |\Delta y_k|$. The smallest value satisfying this inequality is the value $\Delta = \max\limits_k |\Delta y_k|$. Thus, minimizing Δ means selecting the parameter p for which the corresponding maximum

$$\max_k |\Delta y_k| = \max_k |y_k - f(p, x_k)|$$

is the smallest possible; see, e.g., [9].

The corresponding minimax approach is indeed frequently used in data processing; see, e.g., [1], [5], [6], [11], [12], [13], [20], [21], [22], [23].

III. How to Take Outliers into Account

Which cases are possible? In the previous section, we considered possible types of knowledge about the probability distribution. In our analysis, we considered the following four cases, in the decreasing order of the available information about the probabilities:

- we know the exact distribution;
- we know the finite-parametric family of distributions;
- we know the upper bound on the (absolute value) of the corresponding difference; and
- we have no information whatsoever, not even the upper bound.

If we take outliers into account, then, in principle, we may have the same four possible types of information about the corresponding probability density function $\rho_0(\Delta y)$. At first glance, it may therefore seem that we can have $4 \times 4 = 16$ possible combinations. In reality, however, not all such combinations are possible.

Indeed, once we gather enough data, we can determine the corresponding probability distributions. Thus, the fact that we do not yet have detailed information about the probability distribution of the measurement error means that we have not yet collected a sufficient number of measurement results. In this case – since the number of outlier is usually much smaller than the number of actual measurement results – we have even fewer outliers. So, if we cannot determine the probability distribution for the measurement errors, even more so, we cannot determine the probability distribution for the outliers either. In general, for the same reason, the amount of information that we have about the outliers is smaller than the amount of information that we have about the measurement errors.

Hence, instead of 16 options, we only have options in which the amount of information about the outlier-related probability distribution $\rho_0(\Delta y)$ does not exceed the amount of information about the probabilities of measurement errors $\rho_{(}\Delta y)$. Let us consider all these cases one by one.

Full information about both distributions. Let us first consider the ideal case, when we have the complete information about the probabilities. Specifically:

- we know the probability density function $\rho(\Delta y)$ that describes the probability of different values of the measurement error, and
- we know the probability density function $\rho_0(\Delta y)$ that describes the probability of different values of the difference $\Delta y = y - f(p, x)$ corresponding to outliers y.

In this case, once we have the measurement results y_k (some of which may come from malfunctioning and are thus outliers), the probability of these observations occurring depends not only on the parameters p, but also on which of the values y_k are outliers and which are actual measurement results. Once we know the set $M \subseteq \{1, \ldots, K\}$ of indices k for which y_k is the actual measurement, we can then compute the probability L as

$$L = \left(\prod_{k \in M} \rho(\Delta y_k) \right) \cdot \left(\prod_{k \notin M} \rho_0(\Delta y_k) \right).$$

Now, we can use the Maximum Likelihood approach to determine both the parameter tuple p and the set M.

Once p is found, and thus, the values $\Delta y_k = y_k - f(p, x_k)$ are determined, maximizing the product L means:

- selecting $k \in M$ if the value $\rho(\Delta y_k)$ is larger than $\rho_0(\Delta y_k)$, and
- selecting $k \notin M$ if the value $\rho_0(\Delta y_k)$ is larger than $\rho(\Delta y_k)$.

In both cases, the resulting factor in the product L takes the form $\max(\rho(\Delta y_k)), \rho_0(\Delta y_k))$.

The resulting value L takes the following form:

$$L = \prod_{k=1}^{K} \max(\rho(\Delta y_k)), \rho_0(\Delta y_k)) =$$

$$\prod_{k=1}^{K} \max(\rho(y_k - f(p, x_k)), \rho_0(y_k - f(p, x_k))).$$

We thus need to select the parameters p for which this product attains the largest possible value.

Comment. From the computational viewpoint, the corresponding problem is similar to the usual maximum likelihood problem, with a new function $g(\Delta y) \stackrel{\text{def}}{=} \max(\rho(\Delta y), \rho_0(\Delta y))$ instead of the original probability density function $\rho(\Delta y)$. It is worth mentioning, however, that, in contrast to the probability density function $\rho(\Delta y)$ for which $\int \rho(\Delta y)\, dy = 1$, for the new function $g(\Delta y)$, we have, in general,

$$\int g(\Delta y)\, dy > \int \rho(\Delta y)\, dy = 1$$

(as long as the probability densities $\rho(\Delta y)$ and $\rho_0(\Delta y)$ are different).

Full information about $\rho(\Delta y)$, finite-parametric family for $\rho_0(\Delta y)$. In this case, instead of single distribution $\rho_0(\Delta y)$, we have a finite-parametric family of distributions $\rho_0(\Delta y, \varphi)$ with unknown parameters φ. In such a situation, we need to determine all the parameters p and φ from the requirement that the likelihood

$$L = \prod_{k=1}^{K} \max(\rho(\Delta y_k), \rho_0(\Delta y_k, \varphi)) =$$

$$\prod_{k=1}^{K} \max(\rho(y_k - f(p, x_k)), \rho_0(y_k - f(p, x_k), \varphi))$$

attains the largest possible value.

Full information about $\rho(\Delta y)$, bound W on the outlier-related differences Δy_k. In this case, based on the maximum entropy approach, as a distribution $\rho_0(\Delta y)$, we select a uniform distribution on the interval $[-W, W]$, with the probability density $\rho_0(\Delta y_k) = \dfrac{1}{2W}$.

In such a situation, we determine the parameters p from the requirement that the likelihood

$$L = \prod_{k=1}^{K} \max\left(\rho(\Delta y_k), \frac{1}{2W}\right) =$$

$$\prod_{k=1}^{K} \max\left(\rho(y_k - f(p, x_k)), \frac{1}{2W}\right)$$

attains the largest possible value under the constraint that

$$|\Delta y_k| = |y_k - f(p, x_k)| \le W$$

for all $k = 1, \ldots, K$.

Full information about $\rho(\Delta y)$, no information whatsoever about the outlier-related differences Δy_k. In this case, we select the value W for which the likelihood L as described in the previous example if the largest possible – under the constraint that $|\Delta y_k| \le W$ for all k.

One can easily see that the smaller the bound W, the larger the density $\frac{1}{2W}$ and thus, the larger the likelihood function. Thus, to determine the largest possible value of the likelihood function L, we must select the smallest possible value W. The constraints on W have the form that $W \ge |\Delta y_k|$ for all k. The smallest possible value W that satisfies all these constraints is the value

$$W = \max_{\ell} |\Delta y_\ell| = \max_{\ell} |y_\ell - f(p, x_\ell)|.$$

Substituting this expression into the above formula, we conclude that we need to select the parameters p for which the likelihood

$$L = \prod_{k=1}^{K} \max\left(\rho(y_k - f(p, x_k), \frac{1}{2 \cdot \max\limits_{\ell} |y_\ell - f(p, x_\ell)|}\right)$$

attains the largest possible value.

Finite-parametric information about $\rho(\Delta y)$ and about $\rho_0(\Delta)$. In this case, instead of single distributions $\rho(\Delta y)$ and $\rho_0(\Delta y)$, we have finite-parametric families of distributions $\rho(\Delta y, \theta)$ and $\rho_0(\Delta y, \varphi)$ with unknown parameters θ and φ. In such a situation, we need to determine all the parameters p, θ, and φ from the requirement that the likelihood

$$L = \prod_{k=1}^{K} \max(\rho(\Delta y_k, \theta)), \rho_0(\Delta y_k, \varphi)) =$$

$$\prod_{k=1}^{K} \max(\rho(y_k - f(p, x_k), \theta), \rho_0(y_k - f(p, x_k), \varphi))$$

attains the largest possible value.

Finite-parametric information about $\rho(\Delta y)$, bound W on the outlier-related differences Δy_k. In such a situation, we determine the parameters p and θ from the requirement that the likelihood

$$L = \prod_{k=1}^{K} \max\left(\rho(\Delta y_k, \theta), \frac{1}{2W}\right) =$$

$$\prod_{k=1}^{K} \max\left(\rho(y_k - f(p, x_k), \theta), \frac{1}{2W}\right)$$

attains the largest possible value under the constraint that

$$|\Delta y_k| = |y_k - f(p, x_k)| \le W$$

for all $k = 1, \ldots, K$.

Finite-parametric information about $\rho(\Delta y)$, no information about the outlier-related differences Δy_k. In this case, similarly to the above case when we had no information about the outlier-related differences Δy_k, we should select the smallest possible W, i.e., $W = \max\limits_{\ell} |\Delta y_\ell|$. Thus, we need to select the parameters p and θ for which the likelihood

$$L = \prod_{k=1}^{K} \max\left(\rho(y_k - f(p, x_k), \theta), \frac{1}{2 \cdot \max\limits_{\ell} |y_\ell - f(p, x_\ell)|}\right)$$

attains the largest possible value.

Bound Δ on the measurement errors, bound W on the outlier-related differences Δy_k. In this case, by using the maximum entropy approach, we select the following distributions:

- the measurement errors are uniformly distributed on the interval $[-\Delta, \Delta]$, with the probability density
$$\rho(\Delta y) = \frac{1}{2\Delta};$$
- the outlier-related differences Δy_k are uniformly distributed on the interval $[-W, W]$, with the probability density $\rho_0(\Delta y) = \frac{1}{2W}$.

In this case, we need to select the parameters p that maximize the likelihood $L = \prod\limits_{k=1}^{K} g(\Delta y)$, where

$$g(\Delta y) = \max(\rho(\Delta y), \rho_0(\Delta y)).$$

For the above uniform distributions, the auxiliary function $g(\Delta y)$ takes the following form:

- for the values Δy for which $|\Delta y| \le \Delta$, we have
$$g(\Delta y) = \frac{1}{2\Delta};$$
- for the values Δy for which $\Delta < |\Delta y| \le W$, we have $g(\Delta y) = \frac{1}{2W}$; and
- for the values Δy for which $|\Delta y| > W$, we have $g(\Delta y) = 0$.

Thus, maximizing the product $L = \prod\limits_{k=1} g(\Delta y_k)$ means minimizing the number of outliers under the constraint that $|\Delta y_k| = |y_k - f(p, x_k)| \le W$ for all k. In other words, we select p for which, under the above constraints, the number of observations for which $|y_k - f(p, x_k)| > \Delta$ is the smallest possible.

Bound Δ on the measurement errors, no information about the outlier-related differences Δy_k. In this case, since we take $W = \max\limits_{\ell} |y_\ell - f(p, x_\ell)|$, there are no longer any limitations on p.

Thus, in this case, the maximum likelihood method simply means selecting the values of the parameters p for which the number of outliers (i.e., values for which $|y_k - f(p, x_k)| > \Delta$) is the smallest possible.

Comment. This idea has been actively used, as a heuristic idea, to deal with data processing under outliers, see, e.g., [3], [7], [10]. Several practical applications of this heuristic idea are described, e.g., in [3].

Our probability-based justification for this heuristics was first announces in [17] (see also [4]).

Final case, when we have no information about the probabilities. Finally, let us consider the case when we have no information about the probabilities, neither about the probabilities of different values of the measurement errors, nor about the probabilities of different outlier-related differences

$$\Delta y = y - f(p, x).$$

In this case, we need to select the corresponding bounds Δ and W for which the corresponding likelihood function attains its largest possible value. Similar to the previous cases, for each parameter tuple p, the maximum of the likelihood L is attained if we take $W(p) = \max_\ell |\Delta y_\ell|$, so it only remains to select p and Δ.

For each p and Δ, let us denote by $n(p, \Delta)$ the number of values k for which $|y_k - f(p, x_k)| \le \Delta$. In terms of this notation, the desired likelihood value

$$L(p, \Delta) = \prod_{k=1}^{K} g(y_k - f(p, x_k))$$

has the form

$$L(p, \Delta) = \frac{1}{(2\Delta)^{n(p,\Delta)}} \cdot \frac{1}{(2W(p))^{K - n(p,\Delta)}},$$

i.e., equivalently, the form

$$L(p, \Delta) = \frac{1}{(2W(p))^K} \cdot \left(\frac{W(p)}{\Delta}\right)^{n(p,\Delta)}.$$

Maximizing this expression is equivalent to minimizing its minus logarithm

$$\psi(p, \Delta) = -\ln(L(p, \Delta)) =$$

$$K \cdot \ln(2W(p)) + n(p, \Delta) \cdot (\ln(\Delta) - \ln(W(p))).$$

Thus, to get the maximum likelihood, for each p, we need to select Δ for which the expression $\psi(p, \Delta)$ is the smallest possible. We then select the parameters for which the resulting minimum is the smallest possible, i.e., for which the following expression is the smallest possible:

$$\psi(p) = \min_{\Delta}(K \cdot \ln(2W(p)) + n(p, \Delta) \cdot (\ln(\Delta) - \ln(W(p)))),$$

where $W(p) = \max_{\ell} |y_\ell - f(p, x_\ell)|$ and

$$n(p, \Delta) = \#\{k : |y_k - f(p, x_k)| \le \Delta\}.$$

Comment. To check how well our method works, we have applied this idea to the situations when the values Δy_k are distributed according to several reasonable distributions: normal, heavy-tailed power law, etc.

In all these cases, we get 5-20% values classified as outliers. This is in line with the usual case of normal distribution, where 5% of the values lie outside the 2σ interval and are, thus, usually dismissed as outliers,

Acknowledgments

This work was supported in part by the National Science Foundation grants CAREER 0953339, HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation. This research was performed during Anthony Welte's visit to the University of Texas at El Paso.

The authors are thankful to all the participants of the 7th International Workshop on Reliable Engineering Computing REC'2016 (Bochum, Germany, June 15–17, 2016) for valuable discussions.

References

[1] J. Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer Verlag, New York, 1985.

[2] B. Chokr and V. Kreinovich, "Ho far are we from complete knowledge: complexity of knowledge acquisition in Demster-Shafer approach", In: R. R. Yager, J. Kacprzyk, and M. Pedrizzi (eds.), *Advances in the Dempster-Shafer Theory of Evidence*, Wiley, New York, 1994, pp. 555–576.

[3] B. Desrochers, S. Lacroix, and L. Jailin, "Set-membership approach to the kidnapped robot problem", *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'2015*, Hamburg, Germany, September 28 – October 2, 2015, pp. 3715–3720.

[4] V. Drevelle and P. Bonnifait, "A set-membership approach for high integrity height-added satellite positioning", *GPS Solutions*, 2011, Vol. 15, No. 4, pp. 357–368.

[5] L. Jaulin, "Reliable minimax parameter estimation", *Reliable Computing*, 2001, Vol. 7, No. 3, pp. 231–246.

[6] L. Jaulin, M. Kiefer, O. Dicrit, and E. Walter, *Applied Interval Analysis*, Springer, London, 2001.

[7] L. Jaulin and E. Walter, "Guaranteed robust nonlinear mimimax estimation", *IEEE Transactions on Automatic Control*, 2002, Vol. 47, No. 11, pp. 1857–1864.

[8] E. T. Jaynes and G. L. Bretthorst, *Probability Theory: The Logic of Science*, Cambridge University Press, Cambridge, UK, 2003.

[9] V. Kreinovich and S. Shary, "Interval Methods for Data Fitting under Uncertainty: A Probabilistic Treatment", *Reliable Computing*, 2016, Vol. 23, pp. 105–141.

[10] H. Lahanier, E. Walter, and R. Gomeni, "OMNE: a new robust membership-set estimator for the parameters of nonlinear modles", *Journal of Pharmacokinetics and Biopharmacutics*, 1987, Vol. 15, No. 2, pp. 203–219.

[11] El L. Lehmann and G. Casella, *Theory of Point Estimation*, Springer, New York, 2003.

[12] R. McKendall, *Minimax Estimation of a Discrete Location Parameter for a Continuous Distribution*, PhD Dissertation, Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, 1990; available as Teachnical Report MS-CIS-90-28, Computer and Information Science Department, University of Pennsylvania, 1990.

[13] R. McKendall and M. Mintz, "Robust sensor fusion with statistical decision theory", In: M. A. Abidi ad R. C. Gonzalez (eds.), *Data Fusion in Robotics and Machine Intelligence*, Academic Press, Boston, Massachusetts, 1992, pp. 211–244.

[14] R. B. Millar, *Maximum Likelihood Estimation and Inference*, Wiley, Chichester, UK. 2011.

[15] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[16] H. T. Nguyen, V. Kerinovich, B. Wu, and G. Xiang, *Computring Statistics under Intervala nd Fuzzy Uncertainty*, Springer Verlag, Berlin, 2012.

[17] J. Nicola and L. Jaulin, “OMNE is a Maximum Likelihood estimator”, *Abstracts of the 7th International Workshop on Reliable Engineering Computing REC’2016*, Bochum, Germany, June 15–17, 2016.

[18] S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, Berlin, 2005.

[19] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.

[20] A. B. Tsybakov, *Introduction to Nonparametric Estimation*, Springer, New York, 2009.

[21] E. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*, Springer, London, 1997.

[22] G. A. Watson, “The minimax solution of an overdetermined system of nonlinear equations”, *IMA Journal of Applied Mathematics*, 1979, Vol. 23, No. 2, pp. 167–180.

[23] M. A. Wolfe, “On discrete mimimax problems in R using intervak arithmetic”, *Reliable Computing*, 1999, Vol. 5, No. 4, pp. 371–383.