An Advanced Skyline Approach for Imperfect Data Exploitation and Analysis

Saïda Elmi

THESE

Pour l'obtention du Grade de

DOCTEUR DE L'ÉCOLE NATIONALE SUPÉRIEURE DE MÉCANIQUE ET D'AEROTECHNIQUE DE POITIERS et DE L'INSTITUT SUPÉRIEUR DE GESTION De TUNIS

(Diplôme National- Arrêté du 25 mai 2016)

Ecole Doctorale:

Ecole Doctorale: Sciences et Ingénierie pour l'Information et Mathématiques
Secteur de Recherche : INFORMATIQUE ET APPLICATIONS

Présentée par:
SAIDA ELMI

An Advanced Skyline Approach for Imperfect Data Exploitation and Analysis

---

Directeurs de Thèse: **Allel HADJALI** et **Boutheina BEN YAGHLANE**

Soutenue le 15 Septembre 2017, devant le jury composé de:

Présidente :	Prof. Anne DOUCET	Université Pierre et Marie Curie, France
Rapporteurs :	Prof. Arnaud Martin	Université Rennes 1, France
	Prof. Faiz Gargouri	Université de Sfax, Tunisie
Membres du jury :	Prof. Nahla BEN AMOR	Université de Tunis, Tunisie
	Prof. Allel Hadjali	ISAE-ENSMA, France
	Prof. Boutheina Ben Yaghlane	Université de Carthage, Tunisie

Acknowledgements

It would not have been possible to write this dissertation without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here. A special word of thank and gratitude to all, therefore the following list is by no means exhausting.

First of all, I would like to express my gratitude to my supervisor Professor Allel HADJALI who introduced me to the world of uncertainty in Artificial Intelligence and guided me in my research. Special thank to him for his generous support and encouragement. This thesis could have not been completed without his kind-hearted support. I am also very indebted to him for accepting me to work at LIAS Laboratory (Ecole Nationale de Mécanique et d'Aerotechnique, Université de Poitiers), a very stimulating environment, where my thesis was achieved.

I am deeply grateful to my supervisor Professor Boutheina BEN YAGHLAtorNE for her continuous support. Her consistent encouragements, her advice and cooperation are very helpful. It has been a great fortune for me to work under the supervision of Prof. Allel Hadjali and Prof. Boutheina Ben Yaghlane.

This work has been funded in part by ISG-Tunis (Tunisia) and by the laboratory LIAS (France) through the cooperation project France/Tunisia. I am very grateful to LARODEC Laboratory (ISG - Tunis) for providing funding during my visits at LIAS.

I am deeply grateful to Dr. Mohamed Anis Bach Tobji for his continuous support, for his good cooperation, and for the interesting and critical discussions. Not only he guided my work, he has also invested a lot of his time to help me understanding several notions in this field of study.

I wish to thank Professor Didier Dubois from University of Toulouse (Université Paul-Sabatier, France) whose comments and support during the journal preparation were indispensable.

I am very thankful to Dr. Karim Benouaret. I thank him for the knowledge and skills he imparted through our collaboration.

Finally, i would also like to thank my parents, my mother in law, my husband, my sister and brothers for their continuous moral support and encouragement with their best wishes. Their love accompanies me wherever I go.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context and Motivation

The present decade has seen a revival of interest in preference queries that aim at retrieving not necessarily all answers to queries but rather the best, most preferred answers. Skyline queries introduced by (Borzsonyi, Kossmann, & Stocker, 2001), are a popular example of preference queries. They rely on Pareto dominance relationship and they have been shown to be a powerful means in multi-criteria decision-making. The skyline comprises the objects that are not dominated (in Pareto sense) by any other object. Given a set of database objects, defined on a set of attributes, an object U is said to dominate (in Pareto sense) another object V if and only if U is better than or equal to V in all attributes and better in at least one attribute.

On the other hand, due to the exploding number of information stored and shared over Internet, and the introduction of new technologies to capture and transit data, uncertain data analysis is an important issue in many modern real-life applications such as decision-making, data integration, object identification (Bohm, Pryakhin, & Schubert, 2006), moving objects tracking (Cheng, Kalashnikov, & Prabhakar, 2004; Chen, Özsu, & Oria, 2005) and data cleaning (Fuxman, Fazli, & Miller, 2005). Uncertain data in those applications are generally caused by factors like data randomness and incompleteness, limitations of measuring equipment and delay or loss in data transfer. For example, in sensor networks, collected data often contain noise due to environmental factors, device failure or multiple sources of errors.

To deal with imperfect[1] values of database attributes, several models were proposed. The most studied and known models are: probabilistic databases (N. Dalvi & Suciu, 2007; N. N. Dalvi & Suciu, 2007; Aggarwal & Yu, 2009), possibilistic databases (Bosc & Pivert, 2005, 2010) and evidential databases (based on Dempster-shafer theory) (Lee, 1992a; Ee, Srivastava, & Shekhar, 1994, 1996; Bell, Guan, & Lee, 1996; Bach Tobji, Ben Yaghlane, & Mellouli, 2008; Bousnina et al., 2016). The advantage of the evidential database model is twofold: (i) it allows modeling both uncertainty and imprecision (due to the lack of information) in data; and (ii) it represents a generalization of both probabilistic and possibilistic models.

Substantial research work has addressed the problem of skyline analysis on uncertain data from different perspectives and within various communities, including, databases; e.g., (Pei, Jiang, Lin, & Yuan, 2007; Jiang, Pei, Lin, & Yuan, 2012; Lian & Chen, 2008; W. Zhang et al., 2013; Bosc, Hadjali, & Pivert, 2011), Web services; e.g., (Yu & Bouguettaya, 2010; Benouaret, Benslimane, & Hadjali, 2012), and so on. These works are important

[1]Imperfect values stands for uncertain/imprecise values in our study.

and useful, but they focus on either probabilistic data or possibilistic data. However, as mentioned above, these models (i.e., probabilistic and possibilistic) have some limitations to handle some imperfections like imprecision.

As an example, consider plane sensor analysis: to be aware of plane environment in the ordinary situation or to get more information about targets in a state of war, it is absolutely necessary to analyze sensor data to know which target is the most dangerous. To analyze plane sensor observations, we can use some technical criteria (e.g., distance from the target and its altitude). Ideally, we want to find the optimum plane sensor observation that returns, among n targets, the most dangerous in all aspects (i.e., with less distance and altitude). While such target does not exist, the skyline discloses the trade-off between the different preferences. The skyline includes each target U such that there is any other target V that is more dangerous. In other words, a target U is in the skyline if there exists no other target V that dominates U in distance and altitude dimensions.
We argue that skyline analysis is also meaningful on uncertain data. Consider the skyline analysis in sensor observations again, the initial impulse emitted by the sensor loses energy outward journey, on impact with the object and in its return path. Add to that, the climatic conditions and incompleteness in data, all these reasons make sensor observations frequently imperfect. In order to access to the most accurate and reliable information, we need to make many observations, well represent these observations and manage the lack of data. Such imperfection in data can be well modeled by the evidence theory and such imperfect data can be well managed in evidential databases. If observation-by-observation data are considered, we can answer many questions such as: which target should be in the skyline and among these skyline targets, which targets are considered the most dangerous. In other words, which objects in the skyline are the best w.r.t. the skyline criteria.

The main purpose of this thesis is to study an advanced database tool named the skyline operator in the context of imperfect data modeled by the evidence theory. This thesis addresses, on the one hand, the fundamental question of how to extend the dominance relationship to evidential data and to define the semantics of the evidential skyline, and on the other hand, the issue of the optimization techniques for improving the computation of the evidential skyline.

1.2 Thesis Contributions

In this thesis, based on the material presented in (Elmi, Benouaret, Hadjali, Bach Tobji, & Ben Yaghlane, 2014), the following new contributions are made:

- Evidential skyline operator: We tackle the problem of skyline analysis over imperfect data modeled by the evidence theory. Specifically, we address two main challenges. The first is about modeling skyline on evidential data: how can we capture the dominance relationship between the objects of an evidential database? And what should be the skyline on those objects? The second is about computing this kind of skyline: can we provide efficient techniques for computing the skyline on evidential data?

- Distributed evidential skyline: We first define the local evidential skyline. We then introduce an efficient approach for querying and processing the evidential skyline over multiple and distributed servers. This approach is mainly based on the notions of: (i) increasing the parallelism to improve the efficiency of the distributed evidential skyline computation and (ii) marginal points to resume the dominance regions of each local evidential skyline. We also propose the distributed architecture scheme of our distributed skyline processing and we develop efficient algorithms to compute the global evidential skyline. We conduct extensive experiments to show the interest of our approach.

- Evidential skyline maintenance: We propose efficient methods to maintain the skyline results in the evidential database context when a set of objects is inserted or deleted. The idea is to incrementally compute the new skyline, without reconducting an initial operation from the scratch. In addition, we perform an extensive experimental evaluation to demonstrate the scalability of the algorithms proposed.

- Top-k evidential skyline queries: Based on the evidential dominance relationship, we propose a score function reflecting the dominance degree of each object. This score function aims at retrieving the k objects that are expected to believably dominate the more the other objects. In a second step, we develop efficient algorithms to the evidential skyline computation and the top-k query. We conduct extensive experiments to show the efficiency and the effectiveness of our approach. In addition, our extensive experiments reflect the impact of the confidence level on the top-k skyline results.

- Evidential skyline stars: Since the evidential skyline size is often too large to be analyzed, we define the set SKY^2 to refine the evidential skyline and retrieve the best evidential skyline objects (or the stars). In addition, we develop suitable algorithms based on scalable techniques to efficiently compute the evidential SKY^2. We also perform an extensive experimental evaluation to demonstrate the scalability of the proposed algorithms.

1.3 Thesis Structure

The remainder of this thesis is organized as follows:

In Chapter 2, we present the theory of belief functions in a first part. The main objective of this theory is to represent and manage degrees of belief about propositions of a given problem. It is usually regarded as a generalization of the Bayesian approach, but has several interpretations. Then, we briefly describe the evidential databases. In a second part, we present the basic concepts of the skyline operator over certain data as well as uncertain data.

In chapter 3, first we tackle the problem of skyline analysis on evidential databases. We first introduce a skyline model that is appropriate to the evidential data nature. We then develop an efficient algorithm to compute this kind of skyline. Finally, we present a thorough experimental evaluation of our approach.

In chapter 4, we consider the evidential skyline in both centralized and distributed environments. first we define in section 4.2, the notion of marginal points used in the next sections. In section 4.3, as large amounts of distributed data over Internet are communicated and shared, we propose to efficiently compute the global skyline from distributed local sites, using the idea of skyline query over centralized evidential data. The efficiency and effectiveness of our proposal are verified by extensive experimental results. In section 4.4, we address the problem of the skyline objects maintenance of frequently updated evidential databases. In particular, we propose algorithms for maintaining evidential skyline in the case of object insertion or object deletion. Extensive experiments are conducted to demonstrate the efficiency and scalability of the approaches proposed.

In chapter 5, since previous researches showed that the skyline size over uncertain data is too large, we propose to rank the evidential skyline results and retrieve the k skyline objects that are expected to have the highest score with considering the confidence level of the objects. We also study its impact on the top-k result. The efficiency and effectiveness of our proposal are verified by a set of experiments.

In chapter 6, we particularly tackle an important issue, namely the skyline stars (denoted by SKY^2) over the evidential data. This kind of skyline aims at retrieving the best evidential skyline objects (or the stars). Efficient algorithms have been developed to compute the SKY^2. Extensive experiments have demonstrated the efficiency and effectiveness of our proposed approaches that considerably refine the huge skyline.

Finally, the last chapter gives a summary of the results achieved in this thesis. We also provide some research lines for future work.

Chapter 2

Background Material

Contents

2.1 Introduction

The necessary background material will be presented in this chapter. We first recall the basic notions about the evidential databases in section 2.2, then a reminder about the skyline operator will be provided and detailed in section 2.3.

2.2 Evidential Databases

We first start by presenting the basic concepts about the evidence theory.

2.2.1 Evidence theory

Shafer (Shafer, 1976) introduced the mathematical theory of evidence which is a subjective evaluation used to characterize the truth of a proposition. The theory of evidence (also called the belief functions theory) is a generalization of the Bayesian theory of subjective probabilities (Shafer, 1976; Dempster, 1968).

Let Θ be a finite and exhaustive set whose elements are mutually exclusive, Θ is called a frame of discernment (in practice, Θ stands for a set of possible alternatives or propositions). A basic belief assignment (*bba*), also called a mass function, is a mapping $m : 2^{\Theta} \longrightarrow [0, 1]$ such that

$$\begin{aligned} m(\emptyset) &= 0 \\ \sum_{A \subseteq \Theta} m(A) &= 1 \end{aligned} \tag{2.1}$$

An element A of 2^{Θ} is called a focal element whenever $m(A) > 0$. The mass m(A) represents the level of credibility allocated to the subset A. The truth of A is measured thanks to two functions:
The belief of A, denoted by $bel(A)$, is defined as the sum of the masses assigned to every subset B of A, i.e.,

$$bel(A) = \sum_{B \subseteq A} m(B) \tag{2.2}$$

The plausibility of A, denoted by $pl(A)$, is defined as the sum of the masses assigned

to every subset B of Θ that intersects A, i.e.,

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \tag{2.3}$$

One can observe that $bel(A)$ reflects the total weight of evidence in A, while $pl(A)$ reflects the total weight of evidence which is not committed to $\Theta - A$ called the complement of A. It is necessary to note that the belief and plausibility functions are dual functions, i.e., $bel(A) = 1 - pl(\bar{A})$.

Example 2.1. *Let us consider a classification problem of air targets, one can have as frame of discernment:* $\Theta = \{Airliner, Helicopter, Missile\}$. *Then,* $2^{\Theta} = \{\emptyset, A, H, M, A \cup H, A \cup M, H \cup M, \Theta\}$ *where* $A = Airliner$, $H = Helicopter$ *and* $M = Missile$.
The identification of a target can then be expressed by a mass function. Assume that a sensor is available to signal the presence of a fairly quick target. It produces this bba: $m(Airliner) = 0.6$, $m(Airliner \cup Missile) = 0.2$, $m(\Theta) = 0.2$
Note that $m(\Theta)$ *represents the mass allocated to all propositions in* Θ *which reflects the partial ignorance if* $0 < m(\Theta) < 1$ *else the total ignorance* ($m(\Theta) = 1$).
One can easily check that: $bel(Airliner) = m(Airliner) = 0.6$,
$bel(Missile) = bel(Helicopter) = 0$,
$bel(Airliner \cup Missile) = m(Airliner \cup Missile) + m(Airliner) = 0.8$,
$bel(Airliner \cup Helicopter) = m(Airliner) = 0.6$,
$bel(Missile \cup Helicopter) = 0$.
Let us now compute the degree of plausibility of $A \cup H$. *It is easy to see that* $pl(A \cup H) = m(A) + m(H) + m(A \cup H) + m(\Theta) = 0.8$.

As shown in this example, plausibility function measures the intensity with which we do not doubt the proposition $(Airliner \cup Helicopter)$, that is to say intensity with which $(Airliner \cup Helicopter)$ is plausible.

In probability theory, one can easily compute the probabilities of comparisons of two independent probability distributions. For example, we can work out the probability that one of the associated random variables is less than or equal to the other. But in standard Dempster-Shafer theory, the definitions of the bel and pl functions do not handle comparisons like this. When we use mass functions to represent uncertain and imprecise information instead of probability distributions, it is necessary to extend the definition of bel and pl functions to handle the comparison of two independent basic probability assignments.

In (Bell et al., 1996), authors proposed to extend the definition of *bel* and *pl* functions to deal with comparison of two independent *bba*s. This concept is also used in (Y. Zhang,

Jun, Wei, & Wu, 2010; Y. Zhang, Wu, Wei, & Wang, 2011).
Let X and Y be two *bba* which are independent and let $m_X, m_Y : 2^\Theta \to [0,1]$, be their two mass functions, respectively. Then

Definition 2.1. *(Inequality relations) For $A, B \subseteq \Theta$, we have:*

$$bel(X \leq Y) = \sum_{A \subseteq \Theta} (m_X(A) \sum_{B \subseteq \Theta, A \leq^\forall B} m_Y(B)). \tag{2.4}$$

where $A \leq^\forall B$ means that $\forall a \in A$, $\forall b \in B$, such that $a \leq b$. We have also:

$$bel(X < Y) = \sum_{A \subseteq \Theta} (m_X(A) \sum_{B \subseteq \Theta, A <^\forall B} m_Y(B)). \tag{2.5}$$

where $A <^\forall B$ means that $a < b$ for all $a \in A$ and $b \in B$.

$$bel(X = Y) = \sum_{|A|=1} m_X(A) m_Y(A) = \sum_{a \in \Theta} m_X(\{a\}) m_Y(\{a\}). \tag{2.6}$$

We have also:

$$pl(X \leq Y) = \sum_{A \subseteq \Theta} (m_X(A) \sum_{B \subseteq \Theta, A \leq^\exists B} m_Y(B)). \tag{2.7}$$

where $A \leq^\exists B$ means for every $a \in A$, there exists $b \in B$ such that $a \leq b$. Also, we have:

$$pl(X < Y) = \sum_{A \subseteq \Theta} (m_X(A) \sum_{B \subseteq \Theta, A <^\exists B} m_Y(B)). \tag{2.8}$$

where $A <^\exists B$ means for every $a \in A$, there exists $b \in B$ such that $a < b$.

Example 2.2. *Assume that t_1 and t_2 are two target observations defined on the attribute Distance. Assume also that we have two bbas [1] $t_1.d$ and $t_2.d$ defined such that $t_1.d = \langle\{100\}, 1.0\rangle$ and $t_2.d = \langle\{100, 120\}, 0.7, \{120\}, 0.3\rangle$.*
One can check that $bel(t_1.d \leq t_2.d) = 1$, $bel(t_1.d < t_2.d) = 0.3$, $pl(t_1.d \leq t_2.d) = 1$ and $pl(t_1.d < t_2.d) = 1$.

As we can observe, the plausible function returns an optimistic degree. That is because intersections between bba are considered.

With appropriate definitions, we can easily prove that the definitions of bel(X∇Y) and pl(X∇Y), where $\nabla \in (=, \neq, <, \leq)$, satisfy the following properties:

[1] In the following, and for short, we use d to denote the distance from target.

- $0 \leq \text{bel}(\text{X} \nabla \text{Y}) \leq \text{pl}(\text{X} \nabla \text{Y}) \leq 1$;
- $\text{bel}(\text{X} \nabla \text{Y}) + \text{bel}(\neg(\text{X} \nabla \text{Y})) \leq 1$;
- $\text{pl}(\text{X} \nabla \text{Y}) + \text{pl}(\neg(\text{X} \nabla \text{Y})) \geq 1$;
- If all the focal elements of m_X, m_Y are singletons, then $\text{bel}(\text{X} \nabla \text{Y})$ is Bayesian i.e., $\text{bel}(\text{X} \nabla \text{Y}) = \text{pl}(\text{X} \nabla \text{Y}) = \text{Pr}(\text{X} \nabla \text{Y})$.

General Properties

- $bel(\emptyset) = 0, \quad bel(\Omega) = 1.$
- $bel(A) + bel(\overline{A}) \leq 1.$
- $bel(A_1 \cup A_2 \cup ... \cup A_n) \geq \sum_i bel(A_i) - \sum_{i<j} bel(A_i \cap A_j) + ... + (-1)^{n+1} bel(A_1 \cap ... \cap A_n).$

It is worth noticing that the evidence theory is very powerful in terms of resolving many and different problems dealing with uncertain data. Let us enumerate some works using evidential theory. Hong et al. (Hong, He, & Bell, 2010) have used the evidential approach to query interface matching on the deep web. Add to that, evidential theory has also been used to extract frequent items so-called Data Mining (Samet, Lefevre, & Yahia, 2014).

Let now $m_1 : 2^\Theta \longrightarrow [0, 1]$ be a mass function induced on the frame of discernment Θ by a source S_1 (piece of evidence) and $m_2 : 2^\Theta \longrightarrow [0, 1]$ a mass function induced on the frame of discernment Θ by another source S_2. Assuming that all sources are reliable and consistent, Shafer (Shafer, 1976) introduced the rule of combination to merge mass functions provided by distinct and independent sources. On the other hand, Smets (Smets, 2008) proposed also the conjunctive and disjunctive rules of combination for two mass functions m_1 and m_2 defined on the same frame of discernment Θ.

Definition 2.2. *(Conjunctive rule of combination) Considering two mass functions m_1 and m_2 defined on Θ. For all $A \in 2^\Theta$, this rule writes:*

$$m_{\bigcirc\!\!\!\!\cap}(A) = \sum_{B \cap C = A} m_1(B) * m_2(C) \tag{2.9}$$

$m_{\textcircled{\cap}}(A)$ represents the combined mass function attributed to the proposition A.

Definition 2.3. *(The disjunctive rule of combination) Considering two mass functions m_1 and m_2 defined on Θ. For all $A \in 2^{\Theta}$, this rule writes:*

$$m_{\textcircled{\cup}}(A) = \sum_{B \cup C = A} m_1(B) * m_2(C) \tag{2.10}$$

In some cases, we need to combine two bba m_1 and m_2 that are defined on different frames of discernment Θ_1 and Θ_2, respectively. However, rules of combination need that *bbas* have the same frame of discernment. The vacuous extension of belief functions (Shafer, 1976) is a solution for that problem.

Definition 2.4. *(Vacuous extension) The vacuous extension of belief functions defines bbas on a compatible frame of discernment called the join frame of discernment which is an extension of Θ_1 and Θ_2 denoted by Θ where $\Theta = \Theta_1 \times \Theta_2$.*

2.2.2 The evidential database model

As said above, several real-world applications deal with imperfect data. Introduced by Lee (Lee, 1992b), evidential databases aim at modelling imperfect data (i.e., imprecise, missing and uncertain data). The evidential databases provide a solid framework to model and manage such type of information. For instance, many works (Lee, 1992a; Bach Tobji et al., 2008; Bell et al., 1996) have addressed the issue of modeling uncertainty in database systems.

An evidential database can be presented by its compact form as well as a set of possible worlds. We show in the following the two different ways to present an evidential database (EDB).

The compact form of an EDB

The evidential databases can be seen as a set of evidential objects where its attributes values are modeled in the setting of the evidence theory.

Definition 2.5. *An evidential database is a collection of objects $\mathcal{O}$ defined on a set of attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$ where each attribute a_k has a domain Θ_{a_k}. We assume here that Θ_{a_k} is endowed with an ordered relation. The relation between the i^{th} object and the k^{th} attribute is expressed by a bba. The bba may contain one or more focal elements A.*

One can write:
$o_i.a_k = \{\langle A, m_{ik}(A)\rangle \mid A \subseteq \Theta_{a_k}, m_{ik}(A) > 0\}$ *where* $m_{ik} : 2^{\Theta_{a_k}} \longrightarrow [0,1]$, *with* $m_{ik}(\emptyset) = 0$ *and* $\sum_{A \subseteq \Theta_{a_k}} m_{ik}(A) = 1$.

Table 2.1: Evidential database.

Target	Distance (10^3.km)	Altitude (10^3.km)
t_1	$\langle\{150, 160, 180\}, 0.1\rangle$, $\langle\{190, 200\}, 0.9\rangle$	$\langle 60, 0.3\rangle$, $\langle 100, 0.7\rangle$
t_2	$\langle 100, 0.7\rangle$ $\langle\Theta_{Distance}, 0.3\rangle$	$\langle\{70, 80\}, 0.8\rangle$, $\langle 80, 0.2\rangle$
t_3	70	$\Theta_{Altitude}$
t_4	$\langle\{50, 60\}, 0.8\rangle$, $\langle\{65\}, 0.2\rangle$	60
t_5	$\langle\{50\}, 0.5\rangle$, $\langle\{60\}, 0.5\rangle$	$\langle\{60\}, 0.6\rangle$, $\langle\{70\}, 0.4\rangle$

Example 2.3. *Let us consider an example of the sensor data about target observations. Assume that observations are pervaded with uncertainty modelled thanks to evidence theory, as shown in Table 2.1. Each target observation, defined on each attribute, may have one or more focal elements. For example, the distance of target* t_1, *called a bba includes two focal elements* $\langle\{150, 160, 180\}\rangle$ *and* $\langle\{190, 200\}\rangle$ *where their mass functions are respectively* 0.1 *and* 0.9. *That is, we believe that the attribute value is either 150, 160 or 180 with a mass function 0.1 or one of the values 190 or 200 with a mass 0.9. However, we do not know how credible each single element is.*
Note that the evidential databases can store various kind of data imperfections: Probabilistic data when focal elements are singletons (t_1*.Altitude), Possibilistic data when focal elements are nested (*t_2*.Altitude), Partial ignorance:* $0 < m(\Theta_{a_k}) < 1$, *(*t_2*.Distance), Perfect data when focal element is singleton and its mass is equal to one (*t_3*.Distance), or Total ignorance when* $m(\Theta_{a_k}) = 1$ *(*t_3*.Altitude).*

Generally, evidential databases are obtained by collecting different experts opinions/observations (Ha-Duong, 2008). Another way to obtain this type of databases, is the matching of different databases having the same attributes and objects, including different and missing values. But, in our case, evidential databases result from collecting different observation results of radars.

Possible worlds form of an EDB

In this section, we present the possible worlds of an evidential database. This form is very important when modelling uncertain database operations. Indeed, a database operation

performed on the compact form is said to be a strong representation system if it produces the same result (when exploded) when applied on the set of possible worlds of that database. Hence, one can see the importance of the possible worlds form to valid uncertain database operators.

As described in (Bousnina, Bach Tobji, Chebbah, Liétard, & Ben Yaghlane, 2015), the non-compact form of an evidential database $\mathcal{O}$ of n objects is a set of imprecise worlds $\{IW_1, IW_2, .., IW_t\}$ where each IW_i contains n objects such that each object is defined by only one focal element per attribute. Such representation is obtained using the disjunctive rule of combination.

Example 2.4. *Let Table 2.2 describe the evidential database $\mathcal{O}$ defined on the two attributes A and B. The imprecise worlds (IW) derived from $\mathcal{O}$ are shown in Table 2.3*

Table 2.2: Evidential database $\mathcal{O}$.

Objects	A	B
Z	$\langle\{a_1\}, 0.3\rangle$, $\langle\{a_1, a_2\}, 0.7\rangle$	b_1
T	a_5	$\langle\{b_1, b_0\}, 0.4\rangle$, $\langle b_1, 0.6\rangle$

Table 2.3: Imprecise Worlds of $\mathcal{O}$.

IW_1 (0.12)	IW_2 (0.18)	IW_3 (0.28)	IW_4 (0.42)
(Z, a_1, b_1) 0.3	(Z, a_1, b_1) 0.3	$(Z, \{a_1, a_2\}, b_1)$ 0.7	$(Z, \{a_1, a_2\}, b_1)$ 0.7
$(T, a_5, \{b_1, b_0\})$ 0.4	(T, a_5, b_1) 0.6	$(T, a_5, \{b_1, b_0\})$ 0.4	(T, a_5, b_1) 0.6

Each imprecise world IW_i can be expanded to different possible worlds PW as detailed in Table 2.4.

Table 2.4: Possible worlds of $\mathcal{O}$.

IW_1 (0.12)	
PW_1	PW_2
(Z, a_1, b_1)	(Z, a_1, b_1)
(T, a_5, b_1)	(T, a_5, b_0)

IW_2 (0.18)
PW_1
(Z, a_1, b_1)
(T, a_5, b_1)

IW_4 (0.42)	
PW_1	PW_3
(Z, a_1, b_1)	(Z, a_2, b_1)
(T, a_5, b_1)	(T, a_5, b_1)

IW_3 (0.28)			
PW_1	PW_2	PW_3	PW_4
(Z, a_1, b_1)	(Z, a_1, b_1)	(Z, a_2, b_1)	(Z, a_2, b_1)
(T, a_5, b_1)	(T, a_5, b_0)	(T, a_5, b_1)	(T, a_5, b_0)

The basic belief assignment of the different possible worlds are computed in the following way:
$m(IW_1) = m(\{PW_1, PW_2\}) = 0.12$
$m(IW_2) = m(\{PW_1\}) = 0.18$
$m(IW_3) = m(\{PW_1, PW_2, PW_3, PW_4\}) = 0.28$
$m(IW_4) = m(\{PW_1, PW_3\}) = 0.42$

The possible worlds form of an evidential database is essential to prove if a model constitutes a strong representation system (Imieliński & Lipski, 1984). However, this form results in a high computational cost. That is why, in the literature, researchers generally use the EDB compact form for querying (Bell et al., 1996), analyzing (Elmi et al., 2014) and mining (Bach Tobji et al., 2008; Samet et al., 2014) evidential data.

2.3 Skyline Operator

In this section, we present some previous important works proposed to deal with the skyline operator both on certain and imperfect data.

2.3.1 Skyline operator on certain data

Basic Concepts

The idea of Skyline queries has attracted the interest of different communities. It is worth noticing that preferences can be relevant to many types of attributes in order to return

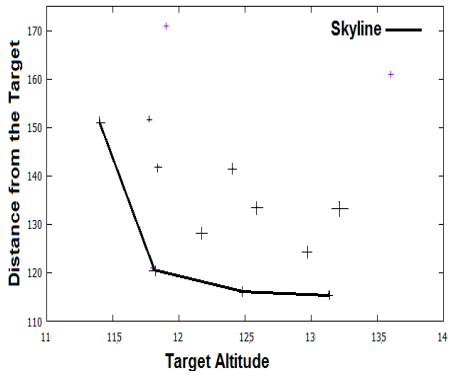


Figure 2.1: Skyline of Certain Sensor Observations

a set of best possible items combining multiple preference criteria. Preferences can be expressed quantitatively by a scoring function (Aggarwal & Yu, 2009), or qualitatively (Kiessling, 2002; Chomicki, 2007) which is the more general approach. Many definitions have been proposed in the literature, but all of them are based on the use of Pareto order. (Borzsonyi et al., 2001) have introduced skyline preferences that aim at filtering out a set of interesting objects from a large set of data. An object is interesting if it is not dominated by any other object in all dimensions. Let say that an object o_1 dominates (in the Pareto sense) another object o_2 if o_1 is at least as good as o_2 in all dimensions and better than o_2 in at least one dimension.

Consider a set $\mathcal{O}$ of n objects defined on a set of d attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$ defined on numerical domains. For simplicity and without loss of generality, we assume throughout all the chapter that the smaller the value the better. Recall that $o_i.a_k$ represents the value of the i^{th} object defined on attribute a_k.

Definition 2.6. *(Pareto Dominance) Given two objects $o_i, o_j \in \mathcal{O}$, o_i dominates o_j (in the sense of Pareto), denoted by $o_i \succ o_j$, if and only if o_i is as good or better than o_j in all attributes a_k $(1 \leq k \leq d)$ and strictly better in at least one attribute a_{k_0} $(1 \leq k_0 \leq d)$, i.e., $\forall a_k \in \mathcal{A} : o_i.a_k \leq o_j.a_k \wedge \exists a_{k_0} \in \mathcal{A} : o_i.a_{k_0} < o_j.a_{k_0}$.*

Definition 2.7. *(Skyline) The skyline of $\mathcal{O}$, denoted by $Sky_{\mathcal{O}}$, includes objects of $\mathcal{O}$ that are not dominated by any other object, i.e.,*

$$Sky_{\mathcal{O}} = \{o_i \in \mathcal{O} \mid \nexists\ o_j \in \mathcal{O}, o_j \succ o_i\}.$$

Example 2.5. *Figure 2.1 shows the skyline of the database $\mathcal{O}$ (depicted in Table 2.5) of flying objects representing the most dangerous targets (i.e., objects with small distance and altitude). Note that objects in the skyline are those that are not dominated by any other object in $\mathcal{O}$ w.r.t. to the skyline attributes, i.e., distance and altitude.*

Table 2.5: Certain Data.

Target	Distance (km)	Altitude (km)
t_1	119	17
t_2	136	16
t_3	114	15
t_4	118	12

SQL Extensions

In order to specify skyline queries, authors in (Borzsonyi et al., 2001) proposed to extend SQL's SELECT statement by an optional SKYLINE OF clause as follows (we denote by d_i the i^{th} dimension or attribute):

```
SELECT ... FROM ... WHERE ...
GROUP BY ... HAVING ...
SKYLINE OF [DISTINCT] d1 [MIN | MAX | DIFF], ..., dm [MIN | MAX | DIFF]
```

where MIN, MAX, DIFF denote minimum, maximum and different, respectively. For instance, to return the skyline objects from the targets database, we write:

```
SELECT * FROM Targets
SKYLINE OF distance [MIN], altitude [MIN]
```

This SQL query can be written without using the Skyline clause but it is very expensive in term of execution time. The following standard SQL query is equivalent to the previous query:

```
SELECT * FROM Targets T
WHERE NOT EXISTS( SELECT * FROM Targets T1
              WHERE T1.distance ≤ h.distance
              AND T1.altitude ≤ h.altitude
              AND (T1.distance < T.distance OR T1.altitude < T.altitude));
```

Skyline Algorithms

In the literature, many algorithms were proposed to compute the skyline from a database. Hereafter, we present some of them:

Divide and Conquer Algorithm

The divide and conquer algorithm can proceed as follows:

- The first step is to get a median value. Then, it divides objects into two partitions

P1 and P2.

- The second step is to compute the Skylines S1 and S2 of P1 and P2, respectively. This is done by recursively applying the whole algorithm to P1 and P2. The recursive partitioning stops if a partition contains few objects.
- Finally, it consists to merge S1 and S2 to compute the overall Skyline. This means eliminating all the objects of S2 which are dominated by objects of S1, i.e, no objects of S1 can be dominated by objects of S2.

BNL Algorithm

Block-Nested-Loop algorithm (BNL) works especially well if the skyline is small, and in the best case the skyline fits into the memory and it stops in one or two iterations. BNL algorithm runtime complexity in the best case is $O(n)$ where n is the number of tuples in the input, and the runtime in the worst case is $O(n^2)$. It shows very good I/O behaviour especially if the window can contain the whole skyline. BNL algorithm (see algorithm 1 and table 2.6) consists of the following steps (Borzsonyi et al., 2001):

- Outer loop: repeat over the input list.
- Inner loop: compare object to all identified candidates.
  - If object is dominated exit to the inner loop.
  - If object dominates candidate from the window, delete the candidate from the window.
- Objects that are remaining in the inner loop are added to the window.
- If window becomes full write the incomparable objects into temporary file.
- After outer loop stop
  - The objects that are completed with comparisons are added to the output list.
  - The temporary file is used as input list to manage the remaining comparisons.
- Repeat until the temporary file is empty.

Table 2.6: The summary of notation.

Notation	Definition
$\mathcal{M}$	Input of the skyline operation
$\mathcal{R}$	Output of the skyline operation
$\mathcal{T}$	The temporary file
$\mathcal{S}$	The main memory
$q \succ p$	point p is dominated by point q

Algorithm 1: SkylineBNL($\mathcal{M}$)

```
Input: R; T; S; CountIn:=0, CountOut:= 0
1  begin
2      while ÆOF(M) do
3          foreach p in S do
4              if TimeStamp(p)=CountIn then
5                  save(R, p), release (p);
6                  load(M, p), TimeStamp(p):=CountOut;
7                  CountIn:= CountIn + 1;
8          foreach q in S \ {p} do
9              if p ≻ q then
10                 release(p), break
11             if q ≻ p then
12                 release(q), break
13         if /MemoryAvalaible then
14             save(T, p), release (p);
15             CountOut:= CountOut + 1;
16         if EOF(M) then
17             M := T, T := ∅ ;
18             CountIn:=0, CountOut:= 0 ;
19     foreach p in S do
20         save(R, p), release (p);
21     return R
```

Three cases can occur for every tuple p at the beginning (Borzsonyi et al., 2001):

- P is dominated by any tuple in the window so p is discarded.
- P dominates any tuple in the window so p is added in the window and all tuples dominated by p are removed from the window.
- P is incomparable with all tuples in the window so p is added to the window if there is space otherwise p is written in the temporary file.

B-trees Algorithm

To compute the skyline, it is also possible to use an ordered index; e.g., a B-tree. One way to use an ordered index for a two-dimensional skyline is to scan through the whole index, get the objects in sorted order and filter out the objects of the skyline. Assuming that each object has d attributes and there is an index for every attribute, the skyline can be computed as follows.

- Scan all the indexes simultaneously to find the first match, i.e., the first object to be seen by all the indexes during the scan.
- The first match is definitely part of the skyline and can be returned immediately, providing a fast initial response
- Scan the rest of the index entries of the first attribute's index. If the object has not been seen before (i.e., the index entries of this object in the other indexes have not been examined prior to the first match), it is definitely not in the skyline and can thus be eliminated. If any of the other indexes contain an index entry to this object prior to the first match, then the object may or may not be in the skyline. To determine whether it is in the skyline, an existing skyline computation algorithm can be applied.

These algorithms have been ameliorated and improved by several research papers (Chomicki, Godfrey, Gryz, & Liang, 2003; Chomicki, 2007; Lin, Yuan, Wang, & Lu, 2005a).

2.3.2 Skyline operator over imperfect data

Many research projects have been conducted to deal with the skyline operator over imperfect database. We can refer to (Alwan, Ibrahim, Udzir, & Sidi, 2017; Bosc & Pivert, 2010; Jiang et al., 2012; Pei et al., 2007; Lian & Chen, 2008; Yu & Bouguettaya, 2010; Yong, Lee, Kim, & won Hwang, 2014; Groz & Milo, 2015; Ilaria, Paolo, & Marco, 2014).

In this section, we present some imperfect skyline modeled by the probability theory as well as the possibility theory. In addition, we review the stochastic skyline introduced by (Lin, Zhang, Zhang, & Cheema, 2011).

Probabilistic Skyline

Probabilistic skyline on uncertain data is first tackled by (Pei et al., 2007) where skyline objects are retrieved based on skyline probabilities. This idea is also developed and improved in (Jiang et al., 2012). There are other studies which have adopted the probabilistic skyline model. Lian et al. (Lian & Chen, 2009) combine reverse skyline with uncertain semantics and study the probabilistic reverse skyline problem in both monochromatic and bichromatic fashion. Atallah and Qi (Atallah & Qi, 2009) develop sub-quadratic algorithms to compute the skyline probabilities for every object. In (W. Zhang et al., 2013), authors tackle the problem of efficiently on-line computing probabilistic skyline over sliding windows. In (Yong et al., 2014), authors studied the problem of supporting skyline queries for uncertain data with maybe confidence.

To illustrate this kind of skyline, let us consider the motivating example shown in (Pei et al., 2007) that aims at analyzing players using the following multiple technical statistics criteria: the number of assists and the number of rebounds.

Let us assume that for both criteria, both the larger the better, to examine the players and to find the perfect one who can achieve the best performance in all aspects. Unfortunately, such a player does not exist. The skyline analysis here is meaningful since it discloses the trade-off between the merits of multiple aspects. A player U is in the skyline if there does not exist other player V such that V is better than U in one aspect, and is not worst than U in all other aspects. In figure 2.2, we plot a few games of 5 synthesis

Figure 2.2: A Set of Players

players to illustrate several important issues. In this example, performances in different

games may vary differently. In Figure 2.2, Arbor's performances are quite consistent while Eddy's performances are quite diverse. Although Eddy's performance in one game (point b) is better than Arbor's performances in all games in both criteria/ dimensions. Our goal is to detect the players that are not dominated by any other players, i.e., in the skyline set of the players data-set.

By default, we consider objects in an n-dimensional numeric space (Attributes) $\mathcal{A} = (a_1, ..., a_n)$. We assume that, on a given attribute, the smaller the value the better.

Definition 2.8. *Let U and V be two uncertain objects, and f and f' be the corresponding probability density functions, respectively. Let $U = \{u_1, ..., u_{l_1}\}$ and $V = \{v_n, ..., v_{l_2}\}$ be two uncertain objects and their instances. Then, the probability that V dominates U is given by (Pei et al., 2007):*

$$Pr[V \prec U] = \int_{u \in D} f(u)(\int_{v \prec u} f'(v)dv)du \tag{2.11}$$

$$Pr[V \prec U] = \frac{1}{l_1 l_2} \sum_{i=1}^{l_1} \mid \{v_j \in V \mid v_j \prec u_i\} \tag{2.12}$$

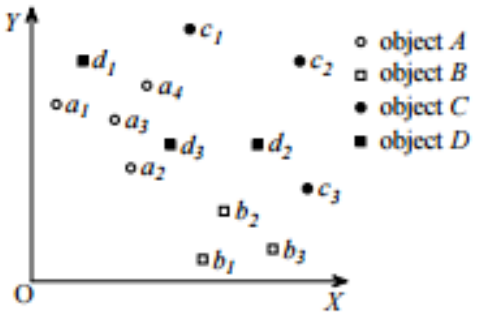


Figure 2.3: Uncertain objects

For instance, in Figure 2.3, let us compute the probability that object A dominates object C: $Pr(A \prec C) = \frac{1}{3}$ (C has 3 instances) $\times \frac{1}{4}$ (A has 4 instances)$\times$
(4 (c_1 is dominated by every instance of A) + 4 (c_2 is dominated by every instance of A) + 0 (c_3 is not dominated by any instance of A))
$= \frac{2}{3}$

Definition 2.9. *(The Skyline probability of U) Let $U = \{u_1, ..., u_{l_1}\}$ be an uncertain object and its instances, the probability that U is in the skyline is given by (Pei et al., 2007):*

$$Pr(U) = \frac{1}{l} \sum_{i=1}^{l} \prod_{V \neq U} (1 - \frac{\mid \{v \in V \mid v \prec u_i \mid\}}{\mid V \mid} \tag{2.13}$$

As another definition of skyline probability of U, the equation 2.13 can be differently written:

$$Pr(U) = \frac{1}{l} \sum_{u \in U} Pr(u) \tag{2.14}$$

Definition 2.10. *(The skyline probability of the instance $u \in U$) Let $U = \{u_1, ..., u_{l_1}\}$ be an uncertain object and its instances, $Pr(u)$ is the probability that u is not dominated by any other objects, i.e., u is in the skyline, is given by:*

$$Pr(u) = \prod_{V \neq U} (1 - \frac{\mid \{v \in V \mid v \prec u \mid\}}{\mid V \mid} \tag{2.15}$$

For uncertain data modeled by probabilistic distribution, (Pei et al., 2007) tackled the problem of computing probabilistic skylines on large uncertain data sets and proposed two efficient algorithms:

- The bottom-up algorithm: computes the skyline probabilities of some selected instances of uncertain objects, and uses those instances to prune other instances and uncertain objects effectively.
- The top-down algorithm: recursively divide the instances of uncertain objects into subsets, and aggressively prunes subsets and objects.

This idea has been improved in (Yong et al., 2014) by adding the maybe confidence measure that indicates the existence probability of a given object. (Fung, Lu, & Du, 2009) proposed an efficient algorithm being able to compare objects that are skyline points using the probabilistic skyline model. Add to that and thanks to the work of (Fung et al., 2009), we are able now to answer to the following question: Which of the objects are the K nearest neighbors to a given object according to their skyline probabilities?
Moreover, (Jiang et al., 2012) proposed a novel probabilistic skyline model where an uncertain object may take a probability to be in the skyline, and a p-skyline contains all objects whose skyline probabilities are at least p ($0 < p \leq 1$). Also, a bounding-pruning-refining algorithms were proposed too. (W. Zhang et al., 2013) studied the problem of efficient processing of continuous skyline queries over sliding windows on uncertain data regarding a given probability thresholds.

Possibilistic Skyline

In contrast with probability theory, when using possibility theory, we expect the following features: (i) the possibility theory offers a qualitative model for uncertainty. (ii) the sum of the possibility degrees is not necessary equal to 1.

For uncertain attribute values that are represented by possibility distribution, (Bosc et al., 2011; Benouaret et al., 2012) proposed to compute the possibility that a given object is not dominated by any other object. In this framework, skyline queries aim at computing the possibility degree that an object from a given relation is possibly not dominated by any other object from that relation. In addition, preferences can be represented in a possibilistic logic way using symbolic weights (Hadjali, Kaci, & Prade, 2008). As defined by (Bosc et al., 2011), the possibilistic dominance can be stated as follows:

Definition 2.11. *Let $(A_1, ..., A_n)$ be the schema of the relation queried. Let $(a_1, ..., a_p)$ be the attributes concerned by a preference in the query and res the result of the query. We denote by $\succ_{a_k}$ the preference relation defined over the attribute domain of attribute a_k.*
A given object o_i is dominated by another object o'_j, denoted by $o_i \prec o'_j$ iff:

$$\forall k \in \{1, ..., p\}, o'_j.a_k \succcurlyeq_{a_k} o_i.a_k \text{ and } \exists q \in \{1, ..., p\}, o'_j.a_q \succ_{a_q} o_i.a_q$$

Let $\Pi(o)$ denotes the degree of possibility that an object o from res be non-dominated by any other object o'. For each interpretation Π_i/o_i of o, we compute the possibility that for every object $o' \neq o$, there exists an interpretation o'_j of o' which does not dominate o_i. The final degree $\Pi(o)$ is the maximum of these degrees, computed over all the interpretations of o. This leads to:

$$\Pi(o) = max_{\Pi_i/o_i \in int(o)} \Pi(\Pi_i/o_i) \tag{2.16}$$

Where int(o) denotes the set of interpretations of o and

$$\Pi(\Pi_i/o_i) = min(\Pi_i, min_{o' \in resn\{o\}} \Pi(o_i \not\prec o')) \tag{2.17}$$

To compute the possibility dominance between two given objects o and o', we first should compute the possibility degree that each interpretation of o is dominated/ not dominated by o'

$$\Pi(o_i \prec o') = \begin{cases} 0, if \{\Pi_j/o'_j \in int(o') \mid o_i \prec o'_j = \emptyset, \\ max_{\Pi/o'_j \in int(o') | o_i \prec o'_j} \Pi_j, otherwise \end{cases} \tag{2.18}$$

Example 2.6. *Let us consider a relation of schema (make, category), the preferences user are as follows: $(VW \succ Ford \succ Opel)$ and $(SUV \succ roadster \succ others)$ and the objects:*

- $o_1 = \langle \{1/Opel, 0.8/VW\}, roadster \rangle$
- $o_2 = \langle Ford, \{1/SUV, 0.7/VW\}, Sedan \} \rangle$

- $o_3 = \langle\{1/VW, 0.6/Opel\}, roadster\rangle$

For computing $\Pi(o_1)$*, we have to identify the interpretations of* o_1*:*
$o_{1_1} = 1/\langle Opel, roadster\rangle$ *and* $o_{1_2} = 0.8/\langle VW, roadster\rangle$
Starting with o_{1_1} *we have:* $\Pi(o_{1_1} \not\prec o_2) = 0.7$ *and* $\Pi(o_{1_1} \not\prec o_3) = 0.6$ *Then, for* o_{1_2} *we have:* $\Pi(o_{1_2} \not\prec o_2) = 1$ *and* $\Pi(o_{1_2} \not\prec o_3) = 1$*, thus:*

$$\Pi(o_1) = max(min(1, min(0.7, 0.6)), min(0.8, min(1, 1))) = 0.8$$

Stochastic Skyline

In the light of utility of applications with uncertain data, the probabilistic skyline model is proposed to retrieve uncertain objects based on skyline probabilities. Nevertheless, skyline probabilities cannot capture the preferences of monotonic utility functions. A novel skyline operator, namely stochastic skyline, provides the minimum set of candidates for the optimal solutions over all possible monotonic multiplicative utility functions (Lin et al., 2011; W. Zhang, Lin, Zhang, Cheema, & Zhang, 2012).

Definition 2.12. *(Skyline) Given a class* F *(family of utility (scoring) functions where* $F = \{\prod_{i=1}^{d} f_i(x_i)\}$*, an uncertain object (random variable)* U *stochastically dominates* V *regarding* F*, denoted by* $U \prec_F V$ *if and only if*

$$E[f(U)] \geq E[f(V)] \text{ for each } f \in F$$

with f *is a given utility function and* E *is the optimal solution.*

$\mathbb{R}^d+$ is used to denote the points in $\mathbb{R}^d$ with non negative coordinate values. An uncertain object U is the set $\{u_1, ..., u_m\}$ instances in $\mathbb{R}^d+$ where for $1 \leq i \leq m$, u_i is in $\mathbb{R}^d+$ and has the probability p_{u_i} $(p_{u_i} > 0)$, and $\sum_{i=2}^{m} p_{u_i} = 1$

Definition 2.13. *(Stochastic Dominance) Given two uncertain objects* U *and* V*,* U *stochastically dominates* V*, denoted by* $U \prec_{sd} V$ *if*

$$U.cdf(x) \geq V.cdf(x), \forall x \in \mathbb{R}^d+ \text{ and } \exists y \in \mathbb{R}^d+ \text{such that } U.cdf(y) > V.cdf(y)$$

The probability mass $U.cdf(x)$ of U is the sum of the probabilities of the instances in $\mathbb{R} = ((0, ..., 0), x)$ where $(0, 0, ..., 0)$ is the origin in $\mathbb{R}^d$.

Definition 2.14. *(Stochastic Skyline) Given a set of uncertain objects* $\mathbf{D}$*, an object* $U \in \mathbf{D}$ *is a stochastic skyline object if there is no object* $V \in \mathbf{D}$ *such that* $V \prec_{sd} U$

2.3.3 Comparative Study

In this section, we present the main models already proposed in the literature that modeled the Skyline Operator in different types of data pervaded with imperfection. Table 2.7 contains the frequently used notation and its meaning.

Table 2.7: The summary of notation.

Notation	**Definition**
U, V, t, t'	Uncertain objects
u_i, v_j	Instances of the uncertain objects U and V, respectively
l_1, l_2	The number of instances of U and V, respectively
f(x)	The continuous object
$int(t)$	interpretation of the object t

The table 2.8 presents in an easy way, the definitions of the different proposed Skyline models.

Table 2.8: Imperfect Skyline Models.

Distributions		Dominance relationship	Degree to be in $\mathcal{S}$
Probabilistic Distribution	**Discrete case**	$Pr[V \prec U] =$ $\frac{1}{l_1 l_2} \sum_{i=1}^{l_1} \mid \{v_j \in V \mid v_j \prec u_i\}$	$Pr(U) =$ $\frac{1}{l} \sum_{u \in U} Pr(u)$
	Continuous Case	$P(f_i(\overrightarrow{x_i}) \prec f_j(\overrightarrow{x_j})) =$ $\int\int f_i(\overrightarrow{x_i}).f_j(\overrightarrow{x_j}).$ $\begin{cases} 1, \overrightarrow{x_i} \prec \overrightarrow{x_j} \\ 0, \overrightarrow{x_i} \not\prec \overrightarrow{x_j} \end{cases} (d\overrightarrow{x_i} d\overrightarrow{x_j})$	$P_S(f(x)) =$ $\int_{\mathbb{R}^d} f(x) \prod_{g(y) \in DB\prime}$ $\left(1 - \int_{\mathbb{R}^d} g(y). \begin{cases} 1, y \prec x \\ 0, otherwise \end{cases} dy\right) dx$
Possibilistic Distribution		$\Pi(t_i \prec t') =$ $\begin{cases} 0, if \\ \{\Pi_j / t'_j \in int(t') \mid t_i \prec t'_j = \emptyset, \\ max_{\Pi / t'_j \in int(t') \mid t_i \prec t'_j} \Pi_j, \\ otherwise \end{cases}$	$\Pi(t) =$ $max_{\Pi_i / t_i \in int(t)} \Pi(\Pi_i / t_i)$

In addition, it details for each uncertain model, the dominance relationship between the objects and its degrees of being in the skyline. The degree can be a probability, if data are modeled by the probabilistic distribution, or a possibility degree if data are represented by the possibilistic distribution.

Table 2.9 presents a comparative study between the different Skyline models proposed in the literature. New concepts mentioned in table 2.9 are presented as follows:

Table 2.9: Comparative table.

	Probabilistic data	Possibilistic Data
Skyline Computing	(Pei et al., 2007) (Jiang et al., 2012) (Atallah & Qi, 2009) (Ilaria et al., 2014)	(Pivert & Prade, 2014) (Bosc et al., 2011) (Hadjali, Souhila, & Henri, 2011) (Lian & Chen, 2013) (Hadjali et al., 2008)
Top-k Skyline	(Lian & Chen, 2009) (Fung et al., 2009) (W. Zhang et al., 2013)	$\emptyset$
Skyline Maintenance	(Zhenjie, Reynold, Dimitris, & Anthony, 2009) (Baichen, Weifa, & Xu, 2009)	$\emptyset$
Distributed Skyline	(Amagata, Sasaki, Hara, & Nishio, 2016) (Ding & Jin, 2010) (Li, Yi, & Jestes, 2009)	$\emptyset$

- **The Top-k Skyline**: We denote by $\mathcal{O}$ the data set of uncertain objects. Let $Sky_{\mathcal{O}}$ be the result of the Skyline Query (i.e., the set of objects that are not dominated by any other objects). The top-k Skyline consists on ranking objects in $Sky_{\mathcal{O}}$ and retrieving the k most interesting objects.
- **The skyline maintenance**: Maintaining the skyline results in the given database when an object is inserted or deleted.
- **The Distributed Skyline**: Computing the skyline operator over multiple and distributed servers.

To the best of our knowledge, there is no previous works that deal with computing the skyline over Evidential Data.

2.4 Conclusion

In this chapter, we first presented the basic concepts of the evidential database. Then, we presented a reminder about the skyline operator in the context of certain and uncertain data. Data can be modeled by different distributions such as the probability theory as well as the possibility theory. Both probabilistic and possibilistic skylines were detailed in this chapter.

As said above, the evidential databases have a powerful mechanism to deal with data uncertainty since it is a generalisation of both probabilistic and possibilistic models. For this reason, we consider that modeling skyline operator over evidential data is really a challenging work.

In the next chapter, we define the evidential skyline model. In addition, we propose new semantics for the skyline operator extended to the evidential data. We also propose efficient algorithms for the evidential skyline computation.

Chapter 3

Evidential Skyline

Contents

3.1 Introduction

Uncertainty is inherent in several applications such as environment surveillance, market analysis, and sensor networks. To deal with uncertain data, several studies were conducted into both possibilistic databases (Bosc et al., 2011; Bosc & Pivert, 2010), probabilistic databases (Pei et al., 2007; Aggarwal & Yu, 2009; N. Dalvi & Suciu, 2007; N. N. Dalvi & Suciu, 2007) and recently evidential databases (Bell et al., 1996; Bousnina et al., 2015). Evidential databases are very interesting since they can handle both possibilistic data, probabilistic data and other types of imperfect data. In addition, authors in (Samet et al., 2016) developed a belief-function based modelling approach to construct a chemical evidential database. Then, they developed a mining process to discover valid association rules between molecule characteristics and properties. In (Smets, 1999), the authors presented examples where the use of the evidence theory provides elegant solutions to real life problems such as the combination of data coming from sensors competent on partially overlapping frames. The author in (Ha-Duong, 2008) introduced a hierarchical fusion method based on the Transferable Belief Model to aggregate expert opinions and obtain a set of more representative belief distributions.

On the other hand, the present decade has seen a revival of interest in preference queries which aim at retrieving the objects from a database which are not dominated by any other object. Many previous studies such as ((Sharif Zadeh & Shahabi, 2006; Lin, Yuan, Wang, & Lu, 2005b; Huang, Jensen, Lu, & Ooi, 2006; Chan, Jagadish, Tan, Tung, & Zhang, 2006)) showed that skyline queries are very useful in multi-criteria decision making applications. Starting with the pioneering work of (Borzsonyi et al., 2001), which provided foundations for a Pareto-order-based preference model for database systems.

3.1.1 Motivating Example

As pointed out above, a large amount of works (Borzsonyi et al., 2001; Chan et al., 2006; Huang et al., 2006; Lin et al., 2005b; Yuan et al., 2005) proved that skyline queries provide an adequate tool that can help users to make intelligent decisions in the presence of multidimensional data where different and often conflicting criteria must be considered.

As an example, consider analyzing sensor data: to be aware of the plane environment in the ordinary situation or to get more information about targets in a state of war, it is absolutely necessary to analyze sensor performances to identify the most dangerous target. To analyze sensor observations, we can use some multiple technical measures criteria (e.g., distance from the target, its altitude, its speed, etc..). We suppose targets having the smaller distance and smaller altitude are the most dangerous. Ideally, we want to find the

perfect sensor observation that returns, among n targets, which is the most dangerous in all aspects. While such target does not exist, the skyline discloses the trade-off between the different preferences. A target U is in the skyline if there exists no other target V such that V dominates (in Pareto sens) U. [1]
We are persuaded that skyline analysis is also meaningful on uncertain data as shown in section 1.1. Our challenge is to answer the following question: which target is the most dangerous if the sensor observations are uncertain and noisy? i.e., which objects are in the evidential skyline ?

3.1.2 Contributions

In this chapter, and starting from our conference work (Elmi et al., 2014), we propose new models to skyline definition and computation over uncertain data where uncertainty is represented thanks to evidence theory. We particularly define two main models, the first model is presented in section 3.2 and has a Boolean semantics. The second model is discussed in section 3.3. The key notion of this model is the concept of knowledge states associated to the real world.

In particular, the following two major contributions are made.

Contribution 1 is about semantics aspect. First, we propose some extensions of Pareto dominance to evidential data. The first skyline semantics respects perfectly the classic skyline definition. A second semantic is based on representing an evidential object U as a set of instances, also called states of knowledge. Each knowledge state is associated with a mass function, which reflects the degree of belief to which it represents the real world, i.e., the object U.

The issue of comparing two evidential objects is discussed by leveraging the knowledge state representation. A belief (resp. plausible) degree that one object dominates another is then defined. Second, an evidential counterpart of the skyline is discussed according to two variants: the belief skyline and the plausible skyline. We show that the latter is more interesting in term of refinement.

Contribution 2 is about algorithmic and computing aspect. We develop efficient algorithms to tackle the problem of evidential skyline computation. In particular, we provide appropriate comparison methods between two evidential objects through reducing the num-

[1] V dominates U if V is smaller than or equal to U in all dimensions, and strictly smaller in at least one dimension.

ber of dominance checks and hence improving the performance of the proposed algorithms.

The contributions presented in this chapter are under consideration in the International Journal of Information Sciences.

3.1.3 Chapter Organisation

The rest of the chapter is organized as follows. I sections 3.2 and 3.3, we introduce the evidential skyline. Then, we present our algorithms to compute this kind of skyline in section 3.4, while in section 3.5, we report our experimental evaluation. Finally, section 3.6 concludes the chapter.

3.2 Evidential Skyline

In this section, and starting from our work (Elmi et al., 2014), the following issues are discussed:

- We first propose a new semantics of the skyline extended to the evidential databases.
- Then, we introduce two kind of evidential skylines the believable skyline (denoted by the b-skyline) and the plausible skyline (denoted by the p-skyline).
- We develop suitable algorithms based on scalable techniques to efficiently compute the b-skyline and the p-skyline.

3.2.1 Belief Skyline

Given a set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ defined on a set of attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$, with $o_i.a_k$ denotes the *bba* of object o_i w.r.t. attribute a_k. According to Definition 2.1, the degree of belief that an object o_i is better than or equal (or strictly better) to another object o_j w.r.t. an attribute a_k writes :

$$bel(o_i.a_k \leq o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A \leq^{\forall} B} m_{jk}(B)) \tag{3.1}$$

Where $A \leq^{\forall} B$ stands for $a \leq b$, $\forall (a, b) \in A \times B$.

$$bel(o_i.a_k < o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A <^{\forall} B} m_{jk}(B)) \tag{3.2}$$

Where $A <^{\forall} B$ stands for $a < b$, $\forall (a, b) \in A \times B$.

Table 3.1: Evidential database example.

Target	Distance (km)	Altitude (km)
t_1	$\langle\{150, 160, 180\}, 0.1\rangle$, $\langle\{190, 200\}, 0.9\rangle$	$\langle 60, 0.3\rangle$, $\langle 100, 0.7\rangle$
t_2	$\langle 100, 0.7\rangle$ $\langle\Theta_{Distance}, 0.3\rangle$	$\langle\{70, 80\}, 0.8\rangle$, $\langle 80, 0.2\rangle$
t_3	70	$\Theta_{Altitude}$
t_4	$\langle\{50, 60\}, 0.8\rangle$, $\langle\{65\}, 0.2\rangle$	60
t_5	$\langle\{50\}, 0.5\rangle$, $\langle\{60\}, 0.5\rangle$	$\langle\{60\}, 0.6\rangle$, $\langle\{70\}, 0.4\rangle$

Example 3.1. *In Table 3.1, one can check that* [2] $bel(t_1.d \leq t_2.d) = 0$, $bel(t_1.a \leq t_2.a) = 0.3$, $bel(t_2.d < t_1.d) = 0.7$, *and* $bel(t_2.a \leq t_1.a) = 0.7$

Let us now present the notion of the belief skyline. This later aims at retrieving the most interesting objects in $\mathcal{O}$ that are not believably dominated by any other objects. We first present the concept of the believable dominance denoted by the b-dominance and then the belief skyline denoted by the b-skyline.

Definition 3.1. *(Believable dominance) Given two objects $o_i, o_j \in \mathcal{O}$ and a belief threshold b, o_i b-dominates o_j denoted by $o_i \succ_b o_j$ if and only if o_i is believably as good or better than o_j in all attributes a_k in $\mathcal{A}$ $(1 \leq k \leq d)$ and strictly believably better in at least one attribute a_{k_0} $(1 \leq k_0 \leq d)$ according to a belief threshold b, i.e.,*
$\forall a_k \in \mathcal{A} : bel(o_i.a_k \leq o_j.a_k) \geq b$ *and* $\exists a_k \in \mathcal{A} : bel(o_i.a_k < o_j.a_k) \geq b$.

Example 3.2. *Let b=0.2. One can check that in Table 3.1, t_5 0.2-dominates t_4 since* $bel(t_5.d < t_4.d) = 0.2$ *and* $bel(t_5.a \leq t_4.a) = 0.6$.

In order to define the b-skyline, it is essential to state the following key property of the b-dominance.

Property 3.1. *The b-dominance relationship does not satisfy the property of transitivity.*

Proof. Consider the objects depicted in Table 3.2. We have, $t_x \succ_{b=0.4} t_y$, $t_y \succ_{b=0.4} t_z$ and $t_x \succ_{b=0.07} t_z$. Observe that, t_x 0.4-dominates t_y and t_y 0.4-dominates t_z, but t_x does not 0.4-dominate t_z. Thus, the b-dominance relationship is not transitive. □

[2] d and a denote the distance and the altitude attributes, respectively

Table 3.2: Example of Evidential Data.

Target	Distance	Altitude
t_x	$\langle\{90, 120\}, 0.7\rangle$, $\langle\{150, 160\}, 0.3\rangle$	$\langle 80, 0.1\rangle$, $\langle\{90, 100\}, 0.9\rangle$
t_y	$\langle\{170, 180\}, 1\rangle$	$\langle\{60, 70\}, 0.6\rangle$, $\langle 100, 0.4\rangle$
t_z	$\langle\{100, 110\}, 0.5\rangle$, $\langle\{190, 200\}, 0.5\rangle$	$\langle 70, 0.3\rangle$, $\langle\{80, 90\}, 0.7\rangle$

Given an object o_i, we denote by $o_i.a_k^-$ and by $o_i.a_k^+$ respectively the minimum value and the maximum value of $o_i.a_k$. For example, in Table 3.2, $t_x.d^- = 90$ and $t_x.d^+ = 160$

Property 3.2. *if* $\exists a_k \in \mathcal{A}$ *where* $o_i.a_k^+ < o_j.a_k^-$ *then* o_j *does not dominate* o_i*, i.e.,* $o_j \not\succ o_i$ *since* $bel(o_j.a_k \leq o_i.a_k) = 0$*.*

Example 3.3. *Let* $o_i.a_k$ *and* $o_j.a_k$ *be two bbas defined on objects* o_i *and* o_j*, respectively, and defined on the attribute* a_k *such that* $o_i.a_k = \{100, 200\}$ *and* $o_j.a_k = \{250, 300\}$*. We have* $bel(o_j.a_k \leq o_i.a_k) = 0$ *since* $200 < 250$*. Thus* $o_j \not\succ_b o_i$*.*

Intuitively, an object is in the believable skyline if it is not believably dominated by another object. Based on the b-dominance relationship, the notion of b-skyline is defined as follows.

Definition 3.2. *(Belief skyline) The believable skyline of* $\mathcal{O}$*, denoted by b-skyline, comprises those objects in* $\mathcal{O}$ *that are not b-dominated by any other object, i.e.,* $b\text{-skyline} = \{o_i \in \mathcal{O} \mid \nexists\, o_j \in \mathcal{O}, o_j \succ_b o_i\}$*.*

Property 3.3. *Given two belief thresholds* b *and* b'*, if* $b < b'$ *then the b-skyline is a subset of the* b'*-skyline, i.e.,* $b < b' \Rightarrow b\text{-skyline} \subseteq b'\text{-skyline}$*.*

Proof. Assume that there exists an object o_i such that $o_i \in b$-skyline and $o_i \notin b'$-skyline. Since $o_i \notin b'$-skyline, there must exists another object, say o_j, that b'-dominates o_i. Thus, $\forall a_k \in \mathcal{A} : bel(o_j.a_k \leq o_i.a_k) > b'$. But, $b < b'$, therefore, $\forall a_k \in \mathcal{A}, bel(o_j.a_k \leq o_i.a_k) > b$ holds also. Hence, $o_j \succ_b o_i$, which leads to a contradiction as $o_i \in b$-skyline. □

Property 3.3 indicates that the size of the b-Skyline is smaller than the b'-Skyline if $b < b'$. Roughly speaking, this means that the threshold b constitutes a tool to control the size of the believable skyline.

Example 3.4. *In Table 3.1, one can observe that* t_1*,* t_2 *and* t_3 *can not be in the 0.5-skyline since they are 0.5-dominated. However, we have* $t_4 \succ_{0.4} t_5$ *and* $t_5 \succ_{0.2} t_4$*. Thus, the 0.5-skyline comprises* t_4 *and* t_5 *since they are not 0.5-dominated by any other object, while the 0.4-Skyline contains only* t_4 *as* t_5 *is 0.4-dominated by* t_4*. 0.4-skyline* $\subseteq$ *0.5-skyline*

3.2.2 Plausible Skyline

Given a set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ defined on a set of attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$, with $o_i.a_k$ denotes the *bba* of object o_i w.r.t. attribute a_k. According to Definition 2.1, the degree of plausibility that an object o_i is better than or equal (or strictly better) to another object o_j w.r.t. an attribute a_k writes :

$$pl(o_i.a_k \leq o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A \leq^{\exists} B} m_{jk}(B)) \tag{3.3}$$

Where $A \leq^{\exists} B$ means for every $a \in A$, $\exists b \in B$ such that $a \leq b$.

$$pl(o_i.a_k < o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A <^{\exists} B} m_{jk}(B)) \tag{3.4}$$

Where $A <^{\exists} B$ means for every $a \in A$, $\exists b \in B$ such that $a < b$.

Example 3.5. *In Table 3.1, one can check that* $pl(t_1.d \leq t_2.d) = 0.3$, $pl(t_1.a \leq t_2.a) = 0.3$, $pl(t_2.d \leq t_1.d) = 1$, $pl(t_2.a \leq t_1.a) = 0.7$ *and* $pl(t_5.d < t_4.d) = 0.6$

Let us now present the notion of the plausible skyline. This later aims at retrieving the most interesting objects in $\mathcal{O}$ that are not plausibly dominated by any other object.

Let us first introduce the notion of the plausible dominance, denoted by p-dominance, and then the plausible skyline, denoted by p-skyline.

Definition 3.3. *(Plausible dominance) Given two objects* $o_i, o_j \in \mathcal{O}$ *and a plausibility threshold* p, o_i *p-dominates* o_j, *denoted by* $o_i \succ_p o_j$, *if and only if* o_i *is plausibly as good or better than* o_j *in all attributes* a_k *in* $\mathcal{A}$ $(1 \leq k \leq d)$ *and strictly plausibly better in at least one attribute* a_{k_0} $(1 \leq k_0 \leq d)$ *according to a plausible threshold* p, *i.e.,*
$\forall a_k \in \mathcal{A} : pl(o_i.a_k \leq o_j.a_k) \geq p$ *and* $\exists a_{k_0} \in \mathcal{A} : pl(o_i.a_{k_0} < o_j.a_{k_0}) \geq p$.

Example 3.6. *Let us again consider the evidential objects depicted in Table 3.1, we have* $t_4 \succ_{p=0.4} t_5$ *since* $pl(t_4.d \leq t_5.d) = 0.8$ *and* $pl(t_4.a < t_5.a) = 0.4$, *however,* $t_5 \succ_{p=0.6} t_4$ *since* $pl(t_5.d < t_4.d) = 0.6$ *and* $pl(t_5.a \leq t_4.a) = 0.6$

Property 3.4. *The p-dominance relationship does not satisfy the property of transitivity as well.*

Proof. One can check that $t_y \succ_{p=0.5} t_z$, $t_z \succ_{p=0.5} t_x$ but t_y does not dominate at all t_x since $pl(t_y.d \leq t_x.d) = 0$. Thus, the p-dominance relationship is not transitive. □

Definition 3.4. *(Plausible skyline) The plausible skyline of $\mathcal{O}$, denoted by p-skyline, comprises those objects in $\mathcal{O}$ that are not p-dominated by any other object, i.e., p-skyline = $\{o_i \in \mathcal{O} \mid \nexists\ o_j \in \mathcal{O}, o_j \succ_p o_i\}$.*

Property 3.5. *Given two plausible thresholds p and p', if $p < p'$ then the p-skyline is a subset of the p'-skyline, i.e., $p < p' \Rightarrow$ p-skyline $\subseteq$ p'-skyline.*

Proof. Similar way as Property 3.2.1. □

Example 3.7. *From Table 3.1, one can check that the 0.7-skyline = $\{t_4, t_5\}$ while the 0.6-skyline = $\{t_5\}$ thus 0.6-skyline $\subseteq$ 0.7-skyline*

For comparison purposes, the algorithms and the experiments of this evidential skyline model are presented and discussed in chapter 6.

3.3 Evidential Skyline oriented Knowledge States

In this section, first, we propose an extension of Pareto dominance to evidential data by representing an evidential object U as a set of instances, also called states of knowledge. Each knowledge state is associated with a mass function, which reflects the degree of belief to which it represents the real world, i.e., the object U. This section is organized in three parts; we first introduce the basic notions about knowledge states of an evidential object. Second, we describe the dominance relationship between two knowledge states. In the third part, we define the new semantics for the evidential dominance between two evidential objects.

3.3.1 Knowledge states

A knowledge state is an instance of an evidential object. An instance reflects a part of evidence of an evidential object defined on the set of attributes $\mathcal{A} = \{a_1, a_2, .., a_d\}$ and on the joint frames of discernment denoted by Θ where $\Theta = \prod_{k=1}^{d} \Theta_{a_k}$ where Θ_{a_k} stands for the domain of the attributes a_k.

Definition 3.5. *(Knowledge State) A knowledge state u of an object U is a combination of focal elements defined on the different Θ_{a_k} in Θ. The mass of a knowledge state u is computed using the conjunctive rule of combination of the extended bba's evidential values.*

$$m : \Theta \rightarrow [0, 1]$$

$$m(\emptyset) = 0$$
$$\sum_{u \in U} m(u) = 1$$

The mass of a knowledge state is the cross product of all the focal elements masses.

$$m(u \in U) = m^{\Theta_{a_k} \uparrow \Theta}(u^{\Theta_{a_k} \uparrow \Theta})$$
$$m(u \in U) = \prod_{k=1}^{d} m(u^{\Theta_{a_k} \uparrow \Theta})$$

Definition 3.6. *(Evidential Object) An evidential object U is a set of knowledge states, i.e, $U = \{u_1, u_2, \ldots, u_n\}$ where $n = \prod_{k=1}^{d} |F_{a_k}|$ and $|F_{a_k}|$ is the number of focal elements on the k^{th} attribute. The frame of discernment of an object is the joint frames of discernment denoted by $\Theta = \prod_{k \leq d} \Theta_{a_k}$.*

Example 3.8. *To illustrate the previous definitions, we present here the knowledge states that constitute object 1 in the database example provided in Table 3.3.*

Table 3.3: Evidential database

Object	**A**	**B**	**C**
U	$\langle A_1, 0.6\rangle$	B_1	$\langle C_2, 0.2\rangle$
	$\langle A_2, 0.4\rangle$		$\langle \{C_1, C_2\}, 0.8\rangle$
V	$\langle A_1, 0.2\rangle$	B_1	$\langle C_1, 0.5\rangle$
	$\langle \{A_2, A_3\}, 0.8\rangle$		$\langle C_2, 0.5\rangle$

The frame of discernment Θ is the cross product of all the attributes' domains Θ_{a_k}. Since knowledge states represent all possible combinations, thus the evidential object U is a set of knowledge states $\{u_1, u_2, u_3, u_4\}$ where:
$u_1 = A_1B_1C_2$, $u_2 = A_1B_1\{C_1, C_2\}$, $u_3 = A_2B_1C_2$ and $u_4 = A_2B_1\{C_1, C_2\}$. In a first step, we extend the bba of each evidential value composing the object U. For example, the extended knowledge states that correspond to the attribute A in the line 1 are:

$$m^{\Theta_A \uparrow \Theta}(A_1\Theta_B\Theta_C) = 0.6 \text{ and } m^{\Theta_A \uparrow \Theta}(A_2\Theta_B\Theta_C) = 0.4$$

Once we have the extended bba of all evidential values, we combine them via the conjunctive rule, we obtain:
$m_1(A_1B_1C_2) = m_1(A_1\Theta_B\Theta_C) \times m_1(\Theta_A B_1 \Theta_C) \times m_1(\Theta_A \Theta_B C_2) = 0.6 \times 1 \times 0.2 = 0.12$,

$m_1(A_1B_1\{C_1, C_2\}) = 0.6 \times 1 \times 0.8 = 0.48$,
$m_1(A_2B_1C_2) = 0.4 \times 1 \times 0.2 = 0.08$ *and*
$m_1(A_2B_1\{C_1, C_2\}) = 0.4 \times 1 \times 0.8 = 0.32$.
As a result, the object U *corresponds to the following knowledge states:*
$U = \{u_1 : \langle\{A_1, B_1, C_2\}, 0.12\rangle, u_2 : \langle\{A_1, B_1, \{C_1, C_2\}\}, 0.48\rangle,$
$u_3 : \langle\{A_2, B_1, C_2\}, 0.08\rangle, u_4 : \langle\{A_2, B_1, \{C_1, C_2\}\}, 0.32\rangle\}$

3.3.2 Belief skyline oriented knowledge states

In this section, we define the belief dominance relationship between knowledge states. We then extend this dominance to evidential objects, which plays a key role to define the belief skyline.

Belief dominance between knowledge states

Given two evidential objects represented by their knowledge states $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_m\}$ defined on d attributes which the joint frame of discernment is $\Theta = \prod_{k \leq d} \Theta_{a_k}$. The *belief dominance* is our belief that a state of knowledge u_i dominates v_j. We say that u_i believably dominates (or not) v_i. The belief dominance is defined as follows:

Definition 3.7. *(Belief dominance) Let* $u_i \in U$*, and* $v_j \in V$ *be two knowledge states whose mass functions are* $m_{u_i}, m_{v_j} : 2^\Theta \to [0, 1]$ *respectively. Let* $u_i.a_k$ *and* $v_j.a_k$ *denote the part of knowledge states* u_i *and* v_j *respectively, defined on the attribute* a_k*.*
The knowledge state u_i *believably dominates* v_j *denoted by* $u_i \succ^{bl} v_j$ *if and only if* u_i *is believably as good or better than* v_j *in all attributes* a_k *in* $\mathcal{A}$ *(*$1 \leq k \leq d$*) and believably strictly better in at least one attribute* a_{k_0} *(*$1 \leq k_0 \leq d$*), i.e.,* $\forall a_k \in \mathcal{A} : u_i.a_k \leq^\forall v_j.a_k$ *and* $\exists a_{k_0} \in \mathcal{A} : u_i.a_{k_0} <^\forall v_j.a_{k_0}$

Table 3.4: Evidential Objects

Target	**Distance**	**Altitude**
U	$\langle\{90, 100\}, 0.7\rangle$, $\langle\{65\}, 0.3\rangle$	$\langle\{1, 2, 3\}, 0.9\rangle$, $\langle\{2\}, 0.1\rangle$
V	$\langle\{100\}, 1\rangle$	$\langle\{1, 2\}, 0.8\rangle$, $\langle\{3, 4\}, 0.2\rangle$

Example 3.9. *Assume we have the evidential objects of Table 3.4. Let's now derive the knowledge states of U and V:*
$u_1 = \langle\{90,100\},\{1,2,3\},0.7\times0.9=0.63\rangle$, $u_2 = \langle\{90,100\},\{2\},0.7\times0.1=0.07\rangle$
$u_3 = \langle\{65\},\{1,2,3\},0.3\times0.9=0.27\rangle$, $u_4 = \langle\{65\},\{2\},0.3\times0.1=0.03\rangle$
$v_1 = \langle\{100\},\{1,2\},0.8\rangle$, $v_2 = \langle\{100\},\{3,4\},0.2\rangle$
Applying definition 3.7, one can write:
$u_1 \not\succ^{bl} v_1$ *since* $\{1,2,3\} \not\leq^{\forall} \{1,2\}$. *Also,* $u_1 \not\succ^{bl} v_2$ *because* $\{90,100\} \leq^{\forall} \{100\}$ *and* $\{1,2,3\} \leq^{\forall} \{3,4\}$, *but* $\{90,100\} \not\leq^{\forall} \{100\}$ *and* $\{1,2,3\} \not\leq^{\forall} \{3,4\}$.
$u_2 \succ^{bl} v_2$ *since* $\{90,100\} \leq^{\forall} \{100\}$ *and* $\{2\} <^{\forall} \{3,4\}$
$u_3 \succ^{bl} v_2$ *since* $\{65\} <^{\forall} \{100\}$ *and* $\{1,2,3\} \leq^{\forall} \{3,4\}$
$u_4 \succ^{bl} v_2$ *since* $\{65\} <^{\forall} \{100\}$ *and* $\{2\} \leq^{\forall} \{3,4\}$.

Belief dominance between evidential objects

The belief dominance between two evidential objects is based on the belief dominance between their instances.

Definition 3.8. *Given two evidential objects U and V defined on a set of attributes $\mathcal{A}$. $\{u_1,u_2,\ldots,u_n\}$ and $\{v_1,v_2,\ldots,v_m\}$ represent the knowledge states of the objects U and V, respectively, defined on d attributes whose discernment frame is $\Theta = \prod_{1\leq k\leq d}\Theta_{a_k}$. Let $u_i \in U$, and $v_j \in V$ be two knowledge states whose mass functions are $m_{u_i}, m_{v_j} : 2^{\Theta} \to [0,1]$, respectively. The degree of belief that U believably dominates V is defined as follows:*

$$bel(U \succ V) = \sum_{i=1}^{n} m(u_i) \sum_{j:u_i \succ^{bl} v_j}^{m} m(v_j) \tag{3.5}$$

where $u_i \succ^{bl} v_j$ means that u_i believably dominates v_j

Example 3.10. *One can see that $bel(U \succ V) = 0.07\times0.2+0.27\times0.2+0.03\times0.2 = 0.074$. One can also easily check that: $bel(V \succ U) = 0.8\times(0+0+0+0)+0.2\times(0+0+0+0) = 0$. This means that the degree of belief that U dominates V is equal to 0.2 (while V does not dominate U at all).*

Modelling the dominance operator on the basis of the interpretations (instances) of the compared evidential objects seems of a great interest. Indeed, an efficient way to implement a database operator (here the dominance operator), is to manipulate the compact form of the database (such in example 2.1). Generating the whole candidate instances to

compare two evidential objects would produce an important requirement in terms of CPU computation and memory consumption. In this context, we introduce Theorem 3.1, to provide an alternative representation of the belief dominance operator, based on the compact form of the evidential objects. We'll show that the introduced formula is equivalent to definition 3.8.

Theorem 3.1.

$$bel(U \succ V) = \prod_{k=1}^{d} bel(U.a_k \leq V.a_k) - \prod_{k=1}^{d} bel(U.a_k \leq V.a_k \text{ and } U.a_k \not\leq V.a_k) \tag{3.6}$$

Proof.

$$\begin{aligned}
bel(U \succ V) &= \sum_{i=1}^{n} m(u_i) \sum_{j:u_i \succ^{bl} v_j}^{m} m(v_j) \\
&= \sum_{i=1}^{n} m(u_i) \sum_{j:(\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k \text{ and } \exists k, u_i.a_k < v_j.a_k)}^{m} m(v_j) \\
&= \sum_{i=1}^{n} m(u_i) \sum_{j:(\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k \text{ and not}(\forall k, u_i.a_k \not< v_j.a_k))}^{m} m(v_j) \\
&= \sum_{i=1}^{n} m(u_i) \left(\sum_{j:\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k} m(v_j) - \sum_{j:(\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k \text{ and } u_i.a_k \not< v_j.a_k)} m(v_j) \right) \\
&= \sum_{i=1}^{n} m(u_i) \sum_{j:\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k} m(v_j) - \sum_{i=1}^{n} m(u_i) \sum_{j:(\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k \text{ and } u_i.a_k \not< v_j.a_k)} m(v_j) \\
&= \sum_{i=1}^{n} \prod_{k=1}^{d} m_{ik}(u^{\Theta_k \uparrow \Theta}) \sum_{j:u_i \leq^{bl} v_j} \prod_{k=1}^{d} m_{jk}(v^{\Theta_k \uparrow \Theta}) - \\
&\sum_{i=1}^{n} \prod_{k=1}^{d} m_{ik}(u^{\Theta_k \uparrow \Theta}) \sum_{j=1}^{m} \prod_{\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k \text{ and } u_i.a_k \not< v_j.a_k}^{d} m_{jk}(v^{\Theta_k \uparrow \Theta}) \\
&= \prod_{k=1}^{d} \left(\sum_{i=1}^{n} m_{ik}(u^{\Theta_k \uparrow \Theta}) \sum_{j:u_i \leq^{bl} v_j} m_{jk}(v^{\Theta_k \uparrow \Theta}) \right) - \\
&\prod_{k=1}^{d} \left(\sum_{i=1}^{n} m_{ik}(u^{\Theta_k \uparrow \Theta}) \sum_{\forall k,\ u_i.a_k \leq^{\forall} v_j.a_k \text{ and } u_i.a_k \not<^{\exists} v_j.a_k} m_{jk}(v^{\Theta_k \uparrow \Theta}) \right) \\
&= \prod_{k=1}^{d} bel(U.a_k \leq V.a_k) - \prod_{k=1}^{d} bel(U.a_k \leq V.a_k \text{ and } U.a_k \not\leq V.a_k)
\end{aligned}$$

(3.7)

□

Table 3.5: Targets Observations

Target	**Distance**	**Altitude**
U	$\langle\{5,20\},0.1\rangle$, $\langle\{30\},0.9\rangle$	$\langle\{20,30\},0.1\rangle$, $\langle\{70\},0.9\rangle$
V	$\langle\{30\},0.1\rangle$ $\langle\{5\},0.9\rangle$	$\langle\{70\},0.1\rangle$, $\langle\{30\},0.9\rangle$

Illustration 3.1. *Consider the targets observations depicted in Table 3.5. We propose to compute the belief degree that U dominates V (Theorem 3.1).*
$\prod_{k=1}^{2} bel(U.a_k \leq V.a_k) - \prod_{k=1}^{2} bel(U.a_k \leq V.a_k \text{ and } U.a_k \not\leq V.a_k) =$
$((0.1 \times 0.1 + 0.9 \times 0.1) \times (0.1 \times (0.1 + 0.9) + 0.9 \times 0.1)) -$
$((0.9 \times 0.1) \times (0.1 \times 0.9 + 0.1 \times 0.9))$
=0.0028
Based on the definition 3.8, to compute the degree $bel(U \succ V)$, it is necessary to derive the knowledge states of U and V.
$U = \{u_1 = \{\{5,20\},\{20,30\},0.01\}, u_2 = \{\{5,20\},\{70\},0.09\},$
$u_3 = \{\{30\},\{20,30\},0.09\}, u_4 = \{\{30\},\{70\},0.81\}\}$
$V = \{v_1 = \{\{30\},\{70\},0.01\}, v_2 = \{\{30\},\{30\},0.09\}, v_3 = \{\{5\},\{70\},0.09\},$
$v_4 = \{\{5\},\{30\},0.81\}\}$
$bel(U \succ V) = 0.01 \times (0.01 + 0.09) + 0.09 \times 0.01 + 0.09 \times 0.01$
=0.0028

Definition 3.9. *(b-dominance) Given two objects $U, V \in \mathcal{O}$ and a belief threshold b, U believably b-dominates V, denoted by $U \succ_b^{bl} V$ if and only if $bel(U \succ V) \geq b$.*

Example 3.11. *In Table 3.4, $U \succ_{0.1}^{bl} V$ since $bel(U \succ V) = 0.2 \geq 0.1$, however, V does not believably 0.1-dominates U denoted by $V \not\succ_{0.1}^{bl} U$*

Belief Skyline

Definition 3.10. *(b-skyline) The belief skyline of the evidential databases, denoted by b-skyline, comprises those objects that are not b-dominated by any other object. Thus, we define the notion of b-skyline as follows:*
b-skyline $= \{U \in \mathcal{O} \mid \nexists V \in \mathcal{O}, V \succ_b^{bl} U\}$.

Proposition 3.1. *Given two belief thresholds b and b', if $b < b'$ then the b-skyline is a subset of the b'-skyline, i.e., $b < b' \Rightarrow b\text{-skyline} \subseteq b'\text{-skyline}$.*

Proof. Assume that there exists an object U such that $U \in b\text{-skyline}$ and $U \notin b'\text{-skyline}$. Since $U \notin b'\text{-skyline}$, there must exists another object, say V, that b'-dominates U. Thus, $bel(V \succ U) > b'$. But, $b < b'$. Therefore, $bel(V \succ U) > b$. Hence, $V \succ_b^{bl} U$, which leads to a contradiction as $U \in b$-skyline. □

Proposition 3.1 indicates that the b-skyline size is smaller than the b'-skyline size if $b < b'$. Roughly speaking, from proposition 3.1, we can see that users have a flexible tool to control the size of the retrieved evidential skyline by varying the belief threshold.

Table 3.6: Evidential database: Example of targets.

Target	Distance	Altitude
t_1	$\langle\{150, 160, 180\}, 0.1\rangle$, $\langle\{190, 200\}, 0.9\rangle$	$\langle 60, 0.3\rangle$, $\langle 100, 0.7\rangle$
t_2	$\langle 100, 0.7\rangle$ $\langle\Theta, 0.3\rangle$	$\langle\{70, 80\}, 0.8\rangle$, $\langle 80, 0.2\rangle$
t_3	$\langle\{50, 60\}, 0.8\rangle$, $\langle\{65\}, 0.2\rangle$	60
t_4	$\langle\{50\}, 0.5\rangle$, $\langle\{60\}, 0.5\rangle$	$\langle\{60\}, 0.6\rangle$, $\langle\{70\}, 0.4\rangle$

Example 3.12. *Consider the set of evidential objects presented in Table 3.6.* 0.5*-skyline* $= \{t_3, t_4\}$ *since* t_3 *and* t_4 *are not believably 0.5-dominated by any other objects in* $\mathcal{O}$*. However,* $\{t_1, t_2\} \notin 0.5$*-skyline since* $t_4 \succ_{0.88}^{bl} t_1$ *and* $t_3 \succ_{0.7}^{bl} t_2$*. The* 0.1*-skyline only contains the object* t_3 *(*0.1*-skyline* $= \{t_3\}$*),* t_4 *is discarded because it is* 0.16*-dominated by* t_3 *(*$t_3 \succ_{0.16}^{bl} t_4$*). Thus,* 0.1*-skyline* $\subseteq 0.5$*-skyline*

3.3.3 Plausible skyline oriented knowledge states

In this section, we define the plausible dominance operator between knowledge states. We then extend it to evidential objects. We define then the notion of the plausible skyline.

Plausible dominance between knowledge states

Given two evidential objects $U = \{u_1, u_2, \ldots, u_n\}$ and $V = (\{v_1, v_2, \ldots, v_m\}$ defined on d attributes which the joint frame of discernment is $\Theta = \prod_{k \leq d} \Theta_{a_k}$. To check if a state of knowledge u_i plausibly dominates v_j or not, we proceed as follows.

Definition 3.11. *(Plausible dominance) Let $u_i \in U$, and $v_j \in V$ be two knowledge states whose mass functions are $m_{u_i}, m_{v_j} : 2^\Theta \to [0,1]$, respectively. Let $u_i.a_k$ and $v_j.a_k$ denote the part of knowledge states u_i and v_j, respectively, defined on the attribute a_k.*
The knowledge state u_i plausibly dominates v_j denoted by $u_i \succ^{pl} v_j$ if and only if u_i is plausibly as good or better than v_j in all attributes a_k in $\mathcal{A}$ $(1 \leq k \leq d)$ and plausibly strictly better in at least one attribute a_{k_0} $(1 \leq k_0 \leq d)$, i.e., $\forall a_k \in \mathcal{A} : u_i.a_k \leq^\exists v_j.a_k$ and $\exists a_{k_0} \in \mathcal{A} : u_i.a_{k_0} <^\exists v_j.a_{k_0}$

Example 3.13. *Let us consider the Table 3.4 again. The knowledge states of the objects U and V are already derived in example 3.9. Applying definition 3.11, we can say that:*
$u_1 \succ^{pl} v_1$ since $\{90,100\} \leq^\exists \{100\}$ and $\{1,2,3\} <^\exists \{1,2\}$
$u_3 \succ^{pl} v_1$ since $\{65\} <^\exists \{100\}$ and $\{1,2,3\} \leq^\exists \{1,2\}$
$u_4 \succ^{pl} v_1$ since $\{65\} <^\exists \{100\}$ and $\{2\} \leq^\exists \{1,2\}$

Plausible dominance between evidential objects

Definition 3.12. *Given two evidential objects U and V defined on a set of attributes $\mathcal{A}$. $\{u_1, u_2, \ldots, u_n\}$ and $\{v_1, v_2, \ldots, v_m\}$ represent the knowledge states of the objects U and V, respectively, defined on d attributes whose discernment frame is $\Theta = \prod_{1 \leq k \leq d} \Theta_{a_k}$. Let $u_i \in U$, and $v_j \in V$ be two knowledge states whose mass functions are $m_{u_i}, m_{v_j} : 2^\Theta \to [0,1]$, respectively. The degree of plausibility that U plausibly dominates V is defined as follows:*

$$pl(U \succ V) = \sum_{i=1}^{n} m(u_i) \sum_{j : u_i \succ^{pl} v_j}^{m} m(v_j) \tag{3.8}$$

where $u_i \succ^{pl} v_j$ means that u_i plausibly dominates v_j

Theorem 3.2.

$$pl(U \succ V) = \prod_{k=1}^{d} pl(U.a_k \leq V.a_k) - \prod_{k=1}^{d} pl(U.a_k \leq V.a_k \text{ and } U.a_k \not\leq V.a_k) \tag{3.9}$$

Proof.

$$
\begin{aligned}
pl(U \succ V) &= \sum_{i=1}^{n} m(u_i) \sum_{j:u_i \succ^{pl} v_j}^{m} m(v_j) \\
&= \sum_{i=1}^{n} m(u_i) \sum_{j:(\exists k,\ u_i.a_k \leq^{\exists} v_j.a_k \text{ and } \exists k, u_i.a_k < v_j.a_k)}^{m} m(v_j) \\
&= \sum_{i=1}^{n} m(u_i) \sum_{j:(\exists k,\ u_i.a_k \leq^{\exists} v_j.a_k \text{ and not}(\forall k, u_i.a_k \not< v_j.a_k))}^{m} m(v_j) \\
&= \sum_{i=1}^{n} m(u_i) \left(\sum_{(j:\exists k,\ u_i.a_k \leq^{\exists} v_j.a_k)} m(v_j) - \sum_{j:(\exists k,\ u_i.a_k \leq^{\exists} v_j.a_k \text{ and } \forall k,\ u_i.a_k \not< v_j.a_k)} m(v_j) \right) \\
&= \sum_{i=1}^{n} m(u_i) \sum_{(j:\exists k,\ u_i.a_k \leq^{\exists} v_j.a_k)} m(v_j) - \sum_{i=1}^{n} m(u_i) \sum_{(j:\forall k,\ u_i.a_k \not< v_j.a_k)} m(v_j) \\
&= \sum_{i=1}^{n} \prod_{k=1}^{d} m_{ik}(u^{\Theta_k \uparrow \Theta}) \sum_{(j:\exists k,\ u^{\Theta_k \uparrow \Theta}.a_k \leq^{\exists} v^{\Theta_k \uparrow \Theta}.a_k)} \prod_{k=1}^{d} m_{jk}(v^{\Theta_k \uparrow \Theta}) - \\
&\sum_{i=1}^{n} \prod_{k=1}^{d} m_{ik}(u^{\Theta_k \uparrow \Theta}) \sum_{j=1}^{m} \prod_{\forall k,\ u^{\Theta_k \uparrow \Theta}.a_k \not<^{\forall} v^{\Theta_k \uparrow \Theta}.a_k}^{d} m_{jk}(v^{\Theta_k \uparrow \Theta}) \\
&= \prod_{k=1}^{d} pl(U.a_k \leq V.a_k) - \prod_{k=1}^{d} pl(U.a_k \leq V.a_k \text{ and } U.a_k \not< V.a_k)
\end{aligned}
\tag{3.10}
$$

□

Example 3.14. *Le us consider the evidential database example shown in Table 3.5. We propose to compute the plausibility that U dominates V according to Theorem 3.2.*
$pl(U \succ V) = \prod_k^2 pl(U.a_k \leq V.a_k) - \prod_k^2 pl(U.a_k \leq V.a_k \text{ and } U.a_k \not< V.a_k) = 0.0361 - 0.0162 = 0.0199$
Using the definition 3.12, that consists in comparing the instances of U and V, we obtain the same result:
$U = \{u_1 = \{\{5, 20\}, \{20, 30\}, 0.01\}, u_2 = \{\{5, 20\}, \{70\}, 0.09\},$
$u_3 = \{\{30\}, \{20, 30\}, 0.09\}, u_4 = \{\{30\}, \{70\}, 0.81\}\}$
$V = \{v_1 = \{\{30\}, \{70\}, 0.01\}, v_2 = \{\{30\}, \{30\}, 0.09\}, v_3 = \{\{5\}, \{70\}, 0.09\},$
$v_4 = \{\{5\}, \{30\}, 0.81\}\}$
$pl(U \succ V) = 0.01 \times (0.01 + 0.09 + 0.09 + 0.81) + 0.09 \times (0.01) + 0.09 \times (0.01 + 0.09) = 0.0199$

Definition 3.13. *(p-dominance) Given two objects $U, V \in \mathcal{O}$ and a plausible threshold p, U plausibly dominates V, denoted by $U \succ_p^{pl} V$ according to the threshold p if and only if $pl(U \succ V) \geq p$.*

Example 3.15. *As we can see in example 3.5, $U \not\succ_{0.9}^{pl} V$ since $pl(U \succ V) = 0.0199$, however, V plausibly 0.9-dominates U since $pl(V \succ U) = 0.9639 > 0.9$*

Plausible skyline

Definition 3.14. *(p-Skyline) The plausible skyline is the set of evidential objects that are not plausibly dominated by any other objects in $\mathcal{O}$, we define the notion of p-skyline as follows:*
p-skyline$= \{U \in \mathcal{O} \mid \nexists V \in \mathcal{O}, V \succ_p^{pl} U\}$.

Proposition 3.2. *Given two plausible thresholds p and p', if $p < p'$ then the p-skyline is a subset of the p'-skyline, i.e., $p < p' \Rightarrow$ p-skyline $\subseteq p'$-skyline.*

Proof. Assume that there exists an object U such that $U \in$ *p-skyline* and $U \notin p'$*-skyline*. Since $U \notin p'$*-skyline*, there must exist another object, say V, that p'-dominates U. Thus, $pl(V \succ U) > p'$. But, $p < p'$. Therefore, $pl(V \succ U) > p$. Hence, $V \succ_p^{pl} U$, which leads to a contradiction as $U \in p$-skyline. □

3.3.4 Analysis of the evidential dominance

In this part, we introduce several fundamental properties of dominance relationship between imperfect objects modeled by the evidence theory. These properties are used later to efficiently perform the main operations on evidential skyline computation, essentially, to minimize the dominance checks.

Table 3.7: Evidential data example.

Target	Distance	Altitude
t_x	$\langle\{90, 120\}, 0.7\rangle$, $\langle\{150, 160\}, 0.3\rangle$	$\langle 80, 0.1\rangle$, $\langle\{90, 100\}, 0.9\rangle$
t_y	$\langle\{170, 180\}, 1\rangle$	$\langle\{60, 70\}, 0.6\rangle$, $\langle 100, 0.4\rangle$
t_z	$\langle\{100, 110\}, 0.5\rangle$, $\langle\{190, 200\}, 0.5\rangle$	$\langle 70, 0.3\rangle$, $\langle\{80, 90\}, 0.7\rangle$

Property 3.6. *The b-dominance relationship does not satisfy the property of transitivity.*

Illustration 3.2. *Consider the objects depicted in Table 3.7. We have,* $bel(t_x \succ t_y) = 0.4$, $bel(t_y \succ t_z) = 0.3$ *and* $bel(t_x \succ t_z) = 0.035$. *Observe that,* $t_x \succ^{bl}_{0.4} t_y$ *and* $t_y \succ^{bl}_{0.3} t_z$, *but* t_x *does not 0.3-dominate* t_z. *Thus, the b-dominance relationship is not transitive.*

Property 3.7. *The p-dominance relationship does not satisfy the property of transitivity.*

Illustration 3.3. *Consider the objects depicted in Table 3.7. We have,* $pl(t_y \succ t_z) = 0.3$, $pl(t_z \succ t_x) = 0.5$ *and* $pl(t_y \succ t_x) = 0$. *Observe that,* $t_y \succ^{pl}_{0.3} t_z$ *and* $t_z \succ^{pl}_{0.5} t_x$, *but* t_y *does not dominate* t_x *at all. Thus, the p-dominance relationship is not transitive.*

Given an object U, we denote by $U.a_k^-$ and by $U.a_k^+$ respectively the minimum value and the maximum value of $U.a_k$, i.e, its *bba* defined on attribute a_k. For example, in Table 3.7, $t_z.d^- = 100$ and $t_z.d^+ = 200$

Property 3.8. *if* $\exists a_{k_l} \in \mathcal{A}$ *such that* $V.a_{k_l}^+ < U.a_{k_l}^-$ *then* $bel(U \succ V) = 0$ *since* $bel(U.a_{k_l} \leq V.a_{k_l}) = 0$.

Property 3.9. *if* $\exists a_{k_l} \in \mathcal{A}$ *such that* $V.a_{k_l}^+ < U.a_{k_l}^-$ *then* $pl(U \succ V) = 0$ *since* $pl(U.a_{k_l} \leq V.a_{k_l}) = 0$.

Property 3.10. *Let* $U.a_k$ *and* $V.a_k$ *be two bba of the evidential objects* U *and* V, *respectively, defined on attribute* a_k. *Let* b *be a threshold.*
If $\exists a_k \in \mathcal{A}$ *such that* $bel(U.a_k \leq V.a_k) < b$ *then* U *does not b-dominate* V.

Proof. We have $bel(U.a_k \leq V.a_k) = x_k \in [0, 1]$. Suppose $\exists k_0 \in \{1, d\}/x_{k_0} < b$ and $\forall$ attribute $a_k \in \mathcal{A} \setminus \{a_{k_0}\}$, $x_k = 1$. Then $bel(U \succ V) = \prod_{k=1}^{d} x_k = x_1 \times x_2 \times \cdots \times x_{k_0} \times \cdots \times x_d = x_{k_0} \times 1 = x_{k_0} < b$ □

3.4 Evidential Skyline Computation

In this section, we develop efficient algorithms to tackle the problem of evidential skyline computation. In particular, we provide appropriate comparison methods between two evidential objects through reducing the number of dominance checks and hence improving the performance of the proposed algorithms.

3.4.1 Belief skyline computation

A straightforward algorithm to compute the belief skyline (denoted by BBS) is to compare each object U against the other objects. If U is not b-dominated, then it belongs to the

b-skyline. However, this approach results in a high computational cost (see Section 3.5) as it needs to compare each object with every other object in $\mathcal{O}$.

Also, while the b-dominance relationship is not transitive (see Property 3.6), an object cannot be eliminated even if it is b-dominated since it will be probably useful to eliminate other objects.
For this reason, we propose the algorithm 2 that follows the principle of the two scans algorithm (Chan et al., 2006).

Our proposed algorithm, named BS, computes the belief skyline through two phases. In the first phase (lines 2–13), a set of candidate objects b-skyline is selected by comparing each object U in $\mathcal{O}$ with those selected in b-skyline. If an object V in b-skyline is b-dominated by U, then V is removed from the set of candidate objects as it is not part of the believable skyline. At the end of the comparison of U with objects of b-skyline, if U is not b-dominated by any object then, it is added to b-skyline as a candidate object. After this first phase, the b-skyline comprises a set of objects that may be part of the b-skyline.

To avoid the situation illustrated by the example in Table 3.7, a second phase is needed (lines 14–17). To determine if an object U in the set b-skyline is indeed a skyline point, it is sufficient to compare U with those in $\mathcal{O} \setminus \{b\text{-skyline} \cup undom(U) \cup \{U\}\}$ that occur earlier than U since the other ones have been already compared against U, where $undom(U)$ is the set of objects that occur before U and that do not b-dominate V. This set is computed in the first phase in order to reduce the dominance checks in the second phase.

Algorithm 2: Belief Skyline BS

Input: Objects $\mathcal{O}$; belief threshold b;
Output: Belief skyline b-skyline;

```
1  begin
2      foreach U in O do
3          isSkyline ← true;
4          foreach V in b-skyline do
5              if isSkyline then
6                  if V ≻_b^bl U then
7                      isSkyline ← false;
8                  else
9                      undom(U) ← undom(U) ∪ {V};
10             if U ≻_b^bl V then
11                 remove V from b-skyline;
12         if isSkyline then
13             insert U in b-skyline;
14     foreach U in b-skyline do
15         foreach V in O \ (b-skyline∪undom(U) ∪ {U}), pos(V) < pos(U) do
16             if V ≻_b^bl U then
17                 remove U from b-skyline;
18     return b-skyline
```

The algorithm denoted BS has a quadratic time complexity, i.e, $T(n) = O(n^2)$.

Algorithm 3: b-dominates(U, V, b)

Input: Objects U,V; belief threshold b;

1 **foreach** a_k *in* $\mathcal{A}$ **do**
2 **if** $U.a_k^- > V.a_k^+$ **then**
3 **return** false;
4 $Dom \leftarrow 0, bel \leftarrow 1, eqbel \leftarrow 1$;
5 **foreach** a_k *in* $\mathcal{A}$ **do**
6 **if** $U.a_k^+ \geq V.a_k^-$ **then**
7 $bel \leftarrow bel \times bel(U.a_k \leq V.a_k)$;
8 **if** $bel < b$ **then**
9 **return** false;
10 **foreach** a_k *in* $\mathcal{A}$ **do**
11 $eqbel \leftarrow eqbel \times bel(U.a_k \leq V.a_k$ and $U.a_k \not\prec V.a_k)$;
12 $Dom \leftarrow bel - eqbel$;
13 **if** $Dom > b$ **then**
14 **return** true;

Even if BS minimizes the number of dominance checks, it also may result in a high computational cost, in particular, when the average number of focal elements and the number of attributes per object are large. Thus, it is crucial to optimize the operator $bel(A \leq B)$ (line 7 in algorithm 3) in order to reduce the dominance checks and improve the performance of the BS algorithm. We devise an efficient method that overcomes this problem using the minimum and maximum values of each *bba* w.r.t. each attribute, according to Property 3.8. The method b-dominates(U, V, b) denoted by $\succ_b^{bl}$ in the algorithm 2 (line 6) is detailed in algorithm 3. Here after some details about this method.

To reduce the complexity of the knowledge states dominance computation, we essentially rely on Theorem 3.1. For each attribute $a_k \in \mathcal{A}$, if the condition $U.a_k^- > V.a_k^+$ holds, then we are sure that U could not dominate V (Property 3.8). The function returns false (line 3). If this first scan ends without satisfying the above-mentioned condition, then we do a second scan (line 5). For each attribute, if $U.a_k^+ < V.a_k^-$ is true, then $bel(U.a_k \leq V.a_k) = 1$. In this case, we should not modify the value *bel* (already initialized to 1 in line 4). The contrary case, however, should be processed (i.e., when $U.a_k^+ \geq V.a_k^-$ is true) by multiplying *bel* by $bel(U.a_k \leq V.a_k)$. If the obtained *bel* does not reach the threshold b (line 8), then it is useless to continue, and the function returns false (according to Property 3.10). In lines 10 to 12, we compute the dominance degree following Theorem 3.1 and check if it exceeds the threshold b (line 13).

3.4.2 Plausible skyline computation

To compute the plausible skyline, we refer to the PS algorithm presented in algorithm 4, in which we follow the same steps as BS algorithm using an efficient method to compute the p-dominance between objects in $\mathcal{O}$.

Algorithm 4: Plausible Skyline PS

Input: Objects $\mathcal{O}$; plausible threshold p;
Output: p-skyline;

```
1  begin
2      foreach o_i in O do
3          isSkyline ← true;
4          foreach o_j in p-skyline do
5              if isSkyline then
6                  if o_j ≻_p^pl o_i then
7                      isSkyline ← false;
8                  else
9                      undom(o_i) ← undom(o_i) ∪ {o_j};
10             if o_i ≻_p^pl o_j then
11                 remove o_j from p-skyline;
12         if isSkyline then
13             insert o_i in p-skyline;
14     foreach o_i in p-skyline do
15         foreach o_j in O \ (p-skyline ∪ undom(o_i) ∪ {o_i}), pos(o_j) < pos(o_i) do
16             if o_j ≻_p^pl o_i then
17                 remove o_i from p-skyline;
18     return p-skyline;
```

The algorithm denoted PS has a quadratic time complexity, i.e, $T(n) = O(n^2)$.

In fact, in algorithm 4, the p-dominance method denoted by $\succ_p^{pl}$ can recall the principle of the b-dominance represented in algorithm 3.

Algorithm 5: *p-dominates*(U, V, p)

Input: Objects U,V; plausible threshold p;

```
1  foreach a_k in A do
2      if U.a_k^- > V.a_k^+ then
3          return false;
4  pDom ← 0, pl ← 1, eqpl ← 1;
5  foreach a_k in A do
6      if U.a_k^+ ≥ V.a_k^- then
7          pl ← pl × pl(U.a_k ≤ V.a_k);
8          if pl < p then
9              return false;
10 foreach a_k in A do
11     eqpl ← eqpl × pl(U.a_k ≤ V.a_k and U.a_k ≰ V.a_k);
12 pDom ← pl − eqpl;
13 if pDom > p then
14     return true;
```

To reduce the complexity of the knowledge states dominance computation, we used Theorem 3.2. Algorithm 5 computes the *p-dominance*, exactly in the similar way of algorithm 3. The only difference is in line 7 where we compute the plausibility of the attributes' comparison, instead of the belief. The pruning techniques are exactly the same.

3.5 Experimental Evaluation

In this section, we have performed an extensive experimental evaluation of the proposed framework. We report empirical study to examine the effectiveness and the efficiency of the evidential skyline analysis on uncertain data modeled by the evidence theory. More specifically, we focus on two issues: (i) the size of the evidential skyline; we show how the p-skyline is more significant than the b-skyline and (ii) the scalability of our proposed techniques for computing the evidential skyline. We implemented two efficient functions, i.e., b-dominates(U,V,b) and p-dominates(U,V,p), to show how perform the b-skyline and the p-skyline algorithms.

For comparison purpose, we implemented the baseline algorithms for computing the b-skyline and the p-skyline referred to as BBS (basic b-skyline) and BPS (basic p-skyline), respectively. BBS and BPS do not use the two phases algorithm, and compare each object

U in $\mathcal{O}$ against all the other objects in $\mathcal{O}$. In addition, BBS and BPS algorithms do not use the methods b-dominates() and p-dominates() and iterate all propositions in a focal element to compute the dominance degrees between objects.

3.5.1 Experimental Setup

The generation of evidential databases is controlled by the parameters in Table 3.8. It lists their notations, range of examined values, and default values. In each experimental setup, we investigate the effect of one parameter, while we set the remaining to their default values. For the data generation, we have considered that an evidential object is an ArrayList of *bba* defined on d dimensions. Each *bba* is represented by an ArrayList of focal elements f whose number is a random value in [1..Max Nbr of focal elmts]. A focal element is a set of random values to which we attribute a mass function.

The data generator and the algorithms, i.e., BBS, BPS, BS and PS were implemented in Java, and all experiments were conducted on a 2.3 GHz Intel Core i7 processor, with 6GB of RAM.

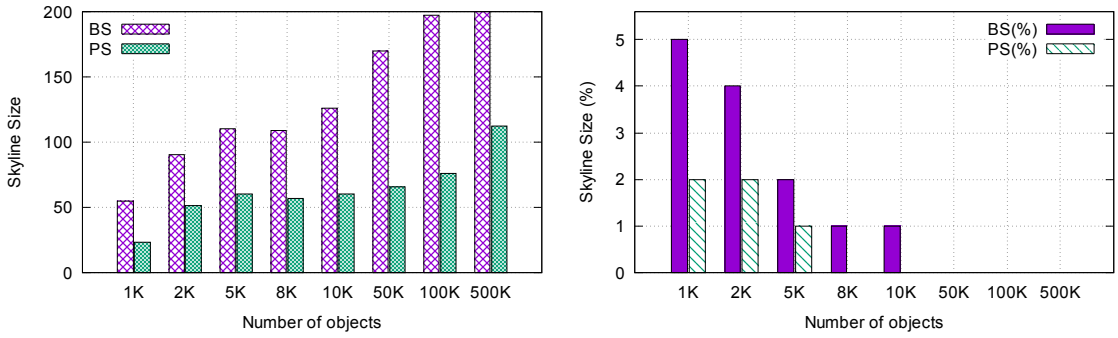
Table 3.8: Parameters for the Skyline Computation.

Parameter	Symbol	Values	Default
Number of objects	n	1K, 2K, 5K, 8K, 10K, 50K, 100K, 500K	10K
Number of attributes	d	2, 3, 4, 5, 6	3
Max Nbr of focal elmts/attr	f	2, 3, 4, 6, 8, 9, 10, 11	5
belief threshold	b	0.01, 0.1, 0.3, 0.5, 0.7, 0.9	0.5
plausible threshold	p	0.5, 0.6, 0.7, 0.8, 0.9	0.7
Theta cardinality/attr	t	10, 50, 100, 150, 200	100

3.5.2 Skyline result size

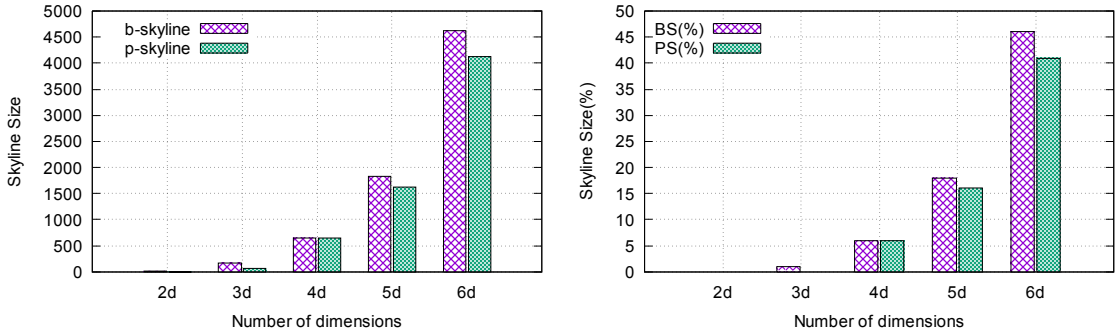
Figures 3.1, 3.2, 3.3 and 3.4 show the b-skyline size (i.e., the number of objects returned in the skyline set) and the p-skyline size w.r.t. n, d, b, p, t and f. Figures 3.1b, 3.2b present the percentage of objects returned by the belief skyline BS and the plausible skyline PS algorithms while varying the parameters in question.

Figure 3.1a shows that the size on the belief and the plausible skyline increases with



(a) Effect of n (b) Effect of $n(\%)$

Figure 3.1: Skyline Size with varying n



(a) Effect of d (b) Effect of $d(\%)$

Figure 3.2: Skyline Size with varying d

higher n since when n increases, more objects have chances to be not dominated. As shown in Figure 3.2a the cardinality of the b-skyline as well as the p-skyline, significantly increases with the increase of d. This is because with the increase of d, an object has better opportunity to be not dominated in all attributes. Figure 3.3 shows that the size of the belief skyline increases with the increase of b since the b-skyline includes the b'-skyline if $b > b'$; (see Proposition 3.1). Figure 3.3 also shows that size of the plausible skyline increases with the increase of p since the p-skyline contains the p'-skyline if $p > p'$; (see Proposition 3.2). Similarly to n, d, and b, as shown in Figure 3.4a, increasing f results in a high number of knowledge states which decreases the object chance to be dominated and thus to be in the skyline.

We also varied the theta cardinality for each attribute, i.e., number of possible propositions for each attribute. In contrast to n, b, p, f and d, the size of the evidential skyline is not affected by the parameter t, as shown in Figure 3.4b.

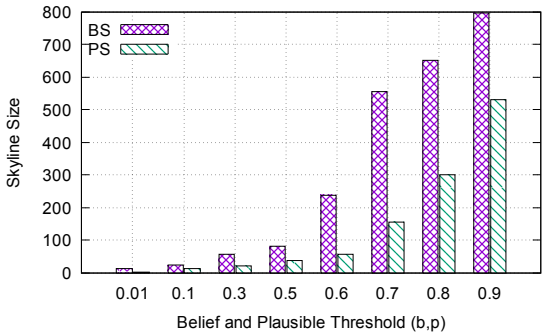
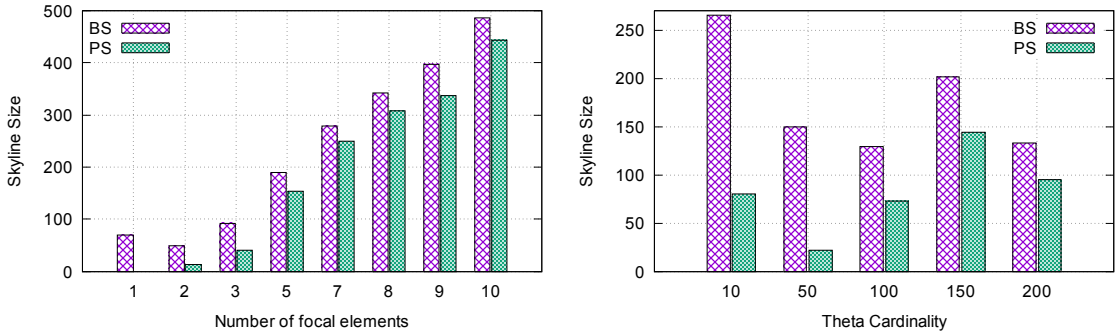


Figure 3.3: Skyline Size with varying the thresholds b and p



(a) Effect of f (b) Effect of t

Figure 3.4: Skyline Size with varying f and t

We can clearly remark that using the p-skyline is more significant than the b-skyline since for all figures, with varying all the parameters, the p-skyline size is significantly smaller than the b-skyline size which can helps more and more the decision maker. In the following, we explain how can the p-skyline be more significant.
Suppose we have four evidential objects in $\mathcal{O}$ such that $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$. We study the belief degrees and the plausibility degrees that an object in $\mathcal{O}$ dominates the others. Table 3.9 shows the belief degrees that each object in lines dominates another object in columns. Table 3.10 shows the plausibility degrees that each object in lines dominates another object in columns. As we can observe, the degrees returned by the plausible dominance are either equal or larger than those returned by the belief dominance. That is because, as we have shown above, the plausibility function gives more chance to objects to dominate the others. For example, o_1 0.2-believably dominates o_2, however, it 0.3-plausibly dominates o_2. Let us compute the skyline objects with the following belief and plausible thresholds: $b = p = 0.3$
In Table 3.9, o_1 cannot be in the b-skyline since $o_2 \succ_{0.4} o_1$. However, o_2, o_3 and o_4 are not 0.3-dominated by any other object. Thus, b-skyline $= \{o_2, o_3, o_4\}$.

In Table 3.10, o_2 cannot be in the p-skyline since $o_3 \succ_{0.5} o_2$ and $o_4 \succ_{0.4} o_2$. As well as o_4 can not be in the p-skyline since $o_1 \succ_{0.4} o_4$, $o_2 \succ_{0.4} o_4$ and $o_3 \succ_{0.5} o_4$. Thus, the p-skyline only contains o_3 since it is not plausibly dominated by any other object with p=0.3 and the p-skyline= $\{o_3\}$.

Table 3.9: Belief Dominance

Objects	o_1	o_2	o_3	o_4
o_1	-	0.2	0.2	0.3
o_2	0.4	-	0.1	0.3
o_3	0.1	0.3	-	0.2
o_4	0.1	0.1	0.15	-

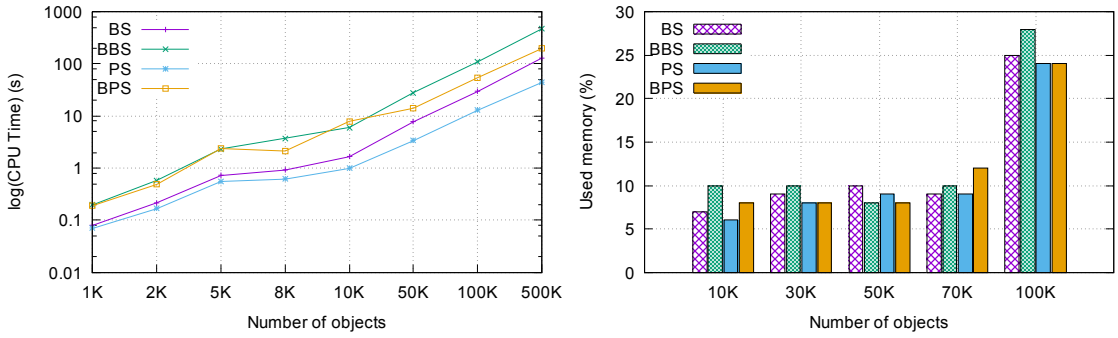
Table 3.10: Plausible Dominance

Objects	o_1	o_2	o_3	o_4
o_1	-	0.3	0.25	0.4
o_2	0.4	-	0.2	0.4
o_3	0.2	0.5	-	0.5
o_4	0.35	0.4	0.15	-

3.5.3 Scalability

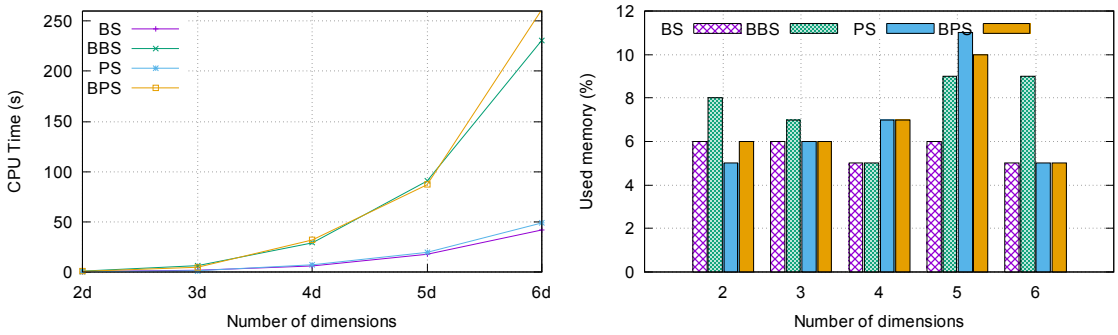
In this subsection, we first show that the b-skyline algorithm BS is more scalable than the basic b-skyline algorithm BBS, and then. In addition, the p-skyline PS algorithm is more scalable than the basic p-skyline algorithm BPS and even than BS.

Figures 3.5, 3.6, 3.7 and 3.8 depict the execution time and the used memory percentage of the implemented algorithms with respect to various parameters (n, d, b, p, t and f). Overall, BS outperforms BBS. More specifically, BS is faster than BBS, which in turn is faster than Basic p-BPS. As expected, Figure 3.5 shows that the performance of all the algorithms decreases with the increase of n. Observe that PS is one order of magnitude faster than BS. In fact, it can quickly identify if an object is plausibly dominated or not. In addition, although the BS algorithm contains more dominance checks than PS algorithm, it outperforms the BBS thanks to properties already presented in section 3.3. In contrast



(a) Effect of n (b) Effect of n

Figure 3.5: Skyline queries with varying n. a. CPU time, b. Used memory (%)



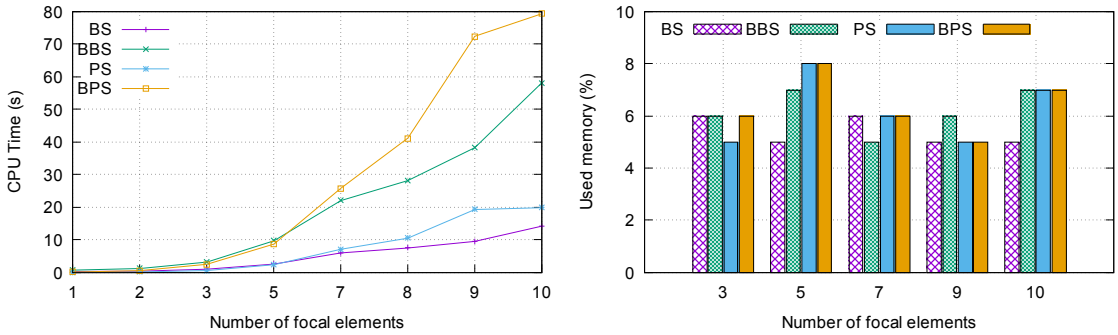
(a) Effect of d (b) Effect of d

Figure 3.6: Skyline queries with varying d. a. CPU time, b. Used memory (%)

to CPU time, varying all parameters has no apparent effect on used memory as shown in Figures 3.5b, 3.6b and 3.7b.

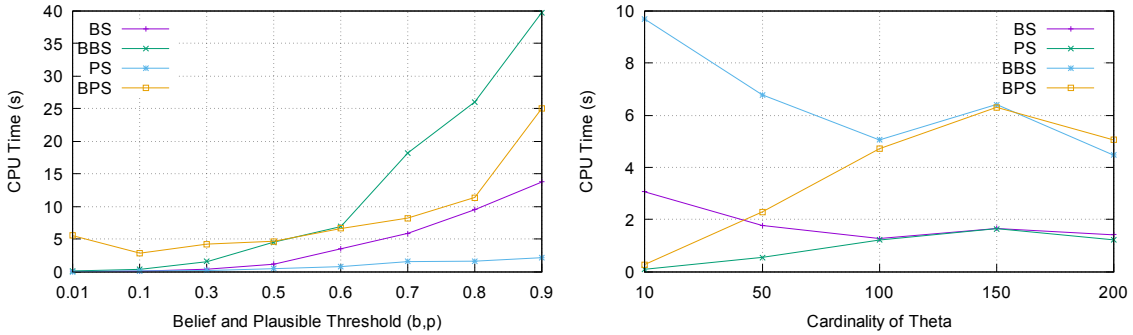
As shown in Figure 3.6, the running time of the algorithms increases when the number of dimensions increases. That is because when d increases, the number of knowledge states increases which explains the increase of dominance checks. BBS and BPS perform a large number of dominance checks. Even if BS performs the same number of dominance checks than BBS, BS is more efficient than BBS since it can detect immediately whether an object does not dominate another. The same logic is used to compare PS and BPS performances. We can note that there is a slight difference in term of CPU time between BS and PS. In addition, varying d has no effect on used memory (Figure 3.6b).

Figure 3.7 shows that, with the increase of f, BS algorithm outperforms the BBS (see Figure 3.7a). The same observation is valid for PS and BPS; PS is more than one order of magnitude faster than BPS. In addition, observe there is a slight difference between BS



(a) Effect of f (b) Effect of f

Figure 3.7: Skyline queries with varying f. a. CPU time, b. Used memory (%)



(a) Effect of b and p (b) Effect of t

Figure 3.8: Skyline queries with varying b, p and t.

and PS algorithms. As shown in Figure 3.8a, the more b and p increase, the more the computation cost increases. In this case, the dominance between objects takes more time, the number of skyline objects increases and consequently the number of dominance checks increases.

3.6 Conclusion

In this chapter, we extended the well-known skyline analysis to imperfect data modeled by the evidence theory. We proposed two main models: a first evidential skyline model which is based on the classic skyline definition and, the evidential skyline oriented knowledge states.

The key concept of this second extension is the states of knowledge derived from each

imperfect object that represents the real world. We explored the belief skyline framework as well as the plausible skyline one. We studied the properties of the belief dominance as well as the plausible dominance and provided efficient methods for performing the dominance checks using some pruning techniques.

We also developed two efficient algorithms to tackle the problem of the evidential skyline computation. On the other hand, we demonstrated the effectiveness of the evidential skyline and the efficiency and scalability of our algorithms.

In the next chapter, we address the following problems:

- Based on the skyline query over centralised imperfect data where imperfection is modeled by the evidence theory, we propose to efficiently compute the global skyline from distributed local sites.
- The maintenance of the skyline objects of frequently updated evidential databases. In particular, we propose algorithms for maintaining evidential skyline in the case of object insertion or deletion.

Chapter 4

Two Variations of the Evidential Skyline

Contents

4.1 Introduction

With the fast development of computing infrastructures and easily available network services, data management and storage have become inevitably more distributed (Hevner & Yao, 1979; Park, Min, & Shim, 2013). Dealing with skyline analysis over the distributed environments faces important challenges. Interestingly enough, many real-life applications where uncertain, imprecise, noisy and error-prone data inherently exist, are distributed, e.g., multiple geographic data sources in road networks (Deng, Zhou, & Tao, 2007), distributed sensor networks with imprecise measurements (Deshpande, Guestrin, Madden, Hellerstein, & Hong, 2004), etc. Distributed skyline computation over uncertain data has received an increasing attention (Endres & Kieling, 2015). Ding et al. (Ding & Jin, 2010) studied the skyline query on probabilistic data from multiple and distributed environments. In (Li et al., 2009), authors were interested in the fundamental problem of retrieving the global top-k objects from uncertain distributed data with minimum communication cost. Despite of the fact that the two research lines often arise concurrently in many applications, uncertainty modeled by the evidence theory and distributed skyline query processing, have been studied separately.

In section 4.3, we first recall the skyline semantic over evidential data and then, we study the problem of distributed skyline queries over imperfect data where imperfection is modeled by the evidence theory.

This contribution is published in the 28th IEEE International Conference on Tools with Artificial Intelligence(Elmi, Tobji, Hadjali, & Yaghlane, 2016a).

The skyline maintenance is not an easy task when the queried database is updated and unfortunately, there exist not much works that can handle skyline queries under database updates. The maintenance of the skyline set is then very useful since it allows users to get informed about the new interesting objects.

Let us mention the work done in (Papadias, Tao, Fu, & Seeger, 2005; Wu, Agrawal, Egecioglu, & Abbadi, 2007) where the authors introduce an optimal skyline deletion maintenance for certain data. Xia et al. (Xia & Zhang, 2006) present efficient update algorithms for compressed skyline Cubes. Closely relating to the maintenance of skyline results, let us point out the study related to the progressive skyline query evaluation and maintenance done in (Zhenjie et al., 2009). However, up to our knowledge, there is no work about the skyline maintenance issue in the uncertain/evidential databases context.

In section 4.4, we address the problem of the maintenance of the skyline objects of frequently updated evidential databases. In particular, we propose algorithms for maintaining evidential skyline in the case of object insertion or deletion.

This work is published in the 16th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (Elmi, Tobji, Hadjali, & Yaghlane, 2016b).

In the next section (section 4.2), we define the notion of the marginal points which are used to prune the research space for the distributed skyline computing and the skyline maintenance computation.

4.2 Marginal Points: Ideal Point and Header Point

An Ideal Point IP and a Header Point HP summarize the region of data explored in earlier iterations required to compute the evidential skyline. This approach is used in (Elmi, Tobji, et al., 2016b) to efficiently maintain the evidential skyline if a newly object is inserted in the evidential database $\mathcal{O}_i$, and in (Elmi, Tobji, et al., 2016a) to prune the research space for the distributed skyline computation. Let us first present the local marginal points.

The Ideal Point is a virtual object having the most interesting values across all attributes.

Definition 4.1. *(Ideal Point) Let b-$SKY(\mathcal{O}_i) = \{s_1, s_2, \ldots, s_u\}$ be the set of objects being in a local believable skyline. An Ideal Point IP of b-$SKY(\mathcal{O}_i)$ is a certain object defined such as: $< MIN(val.a_1), MIN(val.a_2), \ldots, MIN(val.a_d) >, \forall s_j \in$ b-$SKY(\mathcal{O}_i)$ where $MIN(val.a_k)$ is the function which returns the minimum certain value defined on the attribute a_k and $val.a_k$ is the set of distinct values defined on a_k.*

Skyline Objects s_j	Distance	Altitude
s_1	$\langle\{50, 51\}0.7\rangle$, $\langle\{55, 56\}0.3\rangle$	$\underline{20}$
s_2	$\{40, 41\}$	$\langle\{30\}0.9\rangle$, $\langle\{31\}0.1\rangle$
s_3	$\{\underline{10}, 11\}$	$\{60\}$

Table 4.1: 0.4-$SKY(\mathcal{O}_i)$

Example 4.1. *Suppose we have the* 0.4-$SKY(\mathcal{O}_i)$ *objects presented in Table 4.1. The most interesting values defined on distance and altitude attributes, are returned by the MIN() function and appear in an underlined form in Table 4.1. Thus, the Ideal Point is* $IP(10, 20)$.

Let $\mathcal{O}^* \subseteq SKY(\mathcal{O}_i)$ be the set of local skyline points having the most interesting values in one or more attributes, i.e, $\mathcal{O}_i^* = \{s \in SKY(\mathcal{O}_i) / \exists item \in s.a_k, item = MIN(val.a_k)\}$

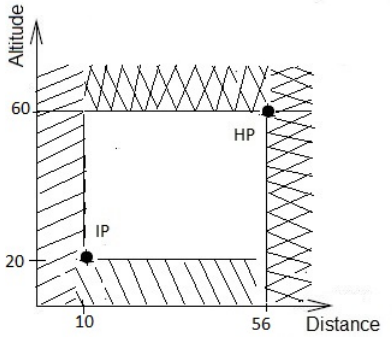


Figure 4.1: Marginal Points

where $val.a_k$ is the set of distinct values in attribute a_k occurring in the skyline and *item* is a single proposition in the *bba* "$s.a_k$".

Example 4.2. *In Table 4.1, $\mathcal{O}_i^*$ comprises the local skyline objects s_1 and s_3 since they have the interesting values* 10 *and* 20 *defined respectively on distance and altitude.*

Definition 4.2. *(Header Point) Let $\mathcal{O}_i^*$ be the set of local skyline points having the most interesting values. A Header Point HP of $SKY(\mathcal{O}_i) = \{s_1, s_2, ..., s_u\}$ is a certain virtual object defined such as:*
$\forall s_j \in \mathcal{O}_i^*$, such that, $< MAX(val.a_1), MAX(val.a_2), \ldots, MAX(val.a_d) >$.

Example 4.3. *One can check that in Table 4.1, $\mathcal{O}_i^* = \{s_1, s_3\}$. The maximum values defined on distance and altitude for s_1 and s_3 are* 56 *and* 60*, respectively. Then the Header Point is $HP(56, 60)$. In Figure 4.1, we present the local skyline region of a given database $\mathcal{O}_i$ mainly presented by the marginal points $IP(10, 20)$ and $HP(56, 60)$. All objects in the crosshatch area can not be in the local evidential skyline and the skyline region is represented by the rectangular including the points IP and HP.*

4.3 Distributed Evidential Skyline (DES)

In this section, we tackle the problem of conducting advanced analysis by means of skyline queries on uncertain data from distributed environments. We address the following major challenges:

- We first recall our new semantics for the evidential dominance by extending the dominance relationship of Pareto to the evidential context. We then present the local evidential skyline.

- We introduce efficient approach for querying and processing the evidential skyline over multiple and distributed servers. This approach is mainly based on the notions of: (i) increasing the parallelism to improve the efficiency and the effectiveness of the distributed evidential skyline computation and (ii) marginal points to resume the dominance regions of each local evidential skyline. We also propose the distributed architecture scheme of our distributed skyline processing.
- We develop efficient algorithms to compute the local evidential skylines, and the global evidential skyline. We conduct extensive experiments to show the efficiency and the effectiveness of our approach.

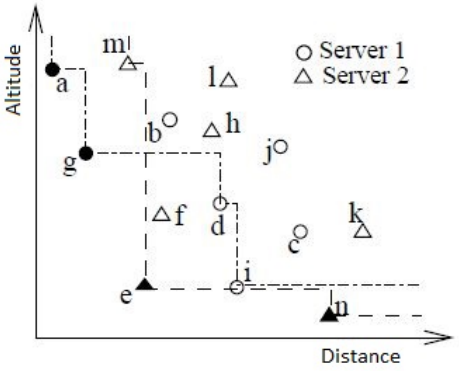


Figure 4.2: Distributed Skyline

Let us recall the example of acoustic sensors (e.g., microphones), they are often used to detect presence of a given object, its altitude, its distance, speed, etc. Due to the nature of acoustic sensing, values returned by microphones are intrinsically ambiguous, noisy and error-prone. Ideally, we want to find the most interesting objects which can meet the user needs in all dimensions. However, the size of data sets become larger and larger, and the skyline computation become more complex and expensive. Hence, to improve its performance, the skyline computation need to be processed on many servers instead of being limited to one. For instance, Figure 4.2 shows the skyline results when the objects are distributed over two servers.

This section is organised as follows. We start by defining the problem and presenting the distributed architecture scheme in section 4.3.1. In section 4.3.2, we first recall the semantics for the dominance relationship over evidential data. Then, we define the problem of the local evidential skyline. In section 4.3.3, we describe our efficient approach to compute the global skyline from multiple servers using the local and global marginal points. We also present our algorithms. An experimental study is reported in section 4.3.4. Finally, at the end of this section, we recall the main contributions.

4.3.1 Problem Definition

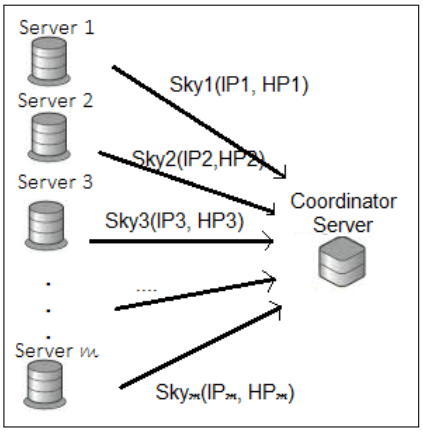


Figure 4.3: Distributed Architecture Scheme

The global skyline is deployed in a share-architecture as shown in Figure 4.3. The servers communicate the local skylines to the coordinator server $\mathcal{H}$. For the remainder of the chapter, we use the notation shown in Table 4.2. Given a set of m distributed servers $S = \{S_1, S_2, .., S_m\}$, each possessing an evidential database $\mathcal{O}_i$ $(1 \leq i \leq m)$ and a coordinator server $\mathcal{H}$ which is responsible for the final execution query and for the global skyline computation. Based on the semantics of the evidential skyline over a centralized database and the independence of those local databases $\mathcal{O}_i$, $\mathcal{H}$ efficiently computes the global skyline as shown in section 4.3.3.

Table 4.2: Frequently Used Symbols.

Symbol	Interpretation
$\mathcal{H}$	The central server (coordinator)
m	The number of local servers
S_i	The i^{th} local server
$\mathcal{O}_i$	The i^{th} evidential database of server S_i
$\mathcal{O}$	The global evidential database
$b\text{-}SKY(\mathcal{H})$	The global skyline objects retrieved by $\mathcal{H}$
$b\text{-}SKY(\mathcal{O}_i)$	The local skyline objects from by $\mathcal{O}_i$
$SR(\mathcal{O}_i)$	The skyline region of the database $\mathcal{O}_i$

A first step is to increase parallelism as shown in Figure 4.3. The local skylines derived from different servers are computed in parallel to improve performance. The coordinator server, in a second step, combines the skyline sets and returns the global skyline.

4.3.2 Local Evidential Skyline

In this section, we first present the problem definition for the local evidential skyline, then, we describe the local skyline computation.

Given a set of objects $\mathcal{O}_i = \{o_1, o_2, \ldots, o_n\}$ defined on a set of attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$, with $o_t.a_r$ denotes the *bba* of object o_t w.r.t. attribute a_r.

To extend the dominance relationship to the evidential data, we refer to the definition 3.1 described in chapter 3.

All properties stated and proved in the previous chapters are always meaningful for the local evidential skyline.

Intuitively, an object is in the believable skyline if it is not believably dominated by any other object. Based on the believable dominance relationship, the notion of $b\text{-}SKY(\mathcal{O}_i)$ is defined as follows.

Definition 4.3. *(The local skyline $b\text{-}SKY(\mathcal{O}_i)$) The believable skyline of $\mathcal{O}_i$ denoted by $b\text{-}SKY(\mathcal{O}_i)$, comprises those objects in $\mathcal{O}_i$ that are not believably dominated by any other object, i.e.,*
$b\text{-}SKY(\mathcal{O}_i) = \{o_t \in \mathcal{O}_i \mid \nexists\ o_h \in \mathcal{O}_i, o_h \succ_b o_t\}$.

For the local skyline, we use the same properties and proofs.

Local Skyline Computation

A straightforward algorithm to compute the local evidential skyline denoted by BLSE, is to compare each object o_t against the other objects. If o_t is not b-dominated by any other object, then it belongs to the evidential skyline. However, this approach results in a high computational cost (see Section 4.3.4). Also, since the b-dominance relationship is not transitive as proved in the previous chapters, an object cannot be eliminated from the comparison even if it is b-dominated since it will be useful for eliminating other objects. For this reason, we propose an algorithm denoted by LSE (see Algorithm 6) in which we follow the principle of the two scans algorithm cited in (Chan et al., 2006) in order to efficiently compute the evidential skyline.

The algorithm denoted LSE has a quadratic time complexity, i.e, $T(n) = O(n^2)$. First, LSE algorithm computes the evidential skyline through two phases. In the first phase (lines 2–13), we compare, each object o_t in $\mathcal{O}_i$ with those selected in $b\text{-}SKY(\mathcal{O}_i)$. If an object o_h in $b\text{-}SKY(\mathcal{O}_i)$ is b-dominated by o_t, we remove o_h from the set of candidates

Algorithm 6: Local Skyline over Evidential Data (LSE)

Input: Objects $\mathcal{O}_i$; belief threshold b
Output: $b\text{-}SKY(\mathcal{O}_i)$;

```
1  begin
2      foreach o_t in O_i do
3          isSkyline ← true;
4          foreach o_h in b-SKY(O_i) do
5              if isSkyline then
6                  if o_h ≻_b o_t then
7                      isSkyline ← false;
8                  else
9                      undom(o_t) ← undom(o_t) ∪ {o_h};
10             if o_t ≻_b o_h then
11                 remove o_h from b-SKY(O_i);
12         if isSkyline then
13             insert o_t in b-SKY(O_i);
14     foreach o_t in b-SKY(O_i) do
15         foreach o_h in O_i \ (b-Sky_{O_i} ∪ undom(o_t) ∪ {o_t}), pos(o_h) < pos(o_t) do
16             if o_h ≻_b o_t then
17                 remove o_t from b-SKY(O_i);
18     return b-SKY(O_i)
```

objects since it is not part of the evidential skyline. At the end of the comparison of o_t with objects of $b\text{-}SKY(\mathcal{O}_i)$, if o_t is not b-dominated by any object then, it is added to $b\text{-}SKY(\mathcal{O}_i)$. To avoid the situation illustrated by the example in Table 3.7, a second phase is needed (lines 14–17). To determine if an object o_t in $b\text{-}SKY(\mathcal{O}_i)$ is indeed in the b-$SKY(\mathcal{O}_i)$, it is sufficient to compare o_t with those in $\mathcal{O} \setminus \{b\text{-}SKY(\mathcal{O}_i) \cup undom(o_t) \cup \{o_t\}\}$ where $undom(o_t)$ is the set of objects that occurs before o_t and that do not b-dominate o_h. Even if, LSE minimizes the number of dominance checks, it also may result in a high computational cost. In particular, when the average number of dimensions is large. Thus, it is crucial to optimize the dominance checks to improve the performance of the local evidential skyline computation and therefore the LSE computation. In the following, in order to check the b-dominance between objects, we devise an efficient method (algorithm 7) that overcomes this problem using the minimum and the maximum values of each *bba* w.r.t. each attribute, according to the property stated in the previous chapters.

Algorithm 7: b-dominates(o_t, o_h, b)

```
1  strict ← False;
2  while a_r in A and strict=False do
3      if o_t.a_r^- > o_h.a_r^+ then
4          return false;
5      if bel(o_t.a_r ≤ o_h.a_r) < b then
6          return false;
7      if bel(o_t.a_r < o_h.a_r) ≥ b then
8          strict ← True;
9  if strict = False then
10     return false;
11 return true;
```

4.3.3 Efficient DES Computation

In this section, we introduce the notions of the global marginal points based on the notion of the local marginal points. Then, we present our methods to efficiently compute the DES.

Because the believable dominance relationship is intransitive as previously proved, a naive approach to retrieve the global skyline from different local skylines from multiple servers, is to compare all objects in the coordinator server $\mathcal{H}$. This way results in a very costly procedure. To avoid the full scan of the database in $\mathcal{H}$, one has to prune the search space in some cases.

We present in this section an approach for optimizing the DES by using the notions of local marginal points: "$IdealPoint$" and "$HeaderPoint$" which keep up a concise summary of the skyline region of each $\mathcal{O}_i$, denoted by $SR(\mathcal{O}_i)$. A skyline region includes the candidate objects that can be in the skyline and excludes the already visited objects (the dominated objects). This summary allows a fraction of objects in each $\mathcal{O}_i$ to be pruned from the global evidential skyline processing phases required in the naive approach, thus reducing the overall cost of the expensive dominance checks.

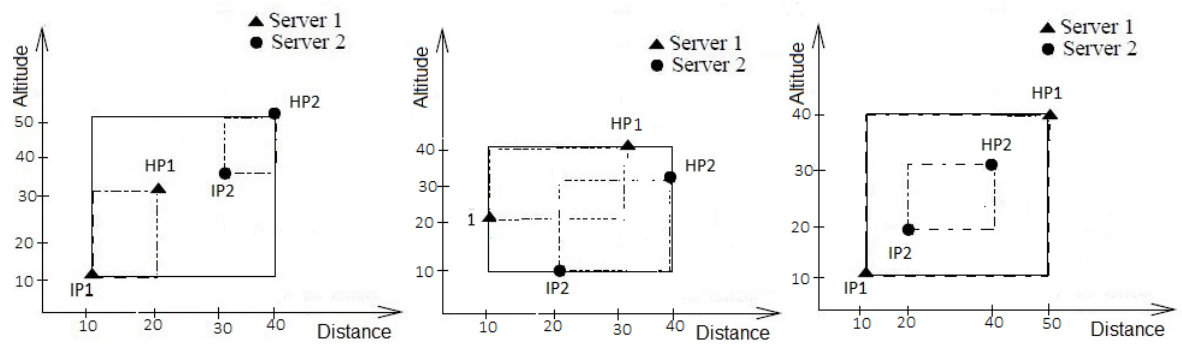
In this section, we use the local marginal points to define the global skyline region of $\mathcal{O}$, i.e., we define the global marginal points of objects in $\mathcal{H}$. Our goal is to leverage the global skyline regions to determine whether an object in $\mathcal{H}$ should be considered as a candidate object for the global skyline or be pruned in advance from the expensive skyline processing phases.

Global Marginal Points

In this section, we define the global skyline over evidential data from multiple and different databases. A key property gives: $b\text{-}SKY(\mathcal{H}) = b\text{-}SKY(\cup_{1\leq i\leq m} b\text{-}SKY(\mathcal{O}_i))$. A straightforward method to compute the global skyline (algorithm BGES) $b\text{-}SKY(\mathcal{H})$ is to consider the local skyline sets $(\cup b\text{-}SKY(\mathcal{O}_i), \forall i \in \{1..m\})$ as an input for the algorithm 6 to retrieve the final skyline objects, but this approach can result in a high computational cost.
Our proposition is that the coordinator server $\mathcal{H}$ computes the global skyline using the global marginal points of $\mathcal{O}$ to improve the performance of our approach. As shown in Figure 4.3, each local skyline is presented by its marginal points, i.e., its skyline region. Our goal is to define the global marginal points derived from $\mathcal{O}_1 \cup \mathcal{O}_2$ to determine whether a local skyline set should be considered as interesting or be pruned in advance from the expensive global skyline processing phases.

Let $\forall i \in \{1..m\}$, IP_i and HP_i be the ideal point and the header point, respectively, of $b\text{-}SKY(\mathcal{O}_i)$. $IP_i.a_r$ (resp. $HP_i.a_r$) denotes the IP (resp. HP) value defined on the attribute a_r.

Definition 4.4. *(Global Ideal Point GIP) The global ideal point is a certain and virtual point which its coordinates are the minimum values defined on the attributes $a_r \in \mathcal{A}$, i.e.,* $\forall a_r \in \mathcal{A}, GIP.a_r = min(IP_i.a_r, \forall i \in \{1..m\})$



(a) Case 1 (b) Case 2 (c) Case 3

Figure 4.4: Skyline Regions and Global Marginal Points

Definition 4.5. *(Global Header Point GHP) The global header point is a certain and virtual point which its coordinates are the maximum values defined on the attributes $a_r \in \mathcal{A}$, i.e.,*
$\forall a_r \in \mathcal{A}, GHP.a_r = max(HP_i.a_r, \forall i \in \{1..m\})$

Suppose we have two skyline sets of the databases $\mathcal{O}_1$ and $\mathcal{O}_2$. Let IP_1 and HP_1 (resp. IP_2 and HP_2) be the marginal points of the database $\mathcal{O}_1$ (resp. $\mathcal{O}_2$).

Example 4.4. *In Figure 4.4a, we have* $IP_1(10, 10)$ *and* $HP_1(20, 30)$. *Also,* $IP_2(30, 35)$ *and* $HP_2(40, 50)$. *Thus,* $GIP(10, 10)$ *and* $GHP(40, 50)$. *Also, in Figure 4.4b, we have* $GIP(10, 10)$ *and* $GHP(40, 40)$ *since* $IP_1(10, 20)$, $HP_1(30, 40)$ *and* $IP_2(20, 10)$, $HP_2(40, 30)$. *In Figure 4.4b, we have* $GIP(10, 10)$ *and* $GHP(50, 40)$. *Observe that the global skyline region of* $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *is described by the rectangle including the marginal points* GIP *and* GHP.

In Figure 4.4a, one can check that any object skyline in $SR(\mathcal{O}_2)$ can dominate those in $SR(\mathcal{O}_1)$.
Let $\mathcal{O}_i$ and $\mathcal{O}_j$ be two evidential databases.

Property 4.1. *If* $(GIP = IP_i$ *and* $GHP = HP_j) \wedge (\forall a_r \in \mathcal{A}, HP_i.a_r < IP_j.a_r)$ *then* $b\text{-}SKY(\mathcal{H}) = b\text{-}SKY(\mathcal{O}_i)$

Example 4.5. *In Figure 4.4a, one can observe that* IP_2 *is the best point in* $SR(\mathcal{O}_2)$, *and* HP_1 *is the worst point in* $SR(\mathcal{O}_1)$. *One can check that* $IP_2 \not\succ HP_1$, *then* $b\text{-}SKY(\mathcal{H}) = b\text{-}SKY(\mathcal{O}_1)$.

Property 4.2. *If* $(GIP \neq \{IP_i, IP_j\}$ *and* $GHP \neq \{HP_i, HP_j\}$ *(Fig. 4.4b))* $\vee$ *(*$GIP = IP_i$ *and* $GHP = HP_i$ *(Fig. 4.4c)), then* $b\text{-}SKY(\mathcal{H}) = b\text{-}SKY(b\text{-}SKY(\mathcal{O}_i) \cup b\text{-}SKY(\mathcal{O}_j))$

Example 4.6. *In Figure 4.4b and Figure 4.4c, all objects in* $SR(\mathcal{O}_1) \cup SR(\mathcal{O}_2)$ *can be candidates for the global skyline.*

Global Evidential Skyline Computation

The properties already described in section 4.3.3 are used in algorithm 8 whose inputs are the marginal points of all $\mathcal{O}_i \in \mathcal{O}$, the best IP and the Best HP denoted by BIP and BHP, respectively. $\forall a_k$ in $\mathcal{A}$, $BIP.a_k$ and $BHP.a_k$ are both initialized by the maximum value in Θ_{a_k}. In lines(2- -6), algorithm 8 returns the best skyline region from $\mathcal{O}$ presented by BIP and BHP. In lines (7- -11), if a given skyline region $SR(\mathcal{O}_i)$ is worst than the best skyline region BSR, i.e., $\forall a_k \in \mathcal{A}, BHP.a_k < IP_i.a_k$, (see property 4.1), objects in $SR(\mathcal{O}_i)$ are pruned in advance from the expensive global skyline processing.

The algorithm denoted GES has the complexity of $O(\mathcal{S} \times d)$ where $\mathcal{S}$ is the number of servers and d the number of attributes.

Algorithm 8: Global Evidential Skyline (GES)

Input: BIP; BHP; IP_i; HP_i
Output: $b\text{-}SKY(\mathcal{H})$;

```
1  begin
2      foreach O_i in O do
3          foreach a_k in A do
4              if IP_i.a_k ≤ BIP.a_k and HP_i.a_k ≤ BHP.a_k then
5                  BIP.a_k ← IP_i.a_k;
6                  HIP.a_k ← HP_i.a_k;
7      foreach O_i in O do
8          foreach a_k in A do
9              if IP_i.a_k < BHP.a_k then
10                 SkyCan ← SkyCan ∪ {b-SKY(O_i)} ;
11                 HIP.a_k ← HP_i.a_k;
12     b-SKY(H) ← b-SKY(SkyCan)
13     return b-SKY(H)
```

4.3.4 Experimental Evaluation

For our evaluation, we use a generated data-sets. We control the generation of evidential data by the parameters in Table 4.3, which provides us with the parameters and their default values.

Table 4.3: Examined Values for the Distributed Skyline Computation.

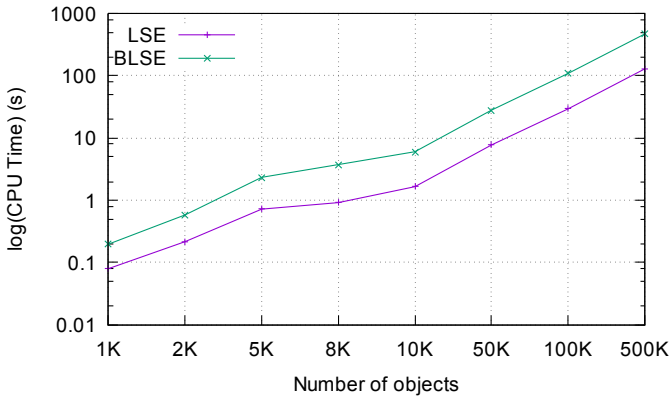
Parameter	Symbol	Values	Default
$\lvert\mathcal{O}_i\rvert$	n	1K, 2K, 5K, 8K, 10K, 50K, 100K, 500K	10K
$\lvert\mathcal{A}\rvert$	d	2, 3, 4, 5, 6	3
$\lvert\mathcal{F}\rvert$	f	2, 3, 4, 6, 8, 9, 10, 11	5
belief threshold	b	0.01, 0.1, 0.3, 0.5, 0.7, 0.9	0.5
$\lvert\Theta_{a_k}\rvert$	t	10, 50, 100, 150, 200	100
Servers	S	3, 5, 10, 15, 20	10

In each experimental setup, we evaluate the effect of one parameter, while we set the other to their default values. $|\mathcal{F}|$ represents the maximum number of focal elements per

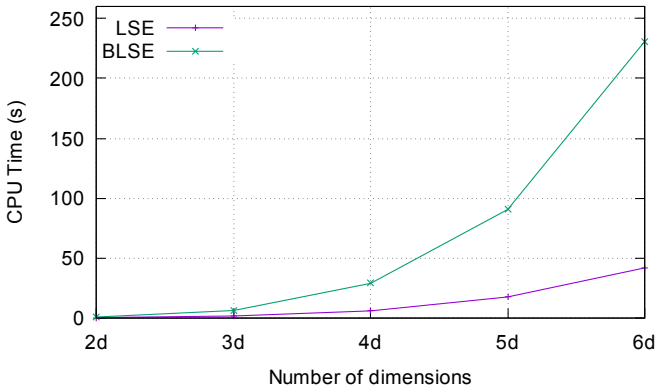
attribute. $|\Theta_{a_k}|$ represents the number of proposition in Θ_{a_k}.

The data generator and the algorithms, i.e, GES, BGES, LSE and BLSE were implemented in Java, and all experiments were conducted on a 2.3 GHz Intel Core i7 processor, with 6GB of RAM.

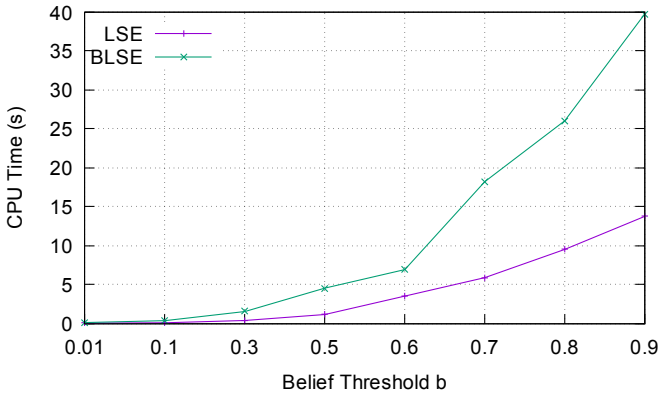
Figures 4.5a, 4.5b, 4.5c, 4.5d and 4.5e depict the execution time the implemented algorithms LSE and BLSE with respect to various parameters (n, d, b, t and f). Overall, LSE outperforms BLSE. More specifically, LSE is faster than BLSE. As expected, Figure 4.5a shows that the performance of the algorithms deteriorates with the increase of n.



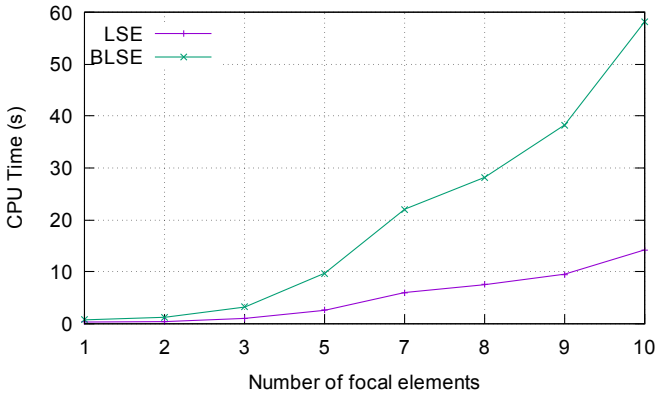
(a) Effect of n



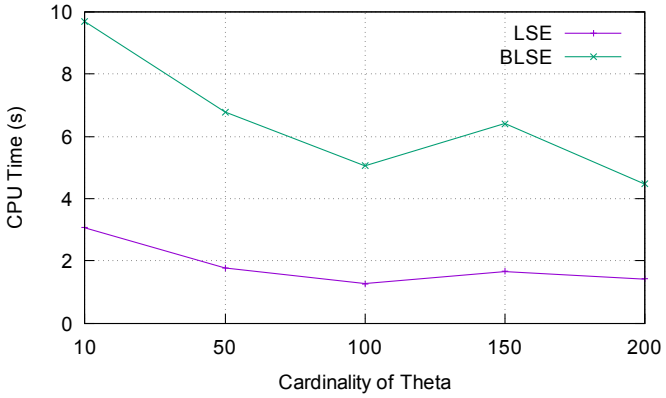
(b) Effect of d



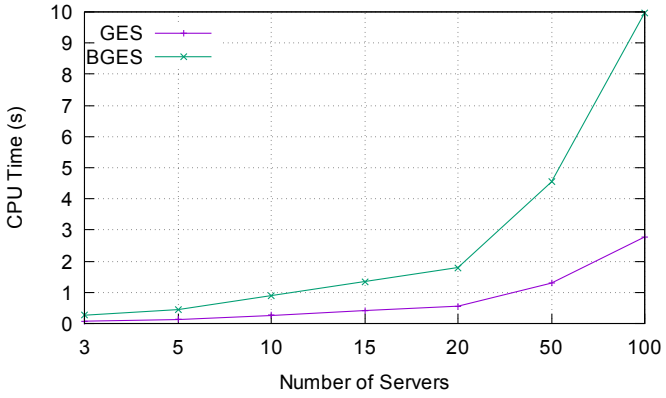
(c) Effect of b



(d) Effect of f



(e) Effect of t



(f) Effect of S

Figure 4.5: Distributed Evidential Skyline

As shown in Fig. 4.5b, the running time of the algorithms increases until the number of

dimensions increases. That is because when d increases, the number of dominance checks increases. Even if LSE performs the same number of dominance checks than BLSE, LSE is more efficient than BLSE since it can detect immediately whether an object does not dominate another.

Figure 4.5d shows that, with the increase of f, LSE algorithm outperforms the BLSE. As shown in figure 4.5c, the more b increases, the more the computation cost increases because the more we have a large number of skyline objects, the more the number of dominance checks increases. In contrast to n, d, b and f, the execution time of the algorithms is large for small number of propositions in Θ_{a_k} (Figure 4.5e). As expected, Figure 4.5f shows that the performance of the algorithms deteriorates with the increase of the servers number. Overall, GES outperforms BGES. More specifically, it is faster than the basic algorithm thanks to the proposed approach.

In this section, we have addressed the problem of the distributed skyline over evidential data. We presented the centring/local skyline analysis when data are imperfect and modeled by the evidence theory. We then defined the global evidential skyline which returns the most interesting objects from distributed environments.

Our experimental evaluation demonstrated the performance of the proposed algorithms. An interesting future direction is to use Hadoop in the distributed evidential skyline processing.

4.4 Evidential Skyline Maintenance

In chapter 3, we introduced a method for extracting skyline objects from an evidential database. When evidential data are updated, the skyline set could be computed, again from the overall updated database. It is the trivial maintenance of the skyline set. In this chapter, the aim is to incrementally maintain the skyline set, without starting from the scratch. Our objective is to reduce the computation cost of the maintenance by using the skyline set already computed. We address then the following major challenges:

- We propose efficient methods to maintain the skyline results in the evidential database context when an object is inserted or deleted.
- We perform an extensive experimental evaluation to demonstrate the scalability of the algorithms proposed for the evidential skyline maintenance.

This section is organised as follows. In subsections 4.4.1 and 4.4.2, we formally propose

a new approach for the incremental maintenance of evidential skyline. Our experimental evaluation is reported in subsection 4.4.3. Finally, we recall the main contributions.

4.4.1 Object Insertion

In this subsection, we discuss the maintenance problem of the evidential skyline after an insertion occurs in the evidential database EDB.

Approach Proposed

Because the dominance relationship is intransitive as proved in (Elmi et al., 2014), a naive approach to see if a new object has an impact on the evidential skyline, is to compare all objects in $\mathcal{O}$ against this inserted object. This way results in a very costly procedure. To avoid the full scan of the database, one has to prune the search space in some cases.

We present in this section an approach for optimizing the evidential skyline updating by using the notions of "$IdealPoint$" and "$HeaderPoint$" already defined in section 4.1. These points keep up a concise summary of the already visited regions of the objects' space. This summary allows a fraction of objects in $\mathcal{O}$ to be pruned from the skyline processing phases required in the naive approach, thus reducing the overall cost of expensive dominance checks.

An $IdealPoint$ and a $HeaderPoint$ summarize the region of data explored in earlier iterations. They enable a newly inserted object to be compared against this summary rather than to perform multiple comparisons against the whole objects in the b-skyline. Our goal is to leverage these two points to determine whether a newly inserted object should be considered as a candidate skyline object or be pruned in advance from the expensive skyline processing phases.

Our goal with the Header Point and the Ideal Point is to determine whether the newly inserted object should be considered as a candidate skyline object or be pruned in advance from the expensive skyline processing phases. As shown in Figure 4.1, if the newly inserted point denoted by P^+ is in the hatched area, i.e, is strictly better than IP in at least one dimension, then it is directly added to the evidential skyline. In this case, P^+ should be compared to all skyline's points because it could dominate one or several of them. We also propose another pruning strategy by adapting the notion of Header Point (HP) to the evidential database context. If P^+ is in the crosshatch area, then it cannot be in the evidential skyline because it has a value attribute which is worse than the HP. In other words, if a newly inserted object is not better than the Ideal Point in at least one dimension

and it is worse than the Header Point, then, the new object cannot be a skyline object and can be discarded.
If an evidential object is inserted, we have to refer to definition 2.1 in order to compare the new object against the Header Point and the Ideal Point. Let $P^+.a_k$ be the new object value defined on attribute a_k. Let also b be the threshold introduced by the user. Note that b is already considered to compute the original set of skyline points b-sky$_{\mathcal{O}}$.

Property 4.3. *If $\exists a_k \in \mathcal{A}$ such that $bel(P^+.a_k < IP.a_k) \geq b$, then P^+ is added to the b-skyline.*

Property 4.4. *If property 4.3 is not satisfied and $\exists a_k \in \mathcal{A}$ such that $bel(P^+.a_k \leq HP.a_k) < b$ then P^+ cannot be in the b-skyline.*

Proof. Suppose $\forall a_i \in \mathcal{A}$, $bel(P^+.a_i \leq HP.a_i) = 1$ and $\exists a_k \in \mathcal{A}$ such that $bel(P^+.a_k \leq HP.a_k) = x < b$.
We have $bel(P^+ \succ HP) = \prod_{i=1}^{d} bel(P^+.a_k \leq HP.a_k) = 1 * 1 * \cdots * 1 * x = x < b$. Thus P^+ does not b-dominates HP. □

Computation Method

Algorithm 9: Incremental maintenance after Insertion (MAI)

Input: b-$Sky_{\mathcal{O}}$, P^+: the object to insert; HP$(a_1, a_2, \ldots, a_d)$: Header Point, IP$(a_1, a_2, \ldots, a_d)$: Ideal Point

```
1  begin
2      var1 ← true;
3      while a_k in A and var1 do
4          if bel(P+.a_k < IP.a_k) ≥ b then
5              b-Sky_O' ← P+;
6              Compare P+ to all point in b-Sky_O;
7              var1 ← false;
8          else
9          if bel(HP.a_k < P+.a_k) ≥ b then
10             P+ cannot be in the b-skyline;
11             var1 ← false;
12         else
13         Execute b-Sky_O();
```

The algorithm denoted MAI has the complexity of $O(d)$ where d is the number of attributes.

A naive approach for skyline insertion maintenance is to recompute from scratch the b-sky$_{\mathcal{O}}$ considering the newly inserted object (Baseline Maintenance Algorithm after Insertion denoted by BMAI). Clearly, this approach may result in a high computational cost since we recompute the b-dominance relationship between all objects in $\{\mathcal{O}\} \cup \{P^+\}$. Algorithm 9 shows the algorithmic description of the proposed method in order to decrease the check space.
As shown in algorithm 9, we compare the *bba*s of the inserted object defined on each attribute against all values of the Ideal Point and the Header Point defined on the set of attributes $\mathcal{A}$. If it does exist an attribute $a_k \in \mathcal{A}$ such that $bel(P^+.a_k < IP.a_k) \geq b$, then the newly inserted object is added to the b-skyline and is compared to all skyline points since it may dominate some of them. If property 4.3 is not satisfied and it does exist an attribute $a_k \in \mathcal{A}$ such that $bel(\text{HP}.a_k <\text{P}^+.a_k) \geq b$, then the newly inserted object is directly discarded from the b-skyline since the Header Point b-dominates the new object P^+ in at least one dimension. Else, because of the well-known non transitivity of the b-dominance relationship, we have to re-compute the b-skyline from scratch which is represented by the function "b-Sky$_{\mathcal{O}}$()".

4.4.2 Object Deletion

In this subsection, we discuss the maintenance problem of the evidential skyline after a deletion occurs in the evidential database EDB.

Approach Proposed

In this section, we study the impact of skyline object deletion on the set of skyline points. Two simple approaches are discussed here. The most straightforward method for skyline deletion maintenance is to recompute from scratch (represented by the Baseline Deletion Algorithm BMAD) the b-Sky$_{\mathcal{O}}$. Clearly, this approach is overly simplistic and may result in a high computational cost because a considerable portion of evidential objects is not affected by the deleted point at all. This BMAD computation can be easily optimized for the purpose of deletion maintenance.
For a given skyline object $S_i \in b\text{-}Sky_{\mathcal{O}}$, we define its b-dominance region designed by b-DR(S_i) as the whole objects space that is dominated by S_i, and its exclusive b-dominance region, designed by EDR(S_i), which contains the objects space that is only dominated by S_i. For instance, in Figure 4.6, the b-DR(a) can be represented as rectangle $ahfe$. However,

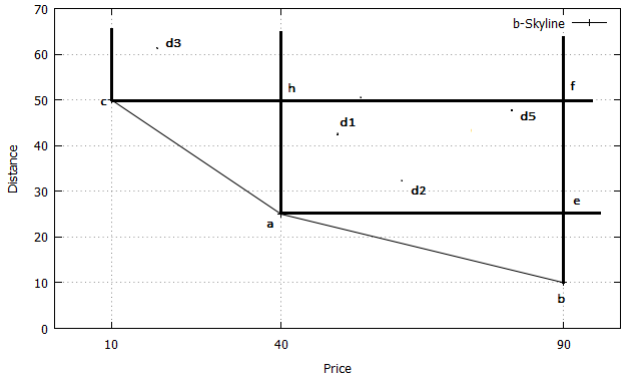


Figure 4.6: Exclusive Dominance Region Example.

objects d_1 and d_2 are exclusively b-dominated by object a. Thus, the exclusive b-dominance region of skyline object a is defined as follows: EDR$(a) = \{d_1, d_2\}$. As a result, both d_1 and d_2 are promoted in the skyline after object a is deleted. EDR(S_i) presents the smallest region that may contain the new skyline objects after deletion of S_i. Intuitively, EDR(S_i) contains those points that must be added to the new skyline after S_i is deleted, since those points are exclusively dominated by S_i.
Let b-$Sky_{\mathcal{O}}$ denotes the original skyline, b-$Sky_{\mathcal{O}}$'(S_i) denotes the new skyline after the deletion of skyline object S_i, i.e, $S_i \in b$-$Sky_{\mathcal{O}}$. Let $\Delta\mathcal{S}$ denotes the skyline objects that are expected to be added in the new skyline b-$Sky_{\mathcal{O}}$', then b-$Sky'_{\mathcal{O}}$ - b-$Sky_{\mathcal{O}} = \Delta S$. ΔS is exactly the exclusive dominance region of deleted object. The key issue now is to compute the EDR of the skyline object S_i in order to find the exact b-$Sky_{\mathcal{O}}$' with optimal I/O performances.

Skyline Objects S_i	b-DR(S_i)
$S_1 = o_2$	$\{o_1, o_3\}$
$S_2 = o_4$	$\{o_3\}$
$S_3 = o_6$	$\{o_5\}$

Table 4.4: b-Dominance Region

Example 4.7. *Suppose we have an evidential database that contains a set of objects* $\mathcal{O} = \{o_1, o_2, \ldots, o_6\}$ *and* b-$Sky_{\mathcal{O}} = \{o_2, o_4, o_6\}$. *Table 4.4 gives the objects that are dominated by each object in* b-$Sky_{\mathcal{O}}$, *i.e, the b-dominance region of skyline objects. Suppose* o_2 *is deleted, only objects that are exclusively dominated by* o_2 *are promoted as new skyline objects. However,* o_3 *can not be promoted to the b-skyline because it is dominated by one or more other objects. Table 4.5 shows that* o_2 *exclusively b-dominates* o_1. *It is then promoted to be in the new skyline* b-$Sky_{\mathcal{O}}$'. *As a result,* b-$Sky_{\mathcal{O}}$'$= \{o_1, o_4, o_6\}$

Object Skyline S_i	EDR(S_i)
o_2	$\{o_1\}$
o_4	$\emptyset$
o_6	$\{o_5\}$

Table 4.5: Exclusive b-dominance region of skyline points.

Computation Method

To compute the b-skyline over an evidential database, we refer to the algorithm proposed in section 3 called BS. In BS algorithm, while computing the b-dominance degrees between objects in $\mathcal{O}$, one can automatically save the exclusive b-dominance region of each skyline point EDR(S_i).

Algorithm 10: Incremental maintenance after Deletion (MAD)

Input: $\mathcal{O}$: evidential database, $b\text{-}Sky_{\mathcal{O}}$: evidential skyline, S_i: the skyline point to delete;

Output: $b\text{-}Sky_{\mathcal{O}}$': the new evidential skyline;

```
1 begin
2     b-Sky'_O ← b-Sky_O ∪ EDR(S_i);
3     return b-Sky'_O;
```

As it can be seen, once information about exclusive b-dominance region of each skyline object is available, we can detect via algorithm 10, which object can be directly promoted as a new skyline object.

4.4.3 Experimental Evaluation

In this section, we present an extensive experimental evaluation of our approaches. More specifically, we focus on the scalability of our proposed methods for maintaining the evidential skyline. For comparison purpose, we also implemented the baseline algorithms for maintenance after insertion and after deletion referred to as BMAI and BMAD.

Experimental Setup

The generation of evidential data sets is controlled by the parameters in Table 4.6. In each experimental setup, we investigate the effect of one parameter, while we set the remaining ones to their default values. K means a thousand of evidential objects.

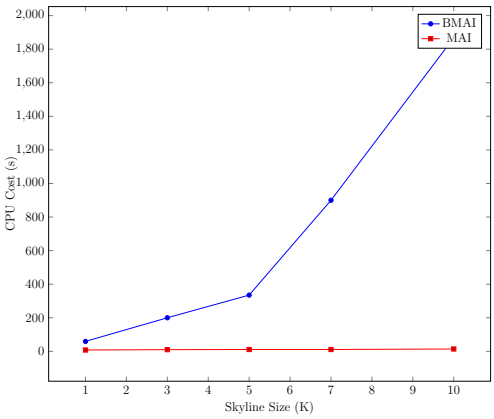
Parameter	Symbol	Values	Default
Skyline size	S	1K, 3K, 5K, 7K, 10K	10K
Number of objects in $\mathcal{O}$	n	10K, 30K, 50K, 70K, 100K	10K
Number of attributes	d	2, 4, 6, 8	4
Number of focal elements/attribute	f	3, 5, 7, 9	5

Table 4.6: Parameters for the Skyline Maintenance Computation

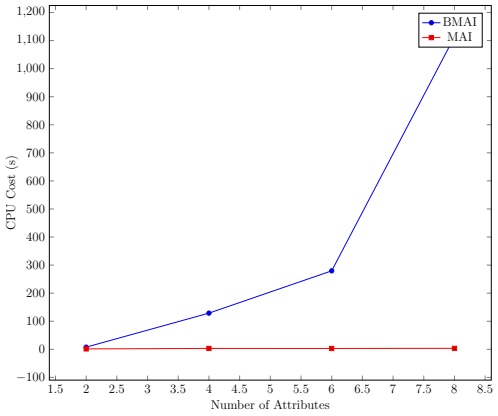
Performance Evaluation

Fig. 4.7 depicts the execution time of the implemented algorithms MAI and BMAI w.r.t. S, d and f. Overall, MAI outperforms BMAI. More specifically, MAI is faster than BMAI since it can detect immediately whether an inserted object is better or not than another existing in the b-Sky$_{\mathcal{O}}$. A simple comparison between the attributes values of P^+ and the certain points IP and HD, makes the algorithm MAI more efficient than BMAI. This later aims at comparing P^+ against all objects in the skyline set in order to check if the newly inserted object is in the evidential skyline or not.

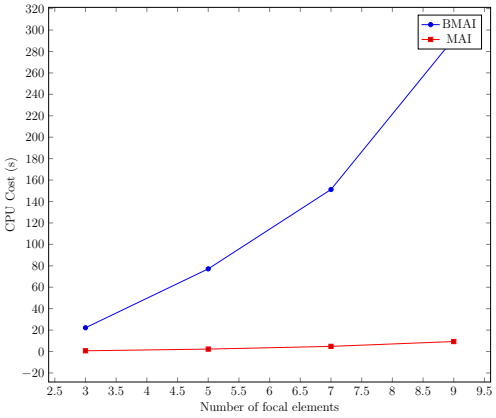
However, MAI decreases the research space by discarding points which are worst than the Header Point on the one hand, and inserting P^+ in the b-skyline if it is better than the Ideal Point on the other hand. As expected, Fig. 4.7a shows that the performance of the algorithm BMAI deteriorates with the increase of the skyline size S. This is because when S increases the number of dominance checks becomes larger. Observe that MAI is one order of magnitude faster than BMAI since it can quickly identify if an object can be in the skyline or not with a simple operation of comparison with both Header Point and Ideal Point. As shown in Fig. 4.7b, and Fig. 4.7c, MAI is not affected by increasing d and f as it makes a simple check. However, BMAI does not scale with d and f. Fig. 4.7 depicts also the execution time of the implemented algorithms MAD and BMAD w.r.t. n, d and f. Fig. 4.7d shows that the execution time of the algorithm MAD slightly increases with the increase of n as we have to check more and more objects in $\mathcal{O}$, but it outperforms



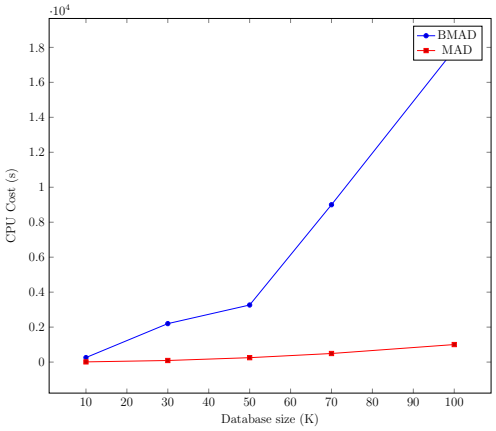
(a) Insertion: Effect of Skyline size



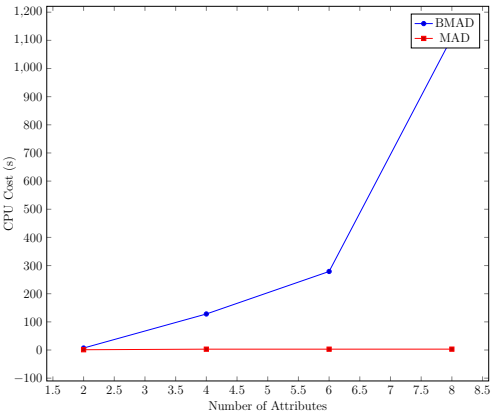
(b) Insertion: Effect of d



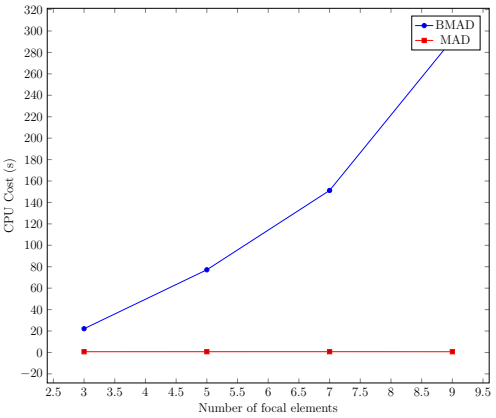
(c) Insertion: Effect of f



(d) Deletion: Effect of database size



(e) Deletion: Effect of d



(f) Deletion: Effect of f

Figure 4.7: Elapsed Time for Maintenance operations.

BMAD. Observe that BMAD has a high computational cost if we increase all parameters. It is not the case for MAD.

4.5 Conclusion

In this chapter, based on the marginal points, i.e., the ideal point and the header point, we have addressed, on the one hand, the problem of the distributed evidential skyline and, on the other hand, the problem of the evidential skyline maintenance.

Our solutions guarantees I/O optimality and can be easily implemented. Experimental results show that our methods outperform the naive methods.

In the next chapter, we will introduce the top-k evidential skyline. We particularly tackle the problem of the evidential skyline ranking, i.e., retrieve the k skyline objects that are expected to have the highest score with considering the confidence level (CL) of the objects. We also study the impact of CL on the top-k results. The efficiency and effectiveness of our proposal are verified by extensive experimental results.

Chapter 5

The Top-k Evidential Skyline

Contents

5.1 Introduction

How to conduct advanced analysis, particularly skyline analysis, remains an open problem at large. In many application domains where data are pervaded with uncertainty, end-users are often interested in the top-k query answers within a potentially large space answer. This requires to introduce a rank-order between the evidential skyline objects.

The skyline ranking has received an increasing importance (Lian & Chen, 2013; Yi, Li, Kollios, & Srivastava, 2008; Yong, Jin, & Seung, 2008; Amagata et al., 2016). Skyline ranking queries has been proposed in classic databases as well as uncertain databases such as probabilistic databases (Liu, Zhang, Xiong, Li, & Luo, 2015). Zhang et al (Ying, Wenjie, Xuemin, Bin, & Jian, 2011), propose to compute the top-k skyline objects for discrete cases. In addition, authors in (Ilaria et al., 2014) tackle the problem of developing a ranking semantics in a probabilistic databases. Moreover, a new approach introducing the skyline ranking for uncertain data with maybe confidence, is proposed in (Yong et al., 2008). Also, Lian and Chen in (Lian & Chen, 2013) provide an effective pruning method for computing the probabilistic top-k dominating queries in uncertain databases. In addition, the authors (Yiu & Mamoulis, 2007) introduce the concept of Top-k dominating queries on certain database to rank skyline objects. In (Ge, Zdonik, & Madden, 2009), authors propose to rank results according to some user-defined score.

5.1.1 Motivating Example

The need to manage uncertain data arises in several real-life applications. For example, acoustic sensors (e.g., microphones) are often used to detect presence of a given object, its altitude, its distance, speed, etc. Ideally, we want to find the most interesting objects which can meet the user needs in all dimensions, suppose that objects having the smaller distance and the smaller altitude are the most interesting objects. The skyline analysis is meaningful here since it discloses the trade-off between the merits of multiple aspects. An object o_i is in the skyline if there exists no object o_j such that o_j is better than o_i in (at least) one dimension, and is not worst than o_i in all other aspects. We argue that skyline analysis is also meaningful on imperfect data. Due to the nature of acoustic sensing, values returned by microphones are intrinsically ambiguous, noisy and error-prone. Add to that, an object in a database may have a confidence level (Bell et al., 1996)(i.e., probability of existence in a table). Often, a query over such data has a large number of result objects. In this context, the top-k query has proven to be useful.

5.1.2 Contributions

In this chapter, we tackle the problem of conducting advanced analysis by means of skyline queries on uncertain data modeled by the evidence theory. We address the following major challenges:

- We recall the evidential dominance relationship, then, the skyline over evidential objects.
- Based on this dominance relationship, we propose a score function reflecting the confidence level of each object. This score function aims at retrieving k objects that are expected to believably dominate the more the other objects.
- We develop efficient algorithms to the evidential skyline computation and the top-k query. We conduct extensive experiments to show the efficiency and the effectiveness of our approach. In addition, our extensive experiments reflects the impact of the confidence level on the top-k skyline results.

This chapter is published in the 13th IEEE/ACS International Conference of Computer Systems and Applications (Elmi, Hadjali, Bach Tobji, & Ben Yaghlane, 2016).

5.1.3 Chapter Organisation

The rest of the chapter is organized as follows. In section 5.2, we recall the evidential dominance definition and the evidential skyline. We conclude the section by defining a score function in order to retrieve the top-k skyline. Section 5.3 is devoted to the top-k skyline computation. An experimental study is reported in section 5.4. Finally, Section 5.5 recalls the main contributions.

5.2 Top-k Skyline over evidential objects

In this section, we present the notion of the evidential skyline. This later aims at retrieving the most interesting objects in $\mathcal{O}$ that are not dominated by any other objects. We first present the concept of the evidential dominance denoted by the b-dominance and then the evidential skyline denoted by the b-sky$_{\mathcal{O}}$. We finally define the score function that allows

us to retrieve the top-k skyline objects in a given database $\mathcal{O}$. The score function is based on the confidence level.

Given a set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ defined on a set of attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$, with $o_i.a_r$ denotes the *bba* of object o_i w.r.t. attribute a_r. The degree of belief that an

Table 5.1: Evidential database example.

Object	Distance	Altitude	Confidence Level CL
o_1	$\langle\{150, 160, 180\}, 0.1\rangle$, $\langle\{190, 200\}, 0.9\rangle$	$\langle 60, 0.3\rangle$, $\langle 100, 0.7\rangle$	[0.1, 0.3]
o_2	$\langle 100, 0.7\rangle$ $\langle\Theta, 0.3\rangle$	$\langle\{70, 80\}, 0.8\rangle$, $\langle 80, 0.2\rangle$	[0.1, 0.2]
o_3	70	Θ	[0.2, 0.5]
o_4	$\langle\{50, 60\}, 0.8\rangle$, $\langle\{65\}, 0.2\rangle$	60	[0.7, 0.7]
o_5	$\langle\{50\}, 0.5\rangle$, $\langle\{60\}, 0.5\rangle$	$\langle\{60\}, 0.6\rangle$, $\langle\{70\}, 0.4\rangle$	[0.9, 1]

object o_i is better than or equal (or strictly better) to other object o_j w.r.t. an attribute a_r is given by (Bell et al., 1996):

$$bel(o_i.a_r \leq o_j.a_r) = \sum_{A \subseteq \Theta_{a_r}} (m_{ir}(A) \sum_{B \subseteq \Theta_{a_r}, A \leq^{\forall} B} m_{jr}(B)) \tag{5.1}$$

Where $A \leq^{\forall} B$ stands for $a \leq b$, $\forall (a, b) \in A \times B$.

$$bel(o_i.a_r < o_j.a_r) = \sum_{A \subseteq \Theta_{a_r}} (m_{ir}(A) \sum_{B \subseteq \Theta_{a_r}, A <^{\forall} B} m_{jr}(B)) \tag{5.2}$$

Where $A <^{\forall} B$ stands for $a < b$, $\forall (a, b) \in A \times B$.

Example 5.1. *In Table 5.1, one can check that*[1] $bel(o_1.d \leq o_2.d) = 0$*,* $bel(o_1.a \leq o_2.a) = 0.3$*,* $bel(o_2.d < o_1.d) = 0.7$*, and* $bel(o_2.a \leq o_1.a) = 0.7$

To extend the dominance relationship to the evidential data, we refer to the definition 3.1 described in chapter 3.

All properties related to the b-dominance relationship already presented in the previous chapters are meaningful.

Intuitively, an object is in the believable skyline if it is not believably dominated by any other object. Based on the b-dominance relationship, the notion of b-$sky_{\mathcal{O}}$ is defined as follows.

[1]For short, we use d and a to denote the distance and the altitude attributes, respectively

Definition 5.1. *(The b-skyline) The b-skyline of $\mathcal{O}$ denoted by b-sky$_\mathcal{O}$, comprises those objects in $\mathcal{O}$ that are not b-dominated by any other object, i.e.,*
b-sky$_\mathcal{O}$ $= \{o_i \in \mathcal{O} \mid \nexists\ o_j \in \mathcal{O}, o_j \succ_b o_i\}$.

For each object S_i in the b-skyline, we need to know how much does it dominate other objects in the b-skyline set. We first introduce the notion of the object dominance score denoted by $\mu(S_i)$. We then introduce a new score function reflecting the confidence level impact.

Definition 5.2. *(Dominance score) Let S_i be in the b-sky$_\mathcal{O}$. The dominance score that reflects how much S_i dominates other skyline objects denoted by $\mu(S_i)$ is defined as follows:*

$$\mu(S_i) = \sum_{\forall S_j \in b-sky_\mathcal{O}, S_i \neq S_j} bel(S_i \succ S_j) \times \frac{1}{|b - sky_\mathcal{O}| - 1} \tag{5.3}$$

where

$$bel(S_i \succ S_j) = \prod_{a_r \in \mathcal{A}} bel(S_i.a_r \leq S_j.a_r) \tag{5.4}$$

Unfortunately, the semantics of dominance score in such data is unclear, due to the fact that both scores and probabilities of objects (confidence level) must be taken in account. For example, it is unclear whether it is better to report highly scored objects with a relatively low belief of existence or a lower -scored objects with high belief of existence. Therefore, we introduce a score dominance with confidence level denoted by $\sigma(S_i)$.

Definition 5.3. *(Dominance score with confidence level) Let S_i be a skyline object. Let $CL[bl(S_i), pl(S_i)]$ be the confidence level of the object skyline S_i where $bl(S_i)$ and $pl(S_i)$ are the belief and plausible degrees, respectively, of the object existence in $\mathcal{O}$. $\sigma(S_i)$ presents the score dominance that reflects the belief degree of the object existence. $\sigma(S_i)$ is defined as follows:*

$$\sigma(S_i) = \mu(S_i) \times bl(S_i) \tag{5.5}$$

Example 5.2. *In Table 5.1, we have 0.5-sky$_\mathcal{O}$* $= \{o_4, o_5\}$. *$bel(o_4 \succ o_5) = 0.4$ since $bel(o_4.d \leq o_5.d) = 0.4$ and $bel(o_4.a \leq o_5.a) = 1$, thus, $\mu(o_4) = 0.4$ and $\sigma(o_4) = 0.4 \times 0.7 = 0.28$*
$bel(o_5 \succ o_4) = 0.36$ since $bel(o_5.d \leq o_4.d) = 0.6$ and $bel(o_5.a \leq o_4.a) = 0.6$, thus, $\mu(o_5) = 0.36$ and $\sigma(o_5) = 0.36 \times 0.9 = 0.324$

Definition 5.4. *(Top-k Skyline query) A top-k skyline query selects the k objects in b-sky$_\mathcal{O}$ with the largest dominance score μ. A TOPk-CL query selects the k objects in b-sky$_\mathcal{O}$ with the largest dominance score with confidence level σ.*

Example 5.3. *Let us recall the example, 0.5-sky$_{\mathcal{O}}$* $= \{o_4, o_5\}$*. One can observe that top-*$1 = \{o_4\}$ *since* $\mu(o_4) > \mu(o_5)$ *and top-*1*-CL*$= \{o_5\}$ *since* $\sigma(o_5) > \sigma(o_4)$

5.3 Top-k Skyline Computation

In this section, we first discuss how our top-k skyline algorithm can highlight the properties of dominance relationship mentioned in section 5.2. Then, we propose an efficient method in order to reduce the complexity of belief function operations.

A straightforward algorithm to compute the top-k skyline denoted by BTOPk-CL (Basic TOPk-CL) is to compare each object o_i against the other objects. If o_i is not b-dominated, then it belongs to the b-skyline and can be in the top-k objects. However, this approach results in a high computational cost (see Section 5.4).

Also, since the b-dominance relationship is not transitive, an object cannot be eliminated from the comparison even if it is b-dominated since it will be useful for eliminating other objects.
For this reason, we propose an algorithm denoted by TOPk-CL (see Algorithm 11). In lines (2–21), we follow the principle of the two scan algorithm cited in (Chan et al., 2006)

in order to efficiently compute the evidential skyline.

Algorithm 11: Top-k skyline with CL (TOPk-CL)

Input: Objects $\mathcal{O}$; belief threshold b; a retrieval size k
Output: TOPk-CL;

```
1  begin
2      foreach o_i in O do
3          isSkyline ← true;
4          s ← 0;
5          foreach o_j in b-Sky_O do
6              if isSkyline then
7                  if o_j ≻_b o_i then
8                      isSkyline ← false;
9                  else
10                     undom(o_i) ← undom(o_i) ∪ {o_j};
11                     s ← s + bel(o_i ≻ o_j);
12             if o_i ≻_b o_j then
13                 remove o_j from b-Sky_O;
14         if isSkyline then
15             insert o_i in b-Sky_O;
16             μ(o_i) ← s/(|b-sky_O| − 1) ;
17             σ(o_i) ← μ(o_i) × bl(o_i) ;
18     foreach o_i in b-Sky_O do
19         foreach o_j in O \ (b-Sky_O ∪ undom(o_i) ∪ {o_i}), pos(o_j) < pos(o_i) do
20             if o_j ≻_b o_i then
21                 remove o_i from b-Sky_O;
22     TOPk-CL← top-k(b-sky_O, k);
23     return TOPk-CL
```

The algorithm denoted TOPk-CL has the complexity of $O(n^2)$ where n is the number of evidential objects.

To compute the TOPk-CL skyline query over evidential data, we need to first compute the evidential skyline. After that, we rank the objects in the skyline according to the dominance score function with CL ($\sigma()$) as defined in the previous section. First, TOPk-CL algorithm computes the evidential skyline through two phases. In the first phase (lines 2–17), we compare in the one hand, each object o_i in $\mathcal{O}$ with those selected in b-$Sky_{\mathcal{O}}$ and

we compute the score dominance attributed for each object. If an object o_j in b-$Sky_{\mathcal{O}}$ is b-dominated by o_i, we remove o_j from the set of candidate objects since it is not part of the evidential skyline. At the end of the comparison of o_i with objects of b-$Sky_{\mathcal{O}}$, if o_i is not b-dominated by any object then, it is added to b-$Sky_{\mathcal{O}}$ with a final score.

To avoid the situation illustrated by the example in Table 3.7, a second phase is needed (lines 18–21). To determine if an object o_i in b-$Sky_{\mathcal{O}}$ is indeed in the b-$sky_{\mathcal{O}}$, it is sufficient to compare o_i with those in $\mathcal{O} \setminus \{b\text{-}Sky_{\mathcal{O}} \cup undom(o_i) \cup \{o_i\}\}$ that occur earlier than o_i since the other ones have been already compared against o_i, where $undom(o_i)$ is the set of objects that occurs before o_i and that do not b-dominate o_j.

Each object in the b-$sky_{\mathcal{O}}$ has a score dominance computed in line 16 and a score dominance withe CL computed in line 17. The method $top\text{-}k(b\text{-}sky_{\mathcal{O}}, k)$ shown in line 22, returns k objects in b-$sky_{\mathcal{O}}$ that are expected to have the largest score dominance value $\sigma()$. Even if, TOPk-CL minimizes the number of dominance checks, it also may result in a high computational cost. In particular, when the average number of dimensions is large. Thus, it is crucial to optimize the dominance checks to improve the performance of the b-skyline computation and therefore the TOPk-CL computation. In the following, in order to check the b-dominance between objects, we devise an efficient method that overcomes this problem using the minimum and the maximum values of each *bba* w.r.t. each attribute.

Algorithm 12: b-dominates(o_i, o_j, b)

```
1  strict ← False;
2  while a_r in A and strict=False do
3      if o_i.a_r^- > o_j.a_r^+ then
4          return false;
5      if bel(o_i.a_r ≤ o_j.a_r) < b then
6          return false;
7      if bel(o_i.a_r < o_j.a_r) ≥ b then
8          strict ← True;
9  if strict = False then
10     return false;
11 return true;
```

To efficiently check if a given object o_i b-dominates or not another object o_j, we propose an efficient method (see Algorithm 12). To determine if an abject b-dominates another, it is not necessary to iterate all *bbas* defined on all attributes.

The details of the b-dominates() function are as follows. For each attribute $a_r \in \mathcal{A}$,

$o_i.a_k^-$ is compared against $o_j.a_k^+$. If there is any attribute a_r for which $o_j.a_k^+ < o_i.a_k^-$ holds then return false (If in line 4); since o_i cannot b-dominate o_j. The method returns "false" as soon as the degree of belief that $o_i.a_r$ is less or equal than $o_j.a_r$ denoted by $bel(o_i.a_r \leq o_j.a_r)$, is less than the threshold b; since that does not satisfy the principle of the b-dominance. Finally, the method returns "false" as well as if it does not exist an attribute where o_i is strictly better than o_j.

5.4 Experimental Evaluation

In this section, we have performed an extensive experimental evaluation of the proposed framework. We report empirical study to examine the effectiveness and the efficiency of the top-k skyline query with confidence level on evidential data. More specifically, we focus on the following important issues: (1) the scalability and the performance of our proposed techniques for computing the TOPk-CL algorithm. For comparison purposes, we implement a baseline algorithm for the TOPk-CL referred to as BTOPk-CL. In addition, we implemented an efficient method; b-dominates function to show how does it perform the TOPk-CL algorithm. (2) We implement the top-k query without confidence level referred to as TOPk and we conduct an analysis about the impact of the confidence level on the top-k skyline results.

5.4.1 Experimental Setup

For our evaluation, we use a generated data-sets. We control the generation of evidential data by the parameters in Table 5.2, which provides us with the parameters and their default values. In each experimental setup, we evaluate the effect of one parameter, while we set the other to their default values.

The data generator and the algorithms, i.e, TOPk, TOPk-CL and BTOPk-CL were implemented in Java, and all experiments were conducted on a 2.3 GHz Intel Core i7 processor, with 6GB of RAM.

5.4.2 Performance and Scalability

In this section, we show on the one hand, that the TOPk-CL algorithm is more scalable than the BTOPk-CL algorithm.

Figure 5.1 depicts the execution time of the implemented algorithms with respect to

Parameter	Symbol	Values	Default
Number of objects	n	1K, 5K, 10K, 50K, 100K	10K
Number of attributes	d	2, 3, 4, 5, 6	3
Nbr of focal elmts/attr	f	2, 3, 4, 6, 8, 10	4
belief threshold	b	0.1, 0.3, 0.5, 0.7, 0.9	0.5
Theta cardinality	t	10, 50, 100, 200	100
Top-k size	k	10, 20, 30, 40, 50	10

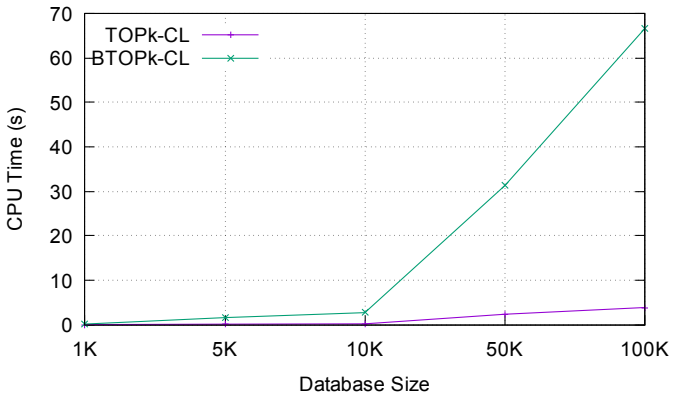
Table 5.2: Examined Values for the Top-k Skyline Computation.

various parameters (n, d, f, b, t and k). Overall, TOPk-CL outperforms BTOPk-CL. More specifically, it is faster than the basic algorithm. As expected, figure 5.1 shows that the performance of all the algorithms deteriorates with the increase of n, d, f and b. That is because the more these parameters increase, the more the dominance checks increase. Observe that TOPk-CL is one order of magnitude faster BTOPk-CL since it can quickly identify whether an object is b-dominated. For example, as shown in figure 5.1b, when d increases, the size of the evidential skyline becomes larger, thus a large number of objects will be selected to the second phase. Hence, BTOPk-CL performs a large number of dominance checks with a basic function. Even if TOPk-CL performs the same number of dominance checks than BTOPk-CL, TOPk-CL is more efficient than BTOPk-CL since it can detect immediately whether an object dominates or not another. In contrast to n, d, f and b, varying t and k, has no apparent effect on CPU Time as shown in figure 5.1e and figure 5.1f, respectively.

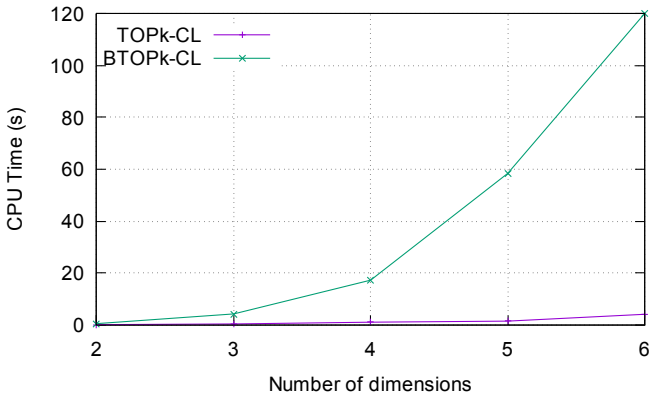
5.4.3 Top$k$-CL VS Top$k$

In this section, we study the impact of the confidence level which reflects the belief and plausibility degrees of the objects existence in the database. For each parameter value, we first obtain the skyline set b-sky$_{\mathcal{O}}$, we then implement two algorithms: TOPk-CL skyline and TOPk skyline which have the same input (b-sky$_{\mathcal{O}}$). The TOPk skyline algorithm is based on the dominance score function $\mu()$ to rank the skyline set. However, the TOPk-CL skyline is based on the dominance score function with confidence level $\sigma()$. For each parameter value, we study the impact of the confidence level by returning the number of common objects between the two result sets denoted by TOPk∩TOPk-CL. Note that for figures 5.2a, 5.2b, 5.2c, 5.2d and 5.2e, we are interested in the top-10 query, i.e., the result set size is $k = 10$.

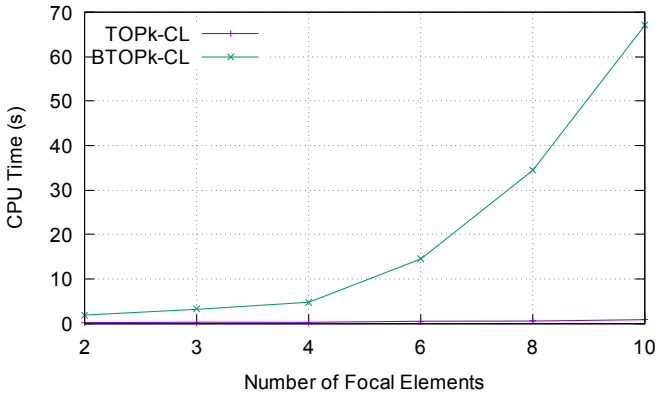
For example, in figure 5.2a, TOP10-CL and TOP10 have 6 common objects when



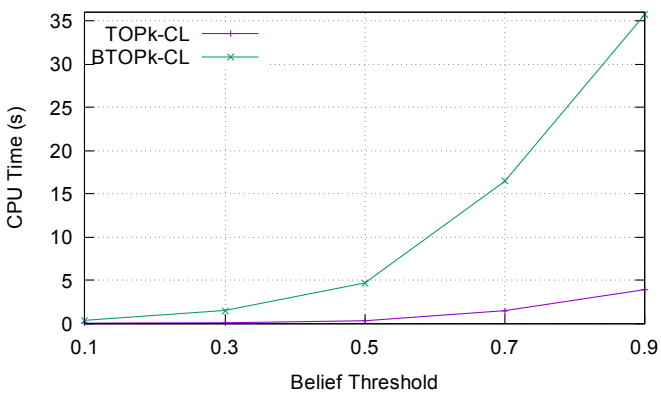
(a) Effect of n



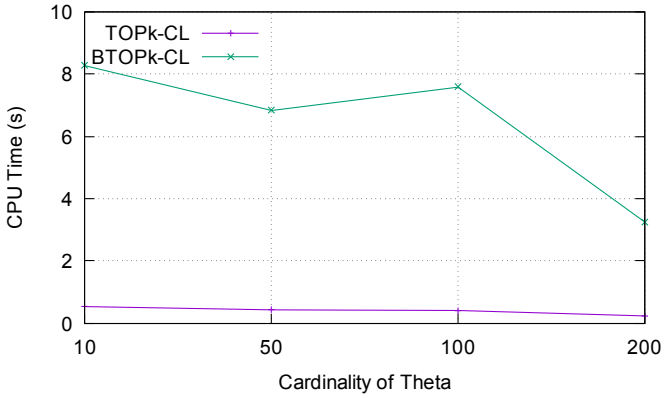
(b) Effect of d



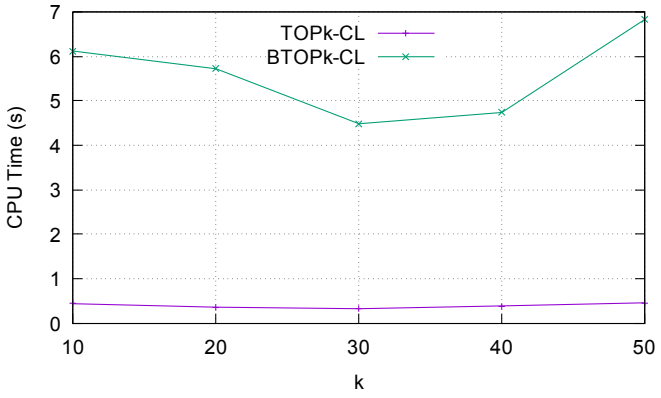
(c) Effect of f



(d) Effect of b



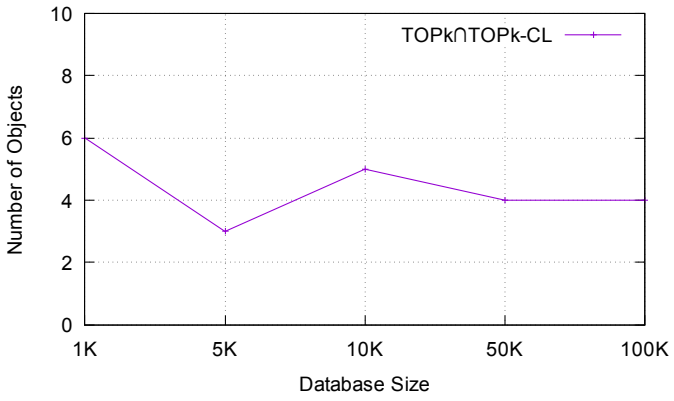
(e) Effect of t



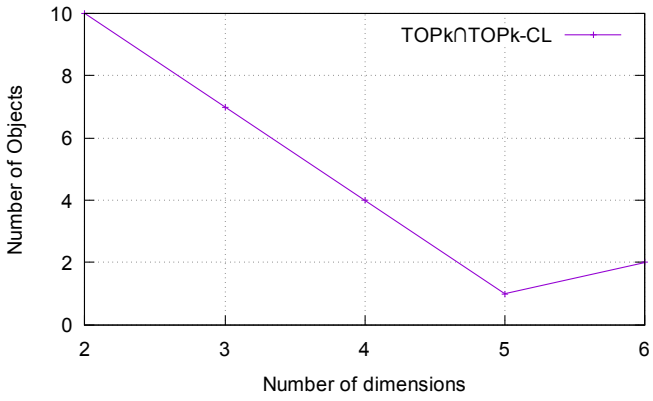
(f) Effect of k

Figure 5.1: Performance of Top-k Skyline varying the parameters

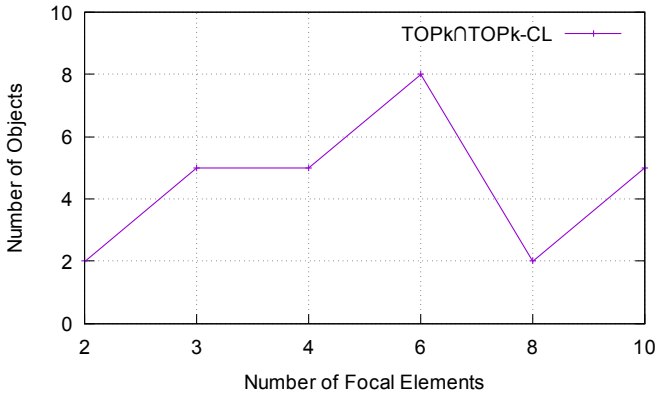
$n = 1K$, i.e., the confidence level changes 40% of the final result.



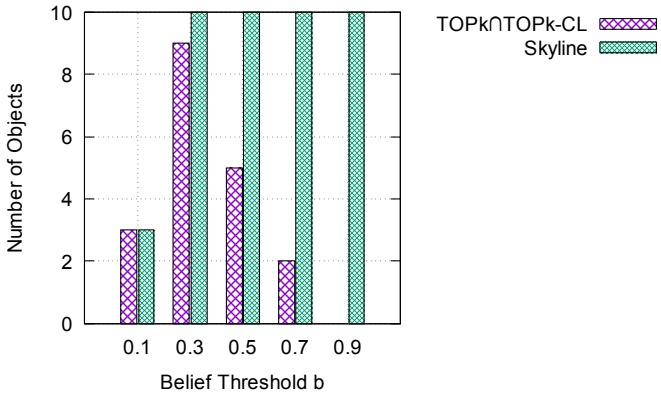
(a) Effect of n



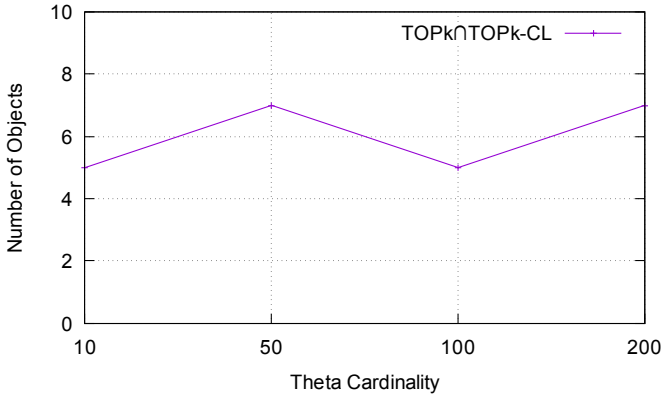
(b) Effect of d



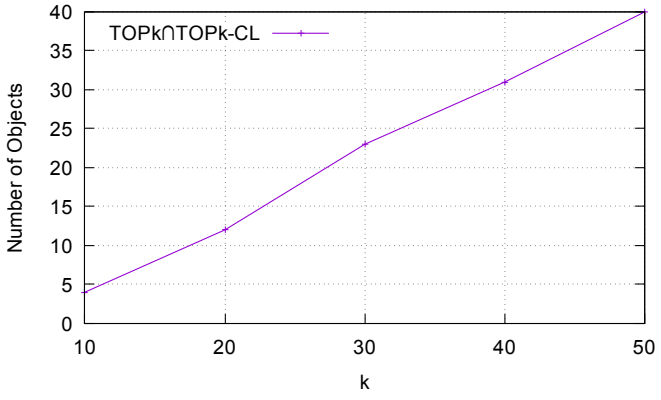
(c) Effect of f



(d) Effect of b



(e) Effect of t



(f) Effect of k

Figure 5.2: TOPk-CL VS TOPk

However, in figure 5.2b, the same result is returned by both TOPk-CL and TOPk when

$d = 2$ (10 common objects). In figure 5.2d, $\forall b \geq 0.3$, the skyline size is larger than $k = 10$. However, when the skyline size is less than k, the TOPk-CL as well as the TOPk skyline return the skyline set as shown in figure 5.2d when $b = 0.1$. Figures 5.2c and 5.2e show that the confidence level may change up to 50% of the TOPk skyline results. Figure 5.2f shows that the more k increases, the more the number of common objects increases.

5.5 Conclusion

In this chapter have addressed the problem of top-k skyline with confidence level on evidential data. After introducing new semantics for the imperfect skyline analysis, we defined the top-k skyline operator which ranks the skyline results and returns the most interesting k objects. We also integrated the notion of the confidence level and we conducted an analysis about its impact on the top-k results.

Our experimental evaluation demonstrated the flexibility of the proposed top-k skyline query over evidential data and the scalability of our algorithms.

In the next chapter, we particularly tackle an important issue, namely the skyline stars (denoted by SKY^2) over the evidential data. This kind of skyline aims at retrieving the best evidential skyline objects (or the stars). Efficient algorithms have been developed to compute the SKY^2. Extensive experiments have demonstrated the efficiency and effectiveness of our proposed approaches that considerably refine the huge skyline set. In addition, the conducted experiments have shown that our algorithms significantly outperform the basic skyline algorithms in terms of CPU and memory costs.

Chapter 6

The Skyline Stars

Contents

6.1 Introduction

In this chapter, we address the issue of refining the skyline on evidential databases.

6.1.1 Motivating example

Let us recall the example presented in chapter 1, we consider again plane sensor analysis. If observation-by-observation data are considered, we can answer many questions such as: which target should be in the skyline. However, the previous studies analysing the skyline size, prove that it is worthwhile to refine this huge skyline set, i.e., among the skyline targets, which targets are considered the most dangerous.

6.1.2 Contributions

In this chapter, the following contributions are made:

- We first give a reminder about the evidential skyline models: the believable skyline (denoted by the b-skyline) and the plausible skyline (denoted by the p-skyline).
- Since the evidential skyline size is often too large to be analyzed, we define the set SKY^2 to refine the evidential skyline and retrieve the best evidential skyline objects (or the stars).
- We develop suitable algorithms based on scalable techniques to efficiently compute the b-skyline and the p-skyline, on the one hand, and the evidential SKY^2 on the other hand. We also perform an extensive experimental evaluation to demonstrate the scalability of the proposed algorithms.

These chapter contributions are published in the International Journal of Applied Soft Computing (Elmi, Tobji, Hadjali, & Yaghlane, 2017).

6.1.3 Chapter Organization

This chapter is organized as follows. In section 6.2, we recall the main notions of the believable skyline and and the plausible skyline, while in section 6.3, we propose the concept of the b-SKY^2 and p-SKY^2. Section 6.4 describe our algorithms to retrieve the skyline stars.

Details about the experimental evaluation are reported in section 6.5. Finally, section 6.6 concludes the chapter and outlines some perspectives for future work.

6.2 Evidential skyline

In this section, we present the notion of the believable skyline, and then, we introduce the notion of plausible skyline over evidential databases.

Given a set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ defined on a set of attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$, with $o_i.a_k$ denotes the *bba* of object o_i w.r.t. attribute a_k. According to Definition 2.1, the degrees of belief and plausibility that an object o_i is better than or equal (or strictly better) to another object o_j w.r.t. an attribute a_k write :

$$bel(o_i.a_k \leq o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A \leq^{\forall} B} m_{jk}(B)) \tag{6.1}$$

Where $A \leq^{\forall} B$ stands for $a \leq b$, $\forall (a, b) \in A \times B$.

$$bel(o_i.a_k < o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A <^{\forall} B} m_{jk}(B)) \tag{6.2}$$

Where $A <^{\forall} B$ stands for $a < b$, $\forall (a, b) \in A \times B$.

Example 6.1. *In Table 6.1, one can check that*[1] $bel(t_1.d \leq t_2.d) = 0$, $bel(t_1.a \leq t_2.a) = 0.3$, $bel(t_2.d < t_1.d) = 0.7$, *and* $bel(t_2.a \leq t_1.a) = 0.7$

Table 6.1: Evidential data for the skyline stars computation.

Target	Distance (10^3.km)	Altitude (10^3.km)
t_1	$\langle\{150, 160, 180\}, 0.1\rangle$, $\langle\{190, 200\}, 0.9\rangle$	$\langle 60, 0.3\rangle$, $\langle 100, 0.7\rangle$
t_2	$\langle 100, 0.7\rangle$ $\langle\Theta_{Distance}, 0.3\rangle$	$\langle\{70, 80\}, 0.8\rangle$, $\langle 80, 0.2\rangle$
t_3	70	$\Theta_{Altitude}$
t_4	$\langle\{50, 60\}, 0.8\rangle$, $\langle\{65\}, 0.2\rangle$	60
t_5	$\langle\{50\}, 0.5\rangle$, $\langle\{60\}, 0.5\rangle$	$\langle\{60\}, 0.6\rangle$, $\langle\{70\}, 0.4\rangle$

$$pl(o_i.a_k \leq o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A \leq^{\exists} B} m_{jk}(B)) \tag{6.3}$$

[1] d and a denote the distance and the altitude attributes, respectively

Where $A \leq^{\exists} B$ means for every $a \in A$, $\exists b \in B$ such that $a \leq b$.

$$pl(o_i.a_k < o_j.a_k) = \sum_{A \subseteq \Theta_{a_k}} (m_{ik}(A) \sum_{B \subseteq \Theta_{a_k}, A <^{\exists} B} m_{jk}(B)) \tag{6.4}$$

Where $A <^{\exists} B$ means for every $a \in A$, $\exists b \in B$ such that $a < b$.

Example 6.2. *In Table 6.1, one can check that* $pl(t_1.d \leq t_2.d) = 0.3$, $pl(t_1.a \leq t_2.a) = 0.3$, $pl(t_2.d \leq t_1.d) = 1$, $pl(t_2.a \leq t_1.a) = 0.7$ *and* $pl(t_5.d < t_4.d) = 0.6$

Let us now discuss how can we extend the dominance relationship to evidential data.

6.2.1 The believable skyline

In this section, we present the notion of the believable skyline. This later aims at retrieving the most interesting objects in $\mathcal{O}$ that are not believably dominated by any other objects. We first present the concept of the believable dominance denoted by the *b*-dominance and then the believable skyline denoted by the *b*-skyline.

Definition 6.1. *(The believable dominance) Given two objects* $o_i, o_j \in \mathcal{O}$ *and a belief threshold* b, o_i *b-dominates* o_j *denoted by* $o_i \succ_b o_j$ *if and only if* o_i *is believably as good or better than* o_j *in all attributes* a_k *in* $\mathcal{A}$ *(*$1 \leq k \leq d$*) and strictly believably better in at least one attribute* a_{k_0} *(*$1 \leq k_0 \leq d$*) according to a belief threshold* b*, i.e.,*
$\forall a_k \in \mathcal{A} : bel(o_i.a_k \leq o_j.a_k) \geq b$ *and* $\exists a_k \in \mathcal{A} : bel(o_i.a_k < o_j.a_k) \geq b$.

In order to define the *b*-skyline, it is essential to illustrate a key property of the *b*-dominance.

Property 6.1. *The b-dominance relationship does not satisfy the property of transitivity.*

Given an object o_i, we denote by $o_i.a_k^-$ and by $o_i.a_k^+$ respectively the minimum value and the maximum value of $o_i.a_k$.

Property 6.2. *if* $\exists a_k \in \mathcal{A}$ *where* $o_i.a_k^+ < o_j.a_k^-$ *then* o_j *does not dominate* o_i*, i.e.,* $o_j \not\succ o_i$ *since* $bel(o_j.a_k \leq o_i.a_k) = 0$.

Intuitively, an object is in the believable skyline if it is not believably dominated by another object. Based on the *b*-dominance relationship, the notion of *b*-skyline is defined as follows.

Definition 6.2. *(The believable skyline) The believable skyline of $\mathcal{O}$, denoted by b-skyline, comprises those objects in $\mathcal{O}$ that are not b-dominated by any other object, i.e., b-skyline = $\{o_i \in \mathcal{O} \mid \nexists\ o_j \in \mathcal{O}, o_j \succ_b o_i\}$.*

Property 6.3. *Given two belief thresholds b and b', if $b < b'$ then the b-skyline is a subset of the b'-skyline, i.e., $b < b' \Rightarrow$ b-skyline $\subseteq$ b'-skyline.*

Property 6.3 indicates that the size of the b-Skyline is smaller than the b'-Skyline if $b < b'$. Roughly speaking, from Property 6.3, we can see that users have the flexibility to control the size of the retrieved believable skyline by varying the believable threshold b.

6.2.2 The plausible skyline

In this section, we present the notion of the plausible skyline which retrieves the most interesting objects in $\mathcal{O}$ that are not plausibly dominated by any other objects. Note that the plausible skyline is semantically different from the believable skyline. The believable skyline returns the objects that are not surely dominated by any other object with respect to some threshold. However, the plausible skyline returns the objects that are not plausibly dominated by any other object according to a threshold. In the following, we show how they are semantically different, and later in section 6.5, we will show how this difference affects on the skyline size.

Let us first introduce the notion of the plausible dominance denoted by the p-dominance and then the plausible skyline denoted by the p-skyline.

Definition 6.3. *(The plausible dominance) Given two objects $o_i, o_j \in \mathcal{O}$ and a plausibility threshold p, o_i p-dominates o_j denoted by $o_i \succ_p o_j$ if and only if o_i is plausibly as good or better than o_j in all attributes a_k in $\mathcal{A}$ $(1 \leq k \leq d)$ and strictly plausibly better in at least one attribute a_{k_0} $(1 \leq k_0 \leq d)$ according to a plausible threshold p, i.e.,*
$\forall a_k \in \mathcal{A} : pl(o_i.a_k \leq o_j.a_k) \geq p$ and $\exists a_k \in \mathcal{A} : pl(o_i.a_k < o_j.a_k) \geq p$.

Property 6.4. *The p-dominance relationship does not satisfy the property of transitivity as well.*

Definition 6.4. *(The plausible skyline) The plausible skyline of $\mathcal{O}$, denoted by p-skyline, comprises those objects in $\mathcal{O}$ that are not p-dominated by any other object, i.e., p-skyline = $\{o_i \in \mathcal{O} \mid \nexists\ o_j \in \mathcal{O}, o_j \succ_p o_i\}$.*

Property 6.5. *Given two plausible thresholds p and p', if $p < p'$ then the p-skyline is a subset of the p'-skyline, i.e., $p < p' \Rightarrow$ p-skyline $\subseteq$ p'-skyline.*

6.3 SKY²: Skyline stars over evidential databases

In this section, we introduce the notion of evidential SKY² which allows refining the evidential skyline results (believable skyline and plausible skyline already introduced in section 6.2). To refine the objects in *b-skyline* (resp. *p-skyline*), we first compute the average score (called the dominance score and denoted by score_1) that every object S_i in the *b-skyline* (resp. *p-skyline*) dominates all objects in $\mathcal{O}$. We then compute its average score of being dominated denoted by score_2. Let us first give the definition of *b*-SKY².

6.3.1 *b*-SKY²: The believable skyline stars

In this section, we present the notions of different score functions used to compute the *b*-SKY². Let us first introduce the concept of the dominance score.

Definition 6.5. *(Believable dominance score) Given an evidential database $\mathcal{O}$, a believable dominance score, denoted by b-score$_1(S_i)$, reflects how much the skyline object S_i believably dominate the rest of objects in $\mathcal{O}$.*

$$b - score_1(S_i) = \sum_{\forall o_j \in \mathcal{O}, S_i \neq o_j, S_i \in b-skyline} bel(S_i \succ o_j) \times \frac{1}{|\mathcal{O}| - 1} \tag{6.5}$$

where $\frac{1}{|\mathcal{O}|-1}$ is used for the normalisation need and the belief degree that a skyline object S_i dominates an object $o_j \in \mathcal{O}$, denoted by $bel(S_i \succ o_j)$, is defined as follows:

$$bel(S_i \succ o_j) = \prod_{a_k \in \mathcal{A}} bel(S_i.a_k \leq o_j.a_k) \tag{6.6}$$

Example 6.3. *Let us come back to Table 6.1. Since $bel(t_2.d \leq t_1.d) = 0.7$ and $bel(t_2.a \leq t_1.a) = 0.7$, the belief degree that t_2 dominates t_1 is given by $bel(t_2 \succ t_1) = 0.7 * 0.7 = 0.49$*

Let us now introduce the concept of the being-dominated score.

Definition 6.6. *(Believable being-dominated score) Given an evidential database $\mathcal{O}$, a believable being-dominated score, denoted by b-score$_2(S_i)$, reflects how much the skyline object S_i is believably dominated by the rest of objects in $\mathcal{O}$.*

$$b - score_2(S_i) = \sum_{\forall o_j \in \mathcal{O}, S_i \neq o_j, S_i \in b-skyline} bel(o_j \succ S_i) \times \frac{1}{|\mathcal{O}| - 1} \tag{6.7}$$

where $bel(o_j \succ S_i)$ recalls the principle of equation 6.6.

One way to select the best objects from the believable skyline (i.e., the skyline stars), is to retrieve those have the highest dominance scores on the one hand, and the smallest scores of being-dominated on the other hand.

Definition 6.7. *(b-SKY²) The refined believable skyline denoted by b-SKY² comprises those objects in the b-skyline such that they have the highest values returned by b-score₁ and the smallest values returned by b-score₂. To compute the b-SKY², it suffices to compute the classical skyline over the data set S^* containing all the object S_i of b-skyline with their degrees b-score₁ and b-score₂.*

See the example bellow for more illustration.

Example 6.4. *Suppose we have the following set of believable skyline objects: b-skyline = $\{S_1, S_2, S_3, S_4, S_5\}$. For each skyline point S_i, we give its dominance score b-score₁(S_i) and its being-dominated score b-score₂(S_i) as described in Table 6.2. One can check that skyline object S_2 dominates skyline objects S_1 and S_4. Skyline objects S_2, S_3 and S_5 do not dominate each other. They thus form the skyline of $\mathcal{S}^*$. As a result, we have b-SKY²=$\{S_2, S_3, S_5\}$*

Table 6.2: The database $\mathcal{S}^*$ for the b-SKY² example

Skyline objects	$b\text{-}score_1(S_i)$	$b\text{-}score_2(S_i)$
S_1	0.16	0.71
S_2	0.65	0.4
S_3	0.5	0.23
S_4	0.23	0.69
S_5	0.6	0.3

6.3.2 p-SKY²: The plausible skyline stars

In this section, we propose how can we refine the plausible skyline results and get the skyline stars.

Definition 6.8. *(Plausible dominance score) Given an evidential database $\mathcal{O}$, a plausible dominance score, denoted by p-score₁(S_i), expresses how much the skyline objects plausibly dominate the rest of objects in $\mathcal{O}$.*

$$p - score_1(S_i) = \sum_{\forall o_j \in \mathcal{O}, S_i \neq o_j, S_i \in p-skyline} pl(S_i \succ o_j) \times \frac{1}{|\mathcal{O}| - 1} \tag{6.8}$$

where the degree of plausibility that a skyline object S_i dominates an object $o_j \in \mathcal{O}$, denoted by $pl(S_i \succ o_j)$, is defined as follows:

$$pl(S_i \succ o_j) = \prod_{a_k \in \mathcal{A}} pl(S_i.a_k \leq o_j.a_k) \tag{6.9}$$

Example 6.5. *From Table 6.1, we have $pl(t_2.d \leq t_1.d) = 1$ and $pl(t_2.a \leq t_1.a) = 0.7$. The plausibility that t_2 dominates t_1 is then given by $pl(t_2 \succ t_1) = 1 * 0.7 = 0.7$. One can also check that $pl(t_1 \succ t_2) = 0.3 * 0.3 = 0.09$*

Let us now introduce the concept of the being-dominated score.

Definition 6.9. *(Plausible being-dominated score) Given an evidential database $\mathcal{O}$, a plausible being-dominated score, denoted by p-score$_2(S_i)$, expresses how much the skyline objects are plausibly dominated by the rest of objects in $\mathcal{O}$.*

$$p - score_2(S_i) = \sum_{\forall o_j \in \mathcal{O}, S_i \neq o_j, S_i \in p-skyline} pl(o_j \succ S_i) \times \frac{1}{|\mathcal{O}| - 1} \tag{6.10}$$

Definition 6.10. *(p-SKY^2) The refine plausible skyline, denoted by p-SKY^2, comprises those objects in the p-skyline such that they have the highest values returned by p-score$_1(S_i)$ and the smallest values returned by p-score$_2(S_i)$. To compute the p-SKY^2, it suffices to compute the classical skyline over the data set S^{**} containing all the object S_i of p-skyline with their degrees p-score$_1$ and p-score$_2$.*

6.4 Skyline Stars Computation

In this section, we first discuss how our evidential SKY^2 algorithms can highlight the properties of dominance relationship mentioned in section 3. Then, we propose an efficient method to reduce the complexity of belief function operations as well as the plausibility function operations.

6.4.1 *b*-SKY^2 computation

In this section, we describe how can we obtain the skyline stars. In a first part, we describe the *b*-SKY^2 algorithm which returns from the *b-skyline* the skyline stars.

A straightforward algorithm to compute the *b-skyline* is to compare each object o_i against the other objects. If o_i is not *b*-dominated, then it belongs to the believable skyline. However, this approach results in a high computational cost (see Section 6.5).

Also, while the b-dominance relationship is not transitive, (see Property 6.1), an object cannot be eliminated from the comparison even if it is b-dominated since it will be useful for eliminating other objects.

For this reason, we propose an algorithm (see Algorithm 13) that follows the principle of the two scan algorithm cited in (Chan et al., 2006) in order to efficiently compute the evidential skyline.

First, algorithm 13 computes the believable skyline through two phases. In the first phase (lines 2–20), we compare each object o_i in $\mathcal{O}$ with those selected in b-*skyline* and we compute the several scores attributed for each object. If an object o_j in b-$Sky_{\mathcal{O}}$ is b-dominated by o_i, we remove o_j from the set of candidate objects since it can not be a part of the believable skyline. At the end of the comparison of o_i with the objects of b-*skyline*, if o_i is not b-dominated by any object then, it is added to b-*skyline* with two scores ($b.\text{score}_1$ and $b.\text{score}_2$).

To avoid the situation of non-transitivity, a second phase is needed (lines 21–24). To determine if an object o_i in b-*skyline* is indeed in the believable skyline it is sufficient to compare o_i with those in $\mathcal{O} \setminus \{b\text{-}skyline \cup undom(o_i) \cup \{o_i\}\}$ that occur earlier than o_i since the other ones have been already compared against o_i, where $undom(o_i)$ is the set of objects that occur before o_i and that do not b-dominate o_j.

In lines 19 and 20, we compute the scores of each skyline object o_i. b-score$_1$ is recovered by summing the belief degrees that o_i b-dominates $o_j \in \mathcal{O}$ where $o_i \neq o_j$. Even, if b-SKY^2 algorithm minimizes the number of dominance checks, it also may result in a high computational cost. In particular, when the average number of focal elements per object is large. Thus, it is crucial to more optimize the dominance checks to improve the performance of b-SKY^2. In the following, we devise an efficient method that overcomes this problem using the minimum and the maximum values of each object w.r.t. each attribute, according to Property 6.2. To determine if an object o_i b-dominates another object o_j, Property 6.2 shows that it is not necessary to iterate all focal elements of each bba. To efficiently check that a given object o_i b-dominates or not another object o_j, it suffices to compare for each attribute $a_k \in \mathcal{A}$, $o_i.a_k^-$ against $o_j.a_k^+$.(see Algorithm 14) which is based on property 6.2. The method returns false (in line 4), if there is any attribute a_k for which $o_i.a_k^- > o_j.a_k^+$; since o_i cannot b-dominate o_j according to Property 6.2. Otherwise, if the belief degree that $o_i.a_k$ is as good or better than $o_j.a_k$, is less than the threshold b, then o_i does not b-dominate o_j (line 6). The method returns "false" as soon as it does not exist an attribute a_{k_0} where o_i is strictly better than o_j (line 7–10).

Algorithm 13: b-SKY^2

Input: Objects $\mathcal{O}$; belief threshold b;
Output: b-SKY^2;

```
1  begin
2      foreach o_i in O do
3          isSkyline ← true;
4          s_1 ← 0;
5          s_2 ← 0;
6          foreach o_j in b-skyline do
7              if isSkyline then
8                  if o_j ≻_b o_i then
9                      isSkyline ← false;
10                 else
11                     undom(o_i) ← undom(o_i) ∪ {o_j};
12                     s_2 ← s_2 + bel(o_j ≻ o_i);
13             if o_i ≻_b o_j then
14                 remove o_j from b-skyline;
15         if isSkyline then
16             insert o_i in b-skyline;
17             foreach o_x in O do
18                 s_1 ← s_1 + bel(o_i ≻ o_x);
19             o_i.b-score_1 ← s_1/(|O| − 1);
20             o_i.b-score_2 ← s_2/(|O| − 1);
21     foreach o_i in b-skyline do
22         foreach o_j in O \ (b-skyline ∪ undom(o_i) ∪ {o_i}), pos(o_j) < pos(o_i) do
23             if o_j ≻_b o_i then
24                 remove o_i from b-skyline;
25     b-SKY^2 ←SKY-OF(b-skyline);
26     return b-SKY^2;
```

The algorithm denoted b-SKY^2 has the complexity of $O(n^2)$ where n is the number of evidential objects.

The method SKY-OF() described in algorithm 15 aims at retrieving the skyline stars which have the highest dominance score and the smallest being-dominated score. As shown in algorithm 15, this set of stars is computed by a simple classic skyline on a data set with

Algorithm 14: b-dominates(o_i, o_j, b)

```
1  strict ← False;
2  while a_k in A and strict=False do
3      if o_i.a_k^- > o_j.a_k^+ then
4          return false;
5      if bel(o_i.a_k ≤ o_j.a_k) < b then
6          return false;
7      if bel(o_i.a_k < o_j.a_k) ≥ b then
8          strict ← True;
9  if strict = False then
10     return false;
11 return true;
```

Algorithm 15: SKY-OF(b-*skyline*)

```
1  Sorted-SKY ← SORT(b-skyline, score_1);
2  pred ← 1;
3  foreach S_i in Sorted-SKY do
4      if S_i.score_2 ≤ pred then
5          pred ← S_i.score_2 ;
6          SKY^2 ← SKY^2 ∪ {S_i} ;
7  return SKY^2;
```

only two dimensions (i.e., b-score$_1$ and b-score$_2$).
A two-dimensional skyline can be computed by sorting the data (line 1 in algorithm 15). If the data is topologically sorted according to one attribute of the database (here in line 1, we sort the skyline points according to the attribute score$_1$), the test of whether an object is a part of the skyline stars does not result in a high cost: we simply need to compare an object with its predecessor. More precisely, we need to compare an object with the last previous object which is part of the skyline stars. Table 6.3 illustrates this approach. Note that the skyline objects are sorted according to the first attribute, i.e., $b\text{-}score_1(S_i)$. So, S_2 can be eliminated because it is dominated by S_1, its predecessor. Likewise, S_3 can be eliminated because it is dominated by S_1, its predecessor after S_2 has been eliminated.

Table 6.3: The database $\mathcal{S}^*$

Skyline objects	$b\text{-}score_1(S_i)$	$b\text{-}score_2(S_i)$
S_1	0.9	0.3
S_2	0.8	0.5
S_3	0.6	0.4
S_4	0.4	0.2

6.4.2 p-SKY^2 computation

To compute the plausible skyline stars, we refer to the p-SKY^2 algorithm. In this algorithm, we follow the same steps as in b-SKY^2 algorithm using an efficient method to compute the p-dominance between objects in $\mathcal{O}$.

In fact, in algorithm 16, the p-dominance method, denoted by $\succ_p$, can recall the principle of the b-dominance discussed in algorithm 14. However, computing the plausibility function can result in a high cost computation because it requires the intersection operations between two *bbas*. In the following, we propose a way to reduce this cost computation. To detect the intersections between two *bba*, it is necessary to iterate all propositions in each focal element.
In the following, we represent a focal element as a decimal number and we reduce the set operations to bit-wise operations[2]. The plausibility function can be then equivalently expressed as follows:

$$pl(A) = \sum_{A\&B\neq 0} m(B) \tag{6.11}$$

where A and B are two decimal numbers and & stands for the logic operator "and".
In this context, we denote the frame of discernment of an attribute a_k as $\Theta_{ak} = \{p_1, p_2, \ldots, p_n\}$ and $\{p_1, p_2, \ldots, p_n\}$ represent the set of propositions in Θ_{ak}. A focal element f can have a relational representation $\mathcal{R}$ where $\mathcal{R}$ is a binary number such as $\mathcal{R} = r_1 r_2 \ldots r_n$ where $r_i = \begin{cases} 1 & if \quad p_i \in f \\ 0 & if \quad p_i \notin f \end{cases}$
However, it is necessary to fix the order of propositions in Θ_{ak} to have a one-to-one correspondence between propositions and n-bits binary numbers.

Example 6.6. *Let* $\Theta_{ak} = \{tall, short, small, big\}$. *The following focal element* $f = \{tall, short, small, big\}$ *can be equivalently represented by the binary representation* $\mathcal{R} =$

[2]This approach is introduced by (Haenni & Lehmann, 2003) for the purpose of reducing the evidential functions computation.

1111. *This binary code can be converted to decimal code denoted by cod such that* $f.cod = (2^0 * 1) + (2^1 * 1) + (2^2 * 1) + (2^3 * 1) = 15$. *Suppose now we have the focal element* $f' = \{tall, big\}$, *its representation as a binary code is* $\mathcal{R} = 1001$. *This later can be converted to decimal as follows:* $1001 = (2^0 * 1) + (2^1 * 0) + (2^2 * 0) + (2^3 * 1) = 9$. *Thus* $f'.cod = 9$.
In order to detect if there is an intersection between the focal elements f *and* f', *it is just sufficient to check if* $(f.cod \& f'.cod)$ *equals or not to zero.*

The correspondence between subsets and decimal codes lies not only at the format but also the operations and relations. Bit-wise operations include intersection AND (&), union OR and complement.

Algorithm 16: p-SKY^2

Input: Objects $\mathcal{O}$; plausible threshold p;
Output: p-SKY^2;

```
1  begin
2      foreach o_i in O do
3          isSkyline ← true;
4          s_1 ← 0;
5          s_2 ← 0;
6          foreach o_j in p-skyline do
7              if isSkyline then
8                  if o_j ≻_p o_i then
9                      isSkyline ← false;
10                 else
11                     undom(o_i) ← undom(o_i) ∪ {o_j};
12                     s_2 ← s_2 + pl(o_j ≻ o_i);
13             if o_i ≻_p o_j then
14                 remove o_j from p-skyline;
15         if isSkyline then
16             insert o_i in p-skyline;
17             foreach o_x in O do
18                 s_1 ← s_1 + pl(o_i ≻ o_x);
19             o_i.p-score_1 ← s_1/(|O| − 1);
20             o_i.p-score_2 ← s_2/(|O| − 1);
21     foreach o_i in p-skyline do
22         foreach o_j in O \ (p-skyline ∪ undom(o_i) ∪ {o_i}), pos(o_j) < pos(o_i) do
23             if o_j ≻_p o_i then
24                 remove o_i from p-skyline;
25     p-SKY^2 ←SKY-OF(p-skyline);
26     return p-SKY^2;
```

The algorithm denoted p-SKY^2 has the complexity of $O(n^2)$ where n is the number of evidential objects.

With relational representation, we can immediately reduce the complexity of plausibility function (see algorithm 17 where $f.mass$ and $f'.mass$ are the masses attributed for the focal elements f and f', respectively.) and hence reduce the complexity of p-SKY^2 algorithm.

We show in section 6.5, that the results clearly indicate that this method significantly save the cost of the plausibility function computation.

Algorithm 17: $pl(o_i.a_k \leq o_j.a_k)$

```
1 pl ← 0;
2 foreach f in o_i.a_k do
3     foreach f' in o_j.a_k do
4         if (f.cod & f'.cod) ≠ 0 then
5             pl ← pl + (f.mass * f'.mass);
6 return pl;
```

The complexity of b-SKY^2 and p-SKY^2 is of the order of O(n^2) where n is the database size. However, algorithm complexity copes with the worst algorithm scenario (a complete database scan). That is why we conduct in section 6.5 an extensive experimentation evaluation, where we consider several aspects of the evidential database to mainly assess the performance of our methods.

6.5 Experimental Evaluation

In this section, we have performed an extensive experimental evaluation of the proposed framework. We report empirical study to examine the effectiveness and the efficiency of computing the skyline stars over evidential data. More specifically, we focus on two issues: (i) the size of the skyline stars; and (ii) the scalability of our proposed techniques for computing the skyline stars. For comparison purposes, we implement the baseline algorithms for b-SKY^2 and p-SKY^2 referred to as BBS and BPS, respectively. In addition, we implement two efficient methods b-dominates function and p-dominates function to show how do they perform the b-SKY^2 and the p-SKY^2 algorithms, respectively.

6.5.1 Experimental Setup

We control the generation of evidential data by the parameters provided in Table 6.4, which lists the parameters and their default values. In each experimental setup, we evaluate the effect of one parameter, while we set the other to their default values.
For the number of focal elements, the values listed in Table 6.4, refer to the maximum number of focal elements that a *bba* may contain. This means that the number of focal

Table 6.4: Examined Values for the Skyline Stars Computation.

Parameter	Symbol	Values	Default
Number of objects	n	1K, 2K, 5K, 8K, 10K, 50K, 100K, 500K	10K
Number of attributes	d	2, 3, 4, 5, 6, 10, 15	3
Max Nbr of focal elmts/attr	f	2, 3, 4, 6, 8, 9, 10, 15, 20	5
belief threshold	b	0.01, 0.1, 0.3, 0.5, 0.7, 0.9	0.5
plausible threshold	p	0.5, 0.6, 0.7, 0.8, 0.9	0.7
Theta cardinality/attr	t	10, 50, 100, 150, 200	100

elements is in $[1..f]$.
In addition, we mean by the theta cardinality, the number of possible propositions for each attribute.

We refer to the work introduced by (Bousnina et al., 2016) to design the evidential database. The data generator and the algorithms, i.e., BBS, BPS, b-SKY^2 and p-SKY^2 were implemented in Java and all experiments were conducted on a 2.3 GHz Intel Core i7 processor, with 6GB of RAM.
Note that the baseline algorithms do not use the methods b-dominates() and p-dominates() and iterate on all the propositions in a focal element to compute the dominance degrees between objects.

6.5.2 Size of the Skyline Stars

Skylines are generally of large size for large sets of objects as well as attributes $\mathcal{A}$. It is larger if the attributes values are uncertain.

Table 6.5: Belief Dominance

Objects	o_1	o_2	o_3	o_4
o_1	-	0.2	0.2	0.3
o_2	0.4	-	0.1	0.3
o_3	0.1	0.3	-	0.2
o_4	0.1	0.1	0.15	-

Table 6.6: Plausible Dominance

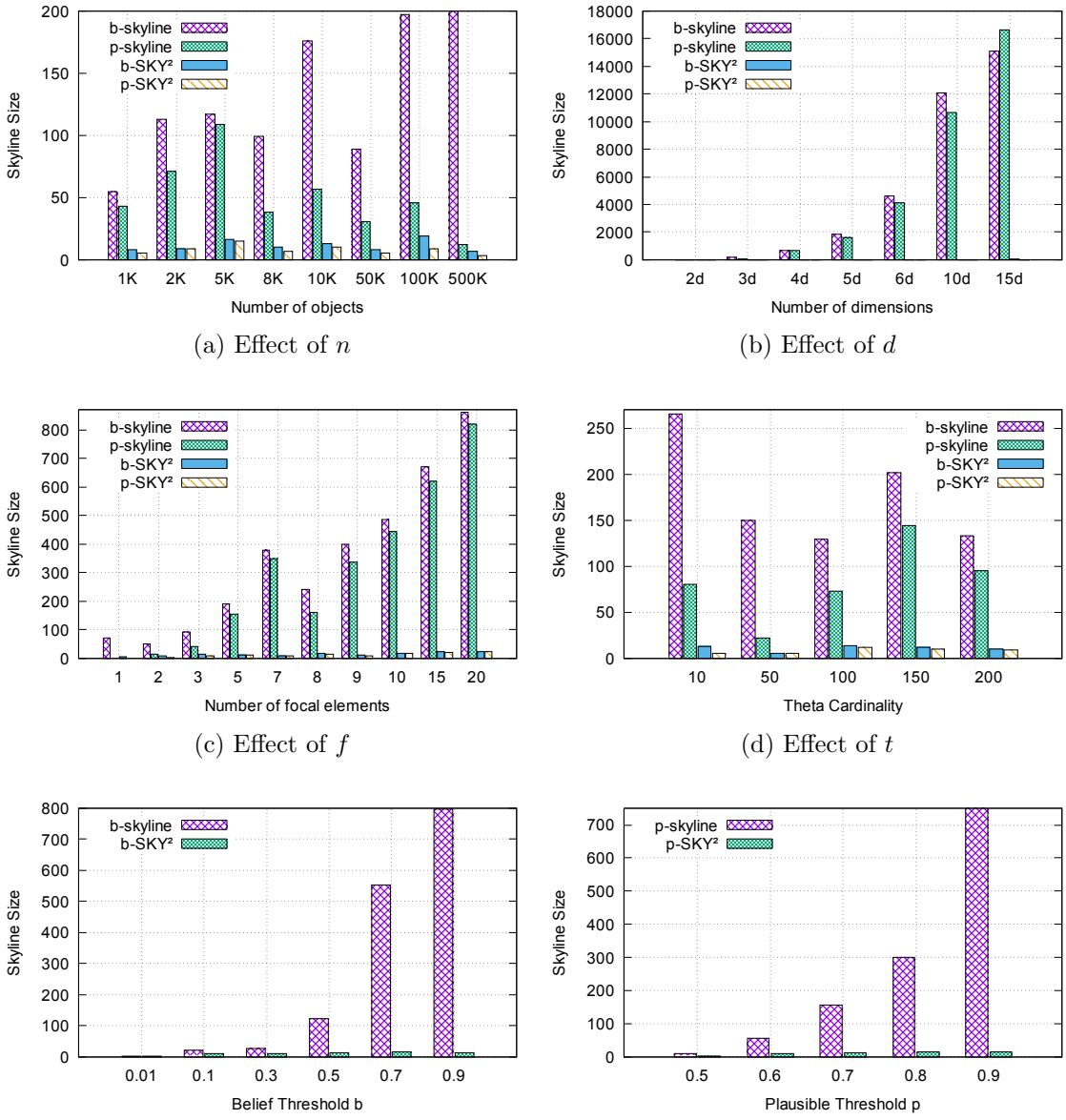
Objects	o_1	o_2	o_3	o_4
o_1	-	0.3	0.25	0.4
o_2	0.4	-	0.2	0.4
o_3	0.2	0.5	-	0.5
o_4	0.35	0.4	0.15	-

The goal of the experiments in this section is twofold. First, we demonstrate that using p-skyline to model the skyline results over evidential data is more significant than the b-skyline in terms of size. Second, we show that the reduction of query result size is more and more significant if we compute the p-SKY^2 and the b-SKY^2. More specially, the p-SKY^2 size is smaller than the b-SKY^2 size.

First of all, let us show how the p-skyline results can be more significant in terms of size than the b-skyline results. Suppose we have four evidential objects in $\mathcal{O}$ such that $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$. We study the believable dominance and the plausible dominance of each object o_i in $\mathcal{O}$. Table 6.5 shows the belief degrees that each object in lines dominates another object in columns. Table 6.6 shows the plausibility degrees that each object in lines dominates another object in columns. As we can observe, the degrees returned by the plausible dominance are either equal or larger than those returned by the belief dominance. That is because, as we have shown in Table 6.5 and Table 6.6, the plausibility function gives more chance to objects to dominate the others. For example, o_1 0.2-believably dominates o_2, however, it 0.3-plausibly dominates o_2. Let us compute the skyline objects with the following belief and plausible thresholds: $b = p = 0.3$
In Table 6.5, o_1 can not be in the b-skyline since $o_2 \succ_{0.4} o_1$. However, o_2, o_3 and o_4 are not 0.3-dominated by any other object. Thus, b-skyline $= \{o_2, o_3, o_4\}$.
In Table 6.6, o_2 can not be in the p-skyline since $o_3 \succ_{0.5} o_2$ and $o_4 \succ_{0.4} o_2$. As well as o_4 can not be in the p-skyline since $o_1 \succ_{0.4} o_4$, $o_2 \succ_{0.4} o_4$ and $o_3 \succ_{0.5} o_4$. Thus, the p-skyline only contains o_3 since it is not plausibly dominated by any other object with p=0.3 and thus the p-skyline= $\{o_3\}$.

Fig. 6.1 shows the size (i.e., the number of objects returned) of the b-skyline, p-skyline, b-SKY^2 and p-SKY^2 w.r.t. n, d, b, p, t and f. All figures show the difference in the size of the results of p-skyline and b-skyline. In all cases, the p-skyline returns less objects than the b-skyline.

Add to that, as the figures suggest, using the b-SKY^2 and the p-SKY^2 has a significant



(a) Effect of n (b) Effect of d

(c) Effect of f (d) Effect of t

(e) Effect of b (f) Effect of p

Figure 6.1: Size on the skyline stars

reduction in the size of skyline query results, in comparison with b-skyline and p-skyline, respectively. In particular, the p-SKY^2 algorithm returns less objects in the skyline stars set. For example, in Figure 6.1a, for a high size of data set (n=500K), the b-skyline algorithm returns 0.04% of the database size (200 objects). However, the b-SKY^2 returns only 0.002% of the database size (less than 10 objects). In addition, the p-skyline algorithm returns 0.003% (15 objects) of the database size. However, the b-SKY^2 returns only 0.001%

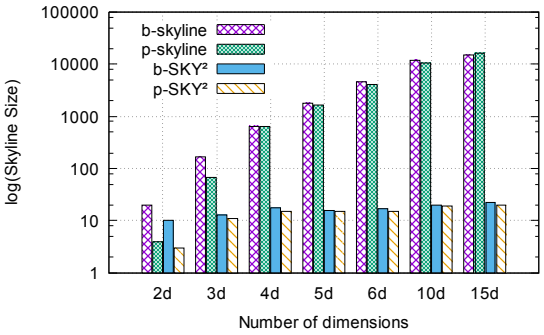


Figure 6.2: Effect of d on Size.

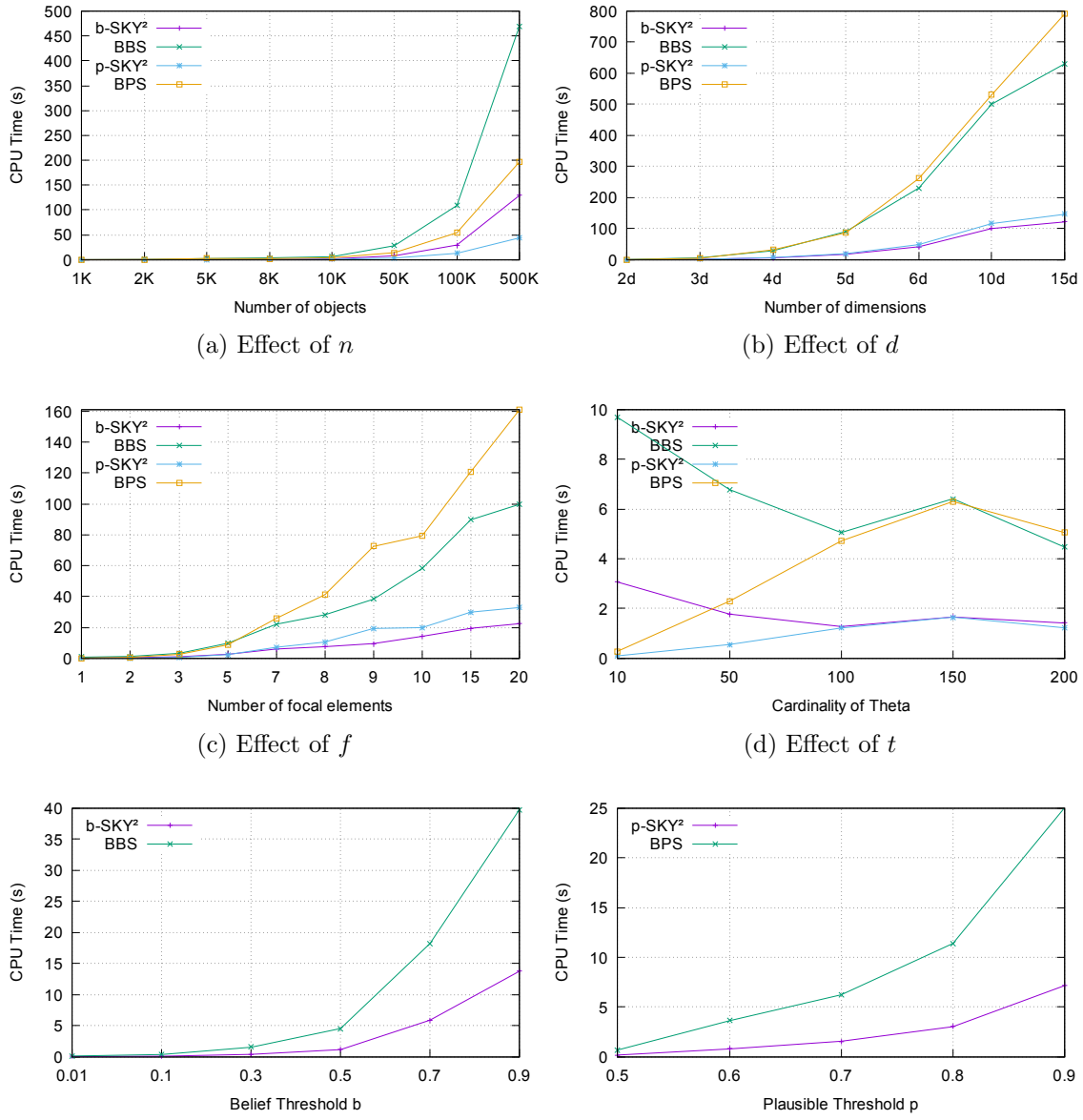
(5 objects), which is interesting for the user.

In general and in all cases, the p-SKY^2 retrieves the best result size for the user (the smallest size). It can be observed in Fig. 6.1b, because using larger sets of attributes $\mathcal{A}$ generally results in a high result size. In this figure, as the linear scale does not reflect the difference on the skyline size between algorithms, we present data on a logarithmic scale in Fig. 6.2. The cardinality of the b-skyline and p-skyline increases significantly with the increase of d. This is because with the increase of d an object has better opportunity to be not dominated in all attributes. However, it does not affect the p-SKY^2 size because for larger attribute sets, p-SKY^2 size grows slowly. The same observation can be done for the f effect in Fig. 6.1c and t effect in Fig. 6.1d.
Another important observation in Fig. 6.1a, is that when varying n, the p-SKY^2 size is always smaller than the b-skyline size, the p-skyline size and also the b-SKY^2 size. Moreover, the figure 6.1c shows that the size of the b-skyline and the p-skyline increases with higher values of n since when n increases, more objects have chances to be not dominated.

Fig. 6.1f shows that the size of the p-skyline increases with the increase of the p since the p-skyline contains the p'-skyline if $p > p'$; see Theorem 6.5. In contrast to d, p and b, f and t have no apparent effect on the size of the b-skyline and the p-skyline as shown in Fig. 6.1c and Fig. 6.1d since varying f and t, some objects have better chances to be not dominated, while other have better chances to be dominated.

6.5.3 Performance and Scalability

To show the efficiency and the scalability of our algorithms, four algorithms are evaluated (the b-SKY^2, the p-SKY^2 and their baseline algorithms referred to BBS and BPS) w.r.t. n, d, b p, t, and f.



(a) Effect of n (b) Effect of d

(c) Effect of f (d) Effect of t

(e) Effect of b (f) Effect of p

Figure 6.3: Elapsed time to compute the skyline stars

Fig. 6.3 investigates the run-time of the algorithms with respect to the plausible and belief threshold. The results clearly indicate that the pruning techniques in those two methods significantly reduce the cost of the skyline stars computation.

Overall, p-SKY^2 outperforms BBS and BPS. Fig. 6.3a shows that the performance of the algorithms deteriorates with the increase of n. In Fig. 6.4, we use the logarithmic scale to well perceive the effect of n. Observe that p-SKY^2 is one order of magnitude faster

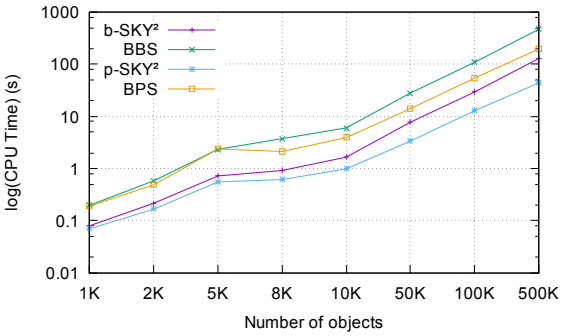


Figure 6.4: Effect of n on CPU Time

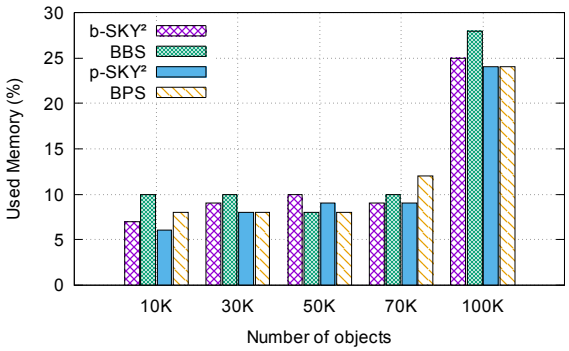
than b-SKY^2, BPS and BBS since it can quickly identifies if an object is dominated or not. Moreover, p-SKY^2 performs b-SKY^2 since the p-dominates() method reduces intersection operations to bit-wise operations which improves the computation cost.

According to Fig. 6.3b, the running time of the algorithms increases when the number of dimensions increases. Moreover, BBS and BPS do not scale with d. This is because when d increases the size of the skyline stars becomes larger, thus a large number of objects will be selected for the the second phase. Hence, BPS performs a large number of dominance checks with a basic function. Even if the p-SKY^2 performs the same number of dominance checks than BPS, p-SKY^2 is more efficient than BPS since it can detect immediately whether an object dominates or not another.

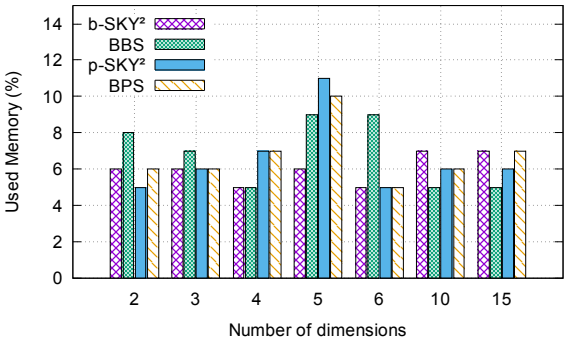
As shown in Fig. 6.3e and Fig. 6.3f, the more b and p increase, the more the computation cost increases. That is because there is more and more dominance checks. The four algorithms are not affected by t as the skyline stars computation is independent of the number of propositions for each attribute.

Fig. 6.3c shows that the execution time of the algorithms p-SKY^2 and b-SKY^2 slightly increase with the increase of f. It is not the case of the BBS and BPS as the baseline algorithms must iterate all focal elements to check if an object dominates other objects or not.

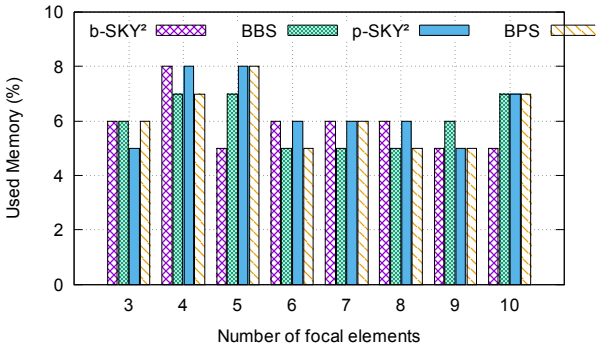
Fig. 6.5 shows the same experiments, but reports the memory cost instead. As we show in Fig. 6.5a, we can note that there is a slight difference in term of memory cost between b-SKY^2 and p-SKY^2. In addition and as expected, larger cardinality increases the percent of used memory because the higher the number of objects, the sparser the data set. However, varying d has no effect on used memory as shown in Fig. 6.5b. The same if varying the parameter f.



(a) Effect of n



(b) Effect of d



(c) Effect of f

Figure 6.5: Used Memory

6.6 Conclusion

In this chapter, we have addressed the problem of selecting skyline stars over evidential data. We introduced new semantics for the skyline analysis over evidential data and we developed efficient algorithms for computation purpose.

Our experimental evaluation demonstrated the interest of the proposed evidential skyline stars and the scalability of our algorithms. In the context of frequently updated database, new methods of maintaining the skyline stars, without restarting the analysis process from the scratch, should be introduced. Thus, the skyline stars computation if the queried database is frequently updated, is left for future work.

Also, another interesting future direction is to include the level confidence of an evidential object in the skyline stars computation in the spirit of (Yong et al., 2014). The confidence level represents the uncertainty of the tuple level in a database.

In the next chapter, we conclude this thesis by summarizing the achieved contributions

and outlining some future works.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

It has been recognized that the Skyline analysis rapidly gains popularity and constituting an integral part of many real-world applications. Therefore, enhancing the capabilities of the current Skyline operator engines with effective and efficient techniques for the Skyline retrieval and selection is an important issue.

In this dissertation, we provided optimization strategies to enable users to select the most interesting database objects in a flexible way based on their preferences. Mainly, we have discussed some aspects related to the skyline analysis over imperfect data modeled by the belief function theory. In addition, we have deeply study the effect of some parameters on the skyline size on the one hand, and the execution time on the other hand, when data are uncertain, imprecise and incomplete. We summarize below our major contributions:

- The evidential skyline: we modeled the skyline operator over imperfect data modeled by the evidence theory. We proposed how can we compute the dominance relationship between the objects of an evidential database and to retrieve the objects that are not dominated by any other objects, i.e., the skyline objects. In particular, we define a first evidential skyline model which is based on the classic skyline definition and second, we propose new semantics for the evidential skyline where objects are considered as a set of knowledge states. Furthermore, we provided new techniques to efficiently compute the skyline on evidential data.
- The evidential skyline maintenance: We proposed efficient methods to maintain the skyline results in the evidential database context when an object is inserted or deleted.

In addition, we performed an extensive experimental evaluation to demonstrate the scalability of the proposed algorithm.

- The Top-k skyline: We proposed a novel ranking criterion based on the evidential dominance relationship, to select the most relevant objects in the evidential database. We proposed a score function reflecting the dominance degree of each object. This score function aims at retrieving k objects that are expected to believably dominate the more the other objects. In a second step, we developed a suitable algorithm to the evidential skyline computation and the top-k query. We evaluated our approach through a set of thorough experiments. In addition, our extensive experiments reflects the impact of the confidence level on the top-k skyline result.

- The distributed evidential skyline: We then introduced efficient approach for querying and processing the evidential skyline over multiple and distributed servers. We proposed the notions of the marginal points to resume the dominance regions of each local evidential skyline. We the developed efficient algorithms to compute the global evidential skyline. We conducted extensive experiments to show the efficiency and the effectiveness of our approach.

- The skyline stars: Since the evidential skyline size is often too large to be improved, we define the set SKY^2 to refine the evidential skyline and retrieve the best evidential skyline objects (or the stars). In addition, we developed suitable algorithms based on scalable techniques to efficiently compute the evidential SKY^2. We evaluated our approach through a set of experiments.

7.2 Future Work

This dissertation leads to various fertile grounds for future research. In the following, we present some ideas which show the direction that future research could follow.

- Interestingly enough, many real-life applications where uncertain, imprecise, noisy and error-prone data inherently exist. By the advent of such applications, the support of advanced analysis queries such as the skyline for imperfect data has become important. Since location based services and GPS devices can easily connect users and make groups, the skyline queries can not answer a users group needs and are not sufficient to obtain a good choice for all group members. As future work, we can propose an imperfect spatial skyline query for group of users located at different positions. For example, if a group wants to find a restaurant, it is important to select

the interesting place for all users of the group. Thus, the skyline computation under this constraint is an interesting future direction.

- Context is an important concept to customize the objects selection. It is thus interesting to consider the context in the object skyline selection. I addition, taking account a user-defined satisfaction functions in the skyline computation seems an interesting direction to rank the objects retrieved by the evidential skyline .
- An interesting future direction is to introduce the notion of confidence level to b-skyline computing and updating in the spirit of (Yong et al., 2008).

Appendix A

The Evidential Skyline System (eSKY)

A.1 Introduction

This annex presents an Evidential Skyline System (eSky) designed to extract the most interesting (skyline) objects over imperfect data. Such data are managed in the evidential databases. eSky proposes a graphical interface including five main modules: (1) Computing the skyline objects over evidential data, (2) Maintaining the skyline set (3) Computing the global skyline from the distributed environments and (4) Retrieving the top-k skyline objects and (5) Retrieving the skyline stars. We also show that the efficiency we achieved is good enough to be used in practice.

A.2 eSKY'S Architecture

The main technical problems addressed by the system eSKY, depicted in Figure A.1, are described in the following.

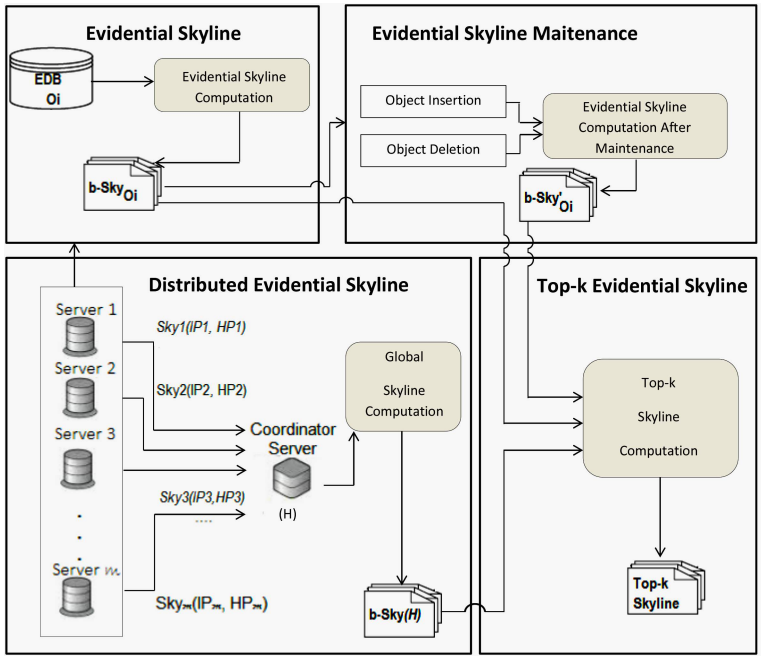


Figure A.1: eSKY's Architecture

A.2.1 Evidential Skyline Computation (ESC)

ESC component is designed to return the skyline objects from an evidential database. This component associates to each object, a belief degree of dominance. The believable dominance, denoted by b-dominance, expresses the extent to which o_i dominates o_j, were defined in the previous chapters.

A.2.2 Evidential Skyline Maintenance (ESM)

The role of ESM component is to compute the new skyline of frequently updated evidential databases. In particular, this component provides the maintaining of the evidential skyline in the two main cases: the object insertion and the object deletion. ESM component is used to efficiently maintain the skyline set. A straightforward method to find the new skyline after insertion (resp.deletion) is to recompute from scratch the b-Sky$_{\mathcal{O}}$, represented by the Basic Maintenance After Insertion Algorithm BMAI (resp. Basic Maintenance After Deletion Algorithm BMAD). However, this approach incurs a high computational cost. To avoid this cost, we have provided in chapter 4, an optimization technique to recompute the skyline. This technique allows decreasing the number of dominance checks.

A.2.3 Distributed Evidential Skyline (DES)

DES component allows users to efficiently compute the evidential skyline from multiple and distributed sources.

Given a set of m distributed servers $S = \{S_1, S_2, .., S_m\}$, each possessing an evidential database $\mathcal{O}_i$ $(1 \leq i \leq m)$ and a coordinator server $\mathcal{H}$ which is responsible for the final execution query and for the global skyline computation as shown in Figure A.1.

DES component is based on ESC component to compute the evidential skyline over independent local databases $\mathcal{O}_i$. A key property gives $b\text{-}SKY(\mathcal{H}) = b\text{-}SKY(\cup_{1\leq i\leq m} b\text{-}SKY(\mathcal{O}_i))$. But, this approach results in a high computational cost. In chapter 4, we have presented new techniques to improve our algorithms.

A.2.4 The Top-k Evidential Skyline (TES)

As shown in Figure A.1, all the result sets of the others components can be an input for TES component to rank the skyline set and retrieve the k skyline objects that are expected

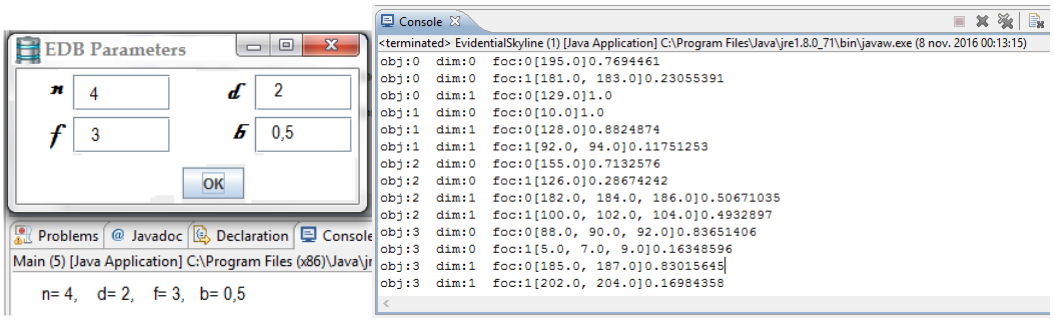
to have the highest score of dominance.

For each object S_i in the b-skyline, we need to know how much does it dominate others objects in the b-skyline set. To this end, we introduce the notion of the object dominance score, denoted by $\mu(S_i)$ as described in chapter 5.

A.2.5 The Evidential Skyline Stars (ESS)

The evidential skyline set computed by the ESC component can be an input for ESS component to retrieve the best evidential skyline objects called the skyline stars and denoted by SKY^2 since there is a double skyline ranking. Our proposed approaches considerably refine the huge skyline set. The conducted experiments have shown in chapter 6.

A.3 Demo Scenarios



(a) EDB Parameters (b) Generated EDB

Figure A.2: EDB Generation

We illustrate in Figure A.2 how users can generate the evidential database and control the generation by the parameters in Figure A.2a. n is the number of objects in the evidential database, d is the attributes number, f presents the number of focal elements in a basic belief assignment bba, and b is the belief threshold to compute the b-dominance.

Figure A.3, presents the user interface of the Evidential Skyline system. Users choose one of the component already described in section A.2. The system components ESM and TES are not active until the users compute the skyline over an independent or multiple databases.

DEMO 1: A straightforward algorithm to compute the b-skyline for the ESC component

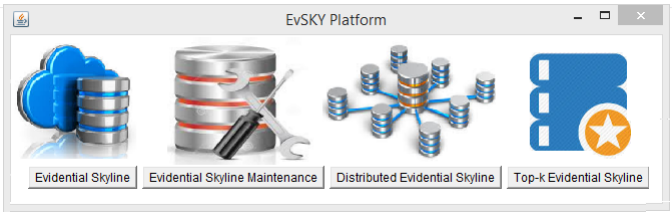


Figure A.3: eSKY Platform

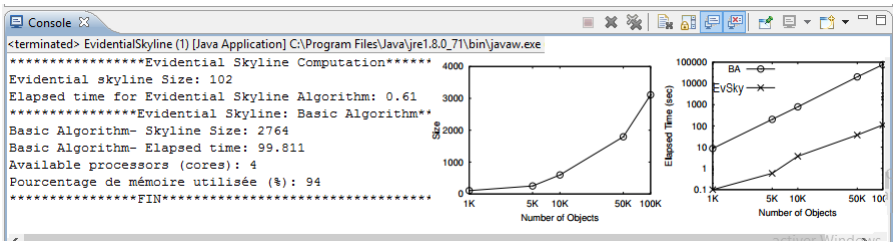
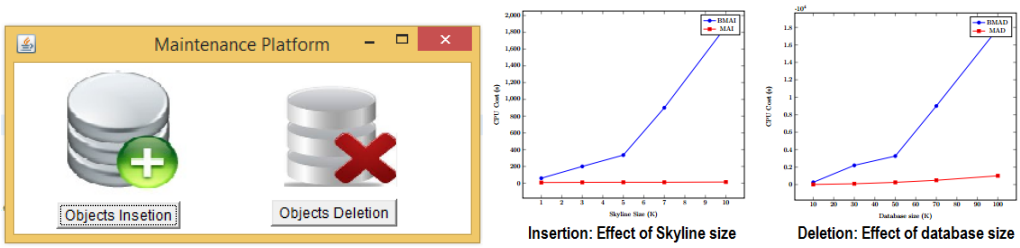


Figure A.4: Evidential Skyline Computation

(denoted by algorithm BA), is to compare each object o_i against the others objects. If o_i is not b-dominated, then it belongs to the evidential skyline. However, this approach results in a high computational cost as it needs to compare each object with every others. For this reason, we propose a two phase algorithm (denoted by EvSky) that follows the principle of the two scan algorithm (Chan et al., 2006). As shown in Figure A.4, we mainly evaluate the effect of the parameter n (number of objects in $\mathcal{O}$) on the skyline size and the execution time.



(a) Maintenance Platform (b) Experiments

Figure A.5: Evidential Skyline Maintenance

DEMO2: In Figure A.5, we present the maintenance platform in the main following cases: object insertion and object deletion. Figure A.5b illustrates that the algorithms (MAI (Maintenance After Insertion) and MAD (Maintenance After Deletion)) using the proposed pruning strategies outperform the basic algorithms (BMAI and BMAD, respectively.)

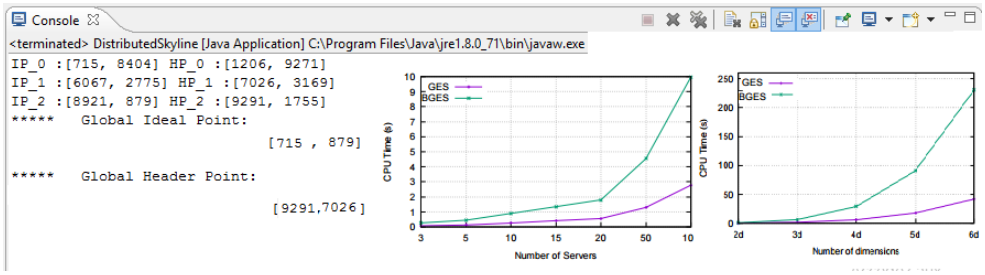


Figure A.6: Distributed Evidential Skyline

DEMO 3: As shown in Figure A.6, the coordinator server $\mathcal{H}$ has as an input the local marginal points of the different database $\mathcal{O}_i \forall i \in \{1..m\}$. The DES component computes the global marginal points and retrieves the global skyline. We evaluate the effect of servers number and the dimensions/attributes number on the CPU time. Our experiments demonstrated the performance of the proposed algorithm GES (Global Evidential Skyline).

DEMO 4: In the Figure A.7, we show the skyline size effect on the CPU Time for the top-k algorithm. The larger skyline set, the larger execution time because the more we have a large number of skyline objects, the more the TES component computes the dominance score of the skyline objects.

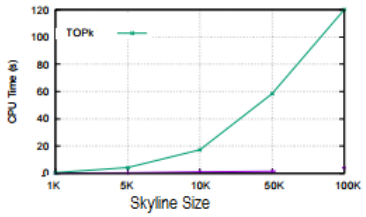


Figure A.7: Top-k Evidential Skyline

A.4 Conclusion

In this annex, we have proposed our evidential skyline system and have shown some demo scenarios. We have proved that our experimental results achieved in the previous chapters are efficient enough to be used in practice.

Appendix A

Uncertainty Models

Data uncertainty is inherent in many real-life applications. Several theories were proposed to handle the data imprecision, uncertainty and imperfection. In this appendix, we present the basic notions about the probability theory as well as the possibility theory, the main uncertainty models.

A.1 Basic Probability Theory

Probability theory began in seventeenth century France when the two great French mathematicians, Blaise Pascal and Pierre de Fermat, corresponded over two problems from games of chance. Problems like those Pascal and Fermat solved continued to influence such early researchers as Huygens, Bernoulli, and De Moivre in establishing a mathematical theory of probability. Today, probability theory is a well established branch of mathematics that finds applications in every area of scholarly activity from music to physics, and in daily experience from weather prediction to predicting the risks of new medical treatments. This appendix is designed for an introductory probability reminder.

The probabilities assigned to events by a distribution function on a sample space ω satisfy the following properties:

1. $P(E) \geq 0$ for every $E \subset \omega$.
2. $P(\omega) = 1$.
3. If $E \subset F \subset \omega$, then $P(E) \leq P(F)$

4. $P(A^{\sim}) = 1 - P(A)$ for every $A \subset \omega$.

5. If A and B are disjoint subsets of ω, then $P(A \cup B) = P(A) + P(B)$.
   Suppose next that A and B are disjoint subsets of ω. Then every element w of $A \cup B$ lies either in A and not in B or in B and not in A. It follows that

$$P(A \cup B) = \sum_{w \in A \cup B} m(w) = \sum_{w \in A} m(w) + \sum_{w \in B} m(w) = P(A) + P(B) \tag{A.1}$$

In summary probability theory

- Dedicated to random phenomena
- Unable to model uncertainty due to lack of knowledge or missing information
- Information demanding (requires to known W or prior probabilities)
- A pure numeric model (very difficult in the case of subjective probabilities)
- Complex computation and reasoning
- Additive: error propagation and amplification
- A single measure to represent uncertainty, i.e., $P(A)$ implies $P(A^c)$, (A^c complementary of A).

A.2 Basic Possibility Theory

The theory of possibility (Zadeh, 1978; Dubois & Prade, 1988) described in this appendix is related to the theory fuzzy sets by defining the concept of a possibility distribution as a fuzzy restriction which acts as an elastic constraint on the values that may be assigned to a variable.

The importance of the theory of possibility stems from the fact that-contrary to what has become a widely accepted assumption much of the information on which human decisions are based is possibilistic rather than probabilistic in nature.

The basic building blocks of possibility theory originate in Zadeh's paper (Zadeh, 1978) and have been more extensively described and investigated in books by Dubois and Prade (Dubois & Prade, 1988). Zadeh starts from the idea of a possibility distribution, to which he associates a possibility measure.

Let U be a set of states of affairs (or descriptions thereof), or states for short. This set can be the domain of an attribute (numerical or categorical), the Cartesian product of attribute domains, the set of interpretation of a propositional language, etc.

A possibility distribution is a mapping π from U to a totally ordered scale $\mathcal{S}$, with top denoted by 1 and bottom by 0. In the finite case $\mathcal{S} = \{1 = \lambda_1 > ...\lambda_n > \lambda_{n+1} = 0\}$. The possibility scale can be the unit interval as suggested by Zadeh, or generally any finite chain, or even the set of non-negative integers. It is assumed that $\mathcal{S}$ is equipped with an order-reversing map denoted by $\lambda \in \mathcal{S} \rightarrow 1 - \lambda$.

The function π represents the state of knowledge of an agent (about the actual state of affairs), also called an epistemic state distinguishing what is plausible from what is less plausible, what is the normal course of things from what is not, what is surprising from what is expected. It represents a flexible restriction on what is the actual state of facts with the following conventions (similar to probability, but opposite to Shackle's potential surprise scale which refers to impossibility):

- $\pi(u) = 0$ means that state u is rejected as impossible.
- $\pi(u) = 1$ means that state u is totally possible (= plausible).

The larger $\pi(u)$, the more possible, i.e., plausible the state u is.

In the Boolean case, π is just the characteristic function of a subset $E \subseteq U$ of mutually exclusive states, ruling out all those states outside E considered as impossible. Possibility theory is thus a (fuzzy) set-based representation of incomplete information.

Given a simple query of the form "does event A occur?" (is the corresponding proposition a true?), where A is a subset of states, the set of models of a, the response to the query can be obtained by computing degrees of possibility (Zadeh, 1978) and necessity (Dubois & Prade, 1988), respectively:

$$\pi(A) = sup_{u \in A} \pi(u); \tag{A.2}$$

$$N(A) = inf_{s \notin A} = 1 - \pi(u). \tag{A.3}$$

$\pi(A)$ evaluates to what extent A is consistent with π, while $N(A)$ evaluates to what extent A is certainly implied by π. The possibility-necessity duality is expressed by $N(A) =$

$1 - \pi(A^c)$, where A^c is the complement of A. Generally, $\pi(U) = N(U) = 1$ and $\pi(\emptyset) = N(\emptyset) = 0$ (since π is normalized to 1). In the Boolean case, the possibility distribution comes down to the disjunctive (epistemic) set $E \subseteq U$, and possibility and necessity are then as follows:

- $\pi(A) = 1$ if $A \cap E = \emptyset$, and 0 otherwise: function π checks whether proposition A is logically consistent with the available information or not.
- $N(A) = 1$ if $E \subseteq A$, and 0 otherwise: function N checks whether proposition A is logically entailed by the available information or not.

Possibility measures satisfy the characteristic "maxitivity" property

$$\pi(A \cup B) = max(\pi(A), \pi(B)) \tag{A.4}$$

Necessity measures satisfy an axiom dual to that of possibility measures, namely

$$N(A \cap B) = min(N(A), N(B)) \tag{A.5}$$

On infinite spaces, these axioms must hold for infinite families of sets. As a consequence, of the normalization of π, $min(N(A), N(A^c)) = 0$ and $max(\pi(A), \pi(A^c)) = 1$, where A^c is the complement of A, or equivalently $\pi(A) = 1$ whenever $N(A) > 0$, which totally fits the intuition behind this formalism, namely that something somewhat certain should be first fully possible, i.e. consistent with the available information. Moreover, one cannot be somewhat certain of both A and A^c, without being inconsistent. Note also that we only have $N(A \cap B) \geq max(N(A), N(B))$: This goes well with the idea that one may be certain about the event $A \cap B$, without being really certain about more specific events such as A and B.

In this appendix, we have presented the basic notions about the probability theory as well as the possibilistic theory. These theories aim at modeling the uncertainty in data.

In summary, the probability theory is dedicated to random phenomena, unable to model uncertainty due to lack of knowledge or missing information. It is really a pure numeric model, thus, very difficult in the case of subjective probabilities. In addition, when talking about the possibilistic setting, it is necessary to note that the knowledge can be then encoded in a pure qualitative way while the probabilistic knowledge must be numeric.

Appendix A

Academic Achievements

1. **Sayda Elmi**, Mohamed Anis Bach Tobji, Allel Hadjali, Boutheina BenYaghlane. Selecting Skyline Stars over Uncertain Databases: Semantics and Refining Methods in the Evidence Theory Setting. In the Journal of Applied Soft Computing. Vol. 57, August 2017, pp. 88-101.

2. **Sayda Elmi**, Allel Hadjali, Mohamed Anis Bach Tobji, Boutheina BenYaghlane. Efficient Distributed Computing and Maintenance of the Skyline over Imperfect Databases. Proc. of The 29th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2017), November 06-08 2017, Boston, MA, USA.

3. **Sayda Elmi**, Allel Hadjali, Mohamed Anis Bach Tobji, Boutheina BenYaghlane. Efficient Distributed Skyline over Imperfect Data Modeled by the Evidence Theory. Proc. of The 28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2016), San Jose, California, USA 2016, pp. 335-342.

4. **Sayda Elmi**, Allel Hadjali, Mohamed Anis Bach Tobji, Boutheina BenYaghlane. Imperfect Top-k Skyline Query with Confidence Level. Proc. of The 13th ACS/IEEE International conference on computer systems and Applications (AICCSA 2016), Agadir, Moroco, November 29th to December 2nd, 2016.

5. **Sayda Elmi**, Mohamed Anis Bach Tobji, Allel Hadjali, Boutheina Ben Yaghlane, Efficient Skyline Maintenance over Frequently Updated Evidential Databases. Proc. of The 15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-2016), Eindhoven, The Netherlands, 2016, pp. 199-210.

6. Fatma Ezzahra Bousnina, **Sayda Elmi**, Mohamed Anis Bach Tobji, Mouna Chebah, Allel Hadjali, Boutheina Ben Yaghlane. Object-relational implementation of evidential databases. Proc. of the International Conference on Digital Economy (ICDEc 2016), Carthage, Tunisia, 2016, pp. 80-87.

7. Fatma Ezzahra Bousnina, **Sayda Elmi**, Mohamed Anis Bach Tobji, Mouna Chebah, Allel Hadjali, Boutheina Ben Yaghlane. Skyline Operator over Combined Reviews of Tripadvisor Travelers under the Belief Functions Theory. Proc. of The 2nd International Conference on Digital Economy (ICDEc 2017), Carthage, Tunisia, 4-6 May 2017.

8. **Sayda Elmi**, Allel Hadjali, Mohamed Anis Bach Tobji, Boutheina Ben Yaghlane, Maintenance du Skyline sur des données évidentielles, 25ème Rencontres Francophones sur la Logique Floue et ses Applications (LFA 2016), Poitiers, France, 3-4 Novembre 2016.

9. **Sayda Elmi**, Karim Benouaret, Allel Hadjali, Mohamed Anis Bach Tobji, Boutheina Ben Yaghlane, Requêtes Skyline en présence des données évidentielles, 15ème Conférence Internationale sur l'Extraction et la Gestion des Connaissances (EGC), Luxembourg, 2015, pp. 215-220.

10. **Sayda Elmi**, Karim Benouaret, Allel Hadjali, Mohamed Anis Bach Tobji, Boutheina Ben Yaghlane, Computing Skyline from Evidential Data, Proc. of the 8th International Conference on Scalable Uncertainty Management (SUM), Oxford, UK, 2014, pp. 148-161.

References

References

Aggarwal, C. C., & Yu, P. S. (2009). A survey of uncertain data algorithms and applications. *Journal of IEEE Transactions on Knowledge and Data Engineering*, *21*(5), 609-623.

Alwan, A. A., Ibrahim, H., Udzir, N. I., & Sidi, F. (2017). Processing skyline queries in incomplete distributed databases. *Journal of Intelligent Information Systems*, *48*(2), 399–420.

Amagata, D., Sasaki, Y., Hara, T., & Nishio, S. (2016). Efficient processing of top-k dominating queries in distributed environments. *Journal of World Wide Web*, *19*(4), 545–577.

Atallah, M. J., & Qi, Y. (2009). Computing all skyline probabilities for uncertain data. In *Proceedings of the 28th acm sigmod-sigact-sigart symposium on principles of database systems* (p. 279-287).

Bach Tobji, M. A., Ben Yaghlane, B., & Mellouli, K. (2008). Frequent itemset mining from databases including one evidential attribute. In *Proceedings of the second international conference on scalable uncertainty management,sum* (p. 1535-1542).

Baichen, C., Weifa, L., & Xu, Y. J. (2009). Progressive skyline query evaluation and maintenance in wireless sensor networks. In *Proceedings of the 18th acm conference on information and knowledge management* (pp. 1445–1448).

Bell, D., Guan, J., & Lee, S. K. (1996). Generalized union and project operations for pooling uncertain and imprecise information. *Journal of Data and Knowledge Engineering*, *18*(2), 89 - 117.

Benouaret, K., Benslimane, D., & Hadjali, A. (2012). Selecting skyline web services from uncertain qos. In *Proceedings of the 9th ieee international conference on services computing scc* (p. 523-530).

Bohm, C., Pryakhin, A., & Schubert, M. (2006). The gauss-tree: Efficient object identification in databases of probabilistic feature vectors. In *Proceedings of the 22nd international conference on data engineering icde* (p. 9-9).

Borzsonyi, S., Kossmann, D., & Stocker, K. (2001). The skyline operator. In *Proceedings of the 17th international conference on data engineering icde* (pp. 421–430).

Bosc, P., Hadjali, A., & Pivert, O. (2011). On possibilistic skyline queries. In *Proceedings of the 9th international conference on flexible query answering systems, fqas* (pp. 412–423).

Bosc, P., & Pivert, O. (2005). About projection-selection-join queries addressed to possibilistic relational databases. *Journal of IEEE T. Fuzzy Systems*, *13*(1), 124-139.

Bosc, P., & Pivert, O. (2010). Modeling and querying uncertain relational databases: a survey of approaches based on the possible worlds semantics. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *18*(05).

Bousnina, F. E., Bach Tobji, M. A., Chebbah, M., Liétard, L., & Ben Yaghlane, B. (2015). A new formalism for evidential databases. In *Proceedings of 22nd international symposium on foundations of intelligent systems, ismis* (pp. 31–40).

Bousnina, F. E., Elmi, S., Bach Tobji, M. A., Chebbah, M., Hadjali, A., & Ben Yaghlane, B. (2016). Object-relational implementation of evidential databases. In *Proceedings of the 1st international conference on digital economy icdec* (p. 80-87).

Chan, C.-Y., Jagadish, H. V., Tan, K.-L., Tung, A. K. H., & Zhang, Z. (2006). Finding k-dominant skylines in high dimensional space. In *Proceedings of the international conference on management of data* (pp. 503–514). ACM.

Chen, L., Özsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the sigmod international conference on management of data* (pp. 491–502). ACM.

Cheng, R., Kalashnikov, D. V., & Prabhakar, S. (2004, September). Querying imprecise data in moving object environments. *Journal of IEEE Transactions on Knowledge and Data Engineering*, *16*(9), 1112–1127.

Chomicki, J. (2007, June). Database querying under changing preferences. *Journal of Annals of Mathematics and Artificial Intelligence*, *50*(1-2), 79–109.

Chomicki, J., Godfrey, P., Gryz, J., & Liang, D. (2003). Skyline with presorting. In *Icde* (p. 717-719).

Dalvi, N., & Suciu, D. (2007, October). Efficient query evaluation on probabilistic databases. *Journal of The Very Large Data Bases (VLDB)*, *16*(4), 523–544.

Dalvi, N. N., & Suciu, D. (2007). Management of probabilistic data: foundations and challenges. In *Proceedings of the 26th acm sigmod-sigact-sigart symposium on principles of database systems* (pp. 1–12).

Dempster, A. P. (1968). A generalization of bayesian inference. *Journal of the Royal Statistical Society*, *30*(B), 205–247.

Deng, K., Zhou, X., & Tao, H. (2007). Multi-source skyline query processing in road networks. In *Ieee 23rd int. conf. on data engineering* (p. 796-805).

Deshpande, A., Guestrin, C., Madden, S. R., Hellerstein, J. M., & Hong, W. (2004). Model-driven data acquisition in sensor networks. In *Proc. int. conf. on very large data bases (vldb)* (pp. 588–599).

Ding, X., & Jin, H. (2010). Efficient and progressive algorithms for distributed skyline queries over uncertain data. In *In the proceedings of distributed computing systems* (p. 149-158).

Dubois, D., & Prade, H. (1988). *Possibility theory: An approach to computerized processing of uncertainty*. Plenum Press.

Ee, P. L., Srivastava, J., & Shekhar, S. (1994). Resolving attribute incompatibility in database integration: An evidential reasoning approach. In *Proceedings of the 10th international conference on data engineering* (p. 154-163).

Ee, P. L., Srivastava, J., & Shekhar, S. (1996). An evidential reasoning approach to attribute value conflict resolution in database integration. *Journal of IEEE Transactions on Knowledge and Data Engineering.*, *8*(5), 707-723.

Elmi, S., Benouaret, K., Hadjali, A., Bach Tobji, M. A., & Ben Yaghlane, B. (2014). Computing skyline from evidential data. In *Scalable uncertainty management: 8th international conference, sum* (pp. 148–161).

Elmi, S., Hadjali, A., Bach Tobji, M. A., & Ben Yaghlane, B. (2016). *Imperfect top-k skyline query with confidence level.* In the Proceedings of the 13th IEEE/ACS International Conference of Computer Systems and Applications, AICCSA, Agadir, Morocco.

Elmi, S., Tobji, M. A. B., Hadjali, A., & Yaghlane, B. B. (2016a). Efficient distributed skyline over imperfect data modeled by the evidence theory. In *28th IEEE international conference on tools with artificial intelligence, ICTAI 2016, san jose, ca, usa, november 6-8, 2016* (pp. 335–342).

Elmi, S., Tobji, M. A. B., Hadjali, A., & Yaghlane, B. B. (2016b). Efficient skyline maintenance over frequently updated evidential databases. In *Proceedings of the 16th international conference on information processing and management of uncertainty in knowledge-based systems, ipmu* (pp. 199–210).

Elmi, S., Tobji, M. A. B., Hadjali, A., & Yaghlane, B. B. (2017). Selecting skyline stars over uncertain databases: Semantics and refining methods in the evidence theory setting. *Applied Soft Computing*, *57*, 88 - 101.

Endres, M., & Kieling, W. (2015, October). Parallel skyline computation exploiting the lattice structure. *Journal of Database Management.*, *26*(4), 18–43.

Fung, G. P., Lu, W., & Du, X. (2009). Dominant and k nearest probabilistic skylines. In *Dasfaa* (pp. 263–277). Berlin, Heidelberg: Springer-Verlag.

Fuxman, A., Fazli, E., & Miller, R. J. (2005). Conquer: Efficient management of inconsis-

tent databases. In *In proceedings of sigmod international conference on management of data* (pp. 155–166). ACM.

Ge, T., Zdonik, S., & Madden, S. (2009). Top-k queries on uncertain data: On score distribution and typical answers. In *Sigmod* (pp. 375–388). ACM.

Groz, B., & Milo, T. (2015). Skyline queries with noisy comparisons. In *Proceedings of the 34th acm sigmod-sigact-sigai symposium on principles of database systems* (pp. 185–198). ACM.

Hadjali, A., Kaci, S., & Prade, H. (2008). Database preferences queries: A possibilistic logic approach with symbolic priorities. In *Proceedings of the 5th international conference on foundations of information and knowledge systems* (pp. 291–310). Berlin, Heidelberg: Springer-Verlag.

Hadjali, A., Souhila, K., & Henri, P. (2011, December). Database preference queries–a possibilistic logic approach with symbolic priorities. *Journal of Annals of Mathematics and Artificial Intelligence*, *63*(3-4), 357–383.

Ha-Duong, M. (2008). Hierarchical fusion of expert opinions in the transferable belief model, application to climate sensitivity. *International Journal of Approximate Reasoning*, *49*(3), 555-574.

Haenni, R., & Lehmann, N. (2003). Implementing belief function computations. *International Journal of Intelligent Systems.*, *18*(1), 31–49.

Hevner, A. R., & Yao, S. B. (1979). Query processing in distributed database system. *Journal of IEEE Transactions on Software Engineering*, *SE-5*(3), 177-187.

Hong, J., He, Z., & Bell, D. A. (2010). An evidential approach to query interface matching on the deep web. *Journal of Information Systems archive*, *35*(2), 140-148.

Huang, Z., Jensen, C. S., Lu, H., & Ooi, B. C. (2006). Skyline queries against mobile lightweight devices in manets. In *Proceedings of the 22nd international conference on data engineering* (pp. 66–74).

Ilaria, B., Paolo, C., & Marco, P. (2014, May). Domination in the probabilistic world: Computing skylines for arbitrary correlations and ranking semantics. *Journal of ACM Transactions Database Systems.*, *39*(2), 14:1–14:45.

Imieliński, T., & Lipski, W., Jr. (1984). Incomplete information in relational databases. *Journal of ACM*, *31*(4), 761–791.

Jiang, B., Pei, J., Lin, X., & Yuan, Y. (2012). Probabilistic skylines on uncertain data: model and bounding-pruning-refining methods. *Journal of Intelligent Information System.*, *38*(1), 1-39.

Kiessling, W. (2002). Foundations of preferences in database systems. In *Proceedings of the 28th international conference on very large data bases* (pp. 311–322). VLDB Endowment.

Lee, S. K. (1992a). An extended relational database model for uncertain and imprecise information. In *Proceedings of the 18th international conference on very large data*

bases vldb (p. 211-220).

Lee, S. K. (1992b). Imprecise and uncertain information in databases: An evidential approach. In *Proceedings of the 8th international conference on data engineering* (pp. 614–621).

Li, F., Yi, K., & Jestes, J. (2009). Ranking distributed probabilistic data. In *Proc. acm sigmod int.ernational conf. on management of data* (pp. 361–374).

Lian, X., & Chen, L. (2008). Monochromatic and bichromatic reverse skyline search over uncertain databases. In *Proceedings of the acm sigmod international conference on management of data* (p. 213-226).

Lian, X., & Chen, L. (2009). Probabilistic inverse ranking queries over uncertain data. In *Proceedings of the 14th international conference database systems for advanced applications, dasfaa* (p. 35-50).

Lian, X., & Chen, L. (2013, March). Probabilistic top-k dominating queries in uncertain databases. *Journal of Information Science.*, *226*, 23–46.

Lin, X., Yuan, Y., Wang, W., & Lu, H. (2005a). Stabbing the sky: Efficient skyline computation over sliding windows. In *Proceedings of the 21st international conference on data engineering* (pp. 502–513). Washington, DC, USA: IEEE Computer Society.

Lin, X., Yuan, Y., Wang, W., & Lu, H. (2005b). Stabbing the sky: efficient skyline computation over sliding windows. In *Proceedings of the 21st international conference on data engineering, icde* (p. 502-513).

Lin, X., Zhang, Y., Zhang, W., & Cheema, M. A. (2011). Stochastic skyline operator. In *Proceedings of the 2011 ieee 27th international conference on data engineering* (pp. 721–732). Washington, DC, USA: IEEE Computer Society.

Liu, J., Zhang, H., Xiong, L., Li, H., & Luo, J. (2015). Finding probabilistic k-skyline sets on uncertain data. In *Proceedings of the 24th acm international on conference on information and knowledge management* (pp. 1511–1520). ACM.

Papadias, D., Tao, Y., Fu, G., & Seeger, B. (2005, March). Progressive skyline computation in database systems. *Journal of ACM Trans. Database System.*, *30*(1), 41–82.

Park, Y., Min, J.-K., & Shim, K. (2013, September). Parallel computation of skyline and reverse skyline queries using mapreduce. *Proc. VLDB Endow.*, *6*(14), 2002–2013.

Pei, J., Jiang, B., Lin, X., & Yuan, Y. (2007). Probabilistic skylines on uncertain data. In *Proceedings of the 33rd international conference on very large data bases vldb* (p. 15-26).

Pivert, O., & Prade, H. (2014). Skyline queries in an uncertain database model based on possibilistic certainty. In *Proceedings of the 8th international conference on scalable uncertainty management* (pp. 280–285). Springer-Verlag New York, Inc.

Samet, A., Gaudin, T., Lu, H., Wadouachi, A., Pourceau, G., Van Hecke, E., ... Dao, T.-T. (2016). Predictive model based on the evidence theory for assessing critical micelle concentration property. In *Proceedings of the 16th international conference on*

information processing and management of uncertainty in knowledge-based systems, ipmu (pp. 510–522).

Samet, A., Lefevre, E., & Yahia, S. B. (2014). Integration of extra-information for belief function theory conflict management problem through generic association rules. *Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *22*(4), 531–552.

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton: Princeton University Press.

Sharif Zadeh, M., & Shahabi, C. (2006). The spatial skyline queries. In *Proceedings of the 32nd international conference on very large data bases* (pp. 751–762). VLDB Endowment.

Smets, P. (1999). Practical uses of belief functions. In *Proceedings of the 15th conference on uncertainty in artificial intelligence* (pp. 612–621).

Smets, P. (2008). Belief functions: The disjunctive rule of combination and the generalized bayesian theorem. In *Classic works of the dempster-shafer theory of belief functions* (Vol. 219, p. 633-664). Springer Berlin Heidelberg.

Wu, P., Agrawal, D., Egecioglu, Ö., & Abbadi, A. E. (2007). Deltasky: Optimal maintenance of skyline deletions without exclusive dominance region generation. In *Proceedings of the 23rd international conference on data engineering, icde* (pp. 486–495).

Xia, T., & Zhang, D. (2006). Refreshing the sky: The compressed skycube with efficient support for frequent updates. In *Sigmod* (pp. 491–502). ACM.

Yi, K., Li, F., Kollios, G., & Srivastava, D. (2008). Efficient processing of top-k queries in uncertain databases. In *Icde* (pp. 1406–1408). Washington, DC, USA: IEEE Computer Society.

Ying, Z., Wenjie, Z., Xuemin, L., Bin, J., & Jian, P. (2011). Ranking uncertain sky: The probabilistic top-k skyline operator. *Journal of Information Systems*, *36*(5), 898–915.

Yiu, M. L., & Mamoulis, N. (2007). Efficient processing of top-k dominating queries on multi-dimensional data. In *Proceedings of the 33rd international conference on very large data bases* (pp. 483–494). VLDB Endowment.

Yong, H., Jin, H. K., & Seung, W. H. (2008). Skyline ranking for uncertain data with maybe confidence. In *Icde* (pp. 572–579).

Yong, H., Lee, J., Kim, J., & won Hwang, S. (2014). Skyline ranking for uncertain databases. *Journal of Information Sciences*, *273*, 247 - 262.

Yu, Q., & Bouguettaya, A. (2010). Computing service skyline from uncertain qows. *Journal of IEEE Transactions on Services Computing*, *3*(1), 16-29.

Yuan, Y., Lin, X., Liu, Q., Wang, W., Yu, J. X., & Zhang, Q. (2005). Efficient computation of the skyline cube. In *Proceedings of the 31st international conference on very large data bases* (pp. 241–252).

Zadeh, L. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*,

100, 9 - 34.

Zhang, W., Lin, X., Zhang, Y., Cheema, M. A., & Zhang, Q. (2012, June). Stochastic skylines. *Journal of ACM Transactions on Database Systems.*, *37*(2), 1–34.

Zhang, W., Lin, X., Zhang, Y., Wang, W., Zhu, G., & Yu, J. X. (2013). Probabilistic skyline operator over sliding windows. *Journal of Information Systems*, *38*(8), 1212 - 1233.

Zhang, Y., Jun, Y., Wei, G., & Wu, L. (2010). Find multi-objective paths in stochastic networks via chaotic immune pso. *Journal of Expert Systems with Applications*, *37*(3), 1911 - 1919.

Zhang, Y., Wu, L., Wei, G., & Wang, S. (2011). A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network. *Journal of Digital Signal Processing*, *21*(4), 517 - 521.

Zhenjie, Z., Reynold, C., Dimitris, P., & Anthony, T. (2009). Minimizing the communication cost for continuous skyline maintenance. In *Proceedings of the acm sigmod international conference on management of data* (pp. 495–508).