A Method for Location Recommendation via Skyline Query Tolerant to Noisy Geo-referenced Data

Welder B. de Oliveira[1], **Helton Saulo**[1],
Sávio S. Teles de Oliveira[2], **Vagner J. Sacramento Rodrigues**[2], **Kleber V. Cardoso**[3]

[1] Instituto de Matemática e Estatística – Universidade Federal de Goiás (UFG)
Caixa Postal 15.064 – 91.501-970 – Goiânia – GO – Brazil

[2]GoGeo
Rua Leopoldo Bulhões, esquina com a Rua 1014
Quadra 31, Lote 07, Sala 9 Setor Pedro Ludovico
CEP 74820-270 – Goiânia – GO – Brazil

[3]Instituto de Informática – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Campus Samambaia
131 - CEP 74001-970 – Goiânia – GO – Brazil

{welder,heltonsaulo}@gmail.com, {savio.teles,vagner}@gogeo.io,
kleber@inf.ufg.br

Abstract. *This work presents a method to perform a location recommendation based on multiple criteria even when there is noise in the coordinates. More specifically, the skyline query is adapted to handle this noise by modeling the errors of geo-referenced points with an appropriate probability distribution and modifying the traditional dominance criterion used by that technique. The method is applied to a scenario in which the coordinates are set by a geocoding process in a sample of schools in a Brazilian city. It enables one to choose the level of confidence in which a point is removed from the skyline solution, i.e., the location recommendation.*

1. Introduction

Database management systems (DBMS) have been increasingly used in recommendation services or applications. Many of these applications are based on multiple, and sometimes conflicting goals, where there may be no single optimal answer. For example, a tourist may be interested in budget hotels with reasonable ratings (e.g., 3-star) that are close to the city. Traditionally, the DBMS supports the recommendation applications by returning all answers that may meet the user's requirement. However, this may not be useful if the user is overloaded by a large amount of information.

Spatial skyline queries [Borzsony et al. 2001] have gained attention due to their efficient solution for this issue. These queries retrieve the desired objects that are no worse than any other object in the database, according to all the criteria under evaluation. In other words, given a set of points, skyline comprises the points that are not dominated by other points. In our example, if there are two hotels, $h1$ and $h2$, with the same rating, such that $h1$ is both cheaper and nearer to the city than $h2$, then $h2$ would not be presented to the user.

The spatial skyline query is directly impacted by the accuracy of the location provided by the database. Data uncertainty inherently exists in a large number of applications [Ding et al. 2014] due to factors such as limitations of the measuring devices (e.g., GPS), and inaccuracy of the geocoding algorithms, when only the street address (e.g., the hotel) is provided. Returning to example, if the hotels $h1$ and $h2$ had been incorrectly located, thus $h2$ could dominate $h1$ due the location error only.

Due to the importance of recommendation applications and the frequent problem of imprecise or noisy data, there is a relevant demand for the creation of automated solutions that are tolerant to the data inaccuracy. How to perform analysis using inaccurate locations, especially the skyline analysis, remains an important and challenging problem. In this paper, we present a novel technique to perform skyline queries over inaccurate locations.

The remainder of this paper is organized as follows. In Section 2 we briefly give an overview of the use of approaches for recommendation services. The problem of skyline queries over inaccurate locations is formally defined in Section 3. Section 4 presents our approach of recommendation service that is tolerant to inaccuracy on the spatial data. Section 5 presents results and discussion of our approach. Finally, we provide some concluding remarks in Section 6.

2. Related work

Since the introduction of the skyline operator [Borzsony et al. 2001], skyline query processing has received considerable attention in multidimensional databases. Several algorithms for skyline computation have been proposed. For example, [Tan et al. 2001] use auxiliary structures on progressive skyline computation, [Kossmann et al. 2002] show a nearest neighbour algorithm for skyline query processing, [Papadias et al. 2003] introduce the branch and bound skyline (BBS) algorithm, [Chomicki et al. 2003] present a sort-filter-skyline (SFS) algorithm leveraging pre-sorting lists, and [Godfrey et al. 2005] propose a linear elimination sort for skyline algorithm with attractive average-case asymptotic complexity. In [Groz and Milo 2015] the true skyline is returned with a high probability with less comparisons required for computing or verifying a candidate skyline.

The concept of spatial skyline query (SSQ) was introduced in [Sharifzadeh and Shahabi 2006], in which given a set of data points P and a set of query points Q, each data point has a number of derived spatial attributes, and each attribute is the point's distance to a query point.

[You et al. 2013] propose the threshold farthest spatial skyline (TFSS) and branch and bound farthest spatial skyline (BBFS) algorithms. The TFSS algorithm uses a standard set of accesses such as sorted access from distributed sources, which uses R-tree for accessing node and retrieves data objects in decreasing order of the attribute value. The BBFS algorithm uses minimum Bounding rectangle (MBR) of an R-tree for batch pruning. Full space skyline can be supported incrementally by using naïve on-line maintenance module (NMA), as described in [Huang et al. 2010].

For a spatial skyline query using Euclidean distance, efficient algorithms have been proposed in [Son et al. 2009, Lee et al. 2011]. [Son et al. 2014] develope an algorithm using the Manhattan distance, which closely reflects road network distance for metro areas with well-connected road networks.

Some studies have also focused on the skyline query processing with moving object. In this sense, a novel probabilistic skyline model is proposed in [Ding et al. 2014] where an uncertain object may take a probability to be in the skyline at a certain point in time. [Huang et al. 2006] had introduced the continuous skyline over precise moving data. [Zhang et al. 2009] present techniques that enable inference of the current and future uncertain locations efficiently.

The present paper brings something new to the area, namely the possibility to perform skyline query even when the data is not precise. For precision we mean that the values provided by the data are exactly what can be found in reality, i.e., the data describe the reality with fidelity (with no errors from measurements, estimation or other sources). The hypothesis of precision is assumed by all current skyline procedures present in literature. We intend to change this scenario providing a more elastic approach to this kind of query, especially concerning spatial attributes.

The two closest related works in literature were made by [Pei et al. 2007] and [Lofi et al. 2013]. The first deal with what they call “uncertain” data, which have different meaning of the “imprecision” data used our work. By uncertain the authors mean that more than one record is available for each attribute for the several objects under evaluation. They provided an example with NBA players data. To each player is collected statistics like number of assists and the number of rebounds, both the larger the better. As players have different performances in different games, the values for the attributes for each player are said to be “uncertain”. [Pei et al. 2007] provide two algorithms to approach the problem. Moreover, the concept of dominance probability is introduced in this paper. On the other hand, [Lofi et al. 2013] use a method based on crowd based data. This way they can perform skyline query for incomplete data. None of them, however, deal with “imprecise” data, i.e., data which contain values with some error.

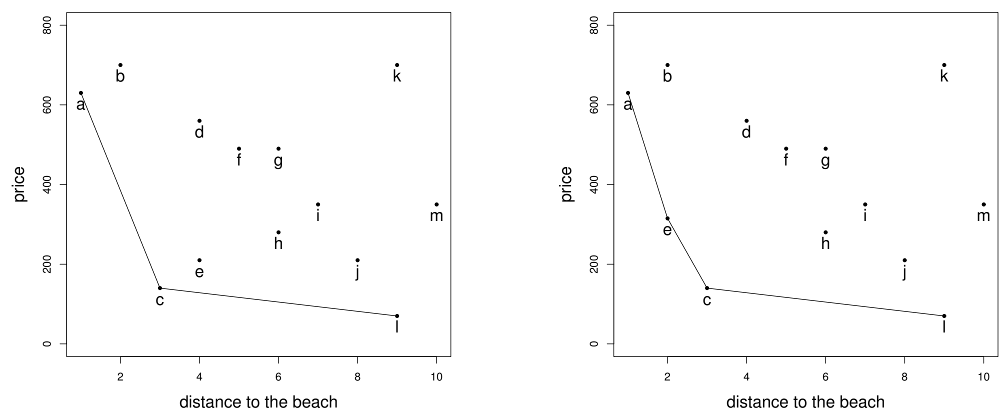
In our paper, the values of spatial attributes are considered random variables and are modelled with a probabilility density function. This enables to compute dominance probabililities even for imprecise data and then provide a more noisy tolerant technique.

3. Formal problem definition

In order to provide a location recommendation tool able to deal with multi-criteria decision analysis, one may face the problem of noisy geo-referenced data. Furthermore, one important step that such a tool should have is discarding points that are not Pareto efficient. Pareto efficiency is an equivalent expression to skyline query and will be explained later. One problem arises in this context: “how to build a skyline query that minimizes the bad effects caused by imprecise geo-referenced data?”

Skyline query is a multi-criteria decision making technique. It aims to find a set S of all points that are not dominated by any other in the database under consideration. According to [Papadias et al. 2003], under the min condition, a point p_i dominates another point p_j if and only if the coordinate of p_i on any axis is not larger than the corresponding coordinate of p_j. In this case, the desirable points are those with the smaller values. In this paper, only the min condition is being considered when the term dominance is mentioned. Naturally, the results are analogous for the max condition.

The Figure 1(a) shows the data for a classical example: "choose a hotel both cheap and near the beach". These are the typical conflicting criteria faced at making a decision concerning multiple goals. The nearest hotel may not be the cheapest. Note that the points a, c and l are not dominated by any other, i.e., no other is better in both dimensions then those. Therefore, $S = \{a, c, l\}$ is the skyline query for these points.



(a) Hotels near the beach: a classical skyline query example.

(b) Hotels near the beach: with a translation in point e.

Figure 1. Skyline query example

However, due to coordinates imprecision, one may not guarantee that point e is in fact dominated by the points in S. The Figure 1(b) shows how S is changed by a relatively small translation in the point e. A translation of this size in e may reflect just the imprecision in its geo-referenced position. The imprecision is measured by the error given by $error = |real - database|$, in which $real$ is the real position and $database$ is the database position. Therefore, a specific point is said to be imprecise if $error > 0$.

Two distinct types of error can occur: I) exclude a point from S when it should be in S; II) include in S a point that should not be there. In this work, we address the type I error by presenting a method to provide a skyline query solution for noisy geo-referenced data.

4. The method

In order to solve the problem proposed by this paper, three steps are required:

1. Model the error with a probability density function (PDF).
2. Generate a table of dominance probabilities via Monte Carlo method.
3. Rewrite skyline query by modifying the dominance criterion in order to take a controlled risk of a type I error.

In the following subsections, these steps will be explained in detail.

4.1. Error modeling

The error must be mathematically modeled, i.e., it is necessary to find a mathematical function that enables one to accurately predicts the behaviour of the error. For instance the probabilility of the error lies in the interval [200, 300] meters. To find this model, one must search for a PDF. A PDF is a function f that associates a value in the interval $[0, 1]$ to each set A of possible values of a random variable X.

With a PDF one can calculate the probability of a value of some random variable lies in a specific range, for example, $error$ be more than 200 and less then 300 meters. In this first step of the method, it is necessary to find a PDF that fits well the curve of the errors. A proper approach to make a guess of such a function is by the histogram or even the kernel estimation of the curve.

Since one suspects that a particular PDF fits well the curve of the errors, a formal hypothesis test may be applied in order to confirm (or not) the guess. A very well recognized hypothesis test in this sense is the Kolmogorov-Smirnov test. As states [Massey Jr 1951],

> If $F_0(x)$ is the population culmulative distribution, and $S_N(x)$ the observed cumulative step-function of a sample (i.e., $S_N(x) = k/N$, where k is the number of observations less than or equal to x), then the sampling distribution of $d = maximum|F_0(x) - S_N(x)|$ is known, and is independent of $F_0(x)$ if $F_0(x)$ is continuous.

Therefore, the distribution of *d* can be used to perform inference related to the hypothesis that $F_0(x)$ is the true populational distribution of the errors.

In our work, we employed the software `R` to perform the Kolmogorov-Smirnov test. The hypothesis may be considered **false** with at least 95% level of confidence if the computed *p-value* is at most 5%. Usually, a 95% level of confidence is considered good enough in order to discard or confirm the use of a specific probability distribution for most applications.

4.2. Table of dominance probabilities via Monte Carlo method

In this subsection, it will be evaluated the probability of a database point P dominates another, say Q in order to construct a table of dominance probabilities. This will be done by the Monte Carlo method. These two points possess the coordinates G and H in the dataset, respectively, which may be imprecise ($error > 0$). Therefore, if one computes the distances between each of those points to a third, say L, the one with the minimum distance to L in the dataset may be not the one with minimum distance in reality. Let p be the probability of P be closer than Q. In this subsection an estimative $\hat{p}$ of p is provided.

The Monte Carlo method aims to estimate a mean $M = E(X)$ by simulations with random numbers, where X is a random variable. About the history and applications of the Monte Carlo method, one can see [Metropolis 1987]. In order to estimate M, n simulated values of X should be performed, say $x_1, x_2, ..., x_n$. After generating the n values for X, its average $m = \sum_1^n x_i/n$ is computed. As n increases, the central limit theorem guarantees that m becomes arbitrarily closer to M, with the difference going to zero as n tends to infinity.

For example, to estimate the probability of getting **head** in a given coin, one can follow the mentioned steps for the variable X defined assuming the valures 0 and 1 with equal probabilility (for example 1 meaning **head**).

1. Let X be 1 if after flipping the coin the result is **head** and zero otherwise;
2. Repeat (1) n times and compute the values of X;
3. Compute the estimate $m = \sum_1^n x_i/n$;
4. Take m as an estimate for M, where M is the true probability of getting **head** by flipping the coin.

In this paper we define a random variable Y and set 1 to it if P is closer than Q in relation to L. Otherwise, the value Y is set to 0. Thus, the mean M of the Y coincides with the probability that P is the closest. As the goal is to estimate this probability, the steps (2), (3) and (4) shown above can be performed. However, it is necessary to define how the simulations will be performed.

In the case of estimating the **head** probability for a given coin, the simulation is simple: just flip the coin and compute the values. To simulate a value for Y, one can follow the paths:

1. generate a value $error_1$ and a value $error_2$ both from the chosen PDF;
2. generate an angle θ_1 and θ_2 both from an uniform distribution with parameters $(0, 2\pi)$, since it is assumed that the error is equally probable in any direction;
3. add to G a vector of length $error_1$ in the direction θ_1, resulting in the point G'. Similarly, add to H a vector of length $error_2$ in the direction of θ_2, resulting in the point H';
4. compute the distances $d_1 = |G' - L|$ and $d_2 = |H' - L|$;
5. if $d_1 < d_2$ then set 1 to Y, otherwise set 0;
6. repeat steps 1 to 5 n times with a large value of n (for instance, $n = 10,000$);
7. calculate $m = \sum_1^n y_i/n$. The value of m is a estimate for the probability that P dominates Q.

In order to provide a table of dominance probabilities as shown in Table 1, the algorithm showed above must be applied for several pairs of points P and Q placed at different distances from a reference place L. In this table, p_{ij} means the probability of P be closer to L than Q, such that in the database the distance of P to L is 100i and from Q to L is 100j. One may construct a more complete table by considering multiples of 1 meter instead of multiples of 100 meters like in this table example. Nevertheless, probabilities for intermediate or even fractional values may be estimated by interpolation.

Table 1. Structure of a dominance table

near / far	200	300	400	500
100	p_{12}	p_{13}	p_{14}	p_{15}
200	p_{22}	p_{23}	p_{24}	p_{25}
300	p_{32}	p_{33}	p_{34}	p_{35}
400	p_{42}	p_{43}	p_{44}	p_{45}
500	p_{52}	p_{52}	p_{54}	p_{55}

Based on the Table 1, the dominance criterion of skyline query can be modified. Now, it is possible to talk about probability of dominance. Instead consider the set S of skyline query points, it is possible to construct the set W_p of points which are not dominated with level of confidence p by any other point. The new criterion is the following: "a point x dominates a point y concerning a dimension i with level of confidence p if $Prob[x_i < y_i] > p$". Therefore, there is a chance of an type I error of $(1-p)$.

For several criteria, say d, and assuming independence between the errors of each of these dimensions, the new dominance criterion may be rewritten like: "a point x dominates y with level of confidence p if the product of the probabilities $Prob[x_i < y_i]$ is greater than p", i.e.,

$$Prob[x_1 < y_1]...Prob[x_d < y_d] > p$$

Thus, the type I error has been controlled as it was the goal of this paper. This new version of skyline query is designed to be more tolerant to geo-referenced imprecise data.

4.3. Birnbaum-Saunders distribution

In this subsection, we discuss the PDF that is used to model the geo-referenced error in the example presented in section 5, more specifically the data displayed by the Figure 2. [Birnbaum and Saunders 1969] introduced a family of Birnbaum-Saunders (BS) distributions motivated by problems of vibration in commercial aircraft that caused fatigue in materials. Although, in principle, its origin is for modeling equipment lifetimes subjected to dynamic loads, the BS distribution has been used for various other purposes, such as finance, quality control, medicine, and atmospheric contaminants. This distribution has two parameters, one of shape a and another of scale b, with b being also the median of the distribution. In addition, the BS distribution is asymmetric with positive skewness and unimodality. If a random variable T follows a BS distribution, denoted by $T \sim \mathrm{BS}(a; b)$, then is cumulative distribution function is given by

$$F_T(t) = P(T \leq t) = \Phi\left(\frac{1}{a}\left[\left(\frac{t}{b}\right)^{1/2} - \left(\frac{b}{t}\right)^{1/2}\right]\right), \quad t > 0, \tag{1}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. The BS model holds proportionality and reciprocal properties given by $k\,T \sim \mathrm{BS}(a, k\,b)$, with $k > 0$, and $1/T \sim \mathrm{BS}(a, 1/b)$, respectively. Also, when a tends to zero, the BS distribution tends to a normal model with mean b and variance τ, where $\tau \to 0$ when $a \to 0$.

5. Results and discussion

The method presented in this paper is exemplified with a data collected from a database of geocoded addresses in Goiânia-GO, Brazil. More specifically, it is a sample of 32 schools in that city. As it is common in geocoding process, the coordinates presents a significant error. To compute this error, it is necessary to have the real position of each point - the schools in this case. For these 32 schools, the right location has been collected by taking its coordinates from Google Maps application. The errors $error_i$ were calculated using the expression $error_i = |real_i - database_i|$, for $i = 1, ..., 32$. Those errors in meters are shown in y-axis of Figure 2. The cut line represents the value 350 of y-axis. Thus, one can see that most of the errors is smaller than 350 meters.

The first step is to find a PDF to model the errors. The Figure 3 shows the histogram in Figure 3(a) and a kernel density estimation with bandwith equals to 91.51 (Figure 3(b)) for the errors verified in school data. The curve suggest a highly heavy tail

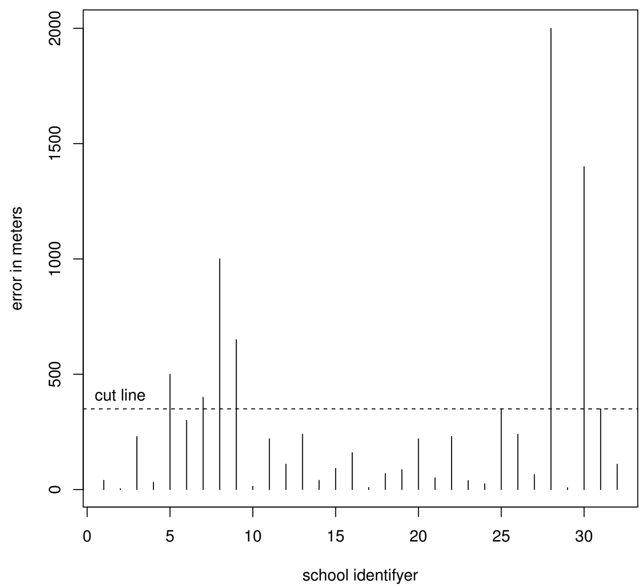


Figure 2. Location error for the 32 schools.

distribution for modeling the errors. Dispite most of the errors are relatively near to zero, there are a reasonable probabilility for some stay far beyond the mean or median of the data. Therefore, simetric options like Normal distribution are not suited for the required model. Thus, the search must rely on assymetric heavy tail PDFs. Bellow some statistics about the errors are presented.

min = 4.0

first quartile = 40

mean = 290.1

median = 135.0

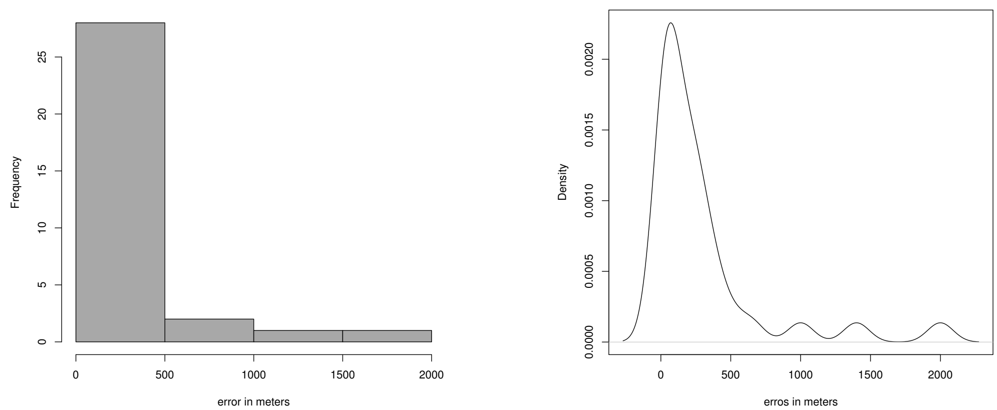
third quartile = 312.5

max = 2000.0

standard deviation = 433.9

pearson's second skewness coefficient = 1.07

As can been seen, the skewness is greater than 0, indicating that most values of the distribution is concentrated left to the mean and that there is a heavy tail to the right. In this context, several heavy tail and assymetric PDFs have been evaluted for modeling the error, until one in particular have been proved to achieve this purpose - the Birnbaum-Saunders (BS) distribution. BS possess the desirable requirements exposed before. In order to confirm (or not) the suspect that the error may be "well" modeled by a BS, the Kolmogorov-Smirnov test was performed. The subsection 4.3 provides more information about this PDF.



(a) Histogram

(b) Density kernel estimation for the 32 schools

Figure 3. Histogram and Kernel density estimation

The parameters a and b of the BS distribution was estimated by the Maximum Likelihood Estimation method (MLE). The parameters estimation in this model is discussed, among others, by [Lemonte et al. 2007]. Here we use a `R` implementation of MLE for this probabilility distribution. The estimated values were 1.88 for a and 100.2 for b.

To decide whether the BS(1.88, 100.2) fits well the data, an `R` implementation of the Kolmogorov-Smirnov test was used. The p-value returned by the test was 0.8745, which it is too high for refuting the hypothesis that the error does not follows a BS distribution. Therefore, the BS(1.88, 100.2) is considered a satisfactory model for the location *error* of the schools. Following the steps reported in subsection 4.2, Table 2 was constructed.

Table 2. A sample of the Dominance Table generated from the school data

near / far	200	300	400	500	600	700
100	0.62	0.71	0.76	0.81	0.84	0.87
200		0.68	0.75	0.80	0.83	0.87
300			0.69	0.77	0.82	0.86
400				0.70	0.77	0.83
500					0.70	0.78
600						0.71

Using the table, the dominance criterion was changed. For example, if P is 200 meters from a reference place L and Q is 500 meters from L, then there is a probability of 80% that P is in fact closer to L than Q. Thus, there is 1 chance in 5 to get a type I error if one considers the point P closer to L than point Q. Therefore, for a cut probability $p = 0.8$, P dominates Q. If more than one dimension was considered, then the product of the probabilities would be used like explained in subsection 4.2.

The example exposed in this section shows how the method can be used. Particularly, the first step presents some issues to be implemented efficiently, since finding a PDF for a random variable (corresponding to the error in our case) is not a trivial task. However, an alternative procedure could be employed in a future work: estimate the empirical distribution function. This approach would enable the method to be applied without the need to look for an ideal and known PDF. Nevertheless, one drawback in doing so is the lack of power that this kind of fitted has relating the a parametric approach like the ones get by finding the PDF. Also, in a future work, it is possible to implement in a programming language, a version of skyline query with the tolerance of errors proposed in this paper.

6. Conclusion

This paper presented a contribution to the area of multi-criteria decision making providing a method to perform skyline queries in the presence of noisy geocorreferenced data. Following the proposed method, it is possible to change the dominance criterion in skyline query, turning this technique more tolerant to location error. Imprecision of this type may be a common feature, for instance with coordinates obtained by a geocoding process. Therefore, the skyline query is generalized in this work from a deterministic to a probabilistic approach.

Despite the cited contribution, the proposed method only can be implemented in cases where the errors can be modeled by some known probabilility distribution. In general, this step may be hard to achieve. However, in a future work one can use the empirical distribution in order to automate this step. An advantage of this last approach would be simplicity and the declared automation. A drawback is the impossibility of choosing a parametric function like those provided by the known PDFs, which are able to provide more power in avoiding type I errors in the statistical hypothesis test (the new criterion created for dominance).

Another suggestion for a future work is to implement an end-to-end location recommendation technique, combining a first step with the noisy tolerant version of skyline query exposed in this paper with a second step related to rank the points with some optimization procedure which also handle noisy data, like, for example, that proposed by [Qin 2013]. The goal of our work has been achieved with the exposure of the method and also with its validation for a real example.

References

Birnbaum, Z. and Saunders, S. C. (1969). A new family of life distributions. *Journal of applied probability*, pages 319–327.

Borzsony, S., Kossmann, D., and Stocker, K. (2001). The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 421–430. IEEE.

Chomicki, J., Godfrey, P., Gryz, J., and Liang, D. (2003). Skyline with presorting. In *ICDE*, volume 3, pages 717–719.

Ding, X., Jin, H., Xu, H., and Song, W. (2014). Probabilistic skyline queries over uncertain moving objects. *Computing and Informatics*, 32(5):987–1012.

Godfrey, P., Shipley, R., and Gryz, J. (2005). Maximal vector computation in large data sets. In *Proceedings of the 31st international conference on Very large data bases*, pages 229–240. VLDB Endowment.

Groz, B. and Milo, T. (2015). Skyline queries with noisy comparisons. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems*, pages 185–198. ACM.

Huang, Z., Lu, H., Ooi, B. C., and Tung, A. K. (2006). Continuous skyline queries for moving objects. *Knowledge and Data Engineering, IEEE Transactions on*, 18(12):1645–1658.

Huang, Z., Sun, S., and Wang, W. (2010). Efficient mining of skyline objects in subspaces over data streams. *Knowledge and information systems*, 22(2):159–183.

Kossmann, D., Ramsak, F., and Rost, S. (2002). Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 275–286. VLDB Endowment.

Lee, M.-W., Son, W., Ahn, H.-K., and Hwang, S.-w. (2011). Spatial skyline queries: exact and approximation algorithms. *GeoInformatica*, 15(4):665–697.

Lemonte, A., Cribari-Neto, F., and Vasconcellos, K. (2007). Improved statistical inference for the two-parameter Birnbaum-Saunders distribution. *Computational Statistics and Data Analysis*, 51:4656–4681.

Lofi, C., El Maarry, K., and Balke, W.-T. (2013). Skyline queries in crowd-enabled databases. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 465–476. ACM.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Metropolis, N. (1987). The beginning of the monte carlo method. *Los Alamos Science*, 15(584):125–130.

Papadias, D., Tao, Y., Fu, G., and Seeger, B. (2003). An optimal and progressive algorithm for skyline queries. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 467–478. ACM.

Pei, J., Jiang, B., Lin, X., and Yuan, Y. (2007). Probabilistic skylines on uncertain data. In *Proceedings of the 33rd international conference on Very large data bases*, pages 15–26. VLDB Endowment.

Qin, Z. (2013). Uncertain random goal programming.

Sharifzadeh, M. and Shahabi, C. (2006). The spatial skyline queries. In *Proceedings of the 32nd international conference on Very large data bases*, pages 751–762. VLDB Endowment.

Son, W., Hwang, S.-w., and Ahn, H.-K. (2014). Mssq: Manhattan spatial skyline queries. *Information Systems*, 40:67–83.

Son, W., Lee, M.-W., Ahn, H.-K., and Hwang, S.-W. (2009). Spatial skyline queries: an efficient geometric algorithm. In *Advances in Spatial and Temporal Databases*, pages 247–264. Springer.

Tan, K.-L., Eng, P.-K., Ooi, B. C., et al. (2001). Efficient progressive skyline computation. In *VLDB*, volume 1, pages 301–310.

You, G.-w., Lee, M.-W., Im, H., and Hwang, S.-w. (2013). The farthest spatial skyline queries. *Information Systems*, 38(3):286–301.

Zhang, M., Chen, S., Jensen, C. S., Ooi, B. C., and Zhang, Z. (2009). Effectively indexing uncertain moving objects for predictive queries. *Proceedings of the VLDB Endowment*, 2(1):1198–1209.