Finding Potential Objects in Uncertain Dataset by Using Competition Power

Sheng-Fu Yang, Guanling Lee, Shou-Chih Lo

Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan
112123@gmail.com, guanling@gms.ndhu.edu.tw, sclo@gms.ndhu.edu.tw

Abstract

In the past studies, it has been proven that skyline queries and dominating queries are very useful in applications such as multi-preference analysis and multi-criteria decision making. In real applications, such as environmental monitoring and market analysis, the data often have uncertain characteristics, and the uncertainty of the data mainly comes from the data randomness, the limitation of the measuring instrument or the delay of updating the data. Therefore, in this paper, by using the dominance concept, we propose an efficient method to help users screen out better data objects in multi-dimensional uncertain dataset. Furthermore, an appropriate probability model is also proposed to objectively calculate the scores of uncertain data. To show the benefit of the approach, a set of experiment is performed on both synthetic and real datasets. According to the experimental results on real dataset, the proposed method can find the potential data objects efficiently.

Keywords: Skyline query, Uncertain data, Competition power, Dominance

1 Introduction

Skyline query and dominance query [1-2] have recently become important in many applications, such as multi-criteria decision making, data mining, and multi-preference analysis. Generally speaking, an object *A* is said to dominate another object *B* if *A* is not worse than *B* on every dimension and *A* is strictly better than *B* on at least one dimension. In skyline query, it returns the points that are not dominated by any other points in the dataset. For example, Table 1 shows the performance of six players on the scoring and rebounding. In Table 1, *A* is a better player than *C*, because *A* and *C* are equally good at scoring, but the number of rebounds is larger than *C*. In this case, we can say that *A* dominates *C*, or *C* is dominated by *A*. Furthermore, *A* and *B* are data points that are incomparable, because *A* is better than *B* in scoring, but *B* is better than *A* in rebounding. In the example, since the two points *B* and *E* are not dominated by other points, {B, E} is the skyline point in this database. By using the skyline concept, we can retrieve excellent data points in the dataset.

Table 1. Player information

Player	Scoring	Rebounding
A	20	6
B	18	18
C	20	3
D	30	6
E	35	6
F	32	4

The concept of the skyline operator was first introduced in [3], which explores three algorithms: the block nested loops (BNL), divide and conquer, and B-tree-based schemes. In [4], by presorting the dataset according to some monotonic scoring function, a variant of BNL named the sort-filter skyline algorithm was proposed. As mentioned in [4], with the resulting order of points, the data object is impossible to dominate the data object listed before it, which simplifies the comparisons. Furthermore, based on the object-based space partitioning concept, an efficient skyline computation method was proposed [5]. Moreover, the problem of how to process skyline query in a parallel way was discussed in [6-9] and several efficient parallel skyline processing algorithms were presented.

A traditional skyline query can only extract data points which are not dominated by other data points in the full space. Therefore, by extending the problem of full-space skyline computation to subspace skyline computation, the concept of subspace skyline was introduced in [10]. Moreover, an efficient subspace approach for calculating the skyline that exploits the seed skyline group lattice formed by the full-space skyline points was proposed in [11]. In [12-13], the interesting problem of computing the skyline according to a user's preference was proposed. Furthermore, because the probability that a data object dominates the other data object decreases as the number of dimension increases. There will be a large amount of skyline points in a high dimensional dataset. To respond this

*Corresponding Author: Guanling Lee; E-mail: guanling@gms.ndhu.edu.tw
DOI: 10.3966/160792642019072004029

problem, in [14-16], the concept of k-dominant skyline was discussed and several efficient algorithm were proposed to solve the problem.

The shortcoming of skyline query is that the number of returned data points cannot be determined. In the above example, if we need to find three outstanding players, the skyline operator will only return two data points. To solve this problem, the concept of dominance query is introduced. In dominance query, the goodness of an object *A* can be naturally measured by the number of other objects dominated by *A* in a set of objects. Refer to Table 2, There is one player ({C}) dominated by *A*, and are two players ({*A*, *C*}) dominated by *D*, the dominant score of *A* and *D* are 1 and 2, respectively. To continue with the above example by using the dominant score as the goodness measure, {*E*, *D*, *F*} can be retrieved as the three outstanding players.

Table 2. The dominant score of players

Player	Dominate Players	Dominant Score
A	C	1
B	-	0
C	-	0
D	A, C	2
E	A, C, D, F	4
F	A, C	2

As mentioned above, in skyline query, it returns the points that are not dominated by any other points in the dataset. In real applications, such as environmental monitoring and market analysis, the data often have uncertain characteristics. The uncertainty of the data mainly comes from the data randomness, the limitation of the measuring instrument or the delay of updating the data. For example, the NBA players may have different performances in different games. If the game-by-game performance data are considered, the player's performance becomes uncertain data.

In the early studies, the average of the instances of each data object was used to represent the uncertain data itself. However, according to the experimental result performed on the database of NBA player performance (scores, assists, rebounds) from 1991 to 2005 [17], it pointed out the inadequacies of using average to represent the uncertain data. For example, a player like Hakeem Olajuwon, because his average performance will be dominated by Shaquille O'Neal, Charles Barkley, Tim Duncan, and Chris Webber, he will not be retrieved in skyline queries if we ignore the uncertanty and use the mean value to represent the performance. However, taking into account the performance of different games, Hakeem Olajuwon does not perform worse than these players in most of games. Just because of the variance of his performance is greater than that of these players, he was not presented in the skyline query. In addition, stars like Kobe Bryant will be excluded from the skyline query if we use average value to represent the performance. Therefore, in [17], the concept of using the probability model to compute the probability which an uncertain object belongs to a skyline point was introduced. And it retrieves the data points (called *p*-skyline) whose probability of becomming a skyline is higher than a predefined threshold (*p*). Furthermore, based on the concept of *p*-skyline, [18-26] continue to propose more efficient algorithms or explore the issues related to the probabilistic skyline, such as subspace skyline and uncertainty preferences.

However, these models have some shortcomings. Refer to Table 1 and Table 2, althought player *B* (special functional player) is in skyline, in dominating query, player B will be excluded from the result (dominant score of *B* is 0). And the same problem will occur for uncertain data. Therefore, in this paper, we discuss the problem of how to efficiently and effectively help users to select better objects in multi-dimensional uncertain data. In this paper, we propose an appropriate probability model to objectively calculate the scores of uncertain data and find objects with considerable potential. Furthermore, by adapting the concept of Minimal Bounding Box and Z-order, an efficient algorithm for retrieving potential objects in a large set of uncertain data is proposed. A set of experiment is performed on both real and syntatic datasets to show the advantage of our approach.

The remainder of this paper is organized as follows. Section 2 provides the main idea and the problem definition of the work. In Section 3, the proposed algorithm is discussed and presented. Section 4 discusses the experimental results and analysis. Conclusions are finally drawn in Section 5.

2 Problem Formulation

Generally speaking, an object *p* is said to dominate another object *q* if *p* is not worse than *q* on every dimension and *p* is strictly better than *q* on at least one dimension. And we use $p \prec q$ to denote *p* dominate *q*. In skyline query, it returns the points that are not dominated by any other points in the dataset. In real applications, such as environmental monitoring and market analysis, the data often have uncertain characteristics, and the uncertainty of the data mainly comes from the data randomness, the limitation of the measuring instrument or the delay of updating the data. For example, Table 3 shows the basketball players performances (score and rebound) in different games. Because each player has different performances in different games, it forms an uncertain dataset. In the following discussion, we use *object* to denote the data we concerned. In the example, we have 6 objects (players *A~F*). Furthermore, (10, 6) is called an instance of *A*, and objects *A* and *E* contain 2 and 3 *instances*, respectively.

Table 3. The performance of 6 players in different games

Player	Score	Rebound
A	10 16	6 7
B	15 15	18 16
C	20 5	3 8
D	30 14	6 16
E	35 32 30	6 6 6
F	32 22	4 3

The concept of probabilistic skylines was first discussed in [13]. It proposed the following equation to calculate the probability of an object U becoming a skyline point. In the equation, U and V are uncertain objects and U contains l instances.

$$P_r(U) = \frac{1}{l}\sum_{i=1}^{l}\prod_{V \neq U}\left(1 - \frac{|\{v \in V \mid v \prec u_i\}|}{|V|}\right) \qquad \textbf{(1)}$$

Although using equation (1) can calculate the probability that an object becomes a skyline, we find in some cases, we will miss some potential objects by using the probability directly. For example, if we have 1,000 basketball players and build information for each basketball player for nearly 3 games. Under such a model, if a basketball player A has 3 games of data that are completely dominated by 3 other players, $P_r(A)$ will be zero. Then even if he is better than the rest of the players, player A will never be included in the result. Furthermore, when there is a super object in the database, then only the super object will be returned as the result. Refer to Table 3, player D's first and second performances are dominated by players E and B, respectively. According to Formula 1, the probability that player D becomes a skyline point will be 0 ($P_r(D)$=0). By comparing the value of player A, ($P_r(A)$=0.5)), the evaluation of play D is worse than Player A. However, player D is a relatively good player in the database, but with such a query, the user will never find player D. That is, the probability of becoming a skyline is not enough to represent the goodness score in the database. To tackle with this problem, we propose the following formula to compute the goodness of the objects.

$$CP(U) = \sum_{U,\ V \in D \wedge U \neq V} CP_{UV}$$

$$CP_{UV} = 1 - \frac{|\{u \in U \text{ and } v \in V \mid v \prec u\}|}{|U| \times |V|} \qquad (2)$$

In the equation, $|U|$ indicates the number of instances of object U, and u represents an instance of object U. Then CP_{uv} is the probability that object U is not dominated by object V, and $CP(U)$ (*Competition Power* of object U) is the sum of CP_{UV} for all objects V in the database. In other words, $CP(U)$ is the expected number of data objects which cannot dominate U. For example, refer to Table 3, the probability which object A is not dominated by object B (CP_{AB}) is

$$CP_{AB} = 1 - \frac{|\{(a_1,b_1),\ (a_1,b_2)\}|}{2 \times 2} = 0.5$$

Similarly,

$$CP_{AC} = 1 - \frac{|\{\}|}{2 \times 2} = 1$$

$$CP_{AD} = 1 - \frac{|\{(a_1,d_1),\ (a_1,d_2)\}|}{2 \times 2} = 0.5$$

$$CP_{AE} = 1 - \frac{|\{(a_1,e_1),\ (a_1,e_2),\ \{a_1,e_3\}|}{2 \times 3} = 0.5$$

$$CP_{AF} = 1 - \frac{|\{\}|}{2 \times 2} = 1$$

Table 4 summarizes the values of CP_{AB} ~ CP_{AF}. With this information, $CP(A)$, the expected value of the number of objects which cannot dominate A, can be calculated by 0.5+1+0.5+0.5+1=3.5.

Table 4. The values of CP_{AB} ~ CP_{AF}

CP_{AB}	CP_{AC}	CP_{AD}	CP_{AE}	CP_{AF}
0.5	1	0.5	0.5	1

In this paper, we use $CP(U)$ as the scoring function to measure the goodness of an uncertain object. The larger the value is, the better the object is. In the experiment result, we will use a real dataset to show the effectiveness of the scoring function.

3 Proposed Method

In this section, we will present the method of finding the object whose competition power is no smaller than a predefined threshold. To calculate the CP value of U, we need to determine the domination relationship between U and the remaining objects V in the dataset. To clarify the discussion, we use *main-object* and *sub-object* to denote objects U and V, respectively.

In uncertain dataset, an object is a collection of instances. To compute the probability of object U dominating object V, we need to verify the domination relationship among the instances in U and the instances in V. To avoid the comparison between every pair of instances in U and V, we adopt the concept of Minimal Bounding Box (MBB). A MBB of an object U is the minimal box which can include every instances of U. For example, refer to Table 3, the lower left corner and upper right corner of MBB of object D is (14, 6) and (30, 16), receptively. As shown in Figure 1, the relationship between two MBBs is either one of the three cases,

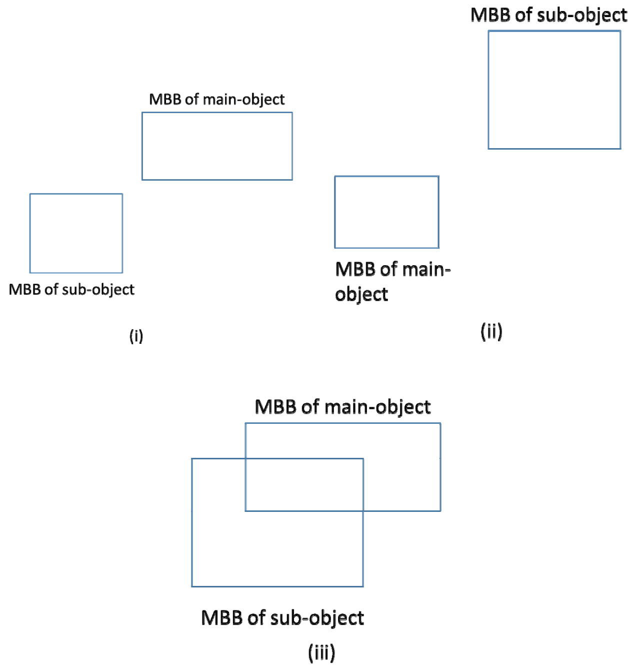

Figure 1. The relationship between two MBBs

(i) main-object fully dominates sub-object
(ii) main-object is fully dominated by sub-object
(iii) main-object and sub-object are incomparable

In cases (i) and (ii), the domination testing is not necessary. Because CP_{UV} (U denotes main-object and V denotes the sub-object) is 1 and 0 in cases (i) and (ii), respectively. On the contrary, CP_{VU} is 0 and 1 in cases (i) and (ii), respectively. For case (iii), we can further partition the MBB into several areas, and avoid the domination relationship checking for the instances falling in the areas which belong to cases (i) or (ii). Refer to Figure 2, the domination relationship between the instances falling in the grey area can be determine directly.

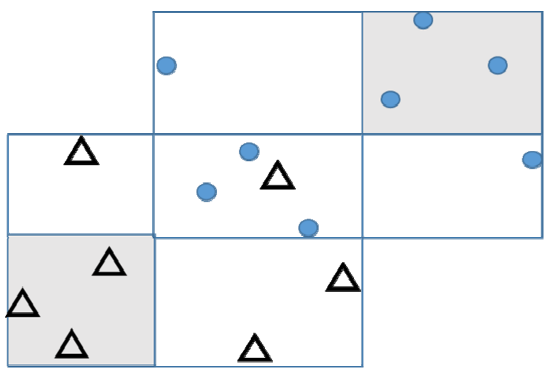

Figure 2. The relationship between the instances

After the previous trimming, for each pair of objects, we only need to consider the instances falling in the incomparable area. In this part, we adopt the advantage of z-order to sort the instances in the incomparable area. As shown in the previous studies, after sorting by z-order, the element is impossible to dominate the other elements listed before it. With this concept, we can get a better MBB and find a tighter bound of the value of *CP*. After this step, we can determine the domination relationship between every pair of instances in main-object and sub-object. For the instance of the main-object which falls in the fully dominating or fully dominated area, we can give the competitiveness score to $1/k_i$ or 0 fairly intuitively. Furthermore, for the instance falls in the incomparable area, we can estimate the best and worst score according to its z-order. Finally, we can get the upper bound and lower bound of *CP*(main-object). If the upper bound of *CP*(main-object) is smaller than the predefined threshold, we can eliminate the object directly. On the contrary, if the lower bound of *CP*(main-object) is no smaller than the threshold, the object can be included in the result directly. For the remaining cases, we need to recursively perform the previous steps to get a tighter bound until the upper bound smaller than the threshold or the lower bound larger than the threshold. The pseudocode is shown in Figure 3.

Input
Uncertain dataset D
Threshold δ
Output
$\{U \in D \mid CP(U) \geq \delta\}$
Method

```
For each U in D
 {
   compute the MBB of U
   CP(U)=0
 }
For each U V pairs in D
 {
   If MBB of U fully dominates MBB of V
     CP(U)=CP(U)+1
   Else if MBB of V fully dominates MBB of U
     CP(V)=CP(V)+1
   Else if MBBs of U and V are incomparable
     {
        Sort the instances according to Z-address
        For i=1 to |U|*|V|
          For j=i to |U|*|V|
          {
            If I_i dominates I_j
              If I_i is an instance in U
                CP(U)=CP(U)+1/(|U|*|V|)
              Else
                CP(V)=CP(V)+ 1/(|U|*|V|)
          }
     }
 }
Output the object with the CP value no smaller
than
```

Figure 3. Proposed algorithm

4 Experimental Results

In this section, a set of experiments was performed to demonstrate the performance of the approach. The simulations were implemented in C and the experiments were run on the computer with a 3.07 GHz Intel Core processor with 16GB of memory.

To show the effectiveness of our approach, following the discussion in [17], we collect the data of National Basketball Association (NBA) statistics from 1991~2005. The dataset contained 1,818 players on 3 attributes (score, assist and rebound). Table 5 lists the top 42 playes ranked by *CP* value. Furthermore, we use bold characters to indicate the player who is included in 0.1 skyline [17]. As shown in the table, the players who are ranked first have a high degree of overlap with the players identified by 0.1 skyline [17]. This result shows that in this model, we can find the players who will also be selected under the model proposed in [17]. In addition, if the basketball Hall of Fame is used as an indicator, according to the experimental result shown in [17], the top 11 players selected by using 0.1 skyline are the hall of fame inductee or hall of fame prospective inductee. And by adapting the CP model, the top 15 players selected by using CP model can reach this criteria. It is worth noting that Reggie Miller (the star of the Hall of Fame) will be ranked as 6 by using our model. However, he will not be selected in a *p*-value skyline (*p* is set to 0.1). By using the proposed model, we can identify the outstanding players and rank them in front of the list.

Table 5. Top 42 players ranked by CP value

Rank	Player Name	CP(K)	Rank	Player Name	CP(K)
1	**Shaquille O'neal**	1.0794	22	Wally Szczerbiak	0.8063
2	**LeBron James**	1.0772	23	**Grant Hill**	0.7894
3	**Michael Jordan**	1.0084	24	Ray Allen	0.7871
4	**Magic Johnson**	1.0067	25	Allan Houston	0.7828
5	**Karl Malone**	0.9934	26	Mark Price	0.7733
6	Reggie Miller	0.9871	27	**Jason Kidd**	0.7689
7	**John Stockton**	0.9682	28	**Paul Pierce**	0.7518
8	**Allen Iverson**	0.9647	29	**Dennis Rodman**	0.7481
9	**Kobe Bryant**	0.9308	30	**Patrick Ewing**	0.7449
10	**Hakeem Olajuwon**	0.9266	31	Terrell Brandon	0.7408
11	**Tim Duncan**	0.9217	32	**Charles Barkley**	0.7393
12	**Steve Nash**	0.9207	33	Ricky Pierce	0.6844
13	**David Robinson**	0.9169	34	Steve Scheffler	0.6754
14	**Clyde Drexler**	0.9126	35	Keith Vanhorn	0.6614
15	**Gary Payton**	0.8965	36	**Dirk Nowitzki**	0.6594
16	**Dwyane Wade**	0.8712	37	Gheorghe Muresan	0.6352
17	Predrag Stojakovic	0.8294	38	Hersey Hawkins	0.6296
18	**Chris Webber**	0.8247	39	Michael Williams	0.5929
19	Alonzo Mourning	0.8165	40	**Scottie Pippen**	0.5632
20	**Kevin Garnett**	0.8126	41	James Blackwel	0.5505
21	Chauncey Billups	0.8106	42	**Lamar Odom**	0.5466

Besides evaluating the effectiveness of the proposed work on real dataset, we use the synthetic datasets to demonstrate the efficiency of our approach. Following the discussion in [14], three optional distributions were considered: correlated, independent, and anticorrelated. The synthetic datasets were generated using the parameters listed in Table 6.

Table 6. Parameters used in the experiments

Parameter	Default Value	Range
Number of objects	8000	2000~10000
Number of instances in an object	20	10~40
Number of dimensions	5	_

Figure 4 shows the effects of the number of objects in the dataset, which indicates that the execution times increase as the number of objects increases for independent, correlated and anticorrelated datasets. Furthermore, as shown in the result, the execution time for independent dataset is larger than those of correlated and anticorrelated datasets. This is because in the independent distribution, the value of the object in each dimension changes drastically, and it is difficult to confirm the relationship between the two objects directly by checking the relationship between the MBBs.

Figure 5 shows the effects of the number of instants of an object in the dataset. The result indicates that the execution times increase as the number of instances increases for independent, correlated and anticorrelated datasets. Furthermore, similar to the previous result, the execution time for independent dataset is larger than those of correlated and anticorrelated datasets.

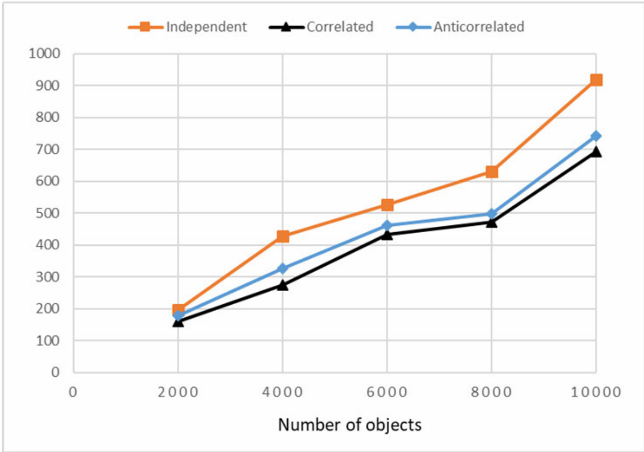

Figure 4. Execution time (in seconds) vs the number of objects

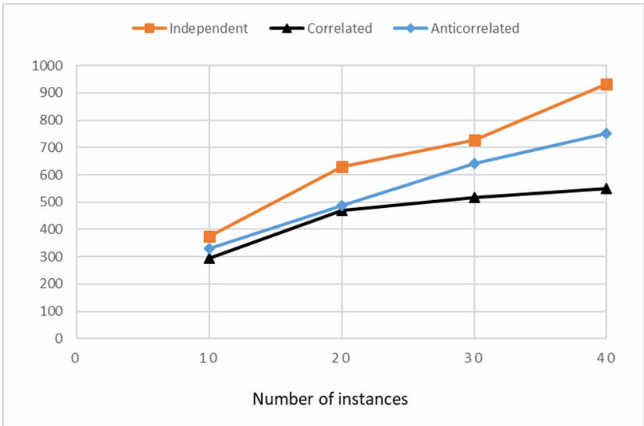

Figure 5. Execution time (in seconds) vs the number of instances of an object

5 Conclusion

In this paper, we discuss the problem of retrieving objects with considerable potential in an uncertain dataset. In real applications, such as environmental monitoring and market analysis, the data often have uncertain characteristics. The uncertainty of the data mainly comes from the data randomness and the limitation of the measuring instrument. By considering the uncertainty property and the concept of domination in skyline query, we propose a measure, named *CP*, to evaluate the goodness of an uncertain object. And the *CP* of an uncertain object *U* is actually the expected number of data objects which cannot dominate *U*.

To show the effectiveness of the proposed method, an experiment was performed on the dataset of National Basketball Association (NBA) statistics from 1991~2005. Experimental results indicate that the proposed method can identify the outstanding players and rank them in front of the list. Furthermore, a set of experiment was performed on a synthetic dataset to demonstrate the efficiency of the proposed method. Experimental results showed that the performance of the proposed method is good in correlated and anticorrelated datasets.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful suggestions. This work was partially supported by the Ministry of Science and Technology, ROC, under contracts MOST 107-2221-E-259-026 and MOST 107-2221-E-259 -008.

References

[1] H. Li, S. Jang, J. Yoo, An Efficient Architecture for Parallel Skyline Computation over Large Distributed Datasets, *Journal of Internet Technology*, Vol. 15, No. 4, pp. 577-588, July, 2014.

[2] T. C. Chang, C. J. Jane, M. Chang, Cluster Validity Indexes to Uncertain Data for Multi-Attribute Decision-Making Datasets, *Journal of Internet Technology*, Vol. 19, No. 2, pp. 533-538, March, 2018.

[3] S. Borzsony, D. Kossmann, K. Stocker, The Skyline Operator, *17th IEEE International Conference on Data Engineering*, Heidelberg, Germany, 2001, pp. 421-430.

[4] J. Chomicki, P. Godfrey, J. Gryz, D. Liang, Skyline with Presorting, *19th IEEE International Conference on Data Engineering*, Bangalore, India, 2003, pp. 717-719.

[5] S. Zhang, N. Mamoulis, D. W. Cheung, Scalable Skyline Computation Using Object-based Space Partitioning, *2009 ACM SIGKDD International Conference on Management of Data*, Rhode Island, USA, 2009, pp. 483-494.

[6] F. N. Afrati, P. Koutris, D. Suciu, J. D. Ullman, Parallel Skyline Queries, *Theory of Computing Systems*, Vol. 57, No. 4, pp. 1008-1037, November, 2015.

[7] K. Mullesgaard, J. L. Pederseny, H. Lu, Y. Zhou, Efficient Skyline Computation in Mapreduce, *17th International Conference on Extending Database Technology*, Athens, Greece, 2014, pp. 37-48.

[8] M. Tang, Y. Yu, W. G. Aref, Q. M. Malluhi, M. Ouzzani, Efficient Parallel Skyline Query Processing for High-Dimensional Data, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 30, No. 10, pp. 1838-1851, October, 2018.

[9] J. Zhang, X. Jiang, W. S. Ku, X. Qin, Efficient Parallel Skyline Evaluation using Mapreduce, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, No. 7, pp. 1996-2009, July, 2016.

[10] J. Pei, W. Jin, M. Ester, Y. Tao, Catching the Best Views of Skyline: A Semantic Approach Based on Decisive Subspaces, *31st International Conference on Very Large Data Bases*, Trondheim, Norway, 2005, pp. 253-264.

[11] J. Pei, A. W. C. Fu, X. Lin, H. Wang, Computing Compressed Multidimensional Skyline Cubes Efficiently, *IEEE International Conference on Data Engineering (ICDE)*, Istanbul, Turkey, 2007.

[12] R. C. -W. Wong, J. Pei, A. W. -C. Fu, K. Wang, Mining Favorable Facets, *13th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, San Jose, CA, 2007, pp. 804-813.

[13] R. C.-W. Wong, A. W.-C. Fu, J. Pei, Y. S. Ho, T. Wong, Y. Liu, Efficient Skyline Querying with Variable User Preferences on Nominal Attributes, *International Conference on Very Large Data Bases*, Auckland, New Zealand, 2008, pp. 1032-1043.

[14] C.-Y. Chan, H. V. Jagadish, K. -L. Tan, A. K. H. Tung, Z. Zhang, Finding k-Dominant Skylines in High Dimensional Space, *2006 ACM SIGMOD International Conference on Management of Data*, Chicago, IL, 2006, pp. 503-514.

[15] G. Lee, Y. -H. Lee, An Efficient Method of Computing the k-Dominant Skyline Efficiently by Partition Value, *3rd International Conference on Information Management*, Chengdu, China, 2017, pp. 416-420.

[16] J. K. Zeng, G. Lee, S. L. Peng, Computing the k-Dominant Skyline Efficiently in the Dynamic Environments, *International Conference on Computational Intelligence and Data Analytics*, Bhubaneswar, India, 2018.

[17] J. Pei, B. Jiang, X. Lin, Y. Yuan, Probabilistic Skylines on Uncertain Data, *33rd International Conference on Very Large Data Bases*, Vienna, Austria, 2007, pp. 15-26.

[18] M. J. Atallah, Y. Qi, H. Yuan, Asymptotically Efficient Algorithms for Skyline Probabilities of Uncertain Data, *ACM Transactions on Database Systems*, Vol. 36, No. 2, pp. 1-28, May, 2011.

[19] M.J. Atallah, Y. Qi, Computing All Skyline Probabilities for Uncertain Data, *28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Rhode Island, USA, 2009, pp. 279-287.

[20] C. Böhm, F. Fiedler, A. Oswald, C. Plant, B. Wackersreuther, Probabilistic Skyline Queries, *18th ACM Conference on Information and Knowledge Management*, Hong Kong, China, 2009, pp. 651-660.

[21] S. Elmi, M. A. B. Tobji, A. Hadjali, B. B. Yaghlane, Selecting Skyline Stars over Uncertain Databases, *Applied Soft Computing*, Vol. 57, No. C, pp. 88-101, August, 2017.

[22] Y. Zhang, W. Zhang, X. Lin, B. Jiang, J. Pei, Ranking Uncertain Skyline: The Probabilistic Top-k Skyline Operator, *Information Systems*, Vol. 36, pp. 898-915, 2011.

[23] K. Zhang, H. Gao, X. Han, Z. Cai, J. Li, Probabilistic Skyline on Incomplete Data, *2017 ACM on Conference on Information and Knowledge Management*, Singapore, Singapore, 2017, pp. 427-436.

[24] Q. Zhang, P. Ye, X. Lin, Y. Zhang, Skyline Probability over Uncertain Preferences, *16th International Conference on Extending Database Technology*, Genoa, Italy, 2013.

[25] J. Zheng, Y. Wang, H. Wang, W. Yu, Asymptotic-Efficient Algorithms for Skyline Query Processing over Uncertain Contexts, *19th International Database Engineering & Applications Symposium*, Yokohama, Japan, 2015, pp. 106-115.

[26] H. Zhu, P. Zhu, X. Li, Q. Liu, Computing Skyline Groups: An Experimental Evaluation, *ACM Turing 50th Celebration Conference - China*, Shanghai, China, 2017, pp. 1-6.

Biographies

Sheng-Fu Yang received the M.S. degree in computer science and information engineering from National Dong Hwa University, Taiwan, in 2013. His research interests are in data mining, and skyline query.

Guanling Lee received the Ph.D. degree in computer science from National Tsing Hua University Taiwan in 2001. She joined National Dong Hwa University in the Department of CSIE, and became a associate professor in 2005. Her research interests include resource management in the mobile environment and data mining.

Shou-Chih Lo received the Ph.D. degree in computer science from National Tsing Hua University, Taiwan. Since 2004, he has been with the Department of CSIE of the National Dong Hwa University, where he is currently an Associate Professor. His research interests are in the area of mobile and wireless networks.