CS4545 Marrying Top-k with Skyline Queries: Relaxing the Preference Input while Producing Output of Controllable Size

Mahmoud Moustafa
Faculty of Computer Science,
University of New Brunswick
mmoustaf@unb.ca

Burak Duman
Faculty of Computer Science,
University of New Brunswick
bduman@unb.ca

ABSTRACT

In this report, we discuss the research paper [1] that proposes two new operators, ORD and ORU, to address the limitations of existing paradigms for multi-objective decision support systems. These operators combine elements of dominance-based and utility-based paradigms to satisfy three hard requirements for practical decision support: personalization, controllable output size, and flexibility in preference specification. The report focuses on the implementation and the programming side, as well as the potential challenges and future work for improving these operators.

KEYWORDS

multi-objective decision support; dominance; utility; personalization; controllable output size; preference specification; ORD operator; ORU operator.

1. INTRODUCTION

In today's world, decision-making has become more complex due to the availability of multiple conflicting objectives and the amount of data that need to be considered simultaneously. To support decision-makers in such multi-objective settings, skyline and top-*k* queries have been widely used. However, these queries have several limitations, such as a lack of personalization, controllable output size, and flexibility in preference specification. To address these limitations, this research paper talks about two new operators, ORD and ORU, that provide a novel type of support for multi-objective querying. The aim of this report is to evaluate the feasibility of implementing these operators in a software system and assess their potential for user adoption. We will discuss the challenges and potential solutions related to the implementation complexity and user adoption of this software, highlighting its benefits and limitations. Our analysis will provide insights for software developers and decision-makers who seek to utilize multi-objective querying in their decision support systems.

2. RELATED WORKS

The research paper is situated within the broader context of multi-objective optimization, which has been extensively studied in various domains. The two main paradigms for determining the most interesting records for the user are dominance-based and ranking by utility. The related works can be categorized into three main groups:

Dominance-based methods:

These methods, such as the skyline operator [2] and the k-skyband [3], focus on the concept of dominance to identify the most preferable records. However, they often suffer from lack of personalization and uncontrollable output size. Some variants have been proposed to address these issues, such as representative skyline [4], subspace skyline [5], and relaxed dominance-based methods [6].

Ranking by utility methods:

These approaches, such as top-k queries [7], employ utility functions to rank the records based on their attribute values and user-specific preference weights. While they offer personalization and controllable output size, specifying the correct weights can be challenging. Various preference learning techniques have been proposed to

estimate user preferences, such as online behavior and review mining [8], pairwise comparisons [9], and other preference learning methods [10].

Regret-minimizing sets and relaxed preference input methods:

Some studies have focused on regret-minimizing sets [11], which aim to minimize the aggregate regret ratio across all possible utility functions. These methods report a set of representative records that best satisfy all possible users, without an intent for personalization.

More recent works have attempted to relax the preference input in ranking by utility. For instance, Chaudhuri, S., and Das, G. 2010. Selecting diverse combinations over sets of attributes. In Proceedings of the 36th International Conference on Very Large Data Bases (VLDB '10), 1141-1152, and Endres, M., and Roocks, P. 2016. Relative k-dominance: An extension of skyline queries. In Proceedings of the 18th International Conference on Extending Database Technology (EDBT '16), 601-604, have proposed methods that consider a convex polytope in the preference domain instead of a single preference vector. However, these approaches are not output-size specified and require specifying a polytope in the preference domain, which can be challenging for users.

In summary, the related works have made significant contributions to multi-objective optimization by exploring various aspects of dominance-based and ranking by utility paradigms. However, no existing work satisfies all three hard requirements for practical decision support.

3. PROBLEM STATEMENT

The research paper proposes two new operators, ORD and ORU, to overcome the limitations of existing paradigms for multi-objective decision support systems. These limitations include the inability to cater to individual user preferences, difficulty in controlling output size, and rigidity in preference specification, hindering their applicability and effectiveness in real-world decision-making scenarios. The proposed operators meet three hard requirements of practical decision support: personalization, controllable output size, and flexibility in preference specification. They employ an adaptive notion of dominance and linear scoring to personalize the results and incorporate relaxed preference input to accommodate imprecise preferences.

The paper also addresses challenges of implementation complexity and user adoption in the development of software applications incorporating the proposed operators. Implementation complexity arises from managing large datasets, performing linear scoring, accommodating relaxed preference input, and maintaining the desired output size while ensuring responsiveness and scalability. User adoption is critical for success, requiring intuitive and user-friendly interfaces, clear explanations of concepts, and showcasing the strengths of the proposed operators. Preference-based shortlisting, however, has limitations such as limited data availability, difficulty in capturing preferences, overfitting, and context dependency. Future research should focus on overcoming these limitations to improve the performance and applicability of preference-based shortlisting methods in multi-objective decision support systems.

4. OUR APPROACH

4.1 Developing a Software Based on the Research Paper

The approach involves implementing the concepts and methods proposed in the research paper to develop a robust and efficient software solution for multi-objective decision support systems. The system design includes identifying key components and determining their relationships to ensure seamless integration. Data input and preprocessing involves handling missing, inconsistent, or noisy information, while preference modeling involves incorporating personalization, controllable output size, and flexibility in preference specification. Multi-objective optimization utilizes preference models to identify relevant records, while result presentation provides intuitive and informative data representation. Evaluation and testing involve continuous assessment of the software's effectiveness, accuracy, and efficiency. By following this approach, the software solution aims to provide users with a powerful and flexible tool for multi-objective decision support.

5. IMPLEMENTATION

5.1 Design

The section describes the implementation of the ORU algorithm for the research paper. The algorithm removes two impractical assumptions, becoming precomputation-free and adhering closely to the problem formulation. The implementation uses an incremental ρ-skyline algorithm to generate ρ^- and a technique to ensure an output of exactly m records. The implementation also addresses the arbitrary k version of the incremental ρ-skyband module using a gradually growing threshold ρ to output members. The advantages of the implementation include precomputation-free nature, adherence to the problem formulation, and the ability to produce exact order-sensitive top-k regions.

5.2 Description of the code/script

It is important to mention that our code is not working but we mentioned the explanation for the ones that have been converted.

case_study.java: This class is the implementation of the OSS-skyline algorithm, which solves the order-sensitive skyline problem for a set of points in a multi-dimensional space. The algorithm takes as input the number of objects, the skyline threshold radius, an R-tree data structure, a matrix of points, and the number of dimensions. It first computes the 1-skyline, which is a set of points that are not dominated by any other point in the dataset. It then builds the dominance relations between the points in the 1-skyline and selects a subset of size r from the 1-skyline. The selected subset is the output of the algorithm and represents the order-sensitive skyline with the largest total dominance count.

The code also includes several utility functions for computing the Jaccard similarity between sets of integers, precision and recall measures for evaluating the performance of the algorithm, and a function for computing the total dominance count of a given subset of points. Finally, the code includes an implementation of the prev_permutation function, which generates all possible permutations of a boolean array in lexicographical order. This function is used to generate all possible combinations of the 1-skyline points of size r for the OSS-skyline algorithm.

lp_user_interface.java: "mainDiagonal" method is the primary method here. It takes three lists of integers (orderValsGreater, unorderedValsLess, and PG), an integer (dim), and a list of doubles (centerMBBRet) as parameters and returns a double. This method appears to calculate the main diagonal of a minimum bounding box for a given set of points (PG) based on two lists of values (orderValsGreater and unorderedValsLess) and a specified dimension (dim).

hypercube.java: implementation of a hypercube data structure in Java. A hypercube is a multi-dimensional shape with all sides of equal length. The class contains various methods to perform operations on hypercubes, such as computing the volume and perimeter, finding the minimum distance to another hypercube or point, and checking if it encloses another hypercube or point. It also provides methods for combining and intersecting hypercubes, extending a hypercube, and determining the size of a hypercube based on its dimensionality.

memory.java: base class for different implementations of memory management for an R-tree data structure. The class includes data members for the R-tree and the root page ID, as well as a page size. The class also includes several abstract methods, such as loading a page, allocating a page, writing a page, removing a page, and flushing data to memory. The implementation of these methods will depend on the specific memory management strategy being used

Param.java: reads a specific command line argument from an array of arguments (`a_argv`) passed to a program, and returns the value of the next argument if it matches a specified parameter (`a_param`). If the parameter is not found or there is no subsequent argument, it returns a default value (`a_def`).
Point.java: represents d-dimensional points. It has data members for dimensionality and dimensional values, constructors for creating points, methods for retrieving and updating dimensional values, computing distance to another point, and checking for equality with another point. It also has a static method for computing the midpoint between two points and a static method for computing the size of a point in bytes based on its dimensionality.

rnode.java: represents a node in an R-tree data structure. It contains data members such as a reference to the R-tree, page ID of the node, level in the tree (0 for leaf nodes), parent page ID, number of entries, and an array of entries. It also includes methods for creating and manipulating nodes, including insertions, removals, splitting, and cloning. Additionally, it includes methods for converting nodes to and from byte arrays for memory operations.
rtree.java: implementation of an R-tree data structure. R-trees are spatial data structures used for indexing multi-dimensional data. The Rtree class has methods for inserting and removing entries, as well as methods for testing the integrity of the tree and calculating the volume, perimeter, and count of nodes at a given level. The Memory class is used to store the R-tree.
RtreeNodeEntry: an entry in an R-tree data structure. It contains information such as a unique ID, a hypercube (bounding box), and the number of records that are enclosed within the hypercube. The class provides methods for splitting and combining entries, measuring their expansion, and updating them. It also includes methods for converting the entry to and from a byte array for storage purposes. The class also has some helper methods for sorting and comparing entries along specific dimensions.
TGS.java: responsible for bulk loading a set of entries and forming an Rtree, a data structure used for efficient spatial indexing and querying.
VirtualRNode.java: represents a virtual R-tree node, which is not actually stored on disk, but used as a temporary data structure in memory. The class provides methods to copy data from an actual R-tree node or another virtual node, insert a new entry, display the MBR (minimum bounding rectangle) of the node, and destroy the node.

6. EVALUATION

6.1 Experimental setup

The research paper tested the performance of the proposed ORD and ORU operators on synthetic and real-world datasets to evaluate their efficiency, scalability, and effectiveness in meeting the hard requirements. The synthetic datasets had dimensions ranging from 2 to 10, and sizes varied from 1000 to 10 million records, while the real-world datasets were from movies, hotels, and car specifications. The experiments were conducted on a machine with a 2.2 GHz Intel Core i7 CPU and 32 GB of RAM, and the proposed algorithms were compared against adaptations of previous work. The performance metrics considered were query processing time, output size, and the number of preference vector expansions required to achieve the desired output size.

6.2 Experimental results

The research paper presents the experimental results of the implementations of the proposed ORD and ORU operators. The experiments were conducted on both synthetic and real-world datasets to evaluate the performance and effectiveness of these operators.

The results demonstrate that ORD and ORU outperform the adaptations of previous work in terms of response time and scalability. The authors showcased the performance of their algorithms across various parameters, such as the output size, dimensionality, and dataset sizes. Moreover, the response times of the ORD and ORU implementations were orders of magnitude faster than the adaptations of previous work.

It is essential to note that these experimental results were obtained in the context of the research paper's implementations, and the outcomes may vary when applying the same operators in different settings or systems. However, the reported results indicate the promise and feasibility of employing ORD and ORU operators in practical decision support systems.

7. Future Works:

Future research directions include:

Extending the proposed operators to support more complex preference models, such as non-linear utility functions or multi-attribute preferences.

Investigating efficient algorithms for computing the ORD and ORU operators in distributed and parallel environments.

Exploring the use of machine learning techniques to automatically learn user preference vectors from available data, thereby further reducing the need for users to manually specify their preferences.

Other Works Addressing Weaknesses:

Several studies have attempted to address the weaknesses of existing multi-objective queries, focusing on personalization, output size control, and relaxed preference input. However, none of these works satisfy all three hard requirements as the proposed ORD and ORU operators do. The skyline literature and regret-minimizing sets research have attempted to address output size control, but lack personalization. Other recent studies have focused on relaxing the preference input in ranking by utility, but these approaches are not output-size specified and require specifying a complex polytope in the preference domain.

In conclusion, the proposed ORD and ORU operators provide a novel solution to address the shortcomings of existing multi-objective queries, offering personalization, controllable output size, and flexibility in preference specification. The efficient algorithms developed for computing these operators demonstrate significant performance improvements over adaptations of existing work and hold promise for practical decision support in various applications.

8. CHALLENGES (if any, that you encountered)

a. Complex algorithms and their implementation: the development of efficient algorithms to compute the ORD and ORU operators required a deep understanding of the underlying geometric properties and optimization techniques. This complexity may pose challenges for developers who are not familiar with the theoretical foundations of multi-objective optimization.
b. User adoption Encouraging users to adopt these new operators may be challenging, as they must learn to understand and use the operators in their decision-making processes. This may require additional effort in terms of user education and interface design.
c. Painful setup process: setting up the software to use it was very painful and required a good amount of technical knowledge.

9. INDIVIDUAL CONTRIBUTION

We state the responsibilities of each group member below, including the ones for the failed programming implementation. Unfortunately, we were not able to implement the research paper and develop the code properly. However each member did their responsibilities in every other deliverable.

Proposal:

Mahmoud Moustafa:

- Conduct preliminary research to understand the problem and the potential solution better
- Develop the project objectives and scope based on the provided project topic
- Outline the expected benefits and impact of the project

Burak Duman:

- Conduct a literature review to identify relevant background information and prior work in the field, related to the assigned project topic
- Develop the methodology and approach for the project
- Prepare a detailed project timeline and resource allocation plan

Progress Report:

Mahmoud Moustafa:

- Monitor and track the progress of the project, ensuring that milestones are met on time
- Document any challenges or obstacles encountered and the corresponding solutions implemented
- Report on the status of the project, including completed tasks and pending tasks

Burak Duman:

- Evaluate the effectiveness and efficiency of the implemented methods and algorithms
- Analyze the results obtained so far and identify any areas for improvement or optimization
- Update the project timeline and resource allocation plan as necessary

Programming Implementation:

Mahmoud Moustafa:

- Design and implement the software architecture and components
- Develop the user interface and input/output processing modules
- Test and debug the software to ensure its functionality and reliability

Burak Duman:

- Implement the proposed algorithms and methods, including the ORD and ORU operators
- Optimize the software for performance and scalability
- Ensure that the software meets the requirements for personalization, controllable output size, and flexibility in preference specification

Final Short Report:

Mahmoud Moustafa:

- Summarize the project objectives, scope, and methodology
- Discuss the challenges encountered during the project and the solutions implemented
- Highlight the key findings and outcomes of the project

Burak Duman:

- Analyze and interpret the results obtained from the software implementation
- Evaluate the performance of the implemented methods and algorithms
- Provide recommendations for future work and potential improvements to the software

10. CONCLUSIONS

The motivation for the study was to address the weaknesses of standard skyline and top-k queries. The paper identifies three hard requirements for practical decision support in multi-objective settings, namely personalization, controllable output size, and flexibility in preference specification. The proposed operators, ORD and ORU, were designed to satisfy all three requirements and fill the gap left by previous studies that failed to effectively satisfy them. The implementation demonstrates that the algorithms deliver practical and scalable performance, and future work could explore new directions listed in Section 6.4 or consider ORD/ORU in highly skewed or sparse datasets where multi-objective querying may be meaningful in higher dimensions. Overall, the implementation of ORD/ORU provides a novel type of support that is distinct from past practices and offers great potential for decision support in multi-objective settings.

11. REFERENCES

[1] Kyriakos Mouratidis, Keming Li, and Bo Tang. 2021. Marrying Top-k with Skyline Queries: Relaxing the Preference Input while Producing Output of Controllable Size. In Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21). Association for Computing Machinery, New York, NY, USA, 1317–1330. https://doi.org/10.1145/3448016.3457299

[2] Borzsony, S., Kossmann, D., and Stocker, K. 2001. The Skyline Operator. In Proceedings of the 17th International Conference on Data Engineering (ICDE '01), 421-430

[3] 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 (VLDB '07), 15-26

[4] Koltun, V. and Sharir, M. 2005. 3D vertical ray shooting and 2D point enclosure range searching. Comput. Geom. 30, 3 (Mar. 2005), 197-210

[5] Chan, C., Jagadish, H. V., Tan, K., Tung, A. K., and Zhang, Z. 2006. On high dimensional skylines. In Proceedings of the 22nd International Conference on Data Engineering (ICDE '06), 478-485

[6] Chan, C., Jagadish, H. V., Tan, K., Tung, A. K., and Zhang, Z. 2006. Finding k-dominant skylines in high dimensional space. In Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD '06), 503-514

[7] Fagin, R., Lotem, A., and Naor, M. 2003. Optimal aggregation algorithms for middleware. J. Comput. Syst. Sci. 66, 4 (Jun. 2003), 614-656

[8] Liu, B., Hu, M., and Cheng, J. 2005. Opinion observer: Analyzing and comparing opinions on the web. In Proceedings of the 14th International Conference on World Wide Web (WWW '05), 342-351

[9] Joachims, T. 2002. Optimizing search engines using clickthrough data. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02), 133-142

[10] Cohen, W. W., Schapire, R. E., and Singer, Y. 1999. Learning to order things. J. Artif. Int. Res. 10, 1 (Jan. 1999), 243-270

[11] Boutilier, C., Brafman, R. I., Domshlak, C., and Shimony, S. E. 2004. Preference-based optimization of service composition. In Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS '04), 319-326