Finding Average Regret Ratio Minimizing Set in Database

Sepanta Zeighami
Hong Kong University of Science and Technology
szeighami@cse.ust.hk

Raymond Chi-Wing Wong
Hong Kong University of Science and Technology
raywong@cse.ust.hk

***Abstract*—Selecting a certain number of data points (or records) from a database which "best" satisfy users' expectations is a very prevalent problem with many applications. One application is a hotel booking website showing a certain number of hotels on a single page. However, this problem is very challenging since the selected points should "collectively" satisfy the expectation of all users. Showing a certain number of data points to a single user could decrease the satisfaction of a user because the user may not be able to see his/her favorite point which could be found in the original database. In this paper, we would like to find a set of k points such that on average, the satisfaction (ratio) of a user is maximized. This problem takes into account the probability distribution of the users and considers the satisfaction (ratio) of *all* users, which is more reasonable in practice, compared with the existing studies that only consider the *worst-case* satisfaction (ratio) of the users, which may not reflect the whole population and is not useful in some applications. Motivated by this, in this paper, we propose algorithms for this problem. Finally, we conducted experiments to show the effectiveness and the efficiency of the algorithms.**

***Keywords*-query processing; average regret ratio;**

I. Introduction

Selecting a certain number of data points (or records) from a database in order to "best" satisfy users' expectations is a prevalent problem with many applications, from recommender systems to search engines. In many situations, the users of such an application is *anonymous*, that is, s/he is not registered on the website or has not logged into his/her account. No personal information is available regarding the specific preferences of an anonymous user. Thus, to select data points for an anonymous user, only information about the users in general (possibly refined by user's location) can be utilized.

In general, when a number of points are selected for a user, no specific information may be known about the user. The websites usually do not have an interface to ask the users to input their preferences (beyond merely refining a query) and even if they did, the users might not be willing or capable to provide their exact preferences. Moreover, a user will have a low *satisfaction* level if s/he cannot see his/her favorite data point among the points shown to the user. Thus, an accurate formulation of the level of satisfaction of the user is required for us to be able to select data points that better satisfy the user, without knowing the user's preferences.

Utility functions are used to quantify a user's satisfaction from a *single point*. Using the concept of utility functions, *regret ratio* [1]–[6] has been proposed that measures how well a *set of points* satisfies a user compared with when the user has seen the entire database. The setting described above corresponds to the problem of selecting a set of points for a user when the user's utility function is unknown (i.e., we have no information about users' preferences). Thus, we cannot select a set of points for the user in order to minimize the user's regret ratio. Instead, we can focus on selecting a set of point that minimize the user's regret ratio on expectation (i.e., on average), based on the probability distribution of the utility functions.

In this paper, we study the problem of *finding the average regret ratio minimizing set (FAM)* [7]. Given a database D and a probability distribution Θ of utility functions, we want to find a set S of k points in D such that the expected regret ratio of a user is the smallest.

Most studies on regret ratio so far have focused on the k-regret queries that minimize the maximum regret ratio. [2]–[4] show that solving a k-regret query is NP-hard and existing studies [1]–[6] focused on improving the efficiency of the algorithms and improving the quality of the result. Minimizing maximum regret ratio provides a worst-case guarantee on the regret ratio of all the users. However, as opposed to average regret ratio, maximum regret ratio disregards the probability distribution of the utility functions. But it can be important to obtain a lower regret ratio for the utility functions that are more probable and occur more frequently. Moreover, maximum regret ratio does not account for the distribution of the regret ratio among the users. Two sets can have the same maximum regret ratio even if the regret ratio of a large proportion of the users in one is smaller than the other. But average regret ratio of the two sets will be different. It is confirmed empirically by our experimental results on real datasets that the vast majority of the users will have a lower regret ratio if we minimize average regret ratio instead of maximum regret ratio (see [8]).

Skyline queries [9] have also been used to address the problem by finding a set of points which are not dominated by any other points in the database. However, the size of the answer of skyline queries is uncontrollable and can be very large. In addition, [10] considered the distribution Θ of the

utility functions and proposed the *k-hit query* which is to find a set S of k points such that the *probability* that at least one point in S is the best point of a user is maximized. The answer to this query becomes less convincing if we care about not only users who regard the points in the answer set as their best points but also users who do not regard the points in the answer set as the best points.

To address the shortcomings of the existing works, in this paper, we study the FAM problem. [7] first studied the FAM problem and proposed a greedy algorithm to solve it. Here, we show that FAM is NP-hard, and we prove a dynamic programming algorithm for the FAM problem when the dataset contains two dimensions, and provide experimental results that show the advantage of average regret ratio over existing methods. Our technical report [8] contains the more detailed theoretical and empirical study of the problem together with proofs of the theoretical results of this paper as well as [7].

The rest of the paper is organized as follows. Section II formally defines our FAM problem. Section III shows the greedy algorithm. Section IV presents our dynamic programming algorithm in a 2-dimensional case. Section V presents our experimental results. Section VI presents the conclusion and the future work.

II. Problem Definition

Given a database D containing n points, we want to select a set S containing k points that best satisfy users' expectations. We first define how users' feelings towards a *single* point are captured. A *utility function*, f, is a mapping $f : D \rightarrow R_{\geq 0}$, that denotes how much a user *likes* a point. The *utility* of a point p with respect to a user with utility function f is denoted by $f(p)$.

To capture a user's feeling towards a *set* of points, f's *satisfaction* with respect to S denoted by $sat(S, f)$, is defined to be $\max_{p \in S} f(p)$, or 0 if S is empty. A point p is said to be f's *best point* in S if $p = \arg\max_{p \in S} f(p)$. Then, the user's regret ratio is defined as follows.

Definition 1 (Regret and regret ratio [6])**.** *Let S be a subset of D. For a user whose utility function is f, when s/he sees the set S instead of D, the* regret *of f with respect to S, denoted by $r(S, f)$, is defined to be $sat(D, f) - sat(S, f)$ and the* regret ratio *of f, denoted by $rr(S, f)$, to be $\frac{r(S,f)}{sat(D,f)}$.*

The regret ratio of a user with respect to a set S captures how dissatisfied this user is if this subset S of the database D, instead of the whole database, is shown to this user.

Consider F, the uncountable set of all possible utility functions (the countable case of F is a simple extension and is analyzed in our technical report [8]). We let Θ denote the distribution of the utility functions in F, and let $\eta(f)$ be the probability distribution function for utility functions f in F corresponding to Θ. Note that $\int_{f \in F} \eta(f) df = 1$.

Definition 2 (Average Regret Ratio)**.** *Let F be a set of users with the probability density function $\eta(.)$ corresponding to a probability distribution Θ and S be a subset of D. The* average regret ratio *of F for S, denoted by $arr(S)$, is defined to be $\int_{f \in F} rr(S, f)\eta(f) df$.*

In the above formulation, $arr(S)$ is the expected value of the regret ratio of a user when the users' utility functions follow the distribution Θ. In the rest of this paper, we focus on the most general case of F (i.e., when F is the set of all continuous utility functions in the space $R^n \geq 0$) and we assume that the utility value for any point is at most 1. Note that the distribution Θ allows us to select for each instance of the problem which utility functions should be considered and how probable they are.

We are ready to present the problem discussed in this paper, called *Finding Average Regret Minimizing Set (FAM)*, as follows.

Problem 1 (Finding Average Regret Minimizing Set (FAM))**.** *Given a positive integer k, and a probability distribution Θ we want to find a set S containing k points in D such that $arr(S)$ is the smallest (i.e., $S = \arg\min_{S' \subseteq D, |S'|=k} arr(S')$).*

The following theorem shows that FAM is NP-hard for a general probability distribution.

Theorem 1. *Problem FAM is NP-hard.*

The NP-hardness result holds when the specification of the general probability distribution is allowed to be of non-constant size, that is, the probability distribution of FAM can be any general probability distribution and it does not have to be specified by at most a constant number of parameters.

III. Algorithm for General Case

In the general case of FAM, when the set of utility functions is continuous and can have any probability distribution, the problem is NP-Hard and we focus on providing an approximate algorithm for the problem. We first presented the algorithm in [7] and its more detailed analysis can be found in [8]. The approximation algorithm Greedy-Shrink initializes the solution set S to the whole database and iteratively removes one point from the current solution set S in a way that the average regret ratio of the resulting set is the smallest. This continues until the number of remaining points in S is at most k. Using the properties of the average regret ratio, it can be theoretically shown that the quality of the solution returned by Greedy-Shrink is within a factor of the optimal solution.

IV. Algorithm on Dataset Containing Two Dimensions

The FAM problem is NP-hard for a general continuous probability distribution. Here, we consider a special case of the problem with continuous distribution of linear utility

functions and provide an exact algorithm that can solve the FAM problem optimally when the dimensionality of the database is two.

First, note that the linear utility functions are of the form $f(p) = w_1 p[1] + w_2 p[2]$ where $p[1]$ and $p[2]$ are the first and second attributes of the point p, and w_1 and w_2 are the weights of the utility function for each dimension. We can consider (w_1, w_2) as a vector, and it is easy to see that scaling the vector does not change the regret ratio of a utility function from any set. Hence, we only need to consider the direction of the vector, which we can measure by the angle it makes with the first dimension. Therefore, in this section, an angle θ is used to represent the set of utility functions that make the angle θ with the first dimension, that is, $\theta = \arctan(\frac{w_2}{w1})$. We let $F_{\theta_l}^{\theta_u}$ be the set of utility functions whose angle is between θ_l and θ_u. Additionally, let $\theta_{i,j} = \arctan(\frac{p_j[2]-p_i[2]}{p_i[1]-p_j[1]})$. Intuitively, $\theta_{i,j}$ is the slope of the line that passes through origin and divides the space of utility functions into two subspaces based on whether they prefer p_i over p_j. Finally, we let $\theta_{i,n+1} = \frac{\pi}{2}$ for simplicity of notation.

In our discussion, we make sure our dataset only includes skyline points and that the points are sorted in descending order of their first dimension. Therefore, if $i < j$, then $p_j[1] \leq p_i[1]$ and because they are in the skyline, $p_j[2] \geq p_i[2]$. Moreover, we limit the set of utility functions to $0 \leq w_1, w_2 \leq 1$.

A. Recursive Formulation

Let $arr^*(r, i, \theta)$ be the optimal solution to the following problem: given that the point p_i is already selected and is the best point for utility function θ, choose at most r points to minimize the average regret ratio of users in $F_{\theta}^{\frac{\pi}{2}}$. Moreover, let $arr(S, F_{\theta_l}^{\theta_u})$ be the average regret ratio of the set S over the utility functions in $F_{\theta_l}^{\theta_u}$. The optimal solution to FAM is now $\min_{1 \leq i \leq n} arr^*(k-1, i, 0)$, and we have the following recursive formulation.

Theorem 2. *Given an integer r, and an angle θ_l, $0 \leq \theta_l \leq \frac{\pi}{2}$, with base cases $arr^*(0, i, \theta_l) = arr(\{p_i\}, F_{\theta_l}^{\frac{\pi}{2}})$ and $arr^*(r, i, \frac{\pi}{2}) = 0$, it hods that $arr^*(r, i, \theta_l) = \min_{i<j \leq n+1, \theta_{i,j} \geq \theta_l} arr(\{p_i\}, F_{\theta_l}^{\theta_{i,j}}) + arr^*(r-1, j, \theta_{i,j})$*

B. Dynamic Programming Algorithm

First we find the skyline of the dataset and sort the points by their first dimension. Then we use the recursive formulation to solve the problem. Finally, when for all i, $arr^*(k-1, i, 0)$ has been calculated, we go through all the n possible values and choose the one with the smallest average regret ratio as the optimal solution.

We need to discuss calculating $arr(\{p_i\}, F_{\theta_l}^{\theta_{i,j}})$. Note that θ_l, in the recursive call is always equal to $\theta_{i,z}$ for some integer z, except for the first function call when $\theta_l = 0$. Therefore, for every i, there are at most $n+1$ possible values for θ_l and there are at most $n+1$ possible values for $\theta_{i,j}$ (because j can be equal to $n+1$). Thus, there are $O(n^3)$ different values of $arr(\{p_i\}, F_{\theta_l}^{\theta_{i,j}})$ possible which we can precompute. Using the definition of the average regret ratio, we can write $arr(\{p\}, F_{\theta_l}^{\theta_u})$ as

$$\sum_{1 \leq i \leq n} \int_0^1 \int_{c_i^l w_1}^{c_i^u w_1} (1 - \frac{w_1 p[1] + w_2 p[2]}{w_1 p_i[1] + w_2 p_i[2]}) \eta(f) dw_2 dw_1$$

for appropriate c_i^l and c_i^u values (our technical report [8] contains the detailed treatment). The exact calculation of the average regret ratio also depends on the choice of $\eta(f)$. For instance, for a uniform distribution where $\eta(f) = 1$, we can integrate the expression exactly and provide a closed-form solution for each integral.

Time Complexity. First, the algorithm finds the skyline points, sorts them and computes $arr(\{p_i\}, F_{\theta_l}^{\theta_{i,j}})$ for all i, j, θ_l, which takes $O(n^4)$ as there are total of $O(n^3)$ different possible values and each arr evaluation takes $O(n)$ (to evaluate the sum of the n integrals). Filling the arr^* table requires filling $O(kn^2)$ elements, each of which take $O(n)$, which is $O(kn^3)$. Finally, finding an i for which $arr^*(k-1, i, 0)$ is minimum needs a linear scan of the elements and $O(n)$ time. Therefore, overall, the algorithm takes $O(n^4)$.

V. Empirical studies

We conducted experiments on a workstation with 2.26GHz CPU and 32GB RAM. All programs were implemented in C++. In Section V-A, we compare the solution set based on the average regret ratio studied in this paper with two solution sets studied in previous papers to study its usefulness. Then, we present the experimental results based on the average regret ratio in Section V-B. Our technical report [8] contains a comprehensive empirical evaluation.

A. Avg. Regret Ratio vs. Max. Regret Ratio vs. k-Hit

In this experiment, we used the NBA dataset from 2013 to 2016 containing statistical records of 664 NBA players. In this experiment, in the absence of any information about utility functions, we used linear and uniformly distributed utility functions.

We executed our proposed algorithm designed for the average regret ratio, greedy algorithm in [6] designed for the maximum regret ratio and algorithm in [10] designed for k-hit to generate the sets S_{arr}, S_{mrr} and S_{k-hit} respectively of 5 NBA players. These three sets could be found in Table I.

In this experiment, we compare the "goodness" of the set S_{arr} with the two sets S_{mrr} and S_{k-hit} based on a "subjective" online survey manner and an "objective" external statistics manner.

We asked 702 participants with basic NBA knowledge to select one set among S_{arr}, S_{mrr} and S_{k-hit} which collectively contains better players. According to the response

S_{arr}	S_{mrr}	S_{k-hit}
Stephen Curry	LaMarcu Aldridge	Stephen Curry
Kevin Durant	DeMarcus Cousins	Kevin Durant
James Harden	Stephen Curry	James Harden
DeAndre Jordan	George Hill	Draymond Green
Russell Westbrook	Ramon Sessions	Russell Westbrook

Table I
THREE SETS OF 5 PLAYERS COMPUTED BASED ON THE AVERAGE REGRET RATIO (ARR), THE MAXIMUM REGRET RATIO (MRR) AND THE k-HIT QUERY (k-HIT) (I.E., S_{arr}, S_{mrr} AND S_{k-hit})

Top 1 to 5	Top 6 to 10
Stephen Curry	Derick Rose
LeBron James	Russell Westbrook
Kobe Bryant	Kyrie Irving
Kristaps Porzingis	James Harden
Kevin Durant	Jimmy Butler

Table II
TOP 10 NBA PLAYERS IN 2016 ACCORDING TO THE NUMBER OF JERSEYS SOLD

Dataset	d	n
House-hold-6d	6	127,931
Forest Cover	11	100,000
Yahoo! Music	-	8,933

Table III
REAL DATASETS' INFORMATION

to the survey question, about 56%, 17% and 27% of the participants preferred S_{arr}, S_{mrr} and S_{k-hit}, respectively, which suggests that the result based on the average regret ratio is more preferred compared with the other two results.

Next, consider the result based on the external statistics about NBA player jersey sales (Table II). Surprisingly, 4 players out of 5 players in S_{arr} and S_{k-hit} are in the top-10 players based on the number of jerseys sold (Table II). However, only 1 player out of 5 players in S_{mrr} is in the top-5 and the top-10.

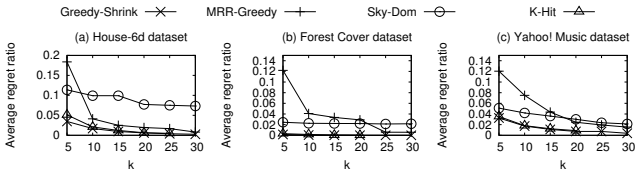

Figure 1. Effect of k on average regret ratio of real datasets

B. Experiments for Average Regret Ratio

We used three real datasets commonly used in the existing studies for skyline queries and top-k queries, namely *Household-6d* (http://www.ipums.org), *Forest Cover*, *Yahoo!music* dataset (http://webscope.sandbox.yahoo.com/catalog.php?datatype=c) to evaluate our algorithm. In the first two datasets, the utility functions used are linear and their distribution is uniform, but for *Yahoo!music* we learn the distribution of users preferences based on the rating they provide. The number of dimensions and the data size of each of these real datasets can be found in Table III.

We compared our proposed algorithms called GREEDY-SHRINK and DP (described in Section III and IV) with the 3 existing algorithms, namely MRR-GREEDY [6], SKY-DOM [11] and K-HIT [10]. MRR-GREEDY is the greedy algorithm [6] designed to find the solution set based on maximum regret ratio. SKY-DOM is an algorithm in [11] which selects k points that together dominate the most number of points in the skyline of a dataset. K-HIT is a top-k algorithm proposed by [10] that uses a probabilistic approach for selecting k points.

In the k-hit algorithm, we set parameters ϵ and δ to be 0.1 such that the setting matches the error and confidence parameter for sampling in GREEDY-SHRINK which is set to $N = 10,000$.

Figure 1 shows the average regret ratios of the solutions returned by different algorithms based on the datasets. GREEDY-SHRINK has the smallest average regret ratio among all the algorithms, and K-HIT has a slightly larger average regret ratio. However, SKY-DOM algorithm does not work well on real datasets and returns an average regret ratio much larger than the other algorithms. Furthermore, the average regret ratio of the points returned by SKY-DOM does not change significantly when the number of points returned increases.

VI. CONCLUSION

We considered the problem of selecting a number of representative points from a database. The problem is concerned with the happiness, or utility, of the users who see the selected points instead of the whole database.

Acknowledgment. The research is supported by HKRGC GRF 16214017 and ITS/227/17FP.

REFERENCES

[1] A. Asudeh, A. Nazi, N. Zhang, and G. Das, "Efficient computation of rregret-ratio minimizing set: A compact maxima representative," *SIGMOD*, 2017.

[2] P. K. Agarwal, N. Kumar, S. Sintos, and S. Suri, "Efficient algorithms for k-regret minmizing sets," *SEA*, 2017.

[3] W. Cao, J. Li, H. Wang, K. Wang, R. Wang, R. Wong, and W. Zhan, "k-regret minimizing set: Efficient algorithms and hardness," *ICDT*, 2017.

[4] S. Chester, A. Thomo, S. Venkatesh, and S. Whitesides, "Computing k-regret minimizing sets," *VLDB*, 2010.

[5] P. Peng and R. C. W. Wong, "Geometry approach for k-regret query," *ICDE*, 2014.

[6] D. Nanongkai, A. D. Sarma, A. Lall, R. J. Lipton, and J. Xu, "Regret-minimizing representative databases," *VLDB*, 2010.

[7] S. Zeighami and R. C.-W. Wong, "Minimizing average regret ratio in database," *SIGMOD*, 2016.

[8] S. Zeighami and R. C. W. Wong, "Technical report," http://www.cse.ust.hk/~raywong/paper/arr-technical.pdf, 2018.

[9] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," *ICDE*, 2001.

[10] P. Peng and R. C. W. Wong, "k-hit query: Top-k query with probabilistic utility function," *SIGMOD*, 2015.

[11] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang, "Selecting stars: The k most representative skyline operator," *ICDE*, 2007.