Minimizing *Average Regret Ratio* in Database

Sepanta Zeighami
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
szeighami@connect.ust.hk

Raymond Chi-Wing Wong
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
raywong@cse.ust.hk

ABSTRACT

We propose "average regret ratio" as a metric to measure users' satisfaction after a user sees k selected points of a database, instead of all of the points in the database. We introduce the average regret ratio as another means of multi-criteria decision making. Unlike the original k-regret operator that uses the maximum regret ratio, the average regret ratio takes into account the satisfaction of a general user. While assuming the existence of some utility functions for the users, in contrast to the top-k query, it does not require a user to input his or her utility function but instead depends on the probability distribution of the utility functions. We prove that the average regret ratio is a supermodular function and provide a polynomial-time approximation algorithm to find the average regret ratio minimizing set for a database.

1. INTRODUCTION

Representing a set of points with a few data points which best satisfy users' expectations is a problem with many applications. As an example, consider a website that allows users to find and book hotels. Imagine the case that users' preferences are not known. The user may not know exactly what kind of a room he or she is looking for, or may not be willing to input his or her preferences. Besides, the website owner may want to give suggestions to a user before he or she inputs his or her criteria. A problem now will be that which hotels should be shown to the user.

This problem has been treated in the literature using different approaches. Skyline queries proposed by Borzsony et al. [1], which select the points to be in the *skyline* of a dataset, and top-k queries, which select k points based on a specific user's preferences, are two ways of addressing the problem.

Another approach towards the problem is using the idea of *regret ratio*. The regret ratio of a user after seeing a subset of the points in a database is a measure of how disappointed the user is with the selected points. Hence, the regret ratio measures how "unhappy" a user is with a subset of points. As such, minimizing the regret ratios of the users is a suitable means of selecting representative points from a database.

In this paper, we use the idea of the regret ratio to select a few representative points from a database. We propose the measure of *average regret ratio* which evaluates how well the points are selected by calculating the average of the regret ratios of all the users. Using the average regret ratio, we can measure how users, on average, feel about the points. Hence, we can select the points that will on average be the most suitable for the users by minimizing the average regret ratio. In addition, the selection of the points using the average regret ratio, unlike the top-k queries, does not require the users to input their preferences, and in contrast to the skyline queries, returns a fixed number of points independent of the number of criteria involved.

Regret ratio has been used by Nanongkai et al. [4] who introduced the k-regret operator, an operator which outputs k points from a database to minimize the maximum regret ratio of a user. However, the k-regret operator suffers from the drawback that it only considers the regret ratio of the most unhappy user, and will be skewed towards the least satisfied users only, ignoring the other users. Hence, with only few very dissatisfied users and many satisfied ones, the maximum regret ratio may not be an accurate measure of the satisfaction of the users. On the other hand, the average regret ratio will not be biased towards a few dissatisfied users, and will give a better impression of how a user in general feel towards the selected points.

In what follows, we use the minimization of the average regret ratio to select k data points from a database. For this, we first define the average regret ratio formally and discuss how it can be calculated. Then, we present an algorithm that returns k data points which minimize the average regret ratio to within a small factor of the best possible solution and we present our experimental results on how well the algorithm works in comparison with other existing methods.

2. PROBLEM DEFINITION

To define the problem, we need to introduce a few concepts. First, we assume that the happiness of the users can be expressed using a utility function. Hence, a utility function, f, is a mapping $f : R_+^d \rightarrow R_+$ where $f(p)$ is the utility of a point p for a user with the utility function f. A utility function shows how "happy" a user is with a data point. Then, we define the regret ratio of a user with the utitliy function f after seeing a subset S of a database D to be $rr_D(S, f) = \frac{\max_{p \in D} f(p) - \max_{p \in S} f(p)}{\max_{p \in D} f(p)}$. As such, the regret ra-

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMOD/PODS'16 June 26 - July 01, 2016, San Francisco, CA, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3531-7/16/06.

DOI: http://dx.doi.org/10.1145/2882903.2914831

tio measures how dissapointed (or unhappy) a user is after seeing a subset of the database, by calculating the maximum possible utility or satisfaction the user can gain from seeing the subset and the original database. Then, we can define the average regret ratio as $arr_D(S) = \int_{f \in F} rr_D(S, f)\eta(f)df$, where $\eta(f)$ is the probability distribution function of the utility functions of different users and F is the set of all of the utility functions. The average regret ratio, therefore, is the expected regret ratio of a user. With this, the general problem of selecting k representative points from a database can be written as follows.

Problem Definition. Given a d-dimensional database D of size n, a set F of utility functions of size N, the probability distribution function of the utility functions $\eta(.)$ and an integer k, find a set S, where $S \subseteq D$, of cardinality k for which $arr_D(S)$ is the least. i.e.

$$S = argmin_{S' \subseteq D, |S'|=k} arr_D(S')$$

3. CALCULATION OF THE AVERAGE REGRET RATIO

The calculation of the average regret ratio, $arr_D(\cdot)$, involves calculating the average of the regret ratios of all the utility functions. If the set of the utility functions is defined on a discrete space, we can calculate the average regret ratio by summing for all the utility functions their regret ratios multiplied by their probabilities. But, if the distribution of the utility functions is continuous, the calculation of the average regret ratio requires the calculation of a d-dimensional integral. Since the calculation of such an integral is costly, we use a sampling approach instead. We select N utility functions based on the probability distribution of the utility functions and calculate the average regret ratio by taking the average of regret ratios for these N sampled utility function. For this, we need to choose a value for N that approximates the true value of the average regret ratio with a high confidence and within a reasonable error parameter. We prove that given a confidence parameter $\sigma \in [0, 1]$ and an error parameter $\epsilon \in [0, 1]$, if the sampling size N is at least $\frac{3ln(\frac{1}{\sigma})}{\epsilon^2}$, then with the confidence of at least 1 - σ, the calculated average regret ratio is within ϵ of its true value.

4. ALGORITHM

We provide an approximation algorithm that outputs k points from a dataset that minimize the average regret ratio of a user. We first prove that the average regret ratio is a monotonically non-increasing supermodular set function. Based on this and an algorithm suggested by Il'ev [2] for minimizing a supermodular set function, we propose a greedy algorithm that minimizes the average regret ratio. This greedy algorithm initializes the solution set S to be equal to the dataset D. Then, it iterates from n to $n - k$ and at each iteration removes the point from S whose removal increases the average regret ratio the least. If Alg is the solution returned by this algorithm and Opt is the optimal solution, and $h(S) = arr_D(S)$, then, $\frac{h(Alg)}{h(Opt)} \leq \frac{e^t - 1}{t}$, where $t = \frac{s}{1-s}$ and s is h's steepness, where steepness is the maximum possible marginal decrease of a function.

The calculation of the average regret ratio based on the method explained above takes time $O(dnN)$ and there are $O(n^2)$ iterations in the greedy algorithm. So, the total running time of the algorithm is $O(dNn^3)$. This, however, can be improved empirically using an R*-tree data structure and by means of heuristics. For instance, at each iteration of the algorithm, instead of going through all the N users to calculate the average regret ratio, we can keep track of the users whose regret ratio changes in each iteration of the algorithm, and calculate the regret ratios only for those users.

5. EXPERIMENTS

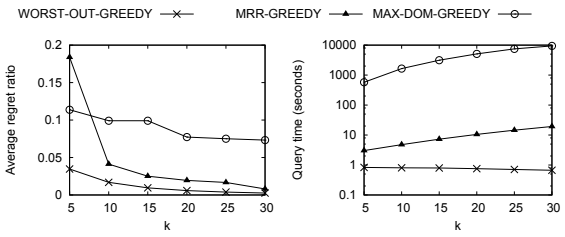

Figure 1: *Results on the House-6d dataset, varying* k

We ran experiments on real and synthetic datasets to measure how well our proposed algorithm works in comparison with other existing algorithms. The algorithms were implemented in C++ and were run on machines with 2.26GHz CPUs and 32GB RAM. For the sake of space, we only present one of the results here. A real dataset with 127,931 points in 6 dimensions called *House-6d* (http://www.ipums.org) was used in the experiment, with 10,000 utility functions sampled from a uniform distribution on the linear class of the utility functions (note that the distribution of the utility functions in real world can be obtained from search logs or rating systems, by recording users' activities and using machine learning techniques). The algorithm proposed by us is called WORST-OUT-GREEDY and the two other algorithms are MRR-GREEDY, proposed by Nanongkai et al. [4], designed for the *maximum regret ratio* operator and MAX-DOM-GREEDY proposed by Lin et al. [3], which selects k points that dominate the most number of points in the skyline of the dataset.

As illustrated by Figure 1, WORST-OUT-GREEDY results in a very low average regret ratio and the average regret ratio for the algorithm MRR-GREEDY drops drastically when k, the number of the selected points, increases. Besides, MAX-DOM-GREEDY results in a larger average regret ratio. The running time of MAX-DOM-GREEDY is smaller than a second for all values of k, while the running time of MRR-GREEDY increases to more than 10 seconds when k increases. However, MAX-DOM-GREEDY has a much larger running time.

References

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

[2] V. P. Il'ev. An approximation guarantee of the greedy descent algorithm for minimizing a supermodular set function. *Discrete Applied Mathematics*, 114(1-3):131–146, October 2001.

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

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