Ranking Object under Team Context

Xiaolu Lu[1], Dongxu Li[1], Xiang Li[†1], and Ling Feng[2]

[1] School of Software, Nanjing University
{mf1232050,mf1332027,lx}@software.nju.edu.cn
[2] Department of Computer Science and Technology, Tsinghua University
fengling@tsinghua.edu.cn

Abstract. Context-aware database has drawn increasing attention from both industry and academia recently by taking users' current situation and environment into consideration. However, most of the literature focus on individual context, overlooking the team users. In this paper, we investigate how to integrate team context into database query process to help the users' get top-ranked database tuples and make the team more competitive. We review naive and propose an optimized query algorithm to select the suitable records and show that they output the same results while the latter is more computational efficient. Extensive empirical studies are conducted to evaluate the query approaches and demonstrate their effectiveness and efficiency.

1 Introduction

Millions of users take portable devices in the palm of their hands. It leads to the rapid development of context-aware database whose users have great expectations of getting suitable query results based on their ambient environment. At the same time, context-aware query has been widely explored to tackle with the many-answers problem to get rid of overwhelming information. Essentially, these applications keep context information to predict users' preferences. Researches in context-aware query have mainly focused on contexts from sensors and user profiles rather than the users' organization-level context i.e the team context. Recently, the problem of context-aware database query has drawn increasing attention from both industry and academia. To cope with the problem many approaches have been proposed and can be divided into two categories: *qualitative* and *quantitative*. Qualitative approaches model the user preference as partial order and apply logic tools to reason the user's intention [14]. On the other hand, quantitative approaches compute the users' satisfaction by score function [8]. However, most of them are based on the individual context. In [7], group context is taken but the group cannot change during the query process. In this paper, we propose the problem of ranking under team context(RTC) which queries the database from a team's perspective and aims to helping the users have a more competitive context by ranking and replace some team component with top-ranked tuples. For example, NBA teams are preparing the roster

[†] Xiang Li is the corresponding author.

and select the prospective player in hope of qualify for the play-offs(finals of NBA) in the next season. To this end, we need to consider the team context in a united group to query the player database for the best player,i.e., who does the team need to acquire in the coming season to make itself a serious candidate for play-offs and which player in the current team should be included in a trade. To the best of our knowledge, this work is the first to focus on team context query while traditional context-aware methods relies on individual context. When taking the whole organization background into consideration, querying becomes more practical and convenient for company customers. Moreover, team context-aware query make it easy to get different query results from different layers of hierarchy which meets perfectly with the innate characteristics of many contexts. The brute force method can be quite inefficient due to excessive I/O overhead. In an effort to handle the limitation, we introduce an I/O-efficient approach RTC* based on Nearest Neighbour. RTC* calculate the exact virtual component the user need to replace with,and map it to the database space. With nearest neighbour technique, we offer the ranking of query results. We prove that RTC* can produce the same results as the brute force method.

We summarize our key contributions as follows:

- We define the RTC problem.
- We propose the solution to the RTC problem based on NN-indexing and prove its correctness.
- We evaluate our algorithms by experiments in terms of effectiveness and efficiency.

The rest of the paper is organized as follows: Section 2 describes related work with a comparison. Section 3 defines the RTC problem and section 4 proposes our method with a review of baseline method. Section 5 presents the experiments with analysis of the results. Section 6 is conclusion and future work.

2 Related Work

Object ranking under team context is a kind of context-aware query processing, aiming at helping systems provide query results after understanding the real intentions behind the queries. To be more specific, our work handled the context that has a team property with a goal of being more competitive and approaching teams with higher rank. Researches in field of context-aware query can be roughly divided into two categories: qualitative and quantitative. In qualitative strategies: preference over database tuples are calculated by score functions. [8, 12] In quantitative strategies: logical rules are hard coded to database system to infer the users' preferences. [15] But group or team context is overlooked for quite a long time. Recently, researches on group preferences have been reported. In [13], Stefanids *et al.* generalized their previous work on hierarchical context model to tackle the needs of a group. In [7], Li and Feng propose several methods to meet most of the people's contexts. However, all these work consider group as union of individuals or most of the members. In our work, team context is take in its entirety. The object selection based on team context is to make the team closer to its rivals. Context tackled in this paper is formed by objects from the object space, which have not been exploited.

k-NN algorithm was one of the most widely used approaches in many fields, first proposed by [1] and continuously improving and refining for specific purpose, especially in spatial databases and sensor networks. Moreover, applying k-NN approach in high dimensional data has raised many attentions. [2] proposed a new method for performing data processing using k-NN in high dimensional data and provided a lower bound of distance between feature vectors. [6, 16] reviewed and put forward a method with hybrid index techniques for solving so called "curse of dimensionality" problems. The state-of-art high dimensional indexing technique iDistance proposed by [4] to enhance efficiency of existing approaches. Recently, [17] propose G-tree index for finding the k nearest objects to the given location. [11] has carefully reviewed the skills in partitioning the data space by iDistance.

3 Prelimilaries

3.1 A Motivated Example

Consider an example in NBA. A fact is that if the games winning of one team ranks top 10 in regular seasons, it would be guaranteed to enter into play-offs. What should a team ranked 11st∼20th do for entering into play-offs?

Assume a team C ranked 17th in NBA wants to enter into play-offs. From the team's view, if the team could approach or even supersede one of the top 10 teams, its chance for entering into play-offs will becomes greater. We refer the team to be surpassed by current one as the target team.

To achieve this goal, usually one player in C will be exchanged with another bought in the transaction. Which pair of players should be selected for fulfilling this goal is a challenging question needed to be answered.

Similar scenarios will also occurred in other area, such as in teams of software developers, clusters of computers etc. Motivated by those ones, Problem solved in this paper can be interpreted as rank the objects and select the ones served as the substitution of a objects in the team.

3.2 Problem Formulation

Given an object space $\mathbb{O}$ with n d-dimensional objects. Team context(TC) in our paper is defined as a context C formed by m objects $\{O_1, O_2, .., O_m, \forall O_m \in C, O_m \in \mathbb{O}\}$ like how teams formed in NBA. Also, define a target team T of C for approaching. Clearly, the contributions of each object differs according to different team contexts, like performance varies of one player in different teams in NBA. Thus, while exchanging objects, a set of exchanging parameters $\Lambda\{\lambda_1, \lambda_2, ..., \lambda_m\}$ is defined for measuring the contribution of $O \in \mathbb{O}$ under current TC. Formally, our problem is:

Problem 1. *Ranking under Team Context*(RTC Problem) Rank the objects in $\mathbb{O}$ and determine a swap-in object which is top-ranked corresponding to a swap-out object in C. After performing the exchanging procedure, C can approach T to its best effort.

4 Solutions to RTC Problem

4.1 Modelling the Problem

Contributions of Objects Since team context C is formed by objects in $\mathbb{O}$, C can also be described by contributions of its components. There exist different ways calculating the contributions of components, which are based on how different contexts are organized. In this paper, we adopt the method which means the team's ability is the accumulation of all its components, since it is the most widely used way in real scenarios. Demonstrate in (1):

$$c_i = \sum_{j=1}^{m} o_{ji} \tag{1}$$

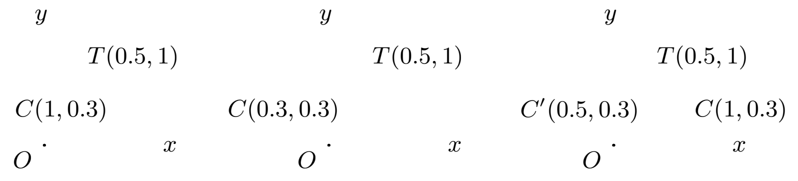
where c_i means value on ith dimension of C and o_{ji} means the value of jth object on its ith dimension.

Contributions of Attributes Although final ranking of the one team depends on values of all attributes, not all of them weight equally. For identifying the importance of each attribute, we adopt the Kendall's tau(τ) coefficients, which is a rank coefficient measuring association between two measured associations [5].

Through calculating the association between each dimension and final ranking of team context pairwisely, an coefficient will be obtained and will be regarded as weight parameter of corresponding dimension i, denote as w_i.

Truncated Distance Usually, we measure the difference between contexts(or objects) by weighted Euclid distance. However, positive distance yields to represent the overall conditions of a TC, especially the case shown in Fig.1(a).

The 2-D case in Fig.1(a) depicted that C exceeds T in dimension x while yields T in dimension y. If we only consider measuring the distance by weighted Euclid distance, C might be far away from T due to abstract advantage on dimension x, therefore, situation such as losing strength on x dimension when approaching T will occur, which contradicts the team's goal. In order to preserve advantage of C while approaching T, using truncated distance as a measurement is adopted as shown in Fig.1(c). As illustrated, if Case 1 happens, we only consider the distance between C' and T rather than C and T. Another case shown in Fig.1(b) is relatively simple for tackling since C lags T in both dimension x and y. So distance is typical weighted Euclid distance.



(a) Case 1 (b) Case 2 (c) Case 1 After Truncation

Fig. 1. Example of Truncated Distance

For a clear expression, we define a 0-1 truncating vector $\overrightarrow{TV}(tv_1, tv_2, ..., tv_d)$ to describe the truncated distance. Denote $\mathrm{diff_i}$ as the difference on dimension i and $\overrightarrow{TV}(i)$ as the ith component of $\overrightarrow{TV}$. Truncated difference $\widetilde{\mathrm{diff_i}}$ on ith dimension is:

$$\widetilde{\mathrm{diff_i}} = \mathrm{diff_i} \times \overrightarrow{TV}(i) = (t_i - c_i) \times \overrightarrow{TV}(i) \tag{2}$$

Notice that dimensions where $t_i - c_i < 0$ will be referred as strong dimensions of C, remaining ones will be referred as weak dimensions accordingly.

According to (2), the truncated weighted Euclid distance $\widetilde{\mathrm{dis}}$ is:

$$\widetilde{\mathrm{dis}} = \sqrt{\sum_{i=1}^{d} (w_i \widetilde{\mathrm{diff_i}})^2} \tag{3}$$

Distance measurements in this paper are all truncated distance. $\overrightarrow{TV}$ is referred as truncating vector henceforth. For better presentation, we denote $\widetilde{\mathrm{oDis}}$ as the truncated distance between two objects which is calculated using (3).

Exchange Procedure Define the exchange procedure as swapping $R(r_1, r_2, ..., r_d)$ in C with $P(p_1, p_2, .., p_d)$ in $\mathbb{O}$, thus new $\mathrm{diff_i}'$ is:

$$\mathrm{diff_i}' = t_i - (c_i - r_i + \frac{\lambda_r}{\lambda_p} p_i) \tag{4}$$

where $\lambda_r, \lambda_p \in \Lambda$ are exchange parameters defined in section 3.2.

Accordingly, truncated difference $\widetilde{\mathrm{diff_i}}'$ after exchanging procedure can be calculated by (2) with a new 0-1 truncating vector $\overrightarrow{TV_1}$ based on situation on each dimension. Hence, new $\widetilde{\mathrm{dis}}'$ after exchanging objects can be calculated with $\widetilde{\mathrm{diff_i}}'$ by applying (3).

4.2 RTC* Method

Before we propose the RTC* method, we define a virtual object as follows:

Definition 1. *(Virtual Object) Define a virtual object $V(v_1, v_2, ..v_d)$ which could make C has the same value of T on each weak dimension after exchanging with object R in C. Thus, value of virtual object on dimension i is:*

$$v_i = \frac{\mathrm{diff_i} + r_i}{\lambda_r} \times \overrightarrow{TV_2}(i) \tag{5}$$

where $\overrightarrow{TV_2}$ is new truncating vector for virtual object and r_i is the value of swap-out object R on dimension i.

Corollary 1. *Assume $\forall w_i \in W(w_1, w_2, ..., w_d), w_i > 0$, denote the truncated distance between objects as $\widetilde{\mathrm{oDis}}$. The nearest neighbours of virtual objects measured by $\widetilde{\mathrm{oDis}}$ is the top-ranked ones who can make C become closer to T.*

Proof. Suppose we can find a nearest neighbour P of V, $\lambda_p P \in \mathbb{O}$, truncated difference between V and P is represented using $\widetilde{\Delta}$ where $\widetilde{\Delta}(i)$ is the truncated value on dimension i. Thus, $\widetilde{\mathrm{diff}}'$ is:

$$\widetilde{\mathrm{diff_i}}' = (\frac{\mathrm{diff_i} + r_i}{\lambda_r} - p_i) \times \overrightarrow{TV_3}(i) \tag{6}$$

where $\overrightarrow{TV_3}$ is truncating vector and $\overrightarrow{TV_3}(i) = 0$ *iff* $\widetilde{\Delta(i)} < 0$, so $\widetilde{\mathrm{dis}}'$ can be represented as:

$$\widetilde{\mathrm{dis}}' = \sqrt{\sum_{i=1}^{d} (w_i \widetilde{\mathrm{diff_i}}')^2} \tag{7}$$

Notice that $\forall w_i \in W(w_1, w_2, ..., w_d), w_i > 0$, (7) also can be represented as:

$$\begin{aligned} \widetilde{\mathrm{dis}}' &= \sqrt{\sum_{i=1}^{d} (w_i((v_i - p_i)\overrightarrow{TV_3}(i))^2} \\ &= \sqrt{\sum_{i=1}^{d} (w_i(\widetilde{\Delta}(i) \times \overrightarrow{TV_3}(i))^2} \\ &= \widetilde{\mathrm{oDis}} \end{aligned} \tag{8}$$

□

So our problem of ranking objects from perspective of team context can be mapped into object space. Which is, by considering nearest neighbours of virtual objects under current team context, we can obtain top-ranked tuples.

We can index the truncated distance between $\forall O_i \in \mathbb{O}$ and the virtual object for convenience of searching:

As presented in Algorithm 1, we first calculate the virtual object based on current team context and index the $\widetilde{\mathrm{oDis}}$ between $O \in \mathbb{O}$ and virtual object in iDistance presented in [16] for processing the query.

It is easy to make the generalization that the query time is only related to the cardinality of our current context C, so RTC* will show high performance and good scalability on very large datasets.

```
Algorithm 1: RTC* Method
    Input: current context $C$,Target Context $T$,$\mathbb{O}$
    Output: $< R_i, P_i >$
foreach $R_i \in C$ do Calculate $V_i$ using (5);
foreach $P_i \in \mathbb{O}$ do Index $\widetilde{\text{oDis}}$ between $P_i$ and $V_i$;
find $< R_i, P_i >$ with Min($\widetilde{\text{oDis}}$);
return $< R_i, P_i >$;
```

5 Experiments

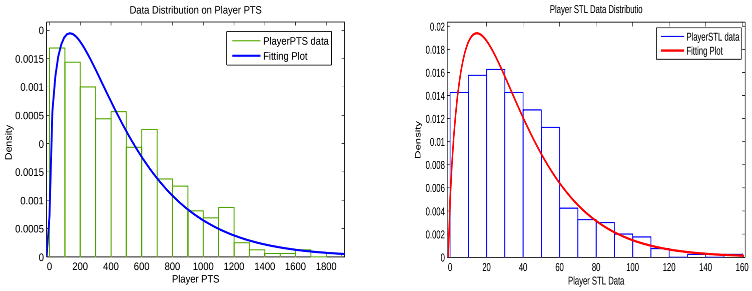
5.1 Experiment Setup

All the experiments were performed on machine with Intel Core(TM) i3 CPU and 4 GB RAM hosted on 32 bit Windows 7.

Datasets We perform our experiments on both real and synthetic data. Real dataset is obtained from [9] which consists total statistic data of NBA regular season from 2011 to 2012. Real dataset contains 400 players with 24 attributes in total and and 30 teams described by 20 dimensions. Size of player dataset is 39.5KB and team dataset is 20KB.

Attributes which can discriminate between season-long successful and unsuccessful basketball teams according to researches on basketball in [3, 10] are FG, 3P, 3PA, BLK, FT, STL, FTA, PTS, AST, DRB and TRB. We use this attribute set for our experiments as well.

Synthetic dataset are generated based on the features of real dataset with total 1.07×10^6 records and 69MB size. Feature of partial dimensions is illustrated in Fig.2. We make hypothesis H_0 that values of dimensions listed in 1 has negative binomial distribution and do distribution fitting accordingly.



(a) Player PTS

(b) Player STL

Fig. 2. Partial Review of Data Distribution

We test the hypothesis using Chi-square goodness-of-fit with parameters estimated in Table 1. H_0 is accepted at 95% significance level. Synthetic data are generated based on the fitted distribution.

Table 1. Estimated Parameters of Data Distribution

Dimension	r	p	Dimension	r	p	Dimension	r	p
FG	1.44	0.008	TRB	1.62	0.008	BLK	0.91	0.004
DRB	1.67	0.01	FT	1.07	0.013	STL	1.70	0.045
FTA	1.16	0.01	PTS	1.40	0.003	AST	0.93	0.0092

Because each dimension contributes differently in teams' final rankings, we adopt Kendall's τ, which is a method measuring the associations between attributes [5], to calculate weight of each dimension. Results are listed in Table 2.

Table 2. Weight for Each Dimension

Dimension	Weight	Dimension	Weight	Dimension	Weight	Dimension	Weight
DRB	0.35	FG	0.2695	3P	0.30	AST	0.24
3PA	0.20	FT	0.2576	TRB	0.1884	STL	0.38
FTA	0.27	PTS	0.4060	BLK	0.24		

5.2 Experiments Implementations

First, we select the target team on real dataset before exchanging players. Results are shown in Table 3 with the initial truncated distance between C and T.

Table 3. Target Context of Each Team

C	T	Distance	C	T	Distance	C	T	Distance
DEN	BOS	6.1096	PHI	BOS	29.5286	UTA	MEM	13.6334
ORL	BOS	51.2774	HOU	ATL	31.2126	DAL	ATL	28.4675
NYK	MEM	23.6467	PHO	BOS	18.3673	MIL	MEM	19.3955
POR	LAC	31.0965						

A Brute Force Method Brute force method is performed for each "mid-class"(teams ranked 11st∼20th based on game winning in season 2011∼2012)

Table 4. Selection For HOU

Roster	Candidate	New Distance
Luis Scola	Josh Smith	0
Patrick Patterson	LeBron James	0

team on real dataset as a baseline method. Take results of HOU listed in Table 4 as an example. There are two pairs of players could be found for making this team approach its target measured by truncated distance. Either pair can be selected to make chance for this team to enter into play-offs.

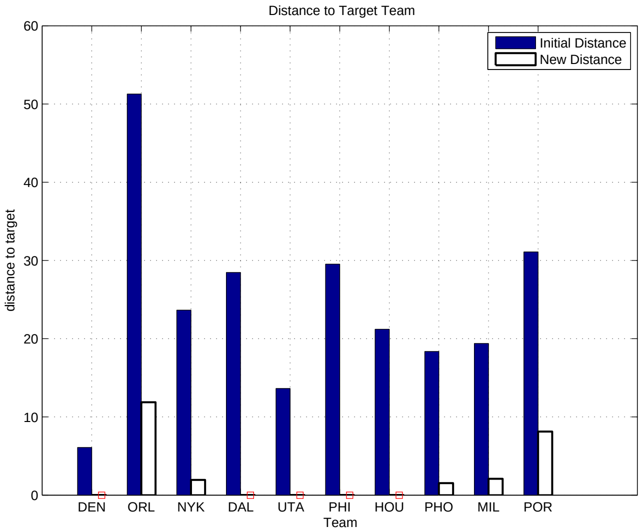


Fig. 3. Comparison of Team’s Distance to Target

Fig.3 depicts between each ”mid-class” team and its target before and after exchanging players. We can observe that many teams will have same value as its targets on their weak dimensions, which are illustrated in square.

RTC* Method We also test RTC* method on the real dataset. In estimating virtual players using (5), minutes played serves as the exchange parameter λ_r.

Also consider HOU(Houston Rockets) as an example. Values on dimension FG,3P,3PA,FT and FTA of virtual player and corresponding swap-in player listed in Table 5 for explanation:

As listed in Table 5, both selected players from player space are better than corresponding virtual players on those dimensions. According to definition of

Table 5. Overview of Virtual Player and Candidate

Name	Attributes					$\widetilde{\text{oDis}}$
	FG	3P	3PA	FT	FTA	
Josh Smith	0.22	0.01	0.05	0.09	0.14	0
Virtual Player of Josh	0.18	0.01	0.00	0.08	0.14	
LeBron James	0.27	0.02	0.06	0.17	0.22	0
Virtual Player of LeBron	0.11	0.01	0.00	0.03	0.10	

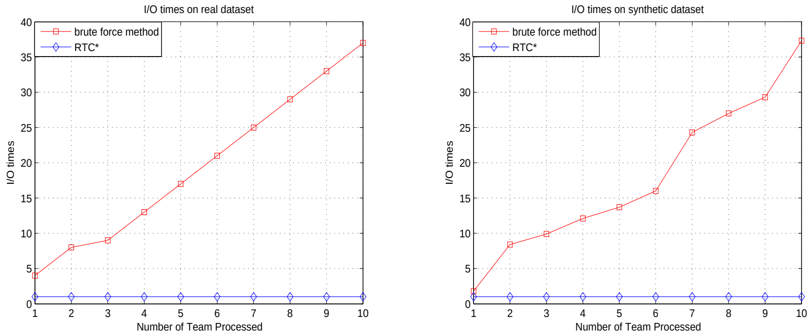
truncated distance, the $\widetilde{\text{oDis}}$ calculated between Josh Smith and its corresponding virtual player, or LeBron James and its virtual player listed, are 0. Therefore, those two pairs are selected as the result which is same as the ones selected using brute force listed in Table 4.

Notice that both results are the nearest neighbours of corresponding virtual player measured by $\widetilde{\text{oDis}}$, which further proves the rationale behind Corollary 1.

5.3 Result Analysis

In this section, we mainly focus on analysing the results of brute force method and RTC* on efficiency and scalability.

We fixed our block size as 100 records per block for real dataset and 10 per block for synthetic data.



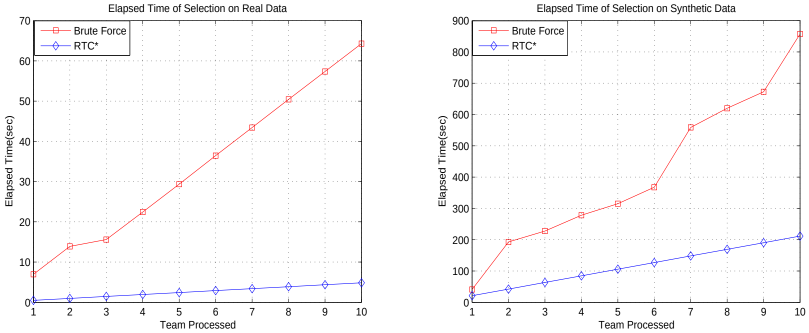
(a) I/O Test on real data

(b) I/O Test on synthetic data

Fig. 4. I/O Performance Of Different Method

Depicted in Fig.4, regardless of the data size and block size, I/O will be performed only once using RTC* as long as we had set up virtual player index, while brute force method varies depending on TC, which shows less robustness.

Like I/O testing, we tested time cost of selection method both on real and synthetic data, illustrated in Fig.5.



(a) Time Cost on Real Data

(b) Time Cost on Synthetic

Fig. 5. Time Cost Of Different Method

It is easy to generalize that the time cost of RTC* grows slowly with a constant rate while brute force increase very fast.

Differs from brute force method which highly depends on the value of team context, RTC* has good robustness and better performance regardless of the data value.

6 Conclusion and Future Work

In this paper, we introduce the problem of object selection under team context. This problem is quite practical in many scenarios of selecting objects to improve the team or organization's competence. We propose the brute force algorithm RTC for the problem. Furthermore, we propose an I/O efficient algorithm RTC* based on NN-indexing with a proof that its output is equivalent to RTC. Extensive experiments are conducted on both synthetic and real datasets to demonstrate the effectiveness and efficiency of our algorithms.

We would like to extend our work from two directions in our future work. First, due to the fact that the probabilistic database tuples are not uncommon, we plan to do probabilistic object selection. Second, query the database objects based on the teams' temporal contexts.

References

1. N. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
2. J. Hu, B. Cui, and H. T. Shen. Diagonal ordering: A new approach to high-dimensional knn processing. In *ADC*, pages 39–47, 2004.
3. S. J. Ibez, J. Sampaio, S. Feu, A. Lorenzo, M. A. Gmez, and E. Ortega. Basketball game-related statistics that discriminate between teams season-long success. *European Journal of Sport Science*, 8(6):369–372, 2008.

4. H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang. idistance: An adaptive b$^+$-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst.*, 30(2):364–397, 2005.
5. M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.
6. N. Kouiroukidis and G. Evangelidis. The effects of dimensionality curse in high dimensional knn search. In *Panhellenic Conference on Informatics*, pages 41–45, 2011.
7. X. Li and L. Feng. Context-aware group top-k query. In *ICDIM*, pages 149–154, 2012.
8. X. Li, L. Feng, and L. Zhou. Contextual ranking of database querying results: A statistical approach. In *EuroSSC*, pages 126–139, 2008.
9. S. R. LLC. Player season finder. `http://www.basketball-reference.com/play-index/psl_finder.cgi`, November 2013.
10. D. Oliver. *Basketball on paper: rules and tools for performance analysis.* Potomac Books, Inc., 2004.
11. M. A. Schuh, T. Wylie, J. M. Banda, and R. A. Angryk. A comprehensive study of idistance partitioning strategies for knn queries and high-dimensional data indexing. In *BNCOD*, pages 238–252, 2013.
12. K. Stefanidis, E. Pitoura, and P. Vassiliadis. Adding context to preferences. In *ICDE*, pages 846–855, 2007.
13. K. Stefanidis, N. Shabib, K. Nørvåg, and J. Krogstie. Contextual recommendations for groups. In *ER Workshops*, pages 89–97, 2012.
14. A. H. van Bunningen, L. Feng, and P. M. G. Apers. A context-aware preference model for database querying in an ambient intelligent environment. In *DEXA*, pages 33–43, 2006.
15. A. H. van Bunningen, M. M. Fokkinga, P. M. G. Apers, and L. Feng. Ranking query results using context-aware preferences. In *ICDE Workshops*, pages 269–276, 2007.
16. C. Yu, B. Cui, S. Wang, and J. Su. Efficient index-based knn join processing for high-dimensional data. *Information and Software Technology*, 49(4):332–344, 2007.
17. R. Zhong, G. Li, K.-L. Tan, and L. Zhou. G-tree: an efficient index for knn search on road networks. In *CIKM*, pages 39–48, 2013.