Volume 2, Issue 8, August 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Validated Nearest Neighbour Search to Minimize The Transmission Overhead

[1]Malatesh S.H, [2]V. Gayathri, [3]Subashini.S [4]Mahesh T.R, [5]Vinayababu M
[1]Department of CSE , MSEC, Bangalore, India
[2]Department of CS, GSC, Bangalore, India
[3]Research Scholar , Singhania University, Rajasthan, India
[4]Department of CS, TJIT, Bangalore, India
[5]Department of MCA, BIGCE, Solapur, India

Abstract: Authenticated multistep NN search required for applications that require a proof of result correctness. Clients issue similarity queries to a provider. The latter returns the result set and additional verification information, based on which the client establishes that the result is indeed correct, i.e., it contains exactly the records of DB that satisfy the query conditions, and that these records indeed originate from their legitimate data source. A similar situation occurs for data replication, i.e., when a data owner stores DB at several servers. Clients issue their queries to the closest server, but they wish to be assured that the result is the same as if the queries were sent to the original source of DB. Notarization services have been proposed to safeguard against tampering in document databases.

KEYWORDS: KNN(K-NEAREST NEIGHBOUR), DB(DATA BASE), DATABASE MANAGEMENT SYSTEM, DST,, MIN-MAX POLYNOMIAL,

I. INTRODUCTION

Authenticated query processing ensures the client that the received result complies with the validated DB. Assuming that that entire DB resides at a single server. An adaptation of a multistep algorithm that is optimal in terms of DST computations. AMN requires transmissions of false hits, i.e., records that are not in the result, but are nevertheless necessary for its verification. In addition to the network overhead, false hits impose a significant burden to the client, which has to verify them. C-AMN alleviates this problem through an elaborate scheme that reduces the size of false hits. The database is horizontally partitioned over several servers.

A kNN query returns the k closest points to Q. NN and kNN queries are common in similarity retrieval. Specifically[1], since similarity between records is inversely proportional to their distance, a kNN query returns the k most similar records to Q. The multistep framework has been proposed for NN and similarity retrieval in domains that entail high dimensional data. We focus on authenticated multistep NN search for applications that require a proof of result correctness. Clients issue similarity queries to a provider. The latter returns the result set and additional verification information, based on which the client establishes that the result is indeed correct, i.e., it contains exactly the records of DB that satisfy the query conditions, and that these records indeed originate from their legitimate data source. A similar situation occurs for data replication, i.e., when a data owner stores DB at several servers.

II. RELATED WORK

A. MAKING THE NEAREST NEIGHBOR MEANINGFUL

The nearest-neighbor problem arises in clustering and other applications. It requires us to define a function to measure differences among items in a data set, and then to compute the closest items to a query point with respect to this measure[1]. Recent work suggests that the conventional Euclidean measure does not adequately model high dimensional data. We present a new, data-driven difference measure for categorical data for which the difference between two data points is based on the frequency of the categories or combinations of categories that they have in common. This measure addresses the main flaw of the Euclidean distance measure—namely, that it treats each dimension independently.

We then provide both brute-force algorithms and an efficient, but approximate, probabilistic algorithm to compute the nearest neighbors of a query point with respect to this measure. Finally, we illustrate a practical application of our approach in a recommendation engine built for the Tower Records online video and DVD catalog.

B.INDEXING SPATIO-TEMPORAL TRAJECTORIES WITH CHEBYSHEV POLYNOMIALS

We attempt to approximate and index a d dimensional ($d \geq 1$) spatio-temporal trajectory with a low order continuous polynomial. There are many possible ways to choose the polynomial, including (continuous) Fourier transforms, splines, non-linear regression, etc.

Some of these possibilities have indeed been studied before. We hypothesize that one of the best possibilities is the polynomial that minimizes the maximum deviation from the true value, which is called the *minimax* polynomial[2]. Minimax

approximation is particularly meaningful for indexing because in a branch-and-bound search (i.e., for finding nearest neighbours) the smaller the maximum deviation, the more pruning opportunities there exist. However, in general, among all the polynomials of the same degree, the optimal minimax polynomial is very hard to compute. However, it has been shown that the Chebyshev approximation is almost identical to the optimal minimax polynomial, and is easy to compute. We explore how to use the Chebyshev polynomials as a basis for approximating and indexing *d*-dimensional trajectories.

The key analytic result of this paper is the Lower Bounding Lemma. That is, we show that the Euclidean distance between two *d*-dimensional trajectories is lower bounded by the weighted Euclidean distance between the two vectors of Chebyshev coefficients. This lemma is not trivial to show, and it ensures that indexing with Chebyshev coefficients admits no false negatives. To complement the analytic result, we conducted comprehensive experimental evaluation with real and generated 1-dimensional to 4-dimensional data sets. We compared the proposed scheme with the Adaptive Piecewise Constant Approximation (APCA) scheme. Our preliminary results indicate that in all situations we tested, Chebyshev indexing dominates APCA in pruning power, I/O and CPU costs.

C. OPTIMAL MULTI-STEP K-NEAREST NEIGHBOR SEARCH

For an increasing number of modern database applications, efficient support of similarity search becomes an important task. Along with the complexity of the objects such as images, molecules and mechanical parts, also the complexity of the similarity models increases more and more.

Whereas algorithms that are directly based on indexes work well for simple medium-dimensional similarity distance functions, they do not meet the efficiency requirements of complex high-dimensional and adaptable distance functions[1]. The use of a multi-step query processing strategy is recommended in these cases, and our investigations substantiate that the number of candidates which are produced in the filter step and exactly evaluated in the refinement step is a fundamental efficiency parameter.

After revealing the strong performance shortcomings of the state-of-the-art algorithm for k-nearest neighboumr search, we present a novel multi-step algorithm which is guaranteed to produce the minimum number of candidates. Experimental evaluations demonstrate the significant performance gain over the previous solution, and we observed average improvement factors of up to 120 for the number of candidates and up to 48 for the total runtime.

D. TAMPER DETECTION IN AUDIT LOGS

Audit logs are considered good practice for business systems, and are required by federal regulations for secure systems, drug approval data, medical information disclosure, financial records, and electronic voting. Given the central role of audit logs, it is critical that they are correct and inalterable. It is not sufficient to say, “our data is correct, because we store all interactions in a separate audit log." The integrity of the audit log itself must also be guaranteed. [4]

A database management system (DBMS), based on cryptographically strong one-way hash functions, which prevent an intruder, including an auditor or an employee or even an unknown bug within the DBMS itself, from silently corrupting the audit log. We propose that the DBMS store additional information in the database to enable a separate audit log validator to examine the database along with this extra information and state conclusively whether the audit log has been compromised. We show with an implementation on a high-performance storage engine that the overhead for auditing is low and that the validator can efficiently and correctly determine if the audit log has been compromised.

E. SPATIAL OUTSOURCING FOR LOCATION-BASED SERVICES

The embedding of positioning capabilities in mobile devices and the emergence of location-based applications have created novel opportunities for utilizing several types of multidimensional data through spatial outsourcing. In this setting, a data owner (DO) delegates its data management tasks to a location-based service (LBS) that processes queries originating from several clients/ subscribers[2].

The LBS is not the real owner of the data, it must prove (to each client) the correctness of query output using an authenticated structure signed by the DO. Currently there is very narrow selection of multi-dimensional authenticated structures, among which the VR-tree is the best choice. Our first contribution is the MR-tree, a novel index suitable for spatial outsourcing. We show, analytically and experimentally, that the MR-tree outperforms the VR-tree, usually by orders of magnitude, on all performance metrics, including construction cost, index size, query and verification overhead. Motivated by the fact that successive queries by the same mobile client exhibit locality, we also propose a synchronized caching technique that utilizes the results of previous queries to reduce the size of the additional information sent to the client for verification purposes.

III. .NEAREST NEIGHBOUR SEARCH

A. SYSTEM ARCHITECTURE

Authenticated multistep NN search for applications that require a proof of result correctness. Clients issue similarity queries to a provider[7].

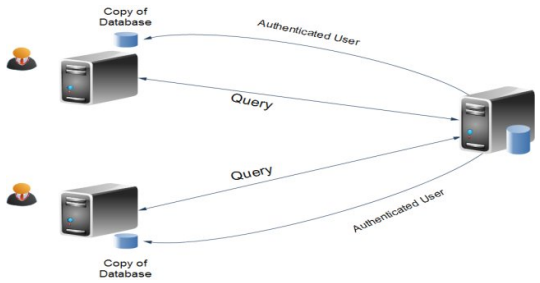

Figure 1 System architecture

B. AUTHENTICATION

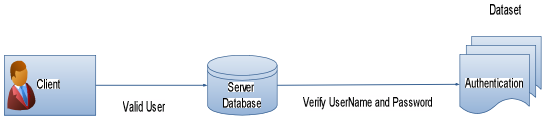

Figure 2 Authentication

The latter returns the result set and additional verification information, based on which the client establishes that the result is indeed correct, i.e., it contains exactly the records of Database that satisfy the query conditions, and that these records indeed originate from their legitimate data source. Clients issue their queries to the closest (in terms of network latency) server, but they wish to be assured that the result is the same as if the queries were sent to the original source of Database[1]. Authenticated query processing ensures the client that the received result complies with the validated Database.

C. IDENTIFYING THE FALSE HITS MODULE

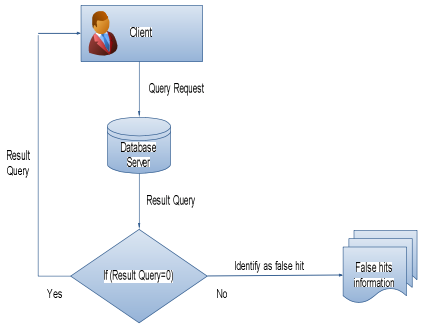

Figure 3 False hit module

Records that are not in the Database, but are nevertheless necessary for its verification. In addition to the network overhead, false hits impose a significant burden to the client, which has to verify them. It initializes a false hit set False Hit (FH) = 0, and a result set Record set (RS) ={P1 . . . . Pk}, and computes DST_{max}, i.e., the DST of the current k^{th} NN P_k. The goal of AMN is to return to the corresponding client the kNNs of Q, in a verifiable manner.

D. NEAREST NEIGHBOR SEARCH

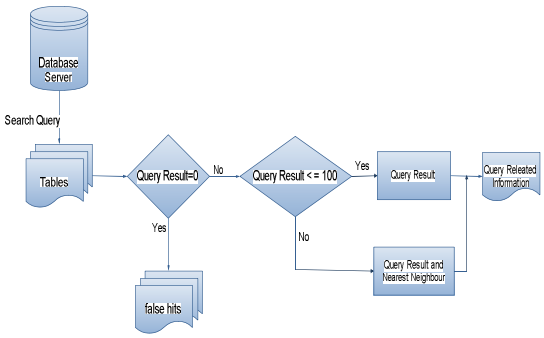

Figure 4 NN

The server receives and processes queries from client[8]s. Each query returns a result set RS_DB that satisfies certain predicates. Moreover, the client must be able to establish that Record Set (RS) is correct, i.e., that it contains all records of Database that satisfy the query conditions, and that these records have not been modified by the server or another entity[3]. There may be numerous false hits in FH information, each containing hundreds or thousands of values. Next, we communication-efficient AMN (C-AMN), which decreases the size of the false hits, significantly reducing the transmission and verification cost without compromising the security of AMN. The main concepts of C-AMN algorithm, for false hit reduction.

E. SEND THE RESULT QUERY TO USER

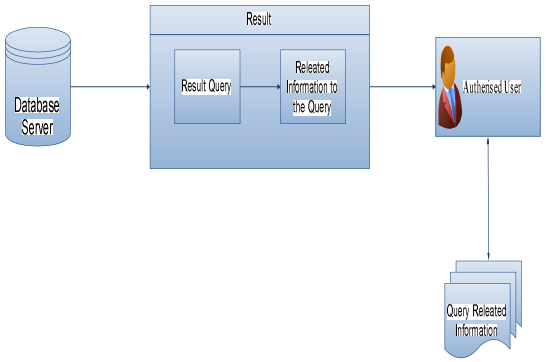

Figure 5 Result query module

The database is horizontally partitioned over M servers using ID-AMN Algorithm. ID-AMN transmits Query (Q) to the server, and each server performs the necessary computations to obtain the k local NNs[5]. ID-AMN transmits Q only to candidate server. In the best case, all results may be found in a server, and the rest are eliminated using the dst bound. The retrieval of a kNN set involves a number of computations linear to (k+|FH|). In ID-AMN a server has to perform a number of computations that is proportional to its contribution $k_i <= k$ in the result set.

IV. AUTHENTICATED SEARCH

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development[6].

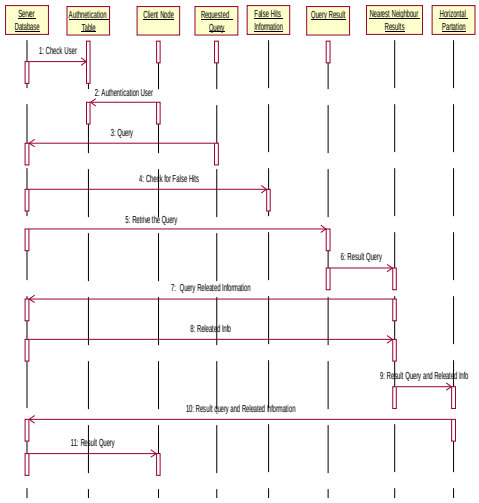

Figure. 6 Sequence diagram

V. CONCLUSION

Addressing authenticated similarity retrieval from such sources using the multistep kNN framework. We developed C-AMN, a technique that addresses the communication specific aspects of NN, and minimizes the transmission overhead and verification effort of the clients. We propose ID-AMN, which retrieves distance information from distributed servers, eliminating those that cannot contribute results.

In future we can distribute the false hit information to the client node so that we can avoid the false hits on the client side itself. So that we can save the time and the network traffic between the Database and the Client. when a data owner stores DB at several servers. Clients issue their queries to the closest server, but they wish to be assured that the result is the same as if the queries were sent to the original source of DB. Notarization services have been proposed to safeguard against tampering in document databases.

REFERENCES

[1] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, *"When is 'Nearest Neighbor' Meaningful?,"* Proc. Int'l Conf. Database Theory (ICDT '99), 1999.

[2] Y. Cai and R. Ng, *"Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials,"* Proc. ACM SIGMOD, 2004.

[3] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, *"Fast Nearest Neighbor Search in Medical Image Databases,"* Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '96), 1996

[4] A. Kundu and E. Bertino, *"Structural Signatures for Tree Data Structures,"* Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '08), 2008.

[5] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, *"Dynamic Authenticated Index Structures for Outsourced Databases,"* Proc. ACM SIGMOD, 2006

[6] C. Martel, G. Nuckolls, P. Devanbu, M. Gertz, A. Kwong, and S. Stubblebine, *"A General Model for Authenticated Data Structures,"* Algorithmica, vol. 39, no. 1, pp. 21-41, 2004.

[7] R. Merkle, *"A Certified Digital Signature,"* Proc. CRYPTO, 1989.

[8] H. Pang and K. Mouratidis, *"Authenticating the Query Results of Text Search Engines,"* Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '08), 2008.

