International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

ONLINE PRODUCT QUANTIZATION

[1]K.SRI LAKSHMI APURUPA, [2]K.JASWANTH
[1]M.TECH DEPT OF CSE, KAKINADA INSTITUTE OF TECHNOLOGICAL SCIENCES, RAMACHANDRAPURAM, ANDHRAPRADESH, INDIA, 533255
[2]ASSISTANT PROFESSOR, KAKINADA INSTITUTE OF TECHNOLOGICAL SCIENCES, RAMACHANDRAPURAM, ANDHRAPRADESH, INDIA, 533255

ABSTARCT

Approximate nearest neighbor (ANN) search has achieved great success in many tasks. However, existing popular methods for ANN search, such as hashing and quantization methods, are designed for static databases only. They cannot handle well the database with data distribution evolving dynamically, due to the high computational effort for retraining the model based on the new database. In this paper, we address the problem by developing an online product quantization (online PQ) model and incrementally updating the quantization codebook that accommodates to the incoming streaming data. Moreover, to further alleviate the issue of large scale computation for the online PQ update, we design two budget constraints for the model to update partial PQ codebook instead of all. We derive a loss bound which guarantees the performance of our online PQ model. Furthermore, we develop an online PQ model over a sliding window with both data insertion and deletion supported, to reflect the real-time behaviour of the data. The experiments demonstrate that our online PQ model is both time-efficient and effective for ANN search in dynamic large scale databases compared with baseline methods and the idea of partial PQ codebook update further reduces the update cost.

I. INTRODUCTION

APPROXIMATE nearest neighbour (ANN) search in a static database has achieved great success in supporting many tasks, such as information retrieval, classification and object detection. However, due to the massive amount of data generation at an unprecedented rate daily in the era of big data, databases are dynamically growing with data distribution evolving over time, and existing ANN search methods would achieve unsatisfactory performance without new data incorporated in their models. In addition, it is impractical for these methods to retrain the model from scratch for the continuously changing database due to the large scale computational time and memory. Therefore, it is increasingly important to handle ANN search in a dynamic database environment.ANN search in a dynamic database has a widespread applications in the real world. For example, a large number of news articles are generated and updated on hourly/daily basis, so a news searching system [1] requires to support news topic tracking and retrieval in a frequently changing news database. For object detection in video surveillance [2],video data is continuously recorded, so that the

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

distances between/among similar or dissimilar objects are continuously changing. For image retrieval in dynamic databases [3], relevant images are retrieved from a constantly changing image collection, and the retrieved images could therefore be different over time given the same image query. In such an environment, real-time query needs to be answered based on all the data collected to the database so far. In recent years, there has been an increasing concern over the computational cost and memory requirement dealing with continuously growing large scale databases, and therefore there are many online learning algorithm works [4], [5],[6] proposed to update the model each time streaming data coming in. Therefore, we consider the following problem. Given a dynamic database environment, develop an online learning model accommodating the new streaming data with low computational cost for ANN search.

Recently, several studies on online hashing [7], [8], [9],[10], [11], [12], [13] show that hashing based ANN approaches can be adapted to the dynamic database environment by updating hash functions accommodating new streaming data and then updating the hash codes of the exiting stored data via the new hash functions. Searching is performed in the Hamming space which is efficient and has low computational cost. However, an important problem that these works have not addressed is the computation of the hash code maintenance. To handle the streaming fashion of the data, the hash functions are required to be frequently updated, which will result in constant hash code recomputation of all the existing data in the reference database. This will inevitably incur an increasing amount of update time as the data volume increases. In addition, these online hashing approaches require the system to keep the old data so that the new hash code of the old data can be updated each time, leading to inefficiency in memory and computational load. Therefore, computational complexity and storage cost are still our major concerns in developing an online indexing model.

Product quantization (PQ) [14] is an effective and successful alternative solution for ANN search. PQ partitions the original space into a Cartesian product of low dimensional subspaces and quantizes each subspace into a number of sub-codewords. In this way, PQ is able to produce a large number of code words with low storage cost and perform ANN search with inexpensive computation. Moreover, it preserves the quantization error and can achieve satisfactory recall performance. Most importantly, unlike hashing-based methods representing each data instance by a hash code, which depends on a set of hash functions, quantization based methods represent each data instance by an index, which associates with a codeword that is in the same vector space with the data instance. However, PQ is a batch mode method which is not designed for the problem of accommodating streaming data in the model. Therefore, to address the problem of handling streaming data for ANN search and

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

tackle the challenge of hash code recomputation, we develop an online PQ approach, which updates the code words by streaming data without the need to update the indices of the existing data in the reference database, to further alleviate the issue of large scale update computational costs

II. EXISTING SYSTEM

Existing hashing methods are grouped in data-independent hashing and data-dependent hashing. One of the most representative work for data-independent hashing is Locality Sensitive Hashing (LSH) [20], where its hashing functions are randomly generated. LSH has the theoretical performance guarantee that similar data instances will be mapped to similar hash codes with a certain probability. Since data-independent hashing methods are independent from the input data, they can be easily adopted in an online fashion.

Data-dependent hashing, on the other hand, learns the hash functions from the given data, which can achieve better performance than data independent hashing methods. Its representative works are Spectral Hashing (SH) [19], which uses spectral method to encode similarity graph of the input into hash functions, IsoH [18] which finds a rotation matrix for equal variance in the projected dimensions and ITQ [17] which learns an orthogonal rotation matrix for minimizing the quantization error of data items to their hash codes.

To handle nearest neighbor search in a dynamic database, online hashing methods have attracted a great attention in recent years. They allow their models to accommodate to the new data coming sequentially, without retraining all stored data points. Specifically, Online Hashing, AdaptHash and Online Supervised Hashing are online supervised hashing methods, requiring label information, which might not be commonly available in many real-world applications. Stream Spectral Binary Coding (SSBC) [9] and Online Sketching Hashing (OSH) are the only two existing online unsupervised hashing methods which do not require labels, where both of them are matrix sketch-based methods to learn to represent the data seen so far by a small sketch.

However, all the online hashing methods suffer from the existing data storage and the high computational cost of hash code maintenance on the existing data. Each time new data comes, they update their hash functions accommodating to the new data and then update the hash codes of all stored data according to the new hash functions, which could be very time-consuming for a large scale database.

Disadvantages

There is no Online Hashing Methods for Online Product Quantization. There is no ANN search instead kNN Search Methods.

IV. PROPOSED SYSTEM

In the Proposed system, the system implemented Product quantization (PQ) which is an effective and successful alternative solution for ANN search. PQ partitions the original space into a Cartesian product of low dimensional subspaces and

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

quantizes each subspace into a number of sub-code words. In this way, PQ is able to produce a large number of code words with low storage cost and perform ANN search with inexpensive computation. Moreover, it preserves the quantization error and can achieve satisfactory recall performance. Most importantly, unlike hashing-based methods representing each data instance by a hash code, which depends on a set of hash functions, quantization based methods represent each data instance by an index, which associates with a codeword that is in the same vector space with the data instance.

However, PQ is a batch mode method which is not designed for the problem of accommodating streaming data in the model. Therefore, to address the problem of handling streaming data for ANN search and tackle the challenge of hash code recomputation, the system develops an online PQ approach, which updates the code words by streaming data without the need to update the indices of the existing data in the reference database, to further alleviate the issue of large scale update computational cost.

Advantages

The Proposed product quantizer in PQ, on the other hand, updates the code Zords in the codebook, but it does not change the index of the updated code words of each data point in the reference database. To handle nearest neighbor search in a dynamic database, online hashing methods have attracted a great attention in the proposed system.

V. IMPLEMENTATION

5. 1 Admin Server

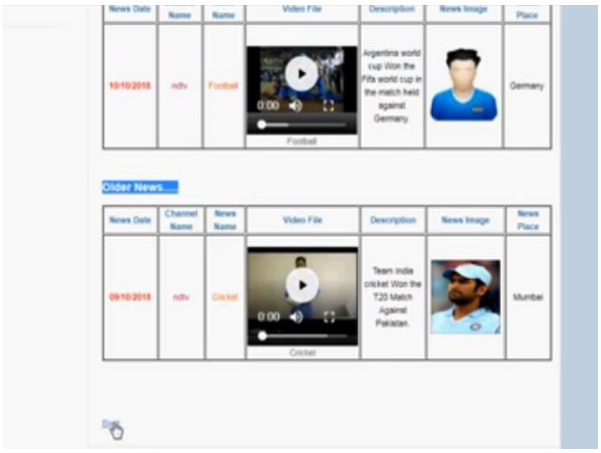
In this module, the Admin has to login by using valid user name and password. After login successful he can perform some operations such as List all users and authorize, Register with News channel name and login, Add News Categories, Set news quantization date, Select category and add news, List all news post and give option to update and delete, List all news post by quantization, List All News Posts by clusters based on news cat, List All Users News transactions by keyword, View online product quantization by chart, View all news post rank in chart.

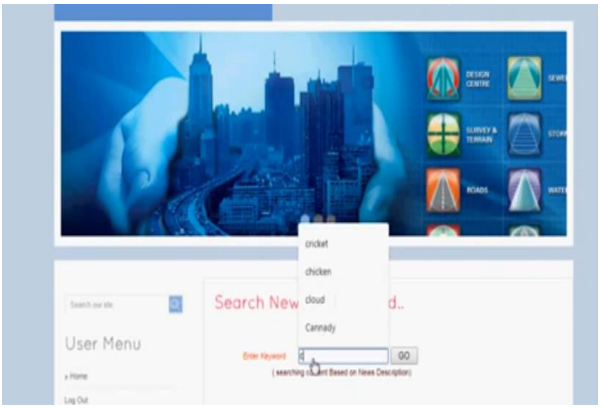
5.2 User

In this module, there are n numbers of users are present. User should register before performing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user can perform some operations like View your profile, Search news by content keyword, select hash code to show all news titles, Show all your search transactions based on keyword and hash code.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

VI SCREEN SHOTS





VI CONCLUSION

In this paper, we have presented our online PQ method to accommodate streaming data. In addition, we employ two budget constraints to facilitate partial codebook update to further alleviate the update time cost. A relative loss bound has been derived to guarantee the performance of our model. In addition, we propose an online PQ over sliding window approach, to emphasize on the real-time data. Experimental results show that our method is significantly faster in accommodating the streaming data, out performs the competing online hashing methods and unsupervised batch mode hashing method in terms of search accuracy and update time cost, and attains comparable search quality with batch mode PQ.

VII. REFERENCES

[1] A. Moffat, J. Zobel, and N. Sharman, "Text compression for Dynamic document databases," TKDE, vol. 9, no. 2, pp. 302–313,1997.

[2] R. Popovici, A. Weiler, and M. Grossniklaus, "On-line clustering for real-time topic detection in social media streaming data," in SNOW 2014 Data Challenge, 2014, pp. 57–63.

[3] A. Dong and B. Bhanu, "Concept learning and transplantation for dynamic image databases," in ICME, 2003, pp. 765–768.

[4] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer,"Online passive-aggressive algorithms," JMLR, vol. 7, pp. 551–585,2006.

[5] C. Li, W. Dong, Q. Liu, and X. Zhang, "Mores: online incremental multiple-output regression for data streams," CoRR, vol.abs/1412.5732, 2014.

[6] L. Zhang, T. Yang, R. Jin, Y. Xiao, and Z. Zhou, "Online stochasticlinear

optimization under one-bit feedback," in ICML, 2016, pp.392–401.
[7] L. Huang, Q. Yang, and W. Zheng, "Online hashing," in IJCAI,2013, pp. 1422–1428.
[8] "Online hashing," NNLS, 2017.
[9] M. Ghashami and A. Abdullah, "Binary coding in stream," CoRR,vol. abs/1503.06271, 2015.
[10] C. Leng, J. Wu, J. Cheng, X. Bai, and H. Lu, "Online sketchinghashing," in CVPR, 2015, pp. 2503–2511.
[11] F. Cakir and S. Sclaroff, "Adaptive hashing for fast similarity search," in ICCV, 2015, pp. 1044–1052.
[12] Q. Yang, L. Huang,W. Zheng, and Y. Ling, "Smart hashing update for fast response," in IJCAI, 2013, pp. 1855–1861.
[13] F. Cakir, S. A. Bargal, and S. Sclaroff, "Online supervised hashing,"CVIU, 2016.
[14] H. J´egou, M. Douze, and C. Schmid, "Product quantization fornearest neighbor search," PAMI, vol. 33, no. 1, pp. 117–128, 2011.
[15] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compactbinary codes," in ICML, 2011, pp. 353–360.
[16] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, "Supervised hashingwith kernels," in CVPR, 2012, pp. 2074–2081.
[17] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in CVPR, 2011, pp. 817–824.
[18] W. Kong and W. Li, "Isotropic hashing," in NIPS, 2012, pp. 1655–1663.
[19] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in NIPS,2008, pp. 1753–1760.
[20] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in VLDB, 1999, pp. 518–529.
[21] A. Babenko and V. S. Lempitsky, "Additive quantization for extreme vector compression," in CVPR, 2014, pp. 931–938.