ONLINE PRODUCT QUANTIZATION

A. Durga Devi Madam [1], K. Mahesh [2],

[1]**Assistant professor , MCA DEPT,** Dantuluri Narayana Raju College**, Bhimavaram, Andharapradesh**
Email:- adurgadevi760@gmail.com
[2]**PG Student of MCA, Dantuluri** Narayana Raju College**, Bhimavaram, Andharapradesh**
Email:- maheshkopanathi01@gmail.com

ABSTRACT

Approximate nearest neighbour (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.

1 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 requires to support news topic tracking and retrieval in a frequently changing news database.

2 RELEATED WORK

AUTHORS: Donna Xu, Ivor W. Tsang, and Ying Zhang

Product quantization 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 codewords 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.

3 implementation study

Existing System:

Approximate nearest neighbor (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.

Disadvantages

•It cannot handle well the database with data distribution evolving dynamically.

Proposed System & alogirtham

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, outperforms 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.

Advantages

•Handle the database with data distribution evolving dynamically.

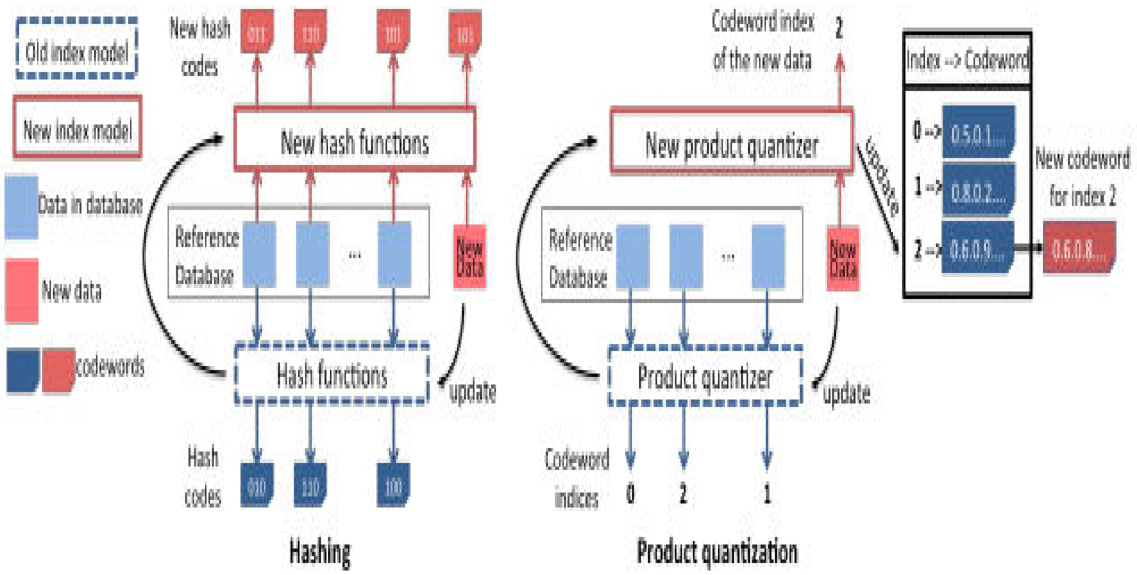

Fig1: SYSTEM ARCHITECTURE

4. IMPLEMENTATION

4.1 MODULES

1.User
2.Admin

MODULES DESCRIPTION

User

In this application the user should register with the application. then only the user can able to login into the homepage.After he gets the access into the home page, he can do the following activities such as view profile, search code word, search products, all purchased products.These are the operation will going to do by the user.

Admin

Admin also one of the module in the project, and the admin can perform the main role in this project.Here the admin can directly login into the application, here the admin can add the category, add product, view all added products, view product count, view Product quantization ranking, view graph. These are the operations will done by the admin.

5 RESULTS AND DISCUSSION

To run project double click on file to get below screens

HOME PAGE

Figure 5.1: Home Page

USER REGISTER PAGE

Figure 5.2: User Register Page

USER LOGIN PAGE

Figure 5.3: User Login Page

ADMIN LOGIN PAGE

Figure 5.4: Admin Login Page

VIEW PRODUCT PAGE

Figure 5.5: View Product Page

PRODUCT RANKING PAGE

Figure 5.6: Product Ranking Page

6.CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In Online Product Quantization 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, outperforms the competing online and batch hashing methods in terms of search accuracy and update time cost, and attains comparable search quality with batch mode PQ.

6.2 FUTURE SCOPE

In our future work, we will extend the online update for other MCQ methods, leveraging the advantage of them in a dynamic database environment to enhance the search performance. Each of them has challenges to be effectively extended to handle streaming data. For example, CQ and SQ require the old data for the codewords update at each iteration due to the constant inter-dictionary-elementproduct in the model constraint. AQ requires a high computational encoding procedure, which will dominate the update process in an online fashion. TQ needs to consider the tree graph update together with the codebook and the indices of the stored data. Extensions to these methods can be developed to address the challenges for online update. In addition, online PQ model can be extended to handle other learning problems such as multi output learning. Moreover, the theoretical bound for the online model will be further investigated.

7.REFERANCES

[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] L. Zhang, T. Yang, R. Jin, Y. Xiao, and Z. Zhou, “Online stochastic linear optimization under one-bit feedback,” in ICML, 2016, pp. 392–401.

[6] L. Huang, Q. Yang, and W. Zheng, “Online hashing,” in IJCAI, 2013, pp. 1422–1428.

[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 sketching hashing,” in CVPR, 2015, pp. 2503–2511.

[11] F. Cakir and S. Sclaroff, “Adaptive hashing for fast similarity search,” in ICCV, 2015, pp. 1044–1052.