E-COMMERCE PRODUCT MANAGEMENT SYSTEM FOR QUANTIFICATION AND STOCK

Dr.M.V.Surya Narayana[1] , T.Sony Priya[2]

[1] Assoc. Professor, Department of Business Management, Aurora's PG College (MBA), Uppal, Hyderabad
Email: mvsuryanarayana09@gmail.com

[2] Assistant Professor, Department of Business Management, Aurora's PG College (MBA), Uppal, Hyderabad
Email: tsonypriya1841990@gmail.com

ABSTRACT

In numerous jobs, the approximate nearest neighbor (ANN) search has shown remarkable success. ANN search techniques that are currently widely used, including quantization and hashing, are only intended for static databases. Because retraining the model on the new database requires a significant computing effort, they are unable to manage databases where the distribution of data is changing dynamically. Our approach to solving the issue in this research is to create an online product quantization (online PQ) model and update the quantization codebook gradually to account for the incoming streaming data. Furthermore, we create two budget restrictions for the model to update a portion of the PQ codebook rather than the entire one in order to further mitigate the problem of large-scale computing for the live PQ update. We construct a loss bound that ensures our online PQ model will function as expected. In addition, we create an online PQ model over a sliding window that supports the insertion and deletion of data in real-time to better represent the data's behavior. In comparison with baseline approaches, the studies show that our online PQ model for ANN search in dynamic large-scale databases is both efficient and effective in terms of time, and the concept of partial PQ codebook updating further minimizes the update cost.

Keywords: Business to customer (approximate nearest neighbor; ANN), shopping cart, payment system, product quantization

1 INTRODUCTION

In a static database, approximate nearest neighbor (ANN) search has shown remarkable success in assisting with a variety of tasks, including item identification, categorization, and information retrieval. However, databases are dynamically growing with data distribution evolving over time due to the massive amount of data generated at an unprecedented rate every day in the big data era, and existing ANN search methods would perform unsatisfactorily without new data incorporated into their models. Moreover, the significant computational time and memory overhead of these methods make it impossible to retrain the model for the constantly changing database. Consequently, managing ANN search in a dynamic database environment is becoming more and more crucial. In the real world, ANN search in a dynamic database is widely used. For example, a news searching system needs to handle news topic monitoring and retrieval in a continuously changing news database since a high number of news articles are published and updated on an hourly or daily basis.

2 RELEATED WORK

Ying Zhang, Donna Xu, and Ivor W. Tsang are the authors.Product quantization is a successful and efficient substitute for ANN search. PQ quantizes each low dimensional subspace into a number of sub-codewords after dividing the original space into a Cartesian product of those subspaces. In this sense, PQ can generate a large number of codewords at low storage costs and carry out ANN search at low computational costs. Additionally, it may accomplish adequate recall performance while maintaining the quantization error. Most crucially, quantization-based approaches represent each data instance by an index that associates with a codeword that is in the same vector space as the data object, in contrast to hashing-based methods that represent each data instance by a hash code, which depends on a set of hash functions.

3 IMPLEMENTATION STUDY

Current System: In a static database, approximate nearest neighbor (ANN) search has shown remarkable effectiveness in a variety of tasks, including item detection, classification, and information retrieval. However, databases are dynamically growing with data distribution evolving over time due to the massive amount of data generated at an unprecedented rate every day in the big data era, and existing ANN search methods would perform unsatisfactorily without new data incorporated into their models. Moreover, the significant computational time and memory overhead of these methods make it impossible to retrain the model for the constantly changing database. Consequently, managing ANN search in a dynamic database environment is becoming more and more crucial.
Cons: • It is unable to manage a database where the distribution of data is changing on the fly.

System Proposal & Alogirtham
In order to support streaming data, we have demonstrated our online PQ technique. To further reduce the update time cost, we also use two budget limits to enable partial codebook updates. We have generated a relative loss bound to ensure our model's performance. Furthermore, we suggest an online PQ using a sliding window method to highlight real-time data. According to experimental results, our method achieves comparable search quality with batch mode PQ, is significantly faster at accommodating streaming data, and performs better than competing online hashing methods and unsupervised batch mode hashing method in terms of search accuracy and update time cost.
Benefits
• Manage the database using a continuously developing data distribution.

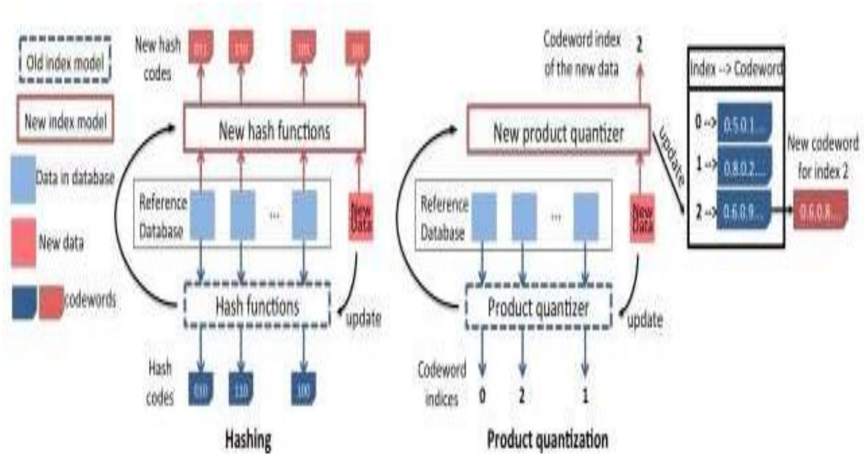

Fig1: SYSTEM ARCHITECTURE

4. IMPLEMENTATION

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

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.

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.