International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 24th Feb 2018. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-02

Title: **ANALYSING ONLINE PRODUCT BY USING PQ MODEL**

Volume 08, Issue 02, Pages: 124–126.

Paper Authors

MS.G.PRASANTHI, K.MAHESH

Vignan's Lara Institute of Technology & Science

International Journal for Innovative Engineering and Management Research

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

ANALYSING ONLINE PRODUCT BY USING PQ MODEL

MS.G.PRASANTHI[1], K.MAHESH[2]
Assistant Professor[1], Department of M.C.A ,Vignan's Lara Institute of Technology & Science
M.C.A Student[2],Department of M.C.A ,Vignan's Lara Institute of Technology & Science

Abstract:
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.
Index Terms—Online indexing model, product quantization, nearest neighbour search.

Introduction

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.

Existing system:

ANN search in a dynamic database has 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. For object detection in video surveillance, video data is continuously recorded, so that the distances

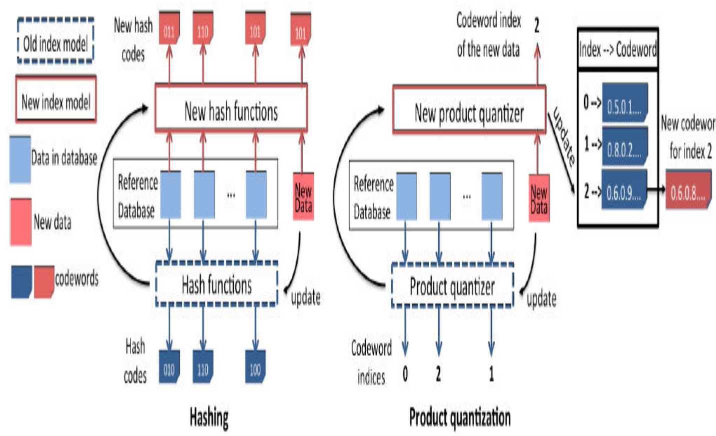
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

between/among similar or dissimilar objects are continuously changing. For image retrieval in dynamic databases, 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.

Proposed system:

Product quantization (PQ) 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, quantizationbased 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, we develop an online PQ approach, which updates the codewords 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.

Architecture Diagram



Modules:

Mini-batch Extension:

In addition to processing one streaming data at a time, our framework can also handle a mini-batch of data at a time. In the case of processing mini-batch of data, we assume that each time we get a new batch of data points. where B is the size of the mini-batch.

Partial Codebook Update:

As we mentioned in the introduction, one of the issues is that online indexing model might incur high computational cost in update. Each new incoming data point might contribute in different significance of changes in different subspaces of nearest sub-codeword update. An obvious example of this is that, given a new streaming data, one of its sub-vector is far from its nearest sub-codeword and another of its sub-vector is close to its nearest sub-codeword then the first one contributes more in PQ index update than the second one.

LOSS BOUND:

Our model, on the other hand, is non-convex and has matrices as variables, which makes the analysis nontrivial to be handled. Moreover, each of the continuously learned codewords may not be consistently matching with each codeword in the best fixed batch model.

Conclusion:

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.

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 stochastic linear 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 sketching hashing," in CVPR, 2015, pp. 2503–2511.