Rong Yan
Snapchat

Multimedia Hashing and Networking

Wei Liu
Tencent AI Lab

Tongtao Zhang
Rensselaer Polytechnic Institute

Many Internet companies frequently handle heterogeneous, multimedia data. Well-known social media websites, including Facebook, Twitter, and YouTube, along with mobile apps such as Instagram, Snapchat, and WeChat, all face the same problem—how can they efficiently and effectively store, index, search, manage, analyze, and understand multimedia data? Here, we attempt to address this problem by studying two popular topics in multimedia: hashing and networking.

Multimedia Hashing

We explore two different methodologies related to multimedia hashing—shallow-learning-based hashing and deep-learning-based hashing—demonstrating state-of-the-art techniques for enabling efficient multimedia storage, indexing, and retrieval.

Hashing by Shallow Learning

Hashing[1] has attracted considerable attention from researchers and practitioners in computer vision, machine learning, data mining, information retrieval, and other related areas. A variety of hashing techniques have been developed to encode documents, images, videos, or other types of data into a set of binary codes (used as hash keys), while preserving certain similarities among the original data. With such binary codes, similarity searches can be rapidly performed over massive datasets, thanks to the high efficiency of pairwise comparison using the Hamming distance.

Early endeavors in hashing concentrated on employing random permutations or projections to construct hash functions. Well-known representatives include Min-wise Hashing (MinHash)[2] and Locality-Sensitive Hashing (LSH).[3] MinHash estimates the Jaccard set similarity, while LSH accommodates various distance or similarity metrics—such as the ℓ_p distance for $p \in (0, \quad 2]$, cosine similarity, and kernel similarity. Due to randomized hashing, more bits per hash table are required to achieve high precision. This typically reduces recall, and multiple hash tables are thus required to achieve satisfactory accuracy of retrieved nearest neighbors. The overall number of hash bits used in one application can easily run into the thousands.

Beyond data-independent randomized hashing schemes, a recent trend in machine learning is to develop data-dependent hashing techniques that learn a set of compact hash codes based on a training dataset (a multimedia database, for example). Binary codes have been popular in this scenario because of their simplicity and efficiency in computation. The compact hashing scheme can accomplish almost a constant-time nearest neighbor search, after encoding the entire dataset into short binary codes and then aggregating them into a hash table. Additionally, compact hashing is particularly beneficial for storing massive-scale data. For example, saving one hundred million samples, each with 100 binary bits, costs less than 1.5 Gbytes, which can easily fit in memory.

To create effective compact hash codes, numerous methods have been presented, including unsupervised and supervised methods. The state-of-the-art unsupervised hashing method, Discrete Graph Hashing (DGH),[4] leverages the concept of "anchor graphs" to capture the neighborhood structure inherent in a given massive dataset, and then formulates a graph-based hashing model over the entire dataset. This model hinges on a novel discrete optimization procedure to achieve nearly balanced and uncorrelated hash bits, where the binary constraints are explicitly imposed and handled. The DGH technique has been demonstrated to outperform the conventional unsupervised hashing methods, such as Iterative Quantization, Spectral Hashing, and Anchor Graph Hashing,[1] which fail to sufficiently

1070-986X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

capture local neighborhoods of raw data in the discrete code space.

The state-of-the-art supervised hashing method, Supervised Discrete Hashing (SDH),[5] incorporates supervised label information and formulates hashing in terms of linear classification, where the learned binary codes are expected to be optimal for classification. SDH applies a joint optimization procedure that jointly learns a binary embedding and a linear classifier. The SDH technique has also been demonstrated to outperform previous supervised hashing methods.[1]

There exist many other interesting hashing techniques, such as document hashing,[6] video hashing,[7] structured data hashing,[8] and intermedia hashing.[9] Note that all of the techniques we have mentioned depend on shallow-learning algorithms. Nonetheless, owing to the high speed of shallow-learning-based hashing, the state-of-the-art hashing techniques have been widely used in high-efficiency multimedia storage, indexing, and retrieval, especially in multimedia search applications on smartphone devices. Several well-known startups, such as Snapchat, Pinterest, SenseTime, and Face++, use proper hashing techniques to manage and search through millions or even billions of images.

Hashing by Deep Learning

Since 2006, *deep learning,*[10] also known as *deep neural networks,* has drawn enormous attention and research efforts in a variety of artificial intelligence areas, including speech recognition, computer vision, machine learning, and text mining. Deep learning aims to learn robust and powerful feature representations for complexly shaped data, so it's natural to leverage deep learning for pursuing compact hash codes, which can be regarded as binarized representations of data. Here, we briefly introduce two recently developed hashing techniques related to deep learning: Convolutional Neural Network Hashing (CNNHash) and Deep Neural Network Hashing (DNNHash).

Previous hashing techniques, relying on deep neural networks, took a vector of hand-crafted visual features extracted from an image as input. The quality of the generated hash codes thus heavily depended on the quality of the hand-crafted features. To remove this barrier, the CNNHash approach was recently developed to integrate image-feature learning and hash-code learning into a joint learning model.[11] This model consists of a stage of learning approximate hash codes given pairwise supervised information and a stage of training a deep Convolutional Neural Network (CNN).[12] Benefiting from the power of CNNs, the latter stage of the joint model can simultaneously learn image features and hash codes, directly working on raw image pixels. The deployed CNN comprises three convolution-pooling layers, a standard fully connected layer, and an output layer with softmax functions. The final hash codes are then produced by quantizing the softmax activations of the output layer.

While the CNNHash approach[11] requires separately learning approximate hash codes to guide the subsequent learning of image representation and finer hash codes, a more recent approach, DNNHash, goes further.[13] With DNNHash, image representation and hash codes are learned in one stage so that representation learning and hash learning are tightly coupled to benefit each other. The DNNHash approach incorporates listwise supervised information to train a deep CNN, leading to a currently deepest architecture for supervised hashing. The pipeline of the deep hashing architecture includes three building blocks:

- a triplet of images, which are fed to the CNN and upon which a triplet ranking loss is designed to characterize the listwise supervised information;
- a shared subnetwork, with a stack of eight convolution layers to generate the intermediate image features; and
- a divide-and-encode module to divide the intermediate image features into multiple channels, each of which is encoded into a single hash bit.

Within the divide-and-encode module, there is one fully connected layer and one hash layer. Eventually, the hash code of an image is yielded by thresholding the output of the hash layer. The DNNHash has been shown to outperform CNNHash and several shallow-learning-based supervised hashing approaches in terms of image search accuracy.[13]

However, for both CNNHash and DNNHash, note that researchers have not yet investigated or reported on the time required for hash code generation. In real-world search scenarios, the speed for generating hashes should be substantially fast. There might be concern about the

hashing speed of these deep-neural-network driven approaches, especially those involving image feature learning, because it might take longer to hash an image with deep learning compared to with shallow-learning-driven approaches.

Multimedia Networking

Here, we introduce the latest Multimedia Information Networks (MINets). As an example of leveraging MINets, we present the cross-media coreference, which incorporates both visual and textual information to reach a sensible event coreference resolution.

Multimedia Information Networks

Recent developments in Web technology—especially in fast connection and large-scale storage systems—have enabled social and news media to publish more in-depth content in a timely manner. However, such developments also raise some issues, such as overwhelming social media information and distracting news media content. In many emergent scenarios, such as encountering a natural disaster (for example, Hurricane Irene in 2011 or Hurricane Sandy in 2012), tweets and news are often repeatedly spread and forwarded in certain circles, so the corresponding content is overlapping. Browsing these messages and pages is unpleasant and inefficient, so an automatic summarization of tweets and news is desired, among which ranking is the most intuitive way to inform users of highly relevant content.

A passive (and common) solution is to prompt users to add more keywords when typing search queries. However, without prior knowledge, and given word limits, it's never trivial to establish a satisfying ranking list for the topics that attract the most public attention. Recent changes in the Google search engine have integrated the image search component and adopted some heterogeneous content analysis. Nevertheless, the connections between images and relevant keywords are still arbitrarily determined by users, so the current search quality is far from optimal.

Active solutions that attempt to summarize information only concentrate on single data modalities. Researchers have developed a context-sensitive topical PageRank method to extract topical key phrases from Twitter as a way to summarize twitter contents.[14] From a new perspective, the Latent Dirichlet Allocation (LDA)[15] model was employed to annotate images,[16] but this doesn't firmly integrate the information across different data modalities. Researchers have also developed a tweet ranking approach,[17] but it only focuses on a single data modality (text).

The connections between images and relevant keywords are still arbitrarily determined by users.

Other conventional solutions for analyzing the relationships or links between data instances include PageRank and VisualRank.[18] The former has been extensively used in heterogeneous networks (webpages and resources), but it mainly concerns linkage. VisualRank, which extends PageRank to the image domain, is a content-based linkage method, but it's confined to homogeneous networks.

A novel MINets[19] representation was recently proposed to create a basic ontology of a powerful ranking system, which aims to integrate cross-media inference and create the linkage among the multimodal information extracted from heterogeneous data. Beyond traditional ranking approaches, designed for homogeneous networks or simple heterogeneous networks, many researchers are developing a series of novel ranking approaches to exploit the properties of MINets, leading to startups such as Toutiao and Tumblr.

Cross-Media Coreference

However, such information networks, where each node represents one event, can suffer from redundant events and low efficiency due to the repeated nodes, because the same stories are often reported by multiple newscast agents. Moreover, to strengthen the impact on audiences and readers, the same stories and events are reported multiple times, especially on TV and in radio broadcasts.

These properties call for automatic methods that can cluster information and remove redundancy. A method has been proposed[20] that not only deals with information from both visual (video contents) and textual (enclosed captions) channels but also analyzes event coreferences.

Thus this method can fully exploit TV news (or newscasts) containing audio and videos.

A good starting point for cross-media coreference is the processing of closed captions (CCs) that accompany videos in a newscast. Such CCs are either generated by automatic speech recognition (ASR) systems or transcribed by a human stenotype operator who inputs phonetics, which are instantly and automatically translated into texts from which events can be extracted. Different from written news, a newscast is often limited in time due to fixed TV program schedules, so anchors and journalists are trained and expected to organize reports that are comprehensively informative with complementary visual and CC descriptions within a short time. These two descriptions have minimal overlap, even though they're interdependent. For example, anchors and reporters introduce background stories that aren't presented in the videos, so the events extracted from the CCs often lack key information about participants.

Another challenge comes from the mistakes that reside in CCs, caused by errors made by human operators or ASR systems. For example, in two similar newscasts, where the death of Jordanian pilot was reported, the closed caption in one cast was mistakenly printed as "It's not clear when it was killed," where "it" should have been "he," referring to the Jordanian pilot.[20] The other newscast had another flawed CC: "Jordan just executed two ISIS prisoners, direct retaliation for the capture of the killing Jordanian pilot" (instead of "capture of the Jordanian pilot"). It's impossible for any existing text-based coreference resolution approach to cluster the two Life.Die event mentions into the same event, because in most natural language processing systems, "it" must not be linked to "Jordanian pilot." Fortunately, videos often illustrate brief descriptions with vivid visual content, and both newscasts adopted the video frames demonstrating the capture of the Jordanian pilot, so these two event mentions can be considered as the same one.

In fact, diverse anchors, reporters, and TV channels tend to use similar or even identical video content to describe the same story, even though they usually use different words and phrases. Therefore, the challenges in coreference resolution methods relying on text information can be addressed by incorporating visual similarity.

Similar work has explored methods for linking visual cues with texts.[21–23] However, these methods mainly focus on connecting image concepts with entities in text mentions, and some didn't clearly distinguish `entity` from `event` in the documents, because the definitions of visual concepts often require both. In addition, the work[21–23] is mostly dedicated to improving visual content recognition by introducing textual features, while the more recent work[20] takes the opposite route by leveraging visual information to improve event coreference resolution.

In the future, we expect to apply the techniques discussed here to make deep learning practical in realistic multimedia applications. For example, we plan to develop deep neural network compressing techniques to endow deep-learning-driven hashing methods with the real-time hashing speed. We also plan to introduce end-to-end memory networks to understand visual and textual information more thoroughly, leading to stronger cross-media event coreference methods. **MM**

References

1. J. Wang et al., "Learning to Hash for Indexing Big Data—A Survey," *Proc. IEEE*, vol. 104, no. 1, 2015, pp. 34–57.
2. A.Z. Broder et al., "Min-Wise Independent Permutations," *J. Computer and System Sciences*, vol. 60, no. 3, 2000, pp. 630–659.
3. A. Andoni and P. Indyk, "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions," *Comm. ACM*, vol. 51, no. 1, 2008, pp. 117–122.
4. W. Liu et al., "Discrete Graph Hashing," *Advances in Neural Information Processing Systems*, vol. 27, 2014, pp. 3419–3427.
5. F. Shen et al., "Supervised Discrete Hashing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 37–45.
6. H. Li, W. Liu, and H. Ji, "Two-Stage Hashing for Fast Document Retrieval," *Proc. Ann. Meeting of the Assoc. Computational Linguistics*, 2014, pp. 495–500.
7. G. Ye et al., "Large-Scale Video Hashing via Structure Learning," *Proc. IEEE Int'l Conf. Computer Vision*, 2013, pp. 2272–2279.
8. W. Liu et al., "Compact Hyperplane Hashing with Bilinear Functions," *Proc. Int'l Conf. Machine Learning* (ICML), 2012, pp. 17–24.
9. J. Song et al., "Inter-Media Hashing for Large-Scale Retrieval from Heterogeneous Data Sources," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2013, pp. 785–796.

10. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, 2015, pp. 436–444.
11. R. Xia et al., "Supervised Hashing for Image Retrieval via Image Representation Learning," *Proc. AAAI Conf. Artificial Intelligence*, 2014, pp. 2156–2162.
12. A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 1106–1114.
13. H. Lai et al., "Simultaneous Feature Learning and Hash Coding with Deep Neural Networks," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 3270–3278.
14. W. X. Zhao et al., "Topical Keyphrase Extraction from Twitter," *Proc. 49th Annual Meeting of the Assoc. Computational Linguistics: Human Language Technologies—Volume* 1, 2011, pp. 379–388.
15. D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, vol. 3, 2003, pp. 993–1022.
16. Y. Feng and M. Lapata, "Topic Models for Image Annotation and Text Illustration," *Proc. Ann. Conf. North Am. Chapter of the Assoc. Computational Linguistics*, 2010, pp. 831–839.
17. H. Huang et al., "Tweet Ranking Based on Heterogeneous Networks," *Proc. Int'l Committee on Computational Linguistics and the Assoc. Computational Linguistics* (COLING), 2012, pp. 1239–1256.
18. Y. Jing and S. Baluja, "Visualrank: Applying Pagerank to Large-Scale Image Search.," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, 2008, pp. 1877–1890.
19. T. Zhang et al., "Cross-Media Cross-Genre Information Ranking Multi-Media Information Networks," *V&L Net*, 2014, p. 74.
20. T. Zhang et al., "Cross-Document Event Coreference Resolution Based On Cross-Media Features," *Proc. Conf. Empirical Methods in Natural Language Processing*, 2015, pp. 201–206.
21. V. Ramanathan et al., "Video Event Understanding Using Natural Language Descriptions," *Proc. Int'l Conf. Computer Vision*, 2013, pp. 905–912.
22. V. Ramanathan et al., "Linking People in Videos with 'Their' Names Using Coreference Resolution," *Proc. European Conf. Computer Vision*, 2014, pp. 95–110.
23. C. Kong et al., "What Are You Talking About? Text-to-Image Coreference," *Proc. Conf. Computer Vision and Pattern Recognition*, 2014, pp. 3558–3565.

Wei Liu is a technical leader and research manager at Tencent AI Lab. Contact him at wliu@ee.columbia.edu.

Tongtao Zhang is a PhD candidate in the Computer Science Department at Rensselaer Polytechnic Institute. Contact him at zhangt13@rpi.edu.