Scalable Mobile Video Retrieval with Sparse Projection Learning and Pseudo Label Mining

Guan-Long Wu, Yin-Hsi Kuo, Tzu-Hsuan Chiu, Winston H. Hsu, and Lexing Xie

Abstract

Retrieving relevant videos from a large corpus on mobile devices is a vital challenge. This paper addresses two key issues for mobile search on user-generated videos. The first is the lack of good relevance measurement, due to the unconstrained nature of online videos, for learning semantic-rich representations. The second is due to the limited resource on mobile devices, stringent bandwidth, and delay requirement between the device and the video server. We propose a knowledge-embedded sparse projection learning approach. To alleviate the need for expensive annotation for hash learning, we investigate varying approaches for pseudo label mining, where explicit semantic analysis leverages Wikipedia and performs the best. In addition, we propose a novel sparse projection method to address the efficiency challenge. It learns a discriminative compact representation that drastically reduces transmission cost. With less than 10% non-zero element in the projection matrix, it also reduces computational and storage cost. The experimental results on 100K videos show that our proposed algorithm is competitive in the performance to the prior state-of-the-art hashing methods which are not applicable for mobiles and solely rely on costly manual annotations. The average query time on 100K videos consumes only 0.592 seconds.

Index Terms

hashing, sparsity, mobile video retrieval, explicit semantic analysis

1 Introduction

Content-based video search is a long-standing research issue, and the remarkable recent growth of user-generated online videos has made searching such collections in real-time a mounting challenge. Furthermore, video applications and search are moving to mobile devices from desktops. This paper sets off to address two challenges for video retrieval in mobile platforms. The first one is efficiency – there are limited computing power and bandwidth on the mobile device, which makes heavy-weight computation on the query video and transmitting the whole video infeasible (i.e., minutes of uploading time over 3G network) [1]. The second challenge is relevance – finding semantically related videos is not a new problem, but doing so in a compact local representation with the help of video metadata is new. This paper proposes two core techniques, sparse projection learning and pseudo label mining, to address the pair of related challenges.

To tackle the efficiency challenge for mobile video search, hash-based approach is a promising way to generate a compact representation (binary codes, fingerprint, signature) from the original feature [2]. The most widely adopted hash-based method utilizes a (dense) projection matrix to generate compact representations (cf. Fig. 1(c)). However, if the original feature space is high dimensional (e.g., the state-of-the-art image/video feature: vector of locally aggregated descriptors - VLAD [3]), the projection matrix costs large amount of memory (e.g., hundreds of MBs) in storage and computational time in generating the fingerprint, which is nearly infeasible on mobile devices. For example, the projection matrix might consume 390MB when the original feature dimension is 200K and the reduced dimension is 256. In this paper, we present a novel knowledge-embedded sparse projection learning algorithm to tackle this problem. The

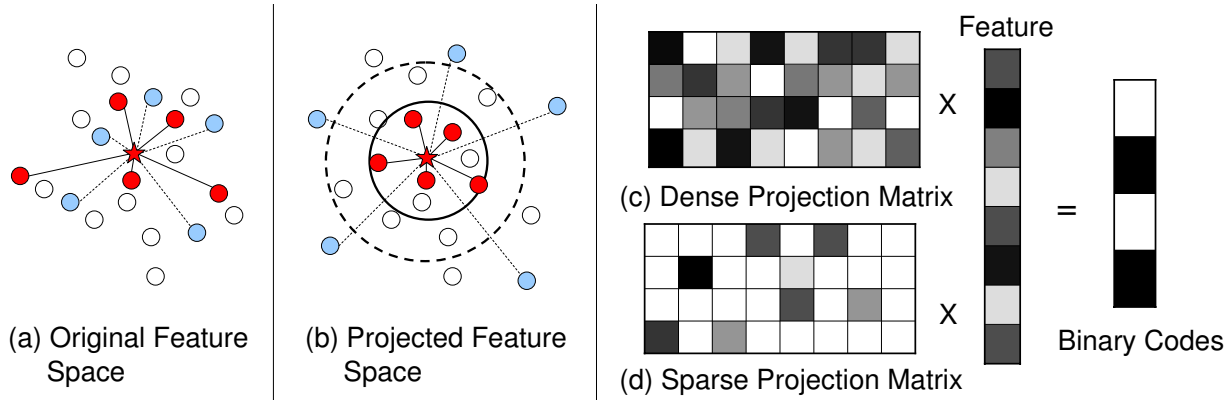


Fig. 1. The main concepts of the knowledge-embedded sparse projection learning approach. (a) When given a query (red star), we will retrieve both semantic related data (red circles, connected by solid lines) and semantic unrelated data (light blue circles) by calculating visual similarity only. (b) By utilizing contextual information (e.g., surrounding text), we help to retrieve more semantic related data in the projected (transferred) feature space. To avoid time-consuming manual annotations, the proposed pseudo label mining algorithm can automatically exploit semantic related data. (c) A widely adopted hash-based approach generates compact representations (binary codes) via the dense projection matrix. (d) However, for mobile video retrieval, the limited memory space and computing power motivate us to learn a sparse projection matrix which also considers semantic relevance in the learning process. Different colors represent different values and white = 0.

advantages of the sparse projection matrix (cf. Fig. 1(d)) are that it can be stored by sparse representation, be loaded to memory of a mobile device more efficiently, and speed up the fingerprint generation. Moreover, only the fingerprint of the query video is transmitted to the server to rank similar videos in real-time. Experimental results show that we can achieve similar retrieval performance as using only 9.45% of memory consumed by the dense projection matrix.

Furthermore, we observe that users tend to prefer top ranked search results which are semantically similar than visually similar. If we only utilize visual similarity in the search process, we will retrieve both semantic related and unrelated videos (cf. Fig. 1(a)). Thus, it is very essential to incorporate partial semantic knowledge when learning the projection matrix. Based on the learned knowledge-embedded projection matrix, we will have better representations in the projected feature space (cf. Fig. 1(b)). Besides, the generated semantic-rich compact representations (i.e., hundreds of bits) can further improve the retrieval accuracy. To incorporate semantic relevance, the most intuitive way is to annotate labels for each video manually. However, it is intractable to acquire human-annotated data especially in real and explosively growing data. For video sharing website, such as YouTube, there are plenty of videos that consist of contextual information (e.g., title, description and tags) provided by users. This plentiful contextual information is beneficial for measuring the semantic relevance between videos. However, the contextual information of videos are often noisy. To overcome this problem, we investigate different pseudo label mining algorithms based on the contextual information to measure the semantic relevance (or labels). We examine the quality of the automated generated annotations, propose how to leverage them for automatically generating semantic-rich fingerprints, and investigate the successful and failure cases in our experimental datasets which consist of 10K and 100K videos.

The primary contributions of the paper include,

- Proposing a knowledge-embedded sparse projection learning approach which generates semantic-rich compact representations for mobile video retrieval under limited resource (Section 3).

- Investigating the help of contextual information for the reliable pseudo label mining algorithm which can avoid time-consuming manual annotations (Section 4).
- Conducting experiments on user-generated videos and showing great reduction of computational cost and memory consumption while achieving similar retrieval accuracy as the state-of-the-art approaches (Section 5).

2 Related Work

Scalable search on mobile platform has been an active research area in the past few years. It generally takes at least dozens of seconds to transmit over 3G network in the original image or video format. Due to the limited bandwidth, it had been argued that instead of sending the entire video or image (several KBs to MBs) back to the server we should send back the compact features (hundreds of bits) [1]. Hence, recent researches focus on extracting and compressing features on mobile devices to reduce the transmission time [1][2]. Progress has been made on the design of hashing algorithms, semantic similarity estimation, and paradigms for mobile visual search.

A widely adopted hash-based approach generates compact (binary) representations via the similarity in the original feature space (e.g., visual similarity) [4]. Authors in [5] further propose to learn the projection matrix and incorporate partial knowledge of the data (label information) to increase the discriminative power of compact representations. However, their method requires manual annotated data which generally overfits the dataset and is not scalable to real and large-scale dataset. Moreover, the learned projection matrix is a dense matrix (cf. Fig. 1(c)), which imposes high storage requirement for the matrix itself, and is demanding in power consumption for computing the hash signature on mobile devices. Therefore, we propose a knowledge-embedded sparse projection learning approach which learns a sparse projection matrix (cf. Fig. 1(d)) with semantic relevance.

Recent advances in online data collection and curation have led to two prevalent methods for computing the semantic relevance between text segments. One relies on knowledge bases such as WordNet [6], ConceptNet [7], and Yago [8]. Most knowledge bases consist of many concepts which are connected to one another as a multi-relational graph. Another approach is data-driven, where semantic similarity is computed from a corpus, such as using Wikipedia [9] or web search results [10]. In particular, explicit semantic analysis [9] represents the meaning of texts in a high-dimensional space of concepts derived from Wikipedia so that the semantic relevance of sentences can be calculated directly. Since Wikipedia consists of a large amount of articles with remarkable quality and large diversity, it is suitable for determining semantic relatedness between videos. Thus, we choose explicit semantic analysis and web-based kernel function [10] using Google search engine to measure the semantic relevance because the corpus representation is intuitive to estimate semantic similarity and contains more information for specific events, persons or objects.

For mobile video retrieval, Chen *et al.* [11] proposed a dynamic feature-rich frame selection algorithm from a sequence of viewfinder frames in a very short temporal window determined by the user-initiated query event. Although it improves search accuracy, the transmission cost of encoded local feature points is much higher than a low-dimensional binary signature. He *et al.* [2] proposed a mobile visual search system based on "bag of hash bits." The mobile client side hashes the local features of a query image to a bag of hash bits using multiple hash tables and transmits it to the server. However, if an image consists of 200 local features, the total transmitted bits (e.g., 200 x 80 bits) of a mobile query is 83 times larger than our proposed method (e.g., 192 bits).

3 KNOWLEDGE-EMBEDDED SPARSE PROJECTION LEARNING

To tackle the challenges for mobile video retrieval, we propose a knowledge-embedded sparse projection learning approach.[1] By considering the limited resource and transmission time, we integrate hash-based method with sparse projection learning to generate compact representation on mobile devices in Section 3.1 and Section 3.2. To further generate semantic-rich compact representations, we propose pseudo label mining algorithm which utilizes contextual information to measure the semantic relevance in Section 4.

3.1 Hash-Based Approaches for Binary Representation

The most widely adopted hashing functions utilize linear projections (projection matrix) to generate compact representations. Assume the original dataset $\mathbf{X} = [\mathbf{x}_i]$, where $\mathbf{x_i} \in \mathbb{R}^D$ and $\mathbf{X} \in \mathbb{R}^{D \times n}$ contains n data instances which has D dimensions. The projection-based hashing functions are calculated by $h_k(\mathbf{x}) = sgn(\mathbf{w_k^T x} + \mathbf{b_k})$, where $\mathbf{W} = [\mathbf{w}_k]$ and $\mathbf{W} \in \mathbb{R}^{D \times K}$. $\mathbf{w}_k \in \mathbb{R}^D$ is the k-th linear projection vector, and $\mathbf{b_k}$ is a threshold value. The corresponding bit for $\mathbf{x_i}$ is expressed as $y_k(\mathbf{x_i}) = (1 + h_k(\mathbf{x_i}))/2$. Note that we can normalize data to have zero mean and set $b_k = 0$ for mean thresholding [5]. The most simple but effective hashing function is designed by random projection (RP) [4] which generates $\mathbf{W}$ with entries in $\{-1, 1\}$ with probability $\{1/2, 1/2\}$. Moreover, they also consider the sparsity of the projection matrix to reduce the computational cost and propose sparse random projection (Sparse RP) which approximates RP by calculating $\mathbf{W}$ with entries in $\{-1, 0, 1\}$ with probability $\{1/6, 2/3, 1/6\}$. To further integrate extra knowledge (e.g., label information), the authors in [5] propose semi-supervised sequential projection learning (S3PLH) algorithm. They design the objective function by maximizing the empirical accuracy in a small portion of neighbor/non-neighbor data pairs and the variance of hash bits in the whole dataset. Therefore, we will compare RP, Sparse RP, and S3PLH in the experiments.

3.2 Sparse Projection Learning (SHP)

For mobile video retrieval, it is not feasible to load the dense projection matrix (cf. Fig. 1(c)) on mobile devices. Prior (linear) projection matrix for generating compact hash bits is only considered on the server environment rather than on mobiles, which have very limited memory space and computing power. Besides, as shown in Fig. 1(a), if we directly apply hashing algorithm in visual domain, some semantic unrelated data might be assigned to similar binary representations as the query. Therefore, we propose to embed the automatically generated semantic relevance with sparse constraint (i.e., sparse projection matrix in Fig. 1(d)) in the learning process. Thus, we can obtain semantic-rich compact representations in the projected feature space as shown in Fig. 1(b). Motivated by [5] which utilizes true label information, we design the automatically generated semantic relevance (pseudo label) matrix ($\mathbf{S}$) as follows

$$\mathbf{S}_{ij} = \begin{cases} 1 & : SIM(\mathbf{x}_i, \mathbf{x}_j) > T_u \\ -1 & : SIM(\mathbf{x}_i, \mathbf{x}_j) < T_l, \\ 0 & : otherwise \end{cases} \tag{1}$$

where $SIM(\mathbf{x}_i, \mathbf{x}_j)$ is the semantic similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. T_u (T_l) is the similar (non-similar) threshold. We will show that such automatically mined pseudo labels (i.e., $\mathbf{S}_{ij}$ - semantic relevance) have competitive performance with time-consuming manual annotations in Section 5.3. Hence, the proposed knowledge-embedded sparse projection learning contains three essential parts: 1) embedding semantic relevance from partial training data (X_l), 2) integrating maximum variance of all the training data (X) in the projected feature space, and 3) enforcing the sparsity

1. Similar to prior hashing learning methods, the proposed method can be implemented on the server sides. Nevertheless, we further consider whether it is feasible for the mobile environment.

(L_1-regularization) for the projection matrix (to be embedded in the mobile device rather than the server). Therefore, we propose to formulate the hashing functions as

$$W^* = \arg\max_W \frac{1}{2} tr\{H(X_l)SH(X_l)^T\} + \frac{\eta}{2} tr\{H(X)H(X)^T\} - \gamma \sum_k ||\mathbf{w}_k||_1, \quad (2)$$

where η modulates the importance between the first and second terms, and γ is the tuning parameter to control the sparsity. The first term attempts to generate similar compact representations if they have the same pseudo label; therefore, it will embed the knowledge (semantic relevance) in the learning process. The second term can also be viewed as minimizing reconstruction error in the original feature space; hence, it will preserve the original visual similarity. The sparse constraint can not only learn a compact projection matrix (i.e., much less memory consumption) but also select few essential dimensions to generate the compact representation. As shown in Fig. 1(d), the first row of the sparse projection matrix (i.e., the first hashing function) only considers the fourth and sixth dimensions to generate the first binary representation. By the definition in Section 3.1 and relaxing the $sgn()$ constraint, we can rewrite the formulation as $\frac{1}{2} tr\{\mathbf{W^T}[\mathbf{X}_l\mathbf{S}\mathbf{X}_l^T + \eta\mathbf{X}\mathbf{X^T}]\mathbf{W}\} - \gamma \sum_k ||\mathbf{w}_k||_1$. To solve this optimization, we find that is can be viewed as a sparse principle component analysis (Sparse PCA) [12]. Because we can rewrite the objective function as

$$\max_{\mathbf{e}} \sqrt{\mathbf{e}^T\mathbf{M}_k\mathbf{e}} - \gamma||\mathbf{e}||_1, \text{ subject to } \mathbf{e}^T\mathbf{e} \leq 1, \quad (3)$$

where $M_k = [\mathbf{X}_l\mathbf{S_k}\mathbf{X}_l^T + \eta\mathbf{X}\mathbf{X^T}]$ and $e = w_k$. Hence, we can solve this optimization problem by [12], and sequential update the S_k (i.e., M_k) followed by [5].

4 Pseudo Label Mining

The proposed method is for scalable retrieval and needs to scale to new dataset by automatically exploiting new annotations; while prior hash learning methods are mostly deployed on small scale data with perfect manual annotations. Therefore, it is essential to generate the semantic relevance matrix $\mathbf{S}$ in Equation (2). We observe that some videos have contextual data provided by users (e.g., title, description and tags) on video-sharing websites (e.g., YouTube). Although the context data of videos are noisy, based on the power of knowledge base, the semantic similarity of data pairs is more convincing. Leveraging semantic relevance measurement, the annotation process can be fully automated. We choose explicit semantic analysis (ESA) based on Wikipedia (Section 4.1) and web-based kernel function (WKF) using Google search engine (Section 4.2) to measure the semantic relevance. Hence, the proposed pseudo label mining algorithm can automatically exploit the semantic related data.

4.1 Explicit Semantic Analysis (ESA)

Motivated by [9], we utilize the knowledge from Wikipedia to generate semantic representations for each video. To obtain the most representative Wikipedia articles (concepts), We use the research-esa [13] library to execute the explicit semantic analysis (ESA) algorithm and an English Wikipedia snapshot as of April 03, 2012. In order to get more general Wikipedia concepts to represent the text input, the filter criteria are as follows,

1) The article is not in the main namespace.
2) The name of article is in month_year (e.g., January 2008), year_in_... (e.g., 2002 in literature), only digits (e.g., 1998) or list (e.g., List of ...) format.
3) The article with number of inlinks < 5 or outlinks < 5.
4) The article with fewer than 100 unique non-stop words and the article $<$ 3KB.

After the pruning process, there are N_{wiki} (e.g., $732,340$) Wikipedia concepts left in the semantic representation. Based on these Wikipedia concepts, we can calculate the similarity between the

contextual information of videos ($V^{context}$) and the concepts (C_n). Hence, we can obtain the semantic representation $V^{\text{wiki}} = \{v_1^{\text{wiki}}, ..., v_{N_{\text{wiki}}}^{\text{wiki}}\}$ by calculating $v_n^{\text{wiki}} = V^{context} \cdot C_n$. Note that we can efficiently calculate the similarity by inverted indexing. Finally, the top 1000 largest values of V^{wiki} are kept as the final semantic representation.

4.2 Web-based Kernel Function (WKF)

Another pre-existing approach for metadata similarity is to use completely unstructured sources via web search. Web-based kernel function (WKF) [10] utilizes the web search engine's results to enrich the semantic meaning of query text effectively. Similarly, we utilize the contextual information of videos ($V^{context}$) to expand more meaningful text from the Google search results. We can obtain n Google snippets returned by a Google search engine (e.g., n=50 in this work). Then, we aggregate all the snippets (R_n) to form a new semantic representation $V^{\text{google}} = \frac{1}{n}\sum_{i=1}^{n} R_i$, where R_i has N_{google} (e.g., $91,004$) dimensions (words).

5 Experiments and Discussions

We evaluate the retrieval performance of the proposed knowledge-embedded sparse projection learning approach on two (online) video datasets in Section 5.1, and summarize the experimental settings in Section 5.2. Section 5.3 evaluates retrieval performances in various hashing algorithms on NUS-WEBV dataset. Section 5.4 reports the discriminative power of pseudo label mining based on the unstructured text information of YouTube videos. Section 5.5 shows the retrieval performance and efficiency on 100K video dataset. Finally, Section 5.6 reports the effect of the sparse constraint for the projection matrix.

5.1 Datasets

To evaluate the performance of our proposed method on retrieving semantic related videos, we conduct the experiments on NUS-WEBV and DS100K datasets.

NUS-WEBV dataset [14] consists of 10,130 YouTube videos with ground truth labels. The videos are crawled from 60 predefined event queries, and labeled with the relevance to the corresponding event query by human annotators. The degree of relevance is divided to 3 levels: very relevant, relevant and irrelevant. The event queries contain diverse topics from Natural, Airshow, Political, Entertainment, Social and Sports.

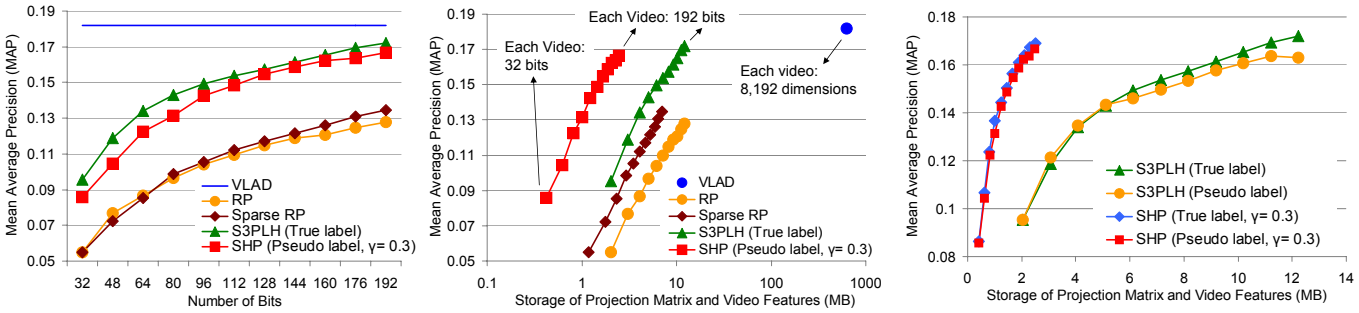
DS100K dataset consists of 104,000 videos which combines with NUS-WEBV, the Iran and SwineFlu videos [15] (56,834 videos), and additional 37,036 videos randomly selected from UQ_VIDEO [16] dataset except CC_WEB_VIDEO dataset to evaluate the scalability issue.[2]

5.2 Experiment Settings

We choose vector of locally aggregated descriptors (VLAD) [3] to describe each video. This is because our recent work [17] reported that VLAD achieves the state-of-the-art video retrieval performance, and lowers the computational cost and memory consumption.[3] Hence, the VLAD-based video feature is used as the visual retrieval baseline in our paper. To extract video feature, the keyframes are extracted by abrupt shot detection. Then, feature points are calculated by hessian-affine detector with SIFT for each keyframe. Finally, video feature (VLAD) is generated

2. The evaluation of UQ_VIDEO is based on CC_WEB_VIDEO which is for near-duplicate video retrieval. Besides, our work [17] reported that VLAD-based features achieve 0.96 retrieval accuracy on CC_WEB_VIDEO so that we do not evaluate on UQ_VIDEO directly.

3. Our work reported that VLAD feature only needs 64 centers to reach the same retrieval performance of bag-of-words (BoW) using 200,000 centers. Besides, BoW requires large vocabulary to achieve good performance which is not feasible to load it on mobile devices. The computational cost of computing nearest neighbor of the codebook is reduced significantly using VLAD.



(a) MAP vs. Number of Bits (b) MAP vs. Storage Consumption (c) True vs. Pseudo Label

(d) Sample Retrieval Results by Different Methods

Fig. 2. Performance comparison on NUS-WEBV dataset: (a) Number of bits vs. MAP. The proposed SHP outperforms RP and Sparse RP. (b) Storage consumption vs. MAP. The results show that our proposed SHP algorithm achieves very competitive performance with S3PLH but using much smaller storage. (c) Performance comparison for S3PLH and SHP using ground truth labels and pseudo label mining. The retrieval performance using pseudo label mining is very similar with that of ground truth labels. In (b) and (c), each curve is obtained by varying the number of hash bits from 32 to 192. (d) Sample retrieval results. The proposed SHP only uses 192 bits to achieve similar results as VLAD baseline (8,192 dimensions).

by aggregating the difference of the feature points over all keyframes and their nearest centroid of codebook. The dimension of a VLAD vector is 8,192 and the distance metric is L_2. Note that the proposed method is not limited to use VLAD feature and it can be easily applied to other state-of-the-art features.

The retrieval evaluation metric is mean average precision (MAP) computed at retrieval depth 10 and averaged over 5 test runs. For the experiment in NUS-WEBV, in each test run, we randomly select 10 "relevant"/"very relevant" videos of each event as testing data and the remaining videos are training data. For constructing pairwise relationship matrix $\mathbf{S}$, we randomly select 25 "relevant"/"very relevant" videos of each event. If the number of the ground truth videos of an event is lower than 10, we randomly select $1/3$ of it as testing data, the remaining $2/3$ of it as training data, and $1/2$ of the training data for $\mathbf{S}$. For the experiment in DS100k, we sample 20K videos as the training data and uniformly sample the different numbers of videos for pseudo labeled training data ($\mathbf{X}_l$).[4] The metric to calculate the distance between two semantic representations is the cosine similarity. Empirically, we choose $T_u = 0.7$ ($T_l = 0.1$) as semantic related (unrelated) pairs. We set $\gamma = 0.3$ which will be discussed in the Section 5.6, and utilize

4. We constrain the ratio of neighbor pair:non-neighbor pair to 1:10 for $\mathbf{S}$. The reason for restricting the ratio is that we hope the neighbor instances in the learned projection vectors can be assigned with the same binary bits.

ESA algorithm to estimate semantic relevance except the results in Section 5.4.

5.3 The Performance of Knowledge-Embedded Sparse Projection Learning on NUS-WEBV

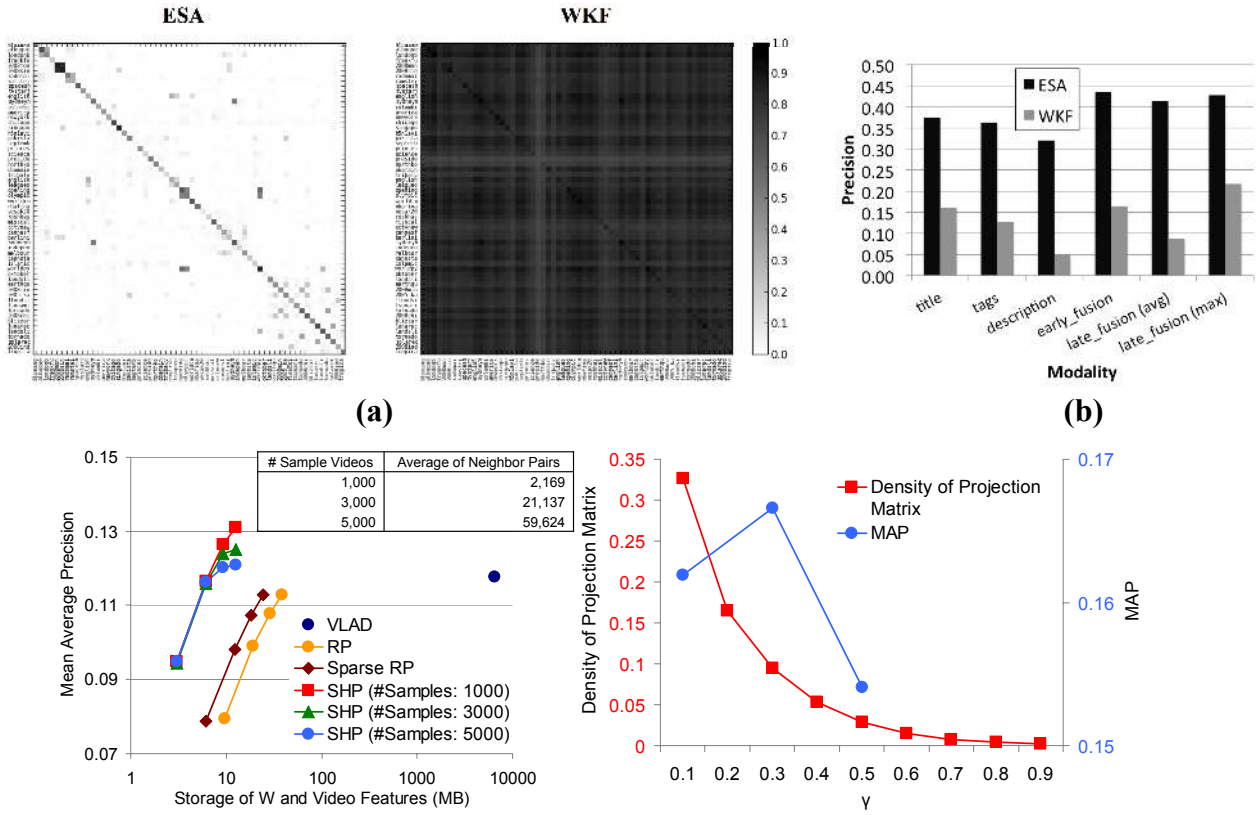
To evaluate the proposed method, we compare with the state-of-the-art approaches on NUS-WEBV. As shown in Fig. 2(a), the retrieval accuracy greatly drops by using random projection (RP) or sparse random projection (Sparse RP) which only considers the visual similarity. To further utilize extra information (e.g., pseudo label information) in the learning process, the proposed SHP can achieve similar retrieval accuracy (e.g., 192 bits) as original high-dimensional VLAD feature (0.182).[5] Besides, our proposed algorithm achieves very competitive MAP performance with S3PLH which needs true label information in the learning process, and outperforms RP and sparse RP. Because we aim at learning a sparse projection matrix which is more applicable for mobile devices, we compare the storage consumption versus MAP by various hashing methods in Fig. 2(b). The MAP increases with the number of compact bits for representation. We find that the proposed method only contains 9.45% non-zero elements in the learned projection matrix; hence, the storage consumption (projection matrix + video features) is much smaller than S3PLH (2.48MB vs. 12.23MB for 192 bits) while achieving competitive MAP performance. The MAP might be low in our experimental results (unlike those near-duplicate cases in CC_WEB_VIDEO dataset, where our VLAD-based features achieve 0.96 retrieval accuracy); however, the top-ranked results are almost relevant to the query as shown in Fig. 2(d). This is because there are too many semantic related videos in the dataset; therefore, the retrieval results will suffer from high precision but low recall. Nevertheless, it is still feasible for mobile video retrieval because users usually care about those top-ranked retrieval results.

To demonstrate the effectiveness of pseudo label mining, Fig. 2(c) shows the MAP performance in different storage for S3PLH and our proposed algorithm between human annotations (true labels, semi-supervised learning) and pseudo label mining (unsupervised learning) for generating compact representations. Note that the mined (pseudo) label information can be applied to S3PLH. It is clear that the MAP of pseudo label mining algorithm is very competitive with the methods using human annotations. The results show that the proposed SHP has similar retrieval accuracy compared to S3PLH but using much smaller storage consumption. This phenomenon is more obvious when we use more bits to describe each video (i.e., the gap between two curves is widening). For example, the reduced storage of 32 and 192 bits is 1.62MB (S3PLH: 2.04MB - SHP: 0.42MB) and 9.75MB, respectively. This also represents that the proposed method only needs a small amount of memory space as increasing the bit numbers. Because we integrate the sparse constraint into the learning process, we can focus on few important dimensions to generate the binary codes (cf. Fig. 1(d)) and achieve similar codes as the dense projection matrix (cf. Fig. 1(c)).

5.4 The Discriminative Power of Semantic Representation by ESA and WKF

To investigate and compare the discriminative power of ESA and WKF for generating the semantic representation, we evaluate the quality of retrieval results by precision at N_e (i.e., N_e = the number of ground truth) for each event category in the NUS-WEBV. The semantic representations of queries are built from the name of the event category. We apply ESA (V^{wiki}) and WKF (V^{google}) to generate three semantic representations based on title, tags, and description ($V^{context}$) for each video in DS100K, and also compare with early fusion (average 3 vectors) and two late fusions (average/maximum 3 similarity scores). As shown in Fig. 3(b), the precision of ESA outperforms WKF significantly in all configurations. The best precision values of ESA and

5. We assume the VLAD baseline is done on the server side because it is a high-dimensional feature which needs higher storage cost (around 633MB for NUS-WEBV).



(c) MAP on DS100K dataset (d) The effect of γ on NUS-WEBV dataset

Fig. 3. (a) Average pair-wise similarity of ESA and WKF for 60 NUS-WEBV categories. (b) Classification precision over 60 categories using ESA and WKF alone. Results show that ESA is notably more discriminative than WKF for different event categories. (c) Performance comparison on DS100k dataset: memory consumption vs. MAP. Each curve is obtained by varying the number of hash bits from 128 to 512. Results show that using SHP with ESA achieves better retrieval performance while using less resource than other hashing methods and VLAD baseline. (d) The density of projection matrix and MAP in different γ on NUS-WEBV. The generated semantic-rich representation is fixed at 192 bits.

WKF are 0.435 (early_fusion) and 0.217 (late_fusion_max), respectively. Note that the precision of WKF with description is only 0.049 because the search engine does not support very long query, which returns a large portion of empty search results. Based on the best configuration in Fig. 3(b), we further compute the average similarity confusion matrix of ESA and WKF for labeled event categories in Fig. 3(a). The ideal case of the confusion matrix should only have black color (similarity = 1) in the diagonal. It is obvious that ESA is more suitable and discriminative than WKF in estimating semantic relevance. Moreover, we find that some ambiguous cases in ESA come from very related categories, such as "2008 Monaco Grand Prix" and "2008 Singapore Grand Prix."

5.5 Retrieval Performance and Efficiency on DS100K

Instead of evaluating on a small dataset (NUS-WEBV), we further experiment on DS100K in Fig. 3(c). Similar to the results shown on NUS-WEBV, the proposed method achieves better retrieval performance while using less resource than other hashing methods and VLAD baseline. Moreover, our SHP (512 bits) further outperforms the VLAD baseline (0.131 vs. 0.118) because the

learning process integrates both visual and semantic information. Besides, the learning process is sequentially learned by solving the Equation (3) so that we can easily include new data (automatically exploited semantic relevance) when increasing the bit numbers. Hence, the proposed method is more scalable than prior hash learning methods which rely on manual annotations. We also investigate the effect by different number of pseudo labeled training samples (X_l) in the right top of Fig. 3(c). The results show that the proposed method only utilizes a small number of X_l (1000 $\sim$ 5000 samples) for generating semantic relevance and achieves good retrieval accuracy.

To examine the video retrieval efficiency between VLAD feature and compact signature, we show the retrieval speed on DS100K dataset. We execute 100 queries and average the query time (distance computation + ranking). The average query time of VLAD (L_2 distance) and our SHP (hamming distance) is 3.201 seconds and 0.592 seconds, respectively.[6] We can not only greatly reduce 81.5% query time but also achieve good retrieval results. The time for calculating hamming distance only takes 0.075 seconds (i.e., 3.13% of the time as VLAD - 2.398s); hence, it is applicable for scalable mobile video retrieval. Moreover, we can further reduce the query time by retrieving videos with the same binary representation (i.e., hamming distance = 0) via lookup table.

5.6 Parameter Sensitivity (γ)

We evaluate the effect of sparsity for the projection matrix by controlling different γ in the experiments on NUS-WEBV. To observe the density of projection matrix, we calculate the number of non-zero elements in the projection matrix. For $\gamma = 0.3$ in Fig. 3(d), we can achieve similar retrieval accuracy as the state-of-the-art S3PLH (0.172) while using 9.45% non-zero elements of the projection matrix. The projection matrix is expressed by sparse matrix representation to reduce the storage cost and computational cost significantly since the zero elements are ignored in the storage and computation. Besides, the generated semantic-rich compact representation (i.e., hundreds of bits) can greatly reduce the transmission time. These properties are especially applicable for the case of high-dimensional features and mobile video (or image) retrieval.

6 Conclusion

We propose a similar video pair generation algorithm based on explicit semantic analysis augmented by contextual data for some of videos, and the experimental result shows this algorithm is effective to generate meaningful similar video pairs without human annotations and still effective in helping unsupervised hash learning. We present a novel sparse projection learning algorithm to reduce the storage and fingerprint computational cost for incorporating knowledge from pseudo label mining and sparsity. The learnt sparse projection matrix is feasible on mobile devices and the retrieval accuracy is ensured. We evaluate our proposed algorithms in NUS-WEBV and a combined video dataset with 100K YouTube videos. The experiment results show the learned projection matrix of our method not only reduces huge storage cost but also reduces the computational cost of fingerprint generation. The MAP performance of our proposed knowledge-embedded sparse projection learning is very competitive to the original video feature. In the future, we will investigate how to incorporate more knowledge to improve retrieval performance systematically.

Acknowledgments

We thank Jun Wang for providing the implementation of [5]. We thank Richang Hong for providing the NUS-WEBV [14] dataset.

6. Our implementation is based on Python and the distance calculation functions of L_2 and hamming distances are optimized by Cython to achieve competitive performance with Native C implementation. The program is run at a computer with Intel Xeon CPU E5620 2.40GHz and only single thread is used.

REFERENCES

[1] B. Girod *et al.*, "Mobile visual search," *IEEE SPM*, 2011.
[2] J. He *et al.*, "Mobile product search with bag of hash bits," in *MM*, 2011, pp. 839–840.
[3] H. Jégou *et al.*, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010, pp. 3304–3311.
[4] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *JCSS*, pp. 671–687, 2003.
[5] J. Wang *et al.*, "Sequential projection learning for hashing with compact codes," in *ICML*, 2010.
[6] C. Fellbaum, *WordNet: An Electronical Lexical Database*. The MIT Press, 1998.
[7] C. Havasi *et al.*, "Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge," in *RANLP*, 2007.
[8] J. Hoffart *et al.*, "YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia," *CACM*, vol. 52, no. 4, pp. 56–64, 2009.
[9] E. Gabrilovich *et al.*, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *IJCAI*, 2007, pp. 1606–1611.
[10] M. Sahami *et al.*, "A web-based kernel function for measuring the similarity of short text snippets," in *WWW*, 2006, pp. 377–386.
[11] D. M. Chen *et al.*, "Dynamic selection of a feature-rich query frame for mobile video retrieval," in *ICIP*, 2010, pp. 1017–1020.
[12] M. Journée *et al.*, "Generalized power method for sparse principal component analysis," *JMLR*, vol. 11, pp. 517–553, Mar. 2010.
[13] P. Sorg. research-esa - an implementation of explicit semantic analysis for research. [Online]. Available: http://code.google.com/p/research-esa/
[14] R. Hong *et al.*, "Exploring large scale data for multimedia qa: an initial study," in *CIVR*, 2010, pp. 74–81.
[15] L. Xie *et al.*, "Visual memes in social media: tracking real-world news in youtube videos," in *MM*, 2011, pp. 53–62.
[16] J. Song *et al.*, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *MM*, 2011, pp. 423–432.
[17] Y.-C. Su *et al.*, "Evaluating gaussian like image representations over local features," in *ICME*, 2012.