Semantics-Preserving Hashing for Cross-View Retrieval

Zijia Lin[1,2], Guiguang Ding[2], Mingqing Hu[3], Jianmin Wang[2]
[1]Department of Computer Science and Technology, Tsinghua University. [2]School of Software, Tsinghua University. [3]Institute of Computing Technology, Chinese Academy of Sciences

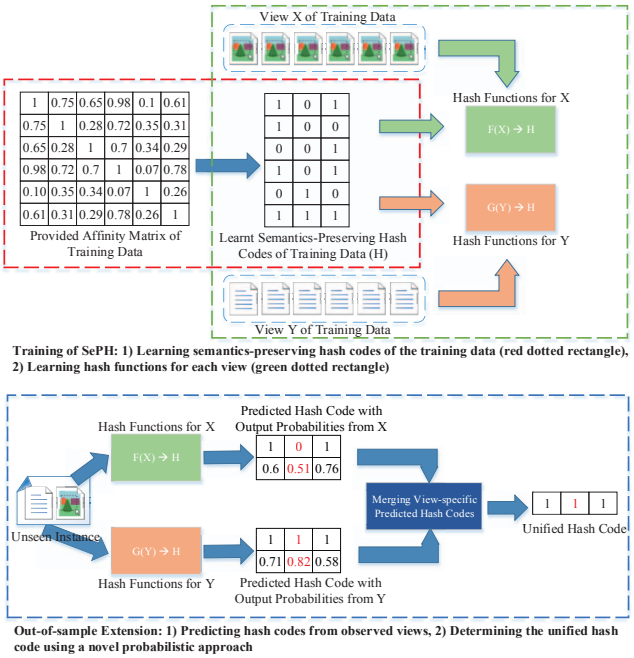

Figure 1: Framework of SePH, illustrated with two-view toy data.

		16 bits	32 bits	64 bits	128 bits
Image Query *v.s.* Text Database	CMSSH	0.4063	0.3927	0.3939	0.3739
	CVH	0.3687	0.4182	0.4602	0.4466
	IMH	0.4187	0.3975	0.3778	0.3668
	LSSH	0.3900	0.3924	0.3962	0.3966
	CMFH	0.4267	0.4229	0.4207	0.4182
	KSH-CV	0.4229	0.4162	0.4026	0.3877
	SCM-Orth	0.3787	0.3668	0.3593	0.3520
	SCM-Seq	0.4842	0.4941	0.4947	0.4965
	SePH$_{rnd}$	0.5394	0.5454	0.5499	0.5556
	SePH$_{km}$	**0.5421**	**0.5499**	**0.5537**	**0.5601**
Text Query *v.s.* Image Database	CMSSH	0.3874	0.3849	0.3704	0.3699
	CVH	0.3646	0.4024	0.4339	0.4255
	IMH	0.4053	0.3892	0.3758	0.3627
	LSSH	0.4286	0.4248	0.4248	0.4175
	CMFH	0.4627	0.4556	0.4518	0.4478
	KSH-CV	0.4088	0.3906	0.3869	0.3834
	SCM-Orth	0.3756	0.3641	0.3565	0.3523
	SCM-Seq	0.4536	0.4620	0.4630	0.4644
	SePH$_{rnd}$	0.6230	0.6331	0.6407	0.6489
	SePH$_{km}$	**0.6302**	**0.6425**	**0.6506**	**0.6580**

Table 1: Cross-view retrieval performance of the proposed SePH (*i.e.* SePH$_{rnd}$ and SePH$_{km}$) and compared baselines on NUS-WIDE with different hash code lengths, in terms of *mAP*.

With benefits of low storage costs and high query speeds, hashing methods are widely researched for efficiently retrieving large-scale data, which commonly contains multiple views, *e.g.* a news report with images, videos and texts. In this paper, we study the problem of cross-view retrieval and propose an effective **Se**mantics-**P**reserving **H**ashing method, termed SePH. Actually, cross-view retrieval is becoming popular in recent years, which can use one view of a query to retrieve relevant data presented in different views, *e.g.* retrieving relevant videos/texts using a query image.

The proposed SePH is supervised, with its framework illustrated in Fig. 1. Note that SePH learns one unified hash code for all views of an instance, rather than learning different hash codes for different views, which can further reduce storage costs.

For training, SePH employs a two-step hash learning process, *i.e.* firstly learning semantics-preserving hash codes of training instances and then learning hash functions for each view to project features into hash codes. Specifically, given the semantic affinities of training data as the supervised information, SePH firstly transforms them into a probability distribution $\mathcal{P}$ and approximates it in Hamming space, via transforming all pairwise Hamming distances between to-be-learnt hash codes into another probability distribution $\mathcal{Q}$ and minimizing its KL-divergence from $\mathcal{P}$. Unlike previous work that utilizes the supervised information for *independently* weighting each pairwise distance/similarity between hash codes, SePH standardizes all Hamming distances by transforming each into a probability that *depends* on all others. And thus correlations between Hamming distances are incorporated for forcing the to-be-learnt hash codes to better preserve the semantic structure of training instances. The objective function of learning semantics-preserving hash codes of training instances is as follows.

$$\Psi = \min_{\hat{H}} D_{KL}(\mathcal{P}\|\mathcal{Q}) + \alpha R(\hat{H}) \quad s.t. \quad \mathcal{Q} \leftarrow \hat{H} \tag{1}$$

where $\hat{H}$ is the relaxed real-valued hash code matrix, $R(\hat{H})$ denotes the quantization loss from $\hat{H}$ to the corresponding binary hash code matrix H, α is a weighting parameter, and $\mathcal{Q} \leftarrow \hat{H}$ means that $\mathcal{Q}$ is derived from $\hat{H}$. The objective function above is non-convex, and by applying gradient descent methods, we can derive a locally optimal $\hat{H}$ and then the binary hash code matrix of training instances, *i.e.* H.

With learnt semantics-preserving hash codes of training data, the learning of hash functions is open for any effective predictive models, like linear regression, SVM, logistic regression, *etc*. In this paper, SePH utilizes kernel logistic regression in each view for modelling the projections from the corresponding features to the hash codes, which can naturally provide output probabilities with the predicted results.

For out-of-sample extension, given any unseen instance, SePH firstly predict view-specific hash codes as well as their corresponding output probabilities from each observed view, using the corresponding learnt kernel logistic regressions. Then its unified hash code $\mathbf{c}$ is determined using a novel probabilistic approach. Specifically, with derivations applying Bayes' theorem, for each bit $\mathbf{c}_k$ of the unified hash code $\mathbf{c}$, its binary value (-1 or 1) is determined using the following formula.

$$\mathbf{c}_k = sign\Big(\prod_{i=1}^{m} p(\mathbf{c}_k = 1|\mathbf{z}^i) - \prod_{i=1}^{m} p(\mathbf{c}_k = -1|\mathbf{z}^i)\Big) \tag{2}$$

where $m \geq 1$ is the number of observed views, $\mathbf{z}^i$ is the ith view, and all needed probabilities like $p(\mathbf{c}_k = 1|\mathbf{z}^i)$ and $p(\mathbf{c}_k = -1|\mathbf{z}^i)$ are provided by the learnt kernel logistic regressions.

Extensive experiments conducted on benchmark Wiki, MIRFlickr and NUS-WIDE well demonstrate the effectiveness of SePH. Here in Table 1 we present the experimental results of the largest NUS-WIDE.

To conclude, the contributions of our work can be summarized as follows. 1) We propose an effective supervised cross-view hashing method termed SePH, which transforms the semantic affinities of training data into a probability distribution and approximates it with to-be-learnt hash codes in Hamming space via minimizing the KL-divergence. 2) We propose a novel probabilistic approach for determining the unified hash code of any unseen instance, using predicted hash codes as well as the corresponding output probabilities from different observed views.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.