Generalized Semantic Preserving Hashing for n-Label Cross-Modal Retrieval

Devraj Mandal Kunal N. Chaudhury Soma Biswas
Indian Institute of Science, Bangalore - 560012
{devraj89,kunal,soma.biswas}@ee.iisc.ernet.in

Abstract

Due to availability of large amounts of multimedia data, cross-modal matching is gaining increasing importance. Hashing based techniques provide an attractive solution to this problem when the data size is large. Different scenarios of cross-modal matching are possible, for example, data from the different modalities can be associated with a single label or multiple labels, and in addition may or may not have one-to-one correspondence. Most of the existing approaches have been developed for the case where there is one-to-one correspondence between the data of the two modalities. In this paper, we propose a simple, yet effective generalized hashing framework which can work for all the different scenarios, while preserving the semantic distance between the data points. The approach first learns the optimum hash codes for the two modalities simultaneously, so as to preserve the semantic similarity between the data points, and then learns the hash functions to map from the features to the hash codes. Extensive experiments on single label dataset like Wiki and multi-label datasets like NUS-WIDE, Pascal and LabelMe under all the different scenarios and comparisons with the state-of-the-art shows the effectiveness of the proposed approach.

1. Introduction

Availability of large volumes of multimedia data have made cross-modal retrieval tasks very important in the field of computer vision. For example, given a text query, we may want to retrieve all semantically similar images from the database. More often than not, single labels are often not sufficient to explain the data, and thus the data are usually annotated with multiple labels (Figure 1). We notice that though the images are not entirely same, they have some common labels signifying that there exists varying amounts of similarity between them. Also, the data (single or multi-label) may be paired or unpaired. For example, for single label data, say 10 images and 5 text data belong to the same category "elephant", which means that the image and text data cannot be paired. Thus cross-modal retrieval tasks can be roughly categorized into the following - (1) single label paired (SL-P), where there is one-to-one correspondence between the data of the two modalities, (2) single label unpaired (SL-U) where such pairing is absent, (3) multi-label paired (ML-P) where the multi-label data is given in pairs with same labels and (4) multi-label unpaired (ML-U) where the number of multi-label data items are different in both the modalities (Figure 2).

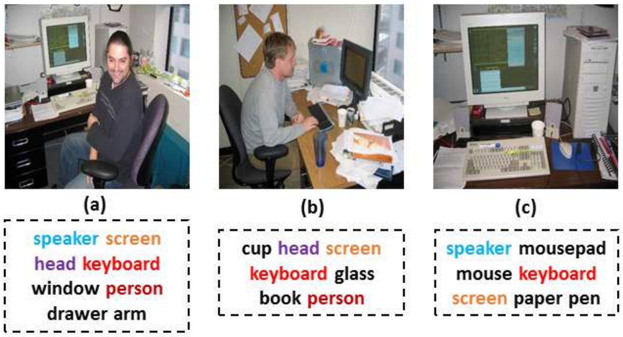

Figure 1. Few examples from the LabelMe dataset [18] shows that a single image usually requires multiple labels to properly describe them. Common tags are represented using the same color.

Most of the approaches in literature have focused on the paired scenarios (first and third problem) addressing both the supervised version [17] [20], where the labels are provided, and the unsupervised version, which does not contain the labels [7] [8]. Very few approaches have been proposed to handle the SL-U [17] and ML-U scenario [15]. Hashing techniques which can efficiently encode the input data into q-bit binary hash code have gained popularity, because of low storage costs and high query speeds, in addition to impressive cross-modal retrieval performance. Hashing techniques for both unsupervised [28] [21] [4] [27] and supervised settings [2] [11] [28] [24] for the SL-P and ML-P problems have been proposed. To the best of our knowledge, the task of SL-U and ML-U are yet to be addressed by the hashing based approaches.

In this work, we propose a simple, yet effective generalized hashing approach which can seamlessly handle all the scenarios described above, while preserving the structure and semantic relationships that exists within the data. Inspired by the success of the recent techniques [12], we propose a two stage hashing framework, which allows for less complex formulations and can be more easily solved in comparison to the coupled formulations [23]. In the first stage, we construct an affinity matrix by utilizing information such as labels or any other similarity measures provided, making the proposed approach a supervised one. The affinity matrix can be square or non-square depending on the availability of paired or unpaired data during the training stage. The algorithm first learns the optimum hash codes simultaneously for the two modalities by minimizing a non-convex optimization problem using alternating minimization techniques. For the second stage, though any binary classifier like support vector machine, deep learning network, etc. can be used to learn the hashing function, in this work, we use the kernel logistic regression for this purpose. We have also provided schemes for out-of-sample extensions and unifying the learned hash codes for the SL-P and ML-P settings. Extensive experiments on four image-text datasets, Wiki [16], NUS-WIDE [3], Pascal [5] and LabelMe [18] for all the different scenarios and comparisons with state-of-the-art cross-modal techniques shows the effectiveness of the proposed approach. The main contributions of the proposed work can be summarized as follows

1. We propose a generalized hashing scheme which can seamlessly handle the different scenarios like SL-P, SL-U, ML-P and ML-U in the same framework while preserving the semantic distance between the data.

2. To the best of our knowledge, this is the first work on hashing for handing the SL-U and ML-U task.

3. Extensive experiments show that the proposed approach compares favorably with respect to the state-of-the-art for all the scenarios.

The remainder of this paper is organized as follows. Section 2 gives an overview of the related works. Section 3 gives details of the proposed approach. The experiments are given in Section 4 and the paper concludes with a brief summary.

2. Related Work

Here we provide pointers to some of the related work in literature on the standard cross-modal techniques and the hashing based approaches.

Standard Cross-modal approaches : Several cross modal techniques like CCA [7] [8] has been developed which are very popular due to its simplicity and wide applicability for different tasks. As CCA [7] [8] is an unsupervised method, it is incapable of utilizing the labels for giving better retrieval performance. It can also work only in the SL-P setting. To facilitate this, CCCA [17] was developed which could use labels to work in the SL-U scenario. The presence of multi-label data provided the basis for the development of FCCA algorithm [15]. It was found in [15] that dealing with multi-label data in a single label setting greatly degrades the retrieval performance. In addition, if clusters are forcibly formed by either utilizing the k-means algorithm or by discarding all labels but one, to generate a single label setting, the performance further decreases [15].

Hashing based approaches: Unsupervised hashing methods [28] [21] [4] [27] uses the inter-modality and intra-modality information present in the training data to learn the hash codes. Inter-media hashing [21] tries to learn functions to map features from different modalities into the common hamming domain. Collective matrix factorization hashing (CMFH) [4] learns a single unified hash code for the different feature domains. Latent semantic sparse hashing (LSSH) [27] uses sparse coding and matrix factorization for image and for text representation respectively. It then maps them to a joint abstraction space to generate an unified hash code. [2] tries to maximize the similarity-agreement criterion to learn hash codes separately for both the modalities. Cross view hashing (CVH) [11] learns hash functions while minimizing the similarity-weighted hamming distances between the hash codes of training data. A probabilistic model for hashing has been proposed in [26]. Recently, very good results have been obtained by quantization techniques [22] [14] [25] for this task. CMSSH [2] and CVH [11] are few of the well known supervised methods which uses the labels. Semantic Preserving Hashing (SePH) [13] transforms the affinity matrix into a probability distribution and finds approximate hash codes while minimizing the Kullback-Leibler divergence. The hashing based approaches typically work in the SL-P and ML-P mode, where there is one-to-one correspondence between the data of the two modalities. Since the data is paired, many of the recent techniques learn a common hash code for both the modalities [14] [13], instead of different hash codes [2] [24]. Also, instead of solving a single objective function to learn both the optimal hash codes and the mapping functions together in a joint framework, the work done in [12] and [23] shows an alternative approach, in which a simpler optimization problem needs to be solved first to learn the hash codes. In the next stage, a set of binary classifiers are learned for either modalities to get the hashing functions.

Inspired by the success of the hashing based approaches, in this work, we propose a generalized hashing technique, which can handle both single and multi-label data, in both paired and unpaired setting. To the best of our knowledge, this is the first attempt in developing a hashing technique which can work in unpaired scenario.

3. The Proposed Approach

Now, we describe in details the proposed hashing framework. Let the two modalities be denoted as $X \in \mathcal{R}^{N_1 \times d_x}$ and $Y \in \mathcal{R}^{N_2 \times d_y}$, with N_1, N_2 being the number of items in either modality and d_x, d_y being the dimensionality of the data (in general $d_x \neq d_y$) respectively. The labels for both the modalities $L_x \in \mathcal{R}^{N_1 \times C}, L_y \in \mathcal{R}^{N_2 \times C}$ are provided, where C denotes the total number of categories. In case of single label data, only one of the C entries is one (eg. L_x^i =[0 0 1 0 0]), while for multi-label data, more than one entries will be equal to one, (eg. L_x^j =[1 0 1 0 1]). Cross-modal retrieval tasks can be categorized as follows:

- **Single Label-Paired (SL-P)**: Here, each data from one modality has a corresponding data in the other modality, i.e. $N_1 = N_2$, and each data belongs to one category. The affinity matrix S of size $N_1 \times N_2$ is constructed as $S_{ij} = 1$ if $L_x^i = L_y^j$, else $S_{ij} = 0$.
- **Single Label-Unpaired (SL-U)**: Here, though each data belongs to one category, pairing of data from the two modalities does not exist, and $N_1 \neq N_2$. Here S is constructed similar to SL-P.
- **Multi Label-Paired (ML-P)**: Here, each data from one modality has a corresponding data in the other modality, i.e. $N_1 = N_2$, but each data belongs to more than one category. Here S can constructed in several ways like (1) $S_{ij} = < L_x^i, L_y^j >$, where $< .,. >$ is the normalized inner product or as (2) $S_{ij} = e^{-||L_x^i - L_y^j||_2^2/\sigma}$, where σ is a constant factor.
- **Multi Label-Unpaired (ML-U)**: Here, each data belongs to multiple categories, but pairing of data from the two modalities does not exist, and $N_1 \neq N_2$. S is constructed as in ML-P.

Our objective is to find the optimal hash codes such that the similarity measure S that is computed is satisfied. In the next stage we use kernel logistic regression to learn the hash functions. A general outline of our procedure is shown in Figure 3.

3.1. Learning the Hash Code

We wish to factorize S as $(1/q)AB^T$, where the factors $A \in \mathcal{R}^{N_1 \times q}$ and $B \in \mathcal{R}^{N_2 \times q}$, N_1 and N_2 are the number of items in X and Y, and q is the length of the hash code. The rows in A (resp. B) are the hash codes for the items in X (resp. Y). Thus, we have the constraint that the elements of A and B should take values in $\{-1, 1\}$. Such a factorization might not exist, and therefore we consider the least squares problem (where $\|\cdot\|_F$ denotes the Frobenius norm):

$$\begin{aligned} \underset{A,B}{\text{minimize}} \quad & \|S - (1/q)AB^T\|_F^2 \\ \text{s.t.} \quad & A \in \{-1,1\}^{N_1 \times q}, \;\; B \in \{-1,1\}^{N_2 \times q}. \end{aligned} \tag{1}$$

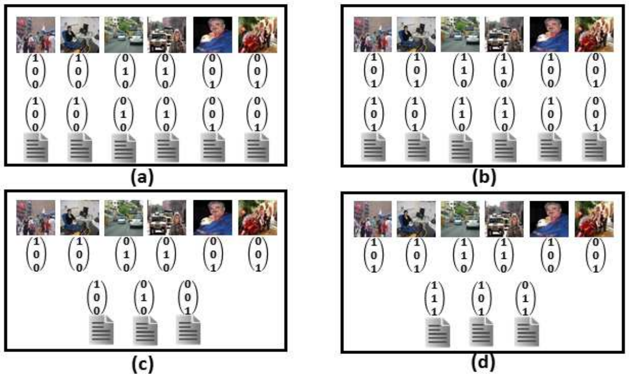

Figure 2. An illustrative example of the four scenarios (a) SL-P (b) ML-P (c) SL-U and (d) ML-U involving images and texts with their labels respectively. Observe that in SL-U and ML-U, the number of items are unequal in both the modalities.

The difficulty is that (1) is a discrete optimization problem, and is known to be computationally intractable [23]. A standard way around is to use some suitable relaxation [23]. In the present case, we replace the constraint set $\{-1, 1\}$ by its convex hull, namely the interval $[-1, 1]$. This gives us the following surrogate of (1):

$$\begin{aligned} \underset{A,B}{\text{minimize}} \quad & \|S - (1/q)AB^T\|_F^2 \\ \text{s.t.} \quad & A \in [-1,1]^{N_1 \times q}, \;\; B \in [-1,1]^{N_2 \times q}. \end{aligned} \tag{2}$$

At the end, we round the solution of (2) to obtain the desired binary solution by simply taking the sign of the matrix elements.

We note that, while the domain of (2) is convex, the objective is non-convex in variables A and B. Nevertheless, due to the bilinear nature of the factorization, the objective is convex in A if we hold B fixed, and vice-versa. Thus, if one of the variables is held fixed, then (2) becomes a convex optimization problem in the other variable. This naturally leads to the idea of alternating minimization [29], where the variables are alternately updated holding the other variable fixed. To further improve the computational efficiency, we propose to use coordinate descent on top of alternating minimization. In particular, for some fixed B, we update the elements of A in a sequential fashion [1]. While it is indeed possible to simultaneously update the elements of A using projected gradient descent [1], this would be computationally expensive given the matrix size. In contrast to this, we now demonstrate how the coordinate-descent updates can be performed analytically (and in a parallel fashion) using simple closed-form expressions.

Consider a single alternating minimization step in which one of the variables, say $B = [b_{ij}]$, is held fixed and we need to update $A = [a_{ij}]$. As mentioned previously, we wish to use coordinate descent for this purpose, whereby the elements of A are updated one at a time, say, in a raster fashion. In particular, suppose that we wish to update the element a_{il}. Notice that the objective in (2) can be expressed (up to a non-negative scaling) as

$$\sum_{j=1}^{N_2} \left(R_l^j + a_{il} b_{jl} \right)^2 + \text{ constant terms,}$$

where the constant terms do not depend on a_{il}, and

$$R_l^j = \sum_{k=1, k \neq l}^{q} a_{ik} b_{jk} - q S_{ij}.$$

Therefore, the coordinate descent with respect to a_{il} results in the subproblem

$$\min_{a_{il} \in [-1,1]} \quad \sum_{j=1}^{N_2} \left(R_l^j + a_{il} b_{jl} \right)^2 . \tag{3}$$

This is a convex problem, involving the minimization of a convex quadratic function over an interval. The unconstrained minimum of (3) is attained at

$$\hat{a}_{il} = - \frac{\sum_{j=1}^{N_2} R_l^j b_{jl}}{\sum_{j=1}^{N_2} b_{jl}^2}, \tag{4}$$

which is precisely the point where the gradient of the objective in (3) is zero. Since the objective is a quadratic function with positive curvature[1], it is not difficult to verify that the unique point where the minimum of (3) is attained is simply the projection of (4) onto $[-1,1]$. Namely, the minimum of (3) is attained at

$$a_{il}^* \quad = \quad \begin{cases} -1, & \text{if } \hat{a}_{il} < -1, \\ \hat{a}_{il}, & \text{if } \hat{a}_{il} \in [-1,1], \\ 1, & \text{if } \hat{a}_{il} > 1. \end{cases} \tag{5}$$

Notice that the denominator of (4) can be precomputed for each row update (during which i is fixed). Moreover,

$$R_{l+1}^j = R_l^j - a_{i\,l+1} b_{j\,l+1} + a_{il}^* b_{jl}. \tag{6}$$

This relation can be used to further speed up the update of successive elements on a given row. An identical strategy is used for updating B (holding A fixed). The entire process is summarized in Algorithm 1. The outer loop corresponds to alternating minimization, while the inner loop corresponds to coordinate updates. An important point to note is that we use just one pass of coordinate descent (one raster update). This is because we noticed that the final solution does not substantially change if we use multiple passes.

Algorithm 1 Alternative Minimization and Coordinate Descent for Hash Code Learning.

```
Input : $S$, $q$, maximum number of iterations $T$.
Initialize: Randomly generated $A$ and $B$.
for $t = 1, 2, ...., T$ do                    ▷ Alternating Minimization
    for $i = 1, 2, ...., N_1$ do              ▷ Coordinate Descent
        for $l = 1, 2, ...., q$ do
            Update $a_{il} \rightarrow a_{il}^*$ using (4) and (5).
    for $j = 1, 2, ...., N_2$ do              ▷ Coordinate Descent
        for $l = 1, 2, ...., q$ do
            Update $b_{jl} \rightarrow b_{jl}^*$ using (4) and (5).
Set $A = \text{sign}(A)$ and $B = \text{sign}(B)$
($\text{sign}(x) = 1$ if $x \geq 0$, and $= -1$ otherwise).
Output : $A \in \{-1,1\}^{N_1 \times q}$ and $B \in \{-1,1\}^{N_2 \times q}$.
```

3.2. Learning the Hash Functions

In this step, we learn the hash functions. Notice that all the bits of the hash codes are independently learned and thus in essence we need to design a bank of q binary classifiers which maps the input data X and Y to $\{-1, 1\}$. Here, we utilize the kernel logistic regression to learn the mappings from features to the hash codes for the input data. Kernel logistic regression exploits the power of kernels to effectively learn a non-linear mapping function. We use it to learn the functions F_X and F_Y independently for both domains. For clarity, we explain the procedure for X modality only.

In kernel logistic regression, kernels, i.e., non-linear functions enable us to map X_i to the Reproducing Kernel Hilbert Space (RKHS) as $\phi(X_i)$. The main objective is to now learn linear functions in the RKHS space which will enable us to go to the hash code domain. To learn the linear projection in RKHS for the lth bit ($1 \leq l \leq q$), we need to solve for w_x^l :

$$\min_{w_x^l} \quad \sum_{i=1}^{N_1} \log(1 + e^{-a_{il} \cdot \phi(X_i) . w_x^l}) + \lambda ||w_x^l||_2^2 \tag{7}$$

where λ is the parameter for the regularization term. For features coming from X, we need to learn the set of hash functions $F_X = \{w_x^1, w_x^2, ..., w_x^q\}$. Similar learning procedure in the Y domain will enable us to learn $F_Y = \{w_y^1, w_y^2, ..., w_y^q\}$. The objective function in (7) is solved by using the minFunc solver [19].

To generate the hash codes for the testing data coming from X or Y modality, compute the hash codes as $H_X = \text{sign}\,(F_X(X))$ and $H_Y = \text{sign}\,(F_Y(Y))$.

3.3. Generation of unified hash codes

In the proposed approach, we keep the option to unify the learned hash codes in settings where it is relevant i.e., the SL-P and ML-P case. Consider the lth bit in this case,

[1] We assume that $\sum_{j=1}^{N_2} b_{jl}^2 > 0$, which is always the case in practice.

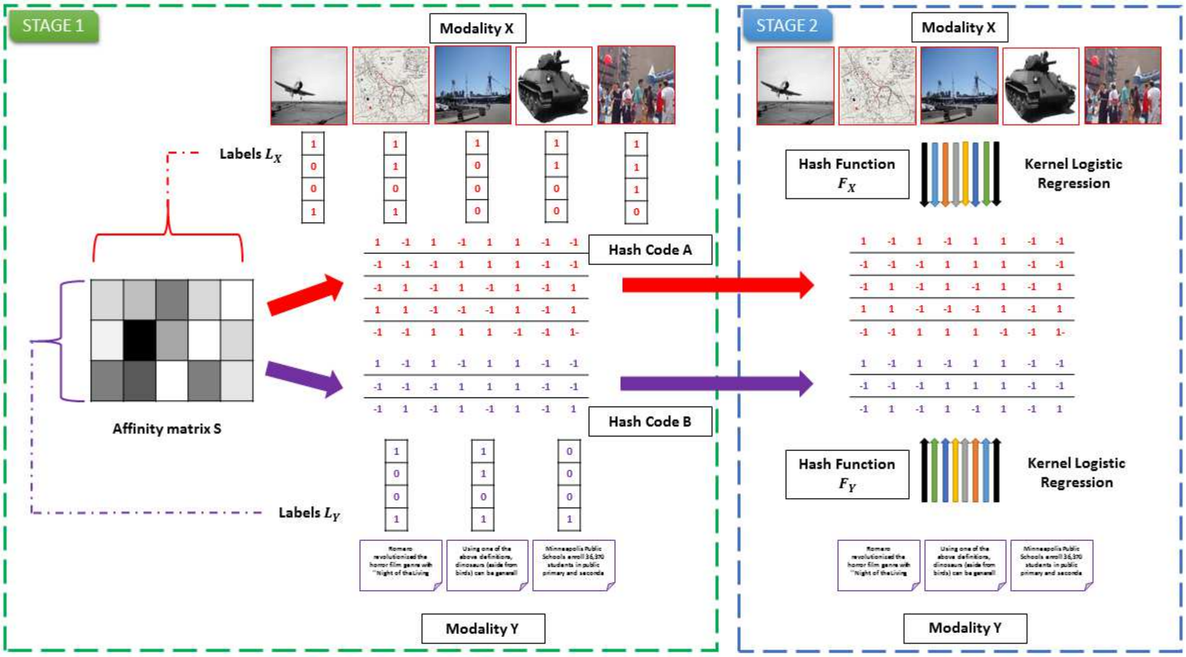

Figure 3. Flowchart of the proposed algorithm. In the first stage, we learn the hash codes from the affinity matrix S and in the next stage we learn the hashing functions.

which needs to be unified for a given data X_i and Y_i. From kernel logistic regression, we can determine the following terms $p(a_{il} = 1|X_i)$, $p(a_{il} = -1|X_i)$, $p(b_{il} = 1|Y_i)$ and $p(b_{il} = -1|Y_i)$. We use a parameter γ in a convex combination setting to combine the above probabilities to get the unified code $c_{il} = a_{il} = bil$ as shown in (8). The weight parameter γ enables us to put more importance to one modality compared to the other.

$$\begin{aligned} c_{il} &= \operatorname{sign}(\gamma\left(p(a_{il}=1|X_i) - p(a_{il}=-1|X_i)\right) \\ &+ (1-\gamma)\left(p(b_{il}=1|Y_i) - p(b_{il}=-1|Y_i)\right)) \end{aligned} \tag{8}$$

4. Experiments

Here, we report the results of extensive experiments performed to test the effectiveness of the proposed approach for all the four scenarios discussed. Specifically, we report results on Wiki [16], which is a single-label dataset, and NUS-WIDE [3], Pascal [5] and LabelMe [18] datasets, which are annotated with multiple labels. All the datasets considered here consist of image and text data, but this approach can be used seamlessly for other cross-modal data also. First, we give a brief description of the datasets with the features used and also the evaluation protocol.

4.1. Datasets and Evaluation Protocol

Wiki Dataset [16] consists of $2,866$ image-text pairs, with images encoded with 128-d SIFT descriptors and texts represented as 10-d topic vectors. The dataset is split into $2,173$ image-text pairs which are used as both the training and retrieval set and the other 693 pairs serves as the query set. Each image-text pair is assigned a single label out of possible 10 semantic classes.

NUS-WIDE Dataset [3] contains $269,648$ images with each image marked with relevant labels. Following the protocol in [13], data containing only the top 10 most popular labels (about $186,577$ pairs) are considered. The images are represented by 500-d bag-of-words features and texts are represented as 1000-d vectors of the most frequent labels. We use 4000 randomly sampled pairs as the query set and the rest as both the training and retrieval set.

LabelMe Dataset [18] consists of a total of 3825 images. For our experiments, following the protocol in [15], we have used bag of visual words, gist, color histogram and CNN features as image features and 209-d absolute tag rank as text features [9]. The ground-truth annotation of the images are used as the labels. A random $50-50$ split is performed to generate the training and testing sets as in [15].

Pascal Dataset [5] consists of 5011 train and 4952 test images. For this dataset also, we follow the protocol as in [15], and use the same features, with the train-test split provided originally.

For evaluation, we follow different performance measures while reporting the results, based on the different scenarios. For comparison against standard hashing techniques for SL-P and ML-P scenario, we report the Mean Average Precision (MAP), i.e., the mean of the average precision of all the queries. Average precision is defined as $AP(q) = \frac{\sum_{r=1}^{R} P_q(r)\delta(r)}{\sum_{r=1}^{R}\delta(r)}$, where R is the number of re-

Table 1. Comparison of cross-view retrieval performance (MAP) of the proposed approach with the state-of-the-art on Wiki [16] dataset for SL-P scenario with different hash code lengths (q). Best results are marked in bold.

	Image-to-Text				Text-to-Image			
	q=16	q=32	q=64	q=128	q=16	q=32	q=64	q=128
CMSSH	0.187	0.177	0.164	0.155	0.163	0.161	0.153	0.151
CVH	0.125	0.121	0.121	0.117	0.118	0.103	0.102	0.099
IMH	0.157	0.157	0.156	0.165	0.146	0.131	0.129	0.130
LSSH	0.214	0.221	0.221	0.221	0.503	0.522	0.529	0.534
CMFH	0.213	0.225	0.236	0.241	0.488	0.513	0.526	0.537
KSH-CV	0.196	0.183	0.170	0.166	0.171	0.166	0.169	0.157
SCM_Orth	0.159	0.146	0.138	0.113	0.155	0.138	0.126	0.109
SCM_Seq	0.221	0.233	0.244	0.259	0.213	0.236	0.247	0.257
SePH_rnd	0.276	0.296	0.304	0.313	0.631	0.658	0.663	0.669
SePH_knn	**0.278**	**0.295**	**0.306**	**0.313**	0.631	0.657	0.664	0.670
Ours_rnd	0.274	0.290	0.300	0.307	0.645	**0.663**	0.669	**0.674**
Ours_knn	**0.278**	0.291	0.301	0.304	**0.646**	**0.663**	**0.670**	**0.674**

trieved items and $P_q(r)$ is the precision at position r for query q. $\delta(r)$ is set to 1 if the r^{th} retrieved item has the same label or shares at-least one label with query q, else it is set to 0. For comparison against the standard cross-modal techniques, the proposed approach is evaluated using two performance metrics, namely, normalized discounted cumulative gain C@K and Precision P@K [15]. P@K corresponds to the number of relevant results in the first K retrieved data, but do not give emphasis on the rank order within the top-K items. C@K uses graded (instead of binary) relevance and puts more emphasis on the rank order of the correctly retrieved items within the top-K results.

4.2. Single Label-Paired (SL-P) Evaluation

Here, we evaluate the proposed approach on the single-labeled Wiki dataset [16] and provide comparisons with the state-of-the-art hashing techniques developed specifically for this scenario. The retrieval performance (MAP) for the proposed approach is reported in Table 1. Since, this evaluation protocol has paired setting, the reported results for the proposed algorithm is using the unified hash code as in [13]. Following the same protocol as in [13], while learning the hash function using Kernel Logistic Regression, we utilize both random sampling and k-means clustering and report both the results. We compare with both state-of-the-art supervised approaches, namely CMSSH [2], CVH [11], KSH-CV [28], SCM [24] and SePH [13] and the unsupervised approaches, namely IMH [21], LSSH [27] and CMFH [4]. The results of all the other approaches have been taken directly from [13]. We observe that for text-to-image setting, the proposed approach performs better than all the other approaches, where-as for image-to-text, it is only second to SePH, while being significantly better than the other techniques. Also, as expected, with the increase in the hash code length, the MAP score increases monotonically.

Table 2. Comparison of cross-view retrieval performance (MAP@50) of the proposed approach with the state-of-the-art on NUS-WIDE [3] for ML-P scenario with different hash code lengths (q). Best results are marked in bold.

	Image-to-Text				Text-to-Image			
	q=16	q=32	q=64	q=128	q=16	q=32	q=64	q=128
CMSSH	0.524	0.521	0.521	0.481	0.417	0.425	0.418	0.420
CVH	0.535	0.525	0.501	0.470	0.560	0.543	0.516	0.482
MLBE	0.447	0.454	0.470	0.402	0.435	0.488	0.502	0.442
QCH	0.509	0.527	0.520	0.513	0.509	0.517	0.509	0.508
LSSH	0.536	0.552	0.567	0.572	0.635	0.663	0.682	0.692
CMFH	0.474	0.482	0.513	0.506	0.510	0.564	0.589	0.594
CMCQ	0.563	0.590	0.599	**0.609**	0.689	0.708	0.719	0.725
SePH	-	0.586	0.601	0.607	-	0.726	0.746	0.746
Ours	**0.608**	**0.597**	**0.602**	0.607	**0.747**	**0.755**	**0.755**	**0.772**

4.3. Multi Label-Paired (ML-P) Evaluation

For the ML-P scenario, we evaluate the proposed approach on three different datasets, NUS-WIDE [3], LabelMe [18] and Pascal [5].

Table 2 shows the performance of the proposed approach on the NUS-WIDE [3] dataset using MAP@50 as the evaluation metric. In this case also, as in [13], we use unified hash code as the input training data is paired in nature. For this dataset, we use the evaluation protocol in [25] so that we can also compare with the recent, very popular quantization approaches (though it works in unsupervised setting). The results of all the other approaches have been taken directly from [25]. We observe from Table 2 that methods like CMFH [4] typically perform worse than methods like SePH [13]. The probable reason for this is that CMFH [4], in addition to being an unsupervised method, considers only the pairwise correspondence between the multi-label data while building the hash codes, whereas techniques like SePH [13] and the proposed approach uses all possible relationships between the data to do the same. We observe that the proposed approach in general outperforms both the state-of-the-art supervised as well as the unsupervised approaches.

For evaluating the hashing techniques, usually the training set consisting of both the image and text data is used as the database from which data is retrieved during testing, while the query is an unknown text or image. There is also another evaluation criteria where the testing data is completely unseen during training, and the retrieval is strictly cross-modal in the sense that given a text query, the task is to retrieve semantically similar image data or vice versa. Here the image and text data used for testing is not present during training. This protocol allows us to evaluate the generalizability of the proposed approach for unseen data and also test the performance for strictly cross-modal setting. Thus, in addition to the above experiments, we perform two additional experiments using this setting on the Pascal [5] and LabelMe [18] datasets, which are both multi-label and compare with some state-of-the-art algorithms. In this set-

Table 3. Performance of CCA [7] [8], CCCA [17], 3-CCA [6], FCCA [15], SePH [13] and the proposed algorithm for the LabelMe [18] dataset. Code length for the hashing based algorithms is 128. Two performance metrics C@30 and P@10 (in brackets) have been evaluated for four different image features for both the image-text and text-image retrieval tasks. Best results are highlighted in bold.

	Image-to-Text				Text-to-Image			
	Bow	Color	Gist	CNN	Bow	Color	Gist	CNN
	C@30(P@10)	C@30(P@10)	C@30(P@10)	C@30(P@10)	C@30(P@10)	C@30(P@10)	C@30(P@10)	C@30(P@10)
CCA	55.2 (38.9)	47.6 (36.0)	53.5 (41.2)	55.8 (41.8)	55.0 (43.1)	51.2 (41.1)	55.1 (42.9)	56.8 (42.1)
3-CCA	42.2 (36.5)	45.7 (30.1)	47.7 (37.6)	54.8 (41.4)	58.2 (43.4)	56.1 (45.6)	54.6 (45.1)	61.7 (43.4)
CCCA	50.2 (36.9)	42.2 (31.2)	43.6 (36.6)	57.4 (39.6)	53.1 (40.8)	48.2 (38.3)	47.2 (38.9)	56.9 (43.1)
FCCA	58.1 (40.8)	47.8 (36.5)	54.0 (40.8)	58.8 (43.1)	61.6 (49.6)	55.3 (43.0)	58.1 (45.2)	61.8 (46.4)
SePH-128	54.0 (41.6)	48.1 (35.2)	50.9 (39.8)	51.5 (36.0)	54.4 (43.1)	49.3 (38.4)	54.1 (41.9)	52.6 (38.1)
Ours-128	**65.0 (49.0)**	**57.6 (44.9)**	**60.6 (48.5)**	**67.2 (50.2)**	**65.7 (54.0)**	**61.6 (51.0)**	**63.2 (51.5)**	**69.2 (55.7)**

Table 4. Performance of CCA [7] [8], CCCA [17], 3-CCA [6], FCCA [15], SePH [13] and the proposed algorithm for the Pascal [5] dataset. Code length for the hashing based algorithms is 128. Two performance metrics C@30 and P@10 (in brackets) have been evaluated for three different image features for both the image-text and text-image retrieval tasks. Best results are highlighted in bold.

	Image-to-Text			Text-to-Image		
	Bow	Color	Gist	Bow	Color	Gist
	C@30 (P@10)	C@30 (P@10)	C@30 (P@10)	C@30 (P@10)	C@30 (P@10)	C@30 (P@10)
CCA	30.1 (26.5)	23.8 (22.6)	31.6 (27.9)	43.6 (43.8)	29.9 (31.8)	42.7 (41.9)
3-CCA	24.2 (24.0)	29.0 (24.2)	30.3 (25.0)	35.2 (31.8)	26.9 (28.1)	39.8 (42.4)
CCCA	27.3 (23.7)	24.2 (22.3)	27.7 (24.1)	37.6 (36.8)	28.8 (28.4)	36.3 (36.8)
FCCA	32.1 (28.2)	26.1 (23.0)	34.0 (29.2)	47.2 (46.1)	32.3 (30.6)	43.4 (43.2)
SePH-128	38.1 (35.1)	31.9 (29.6)	39.8 (36.8)	50.6 (52.3)	35.9 (36.2)	50.6 (49.0)
Ours-128	**38.6 (37.1)**	**32.4 (31.5)**	**41.2 (38.7)**	**52.5 (55.1)**	**37.2 (37.8)**	**52.8 (53.0)**

ting, unified hash codes cannot be used for retrieval since during testing, both the query and the database consists of data from a single modality.

For both the datasets, the results of the proposed approach using Bow, Color and Gist features are given in Table 3 and Table 4. Comparison with CCA [7], cluster-CCA [17], 3-view CCA [6] and FCCA [15] for both text-to-image and image-to-text cross-modal tasks are also reported. The results of the other approaches are directly taken from [15]. For both these tasks, the relevance of any retrieved object is decided based on the similarity between the labels of query and retrieved item. We use both C@30 and P@10 as the performance measure. For the LabelMe [18] dataset, we have also used the convolutional neural network (CNN) provided in [10] to extract image features. We follow the same procedure as in [15] for extracting the CNN features. Unfortunately we could not replicate the results of CNN features for the Pascal dataset [5] as in [15] and so do not report those results here. We also compare with the state-of-the-art supervised hashing technique, SePH [13] for this evaluation protocol using the code provided by the authors. We experimented with different values of α parameter for the SePH algorithm [13] and report the best results here.

The best results for the LabelMe dataset [18] are obtained when CNN [10] features was used. We observe from Table 3 and Table 4 that the proposed algorithm gives superior performance compared to the state-of-the-art for all the features. The top-5 image retrieval results for some textual queries for the Pascal [5] dataset are shown in Figure 4. We observe that for this protocol also, the proposed approach performs better than the state-of-the-art supervised hashing technique SePH [13]. Thus the proposed approach works seamlessly for both the SL-P and ML-P scenarios.

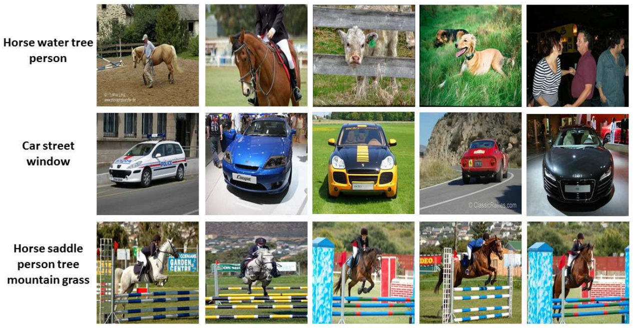

Figure 4. Top-5 image retrieval results for some textual queries for the Pascal [5] dataset.

4.4. Unpaired Scenario (SL-U , ML-U) Evaluation

Finally, we report the results of the proposed approach for unpaired scenarios, both for single and multi-label data (SL-U and ML-U). For the SL-U case, each data point is associated with a single label but there does not exist one-to-one correspondence between the data of the two modalities. As an example, say 10 images and 5 text documents are associated with the tag “building” in the dataset. Most of the algorithms developed for SL-P case are not applicable here, and to the best of our knowledge, no hashing techniques have been developed to handle this unpaired scenario.

For evaluation, we create the SL-P scenario by slightly modifying the experimental training protocol for the Wiki [16] dataset. The training set in one modality is kept same while in the other about 90% of it is retained (case 1) and vice versa (case 2). The training set itself serves as the re-

Table 5. Evaluation of the proposed algorithm in the SL-U mode for the Wiki [16] dataset. MAP@50 is reported with the best results highlighted in bold.

Method	CCA	CCCA	Ours		
			q=16	q=32	q=64
Image-to-Text					
Case1	0.1412	0.2219	0.2314	0.2591	**0.2797**
Case2	0.1486	0.2222	0.2172	0.2453	**0.2624**
Text-to-Image					
Case1	0.1886	0.3541	0.3385	0.5542	**0.6213**
Case2	0.1529	0.3467	0.4355	0.5662	**0.6265**

trieval set while the query set is kept same as in the SL-P case. We compare the proposed method against CCA [7] [8] and CCCA [17]. CCCA has been specifically developed to handle this scenario, and FCCA [15] reduces to CCCA in this situation. For implementing CCA [7] [8], we artificially construct paired training sets for learning the projection matrices. We observe from Table 5, that the proposed approach shows superior performance compared to the other approaches.

We also generate a similar scenario for the multi-label case (ML-U) by using the LabelMe [18] dataset. We use the Gist features for image representation. We use the whole training data in one modality and retain only 90% in the other modality (case 1) and vice versa (case 2) to create the training set. The testing sets remain the same as in ML-P case. We compare our approach with CCA [7] [8] and FCCA [15] for the P@10 metric. For CCA [7] [8] implementation, as before, we construct paired sets. The results in Table 6 shows superior performance of the proposed algorithm over the other baselines. We are not aware of any hashing techniques which can handle the ML-U scenario.

4.5. Analysis and Implementation details

Here we present some analysis of the proposed approach and some implementation details. While learning the hash functions using kernel logistic regression, we have used the radial basis function as the kernel, the numbers of samples was taken as 500 and the regularization parameter $\lambda = 0.01$. We have used the inner product and the exponential function ($\sigma = 1$) for construction of the affinity matrix S for the LabelMe [18] and Pascal [5] datasets respectively.

The performance of the proposed approach in terms of MAP@50 in given in Figure 5 for the Wiki [16] dataset for different number of training pairs. We report both the results before and after unification of the hash codes. For both the modalities, we observe that as the number of training pairs increases, the retrieval results get better. We also observe that the unified code gives a larger improvement in the retrieval performance in case of Text-to-Image, whereas for Image-to-Text, the results obtained are almost equal.

Table 6. Evaluation of the proposed algorithm in the ML-U mode for the LabelMe [18] dataset. P@10 is reported with the best results highlighted in bold.

Method	CCA	FCCA	Ours		
			q=32	q=64	q=128
Image-to-Text					
Case1	36.66	45.66	43.34	44.23	**46.69**
Case2	37.42	43.90	42.52	44.33	**46.53**
Text-to-Image					
Case1	43.82	46.37	49.82	50.62	**52.37**
Case2	42.80	45.47	47.47	50.35	**52.26**

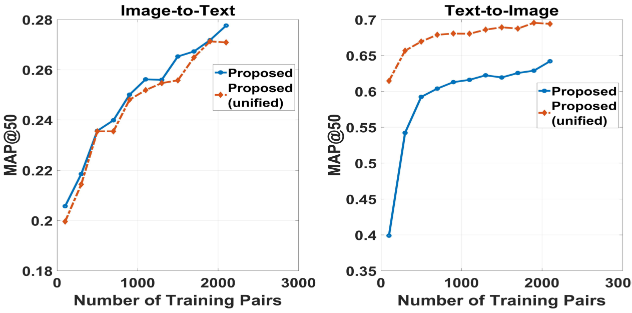

Figure 5. MAP@50 for the Wiki [16] dataset for different number of training pairs. The hash code length is taken as 16 for this experiment.

5. Conclusion

This paper proposes a generalized hashing approach for cross-modal retrieval tasks which can work in multiple settings like single label, multi-label, and both paired and unpaired scenario, while preserving the semantic similarity between the data points. By dividing the entire procedure into two steps - one learning the optimal hash codes and the other learning the hash functions, we gain a two-fold advantage. In the first step, we need to optimize a simple non-convex problem by using alternating minimization technique. The second step ideally can be tailored to suit the user's needs, for example, by fine-tuning a trained deep network to learn more complicated hash functions for the image and text domain data. We also keep an option to unify the learned hash codes at the end of the algorithm. To the best of our knowledge, this is the first time a hashing approach has been used for unpaired scenario. Extensive experiments on several datasets shows the effectiveness of the proposed approach for all the different scenarios. In future, we intend to learn hash codes that reflect both the intra-modality and inter-modality relationships.

Acknowledgements

The authors would like to thank Mr. Viresh Ranjan (Stony Brook University) for useful discussions during the progress of this work.

References

[1] D. P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific Belmont, 2015.

[2] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, volume 1, pages 3594–3601, 2010.

[3] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *ACM-CIVR*, pages 1–9, 2009.

[4] G. Ding, Y. Guo, and J. Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR*, pages 2083–2090, 2014.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[6] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 106(2):210–233, 2014.

[7] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[8] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[9] S. J. Hwang and K. Grauman. Accounting for the relative importance of objects in image retrieval. In *BMVC*, volume 1, pages 58.1–12, 2010.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[11] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, volume 22, pages 1360–1365, 2011.

[12] G. Lin, C. Shen, D. Suter, and A. van den Hengel. A general two-step approach to learning-based hashing. In *ICCV*, pages 2552–2559, 2013.

[13] Z. Lin, G. Ding, M. Hu, and J. Wang. Semantics-preserving hashing for cross-view retrieval. In *CVPR*, pages 3864–3872, 2015.

[14] M. Long, J. Wang, and P. S. Yu. Compositional correlation quantization for large-scale multimodal search. *arXiv preprint arXiv:1504.04818*, 2015.

[15] V. Ranjan, N. Rasiwasia, and C. Jawahar. Multi-label cross-modal retrieval. In *ICCV*, pages 4094–4102, 2015.

[16] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM-MM*, pages 251–260, 2010.

[17] N. Rasiwasia, D. Mahajan, V. Mahadevan, and G. Aggarwal. Cluster canonical correlation analysis. In *AISTATS*, pages 823–831, 2014.

[18] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[19] M. Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab, 2005.

[20] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *CVPR*, pages 2160–2167, 2012.

[21] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*, pages 785–796, 2013.

[22] B. Wu, Q. Yang, W.-S. Zheng, Y. Wang, and J. Wang. Quantized correlation hashing for fast cross-modal search. In *IJCAI*, pages 25–31, 2015.

[23] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, volume 1, pages 2156–2162, 2014.

[24] D. Zhang and W. J. Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, volume 1, pages 2177–2183, 2014.

[25] T. Zhang and J. Wang. Collaborative quantization for cross-modal similarity search. In *CVPR*, pages 2036–2045, 2016.

[26] Y. Zhen and D.-Y. Yeung. A probabilistic model for multimodal hash function learning. In *SIGKDD*, pages 940–948, 2012.

[27] J. Zhou, G. Ding, and Y. Guo. Latent semantic sparse hashing for cross-modal similarity search. In *SIGIR*, pages 415–424, 2014.

[28] J. Zhou, G. Ding, Y. Guo, Q. Liu, and X. Dong. Kernel-based supervised hashing for cross-view similarity search. In *ICME*, pages 1–6, 2014.

[29] X. Zhou, M. Zhu, and K. Daniilidis. Multi-image matching via fast alternating minimization. In *ICCV*, pages 4032–4040, 2015.