Predictable Dual-View Hashing

Mohammad Rastegari MRASTEGA@CS.UMD.EDU
Jonghyun Choi JHCHOI@UMD.EDU
Shobeir Fakhraei SHOBEIR@CS.UMD.EDU
Hal Daumé III ME@HAL3.NAME
Larry S. Davis LSD@UMIACS.UMD.EDU
University of Maryland, College Park, MD 20742 USA

Abstract

We propose a Predictable Dual-View Hashing (PDH) algorithm which embeds proximity of data samples in the original spaces. We create a cross-view hamming space with the ability to compare information from previously incomparable domains with a notion of 'predictability'. By performing comparative experimental analysis on two large datasets, PASCAL-Sentence and SUN-Attribute, we demonstrate the superiority of our method to the state-of-the-art dual-view binary code learning algorithms.

1. Introduction

Binary codes are attractive representations of data for search and retrieval purposes due to their efficiency in computation and storage capacity. For example, 64-bits binary codes can index about 10^{19} images, five times the estimated amount of data created in 2002 and quite likely the total number of digital images in existence (Lyman et al., 2003).

Hashing is a common method for assigning binary codes to data points (*e.g.*, images). The binary codes are used as hash keys where the hash functions are learned to preserve some notion of similarity in the original feature space. Such binary codes should have the general hash property of low collision rates. In addition, suitable binary codes for search and retrieval should also maintain high collision rates for similar data points. The latter property is essential in a similarity based retrieval settings (Gionis et al., 1999b; Gong & Lazebnik, 2011; Weiss et al., 2008).

The binary codes can be learned either in a unsupervised manner that models the distribution of samples in the feature space (Weiss et al., 2008) or in a supervised manner that uses labels of the data points (Liu et al., 2012). Unsupervised methods can be adversely affected by outliers in distributions and noise, and the supervised methods require expensive manual labeling.

It is often the case that information about data are available from two or more views, *e.g.*, images and their textual descriptions. It is highly desirable to embed information from both domains in the binary codes, to increase search and retrieval capabilities. Utilization of such binary codes will create a cross-view Hamming space with the ability to compare information from previously incomparable domains. For example in the text and image domain, image-to-image, text-to-image, and image-to-text comparisons can be preformed in the same cross-view space. Such approaches have received attention recently due to the emergence of large amounts of data in different domains being available on the internet.

To date, most approaches proposed embedding dual-views in Hamming space use canonical correlation analysis (CCA) (Hardoon et al., 2003; Hwang & Grauman, 2010; 2012). The CCA based approaches are less sensitive to feature noise and require no manual labeling. However, bits learned by CCA do not explicitly encode the proximity of samples in the original feature space since CCA enforces orthogonal bases and aims to reduce the modality gap with little consideration of the underlying data distribution.

To address this issue, we propose a dual-view mapping algorithm that represents the distribution of the samples with non-orthogonal bases inspired by a notion of *predictability* proposed in (Rastegari et al., 2012). Predictable codes ensure that small variations of the data point positions in the original space should not result in different binary codes. In other words, a particular bit in the binary code should be identical (predictable) for all data samples that are close to each other in each view. To maintain such predictability, we employ a max-margin formulation that enforces confident pre-

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

diction of bits.

Furthermore, we propose a joint formulation for learning binary codes of data from two different views. We assume that a latent Hamming space exists for the data, and optimize the hash functions that map the data from each view to this common space, while maintaining the predictability of the binary codes. Knowing the hash functions in the original views supports cross-modal searches.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 presents the details of our approach including optimization methods. Experimental analysis and comparisons to state-of-the-art methods are presented in section 4 and we conclude in 5.

2. Related Work

As our work lies in the intersection of hashing methods and mutil-view embedding, we briefly describe related work in both domains. We also review specific applications that could be enabled via our method.

Gionis *et al.* (Gionis et al., 1999a) introduced Locality Sensitive Hashing (LSH) where similar objects have high probability of collision. Along this direction, Shaknarovich *et al.* (Shakhnarovich et al., 2003) use parameter sensitive hashing and apply it to human pose estimation. Kulis and Grauman (Kulis & Grauman, 2009) extend LSH with kernels and show fast image search for example-based searches and content based retrieval. Kulis and Darrell (Kulis & Darrell, 2009) also proposed a binary reconstructive embedding method for minimizing the differences between Euclidean distances in the original feature space and the Hamming distances in the resulting binary space.

Semantic hashing, proposed in (Salakhutdinov & Hinton, 2009), learns compact binary codes that preserve correlations between distances in the Hamming space and semantic similarities approximated by category memberships. This is accomplished by learning a deep generative model, called a Restricted Boltzman Machine (RBM) which has a small number of nodes in a deepest level that produce a small number of binary values. Torralba *et al.*(Torralba et al., 2008) extend this idea to efficient image search method on the scale of millions of images. Nonlinear mapping to binary codes has been addressed in (Salakhutdinov & Hinton, 2007) by stacking multiple RBM's. Norouzi and Fleet (Norouzi & Fleet, 2011) model the problem of supervised learning of compact similarity-preserving binary code using a Latent SVM problem and define a hashing-specific class of loss functions. None of these approaches, however, necessarily captures the semantics of an image. In fact, enforcing preservation of patterns in the original feature space may hurt discrimination in both supervised and unsupervised methods.

Utilization of textual captions for image understanding has recently received considerable attentions in the research community. Farhadi *et al.* (Farhadi et al., 2010) introduce a CRF based method to model a semantic space that text and images can be mapped to via triples of object, subject and verb. In (Rashtchian et al., 2010) strategies of creating image-text datasets via Amazon Mechanical Turk are investigated. Kulkarni *et al.* (Kulkarni et al., 2011) propose a method for generating natural language descriptions from images by parsing a large set of texts and performing object recognition on image sets. Li *et al.* (Li et al., 2011) propose a simple but effective N-gram based method that can produce simple descriptions of pictures. The generated descriptions are not identical to the text corpora, *i.e.*, they compose a sentence entirely from scratch. Recently, several works presented methods for Multi-Modal hashing (Masci et al., 2012; Zhen & Yeung, 2012; Kumar & Udupa, 2011); most of them having high computational complexity which limits their applicability.

Ordonez *et al.* (Ordonez et al., 2011) created a large-scale dataset of images and captions, and proposed a method for generating textual captions for images from this dataset. A method for recognition of visual texts and non-visual texts is proposed in (Dodge et al., 2012). Kuznetsova *et al.* (Kuznetsova et al., 2012) use multiple noisy captions for images from the web and combine them to produce a more meaningful sentence for an image. Berg *et al.* (Berg et al., 2012) approach the problem of text generation to emphasize the visually salient aspects of an image.

3. Our Approach

Without loss of generality, we assume that the two views are visual (image) and textual (description). However, our approach is applicable to any domain, and this assumption only facilitates the discussion.

We use the following notation; $X_{\mathcal{V}}$ represents data in the visual space and $X_{\mathcal{T}}$ indicates data in the textual space. X_* is a $d_* \times n$ matrix whose columns are vectors corresponding to the points in either spaces. d_* is the dimension of either visual or textual space which might be different. x_*^i is the i^{th} column of X_*. $*$ is a placeholder for $\mathcal{V}$ or $\mathcal{T}$.

3.1. Dual-View Embedding

Our goal is to find two sets of hyperplanes $W_{\mathcal{V}}, W_{\mathcal{T}} \in \mathbb{R}^{d_* \times k}$ (k is the dimension of the common subspace, *i.e.*, length of binary code) that map the visual and textual space into a common subspace. Each hyper-

plane (each column of W_*) divides the corresponding space into two subspaces; each point in a space is represented as **-1** or **1** depending on which side of the hyperplane it lies in. w_*^i indicates the i^{th} column of W_*. Among the infinite possible hyperplanes, the ones that binarize the points in the visual space and the textual space consistently are desirable for our purpose. This objective can be achieved by minimizing the following function:

$$\min_{W_\mathcal{V}, W_\mathcal{T}} \|\text{sgn}(W_\mathcal{V}^T X_\mathcal{V}) - \text{sgn}(W_\mathcal{T}^T X_\mathcal{T})\|_2^2 \tag{1}$$

However, Eq.(1) is a non-convex combinatorial optimization problem; it has a trivial solution when both $W_\mathcal{V}$ and $W_\mathcal{T}$ are zero. To avoid the trivial solution and force each bit to carry the maximum amount of information, we add constraints to enforce low correlation of the bits. With these constraints, we can reformulate the problem as:

$$\begin{aligned} &\min_{W_\mathcal{V}, W_\mathcal{T}} \|W_\mathcal{V}^T X_\mathcal{V} - B_\mathcal{T}\|_2^2 + \|B_\mathcal{T} B_\mathcal{T}^T - I\|_2^2 \\ &\quad + \|W_\mathcal{T}^T X_\mathcal{T} - B_\mathcal{V}\|_2^2 + \|B_\mathcal{V} B_\mathcal{V}^T - I\|_2^2 \\ &s.t. \\ &\quad B_\mathcal{T} = \text{sgn}(W_\mathcal{T}^T X_\mathcal{T}) \\ &\quad B_\mathcal{V} = \text{sgn}(W_\mathcal{V}^T X_\mathcal{V}) \end{aligned} \tag{2}$$

where minimizing $\|B_* B_*^T - I\|_2^2$ enforces low correlation of bits. This optimization cannot be directly solvable, but it can be solved approximately by relaxing B_* (Gong et al., 2012) and applying CCA (Hardoon et al., 2003), which leads to the following generalized eigenvalue problem:

$$\begin{pmatrix} S_{\mathcal{V}\mathcal{V}} & S_{\mathcal{V}\mathcal{T}} \\ S_{\mathcal{T}\mathcal{V}} & S_{\mathcal{T}\mathcal{T}} \end{pmatrix} \begin{pmatrix} w_\mathcal{V} \\ w_\mathcal{T} \end{pmatrix} = \lambda \begin{pmatrix} S_{\mathcal{V}\mathcal{V}} & 0 \\ 0 & S_{\mathcal{T}\mathcal{T}} \end{pmatrix} \begin{pmatrix} w_\mathcal{V}, \\ w_\mathcal{T}, \end{pmatrix} \tag{3}$$

where $S_{\mathcal{V}\mathcal{T}} (= X_\mathcal{V} X_\mathcal{T}^T)$ is the covariance matrix between visual and textual features and w_* is a column of W_*.

Although CCA can find the underlying subspace, binarizing data in this subspace by $\text{sgn}(W_*^T X_*)$ suffers from high quantization error. To reduce the quantization error, an iterative method is proposed in (Gong & Lazebnik, 2011) that searches for a rotation of data points. Their approach, however, is not applicable to more than one domain. In addition, the approach assumes orthogonality of all of the projected hyperplanes, *i.e.*, the columns of W_*. But the orthogonality is not always necessary and sometimes harmful. In contrast, we replace orthogonality of the hyperplanes by the notion of predictability of binary codes in the following section.

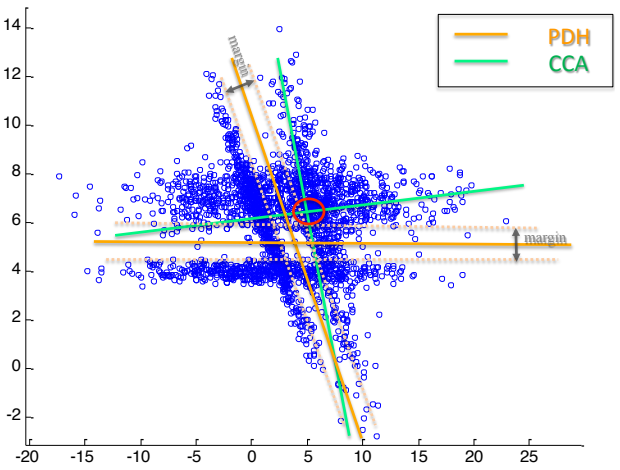

Figure 1. Comparison of learned hyperplanes by our method (PDH) and canonical correlation analysis (CCA). Note that the hyperplanes learned by the PDH divide the space, avoiding the fragmentation of sample distributions by the help of *predictability* constraints implemented by max-margin regularization.

3.2. Predictability

Predictability is the ability to predict the value of a certain bit of a sample by looking at that bit of the nearest neighbors of that sample. For example, if the i^{th} bit in most of the nearest neighbors of a sample is **1** then we would predict that the i^{th} bit of that sample would be also **1**.

Consider the situation where a hyperplane crosses a dense area of samples; there would be many samples in proximity to each other that are assigned different binary values in the bit position corresponding to that hyperplane. Such binary values obtained by that hyperplane are not *predictable*. Intuitively, the binary values determined by a hyperplane are *predictable* when the hyperplane has large margins from samples. Figure 1 illustrates the hyperplanes determined by CCA in green lines in a 2D single domain (view). Note that CCA hyperplanes cross dense areas of samples and are orthogonal to each other whereas our PDH hyperplanes do not. If we binarize the samples by CCA hyperplanes, samples in the red circle will have different binary codes from each other, even though they are strongly clustered. The hyperplanes that are shown by orange lines represent our method (PDH), which enforces large margins from samples.

To learn the predictable W, we regularize the formulation with max-margin constraints. In fact, we learn multiple SVMs in visual space with respect to training labels in the textual space and vice versa. The final

objective function is:

$$\min_{W_{\mathcal{V}},W_{\mathcal{T}},\xi_{\mathcal{V}},\xi_{\mathcal{T}}} \|B_{\mathcal{T}}B_{\mathcal{T}}^T - I\|_2^2 + \|B_{\mathcal{V}}B_{\mathcal{V}}^T - I\|_2^2 + \sum\|w_{\mathcal{V}i}\| + \sum\|w_{\mathcal{T}i}\| + C_1\sum\xi_{\mathcal{V}} + C_2\sum\xi_{\mathcal{T}}$$

$$s.t.$$

$$B_{\mathcal{T}} = \text{sgn}(W_{\mathcal{T}}^T X_{\mathcal{T}}),$$
$$B_{\mathcal{V}} = \text{sgn}(W_{\mathcal{V}}^T X_{\mathcal{V}}),$$
$$B_{\mathcal{T}}^{ij}({w_{\mathcal{V}i}}^T X_{\mathcal{V}}^j) \geq 1 - \xi_{\mathcal{V}}^{ij} \quad \forall i,j,$$
$$B_{\mathcal{V}}^{ij}({w_{\mathcal{T}i}}^T X_{\mathcal{T}}^j) \geq 1 - \xi_{\mathcal{T}}^{ij} \quad \forall i,j. \tag{4}$$

Despite the complex appearance of the optimization, it is a perfect setting for block-coordinate descent and can be solved by an Expectation Maximization (EM) iterative algorithm. A detailed description of our iterative algorithm is as follows:

First, we fix all the variables except $W_{\mathcal{V}}$ and $\xi_{\mathcal{V}}$. Then we solve for these variables, which is multiple linear SVMs; one for each bit. To learn the i^{th} SVM, we use columns of $X_{\mathcal{V}}$ as training data and the elements of the i^{th} row of $B_{\mathcal{T}}$ as training labels. **Second**, using the outputs of these SVMs, $W_{\mathcal{V}}$, we compute $B_{\mathcal{V}} = \text{sgn}({W_{\mathcal{V}}}^T X_{\mathcal{V}})$. **Third**, we update $B_{\mathcal{V}}$ to minimize the correlation between bits via minimizing $\|B_{\mathcal{V}}B_{\mathcal{V}}^T - I\|_2^2$. Since this problem is not trivial to solve, we use spectral relaxation (Weiss et al., 2008) by creating a Gram matrix $S = B_{\mathcal{V}}^T B_{\mathcal{V}}$ and a $n \times n$ diagonal matrix $D(i,i) = \sum_j S(i,j)$ as the relaxed problem:

$$\min_{B_{\mathcal{V}}} \text{tr}(B_{\mathcal{V}}(D-S){B_{\mathcal{V}}}^T) \quad s.t. \quad B_{\mathcal{V}}{B_{\mathcal{V}}}^T = I. \tag{5}$$

The solutions are the k eigenvectors of $D - S$ with minimal eigenvalues, which we binarize by taking the sign of the elements. **Fourth**, we run the same three steps to compute $W_{\mathcal{T}}$. We repeat all the steps until convergence of the objective function. More details of the algorithm are provided in Algorithm 1

For initializing values for optimization, we tried several random values and the values obtained using CCA. But the results are not sensitive to the initialization, since in each block coordinate descent step, the objective function is convex. Thus, we use the values obtained by CCA for all initializations.

Since our objective function is not convex and we use block coordinate descent to optimize, the solution we obtain is not the global minimum. But our experiments suggest that the obtained local minima is good enough.

Algorithm 1 Predictable Dual-View Hashing

Input: $X_{\mathcal{V}}$, $X_{\mathcal{T}} \in \mathbb{R}^{d_* \times n}$.
Output: $B_{\mathcal{V}}$, $B_{\mathcal{T}} \in \mathbb{B}^{d_* \times k}$.
$W_{\mathcal{V}}, W_{\mathcal{T}} \in \mathbb{R}^{d_* \times k} \leftarrow CCA(X_{\mathcal{V}}, X_{\mathcal{T}}, k)$
$B_{\mathcal{V}} \leftarrow \text{sgn}({W_{\mathcal{V}}}^T X_{\mathcal{V}})$
$B_{\mathcal{T}} \leftarrow \text{sgn}({W_{\mathcal{T}}}^T X_{\mathcal{T}})$
repeat
  $W_{\mathcal{V}} \leftarrow$ Weights of k linear SVMs (for i^{th} SVM: training features are columns of$X_{\mathcal{V}}$ and training labels are elements of i^{th} row of $B_{\mathcal{T}}$)
  $B_{\mathcal{V}} \leftarrow \text{sgn}({W_{\mathcal{V}}}^T X_{\mathcal{V}})$
  Update $B_{\mathcal{V}}$ using Eq. (5)
  $W_{\mathcal{T}} \leftarrow$ Weights of k linear SVMs (for i^{th} SVM: training features are columns of$X_{\mathcal{T}}$ and training labels are elements of i^{th} row of $B_{\mathcal{V}}$)
  $B_{\mathcal{T}} \leftarrow \text{sgn}({W_{\mathcal{T}}}^T X_{\mathcal{T}})$
  Update $B_{\mathcal{T}}$ using Eq. (5)
until convergence
$B_{\mathcal{V}} \leftarrow \text{sgn}({W_{\mathcal{V}}}^T X_{\mathcal{V}})$
$B_{\mathcal{T}} \leftarrow \text{sgn}({W_{\mathcal{T}}}^T X_{\mathcal{T}})$

4. Experiments

First, we show that our optimization algorithm solves the proposed objective functions. Then for the empirical validation, we present both quantitative and qualitative results for image category retrieval. In the quantitative analysis, we perform image classification and compare the mean average precision (mAP) obtained by our method with several state-of-the-art binary code methods. In qualitative analysis, we show that the sets of images retrieved by our binary code with both image and text queries contain semantically similar images. Our MATLAB software is available[1].

4.1. Datasets and Experimental Setup

For the dual-view situation, we need a dataset of images that are annotated with sentences. We use two datasets; PASCAL-Sentence 2008 introduced by (Farhadi et al., 2010) (one view is visual and the other is textual) and a recently collected large scale dataset, SUN-Attribute database (one view is visual and the other is semantic (attribute)) (Patterson & Hays, 2012).

4.1.1. PASCAL-Sentence Dataset 2008

The images in the PASCAL-Sentence dataset are collected from PASCAL 2008, which is one of the most popular benchmark datasets for object recognition and detection. For each of the 20 categories of the PASCAL 2008 challenge, 50 images are randomly selected; in total, there are 1,000 images in the dataset. Each image is annotated with 5 sentences using Amazon's Mechanical Turk. These sentences represent the se-

[1] http://umiacs.umd.edu/ mrastega/pdh/

mantics of the image.

Image Features: Our image features, following (Farhadi et al., 2010), are collections of responses from a variety of detectors, image classifiers and scene classifiers. Given an image, we run several object detectors on the image and set the threshold low enough so that each fires at least in one location. Then, we report the location of the most confident detector along with the confidence value. If we have 20 detectors, for each of the detectors we report $[x_i, y_i, c_i]$ which x_i,y_i are the coordinate of the location at which the detectors fired and c_i is the confidence value for that detector. Image and scene classifiers are SVMs trained on each category of objects on the global low-level GIST descriptor (Oliva & Torralba, 2001).

Text Feature: Text features are also from (Farhadi et al., 2010). We construct a dictionary of 1,200 words from the sentences of the entire dataset that are frequent and discriminative with respect to categories. There are no prepositions and stop words in the dictionary. Let us call this set S. For a given sentence, we go through each word and compute its semantic similarity with all the words in S as a feature for that word. As a feature of the sentence, we simply sum all the vectors in each sentence. The semantic distance between two words is computed by the Lin similarity measure (Lin, 1998) on the WordNet hierarchy.

4.1.2. SUN Attribute Dataset

The SUN-Attribute dataset is a large-scale dataset (Patterson & Hays, 2012) that includes 102 attribute labels annotated by 3 Amazon Mechanical Turk worker for each of the 14,340 images from 717 categories, which is a subset of the scene images from the SUN Dataset (Xiao et al., 2010). In total, there are four million (4M) labels. For each of 717 categories, there are 20 annotated scenes.

Image Features: We use the precomputed image features used in (Patterson & Hays, 2012; Xiao et al., 2010), *i.e.*, Gist, 2×2 Histogram of Oriented Gradient, self-similarity measure, and geometric context color histograms.

Attribute Features: Each image has 102 attributes and each attribute has multiple annotations. In total, there are four million labels that are annotated by Amazon Mechanical Turk workers with bad-worker filtering and good-worker cultivating strategies (Patterson & Hays, 2012). Some examples of annotated attributes are vegetation, open area, camping, hiking, natural light, leaves *etc.*

4.1.3. Experimental details

We use Liblinear (Fan et al., 2008) to learn SVMs for learning W_*. The parameters used for linear SVMs are $C_1 = 1$ and $C_2 = 1$ in Eq. 4. We did not tune those parameter. We also used linear SVM for category retrieval. We reduce the dimensionality of visual features in the SUN dataset from 19,080 to 1,000 by PCA.

4.2. Optimization Analysis

As we use a block coordinate descent algorithm to optimize the objective function, we cannot guarantee that our algorithm reaches the global optimum. Our experiments shows that we reach a reasonable local optimum most of the time. To illustrate this, we measure the objective value and see if it decreases (*in the minimization task*) or not. In figure 2, we observe that the objective values does decrease as the iterations go on. After only a few iterations (15) the differences between the textual binary codes (*binary codes extracted from text data*) and the visual binary codes (*binary code extracted from images*) are very small- less than 3 bits. The number of bits we use for this experiments is 32.

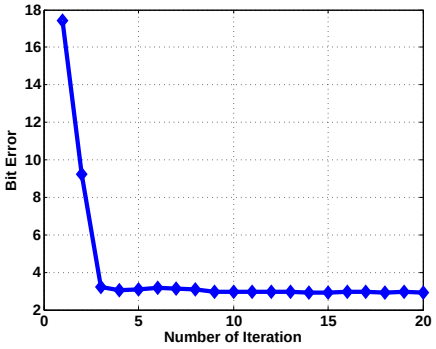

Figure 2. The objective function of Eq.(1) decreases as iterations of our block coordinate descent continue. 'Bit Error' refers to the number of bits that differ in the obtained binary codes from two different views. (32bit code learning)

4.3. Bit Error by Hamming Space Size

We investigate the Hamming distance of two obtained binary codes (value of Eq. 1) as a function of binary code length; 16, 32, 64, 128 and 256. Figure 3 shows that the number of bits that differ between binary codes from visual and text domains is almost always approximately $\frac{1}{10}$ of code length.

4.4. Image Category Retrieval

We retrieve images from an image pool by giving one or more samples (image or text/attribute) of a particular category as a query. In quantitative analysis, we compute the mean average precision (mAP) of re-

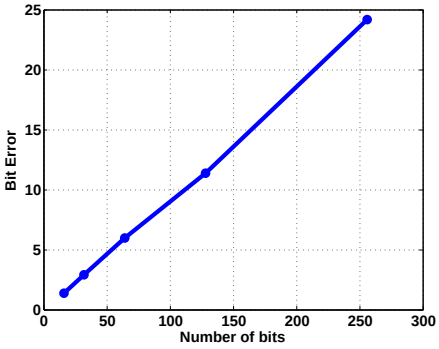

Figure 3. The error between textual and visual binary codes is a linear function of the length of the binary code. 'Bit Error' refers to the number of bits that differ in the obtained binary codes from two different views.

trieved images that belong to the same category of the query. In qualitative analysis, we present the images retrieved for a query by our method.

4.4.1. Quantitative Results

For quantitative analysis, we conduct a category retrieval experiment similar to (Torresani et al., 2010; Rastegari et al., 2012; 2011). We divided the dataset into two train/test segments. We train W_* using the training set. We compute the binary features for all the images (train and test). We take a set of images of a particular category as query set and train a classifier by taking the query set as positive samples and images from other categories in the training set as negative set. Then, we apply the classifier to all the samples of the test set, rank them by their classification confidence value and retrieve the top-K samples. We report precision and recall as an accuracy measure. By varying K in top-K we can draw a precision-recall curve. Since we are considering multiple categories, we report mean precision and recall.

We compare our binary code with several binary code methods including Iterative Quantization (**ITQ**) (Gong & Lazebnik, 2011), Spectral Hashing (**SH**) (Weiss et al., 2008) and Locality Sensitive Hashing (**LSH**) (Gionis et al., 1999a). Our method is referred to as Predictable Dual-view Hashing (**PDH**). We are not comparing our method with (Rastegari et al., 2012) because their method is not applicable to Dual-View. They require category labels of samples as supervision to train their binary codes. We used supervised ITQ coupled with CCA which uses data in two views to construct basis vectors in a common subspace.

Figure 4 and Figure 5 show mean average precision (mAP) of retrieved images by our method and other methods as a function of the number of bits. We presents the results with various numbers of queries given. As shown in the figure, our method (**PDH**) consistently outperforms all other methods.The high ranked images are not necessarily visually similar to the query. When we have few instances in the retrieval set the baseline methods have better precision because the high ranked images are the most visually similar to a query. This is not unexpected, since we optimize for cross-domain similarity, not visual similarity. We can directly compare by average precision(AP). As recall increases and the number of relevant images from the database tat are visually similar to the query are exhausted, the PDH dominates the other methods in precision.

Figure 6. (a) **Image2Image** retrieval. Given an image as a query, we find most similar images by nearest neighbor search in 32 bit PDH. (b) **Text2Image** retrieval. Given a sentence as query, we find the most descriptive images by nearest neighbor search of 32 bit PDH.

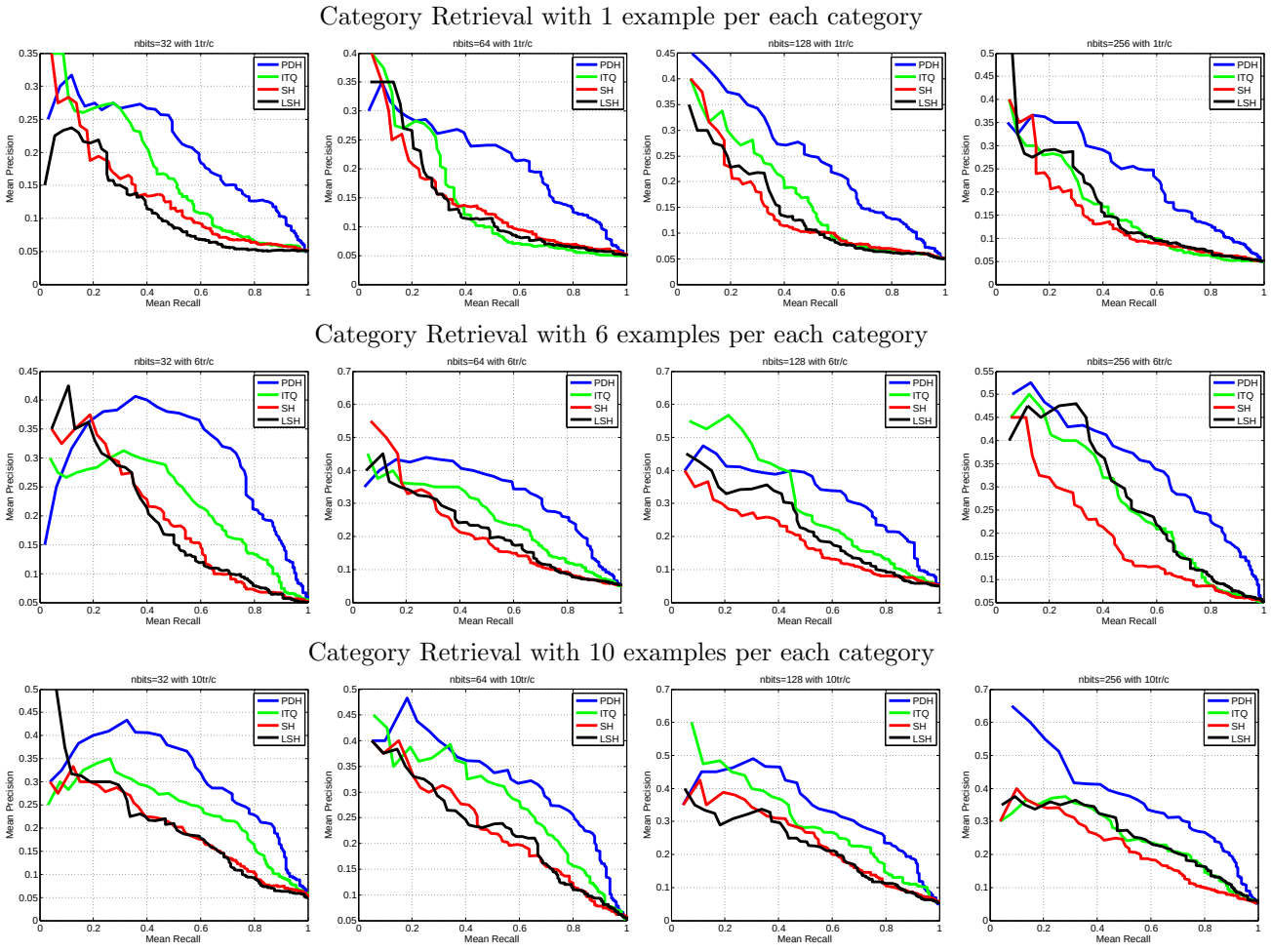

Figure 4. The result of category retrieval on PASCAL-Sentence dataset. Our method (PDH) is compared with three other baselines , Iterative Quantization (ITQ), Spectral Hashing (SH) and Locality Sensitive Hashing (LSH). We run experiments under different settings. We vary the code length (32, 64, 128 and 256) and we also vary the number of examples per each category in query by (1, 6 and 10)

4.4.2. Qualitative Results

We also present qualitative results of how our binary code performs. We perform two qualitative evaluations.

First, we conduct **Image2Image** retrieval. Given an image as a query, we retrieve the top-K closest images. Unlike the previous experiment we do not use an SVM but simply compute the Hamming distance of all other samples to the query sample and report the top-k most similar. Figure 6-(a) shows the retrieval for four query images which are represented by 32 bits. We report the top-5 most similar images. These retrieved images have significant semantic similarity to their query image.

Second, we perform a **Text2Image** retrieval task. Instead of using an image as query we use a sentence as query and we retrieve images for which this query sentence could be a good description. We map the sentence to our binary space and then identify similar points (images) in that space and report the top-k most similar. In figure 6-(b) we illustrate the retrieval set for five different sentences using 32 bit codes. Most of the retrieved images have content that is semantically similar to their query sentence.

5. Conclusion

We proposed a novel binary hashing method from two-views. We formulated an objective function to maintain predictability of the the binary codes and optimized the objective function by applying an iterative optimization method based on block coordinate descent. By conducting experiments on two datasets

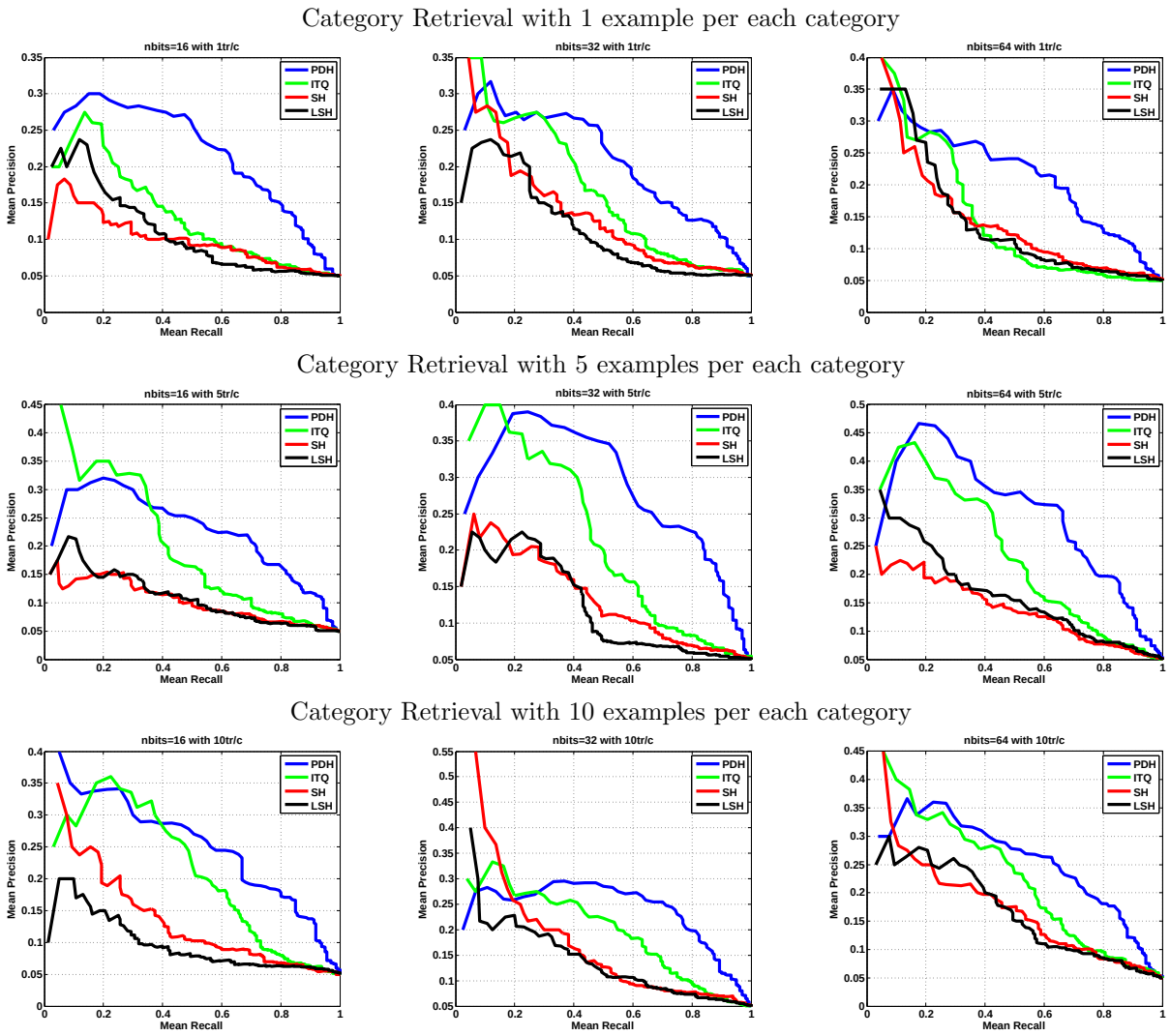

Figure 5. The result of category retrieval on SUN Dataset. Our method (PDH) is compared with three other baselines , Iterative Quantization (ITQ), Spectral Hashing (SH) and Locality Sensitive Hashing (LSH). We run the experiment under different settings of the problem. We changed the code length (32, 64, 128 and 256) and we also changed the number of examples per each category in query by (1, 6 and 10)

from visual-textual domain, we demonstrated the superiority of our method compared to the state-of-the-art binary hashing methods.

Acknowledgments

This work was partially supported by the US Government through NSF Award IIS-0812111 and ONR MURI Grant N000141010934.

References

Berg, Alexander C., Berg, Tamara L., III, Hal Daumé, Dodge, Jesse, Goyal, Amit, Han, Xufeng, Mensch, Alyssa, Mitchell, Margaret, Sood, Aneesh, Stratos, Karl, and Yamaguchi, Kota. Understanding and predicting importance in images. In *CVPR*, pp. 3562–3569, 2012.

Dodge, Jesse, Goyal, Amit, Han, Xufeng, Mensch, Alyssa, Mitchell, Margaret, Stratos, Karl, Yamaguchi, Kota, Choi, Yejin, III, Hal Daumé, Berg, Alexander C., and Berg, Tamara L. Detecting visual text. In *HLT-NAACL*, pp. 762–772, 2012.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. LIBLINEAR: A library for large linear classification. *JMLR*, 2008.

Farhadi, Ali, Hejrati, Mohsen, Sadeghi, Mohammad Amin, Young, Peter, Rashtchian, Cyrus, Hockenmaier, Julia, and Forsyth, David. Every picture tells a story: generating sentences from images. In *ECCV*, pp. 15–29, Berlin, Heidelberg, 2010.

Gionis, A., Indyk, P., and Motwani, R. Similarity search in high dimensions via hashing. In *VLDB*, 1999a.

Gionis, Aristides, Indyk, Piotr, Motwani, Rajeev, and Motwani, Rajeev. Similarity search in high dimensions via hashing. In *VLDB*, pp. 518–529, 1999b.

Gong, Y., Ke, Q., Isard, M., and Lazebnik, S. A Multi-View Embedding Space for Modeling Internet Images, Tags, and their Semantics. *CoRR*, abs/1212.4522, 2012.

Gong, Yunchao and Lazebnik, Svetlana. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, 2011.

Hardoon, D. R., Szedmak, S., Szedmak, O., and Shawe-taylor, J. Canonical correlation analysis; An overview with application to learning methods. Technical report, University of London, 2003.

Hwang, S. J. and Grauman, K. Accounting for the Relative Importance of Objects in Image Retrieval. In *BMVC*, 2010.

Hwang, S. J. and Grauman, K. Learning the Relative Importance of Objects from Tagged Images for Retrieval and Cross-Modal Search. *IJCV*, 100(2):134–153, 2012.

Kulis, Brian and Darrell, Trevor. Learning to hash with binary reconstructive embeddings. In *NIPS*, 2009.

Kulis, Brian and Grauman, Kristen. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, 2009.

Kulkarni, G., Premraj, V., Dhar, S., Li, Siming, Choi, Yejin, Berg, A.C., and Berg, T.L. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, pp. 1601 –1608, june 2011.

Kumar, Shaishav and Udupa, Raghavendra. Learning Hash Functions for Cross-View Similarity Search. In *IJCAI*, 2011.

Kuznetsova, Polina, Ordonez, Vicente, Berg, Alexander C., Berg, Tamara L., and Choi, Yejin. Collective generation of natural image descriptions. In *ACL (1)*, pp. 359–368, 2012.

Li, Siming, Kulkarni, Girish, Berg, Tamara L., Berg, Alexander C., and Choi, Yejin. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, pp. 220–228, 2011.

Lin, D. An Information-Theoretic Definition of Similarity. In *ICML*, pp. 296–304, 1998.

Liu, W., Wang, J., Ji, R., Jiang, Yu-Gang, and Chang, Shih-Fu. Supervised hashing with kernels. In *CVPR*, pp. 2074–2081, 2012.

Lyman, Peter, Varian, Hal R., Charles, Peter, Good, Nathan, Jordan, Laheem L., and Pal, Joyojeet. How much information? 2003, 2003. URL http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/.

Masci, J., Bronstein, M. M., Bronstein, A. A., and Schmidhuber, Jürgen. Multimodal similarity-preserving hashing. *CoRR*, abs/1207.1522, 2012.

Norouzi, Mohammad and Fleet, David. Minimal loss hashing for compact binary codes. In *ICML*, 2011.

Oliva, A. and Torralba, A. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 2001.

Ordonez, Vicente, Kulkarni, Girish, and Berg, Tamara L. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, pp. 1143–1151, 2011.

Patterson, G. and Hays, J. SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes. In *CVPR*, 2012.

Rashtchian, Cyrus, Young, Peter, Hodosh, Micah, and Hockenmaier, Julia. Collecting image annotations using amazon's mechanical turk. In *CSLDAMT*, pp. 139–147, 2010.

Rastegari, Mohammad, Fang, Chen, and Torresani, Lorenzo. Scalable object-class retrieval with approximate and top-k ranking. In *ICCV*, pp. 2659–2666, 2011.

Rastegari, Mohammad, Farhadi, Ali, and Forsyth, David A. Attribute discovery via predictable discriminative binary codes. In *ECCV (6)*, 2012.

Salakhutdinov, Ruslan and Hinton, Geoffrey. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, 2007.

Salakhutdinov, Ruslan and Hinton, Geoffrey. Semantic hashing. *Int. J. Approx. Reasoning*, 2009.

Shakhnarovich, Gregory, Viola, Paul A., and Darrell, Trevor. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, 2003.

Torralba, A., Fergus, R, , and Weiss, Y. Small codes and large image databases for recognition. In *CVPR*, 2008.

Torresani, Lorenzo, Szummer, Martin, and Fitzgibbon, Andrew. Efficient object category recognition using classemes. In *ECCV*, 2010.

Weiss, Yair, Torralba, Antonio, and Fergus, Robert. Spectral hashing. In *NIPS*, pp. 1753–1760, 2008.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In *CVPR*, 2010.

Zhen, Y. and Yeung, Dit-Yan. Co-Regularized Hashing for Multimodal Data. In *NIPS*, 2012.