Semantics-Preserving Bag-of-Words Models and Applications

Lei Wu, Steven C.H. Hoi, and Nenghai Yu

***Abstract*—The Bag-of-Words (BoW) model is a promising image representation technique for image categorization and annotation tasks. One critical limitation of existing BoW models is that much semantic information is lost during the codebook generation process, an important step of BoW. This is because the codebook generated by BoW is often obtained via building the codebook simply by clustering visual features in Euclidian space. However, visual features related to the same semantics may not distribute in clusters in the Euclidian space, which is primarily due to the semantic gap between low-level features and high-level semantics. In this paper, we propose a novel scheme to learn optimized BoW models, which aims to map semantically related features to the same visual words. In particular, we consider the distance between semantically identical features as a measurement of the semantic gap, and attempt to learn an optimized codebook by minimizing this gap, aiming to achieve the minimal loss of the semantics. We refer to such kind of novel codebook as Semantics-Preserving Codebook (SPC) and the corresponding model as the Semantics-Preserving Bag-of-Words (SPBoW) model. Extensive experiments on image annotation and object detection tasks with public testbeds from MIT's *Labelme* and PASCAL VOC challenge databases show that the proposed SPC learning scheme is effective for optimizing the codebook generation process, and the SPBoW model is able to greatly enhance the performance of the existing BoW model.**

***Index Terms*—bag-of-words models, object representation, semantic gap, distance metric learning, image annotation**

I. Introduction

With the advance of cameras and Web 2.0 technology, there has been a proliferation of digital photos on the Web. Massive photos unlabeled or with few tags have posed a great challenge for image retrieval tasks. Automatic image annotation is one promising solution to address this challenge. Generally, automatic image annotation is the process of employing computer programs to automatically assign an unlabeled image a set of keywords or tags, each of which represents certain semantic object/concept. By automatic image annotation, an image retrieval problem is turned into a text retrieval task, which can be effectively resolved by taking advantages of mature text indexing and retrieval techniques.

In the past decade, numerous studies have been focused on automatic image annotation [5], [8], [11], [20], [22]. Some earlier studies often extract global visual features, such as color and texture, from whole images to represent them as data points in vector space. As a result, image annotation is formulated as a supervised classification problem where data are given in some vector space [5]. Such an approach enjoys merits of efficient computation and compact storage, but often works effectively only for annotating scene images or single-object images. They usually performed poorly on generic images that contain multiple objects.

Later, besides extensive studies on global features, more promising studies have been focused on regional features. One typical approach is to partition an image into multiple regions/blobs based on image segmentation and clustering techniques. As a result, image annotation is turned into a machine translation task of classifying regions/blobs into keywords [8]. Along this direction, a variety of statistical learning techniques, such as relevance models [20], [22], have been applied to model the relationships of words and regions/blobs. The performance of these approaches is often sensitive to the quality of image segmentation, which is still an open research challenge in image processing.

Recently, thanks to the advances of powerful local feature descriptors, such as SIFT [27], researchers in computer vision have attempted to resolve object recognition/image annotation problems by a new approach, known as the "Bag-of-Words" (BoW) model, which was derived from natural language processing. Specifically, given an image, BoW first employs some interest point detector, e.g. the DoG (Difference of Gaussians) detector, to detect salient patches/regions in the image. Further, certain feature descriptor, e.g. SIFT, is applied to represent the local patches/regions as numerical feature vectors. The last step of BoW is to generate a codebook by converting the patches to "codewords", e.g. applying k-means algorithm to cluster all the feature vectors into k clusters, and then defining codewords based on the centers of the k resulting clusters. By mapping each visual feature in the image to the codewords, the image is represented by the histogram of the codewords. Based on the BoW representation, some well-known topic models, e.g. probabilistic latent semantic analysis (pLSA) [17], can be applied to analyze the topics of the images [6]. While sacrificing spatial information, BoW has generally shown promising performance for object categorization [34] and image annotation tasks [11].

However, BoW still has several important drawbacks. Other than the ignorance of spatial information that has been widely discussed in many recent papers [4], [26], [37], [24], [42], another critical disadvantage is that semantics of objects is considerably lost during the processes of sub-region detection and visual word generation. Firstly, the detection and segmentation of sub-regions damage the semantic integration.

Mr. Lei Wu is from MOE-MS Key Lab of MCC, Dept of EEIS, University of Science and Technology of China, Hefei China, 230026. This work was performed when Mr. Lei Wu was a research assistant at Nanyang Technological University.

Corresponding Author: Dr. Steven C.H. Hoi, School of Computer Engineering, Nanyang Technological University, Singapore 639798, E-mail: chhoi@ntu.edu.sg

Several methods have been proposed to locate the sub-regions in an image, e.g. regular grid [42], interest point detector [36], [27], random sampling [28], sliding windows [23], other segmentation methods [32], [16] etc. However, due to the lack of human knowledge, these methods cannot locate the semantically intact regions very accurately, which partially causes the semantic gap problem. Secondly, it is problematic for generating the visual words using k-means clustering in Euclidian space, which implicitly assumes that SIFT features of similar semantics are distributed in the same cluster in Euclidian space. This however does not always hold, especially for high dimensional SIFT features. Unlike the completely unsupervised clustering by k-means in visual word generation, we believe that a semi-supervised clustering approach with the aid of side information could lead to more effective codebook for object representation.

To this end, this paper proposes a novel **Semantics-Preserving Bag-of-Words** (SPBoW) model, which considers the distance between the semantically identical features as a measurement of the semantic gap, and tries to learn a codebook by minimizing this semantic gap. We formulate the codebook generation task as a distance metric learning problem, which can be formalized as semi-definite programming (SDP). We then propose an efficient eigen projection algorithm to solve the optimization problem efficiently. With the integrated knowledge and side information, the semantic gap can be minimized and the codebook is able to consistently represent the semantics of the objects. To the best of our knowledge, this is the first distance metric learning approach to overcome the limitation of semantics lost in BoW models.

As a summary, the main contributions of this paper include: (1) we are the first to propose a measurement of the semantic gap; (2) we propose to bridge the semantic gap via distance metric learning method; (4) we propose and implement an efficient algorithm to solve the codebook learning task; (4) we suggest a novel object based codebook scheme; (5) we propose a measurement for visual complexity; (6) the proposed method can automatically decide the size of the codebook for each category; (7) we evaluate and compare a number of different methods for the codebook generation process in building various bag-of-word models towards object annotation tasks.

The rest of the paper is organized as follows. Section II reviews related work. Section III presents the framework of the SPBoW model. Section III-B gives the details of the object representations for this novel model. Section IV elaborates on the codebook learning task and formulates the task as an optimization problem. Section V applies the proposed semantics-preserving codebook (SPC) technique on object annotation tasks. Section VI compares the SPBoW model and the metric learning algorithm with several state-of-the-art methods for object annotation experiments on MIT's Labelme testbed [30] and object categorization on PASCAL VOC challenge testbed [9]. Section VII concludes the paper.

II. Related Work

Our work is related to several research topics, including image annotation/object recognition, and distance metric learning. Below we briefly review the related work of both categories respectively.

A. Image Annotation and Object Recognition

In literature, numerous studies have been devoted to image annotation and object recognition. They can be roughly grouped into three major categories. The first category is based on global features [13]. As a result, regular supervised classification techniques, such as SVM, can be applied to solve the categorization and annotation tasks.

The second category is to extract regional features such that an image can be represented by a set of visual regions/blobs [2], [8], [22]. The image annotation task is thus converted to a problem of learning keywords/tags from visual regions/blobs. For instance, Barnard et al. [2] treated image annotation as a machine translation problem. Jeon et al. [8] proposed the cross-media relevance models (CMRM) model, which combines both surrounding texts and image contents for annotation. Jin et al. [22] studied coherent language models that takes into account the word-to-word correlation.

The last category is focused on applying bag-of-features or bag-of-words representations for image annotation/object recognition [6], [21], [35]. Csurka et al. [6] proposed a bag-of-keypoints approach similar to BoW in text categorization for visual object categorization. Jiang et al. [21] studied some practical techniques to improve the performance of bag-of-features for object recognition and retrieval. Recently, Wu et al. [42] proposed a language modeling approach to address one limitation of the BoW models, i.e., the loss of spatial information. These methods generate the codebook by clustering visual features in the original feature space. Due to the semantic gap, each visual word may contain multiple semantic meanings and the same semantic meaning may be represented by multiple visual words. In these models, each visual word actually does not have correspondence to a precise semantic meaning.

Besides, there are also some emerging paradigms for image annotation, such as search-based annotation [38] that explores WWW images in helping the annotation tasks, and the ALIPR paradigm [25], which used advanced statistical learning techniques to provide fully automatic and real-time annotation for digital pictures. These techniques are not highly relevant to our focus, and are thus out of the discussions in this paper.

B. Distance Metric Learning

From a machine learning point of view, our work is related to supervised distance metric learning (DML). Specifically, consider a set of n data examples $X = \{x_i \in \mathbb{R}^d\}_{i=1}^n$ in d-dimensional vector space, the objective of DML is to find an optimal Mahalanobis metric M from training data with side information that can be either class labels or general pairwise constraints [43].

In literature, DML has been actively studied recently. Existing DML studies can be roughly grouped into two major categories. One category is to learn metrics with class labels, such as Neighbourhood Components Analysis (NCA) [14], which

are often studied for classification [12], [39], [44]. Neighborhood Component Analysis (NCA) [14] learns a distance metric by extending the nearest neighbor classifier. The maximum-margin nearest neighbor (LMNN) classifier [39] extends NCA through a maximum margin framework. Information-Theoretic Metric Learning (ITML) [7] presented the metric learning problem from the information theory, and achieved the optimal metric by minimizing the differential relative entropy between two multivariate Gaussians under constraints on the distance function. The other category is to learn metrics from pairwise constraints that are mainly used for clustering and retrieval. Examples include Relevant Components Analysis (RCA) [1] and Discriminative Component Analysis (DCA) [19], amongst others [43], [33], [18], [40]. RCA learns a global linear transformation from the equivalence constraints. The learned linear transformation can be used directly to compute distance between any two examples. DCA and Kernel DCA [19] improve RCA by exploring negative constraints and aiming to capture nonlinear relationships using contextual information. Essentially, RCA and DCA can be viewed as extensions of Linear Discriminant Analysis (LDA) by exploiting the must-link constraints and cannot-link constraints.

III. Framework of Semantics-Preserving Bag-of-Words Models

A. Overview

The BoW model treats an image as a bag of "code-words", which essentially consists of a set of independent local appearance features. These features are either located by salient region detectors like SIFT, random samplings like random windows, segmentation, or regular grid. These high-dimensional features may contain much noise and redundancy, and are often difficult to store and use directly. Hence, visual words are further generated by performing clustering on these features. Through feature clustering, each visual word usually corresponds to a cluster in the feature vector space. Based on the visual words, each of the features detected from the image can be mapped to one of the most similar visual words by measuring the distance between the feature and all visual words. Consequently, a histogram of visual words can be calculated to represent an image.

BoW can be applied for object annotation by either a naïve Bayes classifier [41] or more complex latent topic analysis methods, such as pLSA [34] and LDA [3]. For example, by a naïve Bayes classification approach, object annotation is equivalent to matching the visual word histogram of an image with respect to the visual word histograms of semantic objects. The name of an object is annotated to the image if the visual word histogram of the object is matched from the visual word histogram of the image.

In this paper, we aim to investigate a new BoW framework for object representation to overcome the limitations of existing BoW with applications to image annotation and object detection tasks. In particular, we propose a novel **Semantics-Preserving Bag-of-Words** (SPBoW) framework. Fig. 1 illustrates the flowchart of our framework. First of all, in the training process, objects in the images are segmented and tagged by users. SIFT features are extracted from the images to represent these objects. The SIFT features that are located at the same semantic parts of objects are considered as *relevant* to each other, and will be used as the *similar* pairwise constraints in our learning task; on the other hand, any two SIFT features that are located at different semantic parts of objects are considered as *irrelevant*, and will be treated as the *dissimilar* pairwise constraints in our learning task. We refer to the collections of similar and dissimilar pairwise constraints as "side information".

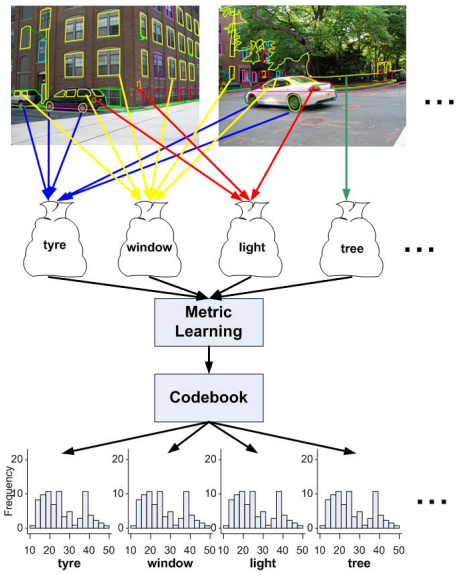


Fig. 1. The process of building the semantics-preserving bag-of-words model.

In this paper, we propose a novel learning scheme to optimize the distance metric from the side information. By minimizing the semantic loss, the optimized distance metric aims to achieve the Semantics-Preserving Codebook (SPC) representation, which can be beneficial for image annotation and object categorization tasks.

B. SPBoW for Object Representation

In traditional BoW, an image is represented by the histogram of visual words from a codebook. This simple representation has some drawbacks. First of all, both the visual words extracted from the object regions and the visual words extracted from the background regions are all incorporated for generating the BoW model. Such a simple approach however brings the background noise into the resulting model which is supposed to describe only the object. Moreover, this representation may be influenced if an image contains multiple objects. However, many real-world images usually contain multiple objects. As a result, all other irrelevant objects in the images will become noises when building the regular BoW model for certain objects. Although this problem may be partially resolved by the latent topic analysis, it also faces a number of challenges, e.g. how to determine the number of latent topics.

For the above reasons, our new SPBoW approach aims to preserve the semantics by modeling each individual object rather than simply modeling a whole image. In particular,

we adopt some images from MIT's Labelme testbed [30] as training data, in which objects are well segmented and labeled by users. By the proposed SPBoW framework, we first apply SIFT to extract features from each image. The SIFT features that are located at the regions of the same semantics (label) in all the images are collected to represent the semantics. In order to preserve the semantics in the BoW model, all the collected features related to the same semantics are clustered into one or several discriminative visual words for representing the object based on an optimized distance metric that aims to minimize the overall semantic loss. The visual words used for representing an object may describe different semantic parts or different views of the object. Finally, we note that the set of visual words used for one object is often different from the set used for another. This is very different from the regular BoW model where all objects share the same set of visual words. Next we present a novel learning technique that aims to find an optimal distance metric to overcome the limitation of semantic loss during the codebook generation process.

IV. Learning to Optimize Codebooks

Codebook generation is a critical step of building the BoW model. Instead of generating the codebook by applying simple k-means clustering in Euclidean space which often leads to much semantic loss, in this paper, we suggest a novel metric learning scheme that exploits side information for minimizing the semantic loss in the codebook generation process.

A. Problem Formulation

We first formalize the representation of side information, which is illustrated in Fig 2. Assume we are given a set of pairwise feature instances $\{(x_{i1}, x_{i2})\}_{i=1}^{N}$ and a set of corresponding instance constraints, $\{(z_{i1}, z_{i2}, y_i)\}_{i=1}^{N}$, where $x_{i1} \in \mathbb{R}^d$ and $x_{i2} \in \mathbb{R}^d$ are two d-dimensional feature instances, e.g. SIFT feature vectors; x_{i1} indicates the first feature vector in the pair, and x_{i2} is the second feature vector in the pair; z_{i1} and z_{i2} are binary indicators to indicate whether a feature instance is located at the object region or the background region in the image. As shown in Fig 2(a), if feature instance x_{i1} is on the object region, then $z_{i1} = 1$; otherwise $z_{i1} = 0$. The variable y_i indicates whether the feature instances in pair (x_{i1}, x_{i2}) are of the same semantics. If both x_{i1} and x_{i2} are on the same semantic parts of objects, e.g. *tyre* of cars as shown in Fig 2(d), then $y_i = 1$. If two features are of different semantics, i.e., they appear on different semantic parts of two different objects (Fig 2(c)) or they are located at the same object but on different semantic parts, e.g. *tyre* and *window* of cars as shown in Fig 2(b), then $y_i = -1$.

In general, side information can be generated automatically from the locations of feature points in the well-segmented images. For example, in the Labelme testbed, objects and background regions are manually separated for each image, and different parts of the objects are also manually segmented by users. Hence, if two feature vectors are located at the same region or at the regions of the same semantic label, they will be considered as the same semantic meaning, i.e., $y_1 = 1$. Similarly, in PASCAL VOC2006 datasets, objects in each image are separated from the background by a bounding box. Thus, if two features are in the same bounding box or in the bounding boxes with the same label are treated of the same semantic meanings.

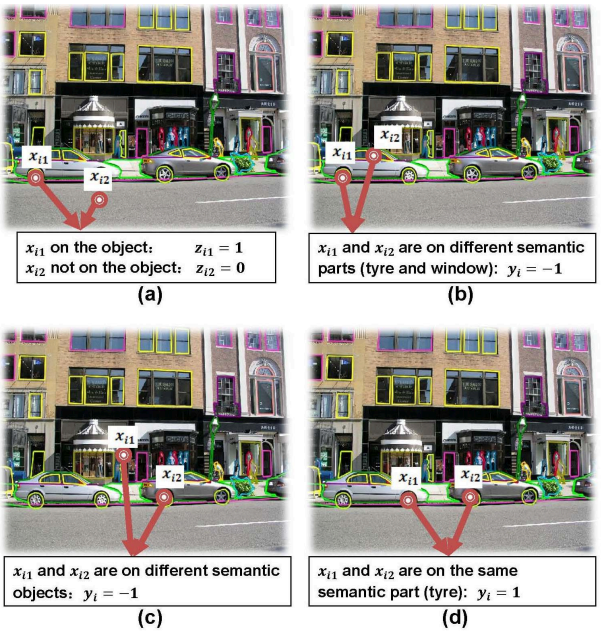


Fig. 2. Illustration of side information between objects and feature instances.

Given the above side information, the goal of our task is to learn a distance metric A to effectively measure distance between any two visual features x_{i1} and x_{i2} that is often represented in the following framework:

$$d(x_{i1}, x_{i2}) = \sqrt{(x_{i1} - x_{i2})^\top A (x_{i1} - x_{i2})} \tag{1}$$

where matrix $A \in \mathbb{R}^{d \times d}$ is the target distance metric that must be positive and semi-definite w.r.t. the properties of a valid metric, i.e., $A \succeq 0$. To find an optimal metric A, the basic principle of our metric learning task is that distances between visual feature vectors of the same semantics should be minimized, and meanwhile distances between feature vectors of different semantics should be maximized. Based on this principle, we can search for the optimal metric that facilitates clustering the feature vectors of the same semantics into the same visual words, in which each visual word has certain specific semantic meaning. To this end, we formulate our distance metric learning problem into the following optimization:

$$\min_{A \succeq 0, b} \quad \sum_i z_{i1} z_{i2} \xi_i + \frac{\lambda}{2} tr(AA^\top) \tag{2}$$

$$s.t. \quad y_i(\|x_{i1} - x_{i2}\|_A - b) \leq \xi_i, \xi_i \geq 0, i = 1, \ldots, n \tag{3}$$

$$\|A\| = 1/\sqrt{\lambda} \tag{4}$$

where $\|\cdot\|_A$ is the Mahanalobis distance between two features under metric A. The first term of the objective function is the slack variable which accounts for the semantic loss w.r.t. the side information of n pairwise constraints $\{(x_{i1}, x_{i2}, z_{i1}, z_{i2}, y_i)\}_{i=1}^{n}$. With the first inequality constraint, minimizing this term will make the distance between two

semantically identical features closer and thus more likely to be assigned to the same visual word. The second term of the objective function is the regularization term, which prevents the overfitting by minimizing the complexity of the model. The second equality constraint is introduced to prevent the trivial solution by shrinking metric A into a zero matrix, and λ is a constant parameter. By solving the optimization problem, we can obtain the optimized distance metric A and the threshold variable b that could be used to determine whether two features are similar or dissimilar. In general, the above optimization problem belongs to a general semi-definite programming (SDP), which is often difficult to solve with global optima for large applications.

B. Optimization

In this section, we present a stochastic gradient search algorithm by combining with an active constraint selection scheme to efficiently solve the above optimization problem. To simplify the formulation, we denote the feature matrix as $X \in \mathbb{R}^{N_{tr} \times d}$ where N_{tr} is the number of SIFT features in the training set, and d is the feature dimension. We also represent all the feature pairs (x_{i1},x_{i2}) in the training data by two feature matrices $X_1 = [x_{11}, x_{21}, \cdots, x_{n1}]^\top$ and $X_2 = [x_{12}, x_{22}, \cdots, x_{n2}]^\top$, and similarly their constraints by three matrices $Z_1 = \mathrm{diag}(z_{11}, z_{21}, \cdots, z_{n1})$, $Z_2 = \mathrm{diag}(z_{12}, z_{22}, \cdots, z_{n2})$ and $Y = \mathrm{diag}[y_1, \cdots, y_n]$. The proposed iterative optimization scheme is described in the following steps.

First of all, we actively choose a subset of informative side information from the training data as the training instances. In particular, the training instances must satisfy either one of the two criterions: (1) the features are of the same semantics but with large distance in the current metric space; or (2) the features are of different semantics but with small distance in the current metric space.

Based on the selected training dataset S_t in the t-th iteration, we then apply the gradient descent technique to search for the optimal metric A and threshold b.

Finally, to enforce the valid metric constraint, we project the current solution of metric A back to a positive semidefinite (PSD) cone by an eigen decomposition approach. The details of the proposed Semantics-Preserving Metric Learning (SPML) algorithm are described in Algorithm 1, in which γ is a learning rate variable that is determined empirically. $D_X = X_1 - X_2$ is the difference between the two feature matrices X_1 and X_2. Empirically, this iterative algorithm converges quickly with no more than 5 iterations.

C. Convergence Analysis

We now analyze the convergence of the algorithm. Let us denote the objective function in the t-th iteration as follows:

$$\mathcal{L}(A_t, \mathcal{S}_t) = \sum_{(x_{i1},x_{i2},y_i)\in\mathcal{S}_t} y_i(\|x_{i1} - x_{i2}\| - b_t) + \frac{\lambda}{2} tr(A_t A_t^\top) \tag{5}$$

To prove the convergence of the algorithm, we first calculate the bound of the objective function after T iterations. Here we adopt the following theorem proposed in [15], which provides a bound for a general sub-gradient method. The detailed proofs and explanations can be found in [31].

Algorithm 1 The Semantics-Preserving Metric Learning (SPML) algorithm

INPUT:

- SIFT feature matrix: $X \in \mathbb{R}^{N\times d}$
- pairwise constraint $(x_{i1}, x_{i2}, z_{i1}, z_{i2}, y_i)$, where x_{i1} is the i_1^{th} SIFT feature, z_{i1} indicate whether the location of feature x_{i1} is on the semantic object, and constraints $y_i = \{+1, 0, -1\}$ represents feature x_{i1} and x_{i2} are on the same semantic part of the object, not known, or on different semantic parts.
- regularization parameter λ
- learning rate parameter γ

PROCEDURE:

1: initialize metric and threshold: $A = I$, $b = b_0$
2: set iteration step t = 1;
3: **repeat**
4: (1) update the learning rate: $\gamma = \gamma/t$, t = t + 1
5: (2) update the subset of training instances: $\mathcal{S}_t^+ = \{(x_{i1}, x_{i2}, y_i) | (1 + y_i)\|x_{i1} - x_{i2}\|_A^2 > 1\}$ $\mathcal{S}_t^- = \{(x_{i1}, x_{i2}, y_i) | (1 - y_i)\|x_{i1} - x_{i2}\|_A^2 < 1\}$ $\mathcal{S}_t = \mathcal{S}_t^+ \bigcup \mathcal{S}_t^-$
6: (3) compute the gradients w.r.t. A $\nabla_A \mathcal{L} \leftarrow Z_1 Z_2(\lambda A + D_X^\top Y^\top D_X)$, $D_X = X_1 - X_2$,
7: (4) compute the gradients w.r.t. b $\nabla_b \mathcal{L} \leftarrow \mathrm{tr}(Z_1 Z_2 Y)$
8: (5) update metric and threshold: $A_{t+1} \leftarrow A_t - \frac{\gamma}{t}\nabla_A \mathcal{L}$, $\quad b_{t+1} \leftarrow b_t - \frac{\gamma}{t}\nabla_b \mathcal{L}$
9: (6) project A back to the PSD cone: $A_{t+1} = \sum_{i=1}^d \lambda_i \phi_i \phi_i^\top$ $A_{t+1} \leftarrow \sum_i \max(0, \lambda_i)\phi_i \phi_i^\top$
10: (7) normalize A_{t+1} to satisfy $\|A_{t+1}\| = \frac{1}{\sqrt{\lambda}}$: $A_{t+1} \leftarrow \frac{1/\sqrt{\lambda}}{\|A_{t+1}\|} A_{t+1}$
11: **until** convergence

OUTPUT:

- feature metric A, threshold variable b

Theorem 1: Let $\mathcal{L}_1, \cdots, \mathcal{L}_T$ be a sequence of λ-strongly convex functions w.r.t the objective function $\frac{1}{2}tr(\cdot)$, where $\mathcal{L}_t = \mathcal{L}(A, \mathcal{S}_t)$. Let $\mathcal{A}$ be a closed convex set and define $\prod_{\mathcal{A}}(A) = \arg\min_{A'\in\mathcal{A}} \|A - A'\|$. Let $A_1, \cdots, A_{T+1}$ be a sequence of vectors such that $A_1 \in \mathcal{A}$ and for $t \geq 1$,$A_{t+1} = \prod_{\mathcal{A}}(A_t - \frac{\gamma}{t}\nabla_t)$,where ∇_t is a subgradient of $\mathcal{L}_t$ at A_t. Assume that for all t, $\|\nabla_t\| \leq G$. Then, for all $u \in \mathcal{A}$ we have

$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}_t(A_t) \leq \frac{1}{T}\sum_{t=1}^T \mathcal{L}_t(u) + \frac{G^2(1 + ln(T))}{2\lambda T} \tag{6}$$

□

By applying Theorem 1, we can prove the following corollary.

Corollary 2: Let $\mathcal{L}_1, \cdots, \mathcal{L}_T$ be a sequence of λ-strongly convex functions. Let $\mathcal{A}$ be a closed convex set and define $\prod_{\mathcal{A}}(A) = \arg\min_{A'\in\mathcal{A}} \|A - A'\|$. Let $A_1, \cdots, A_{T+1}$ be a sequence of matrics such that $A_1 \in \mathcal{A}$ and for $t \geq 1$, $A_{t+1} = \prod_{\mathcal{A}}(A_t - \frac{\gamma}{t}\nabla_t)$, where ∇_t is a subgradient of $\mathcal{L}_t$ at A_t. Then, the bound for the proposed objective function is

$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}_t(A_t) \leq \frac{1}{T}\sum_{t=1}^T \mathcal{L}_t(A^*) + \frac{(\sqrt{\lambda} + \sum_i \xi_i)^2(1 + ln(T))}{2\lambda T}$$

where A^* is the optimal solution. □

Using Corollary 2 and the convexity property of the objective function $\mathcal{L}$, i.e., $\frac{1}{T}\sum_{t-1}^{T}\mathcal{L}_t(A_t) \leq \mathcal{L}(\frac{1}{T}\sum_{t=1}^{T}A_t)$ we can further show the following corollary of the optimization bound.

Corollary 3: Let $\mathcal{L}(A_t, \mathcal{S}_t) = \mathcal{L}_t(A_t)$, and $\mathcal{L}(A_t) = \mathbb{E}_{S_t}\mathcal{L}(A_t, S_t)$. Assume the conditions stated in Corollary 2 and denote by $\bar{A} = \frac{1}{T}\sum_{t=1}^{T}A_t$, and $G = (\sqrt{\lambda} + \sum_i \xi_i)$, then we have the following result:

$$\mathcal{L}(\bar{A}) \leq \mathcal{L}(A^*) + \frac{G^2(1+ln(T))}{\lambda T}$$

□

The proofs to the above two corollaries can be found in http://www.cais.ntu.edu.sg/~chhoi/SPBOW/proofs.pdf. By denoting $\eta(T) = \frac{G^2(1+ln(T))}{2\lambda T}$, we can see that when the iteration number $T \to \infty$, $\eta(T) \to 0$. This corollary thus proves the convergence of the algorithm. Finally, by applying Corollary 3 and using the first order Taylor expansion of function $ln(T)$, we obtain that for achieving a solution with accuracy ϵ, the algorithm requires $O(\frac{G^2}{\epsilon\lambda})$ iterations.

D. Codebook Generation

A codebook can be generated by clustering the features under the learned distance metric into some visual words or codes. Different visual words could represent different views or different parts of an object. In this paper, we propose to generate the codebook for each object category such that the linkage between the codewords in the codebook and the high-level semantics of object category can be established effectively, which is essential to bridge the gap between low-level features and high-level semantics.

Specifically, for each object category, we first collect all the related features from the same object regions, and then perform the k-means clustering based on the optimized distance metric A that is obtained from the proposed SPML scheme. By the k-means clustering, we can obtain a set of k clusters (i.e., visual words or codewords) for this object category. Finally, we form a global codebook by gathering all codewords from all object categories. We thus refer to the resulting codebook as "Semantics-Preserving Codebook" (SPC).

In general, there are two important issues for SPC, including (1) codebook size assignment, and (2) visual word generation.

1) Codebook Size Assignment: This is to determine how many codes should be assigned for each object category. One straightforward approach is to uniformly assign the same number of codes for every object. This however does not explore the difference of complexity in semantic understanding for different objects. A more desirable approach is to assign varied codebook sizes for different objects. To address this challenge, we introduce two principles for the assignment task:

(1) The number of codes increases linearly w.r.t. the visual complexity of an object category
(2) Visual complexity of an object category can be measured by the diversity of its associated features.

In this paper, we suggest to measure the visual complexity of an object category by applying the information theory. In particular, consider each object category as a bag of features, each feature in the object category has a probability of being generated from the bag. Such a probability can be estimated by either the distance to the mean of all features or the frequency of the features. For example, let us denote by C_i an object category and x_j some feature, we can estimate the generative probability $p(x_j|C_i)$ as follows:

$$p(x_j|C_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{\|x_j-\hat{x}\|_A^2}{2\sigma^2}} \tag{7}$$

where $\hat{x} = \frac{1}{n_{C_i}}\sum_{x_j \in C_i} x_j$, and n_{C_i} is the total number of features related to the objects from C_i. Based on the above estimated probability, we calculate the information entropy of the bag as a measurement of the object's visual complexity:

$$H(C_i) = -\sum_{x_j \in C_i} p(x_j|C_i)\log p(x_j|C_i) \tag{8}$$

Finally, we assign object C_i the number of codes L_{C_i} that is proportional to its visual complexity, i.e.,

$$L_{C_i} = \lfloor L_{max} \times \frac{H(C_i)}{\log n_{C_i}} \rfloor \tag{9}$$

where L_{max} is the maximum size of the SPC for each category. The total number of visual words for all categories is $L_{max} \times M$, where M is the number of categories.

Algorithm 2 Codebook Generation Algorithm

INPUT:
- features and their object labels $\{(x, y), x \in \mathcal{X}, y \in \mathcal{C}\}$
- optimized distance metric A
- codebook size assigned for each object $L_{C_i}, i = 1, \cdots, M$
- the number of clusters for clustering $K > \max_i L_{C_i}$

PROCEDURE:
1: initialize the number of visual words $L = 0$
2: **for** $i = 1 : M$ **do**
3: clustering features of the i-th object $X_i = \{(x, C)|C = C_i\}$ into K clusters
$$[c_{ij}, r_{ij}] = kmeans(X_i, K)$$
4: calculate the size of each cluster:
$$S_{ij} = \sum_x \delta(\|x - c_{ij}\|_A, r_{ij})$$
5: sort clusters by their sizes
$$c_{ij} \leftarrow sort(c_{ij}, S_{ij}) \qquad r_{ij} \leftarrow sort(r_{ij}, S_{ij})$$
6: adopt top L_{C_i} largest clusters as visual words for the category
$$w_{L+j} = c_{ij}, r_{L+j} = r_{ij}, j = 1, \cdots, L_{C_i}$$
7: update the number of visual words $L = L + L_{C_i}$
8: **end for**

OUTPUT:
- the centers of visual words w_k and their range radius r_k, $k = 1, \cdots, L_{max}$

2) Visual Word Generation: This task aims to build the codebook for each object category C_i by applying the k-means clustering on the associated features to generate a set of L_{C_i} visual words.

Let us denote by X_i a collection of features belonging to object category C_i, i.e., $X_i = \{(x, y)|x \in \mathcal{X}, y = C_i, C_i \in \mathbb{C}\}$, where y denotes the object category label of feature x, $\mathcal{X}$ is the feature space, and $\mathbb{C}$ is the label space. The proposed algorithm first applies the k-means clustering on X_i with the optimized metric A to generate a set of K clusters, denoted by $\{c_{ij}, r_{ij}|j = 1 \cdots, K\}$, where K is set to be larger than

$\max_i L_{C_i}$, c_{ij} denotes the center of the j-th cluster and r_{ij} denotes the range radius of the cluster, which is defined as the largest distance from the features to the cluster center.

To reduce noisy clusters, we further sort the K clusters according to their sizes S_{ij} that are calculated below:

$$S_{ij} = \sum_x \delta(\|x - c_{ij}\|_A, r_{ij}), \text{where} \quad \delta(a,b) = \begin{cases} 1, & a \le b; \\ 0, & \text{otherwise.} \end{cases}$$

The algorithm then chooses top L_{C_i} largest clusters as the set of visual words for the codebook of category C_i. Finally, the algorithm gathers all the visual words from every object category and outputs the set of visual words along with their ranges, i.e., $\{w_k, r_k\}_{k=1}^{L_{max}}$, as the final SPC. The algorithm of visual word generation is summarized in Algorithm 2.

E. Visual Word Histogram

To apply SPC in the test phase, the key task is to generate the visual word histogram for a novel test image. In particular, we first extract SIFT features from the novel image, and then map each of the SIFT features $x \in \mathcal{R}^d$ to the visual word id k in the cookbook. Different from traditional BoW, in our approach, one visual feature can be assigned to multiple visual words in different object categories. This is because the ranges of visual words may overlap each other and the same semantics may appear in different objects. For example, "*window*" can appear in both "*building*" and "*car*" objects. Hence, instead of assigning a feature to the closest visual word, we suggest to assign the feature to a visual word when the distance between the feature and the visual word is smaller than the range radius.

Specifically, we define a mapping function $\pi(x,k)$ between feature x and visual word w_k as follows:

$$\pi(x,k) = \begin{cases} 1, & \|x - w_k\|_A < r_k; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

By the mapping function, we calculate the frequency of a visual word w_k appearing in image I as: $f_I(k) = \sum_{x \in I} \pi(x,k)$. Finally, we can obtain the visual word histogram by normalizing the visual word frequencies as follows:

$$h_I(w_k) = \frac{f_I(k)}{\sum_{v=1}^{L_{max}} f_I(v)} \tag{11}$$

V. Generative and Discriminative Models with SPBoW

Similar to existing BoW models, the proposed SPBoW can also be easily adopted for existing classification methods, including both generative and discriminative models. Fig. 3 illustrates the idea of applying SPBoW for annotating a novel image in an object annotation task. Below we discuss two representative methods for applying SPBoW in image annotation and object categorization applications.

A. Generative Model

Based on the SPBoW technique, we now discuss how to apply the resulting semantics-preserving codebook for building generative models in an object annotation task. Assume that we are given a set of labeled image regions $\{(I_j, C(I_j))\}_{j=1}^{N_{tr}}$. Our goal is to automatically annotate a novel image I.

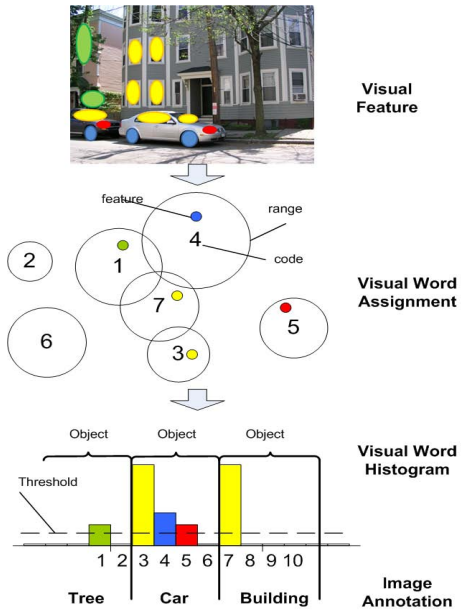


Fig. 3. Illustration of object annotation using the SPBoW representation.

First of all, we extract SIFT features from these training regions $\{x \in I_j, j = 1, \ldots, N_{tr}\}$. For each feature, we then find its mappings to the visual words in the codebook, which is based on the mapping function defined in (10). We also translate the region's object labels from the feature x to the mapping visual word w_k when the mapping result is nonzero, i.e., $\pi(x,k) = 1$. Finally, by gathering all visual words associated with a certain semantic object, we estimate the visual word's conditional distribution:

$$p(w_k|C_i) = \frac{\sum_{\{x|C(x)=C_i\}} \pi(x,k) + 1}{\sum_k \sum_{\{x|C(x)=C_i\}} \pi(x,k) + V}$$

where V is the vocabulary size. In the above probability formula, we adopt the Laplace smoothing to avoid the zero probability issue. With the assumption of uniform distribution of images, the likelihood of object category C_i appearing in image I can be calculated by a Naïve Bayes model as follows:

$$p(C_i|I) \propto p(I|C_i)p(C_i) \propto p(C_i) \prod_k p(w_k|C_i)^{f_I(k)} \tag{12}$$

where $f_I(k)$ is the frequency of visual word w_k appearing in the test image I, and prior $p(C_i)$ can be calculated based on the normalized frequencies of the object category that appears in the training data. Finally, we rank the object categories by their likelihood $p(C_i|I), C_i = 1 \cdots, K$, and top N ($N = 1, \cdots, 10$) ranked categories are used to annotate the image.

B. Discriminative Models

The learned SPC can also be used in a discriminative learning setting. To illustrate this property, we apply the codebook to train SVM models for classifying the visual objects. Similarly, we are given a set of training images (or image

regions) and their semantic categories $\{(I_j, C(I_j))\}_{j=1}^{N_{tr}}$. Based on the SPBoW representation, we can represent each image I_j by an $L_{\max}$-dimensional vector, which is also called visual word histogram $\mathbf{h}_I = [h_I(w_1), h_I(w_2), \cdots, h_I(w_{L_{max}})]$ by Algorithm 3. Since it is a multi-class classification task, we then train multiple binary SVM models by one-against-all. Specifically, for the i-th category, we build a binary SVM classifier as follows:

$$\min_{\omega,b} \quad \frac{1}{2}\|\omega\|^2 + C\sum_j \xi_j \tag{13}$$

$$s.t. \quad y_j(i)(\omega \cdot \mathbf{h}_{I_j} - b) \geq 1 - \xi_j, \xi \geq 0, 1 \leq j \leq N_{tr} \tag{14}$$

where ω is the weight vector in SVM, C is the penalty constant, ξ_j are slack variables, $y_j(i)$ is a binary label function for the i-th category such that $y_j(i) = 1$ if $C(I_j) = i$; and $y_j(i) = -1$ otherwise. In the object detection phase, by the similar representation, each novel test image will be classified by all of the binary SVM classifiers, in which a positive output indicates a specific object is detected on the image.

VI. Experiments

In this section, we conduct extensive experiments to empirically evaluate the performance of the proposed SPBoW model and the existing BoW model for image annotation and object categorization tasks. In addition to the proposed metric learning algorithm, we also show that our SPBoW framework can be integrated with other existing DML techniques. In our experiments, we extensively evaluate different implementations of SPBoW models by adapting other existing DML algorithms in our framework.

A. Experimental testbed

We adopt a dataset from MIT's Labelme project [30], which consists of 495 objects and 185 images that are mostly related to downtown streets. The objects include cars, trees, buildings, persons, lights, ladders, sidewalks, air conditions, mail box, signs, bicycle, umbrella, etc. In total, there are more than 400,000 local appearance features extracted from these images.

We choose this dataset due to several reasons. First of all, this dataset has high-quality user-generated object segmentation and labeling information. The segmentation and labeling information can be as detailed as parts of the objects, such as the front light of the car, the door of a building, etc. Such detailed labeling information can help to generate high quality side information for learning the distance metric. Secondly, it contains around 495 common objects, which frequently appear in daily life. For each image, there are on average 12 objects positioned and occluded as they used to be in the real world. It is a great challenging for any model to detect and annotate these objects in such a complex situation. Finally, all the images are of high resolution and generated from real world, which can help us to examine the performance and applicability of our technique to real applications.

B. Image Representation

In our experiments, we adopt SIFT to represent the local visual features. For each image, 1,000 SIFT features are extracted in 128-dimensional vector space. We use SIFT for three reasons. Firstly, it is invariant to object scaling, rotation and affine invariance changes, which is relatively more robust than other feature descriptors, especially for object representation. Secondly, SIFT usually performs very well on street scenarios, which accounts for a large portion of images in our dataset. Finally, as the regular BoW model often uses SIFT, we also adopt the same technique to ensure a fair comparison.

C. Experimental Settings

In the experiments, for the regular BoW model, the codebook is generated by performing k-means over all the SIFT features extracted from the training dataset. The centers of the resulting clusters are collected to form a set of k visual words as the codebook, in which each cluster represents one visual word. For the BoW representation, each feature in an image is then mapped to the nearest visual word in the codebook, and finally a visual word histogram can be generated by summarizing the mapping results of all features of the images.

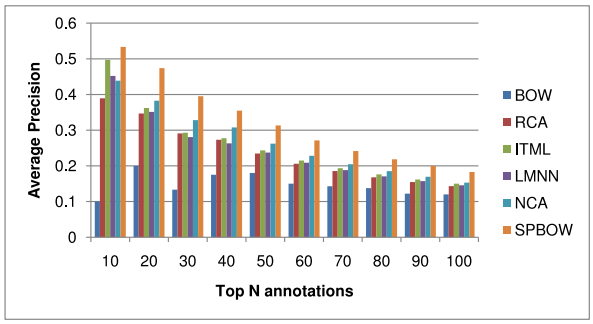
To examine how SPBoW can also benefit from existing DML methods, we implement several different SPBoW methods by adapting four state-of-the-arts metric learning algorithms, including Relevant Component Analysis (RCA) [1], Information Theoretic Metric Learning algorithm (ITML) [7], Large Margin Nearest Neighbor (LMNN) [39], and Neighborhood Components Analysis (NCA) [14]. All of them were implemented in the same experimental settings.

D. Experiment I: Annotation Performance

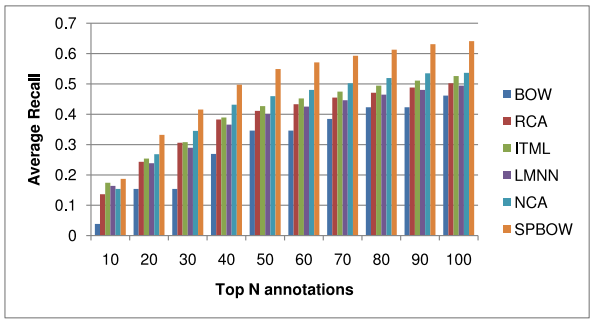
In this experiment, we evaluate the image annotation performance of the proposed techniques. The ground truth was generated by web users from Labelme project [30]. We adopt standard performance metrics, i.e., *Average Precision* (AP@N) and *Average Recall* (AR@N), to evaluate the annotation performance at the top N annotated semantic labels/tags.

In our experiment, we perform 5-fold cross validation, in which 4 folds are used for building the codebook and 1 fold is used for testing the annotation performance. In our methods, there are 2 key parameters: the number of sampled pairwise constraints and the codebook size. In this experiment, we simply fix the constraint size to 10,000 and the codebook size to 2,500. We will examine their effects in subsequent experiments. Fig. 4 shows the comparison results of different approaches, including a regular BoW method and five implementations of SPBoW with different DML techniques.

From Fig. 4, we found that most SPBoW algorithms significantly improve the annotation performance of the regular BoW in both precision and recall. Comparing with other existing DML algorithms, SPBoW with the newly proposed metric learning algorithm also has the significant advantage. These results show that the codebook generated by our SPBoW technique is more discriminative than the regular BoW, and SPBoW is effective in reducing the semantic loss during the codebook generation process.

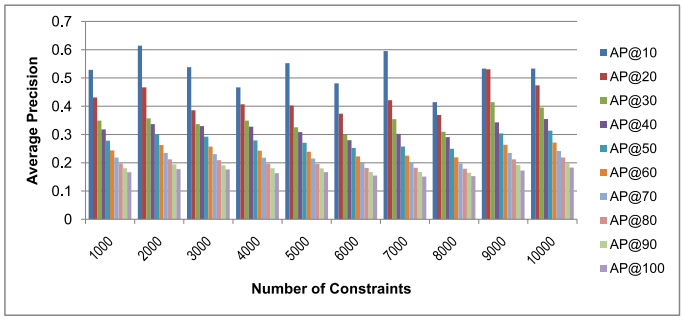


(a) Average Precision

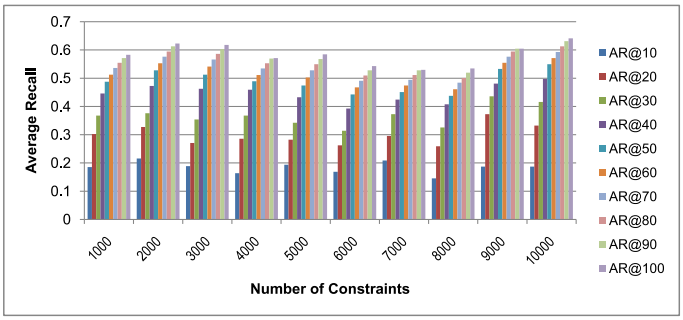


(b) Average Recall

Fig. 4. Performance comparison of different approaches for image annotation



(a) AP@N of SPBoW method under different constraints



(b) AR@N of SPBoW method under different constraints

Fig. 5. Evaluation of constraint sizes on the image annotation performance

E. Experiment II: Evaluation of Varied Constraint Sizes

In this experiment, we study the influence of the number of constraints on the final annotation performance. We sample a certain number of constraints from all the user-generated labels, and gradually increase the number from 1,000 to 10,000 with interval of 1,000 constraints. Under each number of constraints, we evaluate the performance of object annotation by the resulting SPBoW. The average precision and average recall at top N ($N = 10, \cdots, 100$) are summarized in Fig. 5.

From the results, we can see that increasing the number of constraints in general results in the improvements of the annotation performance in terms of both precision and recall. This is reasonable as when more side information is included for the metric learning task, we expect to learn a better metric, which is essential to generate the SPC for the annotation task. In practice, the selection of the number of constraints is a tradeoff between efficacy and efficiency.

F. Experiment III: Evaluation of Different Codebook Sizes

In this section, we evaluate the influence of codebook sizes on the final annotation performance. We gradually increase the codebook size from 2,500 to 4,500, and evaluate the average precision and recall results under each setting of codebook size. Fig. 6 shows the experimental results.

The results show that the codebook size does affect the annotation performance. In particular, we observe that the performance is first improved when increasing the codebook size from $2,500$ to $3,000$, but is degraded when the size is larger than $3,000$. From the empirical results, the best codebook size is around 3,000 on this dataset.

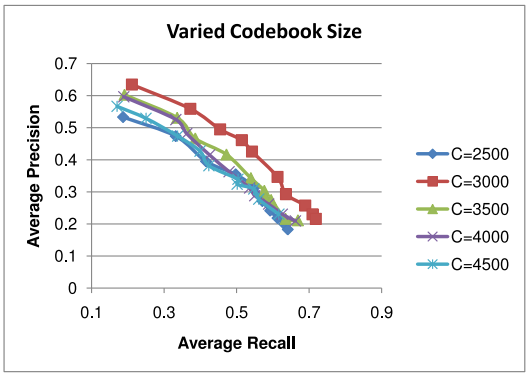
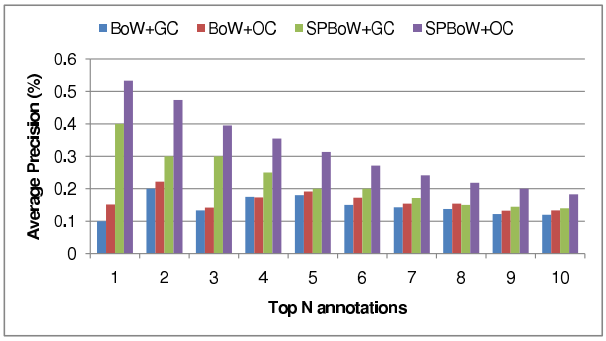


Fig. 6. Evaluation of varied codebook sizes (C) on image annotation.

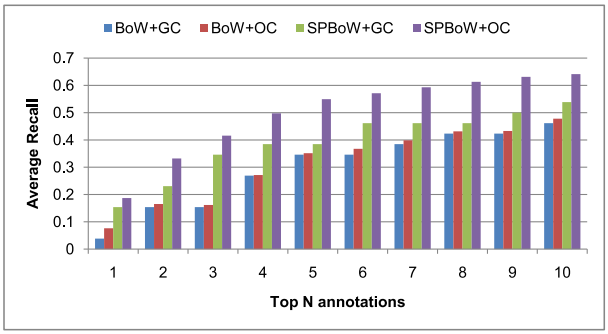
G. Experiment IV: Object Codebook vs. General Codebook

Our SPC solution is in general an object based codebook, which is denoted as "object codebook". Unlike the regular BoW that uses a general codebook without considering specific objects, our object codebook enjoys a number of advantages, such as high efficiency and scalability. In addition, similar to the regular BoW, we can also generate a "general codebook" by applying the similar metric learning in SPBoW. This experiment aims to compare the performance between object codebook and general codebook.

We implement two kinds of SPC. One is an object codebook similar to the previous experiments, and the other is a global SPC similar to the regular BoW except for the usage of the optimized distance metric. Finally, we also include the regular BoW codebook into the comparison. Fig. 7 summarizes the comparison results. We first observe that both SPC approaches perform considerably better than the regular BoW codebook.



(a) Average Precision



(b) Average Recall

Fig. 7. Comparison between general codebook and object codebook.

Further, by comparing the difference between object and global codebooks, we found that both of the two object codebooks consistently surpasses their corresponding global codebooks in all of top annotation results. These results again validate the effectiveness of the SPBoW technique.

Remark. We briefly explain why the object codebook outperforms the global codebook. Firstly, the visual words of the object codebook are obtained by clustering features related to the same semantic concept, and thus they correspond to the same semantic meaning; however, in global codebook, visual words are obtained by clustering features from various semantic concepts, and thus each visual word may relate to multiple semantic meanings. Hence, the object codebook is thus less likely to cause semantic loss. Further, another advantage of the object codebook is that it could be more robust than the global codebook since in the object codebook only the semantics-related features will be engaged for clustering, while for global codebook, all kinds of features including the background features will be engaged for clustering, which thus more likely suffers from noisy background features.

H. Experiment V: Fixed vs. Varied Codebook Sizes

One key step of generating our SPC is the codebook size assignment, which decides how many visual words (codes) should be assigned to each object. In our approach, we have proposed the varied codebook size assignment approach based on the information theory approach. Hence, this experiment aims to examine if the proposed varied code size approach is better than a simple fixed codebook size approach that assigns the *uniform* number of visual words for every object. In this experiment, we fix the total codebook size to 2,500. Fig. 8 shows the experimental results of average precision and recall.

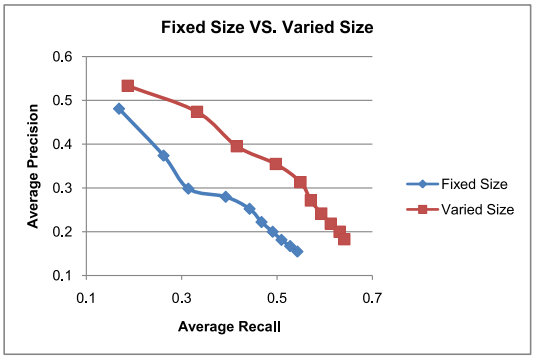


Fig. 8. Comparison between fixed codebook and varied codebook schemes.

The results show that the varied codebook approach outperforms the fixed codebook approach by around 22% on average in terms of both average precision and recall performance.

I. Experiment VI: Application to Object Recognition

In this experiment, we apply the proposed SPBoW on the PASCAL VOC2006 challenge for object recognition to further compare its performance with other algorithms. Note that there are some difference between the VOC2006 dataset and the Labelme dataset. Different from the manually well segmented objects in the Labelme dataset, objects in the VOC2006 data are marked in the images only with a rough bounding box. Also the number of object categories is only 10 for VOC2006 data, which is much smaller than the Labelme dataset. Finally, since these two datasets have different data distributions and different number of categories, we believe using both of them can examine the robustness of our techniques.

We employ the discriminative model for object detection. Specifically, we use all the VOC2006 training data to learn the codebook as well as to train a set of binary SVM classifiers. Each SVM classifier is then used to detect one object category. We then test the performance on the VOC2006 test dataset, and compare with the existing BoW model as well as some state-of-the-art object recognition methods, such as AP06-Lee (Lee et al.) [10], QMUL-LSPCH (Zhang et al.) [10], and XRCE (Perronnin et al.) [29]. We set the codebook size to 500 in the codebook learning process, since there are only 10 different categories. And we use the default SVM settings ($C = 1$) with the RBF kernel of $\gamma = 0.07$. The detection performance is measured by the area under the ROC curve (AUC).

Table I summarizes the AUC results. First, we found that most of the proposed DML based approaches significantly outperform the regular BoW. Second, among different DML approaches, the proposed SPBoW yields the best average performance. Finally, compared to other state-of-the-art approaches, SPBoW also performs the best in most cases.

J. Experiment VII: Evaluation of Computational Cost

This experiment is to evaluate the time cost performance. As we adopt 5-fold cross validation approach, in which 4 folds

Categories	BOW	AP06-Lee	QMUL-LSPCH	XRCE	RCA	ITML	LMNN	NCA	SPBoW
bicycle	56.91	79.10	94.80	94.30	93.45	96.98	94.12	95.34	**99.89**
bus	56.61	63.70	98.10	97.80	97.57	**98.17**	97.79	95.98	97.15
car	60.31	83.30	**97.50**	96.70	94.42	93.17	93.13	93.13	94.54
cat	61.08	73.30	93.70	93.30	92.19	**94.15**	92.97	93.32	93.33
cow	68.53	75.60	93.80	94.00	93.91	92.18	92.77	92.75	**94.18**
dog	73.22	64.40	87.60	86.60	87.77	92.11	90.06	89.97	**94.42**
horse	28.83	60.70	92.60	92.50	93.22	95.58	**96.18**	93.85	95.18
motorbike	36.01	67.20	96.90	95.70	92.19	94.37	94.75	94.19	**96.97**
person	60.78	55.00	85.50	86.30	92.18	93.33	**94.18**	91.31	92.68
sheep	60.74	79.20	95.60	95.10	97.19	97.15	92.39	95.67	**97.44**
Average	56.30	70.15	93.61	93.23	93.41	94.72	93.83	93.55	**95.58**

TABLE I
AUC RESULTS ON THE VOC2006 DATASET.

of the data are used to generate the codebook and 1 fold is used for object annotation, we thus focus on measuring the computational time on codebook generation by the methods. We omit the results of the annotation time cost since they are almost similar for all the compared methods.

Method	BoW	RCA	ITML	LMNN	NCA	SPBoW
Time Cost (s)	121	3	96	1759	457	8

TABLE II
TIME EVALUATION OF CODEBOOK GENERATION BY DIFFERENT METHODS.

Table II shows average computational time for generating the codebook. It consists of time costs of both metric learning and the k-means clustering. 50,000 random features are used to generate 2,500 visual words by k-means algorithm. There are two kinds of codebook generation schemes: the global codebook and the object codebook. The global codebook scheme uses k-means to cluster the 50,000 features into 2,500 clusters. The object codebook scheme generates clusters within each category and then combines them to a codebook of size 2,500. So for each category, we only need to generate around $2,500/|C|$ clusters from around $50,000/|C|$ features, where $|C| = 495$ is the size of the categories. The global codebook scheme requires to compute the distances $2,500 \times 50,000 \sim \mathcal{O}(10^8)$ times per iteration, but the proposed object codebook scheme only needs $2,500/|C| \times 50,000/|C| \times |C| \sim \mathcal{O}(10^5)$ times per iteration. BoW adopts the global codebook, while the other methods employ the object codebook.

From the results, we found that BoW takes even more time than some of the SPBoW models due to the limitation of the global codebook, even it does not have any cost for metric learning. This again shows that the object codebook is not only more effective, but also more efficient than the regular BoW method. Finally, by comparing the time cost between different DML techniques, we can see that our algorithm is comparable to the simple RCA method, and is significantly more efficient than the other state-of-the-art metric learning techniques that are usually computationally intensive.

VII. CONCLUSION

This paper proposed a novel framework of Semantics-Preserving Bag-of-Words (SPBoW) for object representation. Unlike conventional Bag-of-Words (BoW) models that usually suffer from the semantic loss in the codebook generation process, our new technique overcomes this drawback by learning an effective distance metric that aims to bridge the semantic gap between low-level features and high-level semantics. We propose a novel measurement of semantic gap and then try to minimize the gap via distance metric learning. In addition to the new efficient algorithm for solving the challenging distance metric learning task, we also propose the object based codebook generation scheme, which not only improves the efficacy, but also significantly reduces the computational cost. Extensive experiments have been done on both image annotation and object categorization applications, in which encouraging results show that the new SPBoW technique is effective and promising for object representation in a large range of multimedia applications. Future work will study more advanced approaches of improving the estimation of distribution for the measurement of visual complexity, and investigate other distance metric learning techniques for improving the performance.

ACKNOWLEDGEMENTS

The work was supported by Singapore NRF Interactive Digital Media R&D Program, under research grant NRF2008IDM-IDM004-006, MOE tier-1 Research Grant (RG67/07), and the National High Technology Research and Development Program of China(863)(No. 2008AA01Z117).

REFERENCES

[1] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *In Proceedings of the Twentieth International Conference on Machine Learning*, pages 11–18, 2003.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. D. Freitas, D. M. Blei, J. K, T. Hofmann, T. Poggio, and J. Shawe-taylor. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[3] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[4] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[5] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *IEEE CVPR*, pages 163–168, 2005.

[6] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML'07*, pages 209–216, 2007.

[8] P. Duygulu, K. Barnard, J. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, pages 97–112, 2002.

[9] M. Everingham, C. W. A Zisserman, and L. V. Gool. The 2006 pascal visual object classes challenge. In *Workshop in ECCV'06*, 2006.

[10] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[11] J. Fan, Y. Gao, and H. Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *ACM Multimedia*, pages 540–547, 2004.

[12] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS'05*, 2005.

[13] K.-S. Goh, B. Li, and E. Chang. Using one-class and two-class svms for multiclass image annotation. *IEEE Trans. on Knowl. and Data Eng.*, 17(10):1333–1346, 2005.

[14] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighborhood component analysis. In *NIPS*, 2004.

[15] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Mach. Learn.*, 69(2-3):169–192, 2007.

[16] V. Hedau, H. Arora, and N. Ahuja. Matching images under unstable segmentations. In *IEEE CVPR*, pages 1–8, 2008.

[17] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR'99*, pages 50–57, Berkeley, CA, 1999.

[18] S. C. H. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*, June 2008.

[19] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *Proc. CVPR2006*, New York, US, June 17–22 2006.

[20] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proc. 26th ACM SIGIR Conference*, pages 119–126, 2003.

[21] Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proc. 6th ACM Int. Conf. on Image and video retrieval*, pages 494–501, Amsterdam, The Netherlands, 2007.

[22] R. Jin, J. Y. Chai, and L. Si. Effective automatic image annotation via a coherent language model and active learning. In *Proc. 12th ACM International Conference on Multimedia*, pages 892–899, New York, NY, USA, 2004.

[23] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: object localization by efficient subwindow search. In *CVPR*, 2008.

[24] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. volume 2, pages 2169–2178, 2006.

[25] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 911–920, Santa Barbara, CA, USA, 2006.

[26] J. Li, W. Wu, T. Wang, and Y. Zhang. One step beyond histograms: Image representation using markov stationary features. In *IEEE CVPR Conference*, pages 1–8, 2008.

[27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[28] R. Maree, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, pages 34–40, Washington, DC, USA, 2005.

[29] F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *In ECCV*, pages 464–475, 2006.

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

[31] S. Shalev-Shwartz and Y. Singer. Logarithmic regret algorithms for strongly convex repeated games (technical report). *The Hebrew University.*, 2007.

[32] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[33] L. Si, R. Jin, S. C. H. Hoi, and M. R. Lyu. Collaborative image retrieval via regularized metric learning. *ACM Multimedia Systems Journal (MMSJ)*, 12(1):34–44, 2006.

[34] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[35] P. Tirilly, V. Claveau, and P. Gros. Language modeling for bag-of-visual words image categorization. In *Proc. ACM Int. Conf. on Content-based image and video retrieval*, pages 249–258, Niagara Falls, Canada, 2008.

[36] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*, pages 1–8, 2008.

[37] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

[38] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. In *CVPR'06*, pages 1483–1490, 2006.

[39] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems*, 18:1473–1480, 2006.

[40] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu. Distance metric learning from uncertain side information with application to automated photo tagging. In *Proceedings of the seventeen ACM international conference on Multimedia (MM'09)*, pages 135–144, Beijing, China, 2009.

[41] L. Wu, Y. Hu, M. Li, N. Yu, and X.-S. Hua. Scale-invariant visual language modeling for object categorization. *Multimedia, IEEE Transactions on*, 11(2):286–294, Feb. 2009.

[42] L. Wu, M. Li, Z. Li, W.-Y. Ma, and N. Yu. Visual language modeling for image classification. In *Proc. Int. workshop on multimedia information retrieval (MIR'07)*, pages 115–124, Augsburg, Bavaria, Germany, 2007.

[43] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS2002*, 2002.

[44] L. Yang, R. Jin, R. Sukthankar, and Y. Liu. An efficient algorithm for local distance metric learning. In *AAAI*, 2006.

Lei Wu received the Bachelor degree in Special Class for Gifted Young (SCGY) in 2005 from University of Science and Technology of China (USTC), from which he is now pursuing his Ph.D degree in Electronic Engineering and Information Science. From 2006 to 2008, he has been a research intern at Microsoft Research Asia working on image annotation and tagging. From 2008 to 2009, he was visiting Nanyang Technological University working on distance metric learning and multimedia retrieval. His research interests include machine learning, multimedia retrieval, and computer vision. He received Microsoft Fellowship in 2007.

Steven C.H. Hoi is currently an Assistant Professor in the School of Computer Engineering of Nanyang Technological University, Singapore. He received his Bachelor degree in Computer Science from Tsinghua University, Beijing, P.R. China, and his Master and Ph.D degrees in Computer Science and Engineering from Chinese University of Hong Kong. His research interests include statistical machine learning, multimedia information retrieval, Web search and data mining. He is a member of IEEE and ACM.

Nenghai Yu is currently a Professor in the Department of Electronic Engineering and Information Science of University of Science and Technology of China (USTC). He is the Executive Director of MOE-Microsoft Key Laboratory of Multimedia Computing and Communication, and the Director of Information Processing Center at USTC. He graduated from Tsinghua University,Beijing, China, and obtained his M.Sc. Degree in Electronic Engineering in 1992, and then he joined in USTC and worked there until now. He received his Ph.D. Degree in Information and Communications Engineering from USTC, Hefei, China, in 2004. His research interests are in the field of multimedia information retrieval, digital media analysis and representation, media authentication, video surveillance and communications etc. He has been responsible for many national research projects. Based on his contribution, Professor Yu and his research group won the Excellent Person Award and Excellent Collectivity Award simultaneously from the National Hi-tech Development Project of China in 2004. He has contributed more than 80 papers to journals and international conferences.