This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

Entropy Optimized Feature-based Bag-of-Words Representation for Information Retrieval

Nikolaos Passalis and Anastasios Tefas

Abstract—In this paper, we present a supervised dictionary learning method for optimizing the feature-based Bag-of-Words (BoW) representation towards Information Retrieval. Following the cluster hypothesis, which states that points in the same cluster are likely to fulfill the same information need, we propose the use of an entropy-based optimization criterion that is better suited for retrieval instead of classification. We demonstrate the ability of the proposed method, abbreviated as EO-BoW, to improve the retrieval performance by providing extensive experiments on two multi-class image datasets. The BoW model can be applied to other domains as well, so we also evaluate our approach using a collection of 45 time-series datasets, a text dataset and a video dataset. The gains are three-fold since the EO-BoW can improve the mean Average Precision, while reducing the encoding time and the database storage requirements. Finally, we provide evidence that the EO-BoW maintains its representation ability even when used to retrieve objects from classes that were not seen during the training.

Index Terms—Information Search and Retrieval, Dictionary Learning, Entropy Optimization, Image Retrieval, Time-series Retrieval.

1 Introduction

Information retrieval (IR) is the task of retrieving objects, e.g., images, from a database given the user's information need [28]. Early research focused mostly on text retrieval [20], [28], but then quickly expanded to other areas, such as image retrieval [7], and video retrieval [15], since the availability of digital technology led to a great increase of multimedia data. A similar growth in the availability of time series data, i.e., data that are composed of a sequence of measurements, such as medical monitoring data [2], also raised the interest in the retrieval of time series [5], [18].

We can more formally define IR as follows: Given a collection of objects $\mathcal{D} = \{d_1, d_2, ..., d_n\}$ and a query object q, rank the objects in $\mathcal{D}$ according to their relevance to q and select a subset of the most relevant objects. As we already mentioned, the collection can contain any type of objects, such as images, videos, time-series or text documents. However, we focus mainly on image retrieval since it is the most studied and challenging aspect of multimedia information retrieval. This is without loss of generality as the proposed method can be applied to several other types of data, such as video, audio and time-series, with minor modifications. The proposed method cannot be directly used for text retrieval, however it can utilize word-embedding techniques, such as [30], to create optimized representations for text documents.

One of the most important challenges in image retrieval is the so-called *semantic gap* [40], between the low-level representation of an image and its higher level concepts. In other words, the semantic gap describes the phenomenon in which images with similar semantic content have very different low-level representations (e.g., color information) and vice versa. The high dimensionality of images reduces the retrieval performance even more, both in terms of query time and precision.

Several approaches have been proposed to extract meaningful low-dimensional features from images. The purpose of feature extraction is to lower the dimensionality of the images, which reduces the storage requirements and the retrieval time, and to bridge the semantic gap, which increases the retrieval precision. Perhaps the most widely used and successful method for this task is the feature-based Bag-of-Words model [39], also known as Bag-of-Features (BoF) or Bag-of-Visual Words (BoVW). The feature-based BoW approaches, described in detail in Section 3.1, should not be confused with the standard textual BoW approaches [28], that are discussed later on in this Section. In the rest of the manuscript we abbreviate the feature-based Bag-of-Words models as BoW. The BoW model treats each image as a document that contains a number of different "visual" words. Then an image is represented as a histogram over a set of representative words, known as *dictionary* or *codebook*. These histograms describe the corresponding images and they can be used for the subsequent retrieval tasks. The BoW pipeline can be summarized as follows:

1) *feature extraction*, in which multiple features, such as SIFT descriptors [26], are extracted from each image. That way, the *feature space* is formed where each image is represented as a set of features.
2) *dictionary learning*, in which the extracted features are used to learn a dictionary of representative features (also called *words* or *codewords*),
3) *feature quantization* and *encoding*, in which each feature is represented using a codeword from the learned dictionary and a histogram is extracted for each image. That way, the *histogram space* is formed where each image is represented by a constant dimensionality histogram vector.

Unlike textual words, visual words are not predefined and the quality of the extracted representation critically

• *Nikolaos Passalis and Anastasios Tefas are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece. email: passalis@csd.auth.gr, tefas@aiia.csd.auth.gr*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

relies on how these words are chosen. Early feature-based BoW approaches (e.g., [23], [39]) used unsupervised clustering algorithms, such as k-means, to cluster the set of features and learn a dictionary. These unsupervised approaches achieved promising results and produced codebooks that were generic enough to be used for any task. However, learning a discriminative dictionary tailored to a specific problem is expected to perform significantly better.

Indeed, supervised dictionary learning, is shown to achieve superior performance in terms of classification accuracy [13], [24], [25]. These methods produce discriminative dictionaries that are useful for the given classification problem. Even though a highly discriminative representation is desired for classification tasks, it is not always optimal for retrieval since it might severely distort the similarity between images in order to gain discrimination ability. This can be better understood by an example. Suppose that we want to learn a dictionary that distinguishes apples from oranges using only two codewords and a perfect discriminative dictionary learning algorithm exists. After the training process, each apple is represented by a vector $(x, y) \in \mathcal{N}(0, 0.1) \times \mathcal{N}(1, 0.1)$ and each orange by a vector $(x, y) \in \mathcal{N}(1, 0.1) \times \mathcal{N}(0, 0.1)$, where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2. This codebook would be excellent for classification and retrieval of oranges and apples, as it manages to linearly separate the two classes with a large margin. Now, what would happen if we use this representation to retrieve an another fruit, such as pears or bananas? Does this dictionary gained its discrimination ability at the expense of its representation ability and would actually perform worse (than an unsupervised dictionary) on this task? Our experiments (Section 4.2.2) confirm this hypothesis, since we found that a highly discriminative representation excels at its learned domain, but its discriminative ability outside this domain is severely limited. The interested reader is referred to [34], where the problems of *transfer learning* and *domain adaptation* are discussed. We also expect relevance feedback techniques, i.e., methods that are used to better identify the user's information need [10], [11], [45], not to work correctly outside the training domain, since the representation ability is already lost.

Ideally, the aforementioned problem would be solved if we could optimize the learned representation for every possible information need. Since this is rather infeasible, we can learn a representation using a large-scale database with annotated images, such as ImageNet [37], that covers a broad range of information needs. However, it is not always possible to acquire a large set of annotated training data, mostly because of the high cost of annotation. Therefore, a good representation for retrieval should a) improve the retrieval metrics inside its training domain and b) be able to "transfer" the learned knowledge to other similar domains. The latter is especially important as it ensures that we can learn using a small and representative training set.

To this end, we propose a supervised dictionary learning method that produces *retrieval-oriented* codebooks by adhering to the *cluster hypothesis*. Cluster hypothesis states that points in the same cluster are likely to fulfill the same information need [43]. We select a set of centroids in the histogram space and we learn a codebook that minimizes the entropy of each cluster. Each centroid can be considered as a *representative query* and the optimization aims to maximize the relevant information around it. The entropy objective acts as a mild discrimination criterion, which try to make the clusters as pure as possible. To understand why this criterion differs from other more discriminative criteria, such as the Fisher's ratio [13], or max-margin objectives [24], [25], note that we do not push clusters or points away from each other. Instead, the clusters are fitted to the existing data distribution and we only try to move irrelevant points to the nearest relevant cluster. This allows us to optimize the codebook without over-fitting the representation over the training domain. We validate our claims by demonstrating the ability of the proposed method to successfully retrieve images that belong to classes that are not seen during the training process. We should mention that our method is not limited to image retrieval. It is general enough to be applied to any task that involves BoW representations, such as video [15], audio [36], and time series retrieval [12].

The term Bag-of-Words is also used to refer to the natural language processing methods that handle a text document as collection of its words [28]. One of the most widely used such methods uses a term frequency (tf) histogram, which counts the appearances of each word, to represent a document as a vector. In these approaches the words of the dictionary are predefined and cannot be altered. Dictionary learning for this representation aims mainly to prune the dictionary by selecting the most useful features [27], instead of altering the words and extracting a new representation. Although the proposed method cannot be applied when the textual BoW representation is used, with the advent of the word-embedding models, e.g., [30], is possible to map each word to a "meaningful" low-dimensional continuous vector. Then, the proposed method can be used by extracting a feature vector from each word of a text document. A similar approach, described in Section 4.1.1, is used to apply the proposed method to a dataset that contains movie reviews in the form of free text.

The main contribution of this paper is the proposal of a novel retrieval-oriented dictionary learning method which can a) improve the retrieval precision over the state-of-the-art methods, b) reduce the storage requirements and query time by using smaller dictionaries, and c) transfer the learned knowledge to previously unseen classes without retraining.

The rest of the paper is structured as follows. The related work is discussed in Section 2 and the proposed method is presented in Section 3. The experimental evaluation using two multi-class image datasets, the UCR time-series dataset collection, the RT-2k movie review dataset and the KTH video dataset is presented in Section 4. Finally, conclusions are drawn in Section 5.

2 Related Work

Our work concerns supervised dictionary learning for information retrieval. To the best of our knowledge, this is the first attempt to learn a codebook by optimizing a retrieval-oriented objective. The related supervised dictionary learning methods can be divided into two categories according to the used optimization criterion.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

The first category uses a discrimination criterion in the feature space. In [22], the optimization aims to increase the mutual information between each codeword and the feature labels and in [19], to minimize the cluster entropy of each codeword. These methods assume that each feature carries the label of the image in which it appears. This assumption does not always hold and it can negatively affect the quality of the learned codebook. For example, a sky patch may both appear in a sea scene and in a forest scene and it would be wrong to assume that it discriminates these two classes.

The second category uses a discrimination criterion directly in the histogram space to improve the codebook construction. In [24], a maximum margin hyperplane is learned and at the same time the codebook is adjusted in order to maximize this margin. However, since the learned hyperplane can separate only two classes, this approach requires the construction of $C(C-1)/2$ codebooks. This problem is addressed in [25], where a multi-class SVM formulation is used. Other methods optimize the LDA/CDA discrimination criterion [12], [13], or the logistic regression loss [4], in the histogram space. In contrast with the previous category, these methods are usually used for classification instead of retrieval.

Although both our method and [19] use entropy as the objective function, our approach is significantly different. We want to increase the number of relevant histograms around each representative query in the histogram space, while [19] aims to increase the purity of each codeword's cluster in the feature space. Also, our work takes the opposite approach to max-margin techniques [24], [25]. Instead of pushing histograms away from the separating hyperplane, we move them towards their representative query (centroid) and gather them in pure clusters.

Other works in the field of BoW representation focused on improving the extracted features. In [15], and [44], different weighting schemes were proposed and evaluated. A pyramid matching scheme that incorporates spatial information was proposed in [23]. Other extensions to the BoW model that achieve state-of-the-art performance include the vector of locally aggregated descriptors (VLAD) [14], and the vector of locally aggregated tensors (VLAT) descriptors [32]. We should stress that these methods do not improve the learned dictionary. Instead, they improve the extracted representation for a given dictionary.

This work also concerns time series retrieval [5], [18]. The BoW model can be applied to time series by using the method described in [12]: each 2-d (or higher-dimensional) point of a time series is quantized into its nearest codeword and a histogram over the codewords is extracted. Since our method can be readily applied to every BoW representation, we can also optimize time series representations for retrieval.

3 Proposed Method

In this section we present the proposed method for learning entropy-optimized BoW representations. Again, we restrict the presentation of our approach in the context of image retrieval without loss of generality. First, we introduce the used notation and briefly describe the standard BoW model. Then, we formally define the Entropy Optimized BoW (EO-BoW) model and discuss its computational complexity. Finally, we derive a learning algorithm for EO-BoW. The notation used in this paper is summarized in Table 1.

TABLE 1
Used notation

Symbol	Meaning
N	number of images
N_i	number of features of i-th image
N_K	number of codewords
N_C	number of training classes
N_T	number of representative queries
Tp	number of representative queries per class
N_{iters}	number of iterations
g	quantization parameter
m	entropy smoothness parameter
η	learning rate
$\mathbf{V}$	codebook
$\mathbf{v}_k$	k-th codeword
$\mathbf{x}_{ij}$	j-th feature of i-th image
d_{ijk}	distance between $\mathbf{v_k}$ and $\mathbf{x}_{ij}$
$\mathbf{u}_{ij}$	codeword membership vector for $\mathbf{x}_{ij}$
$\mathbf{s}_i$	histogram of i-th image
$\mathbf{q}_i$	cluster membership vector of $\mathbf{s}_i$
$\mathbf{w}_i$	normalized cluster membership vector of $\mathbf{s}_i$

3.1 Standard Feature-based BoW Model

Let N be the number of images that we want to encode using the BoW model. The i-th image is described by N_i feature vectors (e.g., SIFT descriptors [26]): $\mathbf{x}_{ij} \in \mathbb{R}^D$ ($i = 1...N$, $j = 1...N_i$), where D is the dimensionality of the features. The BoW model represents each image using a fixed-length histogram of its quantized features, where each histogram bin corresponds to a codeword. In hard assignment each feature is quantized to its nearest codeword/histogram bin, while in soft-assignment every feature contributes, by a different amount, to each codeword/bin.

In order to learn a codebook we cluster the set of all feature vectors, $\mathcal{S} = \{\mathbf{x}_{ij} | i = 1...N, j = 1...N_i\}$, into N_K clusters and use the corresponding centroids (codewords) $\mathbf{v}_k \in \mathbb{R}^D (k = 1...N_K)$ to form the codebook $\mathbf{V} \in \mathbb{R}^{D \times N_K}$, where each column of $\mathbf{V}$ is a centroid vector. These centroids are used to quantize the feature vectors. It is common to cluster only a subset of $\mathcal{S}$ since this can reduce the training time with little effect on the learned representation. It should be also noted that the codebook is learned only once and then it can be used to encode any image.

To encode the i-th image, we compute the similarity between each feature vector $\mathbf{x}_{ij}$ and each codeword $\mathbf{v}_k$ as:

$$d_{ijk} = exp(\frac{-||\mathbf{v}_k - \mathbf{x}_{ij}||_2}{g}) \in \mathbb{R}$$

The parameter g controls the quantization process: for harder assignment we use a small value, i.e., $g < 0.01$, while for softer assignment we use larger values, i.e., $g > 0.01$. Then, we obtain the l^1 normalized membership vector of each feature vector $\mathbf{x}_{ij}$ by: $\mathbf{u}_{ij} = \frac{\mathbf{d}_{ij}}{||\mathbf{d}_{ij}||_1} \in \mathbb{R}^{N_K}$. This vector describes the similarity of feature vector $\mathbf{x}_{ij}$ to each codebook vector. Finally, we extract the histogram $\mathbf{s}_i$ (or the mean membership vector of its features):

$$\mathbf{s}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{u}_{ij} \in \mathbb{R}^{N_K} \quad (1)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

The histogram $\mathbf{s}_i$ has unit l^1 norm, since $||\mathbf{u}_{ij}||_1 = 1$ for every j. These histograms describe each image and they can be used to classify or retrieve images. Finally, note that the training and the encoding process are unsupervised and no labeled data are required. In the next subsection we describe the proposed approach for using annotated training data to improve the learned codebook.

3.2 Entropy Optimized BoW Model

The goal of the EO-BoW technique is to learn a codebook that minimizes the entropy in the histogram space by using a training set of images, where the i-th image is annotated by a label $l_i \in \{1, ..., N_C\}$ and N_C is the number of training classes. Intuitively, the entropy in the histogram space is minimized when the image histograms are gathered in pure clusters, i.e., each cluster contains images of the same class.

In order to measure the entropy in the histogram space, the $\mathbf{s}_i$ vectors are clustered into N_T clusters. The centroid of the k-th cluster is denoted by $\mathbf{c}_k$ $(k = 1...N_T)$. Then, the entropy of the k-th cluster can be defined as:

$$E_k = -\sum_{j=1}^{N_C} p_{jk} \log p_{jk} \tag{2}$$

where p_{jk} is the probability that an image of the k-th cluster belongs to the class j. This probability is estimated as $p_{jk} = h_{jk}/n_k$, where n_k is the number of image histograms in cluster k and h_{jk} is the number of image histograms in cluster k that belong to class j.

As it was already mentioned, each centroid can be considered as a representative query for which we want to optimize the codebook. According to the cluster hypothesis low-entropy clusters, i.e., clusters that contain mostly vectors from images of the same class, are preferable for retrieval tasks to high-entropy clusters, i.e., clusters that contain vectors from images that belong to several different classes. Therefore we aim to learn a codebook that minimizes the total entropy of a cluster configuration, which is defined as:

$$E = \sum_{k=1}^{N_T} r_k E_k \tag{3}$$

where $r_k = n_k/N$ is the proportion of images in cluster k.

3.2.1 Entropy Optimization

By substituting r_k and p_{jk} into the entropy definition given in (3) we obtain the following objective function:

$$E = -\frac{1}{N} \sum_{k=1}^{N_T} \sum_{j=1}^{N_C} h_{jk} \log p_{jk} \tag{4}$$

We seek to learn a codebook $\mathbf{V}$ that minimizes this objective. However, it is not easy to directly optimize (4) as this function is not continuous with respect to $\mathbf{s}_i$ (each $\mathbf{s}_i$ is a function of $\mathbf{V}$ and therefore (4) is neither continuous with respect to $\mathbf{V}$). To this end, we introduce a continuous smooth approximation of the previously defined entropy. To distinguish the previous definition from the smooth entropy approximation, we call the former *hard entropy* and the latter *soft entropy*.

We define a smooth cluster membership vector $\mathbf{q}_i \in \mathbb{R}^{N_T}$ for each histogram $\mathbf{s}_i$, where

$$q_{ik} = exp(\frac{-||\mathbf{s}_i - \mathbf{c}_k||_2}{m})$$

and the corresponding smooth l^1 normalized membership vector $\mathbf{w}_i$ as:

$$\mathbf{w}_i = \frac{\mathbf{q}_i}{||\mathbf{q}_i||_1} \in \mathbb{R}^{N_T}$$

The parameter m controls the fuzziness of the assignment process: for $m \to 0$ each histogram is assigned to its nearest cluster, while larger values allow fuzzy membership.

Then, we can redefine $n_k = \sum_{i=1}^{N} w_{ik}$ and $h_{jk} = \sum_{i=1}^{N} w_{ik}\pi_{ij}$, where π_{ij} is 1 if the i-th object belongs to class j and 0 otherwise. These modifications lead to a smooth entropy approximation that converges to hard entropy as $m \to 0$.

We can now calculate the derivative of E with respect to $\mathbf{v}_m$ as the product of two other partial derivatives:

$$\frac{\partial E}{\partial \mathbf{v}_m} = \sum_{l=1}^{N} \sum_{\kappa=1}^{N_K} \frac{\partial E}{\partial s_{l\kappa}} \frac{\partial s_{l\kappa}}{\partial \mathbf{v}_m} \tag{5}$$

In order to reduce the entropy in the histogram space we have to shift the histograms $\mathbf{s}_l$ that, in turn, depend on the codebook $\mathbf{V}$. The partial derivative $\frac{\partial E}{\partial \mathbf{s}_l}$ provides the direction in which the histogram $\mathbf{s}_l$ must be moved. Since each codeword $\mathbf{v}_m$ lies in the feature space, the derivative $\frac{\partial s_{l\kappa}}{\partial \mathbf{v}_m}$ projects the previous direction into the feature space. The detailed calculation of these derivatives is given in the Appendix A. The codebook can be optimized using gradient descent, i.e., $\mathbf{V}_{t+1} = \mathbf{V}_t - \eta \frac{\partial E}{\partial \mathbf{V}_t}$, where η is the learning rate.

3.2.2 Learning with the EO-BoW

There are some additional aspects of the learning algorithm that we should consider. First, the codebook could be either initialized at random or by using k-means. We use k-means initialization as we observed that this systematically yields better results in terms of retrieval precision. This can be attributed to the fact that the optimization problem is non-convex and convergence to a global minimum is not guaranteed. The k-means initialization provides a good starting point for the optimization.

We should also consider how to select the centers in the histogram space. It is reasonable to assume that each class should be represented by at least one cluster. Therefore, we use at least N_C centers. However, since the distribution of some classes may be multimodal we can achieve better modeling of the data by using more than one cluster per class. In order to select the centers we can use any clustering algorithm. In this paper we opt for selecting N_{Tp} centers per class by running k-means over the histograms of each class. That way, each class is equally represented and the total number of representative queries is $N_T = N_C N_{Tp}$. This is equivalent to minimizing the following objective function

$$J_{center} = \sum_{i=1}^{N} \sum_{j=1}^{N_C} \sum_{k=1}^{N_{Tp}} \pi_{ij}\theta_{ijk} ||\mathbf{s}_i - \mathbf{c}_k^{(j)}||_2^2. \tag{6}$$

where $\mathbf{c}_k^{(j)}$ is the k-th center of class j and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

$$\theta_{ijk} = \begin{cases} 1 & \text{if the } i\text{-th image belongs to the cluster of } \mathbf{c}_k^{(j)} \\ 0 & \text{otherwise} \end{cases}$$

We can also update the centers after each gradient descent's iteration by re-running k-means. This ensures that the centers "follow" the distribution of the histograms during the optimization. However, the k-means algorithm might converge to different local minima of (6) depending on its initialization. To avoid this phenomenon and ensure smooth convergence, the k-means is initialized using the centers of the previous iteration.

To increase the convergence speed and remove η from the parameter selection procedure we use a variant of backtracking line search [33]. The search starts with an initial (large) estimation of the learning rate η and the learning step is gradually reduced until the objective function decreases. The learning rate is set to $\eta = 1$ before each iteration. The used line search procedure is shown in lines 10-17 of the Algorithm 1.

Finally, we have to choose the softness parameters m and g. The parameter g controls the softness of the quantization process, while m controls the histogram's membership fuzziness and should be set to small enough value in order to closely approximate the hard entropy. Note that very small values of m or g may cause numerical instability issues. A detailed evaluation on how these parameters affect the performance of EO-BoW is presented in Section 4.

3.2.3 Complexity Analysis

The complexity of calculating the soft entropy in the feature space is $O(NN_T^2N_KN_C)$. To compute the gradient (5) $O(NN_K^2N_T^2N_C(N + N_T) + NN_K^2N_LD)$ time is required, since the complexity of computing the gradients $\frac{\partial E}{\partial \mathbf{s}_l}$ and $\frac{\partial s_{Ik}}{\partial \mathbf{v}_m}$ is $O(N_T^2N_KN_C(N + N_T))$ and $O(N_LN_KD)$ respectively, where N_L is the maximum number of features extracted from an image. Since $N > N_T$, the asymptotic complexity of the proposed method becomes $O(N_{iters}(N^2N_K^2N_T^2N_C + NN_K^2N_LD))$, where N_{iters} is the number of iterations. For comparison, the complexity of the k-means algorithm, when used to learn a dictionary, is $O(N_{iters}N_FN_KD)$, where N_F is the number of sampled feature vectors.

Our method requires quadratic time with respect to the number of training data. However, the annotated training data are usually limited, since acquiring them is costly in the first place. Also, it worths noting that the EO-BoW method does not increase the retrieval time as the algorithm runs only once and then the objects are encoded using the learned dictionary without any extra cost. Actually, to the contrary, the EO-BoW has the potential of reducing retrieval time since it can achieve higher mAP with reduced dictionary size. This reduces a) the storage requirements, b) the nearest neighbor search time and c) the encoding time (an image is encoded in $O(N_iN_KD)$ time, where N_i is the number of its features). Finally, the optimization of the EO-BoW can be stopped after each iteration of the algorithm, allowing the method to adapt to the available computational resources.

Fig. 1. EO-BoW Learning Algorithm

Input: A set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ of N training images and their class labels $\mathcal{L} = \{l_1, ..., l_N\}$
Parameters: N_K, m, g, N_{Tp} and N_{iters}
Output: The optimized codebook V

1: **procedure** EO-BOW
2: Initialize V by running k-means on training features
3: $S \leftarrow$ ENCODE(X, V, g)
4: $C \leftarrow []$
5: **for** $i \leftarrow 1; i \leq N_C; i++$ **do**
6: Run k-means on the histograms S that belong to class i
7: Append the new N_{Tp} centers to C
8: **for** $i \leftarrow 1; i \leq N_{iters}; i++$ **do**
9: $grad \leftarrow$ GRADIENT(X, S, Y, V, C, m, g)
10: $\eta \leftarrow 1$
11: $prevEntropy \leftarrow$ENTROPY(S, C, m)
12: **repeat**
13: $newV \leftarrow V - \eta * grad$
14: $newS \leftarrow$ ENCODE$(X, newV, g)$
15: $newEntropy \leftarrow$ENTROPY$(newS, C, m)$
16: $\eta \leftarrow \eta/2$
17: **until** $newEntropy < prevEntropy$
18: $S \leftarrow newS$
19: $V \leftarrow newV$
20: Update centers (lines 5-7)
return V
21: **procedure** ENCODE(X, V, g) **return** the image histograms according to (1)
22: **procedure** GRADIENT(X, S, Y, V, C, m, g) **return** the gradient according to (5)
23: **procedure** ENTROPY(S, C, m) **return** the entropy according to (4)

3.2.4 Implementation Details

The complete proposed algorithm is shown in Algorithm 1. First, we initialize the dictionary using k-means (line 2). Then, we compute the histogram space centers by separately running k-means for each class (lines 3-7). In the subsequent lines (8-20) we implement the gradient descent method with backtracking line search (lines 12-17). After each iteration we update the histogram space centers by running k-means again (line 20).

We have implemented several optimizations to improve the running time of the algorithm. First, the gradient computation is readily parallelized since the inner sum of (5) can be independently computed for each training image. Similarly, we parallelized the image encoding procedure, which is called several times during the optimization. Finally, to further decrease training time we randomly subsample the features of each image. Experimentally, we found that we can decrease the number of features to less than 100 features per image with negligible effect on the final precision.

3.2.5 Toy Example

To better understand how the proposed method works we visualize the image histograms and the objective function during the optimization process for a simple binary problem. We choose two classes (suburb vs. coast) from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

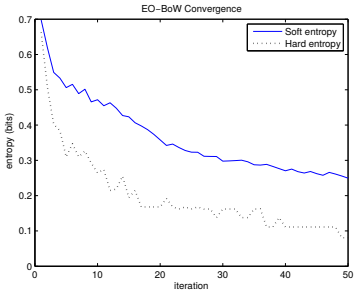

Fig. 2. Soft and hard entropy during the optimization process on a binary problem (suburb vs. coast).

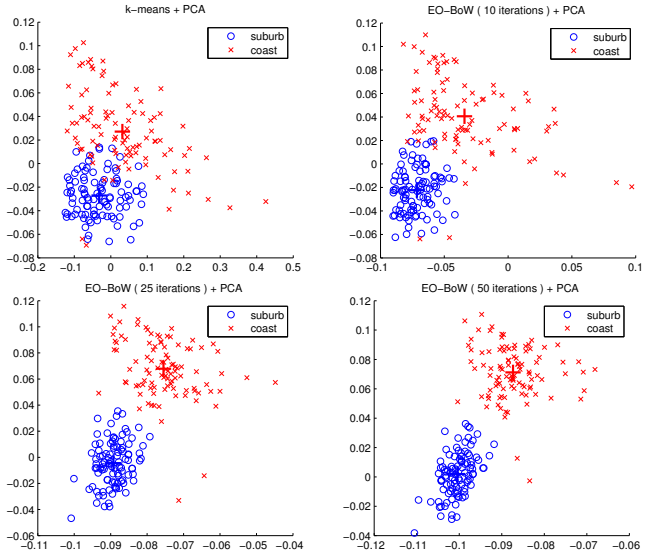

Fig. 3. Image histograms during the optimization process (we plot the first two principal components).

the 15-Scene dataset, which is described in Section 4, and we sample 100 images per class. Also, we set $N_K = 32$ codewords, $N_C = 2$ histogram-space centers (one per class), $g = 0.1$ and $m = 0.01$. Figure 2 plots the objective function. Both the soft entropy and the hard entropy are decreasing in the long term. Since EO-BoW optimizes the soft entropy, it is not expected that the hard entropy will monotonically decrease. Also, the update of histogram centers after each iteration can lead to small increases in soft entropy as well.

In order to visualize the resulting histograms, we project them into a 2-d subspace using PCA [16]. Figure 3 shows the first two principal components of the training histograms during the optimization process. It is evident that the EO-BoW learns a codebook that successfully lowers the entropy in the histogram space since the overlapping of histograms that belong to different classes is gradually reduced and two (almost) pure clusters are formed.

4 Experiments

In this section, we evaluate the proposed method using datasets from various domains and we compare it to other dictionary learning and state-of-the-art representation methods. First, we briefly describe the used datasets, the evaluation metrics and the other evaluated methods. In the next five subsections, we present the performance of our method on each dataset under different evaluation setups. Finally, we validate the statistical significance of our results, we evaluate the effect of parameter selection on the quality of the learned codebook and we present the selected parameters for the conducted experiments.

4.1 Evaluation Setup

4.1.1 Datasets

We evaluate the proposed method using extensive experiments on two multi-class image datasets, the 15-Scene dataset [23], and the COREL CBIR dataset [3], [42], and a collection of time series, the UCR time series collection [17]. Then, we also provide an experimental evaluation of the method using one text dataset, the RT-2k movie review dataset [35] and one multi-class video dataset, the KTH action recognition database [38].

The 15-Scene dataset contains 15 different scene categories: *office*, *kitchen*, *living room*, *bedroom*, *store*, *industrial*, *tall building*, *inside city*, *street*, *highway*, *coast*, *open country*, *mountain*, *forest*, and *suburb*. The total number of images is 4,485 and each category has 200 to 400 images. From each image we densely extract SIFT descriptors of 16 x 16 patches sampled over a grid with spacing of 8 pixels. The COREL Database for Content based Image Retrieval contains 10,800 images from 80 different concepts. We follow the same feature extraction method as in the 15-Scene dataset. Since we use random sub-sampling to create the train and test splits, we repeat each experiment 5 times and report the mean and the standard deviation of each metric.

The UCR time series collection contains 45 different time series datasets from a diverse range of domains, e.g., different types of ECG data, chemical concentrations, star light curves, etc. Some of them are well suited for retrieval tasks, such as the star light curves or the ECG data, while for others the current applications are mostly limited to classification. However, we choose to evaluate the proposed method on every dataset to provide a more comprehensive evaluation. We use the train and test splits that are provided in [17], for each dataset.

The RT-2k [35], is a sentiment dataset that contains 1000 positive and 1000 negative movie reviews in the form of free text. The dataset is used to retrieve reviews with similar sentiment content. This process is also known as *sentiment retrieval* [8]. The performance of the evaluated methods are measured using 10-fold cross-validation. The mean number of the sentences of each movie review is 34.1 $\pm$ 15.5. We followed the feature extraction process proposed in [6], i.e., the RNTN model [41], is used to extract a 25-value sentiment vector from every sentence of each review. The pre-trained RNTN model of the Stanford CoreNLP library [29], is utilized to this end.

The KTH action recognition dataset [38], contains 2391 video sequences of six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping). The dataset is already split to a train+validation set (1528 sequences) and a test set (863 sequences). From each video we extract HOF descriptors [21], and these descriptors are used as feature vectors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

4.1.2 Evaluation Metrics

Throughout this paper, we use three evaluation metrics: precision, recall and mean average precision (mAP). The precision is defined as $Pr(q,k) = \frac{rel(q,k)}{k}$, where k is the number of retrieved objects and $rel(q,k)$ is the number of retrieved objects that belong to the same class as the query q. The recall is similarly defined as $Rec(q,k) = \frac{rel(q,k)}{nclass(q)}$, where $nclass(q)$ is the total number of database objects that belong to the same class as q. We choose to use the interpolated precision, $Pr_{interp}(q,k) = \max_{k',\, k' \geq k} Pr(q,k')$, which is preferred instead of the raw precision since it reduces the precision-recall curve fluctuation [28]. We compute the average precision (AP) for a given query at eleven equally spaced recall points (0, 0.1, ..., 0.9, 1) and the mean average precision as the mean of APs for all queries. We plot two types of curves: the precision-recall curve and the precision-scope curve. The scope refers to the number of objects returned to the user and the precision-scope curve allow us to evaluate the precision at lower recall levels, which is the typical case for an image retrieval system. For the KTH and the RT-2k datasets the precision-recall curves are omitted due to the limited space in the paper.

4.1.3 Baseline and Competitive State-of-the-Art Methods

To learn a baseline dictionary we randomly sample 100 features per image and we use k-means. The clustering process is repeated 5 times and we keep the codebook that yields the lowest reconstruction error. One of the best performing methods for image classification/retrieval is the spatial pyramid matching scheme [23]. Since it can be easily combined with the proposed method (we use the learned codebook for every pyramid level) we evaluate the retrieval performance at three different levels: L = 0 (no pyramid), L = 1 and L = 2. Also, we compare the proposed method to the VLAD method which is among the state-of-the-art representation techniques for image retrieval [14]. We normalize the VLAD vectors to have unit l^2 norm. Furthermore, we compare the EO-BoW against another dictionary learning method [19], abbreviated as 'f-ent', which optimizes the entropy in the feature space instead of the histogram space. We run this method for 100 iterations and we use $m = 0.1$ for both the image datasets. 30 iterations and $m = 0.01$ are used for the KTH dataset and 600 iterations and $m = 0.01$ for the RT-2k dataset. For the UCR datasets we run the method for 100 iterations and we use the same procedure as in EO-BoW (Subsection 4.8) to dynamically select the best value for m. The entropy calculations for the f-ent method [19], are slower than for the EO-BoW, since the number of features are much larger than the number of histograms. Therefore, we can further down-sample the images / time series by selecting 10 features per image / time series (100 features per video are used for the KTH dataset). Experimentally, we found that this reduces the training time without significantly affecting the quality of the learned codebook. For the RT-2k dataset we evaluate the textual BoW method using the term frequency (tf) weighting scheme and a fixed dictionary that is composed of the top 10.000 unigrams appearing in the corpus [28].

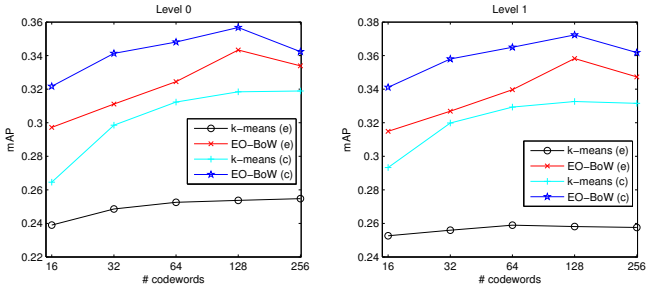

Fig. 4. mAP for different codebook sizes and distance metrics using in-domain evaluation on the 15-Scene dataset.

4.2 15-Scene Dataset

We evaluate the proposed method using two different setups on the 15-Scene dataset. First, we follow the standard evaluation setup [23] and we use 100 images from each category for the training process and the rest of them as testing data. The database consists of the training data and we use the testing data to query the database and evaluate the retrieval performance. We refer to this setup as *in-domain evaluation*.

Then, we use a second evaluation setup, *out-of-domain evaluation*, in which we exclude 5 classes from the training data (suburb, forest, open country, industrial and store) and we evaluate the performance using test queries only from the excluded classes. The database contains images from the excluded classes, but we do not use these images to train the codebook. This setup evaluates the representation ability of the learned codebook, since we retrieve images from classes that are not seen during the training, and corresponds to the realistic scenario in which only a subset of the database images is annotated.

4.2.1 In-domain Evaluation

In Figure 4 we compare the mAP between the proposed method and the k-means method for different codebook sizes and pyramid levels (the results for the pyramid level 2 are omitted since they are similar to those of level 1). Soft quantization is used for both methods. We use two distance metrics for retrieval: the euclidean distance, denoted by '(e)' and the chi-square distance, denoted by '(c)'. We have also evaluated the retrieval performance using two other distance metrics, the histogram intersection and the cosine distance, but we omit them since we observed that the chi-square distance performs better. We observe that the EO-BoW method performs always significantly better than the k-means for any combination of codebook size, distance metric and pyramid level. It worths noting that even if we reduce the size of EO-BoW dictionary by a factor of 8-16, it still performs better than any codebook learned using k-means. That is especially important since using a smaller dictionary decreases the required storage and the retrieval time. The precision gains are larger for the euclidean distance, as we optimize the codebook for it. Nonetheless, the chi-square distance still benefits from the optimization and achieves the best overall mAP.

Next, we use a 128-word dictionary, which provided the best mAP in the previous experiment, and we compare the mAP for different methods. The results are shown in Table 2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

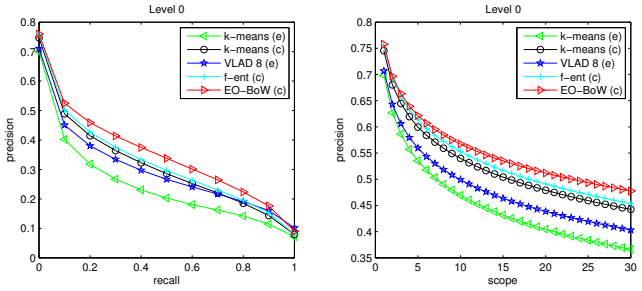

Fig. 5. Precision-recall and precision-scope curves for in-domain evaluation on the 15-Scene dataset.

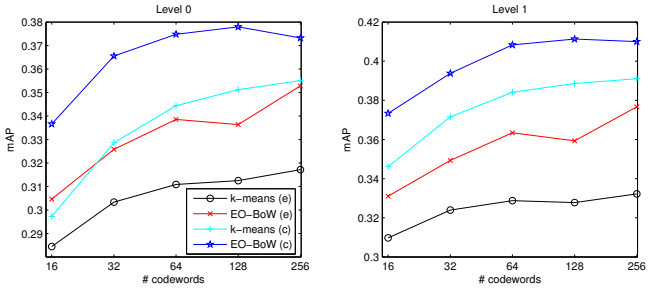

Fig. 6. mAP for different codebook sizes and distance metrics for out-of-domain evaluation on the 15-Scene dataset.

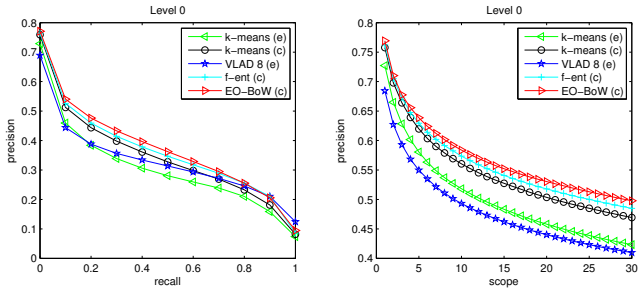

Fig. 7. Precision-recall and precision-scope curves for out-of-domain evaluation on the 15-Scene dataset.

We also report the representation length at each level of the pyramid, since the retrieval time is proportional to it. The f-ent method slightly increases the mAP over the k-means. The VLAD method performs better than the k-means when the euclidean metric is used and slightly better than the k-means with the chi-square distance. We do not use the chi-square distance for retrieval of VLAD vectors since the VLAD representation is not a histogram, but a vector of aggregated reconstruction losses. Indeed, we found that the retrieval results are worse when the chi-square distance was used. The EO-BoW method outperforms all the other competitive methods. Note that the EO-BoW using only a 128-bin histogram (level 0) performs better than almost any other combination of methods. Although the chi-square distance yields the best retrieval results, using the euclidean distance does not harm the mAP too much. Figure 5 plots the precision-recall and precision-scope curves, which confirm the previous findings.

The dimensionality of the image histograms can be reduced using dimensionality reduction techniques, such as PCA [16]. Experimentally we found that reducing the dimensionality of the extracted representation does not significantly affect the mAP. Also, to further increase retrieval speed, approximate nearest neighbor methods [1], [31], [9], can be used.

4.2.2 *Out-of-domain Evaluation*

We also wanted to examine the ability of our method to retrieve classes outside the training domain. Figure 6 compares the mAP for different codebook sizes between the proposed method and the k-means. Even thought the EO-BoW codebook is not optimized for retrieval on the query classes, it does not harm the mAP. Quite the contrary, it actually increases the retrieval precision on the unseen data. In Table 3, we compare the mAP between different methods for the same evaluation setup. The results are similar to those of the Table 2. Although the lead of EO-BoW is somewhat reduced, it still increases the mAP over the other methods. The precision-recall and precision-scope curves are shown in Figure 7. An interesting observation is that the VLAD method is not performing well at low recall levels, but manages to maintain a better precision at higher recall levels. The curves for the level 1 and the level 2 of the SPM are similar to those of the level 0 and, therefore, they are omitted due to space constraints.

To demonstrate the loss of representation ability when a highly discriminative method is used, we also extract a representation using a SVM (chi-square kernel and $C = 50$) trained on the BoW representation learned with k-means (128-word codebook). We train the SVM using the 1-vs-all technique and a vector with class probabilities is extracted from each image. This vector, which can be thought as a histogram over the classes, is used for the retrieval. The results are shown in Table 4. We observe that the SVM representation excels at the learned domain, but its performance drops sharply when used to retrieve images from unseen classes. In fact, even the k-means representation, which was used to train the SVM, achieves better mAP on the unseen data. This experiment confirms the claim that a highly discriminative representation can severely harm the retrieval performance outside the training domain.

4.3 Corel Dataset

In this subsection we repeat the previous experiments using the Corel dataset. We use 6000 randomly selected images to build the database and the rest of them to query the

TABLE 4
Comparing methods with different representation and discrimination ability using the 15-scene dataset.

	In-domain evaluation (mAP)		
	L=0	**L=1**	**L=2**
k-means (c)	31.84 ± 0.17	33.26 ± 0.14	33.41 ± 0.10
EO-BoW (c)	35.68 ± 0.49	37.23 ± 0.40	38.10 ± 0.34
SVM (c)	$\mathbf{46.86 \pm 10.75}$	$\mathbf{63.16 \pm 8.50}$	$\mathbf{73.82 \pm 5.80}$
	Out-of-domain evaluation (mAP)		
	L=0	**L=1**	**L=2**
k-means (c)	35.12 ± 0.49	38.86 ± 0.68	40.50 ± 0.71
EO-BoW (c)	$\mathbf{37.79 \pm 0.11}$	$\mathbf{41.13 \pm 0.44}$	$\mathbf{43.01 \pm 0.53}$
SVM (c)	29.31 ± 4.03	31.77 ± 2.24	34.55 ± 2.31

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

TABLE 2
Comparing EO-BoW to other dictionary learning and representation methods for in-domain evaluation on the 15-Scene dataset.

Method	Dictionary Size	Representation Length	mAP at L=0	mAP at L=1	mAP at L=2
k-means (e)	128	(128, 640, 2688)	25.37 ± 0.37	25.81 ± 0.43	26.07 ± 0.44
k-means (c)	128	(128, 640, 2688)	31.84 ± 0.17	33.26 ± 0.14	33.41 ± 0.10
VLAD (e)	8	(1024, 5120, 21504)	30.45 ± 0.20	34.45 ± 0.37	35.82 ± 0.58
VLAD (e)	16	(2048, 10240, 43008)	29.77 ± 0.23	32.62 ± 0.25	33.04 ± 0.28
f-ent (e)	128	(128, 640, 2688)	23.43 ± 0.30	23.77 ± 0.45	24.16 ± 0.48
f-ent (c)	128	(128, 640, 2688)	32.84 ± 0.24	34.10 ± 0.17	34.24 ± 0.23
EO-BoW (e)	128	(128, 640, 2688)	34.33 ± 0.52	35.83 ± 0.60	36.48 ± 0.63
EO-BoW (c)	128	(128, 640, 2688)	$\mathbf{35.68 \pm 0.49}$	$\mathbf{37.23 \pm 0.40}$	$\mathbf{38.10 \pm 0.34}$

TABLE 3
Comparing EO-BoW to other dictionary learning and representation methods for out-of-domain evaluation on the 15-Scene dataset.

Method	Dictionary Size	Representation Length	mAP at L=0	mAP at L=1	mAP at L=2
k-means (e)	128	(128, 640, 2688)	31.25 ± 0.43	32.78 ± 0.53	33.48 ± 0.56
k-means (c)	128	(128, 640, 2688)	35.12 ± 0.49	38.86 ± 0.68	40.50 ± 0.71
VLAD (e)	8	(1024, 5120, 21504)	33.35 ± 0.40	36.97 ± 0.23	38.19 ± 0.30
VLAD (e)	16	(2048, 10240, 43008)	34.78 ± 0.27	37.07 ± 0.40	37.38 ± 0.46
f-ent (e)	128	(128, 640, 2688)	29.52 ± 0.13	31.12 ± 0.31	31.71 ± 0.35
f-ent (c)	128	(128, 640, 2688)	36.78 ± 0.34	40.35 ± 0.44	41.93 ± 0.54
EO-BoW (e)	128	(128, 640, 2688)	33.64 ± 1.15	35.93 ± 1.20	36.90 ± 1.29
EO-BoW (c)	128	(128, 640, 2688)	$\mathbf{37.79 \pm 0.11}$	$\mathbf{41.13 \pm 0.44}$	$\mathbf{43.01 \pm 0.53}$

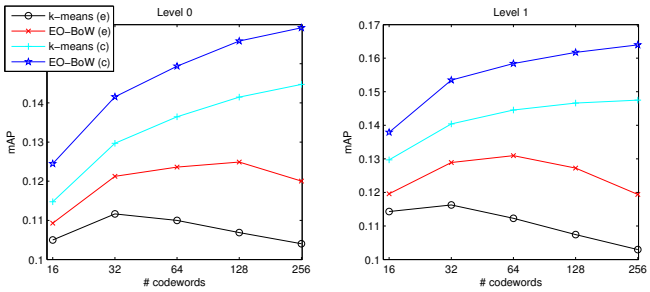

Fig. 8. mAP for different codebook sizes and distance metrics using the Corel dataset.

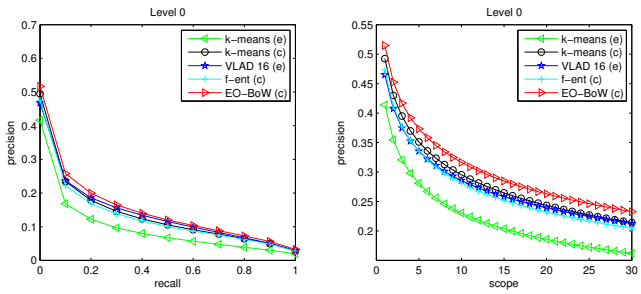

Fig. 9. Precision-recall and precision-scope curves for the Corel dataset.

database. Instead of using a large training set, we choose to use a relative small subset of database images (500 images). Since most of the classes are equally represented in the dataset, we expect that the training set contains 6-7 images per class.

Figure 8 compares the retrieval precision between the k-means and the EO-BoW method for different codebook sizes. Again, we observe that the EO-BoW always increases the mAP and we can reduce the codebook size by a factor of 8 and still exceed the performance of k-means.

In Table 5 we compare the mAP between different methods. The EO-BoW performs better than any other evaluated method. It worths noting that the feature entropy optimization [19], harms the retrieval performance. We attribute this to the (wrong) assumption behind f-ent which states that each feature should carry the label of the image in which it appears. The precision-recall and precision-scope curves are shown in Figure 9. Again, the EO-BoW method achieves better precision than the other methods.

We also wanted to examine the effect of varying the amount of training data on the mAP. To this end, we use three more evaluation setups: in the first we further limit the training data to only 50 and in the others we increase the available training data to 1000 and 3000. We compare the new setups to the best performing methods. The results are shown in Table 6. As it was expected, the increased number of training examples benefits the EO-BoW method. However, this is not true for the f-ent method, which seems to perform better with only 50 training images. In any case, the representation ability of EO-BoW is not reduced since it performs better than the k-means, even when we are using less than 1 training example per class (50 training images). However, in order to match and exceed the performance of the VLAD our method needs a representative enough sample of training data (500 or more training images).

4.4 UCR Datasets

We also evaluate the proposed method using the UCR time series collection. A (one-dimensional) time series is a sequence of numbers $\{t_i\}_{i=1\ldots N_{ts}}$, where N_{ts} is its length. In order to apply the BoW model [12] we can regard each time series as a bag of 2d-points (or feature vectors), $\{(i, t_i)\}_{i=1\ldots N_{ts}}$ that are quantized into their nearest centroid. This allow us to use the proposed method to learn a codebook for time series retrieval. Note that we scale the extracted feature vectors between -1 and 1.

The results of the evaluation are shown in Table 7. Since, the most straightforward way to represent a time series is as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

TABLE 5
Comparing EO-BoW to other dictionary learning and representation methods using the Corel dataset (mAP).

Method	Dictionary Size	Representation Length	mAP at L=0	mAP at L=1	mAP at L=2
k-means (e)	256	(256, 1280, 5376)	10.40 ± 0.26	10.30 ± 0.24	10.07 ± 0.23
k-means (c)	256	(256, 1280, 5376)	14.47 ± 0.19	14.75 ± 0.16	14.57 ± 0.18
VLAD (e)	8	(1024, 5120, 21504)	14.27 ± 0.22	15.82 ± 0.21	15.83 ± 0.20
VLAD (e)	16	(2048, 10240, 43008)	14.77 ± 0.20	15.96 ± 0.22	15.59 ± 0.24
f-ent (e)	256	(256, 1280, 5376)	10.07 ± 0.21	10.38 ± 0.20	10.50 ± 0.21
f-ent (c)	256	(256, 1280, 5376)	13.83 ± 0.18	13.94 ± 0.21	13.66 ± 0.23
EO-BoW (e)	256	(256, 1280, 5376)	12.00 ± 0.28	11.94 ± 0.31	11.66 ± 0.30
EO-BoW (c)	256	(256, 1280, 5376)	$\mathbf{15.92 \pm 0.21}$	$\mathbf{16.40 \pm 0.24}$	$\mathbf{16.24 \pm 0.26}$

TABLE 6
Effect of varying the amount of labeled training data for the Corel dataset.

Method	Dictionary Size	# labeled data	mAP at L=0	mAP at L=1	mAP at L=2
k-means (c)	256	-	14.47 ± 0.19	14.75 ± 0.16	14.57 ± 0.18
VLAD (e)	16	-	14.77 ± 0.20	15.96 ± 0.22	15.59 ± 0.24
f-ent (c)	256	50	13.83 ± 0.13	14.27 ± 0.17	14.23 ± 0.21
f-ent (c)	256	500	13.83 ± 0.18	13.94 ± 0.21	13.66 ± 0.23
f-ent (c)	256	1000	13.62 ± 0.17	13.61 ± 0.15	13.27 ± 0.16
EO-BoW (c)	256	50	14.48 ± 0.27	14.90 ± 0.29	14.76 ± 0.30
EO-BoW (c)	256	500	15.92 ± 0.21	16.40 ± 0.24	16.24 ± 0.26
EO-BoW (c)	256	1000	16.12 ± 0.29	16.58 ± 0.33	16.41 ± 0.31
EO-BoW (c)	256	3000	$\mathbf{16.81 \pm 0.18}$	$\mathbf{17.23 \pm 0.19}$	$\mathbf{17.02 \pm 0.20}$

a vector $\mathbf{t} = (t_1, ..., t_{N_{ts}})$ we include this *raw* representation as baseline. The optimal dictionary size for each dataset is selected using 10-fold cross validation. It should be noted that for some datasets, the f-ent method stopped early due to numerical stability issues. The EO-BoW with chi-square distance achieves higher mAP than any other combination of representation method and retrieval metric for 26 out of 45 datasets, while the EO-BoW with the euclidean distance for 24 datasets. The performance lead of EO-BoW is clear, as no other method wins more than 6 datasets.

It is impractical to plot the precision-recall curves for 45 different datasets. Instead, we compare the precision between two methods, the EO-BoW (c) and the k-means(c), at four different recall points (0, 0.1, 0.2, 0.3). Since we use interpolated precision the precision at recall level 0 is well defined and equals to the maximum precision. The results are shown in Figure 10. The points that fall in the upper left part of the figure correspond to the datasets for which the EO-BoW performs better and the points that fall in the lower right to the datasets for which the k-means method performs better. We also report the number of wins. The EO-BoW provides increasingly better precision for any recall level equal or higher than 0.1. However, the opposite is true for the precision at recall level 0. To understand this behaviour we have to consider how the entropy objective works. The (hard) entropy is minimized when all the points inside a cluster belong to the same class. The distance between two points of different classes does not alter the total entropy as long as each one belongs to a different pure cluster. The same is also true for the soft approximation of the entropy, although the softness of the membership function might mitigate this to some extent. Therefore, the EO-BoW can actually harm the 1-nn accuracy (which is related to the precision at recall 0) since a point that belongs to a wrong cluster contributes much more to the soft entropy than a point that belongs to the right cluster but it is near the cluster boundary.

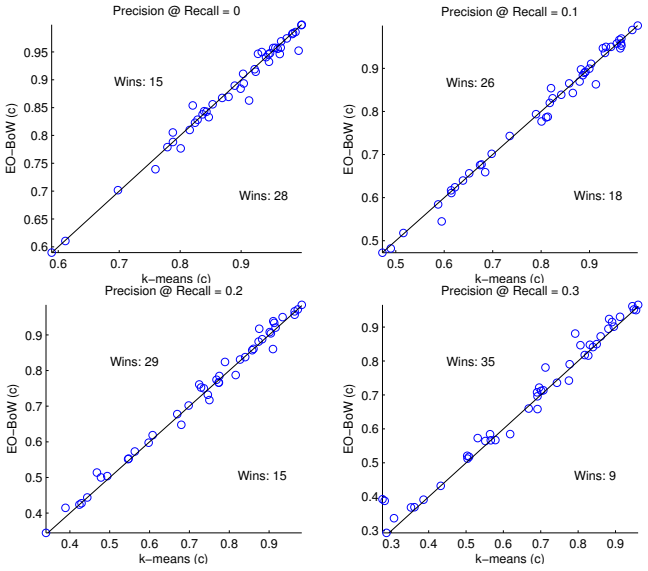

Fig. 10. Comparison of EO-BoW and k-means on the UCR datasets.

4.5 RT-2k Dataset

We also conducted the experiments using the RT-2k dataset. As before, the training split was used to build the database and the test split to query the database. In Table 8 we compare the mAP between different methods. The EO-BoW outperforms the baseline (k-means), as well as the other evaluated methods. We also evaluated these methods for different codebook sizes (the results are omitted due to the limited space) and the same behaviour was observed (the EO-BoW always performs better). The precision-recall curves (not shown) also confirm this. Furthermore, the dictionary which is composed of the top 10.000 unigrams performs significantly worse than the BoW method using the sentiment feature vectors. This is expected, since most unigrams carry mainly topic information and only a small

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

TABLE 7
mAP for the UCR datasets.

	Raw	k-means (e)	k-means (c)	VLAD (e)	f-ent (e)	f-ent (c)	EO-BoW (e)	EO-BoW (c)
50words	41.71	54.77	55.86	45.39	54.35	55.86	55.60	**56.53**
Adiac	41.59	51.36	51.57	42.70	52.03	**52.12**	48.70	48.88
Beef	55.31	54.65	54.71	**57.14**	55.14	54.46	57.10	55.67
CBF	71.33	84.59	84.44	83.20	83.79	83.44	89.35	**89.42**
ChlorineConcentration	**46.25**	45.96	45.91	45.79	45.96	45.91	45.96	45.91
CinC_ECG_torso	**57.44**	56.78	49.95	56.85	54.61	49.75	56.96	52.93
Coffee	68.76	67.31	66.61	**69.78**	67.33	66.62	68.28	67.21
Cricket_X	24.49	26.63	28.12	25.39	26.63	28.03	33.01	**34.26**
Cricket_Y	27.77	31.80	32.33	30.82	31.89	32.14	32.87	**33.29**
Cricket_Z	25.02	27.77	29.05	27.45	27.83	28.78	34.24	**34.98**
DiatomSizeReduction	90.85	87.37	87.72	**93.70**	87.68	87.72	89.17	89.28
ECG200	76.68	76.04	75.95	74.43	75.94	75.84	**77.08**	76.92
ECGFiveDays	67.46	76.35	75.63	77.10	73.90	73.71	**77.15**	76.34
FaceAll	31.54	57.81	57.16	52.17	56.62	56.47	**60.41**	59.10
FaceFour	69.48	79.81	75.96	**85.00**	80.88	76.40	82.41	78.07
FacesUCR	42.35	70.65	69.90	63.37	70.65	69.90	**73.22**	72.37
Gun_Point	70.21	72.86	73.06	**78.04**	72.26	72.30	75.04	75.36
Haptics	34.83	35.73	35.39	**36.09**	34.96	35.15	36.07	35.89
InlineSkate	26.44	28.50	28.49	28.20	28.50	28.49	**29.11**	28.88
ItalyPowerDemand	81.35	77.05	77.32	81.97	77.05	77.32	**82.51**	82.50
Lighting2	64.91	66.56	65.96	68.24	66.56	65.96	**68.40**	67.63
Lighting7	49.39	59.96	60.61	57.52	59.96	60.61	60.45	**61.09**
MALLAT	91.23	89.15	91.62	83.92	89.46	91.74	90.31	**92.36**
MedicalImages	**48.60**	46.91	47.52	46.05	46.91	47.52	47.70	48.19
MoteStrain	77.47	82.70	82.70	82.27	82.70	82.70	**85.86**	**85.85**
NonInvasiveFatalECG_Thorax1	54.25	55.46	**57.21**	48.79	55.46	**57.21**	50.48	53.73
NonInvasiveFatalECG_Thorax2	61.47	65.25	**67.15**	55.25	65.25	**67.15**	58.51	62.49
OSULeaf	32.74	35.21	34.87	31.00	**35.27**	34.83	34.95	34.66
OliveOil	**78.15**	72.66	74.98	71.70	72.66	74.98	72.68	74.97
SonyAIBORobotSurface	66.37	73.62	74.27	69.26	74.08	74.05	77.14	**77.45**
SonyAIBORobotSurfaceII	82.33	80.96	80.86	81.71	80.78	80.38	**83.50**	83.05
StarLightCurves	79.79	82.47	82.36	80.56	82.47	82.36	**82.87**	82.68
SwedishLeaf	45.01	56.26	56.05	48.70	56.30	55.91	**57.63**	57.19
Symbols	86.13	91.96	92.80	85.26	91.57	92.33	92.52	**93.38**
Trace	61.57	65.74	65.99	64.98	65.74	65.99	**72.63**	70.03
TwoLeadECG	68.93	74.00	74.40	68.24	73.10	73.50	79.79	**79.97**
Two_Patterns	35.71	38.16	37.41	37.26	38.09	37.34	**38.54**	37.42
WordsSynonyms	32.55	39.56	40.40	36.05	39.56	40.40	40.41	**41.28**
Fish	45.84	**55.69**	55.09	52.04	52.62	51.77	55.36	54.34
Synthetic_Control	63.17	77.07	79.09	61.88	77.07	79.09	78.34	**80.26**
uWaveGestureLibrary_X	47.93	47.59	48.51	41.96	47.59	48.51	50.94	**51.16**
uWaveGestureLibrary_Y	44.08	45.60	45.57	43.13	45.26	45.37	**46.79**	46.53
uWaveGestureLibrary_Z	44.47	45.23	46.12	42.52	45.34	46.18	48.76	**48.83**
Wafer	90.07	90.45	90.25	90.52	90.19	89.88	93.12	**93.16**
Yoga	58.34	**58.78**	58.58	58.66	58.75	58.53	**58.78**	58.58
Total Wins	4	2	2	6	1	3	15 (24)	**16 (26)**

TABLE 8
Comparing EO-BoW to other dictionary learning and representation methods using the RT-2k dataset (mAP). The (cos) abbreviation refers to the cosine distance.

Method	# codewords	Repres. Length	mAP
unigram-tf (e)	10000	10000	55.84 ± 0.14
unigram-tf (cos)	10000	10000	56.21 ± 0.15
k-means(e)	8	8	72.26 ± 1.03
k-means(c)	8	8	72.73 ± 1.01
VLAD	4	100	61.56 ± 0.47
VLAD	8	200	59.92 ± 0.45
f-ent(e)	8	8	72.35 ± 0.94
f-ent(c)	8	8	73.72 ± 0.97
EO-BoW (e)	8	8	75.02 ± 0.95
EO-BoW (c)	8	8	$\mathbf{75.07 \pm 0.92}$

number of them is associated with the sentiment of the text.

4.6 KTH Dataset

The previous experiments were repeated using the KTH dataset, which is described in Section 4.1.1. The training split was used to build the database and the test split to query the database. In Table 9 we compare the mAP for different methods. The EO-BoW outperforms the baseline (k-means) by more than 14% when the euclidean distance is used, as well as the other evaluated methods. Again, the proposed method allows to reduce the used codebook size without harming the retrieval performance. Note that we set the initial estimation of η to 10 (line 10 of the algorithm in Figure 1), instead of 1, due to the smaller magnitude of the gradients.

4.7 Statistical Significance Analysis

We use Friedman's test to statistically validate the obtained results using all the evaluated datasets. We define the null hypothesis H_0: *There is no difference between the k-means, the VLAD, the f-ent and the EO-BoW methods*. The null hypothesis is rejected ($a = 0.01$). Then, we applied the Nemenyi post hoc test to evaluate the differences between the algorithms. The results are shown in Figure 11. The difference of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

TABLE 9
Comparing EO-BoW to other dictionary learning and representation methods using the KTH dataset (mAP).

Method	# codewords	Representation Length	mAP
k-means(e)	128	128	58.61
k-means(c)	128	128	61.65
k-means(e)	512	512	67.67
k-means(c)	512	512	71.88
VLAD (e)	64	6912	62.49
VLAD (e)	128	13824	62.71
f-ent(e)	128	128	69.93
f-ent(c)	128	128	71.39
EO-BoW (e)	128	128	**73.17**
EO-BoW (c)	128	128	71.74

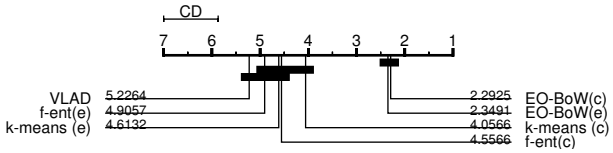

Fig. 11. Nemenyi post hoc test

EO-BoW method (using either the euclidean or the chi-square distance) is statistically significant ($a = 0.01$). We also repeated the same analysis using only the time-series datasets and including the raw time-series representation. Again, the null hypothesis is rejected ($a = 0.01$) and the difference of the EO-BoW is statistically significant.

4.8 Parameter Selection

Finally, we examine the effect of parameter selection on the quality of the learned codebook. The 15-Scene dataset is utilized and the retrieval performance is evaluated for various values of m, g and N_T. We repeat each experiment 5 times and we plot the mean mAP and the mean hard entropy on the training data. The EO-BoW runs for 10 iterations and we down-sample the images by randomly selecting 20 features per image.

First, we fix $g = 0.1$, $N_T = 1 \times Nc = 15$ and vary m from 10^{-3} to 1. Figure 12 shows the effect of m on mAP and hard entropy. The best value of m seems to depend on the size of the learned dictionary. For codebooks smaller than 64 the mAP peaks at $m = 0.01$, while for a codebook with 128 and 256 words the mAP peaks at $m = 0.05$ and $m = 0.001$ respectively. A similar behavior is also observed for the hard entropy. We notice that as the dictionary size increases, the value of m should be gradually reduced. A possible explanation for this behavior is that the increase of dimensionality of the histogram space smooths the membership function, i.e., any histogram belongs to almost every cluster. Decreasing the value of m counteracts this and allows for better optimization.

Next, we use the best values of m, selected according to the previous experiment, and we vary g from $\frac{1}{2}10^{-2}$ to 1. We restrict the lower limit to $\frac{1}{2}10^{-2}$ because numerical stability issues arise for smaller values. The results are shown in Figure 13. The best overall mAP is achieved for $g = 0.1$ and a steep reduction in the hard entropy is observed for the same value regardless the dictionary size. Therefore, we choose to use $g = 0.1$ for this dataset.

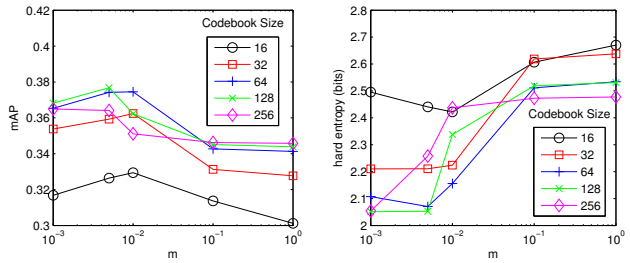

Fig. 12. Effect of m on mAP and hard entropy.

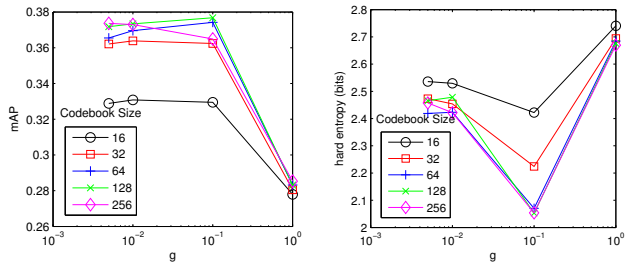

Fig. 13. Effect of g on mAP and hard entropy.

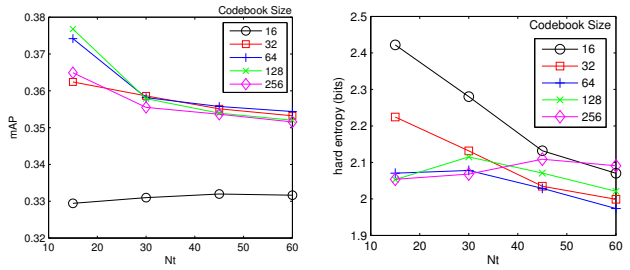

Fig. 14. Effect of N_T on mAP and hard entropy.

Finally, we examine the influence of N_T (number of representative queries) on the optimization process (Figure 14). Since the 15-Scene dataset contains 15 different classes and we equally distribute the centers over these classes, we can use a total number of 15, 30, 45 and 60 centers. As it was expected, using more centers usually lowers the total entropy since each cluster contains fewer points. However, the opposite behavior is observed for the mAP. The best mAP is achieved using only one center per class, except for the 16-word codebook. This result possible suggests that the class distributions are multimodal for the 16-word codebook, but they turn into unimodal as the dimensionality of the histograms increases. Nonetheless, we cannot draw this conclusion without further statistical testing.

The experiments were also repeated for the Corel, the RT-2k and the KTH dataset and the selected parameters are summarized in Table 10. For the UCR datasets we choose $g = 0.1$, $N_T = N_C$ and we dynamically select the value of m that minimizes the entropy for each training set. In order to reduce the computational time during the codebook training process we randomly sample 100 features from each image/time series/video. Then, we run the EO-BoW method for 100 iterations (30 for the KTH dataset, 600 for the RT-2k dataset). After the training phase, we use all the features to extract the representation of each object. Finally, we update the histogram-space centers (line 20 of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2545657, IEEE Transactions on Knowledge and Data Engineering

TABLE 10
Selected parameters for the used datasets.

Parameter	Dict. Size	15-Scene	Corel	KTH	RT-2k
m	≤ 32	0.01	0.01	-	0.01
m	≤ 128	0.005	0.005	0.01	-
m	≥ 256	0.001	0.005	-	-
g	every	0.1	0.05	0.05	1
N_T	≤ 16	3×15	1×80	-	2
N_T	> 16	1×15	1×80 5	5	-

Algorithm 1) after each iteration for the 15-Scene and the Corel datasets. However, we do not update the histogram-space centers for the UCR and the RT-2k datasets, since we observed that this slightly harms the retrieval precision in some cases. Avoiding the histogram-space center updates can act as a form of regularization that prevents overfitting to the training data.

5 Conclusion

In this paper we proposed a supervised dictionary learning method, the EO-BoW, which optimizes a retrieval-oriented objective function. We demonstrated the ability of the proposed method to improve the retrieval performance using two image datasets, a collection of time-series datasets, a text dataset and a video dataset. First, for a given dictionary size, it can improve the mAP over the baseline methods and other state-of-the-art representations. Second, these improvements allow us to use smaller representations which readily translates to lower storage requirements and faster retrieval. In exchange for these, our method requires a small set of annotated training data. Although the gained performance is correlated to the size and the quality of the training dataset, we showed that the proposed method does not lose its representation ability even when a small training dataset is used. Finally, we demonstrated that the EO-BoW improves the retrieval performance using two different similarity metrics, the euclidean and the chi-square distance. Therefore, it can be combined with any approximate nearest neighbor technique that works with these similarity metrics to further increase the retrieval speed.

Appendix A

In this Appendix we calculate the derivatives of (5). First, we compute the histogram space derivative:

$$\frac{\partial E}{\partial s_{l\kappa}} = -\frac{1}{N}\sum_{k=1}^{N_T}\sum_{j=1}^{N_C}\frac{\partial(h_{jk}\log p_{jk})}{\partial s_{l\kappa}}$$

$$= -\frac{1}{N}\sum_{k=1}^{N_T}\sum_{j=1}^{N_C}\log p_{jk}\frac{\partial h_{jk}}{\partial s_{l\kappa}} - \frac{1}{N}\sum_{k=1}^{N_T}\sum_{j=1}^{N_C}h_{jk}\frac{\partial \log p_{jk}}{\partial s_{l\kappa}}$$

where $\sum_{k=1}^{N_T}\sum_{j=1}^{N_C} h_{jk}\frac{\partial \log p_{jk}}{\partial s_{l\kappa}} = \sum_{j=1}^{N_C}\sum_{k=1}^{N_T}\frac{h_{jk}}{p_{jk}\ln 2}\frac{\partial p_{jk}}{\partial s_{l\kappa}} = \sum_{j=1}^{N_C}\sum_{k=1}^{N_T}\frac{n_k}{\ln 2}\frac{\partial p_{jk}}{\partial s_{l\kappa}} = 0$, since $\sum_{k=1}^{N_T} n_k p_{jk} = \sum_{k=1}^{N_T} h_{jk}$ is constant and equal to the number of the images that belong to the class j. By taking the partial derivative with respect to $s_{l\kappa}$ we have $\sum_{k=1}^{N_T} n_k \frac{\partial p_{jk}}{\partial s_{l\kappa}} = 0$.

Also, we have $\frac{\partial h_{jk}}{\partial s_{l\kappa}} = \sum_{i=1}^{N}\pi_{ij}\frac{\partial w_{ik}}{\partial s_{l\kappa}} = \pi_{lj}\frac{\partial w_{lk}}{\partial s_{l\kappa}}$, since $\frac{\partial w_{ik}}{\partial s_{l\kappa}} = 0$ for $i \neq l$.

Therefore, the histogram space derivative is

$$\frac{\partial E}{\partial s_{l\kappa}} = -\frac{1}{N}\sum_{k=1}^{N_T}\sum_{j=1}^{N_C}\log p_{jk}\pi_{lj}\frac{\partial w_{lk}}{\partial s_{l\kappa}}$$

where $\frac{\partial w_{lk}}{\partial s_{l\kappa}} = -\frac{w_{lk}}{m}\left(\frac{s_{l\kappa}-c_{k\kappa}}{||\mathbf{s}_l-\mathbf{c}_k||_2} - \sum_{k'=1}^{N_T} w_{lk'}\frac{s_{l\kappa}-c_{k'\kappa}}{||\mathbf{s}_l-\mathbf{c}_{k'}||_2}\right)$ Then, we derive the feature space projection derivative (last term of (5)):

$$\frac{\partial s_{l\kappa}}{\partial \mathbf{v}_m} = \frac{1}{N_l}\sum_{j=1}^{N_l}\frac{\partial u_{lj\kappa}}{\partial \mathbf{v}_m}$$

where $\frac{\partial u_{lj\kappa}}{\partial \mathbf{v}_m} = \frac{\partial u_{lj\kappa}}{\partial d_{ljm}}\frac{\partial d_{ljm}}{\partial \mathbf{v}_m} = -\frac{1}{g}u_{ljm}(\delta_{\kappa m}-u_{lj\kappa})\frac{\mathbf{v}_m-\mathbf{x}_{lj}}{||\mathbf{v}_m-\mathbf{x}_{lj}||_2}$ and $\delta_{\kappa m} = \begin{cases}1 & \text{if } \kappa = m\\ 0 & \text{otherwise}\end{cases}$

Note that the feature space derivative does not exist when a codebook center and a feature vector coincide (since in that case $\frac{\partial d_{ljm}}{\partial \mathbf{v}_m} \rightarrow \infty$). The same holds for the histogram space derivative when a histogram and a centroid also coincide. When that happen, we set the corresponding derivatives to 0.

References

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *47th Annual IEEE Symposium on Foundations of Computer Science*, 2006, pp. 459–468.

[2] M. M. Baig, H. Gholamhosseini, and M. J. Connolly, "A comprehensive survey of wearable and wireless ECG monitoring systems for older adults," *Medical & Biological Engineering & Computing*, vol. 51, no. 5, pp. 485–495, 2013.

[3] W. Bian and D. Tao. (2009) The COREL database for content based image retrieval. [Online]. Available: https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval

[4] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2559–2566.

[5] F. K.-P. Chan, A. W.-C. Fu, and C. Yu, "Haar wavelets for efficient similarity search of time-series: with and without time warping," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 686–705, 2003.

[6] D. Chatzakou, N. Passalis, and A. Vakali, "Multispot: Spotting sentiments with semantic aware multilevel cascaded analysis," in *Big Data Analytics and Knowledge Discovery*, 2015, pp. 337–350.

[7] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, 2008.

[8] K. Eguchi and V. Lavrenko, "Sentiment retrieval using generative models," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 345–354.

[9] D. Gorisse, M. Cord, and F. Precioso, "Locality-sensitive hashing for chi2 distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 402–409, 2012.

[10] X. He, D. Cai, and J. Han, "Learning a maximum margin subspace for image retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 189–201, 2008.

[11] S. C. Hoi, M. R. Lyu, and R. Jin, "A unified log-based relevance feedback scheme for image retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 509–524, 2006.

[12] A. Iosifidis, A. Tefas, and I. Pitas, "Multidimensional sequence classification based on fuzzy distances and discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2564–2575, 2013.

[13] ——, "Discriminant bag of words based representation for human action recognition," *Pattern Recognition Letters*, vol. 49, pp. 185–192, 2014.

[14] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

[15] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM international conference on Image and Video retrieval*, 2007, pp. 494–501.

[16] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. New York: Springer-Verlag, 2002.

[17] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana. (2006) The UCR time series classification/clustering homepage. [Online]. Available: http://www. cs. ucr. edu/~ eamonn/time_series_data

[18] E. J. Keogh and M. J. Pazzani, "Relevance feedback retrieval of time series data," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 183–190.

[19] Y. Kuang, M. Byrod, and K. Astrom, "Supervised feature quantization with entropy optimization," in *IEEE International Conference on Computer Vision Workshops*, 2011, pp. 1386–1393.

[20] W. Lam, M. Ruiz, and P. Srinivasan, "Automatic text categorization and its application to text retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 6, pp. 865–879, 1999.

[21] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[22] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1294–1309, 2009.

[23] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[24] X.-C. Lian, Z. Li, B.-L. Lu, and L. Zhang, "Max-margin dictionary learning for multiclass image categorization," in *Proceedings of the 11th European Conference on Computer Vision*, 2010, pp. 157–170.

[25] H. Lobel, R. Vidal, D. Mery, and A. Soto, "Joint dictionary and classifier learning for categorization of images using a max-margin framework," in *Proceedings of the 6th Pacific-Rim Symposium*, 2014, pp. 87–98.

[26] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE international conference on Computer vision*, vol. 2, 1999, pp. 1150–1157.

[27] R. E. Madsen, S. Sigurdsson, L. K. Hansen, and J. Larsen, "Pruning the vocabulary for better context recognition," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, 2004, pp. 483–488.

[28] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st ed. Cambridge: Cambridge University Press, 2008.

[29] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[31] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application*, 2009, pp. 331–340.

[32] R. Negrel, D. Picard, and P.-H. Gosselin, "Web-scale image retrieval using compact tensor aggregation of visual descriptors," *MultiMedia, IEEE*, vol. 20, no. 3, pp. 24–33, 2013.

[33] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. New York: Springer-Verlag, 2006.

[34] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[35] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, 2004, p. 271.

[36] M. Riley, E. Heinen, and J. Ghosh, "A text retrieval approach to content-based audio retrieval," in *Proceedings of the 9th International Conference on Music Information*, 2008, pp. 295–300.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, pp. 1–42, 2014.

[38] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, 2004, pp. 32–36.

[39] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[40] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[41] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, vol. 1631, 2013, p. 1642.

[42] D. Tao, X. Tang, X. Li, and Y. Rui, "Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 716–727, 2006.

[43] E. M. Voorhees, "The cluster hypothesis revisited," in *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, 1985, pp. 188–196.

[44] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, 2007, pp. 197–206.

[45] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia systems*, vol. 8, no. 6, pp. 536–544, 2003.

Nikolaos Passalis obtained his B.Sc. in informatics in 2013 and his M.Sc. in information systems in 2015 from Aristotle University of Thessaloniki, Greece. He is currently pursuing his PhD studies in the Artificial Intelligence & Information Analysis Laboratory in the Department of Informatics at the University of Thessaloniki. His research interests include machine learning, computational intelligence and information retrieval.

Anastasios Tefas received the B.Sc. in informatics in 1997 and the Ph.D. degree in informatics in 2002, both from the Aristotle University of Thessaloniki, Greece. Since 2013 he has been an Assistant Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 2008 to 2012, he was a Lecturer at the same University. From 2006 to 2008, he was an Assistant Professor at the Department of Information Management, Technological Institute of Kavala. From 2003 to 2004, he was a temporary lecturer in the Department of Informatics, University of Thessaloniki. From 1997 to 2002, he was a researcher and teaching assistant in the Department of Informatics, University of Thessaloniki. Dr. Tefas participated in 12 research projects financed by national and European funds. He has co-authored 64 journal papers, 161 papers in international conferences and contributed 8 chapters to edited books in his area of expertise. Over 3000 citations have been recorded to his publications and his H-index is 28 according to Google scholar. His current research interests include computational intelligence, pattern recognition, statistical machine learning, digital signal and image analysis and retrieval and computer vision.n.