Information Clustering using Manifold-based Optimization of the Bag-of-Features Representation

Nikolaos Passalis and Anastasios Tefas

Abstract—In this paper a manifold-based dictionary learning method for the Bag-of-Features (BoF) representation optimized towards information clustering is proposed. First, the spectral representation, which unwraps the manifolds of the data and provides better clustering solutions, is formed. Then, a new dictionary is learned in order to make the histogram space, i.e., the space where the BoF historgrams exist, as similar as possible to the spectral space. The ability of the proposed method to improve the clustering solutions is demonstrated using a wide range of datasets: two image datasets, the 15-scene dataset and the Corel image dataset, one video dataset, the KTH dataset, and one text dataset, the RT-2k dataset. The proposed method improves both the internal and the external clustering criteria for two different clustering algorithms, the k-means and the spectral clustering. Also, the optimized histogram space can be used to directly assign a new object to its cluster, instead of using the spectral space (which requires re-applying the spectral clustering algorithm or using incremental spectral clustering techniques). Finally, the learned representation is also evaluated using an information retrieval setup and it is demonstrated that improves the retrieval precision over the baseline BoF representation.

Index Terms—Information Clustering, Dictionary Learning, Manifold-based Optimization, Spectral Clustering, Bag-of-Features Representation.

1 Introduction

Clustering is the task of grouping a set of objects into groups, also known as *clusters* [1]. Each object should be similar to the objects of its cluster and dissimilar to the objects of the other clusters. Several tasks require the clustering of different types of objects that can range from text documents [2], images [3], and video [4], to time-series [5], and facial data [6]. As a result, a rich literature exists in the field of clustering [7], [8]. Perhaps the most commonly used clustering algorithm is the k-means algorithm [9], and its extensions [2], [10], [11].

A more advanced technique, that is also capable of exploiting the manifold structure of the data, is the *spectral clustering* [12]. The term manifold is used to refer to the topological spaces that are locally Euclidean, but might exhibit more complicated global structure. For example, a circle forms a one-dimensional manifold embedded in a two dimensional space. In the spectral clustering the *similarity matrix* of the data, i.e., the matrix that describes the similarity between every pair of objects, is constructed and then its spectrum, i.e., its eigenvalues, is used to create a new low-dimensional representation of the data, also called *spectral representation*. Then, any clustering algorithm, such as the k-means, can be used to cluster the data. This spectral representation captures the local properties of the input (original) space allowing to extract the manifold structures of the data. As a result, better clustering solutions can be obtained by using the spectral representation instead of the original representation.

In real world problems the data are usually complex objects (e.g., images, video, text documents, etc.) that can be represented as a collection of feature vectors. These feature vectors can be either extracted using handcrafted extractors, such as HoG [13], HoF [14], or by using trainable feature extractors such as convolutional neural networks [15]. The interested reader is referred to [16] for an extensive review of the feature extraction methodologies.Note that a different number of feature vectors might be extracted from each image. Clustering such type of objects usually requires the extraction of an appropriate representation (information), since it is not straightforward how to directly cluster collections of feature vectors. Therefore, the term *information clustering* is used to refer to the clustering of the extracted information, while the term *data clustering* is used to refer to the clustering of the feature vectors (raw data).

The most commonly used approach to create a constant-length representation of such objects is the *Bag-of-Features* (BoF) technique [17], also known as Bag-of-Visual-Words (BoVW) when applied in the context of images. The BoF model considers each object as a document that contains a number of different words. Each object is then represented as a histogram over a set of representative words, known as *dictionary* or *codebook*. The histogram reprensetation can be then used for the subsequent classification, retrieval or clustering tasks. The BoF model is composed of the following:

1) *feature extraction*, in which multiple feature vectors, e.g., HoG descriptors, are extracted from each object, e.g., an image. The feature vectors lie in the *feature space*, where each object is represented as a set of features.
2) *dictionary learning*, in which the extracted feature vectors are utilized to learn a dictionary of representative features (also called *codewords*),

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

3) *feature quantization* and *encoding*, in which each feature vector is represented using a codeword and a histogram is compiled for each object. The histograms lie in the *histogram space* (also known as *BoF space*), where each object is represented by a constant dimensionality histogram vector

The application of the BoF model is not restricted to image representation. Several types of objects, such as video [18], audio [19], and time-series [20], can be represented using the BoF model. For example, HoG or HoF features can be extracted from a video [14], while each multi-dimensional point of a timeseries can be regarded as a feature vector [20]. After the feature extraction step, the BoF encoding process proceeds as usual by learning a dictionary and compiling a histogram for each object. The BoF model can be considered as an information extraction procedure that takes the raw data (feature vectors) and extracts the useful information (histogram vector).

The early BoF approaches (e.g., [17], [21]) used data clustering algorithms, such as the k-means algorithm, to learn a "generic" dictionary that minimizes the reconstruction loss of the quantized feature vectors. The term generic is used to indicate that the dictionary is not optimized for a specific task. Although these methods achieved promising results it was later established that supervised dictionary learning (e.g., [20], [22], [23], [24]), that optimizes the dictionary towards a specific task, performs significantly better. The success of the supervised dictionary learning methods for classification, which proves that a generic dictionary is not optimal for every task, raises the following questions: Is a generic dictionary optimal for information clustering tasks? If not, is it possible to optimize a dictionary towards information clustering instead of classification? Since no supervised information, i.e., labels, is available for clustering tasks, the dictionary learning procedure must rely on a different source of information. The structure of the histogram space can be exploited to this end. For example, two points that are close in the histogram space are more likely to belong to the same class/cluster, while this is less probable for two distant points. The data in the histogram space can also embed manifolds (this hypothesis is confirmed in Section 4 using four different datasets). Using spectral clustering allows to discover these manifolds and to optimize the histogram space by a learning an appropriate dictionary that unwraps them. The optimized histogram space is expected to provide better clustering solutions. The motivation behind the proposed optimization method is illustrated in the toy example of Figure 1. In the upper left subfigure the initial histogram space is depicted, where two clusters in the form of two (distorted) half moons exist. The rest of the subfigures shows three different ways to unwrap the manifolds that exist in the original space. Note that the histogram space is gradually restructured in a way that allows the easier identification of these clusters by a centroid-based approach, such as the k-means. This toy example is further discussed in Section 3.3.2.

The main contribution of this paper is the proposal of a dictionary learning method, called Manifold Optimized BoF (MO-BoF), for optimizing the BoF representation towards information clustering by exploiting the manifold structure

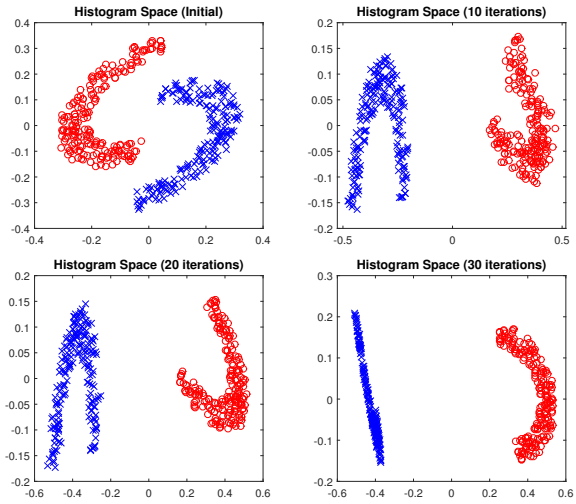
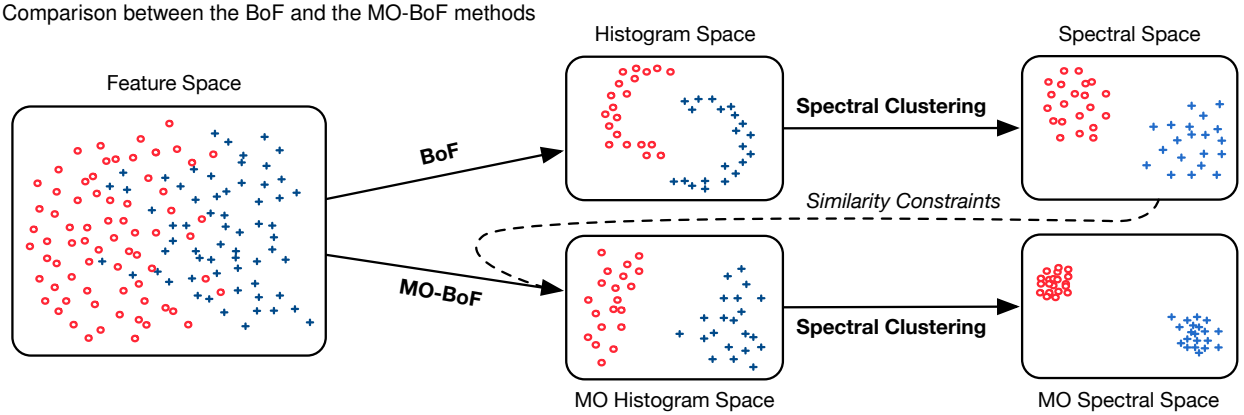


Fig. 1. Histogram space during the optimization using the Manifold Optimized BoF method

of the histogram space. This work focuses on information clustering altering the information extraction procedure, i.e., the BoF model, in a way that makes the extracted information more suitable for clustering. Spectral clustering is used to form the spectral representation of the data, which is better suited for clustering, and then a dictionary is learned in order to make the manifold optimized (MO) histogram space as similar as possible to the spectral space (by enforcing similarity constraints according to the spectral space). It is expected that the MO histogram space will perform better for clustering tasks, since it mimics the spectral space, which is better suited for clustering that the initial histogram space. The proposed method is illustrated and compared to the BoF method in Figure 2. The manifolds are unwrapped in the spectral space (using Spectral Clustering) and then an optimized histogram space is learned using similarity constraints from the spectral space. The proposed method can increase both the internal and the external clustering criteria for different clustering algorithms and a diverse collection of datasets (images, video and text). Also, a limitation of the spectral clustering is the need to recluster all the objects (or use incremental techniques, such as [25], [26]), when a new object arrives. The proposed method can overcome this limitation by constructing a histogram space similar to the spectral space. Then, the cluster of any new object can be quickly and accurately identified in the histogram space, instead of using the spectral space. This provides a straightforward out-of-sample extension for spectral clustering. Although the proposed method is oriented towards clustering its unsupervised nature allows to be used for other tasks too, such as information retrieval. Therefore, the proposed method is evaluated using two different experimental setups: a clustering setup and a retrieval setup.

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 image datasets, the 15-scene recognition dataset and the Corel database for content based image retrieval, a video dataset, the KTH activity recognition dataset, and a

Fig. 2. Comparison between the BoF and the MO-BoF methods



text dataset, the RT-2k movie review dataset, is presented in Section 4. Finally, conclusions are drawn and future research directions are discussed in Section 5.

2 Related Work

The proposed method concerns both spectral clustering and dictionary learning for the BoF representation. Several spectral clustering algorithms have been proposed [12], [27], [28], and any of them can be used in conjunction with the method proposed in this paper. However, an extensive review of the spectral clustering approaches is out of the scope of this paper. The unnormalized spectral clustering algorithm, as described in [12], is utilized in this work and it is presented in detail in Section 3.

A rich litetature also exists in the field of supervised dictionary learning for the BoF representation. The related methods can be classified into two categories according to the used optimization objective. The methods of the first category assume that the feature vectors of an object carry the same label as the object and set the optimization objective in the feature space [29], [30], [31]. The methods of the second category tie the classifier and the codebook learning and rely on the classifier's decisions to optimize the codebook. In [22], and [23], max-margin formulations are used to learn the codebook, while in [32], a multilayer perceptron (MLP) is used to backpropagate the error to the dictionary. In [33], multiple dictionaries with complementary discriminative information are learned. Some other approaches used a discriminative criterion in the histogram space instead of a classifier to optimize the codebook. These methods focused on learning class-specific dictionaries [34], [35], or adjusting the dictionary in order to increase a specific objective, such as the mutual information [36], or the ratio of intra/inter class variation [20], [24].

All the methods presented above are supervised, i.e., they assume that labels are available for the objects during the dictionary learning, and optimize the dictionary towards classification. Little work has been done in the area of dictionary learning for clustering. In [37], and [38], sparse coding is used and several dictionaries are constructed in order to minimize the reconstruction loss of the sparse representation of the objects in each cluster according to a preselected clustering solution. However, these works concern sparse coding instead of the BoF representation. In [39], the histogram space is optimized using a naive spectral criterion that concerns only the top k-nearest neighbors in the spectral space. To the best of our knowledge, the proposed method is the first one that combines dictionary learning for the BoF representation with a manifold-based objective function oriented toward clustering that attempts to fully reconstruct the spectral space using the histogram space.

One shortcoming of the spectral clustering is its inability to assign a new point, i.e., a point that was not available during the optimization, to a cluster without recalculating the spectral space. This problem can be addressed using incremental or online spectral clustering techniques that are able to update the spectral space [26], [40], [41], and avoid the costly recalculation of the eigenvectors. The proposed MO-BoF method can also partially provide the benefits of these methods since the optimized histogram space, which improves the clustering solutions over the original histogram space and allows for the direct assignment of new points to the existing clusters, can be used instead of the spectral space (out-of-sample extension).

3 Proposed Method

In this Section the manifold-based optimization of the BoF model is presented. First, the standard BoF model and the spectral clustering method are introduced. Then, the optimization of the BoF model towards clustering is described and the proposed dictionary learning algorithm is derived. Finally, a toy example using synthetic data is presented to provide a better understanding of the proposed method.

3.1 BoF Model

Suppose that $\mathcal{X} = \{x_i\}_{i=1}^{N}$ is a set of N objects to be represented using the BoF model. Each object x_i consists of N_i feature vectors: $\mathbf{x}_{ij} \in \mathbb{R}^D$ $(j = 1...N_i)$, where D is the dimensionality of the extracted features. For example, HoG feature vectors can be extracted from images, while HoF feature vectors can be extracted from videos. A fixed-length histogram is calculated for each object by quantizing its feature vectors into a predefined number of histogram bins/codewords. When hard quantization is used each feature vector is assigned to its nearest codeword, while in

soft quantization every feature contributes, by a different amount, to each histogram bin/codeword.

Let $\mathcal{T} = \{\mathbf{x}_{ij} | i = 1...N, j = 1...N_i\}$ be the set of all feature vectors of the objects in $\mathcal{X}$. To learn a codebook the vectors of $\mathcal{T}$ are clustered into N_K clusters and the corresponding centroids (codewords) $\mathbf{v}_k \in \mathbb{R}^D$ $(k = 1...N_K)$ are used to form the codebook $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ ...\ \mathbf{v}_{N_K}] \in \mathbb{R}^{D \times N_K}$, where each column of $\mathbf{V}$ is a centroid. These centroids are used to quantize the feature vectors. To reduce the computational time, only a subset of $\mathcal{T}$ is usually clustered. This has little effect on the learned representation. The codebook learning is done only once and the extracted codebook can be used to represent any object.

To encode the i-th object the following process is used. First, the similarity between each feature vector $\mathbf{x}_{ij}$ and each codeword $\mathbf{v}_k$ is calculated as:

$$[\mathbf{d}_{ij}]_k = exp(\frac{-||\mathbf{v}_k - \mathbf{x}_{ij}||_2}{g}) \in \mathbb{R} \quad (1)$$

where the notation $[\mathbf{d}_{ij}]_k$ is used to denote the k-th element of the vector $\mathbf{d}_{ij}$. The hyperparameter g controls the quantization process: for harder assignment $g << 1$ is used, while for softer assignment larger values are used. Then, the normalized membership vector of each feature vector $\mathbf{x}_{ij}$ is computed as:

$$\mathbf{u}_{ij} = \frac{\mathbf{d}_{ij}}{||\mathbf{d}_{ij}||_1} \in \mathbb{R}^{N_K} \quad (2)$$

This vector describes the similarity of the feature vector $\mathbf{x}_{ij}$ to each codeword. Finally, the histogram $\mathbf{t}_i$ is extracted for an object x_i:

$$\mathbf{t}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{u}_{ij} \in \mathbb{R}^{N_K} \quad (3)$$

The histogram $\mathbf{t}_i$ has unit l^1 norm, since $||\mathbf{u}_{ij}||_1 = 1$ for every j. These histograms describe the objects and they can be used for the subsequent clustering, classification or retrieval tasks. The dictionary learning and the histogram extraction are fully unsupervised and no labeled data are required.

3.2 Spectral Clustering

The extracted histograms can be clustered using any clustering algorithm. In this work, the unnornmalized spectral clustering algorithm, as described in [12], is used. The spectral clustering algorithm is shown in Figure 3. First, the similarity graph of the input points is formed (Figure 3, line 2). Several options exist to create the similarity graph [12]. In this paper, the k-nearest neighbor graph is used. In the k-nearest neighbor graph two points, $\mathbf{t}_i$ and $\mathbf{t}_j$ are connected if $\mathbf{t}_i$ is among the k nearest neighbors of $\mathbf{s}_j$ (or vice versa). Also, several methods exist for setting the weights of the similarity graph. Among the most commonly used methods is the Gaussian (or Heat) kernel [12]. Therefore, the adjacency matrix of the similarity graph is defined as:

$$[\mathbf{W}]_{ij} = \begin{cases} \exp(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)}{\sigma_h}) & \text{if } i \text{ is connected to } j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $d(\mathbf{t}_i, \mathbf{t}_j)$ is a distance metric in the histogram space and σ_h is a scaling hyperparameter for the histogram space.

Input: A set of points, i.e., histograms, $\mathcal{S} = \{t_1, ..., t_N\}$ to cluster, the number of clusters K
Output: The formed clusters

1: **procedure** SPECTRALCLUSTERING
2: Construct the similarity graph of the points in $\mathcal{S}$ and calculate its adjacency matrix $\mathbf{W}$ according to equation (4)
3: Calculate the Laplacian $\mathbf{L}$ according to equation (6)
4: Compute the first K eigenvectors $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_K$ of $\mathbf{L}_{sym}$
5: Form the matrix $\mathbf{U} \in \mathbb{R}^{N \times K}$ by stacking the eigenvectors as columns
6: Cluster the rows of T into K clusters using the k-means algorithm
7: Return the clusters

Fig. 3. Spectral Clustering Algorithm

Note that the matrix $\mathbf{W}$ is symmetric since the similarity graph is undirected and $d(x, y) = d(y, x)$. Since the vectors $\mathbf{t}_i$ are histograms it is natural to use a histogram-oriented metric, such as the χ^2 distance [42], instead of the euclidean distance. The χ^2 distance is defined as:

$$d(\mathbf{t}_i, \mathbf{t}_j) = \frac{1}{2} \sum_{l=1}^{N_K} \frac{([\mathbf{t}_i]_l - [\mathbf{t}_j]_l)^2}{[\mathbf{t}_i]_l + [\mathbf{t}_j]_l} \quad (5)$$

Indeed, during our experiments the χ^2 distance yielded better clustering solutions than the euclidean distance. Note that the histogram intersection distance could be also used. However, the χ^2 distance is preferred since it yields similar results and it is also continuous and differentiable.

To solve the spectral clustering optimization problem [12], the Laplacian matrix of the similarity graph is calculated (Figure 3, line 3):

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \quad (6)$$

where $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{W}$ defined as $[\mathbf{D}]_{ii} = \sum_{j=1}^{N} [\mathbf{W}]_{ji}$. Then, the optimization reduces to solving the following eigenproblem:

$$\mathbf{L}\mathbf{e} = \lambda \mathbf{L}\mathbf{e} \quad (7)$$

where $\mathbf{e}$ is an eigenvector and λ its eigenvalue. The first K eigenvectors are calculated (Figure 3, line 4) and the representation of the i-th histogram $\mathbf{t}_i$ in the spectral space is computed as $\mathbf{s}_i = ([\mathbf{e}_1]_i, [\mathbf{e}_2]_i, ..., [\mathbf{e}_K]_i) \in \mathbb{R}^K$, where $\mathbf{e}_i$ is the i-th eigenvector (Figure 3, line 5). Finally, the vectors $\mathbf{s}_i$ are clustered using the k-means algorithm into K clusters (Figure 3, lines 6-7).

The complete clustering procedure, when the BoF representation is used, is described in Figure 4. The codebook is learned by minimizing the reconstruction loss of the feature vectors (Figure 4, line 2), the histogram space is formed (Figure 4, line 3) and then the histograms are clustered (Figure 4, line 4). Note that any other dictionary learning technique can be used (Figure 4, line 2) or any other clustering technique (Figure 4, line 4).

After learning the dictionary, any new object can be directly encoded in the histogram space (Section 3.1) using the non-linear mapping given in (3). Then, it can be readily assigned to its cluster. Note that this is in contrast with the

Input: A set of objects $\mathcal{X}$ to be clustered, the number of clusters K
Output: The formed clusters

1: **procedure** CLUSTERING USING THE BOF REPRESENTATION
2: Learn an unsupervised dictionary $\mathbf{V}$ using the k-means algorithm (the method is described in Section 3.1)
3: Encode the objects using the equation (3) and the dictionary $\mathbf{V}$
4: Cluster the resulting histograms using either the k-means algorithm or the spectral clustering algorithm (described in Figure 3)
5: Return the formed clusters

Fig. 4. Clustering using the BoF representation

spectral space (Section 3.2) that requires resolving the eigen-problem given in (7) to encode a new object. Also, the objects must be clustered again, since their spectral representation is altered. However, the spectral space is more appropriate for clustering, since it unwraps the manifolds of the data. That is not true for the histogram space, when simple data clustering algorithms are used to learn the dictionary.

3.3 Manifold Optimized BoF Model

In this subsection the manifold-based optimization of the BoF model, abbreviated as MO-BoF, is introduced. Then, the application of the MO-BoF method is demonstrated using a toy clustering problem.

3.3.1 Manifold Optimization of the BoF Model

The previous clustering approach used a generic dictionary that was learned by minimizing the reconstruction loss of the quantized feature vectors in the feature space and ignoring the distribution of the data in the histogram space. To this end, the MO-BoF model is introduced in this subsection. The main idea behind the optimization is to create the spectral space, which unwraps the manifold structure of the histogram space, and then to restructure the histogram space in a way that resembles the spectral space. Intuitively, this happens when every pair of points have the same similarity both in the histogram and the spectral space. Therefore, the optimization aims to copy the spectral space into the histogram space by learning a new dictionary optimized toward this task. It should be noted that the new histogram space is expected to perform better, in terms of clustering, only if the spectral clustering achieves better clustering solutions, i.e., if manifolds do exist in the histogram space.

First, the adjacency matrix of the fully connected similarity graph of the histogram space is defined as:

$$[\mathbf{W}_h]_{ij} = \exp(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)}{\sigma_h}) \tag{8}$$

where $d(\mathbf{t}_i, \mathbf{t}_j)$ is the χ^2 distance, as defined in equation (5). Similarly, the fully connected similarity graph of the spectral space is defined as:

$$[\mathbf{W}_s]_{ij} = \exp(-\frac{||\mathbf{s}_i - \mathbf{s}_j||_2}{\sigma_s}) \tag{9}$$

where σ_s is a scaling hyperparameter for the spectral space and $\mathbf{s}_i$ is the spectral representation of the i-th object (as described in Section 3.2). Note that the points that lie in the spectral space are not histograms and, as a result, the χ^2 distance is not appropriate for measuring the distance between them. Therefore, the euclidean distance is used to define the similarity matrix in the spectral space.

In order to make the histogram space as similar as possible to the spectral these two matrices must have approximately the same values, i.e., $\mathbf{W}_h \approx \mathbf{W}_s$. The fully connected similarity graph is used instead of the k-nearest neighbor graph, which is used in the spectral clustering, since the whole histogram space should resemble the spectral space and not only its local structures (that are expected to be more or less the same). Therefore, the following objective function is defined:

$$J = \frac{1}{2}||\mathbf{W}_h - \mathbf{W}_s||_F^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}([\mathbf{W}_h]_{ij} - [\mathbf{W}_s]_{ij})^2 \tag{10}$$

where $||\mathbf{A}||_F$ denotes the Frobenius norm of the matrix $\mathbf{A}$. This objective is minimized when every pair of points has the same similarity both in the histogram and the spectral space, i.e., the points that are similar in the spectral space are also similar in the histogram space and the points that are dissimilar in the spectral space are also dissimilar in the histogram space. Note that the similarity in the spectral space is calculated using the initial dictionary and the matrix $[\mathbf{W}_s]_{ij}$ is not optimized, i.e., it is not a parameter of the problem, since it provides the target similarity. To make the optimization tractable the soft-quantized BoF model is used, as described in Section 3.1, since the objective function is not continuous in the case of the hard-quantized BoF. The gradient descent technique [43], can be used to optimize the objective defined in equation (10):

$$\Delta\mathbf{V} = -\eta\frac{\partial J}{\partial \mathbf{V}} \tag{11}$$

where η is the learning rate. Instead of using the simple gradient descent, the Adam algorithm is utilized [44]. The Adam algorithm computes adaptive learning rates for each of the optimization parameters using estimates of the first and second moments of the gradient. The default parameters for the decay rate of the first and second order estimates, i.e., $\beta_1 = 0.9$ and $\beta_2 = 0.999$, are used and a small value $\epsilon = 10^{-8}$ is utilized to ensure numerical stability. Note that if trainable extractors are utilized for the feature extraction the MO-BoF method can be used to jointly optimize both the dictionary and the feature extractor by backpropagating the gradients to the feature extractor.

Both the gradient descent and the Adam technique require the calculation of the derivate $\frac{\partial J}{\partial \mathbf{V}}$. This derivative (with respect to each codeword $\mathbf{v}_m$, since $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ ...\ \mathbf{v}_{N_K}]$) is calculated as the product of two other partial derivatives using the chain rule:

$$\frac{\partial E}{\partial \mathbf{v}_m} = \sum_{l=1}^{N}\sum_{\kappa=1}^{N_K}\frac{\partial J}{\partial [\mathbf{t}_l]_\kappa}\frac{\partial [\mathbf{t}_l]_\kappa}{\partial \mathbf{v}_m} \tag{12}$$

The first partial derivative, i.e., $\frac{\partial J}{\partial [\mathbf{t}_l]_\kappa}$, provides the direction in which the histograms must be moved in order to minimize the objective.

This derivative is calculated as: $\frac{\partial J}{\partial [\mathbf{t}_l]_\kappa} = -\frac{2}{\sigma_h} \cdot \sum_{i=1}^{N} \left(([\mathbf{W}_h]_{li} - [\mathbf{W}_s]_{li}) [\mathbf{W}_h]_{li} \frac{([\mathbf{t}_l]_\kappa - [\mathbf{t}_i]_\kappa)([\mathbf{t}_l]_\kappa + 3[\mathbf{t}_i]_\kappa)}{([\mathbf{t}_l]_\kappa + [\mathbf{t}_i]_\kappa)^2} \right)$. The second partial derivative ($\frac{\partial [\mathbf{t}_l]_\kappa}{\partial \mathbf{v}_m}$) projects the previous direction into the feature space and provides the direction in which the codewords must be moved. The feature space projection derivative is calculated as: $\frac{\partial [\mathbf{t}_l]_\kappa}{\partial \mathbf{v}_m} = \frac{1}{N_l} \sum_{j=1}^{N_l} \frac{\partial [\mathbf{u}_{lj}]_\kappa}{\partial \mathbf{v}_m}$, where $\frac{\partial [\mathbf{u}_{lj}]_\kappa}{\partial \mathbf{v}_m} = \frac{\partial [\mathbf{u}_{lj}]_\kappa}{\partial [\mathbf{d}_{lj}]_m} \frac{\partial [\mathbf{d}_{lj}]_m}{\partial \mathbf{v}_m} = -\frac{1}{g} [\mathbf{u}_{lj}]_m (\delta_{\kappa m} - [\mathbf{u}_{lj}]_\kappa) \frac{\mathbf{v}_m - \mathbf{x}_{lj}}{||\mathbf{v}_m - \mathbf{x}_{lj}||_2}$ and $\delta_{\kappa m}$ is the Kronecker delta function (If $\kappa = m$, then $\delta_{\kappa m} = 1$. If $\kappa \neq m$, then $\delta_{\kappa m} = 0$).

The learning algorithm for the manifold optimization of the BoF representation is shown in Figure 5. The dictionary learning method initializes the dictionary using the k-means algorithm (Figure 5, line 1). Other methods, such as random initialization, can be also used. However, the k-means provides a better starting point for the optimization. Then, the histogram space is formed and the histograms are extracted (Figure 5, line 3). The extracted histograms are clustered using spectral clustering and the similarity matrix $\mathbf{W}_s$ is calculated (Figure 5, lines 4-5). Typically, the number of dimensions of the spectral space (N_s) is set to the number of clusters. The optimization consists of calculating the derivative of the objective function (Figure 5, line 8), applying the Adam algorithm (Figure 5, line 9), updating the histogram space using the new dictionary (Figure 5, line 10) and re-calculating the similarity matrix in the histogram space (Figure 5, line 11). This process is repeated for a predefined number of iterations. If large dictionaries are used, the method has the potential of overfitting the histogram representation. Therefore, usually a small number of iterations, e.g., 10-20, are used. The spectral space, and the corresponding similarity matrix $\mathbf{W}_s$, can be also updated during the optimization. However, it was experimentally established that this makes the method more prone to overfitting. The hyperparameter selection procedure and the selected hyperparameters are presented in Section 4.1.3.

The complete clustering algorithm when the MO-BoF method is utilized is described in Figure 6. First, the dictionary is learned using the MO-BoF dictionary learning procedure (line 2) and then the objects are encoded and clustered (lines 3-4). Note that any clustering algorithm, e.g., k-means, spectral clustering, etc., can be used for the clustering process.

3.3.2 Toy Example

To provide a better insight on how the proposed method works, a simple example of the optimization using a synthetic dataset is provided. The synthetic data lie on two manifolds. To create the data a random dictionary with three codewords is created (using a gaussian distribution with mean 0 and standard deviation 1) and then feature vectors are randomly generated around each codeword. Finally, 200 (distinct) feature vectors are chosen for each object. To discriminate the two manifolds a different gaussian distribution is used for the feature vectors of an object that should be encoded to a point of the first manifold and a different for the objects of the second manifold (the means of the two distributions are shifted 0.5 apart). The resulting histograms are shown in Figure 7a. Note that the data are projected back to 2 dimensions using the PCA method.

Input: A set of objects $\mathcal{X}$ to be encoded using the MO-BoF, the number of dimensions to keep in the spectral space N_s, the number of iterations N_{iters}, the learning rate η, the scaling hyperparameters σ_h, σ_s, the number of nearest neighbors k for calculating the similarity matrix $\mathbf{W}$
Output: The optimized dictionary V

1: **procedure** MO-BOF DICTIONARY LEARNING
2: Initialize the dictionary $\mathbf{V}$ by running the k-means algorithm on a subsample of the feature vectors (as described in Section 3.1)
3: Encode the objects using the equation (3) and the dictionary $\mathbf{V}$
4: Calculate the nearest neighbors similarity matrix $\mathbf{W}$
5: Using the matrix $\mathbf{W}$ and the technique described in Figure 3 form the spectral space keeping the first N_s eigenvectors (dimensions of the spectral space)
6: Calculate the similarity matrices $\mathbf{W}_h$ and $\mathbf{W}_s$
7: **for** $i = 1$ to N_{iters} **do**
8: Calculate the derivative defined in equation (12)
9: Apply the ADAM algorithm with learning rate η to update $\mathbf{V}$
10: Re-encode the objects using the equation (3) and the updated dictionary $\mathbf{V}$
11: Update the similarity matrix $\mathbf{W}_h$
12: Return the learned dictionary $\mathbf{V}$

Fig. 5. MO-BoF Learning Algorithm

Input: A set of objects $\mathcal{X}$ to be clustered, the number of clusters K, the hyperparameters for the MO-BoF learning algorithm
Output: The formed clusters

1: **procedure** CLUSTERING USING THE MO-BOF REPRESENTATION
2: Use the MO-BoF learning algorithm (Figure 5) to learn a manifold-optimized dictionary $\mathbf{V}$
3: Encode the objects using the equation (3) and the manifold-optimized dictionary $\mathbf{V}$
4: Cluster the resulting histograms using either the k-means algorithm or the spectral clustering algorithm (described in Figure 3)
5: Return the formed clusters

Fig. 6. Clustering using the MO-BoF representation

The spectral representation of the generated histograms (5 neighbors and $\sigma_h = 10$ are used to create the similarity matrix) is shown in Figure 7b. The spectral representation collapses the two manifolds into just two points in the spectral space. Figure 1 illustrates the histogram space during the optimization using the MO-BoF method (the following hyperparameters are used: $\eta = 0.1$, $Ns = 2$, $\sigma_s = 0.1$, $\sigma_h = 10$ and $k = 5$). Again, the data are projected to 2-dimensions using the PCA method. The two manifolds gradually move away from each other and they are partially unwrapped. Therefore, the resulting histogram space is better suited for clustering than the original (especially if centroid-based clustering methods, such as the k-means, are used). It should be noted that the optimization problem is not convex and, as a result, the optimization might converge to a local minimum instead of the global minimum, leading to sub-optimal solutions. Furthermore,

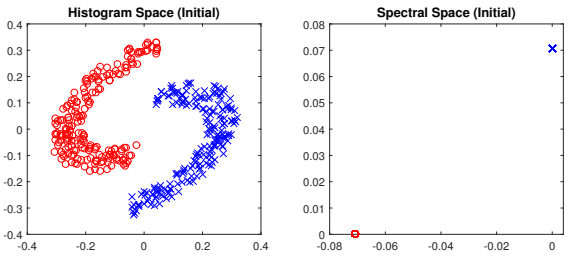


Fig. 7. Histogram representation of the synthetic data (a) (projected to 2-d using PCA) and spectral representation of the synthetic data (b).

other constraints, such as the number of the used codewords and the distribution of the feature vectors, might limit the ability of the proposed method to improve the histogram space even more. Nonetheless, as it is demonstrated in the next Section, the solutions provided by the MO-BoF always improves the histogram space and the clustering metrics for a wide range of problems and experimental setups.

4 Experiments

In this Section the proposed method is evaluated using four different datasets. First, the used datasets, the evaluation metrics and the selected hyperparameters are briefly described. In the next subsections the performance of the proposed method on each dataset is evaluated and discussed. Finally, the proposed method is combined with a trainable extractor and its performance is evaluated.

4.1 Evaluation Setup

4.1.1 Datasets

The proposed method is evaluated using two multi-class image datasets, the 15-scene dataset [21], and the Corel dataset [45], [46], one multi-class video dataset, the KTH action recognition database [47], and one text dataset, the RT-2k movie review dataset [48].

The 15-scene dataset [21], 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*. There are 4485 images and each category has 200 to 400 images. HoG [13], and LBP features [49], of 8×8 non-overlapping patches are densely extracted from each image. The two feature vectors extracted from each patch are fused together to form the final feature vector. The dimensionality of the fused vector is 89. The Corel dataset [45], [46], contains 10800 images from 80 different concepts. Again, HoG and LBP features of 8×8 patches are densely extracted from each image.

The KTH action recognition dataset [47], contains 2391 video sequences of six types of human actions (*walking*, *jogging*, *running*, *boxing*, *hand waving and hand clapping*). From each video HoG descriptors are extracted [14], and these descriptors are used as feature vectors.

The RT-2k [48], is a sentiment dataset that contains 1000 positive and 1000 negative movie reviews in the form of free text. The mean number of the sentences of each movie review is 34.1 ± 15.5. The feature extraction process proposed in [50] is used, i.e., the RNTN model [51], 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 [52], is utilized to this end.

For the 15-scene dataset 3000 images are randomly sampled (5000 for the Corel dataset) for each experiment and the MO-BoF method is used to optimize the initial dictionary. The rest of the images are used as test queries when the method is evaluated in the information retrieval setup. For the KTH dataset 1500 randomly sampled videos are used for the optimization, while for the the RT-2k dataset 1800 randomly sampled reviews are used for training. Again, the rest of the objects are used as test queries. Each experiment is repeated 5 times and the mean and the standard deviation of the evaluation criteria are reported.

4.1.2 Evaluation Metrics

Clustering is evaluated using *internal* and *external* criteria [53], [54]. The internal criteria measure how "well" the data are clustered without using any ground-truth information. Typically, the internal criteria are maximized when the clusters have high intra-cluster similarity, i.e., the objects within each cluster are similar to each other, and low inter-cluster similarity, i.e., two objects that belong to different clusters are dissimilar. The following internal criteria are evaluated in this work [54]:

- *Calinski-Harabasz Criterion*: The Calinski-Harabasz criterion, abbreviated as *CH*, measures the ratio between the overall between-cluster variance and the overall within-cluster variance. Clusters with well defined shapes that produce good partitions of the data maximize this criterion.
- *Silhouette Index*: The Silhouette index measures how similar is a point to the points of its cluster and how dissimilar is to its neighboring clusters. The values of the Silhouette index range from -1 to 1 and a high value means that the point fits well its cluster, while being dissimilar to the other cluster. The mean Silhouette index for all the points is reported.
- *Davies-Bouldin Criterion*: The Davies-Bouldin criterion, abbreviated as *DB*, measures the "worst-case" ratio of the within-cluster and the between-cluster distances for every cluster. Since the Davies-Bouldin criterion takes into account only the worst performance for each cluster, a good clustering solution will have a low Davies-Bouldin value.

On the other hand, the external criteria measure how well the clusters correspond to a specific application. For example, in the case of scene recognition the external criteria measure how well each cluster corresponds to a scene category (or a set of scene categories). In this work three external clustering criteria are used [53]:

- *Rand Index*: The Rand Index is a measure of similarity between two different clustering solutions (or a clustering solution and the ground truth set). The values of the Rand Index ranges from 0 (the solutions are completely dissimilar) to 1 (the clusters are the same).
- *Normalized Mutual Information*: The Mutual Information (MI) is an information theoretic criterion that

measures how much information is shared between two different clustering solutions. Since the MI is not bounded, a normalized variant is usually used, the Normalized Mutual Information, abbreviated as NMI, which ranges from 0 (no mutual information between the corresponding clusters of the two clustering algorithms) to 1 (the corresponding clusters of the two clustering algorithms are identical).

- *Purity*: The purity measures the percentage of correctly assigned objects (according to their label), when each cluster is assigned to its most frequent label. The purity values range from 0% to 100%. The mean purity is reported.

Since the proposed method is fully unsupervised, it can be also used to optimize the BoF representation for information retrieval. Therefore, the proposed method is also evaluated using an information retrieval setup. The database consists of the objects used for the optimization (3000 images for the 15-scene dataset, 5000 for the Corel dataset, 1500 for the KTH dataset and 1800 for the RT-2k dataset) and the rest of them are used to query the database and measure the retrieval precision. Two evaluation metrics are used to plot the precision-recall curves:

- *precision*, which 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 object q, and
- *recall*, which is 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.

In this paper, the interpolated precision, $Pr_{interp}(q,k) = \max_{k',\, k' \geq k} Pr(q,k')$, is used instead of the raw precision since it reduces the precision-recall curve fluctuation [1]. Also, the mean average precision (mAP) is calculated as the mean of the average precision for all queries, which in turn is defined as the mean precision at eleven equally spaced recall points (0, 0.1, ..., 0.9, 1) for a given query.

4.1.3 Hyperparameter Selection

The behavior of the proposed method is stable with regard to most of its hyperparameters and it is expected that the selected hyperparameters will work for a wide range of problems. For all the conducted experiments $\eta = 0.01$ and 10 iterations are used. Using more iterations can further improve the internal evaluation metrics. However, it also increases the risk of overfitting the data and harming the generalization ability of the MO-BoF. The dimensionality of the spectral space (N_s) is set to the expected number of clusters/classes of the data. For the conducted experiments, the N_s is set to the number of the classes that exist in each dataset. However, experimentally it was established that the performance of the method remain stable as long as the dimensionality of the spectral space is greater than the number of the classes. Also, any method that estimates the number of the clusters in the data can be used, e.g., selecting the number of clusters that maximizes the Silhouette criterion. The scaling hyperparameter of the spectral space is fixed to $\sigma_s = 0.1$. The value of σ_h seems to depend on the used dataset and it is set to 10 for the 15-scene dataset and the Corel dataset, 0.5 for the KTH dataset and 0.1 for the RT-2k dataset. The performance of the proposed method is slightly reduced when other values are used for the σ_h. The number of the neighbors (k) used to calculate the similarity matrix for the spectral clustering also seems to depend on the used dataset ($k = 10$ is used for the 15-scene, the Corel and the RT-2k dataset and $k = 100$ for the KTH dataset). The quantization hyperparameter of the BoF model is set to a small value that provided numerically stable results, i.e., $g = 0.05$ for the 15-scene dataset and the Corel dataset and $g = 0.1$ for the KTH dataset and the RT-2k dataset.

To reduce the time needed for the optimization procedure the feature vectors of the objects are subsampled (200 feature vectors are sampled from each object). This technique allows to speed up the optimization process, without significantly reducing the quality of the learned codebook. Finally, since the results provided by the k-means algorithm depend on the used initialization,, the k-means algorithm is initialized 5 different times and the best solution, in terms of clustering loss used by the k-means, is selected. This procedure is followed whenever the k-means algorithm is used in this work.

4.2 Experimental Evaluation

In the following subsection the evaluation of the proposed method using extensive experiments on one image dataset, the 15-scene dataset, is provided. Next, the proposed method is also evaluated using three other datasets from three different domains (image, video and text). The statistical significance of the obtained results is validated in the supplementary material.

4.2.1 15-scene dataset

First, the proposed method is evaluated using the 15-scene dataset. The results using different clustering algorithms and 15 clusters are presented in Table 1. The BoF + kmeans method refers to learning a codebook using the k-means algorithm (as described in Section 3.1) and then clustering the histograms using the k-means algorithm, while the MO-BoF + kmeans refers to using the MO-BoF dictionary learning method and then utilizing the k-means algorithm to cluster the resulting histograms as before. The learned representations are also compared using the Spectral Clustering (SC) algorithm, as described in Figure 3. The spectral clustering solutions are evaluated using the representation of the objects both in the histogram space, annotated by "(h)", and in the spectral representation, annotated by "(s)". Note that the space which is used for the evaluation alters only the internal clustering criteria, i.e., the CH, the Silhouette and the DB. The external evaluation depends only on the clustering solutions, and as a result it is invariant to the space used for the evaluation. The same hyperparameters (k and σ_h) are used for both the spectral clustering and the MO-BoF method. Finally, note that the external criteria are not directly comparable when they are calculated in spaces of different dimensionality.

Several conclusions can be drawn from Table 1. First, the spectral clustering improves the external criteria over the histogram space clustering, which confirms the existence of manifolds in the histogram space. Therefore, it is expected

that the MO-BoF method will improve the learned representation, since it uses the spectral space to optimize the dictionary. Indeed, the proposed method improves every evaluated internal and external criteria in the histogram space using the k-means algorithm. Also, the improved histogram space allows the spectral clustering to extract better solutions. Note that the internal criteria for the spectral clustering are improved with the use of the MO-BoF method both in the histogram space, which is expected since the histogram space is directly optimized, as well as in the spectral space. Therefore, the optimization of the histogram space can indirectly lead to a better spectral space, in terms of the clustering criteria, as well. Note that due to the non-convexity of the optimization problem and the constraints imposed by the feature vectors and the codewords (as discussed in Section 3.3.2), it is not possible to exactly "re-create" the spectral space in the histogram space and match the performance of the spectral clustering. Nonetheless, the optimization improves the learned representation in any case.

Then, the learned representations are also evaluated for different numbers of clusters using one internal and one external criterion. The results are illustrated in Figures 8a and 8b respectively. The MO-BoF improves the CH criterion, especially in the histogram space. The improvement in the spectral space is marginal, which is expected since the optimization does not directly alter the spectral space. Also, regardless the used clustering algorithm, the proposed method leads to a significant improvement of the evaluated internal criterion (Rand index). These figures demonstrate that the manifold optimization of the histogram space improves the clustering criteria over the baseline histogram space (BoF) for different number of clusters.

The previous experiments were conducted using a 128-word dictionary. To demonstrate the ability of the proposed method to optimize dictionaries with different number of codewords, the experiments are repeated using 16, 32 and 64 codewords. For these experiments 15 clusters are used. The results are illustrated in Figures 8c and 8d. Again, the MO-BoF improves both the internal and the external criteria. The larger dictionaries allow to better optimize the histogram space, both by providing more accurate spectral solutions and by increasing the number of the available parameters for the optimization. This is especially evident in Figure 8d, where using more than 32 codewords increases only slightly the Rand index when the BoF + k-means algorithm is used, but it greatly improves the k-means solutions when the MO-BoF representation is used instead. The evaluation using different number of codewords and clusters are omitted for the next three datasets, since the results are similar to those presented in this section.

Finally, the MO-BoF representation is also evaluated using the information retrieval setup described in Section 4.1.2. The mAP of the BoF and the MO-BoF methods using two different distance metrics, the euclidean distance (e) and the χ^2 distance (c), are shown in Table 2. The unsupervised optimization of the codebook does not only increase the clustering metrics, but also increases the retrieval precision by more than 2.5%. This is also confirmed by the precision-recall curves (available in Section 3 of the supplementary material).

TABLE 2
Retrieval evaluation using the mAP

Method	Dataset	mAP (e)	mAP (c)
BoF	15-scene	26.59 ± 0.33	34.10 ± 0.26
MO-BoF	15-scene	$\mathbf{29.29 \pm 0.32}$	$\mathbf{36.57 \pm 0.22}$
BoF	Corel	14.73 ± 0.10	18.04 ± 0.12
MO-BoF	Corel	$\mathbf{15.57 \pm 0.14}$	$\mathbf{19.55 \pm 0.17}$
BoF	KTH	38.03 ± 0.35	39.26 ± 0.35
MO-BoF	KTH	$\mathbf{38.90 \pm 0.65}$	$\mathbf{41.15 \pm 0.29}$
BoF	RT-2k	68.48 ± 0.47	70.08 ± 0.60
MO-BoF	RT-2k	$\mathbf{69.41 \pm 0.65}$	$\mathbf{70.37 \pm 0.60}$

4.2.2 Corel dataset

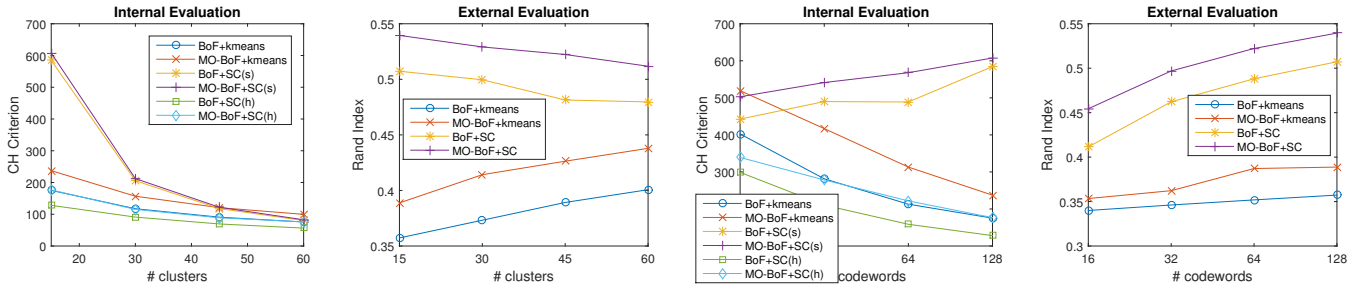
The experimental evaluation using the Corel dataset and 128 codewords is provided in Table 3. Again, the proposed method improves the evaluated clustering metrics in the histogram space, as well as in the spectral space (using spectral clustering). The slight increase in the DB criterion (spectral clustering evaluated in the histogram space) can be attributed to the nature of this criterion: the DB measures the worst case ratio of the within-cluster and the between-cluster distances for every cluster. Therefore, even if the ratio for most of the clusters is improved (as hinted by the other internal criteria), a single sub-optimal cluster can increase the value of this criterion. The mAP for the two evaluated methods is shown in Table 2. Again, the MO-BoF improves the retrieval precision.

4.2.3 KTH dataset

The proposed method is evaluated in the KTH dataset using a dictionary with 128 codewords. The results are shown in Table 4. There is a large improvement in the evaluated criteria (more than 45% improvement in the NMI) when the k-means algorithm is used. The clustering metrics also improve when spectral clustering is used (both in the histogram space and the spectral space). Note that the purity slightly decreases when the MO-BoF is combined with the spectral clustering. However, this concerns only the spectral space (which is not directly optimized by the MO-BoF method) and the decrease is relatively small compared to the attained improvement in the histogram space (-0.18% for the spectral clustering vs. +20.24% for the k-means clustering). The mAP for the information retrieval setup is shown in Table 2. The MO-BoF method increases the mAP by 1.5% when the χ^2 distance is used. The improvements in the precision-recall curves (available in Section 3 of the supplementary material) are smaller. This can be attributed to the fact that the optimization criterion is mainly oriented towards clustering instead of information retrieval.

4.2.4 RT-2k dataset

The clustering evaluation for the RT-2k dataset using 2 clusters and 4 codewords is presented in Table 6. Unlike the previous datasets, the performance of the BoF method for this dataset peaks relatively soon (using 4-8 codewords instead of 128). Again, all the internal and the external evaluation criteria improve when the MO-BoF method is used regardless the utilized clustering algorithm and the space used for the evaluation. The mAP evaluation is presented in Table 2, while the precision-recall curves are available in the



(a) Internal Evaluation (clusters) (b) External Evaluation (clusters) (c) Internal Evaluation (codewords) (d) External Evaluation (codewords)

Fig. 8. Comparison between BoF and MO-BoF (15-scene dataset) using different clustering algorithms, number of clusters and number of codewords

TABLE 1
15-scene dataset - Comparison between BoF and MO-BoF (128 codewords) using different clustering algorithms and 15 clusters

Method	CH	Silhouette	DB	Rand Index	NMI	Purity
BoF+kmeans	175.15 ± 3.69	0.139 ± 0.007	2.095 ± 0.076	0.357 ± 0.004	0.171 ± 0.004	37.07 ± 1.24
MO-BoF+kmeans	$\mathbf{237.23 \pm 7.21}$	$\mathbf{0.186 \pm 0.015}$	$\mathbf{1.927 \pm 0.080}$	$\mathbf{0.389 \pm 0.014}$	$\mathbf{0.191 \pm 0.013}$	$\mathbf{38.80 \pm 2.20}$
BoF+SC(s)	584.41 ± 59.61	0.513 ± 0.027	0.911 ± 0.077	0.507 ± 0.016	0.282 ± 0.026	51.89 ± 2.38
MO-BoF+SC(s)	$\mathbf{607.56 \pm 57.90}$	$\mathbf{0.527 \pm 0.030}$	$\mathbf{0.900 \pm 0.060}$	$\mathbf{0.540 \pm 0.012}$	$\mathbf{0.300 \pm 0.013}$	$\mathbf{52.24 \pm 2.38}$
BoF+SC(h)	128.10 ± 8.08	0.052 ± 0.020	2.405 ± 0.117	0.507 ± 0.016	0.282 ± 0.026	51.89 ± 2.38
MO-BoF+SC(h)	$\mathbf{176.65 \pm 4.30}$	$\mathbf{0.086 \pm 0.010}$	$\mathbf{2.110 \pm 0.085}$	$\mathbf{0.540 \pm 0.012}$	$\mathbf{0.300 \pm 0.013}$	$\mathbf{52.24 \pm 2.38}$

TABLE 3
Corel dataset - Comparison between BoF and MO-BoF (128 codewords) using different clustering algorithms and 80 clusters

Method	CH	Silhouette	DB	Rand Index	NMI	Purity
BoF+kmeans	108.85 ± 1.64	0.108 ± 0.004	2.111 ± 0.027	0.391 ± 0.003	0.085 ± 0.003	25.56 ± 0.48
MO-BoF+kmeans	$\mathbf{118.50 \pm 4.95}$	$\mathbf{0.135 \pm 0.007}$	$\mathbf{2.023 \pm 0.038}$	$\mathbf{0.405 \pm 0.004}$	$\mathbf{0.103 \pm 0.003}$	$\mathbf{27.05 \pm 0.64}$
BoF+SC(s)	113.48 ± 5.03	0.310 ± 0.009	1.352 ± 0.074	0.436 ± 0.003	0.104 ± 0.006	30.79 ± 0.47
MO-BoF+SC(s)	$\mathbf{117.01 \pm 2.76}$	$\mathbf{0.314 \pm 0.015}$	$\mathbf{1.309 \pm 0.067}$	$\mathbf{0.453 \pm 0.003}$	$\mathbf{0.119 \pm 0.010}$	$\mathbf{32.55 \pm 0.74}$
BoF+SC(h)	81.45 ± 1.07	0.018 ± 0.017	$\mathbf{2.303 \pm 0.038}$	0.436 ± 0.003	0.104 ± 0.006	30.79 ± 0.47
MO-BoF+SC(h)	$\mathbf{87.63 \pm 3.87}$	$\mathbf{0.036 \pm 0.010}$	2.321 ± 0.041	$\mathbf{0.453 \pm 0.003}$	$\mathbf{0.119 \pm 0.010}$	$\mathbf{32.55 \pm 0.74}$

TABLE 4
KTH dataset - Comparison between BoF and MO-BoF (128 codewords) using different clustering algorithms and 6 clusters

Method	CH	Silhouette	DB	Rand Index	NMI	Purity
BoF+kmeans	283.98 ± 14.19	0.354 ± 0.045	1.525 ± 0.112	0.213 ± 0.038	0.111 ± 0.034	34.01 ± 3.24
MO-BoF+kmeans	$\mathbf{483.19 \pm 40.71}$	$\mathbf{0.421 \pm 0.044}$	$\mathbf{1.370 \pm 0.057}$	$\mathbf{0.291 \pm 0.040}$	$\mathbf{0.180 \pm 0.028}$	$\mathbf{41.67 \pm 3.26}$
BoF+SC(s)	2162.34 ± 225.28	0.800 ± 0.016	0.463 ± 0.028	0.320 ± 0.007	0.185 ± 0.008	$\mathbf{44.99 \pm 0.70}$
MO-BoF+SC(s)	$\mathbf{2708.24 \pm 215.56}$	$\mathbf{0.832 \pm 0.010}$	$\mathbf{0.411 \pm 0.018}$	$\mathbf{0.326 \pm 0.008}$	$\mathbf{0.191 \pm 0.005}$	44.93 ± 0.32
BoF+SC(h)	266.19 ± 10.63	0.308 ± 0.011	1.602 ± 0.038	0.320 ± 0.007	0.185 ± 0.008	$\mathbf{44.99 \pm 0.70}$
MO-BoF+SC(h)	$\mathbf{444.92 \pm 20.84}$	$\mathbf{0.440 \pm 0.011}$	$\mathbf{1.317 \pm 0.028}$	$\mathbf{0.326 \pm 0.008}$	$\mathbf{0.191 \pm 0.005}$	44.93 ± 0.32

Section 3 of the supplementary material. The retrieval metrics follow the same trend with the clustering metrics and they are also improved when the MO-BoF representation is used instead of the BoF representation.

4.2.5 *MO-BoF evaluation using trainable feature extractors*

When trainable feature extractors, such as convolutional neural networks [15], are used the gradients can backpropagate to the feature extractor further finetuning the extracted features toward clustering. To this end, a neural network is combined with the proposed MO-BoF approach and tested under two different scenarios: a) only training the codebook (abbreviated as MO-BoF (cod.)) and b) jointly optimizing both the codebook and the convolutional feature extractor (abbreviated as MO-BoF(all)). The MNIST dataset [55], that contains images of handwritten digits (0 to 9), is utilized for evaluating the methods. The evaluation results are summarized in Table 5. The baseline BoF technique uses the feature vectors extracted from the CNN without further training the CNN. The MO-BoF method performs better than the baseline BoF method. Jointly learning the codebook and the convolutional feature extractor (MO-BoF (all)) leads to even better clustering results. The exact experimental setup and the evaluation results are analytically reported in the supplementary material.

TABLE 5
MNIST dataset - Spectral clustering evaluation in the spectral space using a trainable feature extractor

Method	CH	Silh.	DB	Rand.	NMI
BoF	223.934	0.405	0.996	0.849	0.462
MO-BoF (cod.)	231.997	0.413	1.074	0.858	0.484
MO-BoF (all)	**252.296**	**0.441**	**0.988**	**0.862**	**0.495**

TABLE 6
RT-2k dataset - Comparison between BoF and MO-BoF (4 codewords) using different clustering algorithms and 2 clusters

Method	CH	Silhouette	DB	Rand Index	NMI	Purity
BoF+kmeans	921.73 ± 19.19	0.458 ± 0.005	1.245 ± 0.014	0.223 ± 0.007	0.284 ± 0.009	76.66 ± 0.43
MO-BoF+kmeans	$\mathbf{1238.07 \pm 38.34}$	$\mathbf{0.530 \pm 0.005}$	$\mathbf{1.072 \pm 0.013}$	$\mathbf{0.272 \pm 0.004}$	$\mathbf{0.350 \pm 0.004}$	$\mathbf{79.58 \pm 0.19}$
BoF+SC(s)	5161.50 ± 80.28	0.785 ± 0.003	0.506 ± 0.003	0.305 ± 0.007	0.390 ± 0.009	81.23 ± 0.37
MO-BoF+SC(s)	$\mathbf{5444.17 \pm 179.72}$	$\mathbf{0.792 \pm 0.006}$	$\mathbf{0.483 \pm 0.009}$	$\mathbf{0.315 \pm 0.006}$	$\mathbf{0.402 \pm 0.008}$	$\mathbf{81.72 \pm 0.31}$
BoF+SC(h)	830.33 ± 14.94	0.431 ± 0.004	1.326 ± 0.012	0.305 ± 0.007	0.390 ± 0.009	81.23 ± 0.37
MO-BoF+SC(h)	$\mathbf{1022.58 \pm 43.43}$	$\mathbf{0.472 \pm 0.011}$	$\mathbf{1.169 \pm 0.022}$	$\mathbf{0.315 \pm 0.006}$	$\mathbf{0.402 \pm 0.008}$	$\mathbf{81.72 \pm 0.31}$

5 Conclusions and Future Work

In this paper a manifold-based dictionary learning method oriented toward information clustering was proposed. First, the spectral representation of the data is formed. This representation unwraps the manifolds that exist in the histogram space. Then, a new codebook is learned in order to make the histogram space as similar as possible to the spectral space. The proposed method was evaluated using four different datasets from three different domains (image, video and text). The optimization greatly improves the learned histogram space both in terms of the internal clustering criteria and in terms of the external clustering criteria. Although the proposed method does not directly optimize the spectral space, it increases the evaluated criteria in the spectral space. Furthermore, the optimized histogram space can be used to directly assign a new object to a cluster, instead of using the spectral space, avoiding the need to re-apply the spectral clustering or use incremental spectral clustering techniques. Finally, the learned representation was also evaluated using an information retrieval setup and it was confirmed that it improves the retrieval precision over the baseline BoF representation.

There are several interesting future research directions. First, instead of using handcrafted features, such as HoG, HoF, etc., trainable feature extractors, such as deep convolutional neural networks [15], can be used. The gradients can backpropagate to the feature extractor further increasing the accuracy of the model by learning clustering-oriented features. The potential of this approach is demonstrated in the supplementary material using the MNIST dataset and deep convolutional neural networks as feature extractors. Also, instead of using spectral clustering, other manifold-based techniques, such as ISOMAP and its extensions [56], can be used to learn the optimized histogram space. Finally, the proposed technique can be extended to work in a semi-supervised setting by exploiting both labeled and unlabeled samples.

References

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

[2] J. Cao, Z. Wu, J. Wu, and H. Xiong, "Sail: Summation-based incremental learning for information-theoretic text clustering," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 570–584, 2013.

[3] B. Gao, T.-Y. Liu, T. Qin, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Web image clustering by consistent utilization of visual features and surrounding texts," in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 112–121.

[4] N. Guil, J. M. González-Linares, J. R. Cózar, and E. L. Zapata, "A clustering technique for video copy detection," in *Pattern Recognition and Image Analysis*, 2007, pp. 451–458.

[5] T. W. Liao, "Clustering of time series data survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[6] X. Cao, X. Wei, Y. Han, and D. Lin, "Robust face clustering via tensor decomposition," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2546–2557, 2015.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[8] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[9] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[10] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[11] R. J. Hathaway and J. C. Bezdek, "Fuzzy c-means clustering of incomplete data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 31, no. 5, pp. 735–744, 2001.

[12] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[14] 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.

[15] H. Qiao, X. Xi, Y. Li, W. Wu, and F. Li, "Biologically inspired visual model with preliminary cognition and active attention adjustment," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2612–2624, 2015.

[16] S. Krig, "Interest point detector and feature descriptor survey," in *Computer Vision Metrics*. Springer, 2014, pp. 217–282.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] H. Lobel, R. Vidal, D. Mery, and A. Soto, "Joint dictionary and classifier learning for categorization of images using a max-margin framework," in *Image and Video Technology*, 2014, pp. 87–98.

[24] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proceedings of the 10th IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1800–1807.

[25] T. Kong, Y. Tian, and H. Shen, "A fast incremental spectral clustering for large data sets," in *12th international conference on Parallel and distributed computing, applications and technologies*, 2011, pp. 1–5.

[26] C. Dhanjal, R. Gaudel, and S. Clémençon, "Efficient eigen-updating for spectral graph clustering," *Neurocomputing*, vol. 131, pp. 440–452, 2014.
[27] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1083–1094, 2015.
[28] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2015.
[29] 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.
[30] Y. Kuang, K. Åström, L. Kopp, M. Oskarsson, and M. Byröd, "Optimizing visual vocabularies using soft assignment entropies," in *Proceedings of the 10th Asian Conference on Computer Vision*, 2011, pp. 255–268.
[31] Y. Kuang, M. Byröd, and K. Åström, "Supervised feature quantization with entropy optimization," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1386–1393.
[32] M. Jiu, C. Wolf, C. Garcia, and A. Baskurt, "Supervised learning and codebook optimization for bag-of-words models," *Cognitive Computation*, vol. 4, no. 4, pp. 409–419, 2012.
[33] W. Zhang, A. Surve, X. Fern, and T. Dietterich, "Learning non-redundant codebooks for classifying complex objects," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 1241–1248.
[34] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *Proceedings of the 9th European Conference on Computer Vision*, 2006, pp. 464–475.
[35] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, 2008.
[36] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *Proceedings of the 10th European Conference on Computer Vision*, 2008, pp. 179–192.
[37] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3501–3508.
[38] P. Sprechmann and G. Sapiro, "Dictionary learning and sparse coding for unsupervised clustering," in *IEEE International Conference on Acoustics Speech and Signal Processing*, 2010, pp. 2042–2045.
[39] N. Passalis and A. Tefas, "Spectral clustering using optimized bag-of-features," in *Proceedings of the 9th Hellenic Conference on Artificial Intelligence*, 2016, pp. 19:1–19:9.
[40] R. Langone, O. M. Agudelo, B. De Moor, and J. A. Suykens, "Incremental kernel spectral clustering for online learning of non-stationary data," *Neurocomputing*, vol. 139, pp. 246–260, 2014.
[41] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. S. Huang, "Incremental spectral clustering by efficiently updating the eigen-system," *Pattern Recognition*, vol. 43, no. 1, pp. 113–127, 2010.
[42] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.
[43] S. S. Haykin, *Neural networks and learning machines*. Pearson Education Upper Saddle River, 2009, vol. 3.
[44] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.
[45] 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.
[46] W. Bian and D. Tao, "The COREL database for content based image retrieval," https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval, 2009.
[47] 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.
[48] 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, pp. 271–278.
[49] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
[50] 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.
[51] 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.
[52] 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.
[53] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for k-means clustering," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 877–886.
[54] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Understanding and enhancement of internal clustering validation measures," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 982–994, 2013.
[55] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.
[56] Z. Zhang, T. W. Chow, and M. Zhao, "M-isomap: Orthogonal constrained marginal isomap for nonlinear dimensionality reduction," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 180–191, 2013.

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 Ph.D. 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 60 journal papers, 145 papers in international conferences and contributed 8 chapters to edited books in his area of expertise. Over 2800 citations have been recorded to his publications and his H-index is 27 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.