Multi-Attributed Dictionary Learning for Sparse Coding

Chen-Kuo Chiang, Te-Feng Su, Chih Yen and Shang-Hong Lai
National Tsing Hua University, Hsinchu, 300, Taiwan
{ckchiang, tfsu, lai@cs.nthu.edu.tw}

Abstract

We present a multi-attributed dictionary learning algorithm for sparse coding. Considering training samples with multiple attributes, a new distance matrix is proposed by jointly incorporating data and attribute similarities. Then, an objective function is presented to learn category-dependent dictionaries that are compact (closeness of dictionary atoms based on data distance and attribute similarity), reconstructive (low reconstruction error with correct dictionary) and label-consistent (encouraging the labels of dictionary atoms to be similar). We have demonstrated our algorithm on action classification and face recognition tasks on several publicly available datasets. Experimental results with improved performance over previous dictionary learning methods are shown to validate the effectiveness of the proposed algorithm.

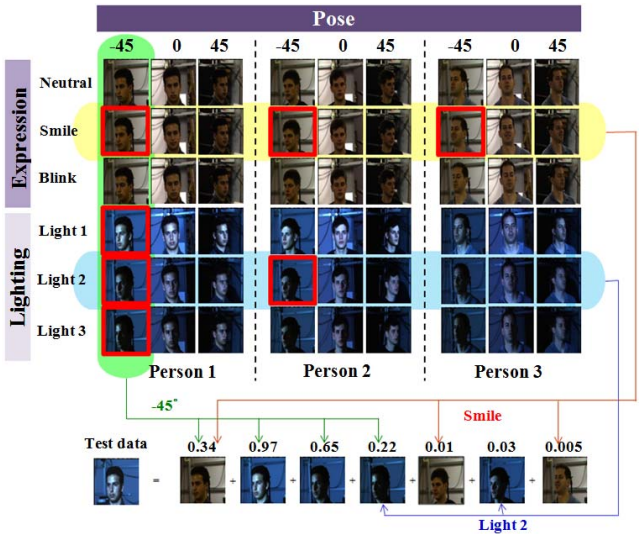


Figure 1. Example of utilizing multiple attributes in dictionary learning for sparse representation with attributes of facial expressions, pose variations and lighting conditions.

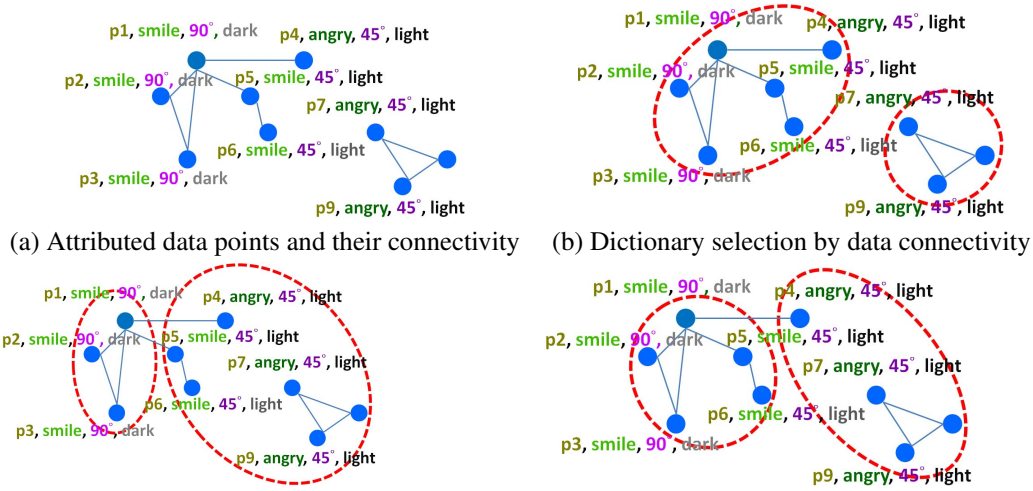
1. Introduction

Sparse coding technique attracts more and more attention because of its success in a variety of image processing and computer vision applications. It recovers a sparse linear representation of a query datum with respect to a set of non-parametric basis set, known as *dictionary*. Originally, predefined dictionaries based on various types of wavelets have been used. Lately, learning the dictionary instead of using predefined bases has been shown to improve signal reconstruction significantly.

Dictionary learning of sparse representation is aimed to find the optimal dictionary that leads to the lowest reconstruction error with a set of sparse coefficients. Wright et al. [18] exploited the entire training set as the dictionary and proposed the sparse representation classification (SRC) for robust face recognition. Some algorithms learn category-dependent dictionaries [12, 10, 14] since solving the sparse coding problem with multiple sub-dictionaries has the advantages of lower computational complexity and straightforward implementation with parallel computing compared to that with a single universal dictionary. Mairal et al. [12] assumed a correct dictionary associated with one class should provide better reconstruction than those using incorrect dictionaries. An additional term was introduced into the cost function to improve the discrimination power.

The K-SVD algorithm [1] learns an over-complete dictionary from a set of signals. It has achieved good performance for the image denoising problem. Since it focuses on the representation power of the dictionary without considering the discrimination capability, the Discriminative K-SVD algorithm (D-KSVD) [20] achieved the representational and discriminative dictionary learning in a unified process. Other algorithms adding the discriminative term into the objective function can be found in [19]. Submodular dictionary learning [9] models the selection of the dictionary columns and the sparse representation of signals as a joint combinatorial optimization problem. Later, a compact and discriminative submodular dictionary learning was proposed by a greedy-based approach [6].

In addition to the criteria of reconstruction and discrimination, data labels have also been considered in dictionary



(a) Attributed data points and their connectivity (b) Dictionary selection by data connectivity

(c) Dictionary selection by data attributes (d) Dictionary selection by data connectivity and attributes

Figure 2. Dictionary selection by considering data connectivity and attribute similarity.

learning. A label consistent K-SVD (LC-KSVD) algorithm [5] associated the class labels with each dictionary atom to enforce discrimination in sparse codes. A similar approach can be also found in [6]. A recent work [16] learned a context aware dictionary by a set of labeled training images to predict the presence of objects in images. In the existing methods, only single attribute or class label is considered in the dictionary learning problem. However, in real-world applications, data samples usually contain multiple attributes. For the face recognition problem, as depicted in Figure 1, a test face image from a person with the smile expression and -45° head pose under a specific lighting condition may be better reconstructed using dictionary atoms from close pose (-45° in the green region), similar lighting (Light 2 in the blue region) and same expression (smile in the yellow region). Obviously, linear combination of data from different geometrical configurations, such as poses, expressions or others, could never give a new correct data point but only blurry images. It is better not to mix data points with different attributes. We argue that the problem of dictionary learning can be beneficial by considering multiple attributes of data points simultaneously with the reconstruction to enhance the discrimination power of dictionary, which has not been well addressed in the previous literature or handled in an effective way.

In this paper, we present an algorithm for learning discriminative category-dependent dictionaries from a set of training images which are labeled with multiple attributes. A novel objective function is proposed for the above purpose. It consists of three terms. The compact term favors close dictionary atoms by utilizing both data and attribute similarity into one unified distance measure. The reconstruction term introduces the representative ability by selecting dictionary atoms with minimal reconstruction errors. Last, the label term enforces label-consistent dictionary atoms from multi-attributed training samples. The main contributions of this paper are:

- A unified distance measure is proposed by mapping the data points and their attributes into a graph. The transition probability of the graph is utilized to measure a new distance on how close the sample pair is and how similar the attributes they share simultaneously.
- We present an objective function for dictionary learning that considers the data representation capability, the discrimination power, and label consistency of multiple attributes in a unified framework.
- We demonstrate the effectiveness of our method through the action classification and face recognition experiments on several publicly available datasets, including IXMAS [17], AR Dataset [13], and CMU PIE dataset [15]. Our algorithm achieves state-of-the-art performance.

2. Problem Statement

Given a signal x in R^m, a sparse approximation over a dictionary D in $R^{m\times k}$ is to find a linear combination of a few atoms from D that is close to the signal x, where the k columns selected from D are referred to as dictionary atoms. It optimizes the following cost function:

$$R(x, D) = \min_{\alpha \in R^k} \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda \|\alpha\|_1 \qquad (1)$$

where λ is a parameter that balances between reconstruction error and sparsity. The l_1-constraint induces sparse solutions for the coefficient vector α. In this paper, we aim to

learn K category-dependent dictionaries. Therefore, dictionary D can be represented as $D = [D^{(1)}, ..., D^{(K)}]$, where K is the number of class. A set of training data is defined as $X^{(k)} = [x_1^k, ..., x_{n_k}^k] \in R^{m \times n_k}$, where $X^{(k)}$ is the samples from class k, n_k is the number of total training samples in class k. The whole training set is defined as $X = [X^{(1)}, ..., X^{(K)}]$. In this work, we consider our data has multiple *attributes*. For the face recognition problem, the attributes could be facial expression, face pose or a lighting condition, etc. Denote the attribute set $A = [a_1, ..., a_r]$ of r attributes. In each attribute, it may have several *types*. For example, the attribute for facial expression may be the smile, angry or screaming type. The types in attribute a_i are defined as $a_i = [a_1^i, ..., a_{n_i}^i]$.

Considering data distance and attribute similarity in dictionary learning problem, we can combine these two terms with appropriate weighting. However, it is difficult to tune the weighting coefficients to achieve optimal performance as the number of attribute increases. To learn the dictionary automatically and deterministically, we model the dictionary learning as a clustering problem. First, data samples and their multiple attributes are mapped into a graph. A new distance of measuring the pairwise relationship is proposed by considering both the Euclidean distance and shared attributes between a pair of data points. Then, the dictionaries are learned by partitioning the graph into K clusters via minimizing the objective function which enforces the dictionary to be compact, reconstructive and label-consistent.

3. Distance Measure of Data and Attributes

To realize how dictionary can be selected by graph clustering based on data connectivity (the k-nearest-neighbor relationship) and shared attributes, a simple example is depicted in Figure 2. In Figure 2 (a) data points are connected to their nearest neighbors by edges. Each data point is labeled by their multiple attributes. (p1, smile, 90°, dark) represents that the image is from person_1 with smile expression, 90° face pose and captured under dark lighting condition. In Figure 2 (b), dictionary selection considers only data connectivity. In Figure 2 (c), data points sharing two out of three attributes are clustered. We argue that dictionary selection can be better achieved based on data connectivity and their multiple attributes, which is illustrated in Figure 2 (d). In this paper, we integrate the data distance and attribute similarity into a unified framework based on the construction of an augmented graph.

3.1. Graph Construction

We construct a directed graph by mapping the training set X into the graph $G = (V, E)$, where V is the set of vertices, E is the set of edges. An edge $e_{ij} \in E$ exists between vertex v_i and v_j if v_j is the k-nearest-neighbor of v_i. Except for data vertices, we also add vertices for attributes

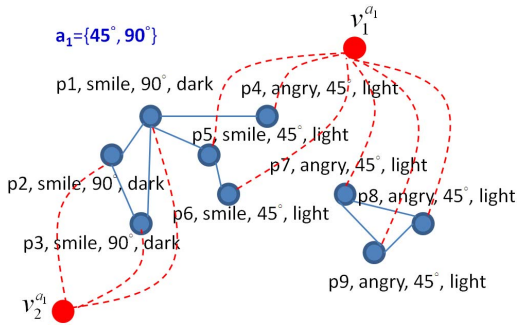


Figure 3. Example of adding attribute vertices into the graph.

into graph G. Assume an attribute set $A = [a_1, ..., a_r]$ containing a total number of r attributes which are associated with vertices in V. Denote the types in attribute a_i by $[a_1^i, ..., a_{n_i}^i]$. Attribute vertices V_a can be defined to associate with type j, j=1,...,n_i, of attribute i as $V_a = \{\{v_j^{a_i}\}_{j=1}^{n_i}\}_{i=1}^{r}$. An edge between a data vertex and an attribute vertex $(v_i, v_j^{a_i}) \in E_a$ is constructed if the data vertex v_i has the attribute a_i with the type a_j^i. There is no edge between two attribute vertices. Then, we can define an augmented graph with attributes $G_a = (V \bigcup V_a, E \bigcup E_a)$. Figure 3 depicts an example that adds a pose attribute a_1 into the graph. $(v_1^{a_1}, v_2^{a_1})$ represents the types (45°, 90°) of the pose. For brevity, the data vertices are called *D_node* and the attribute vertices are called *A_node* for the rest of this paper.

3.2. Unified Distance Measure

In the augmented graph, if a path exists between two vertices, they are connected either by edges in edge set E from k-nearest-neighbors or by E_a which indicates they share the same type of attribute. Therefore, if there are multiple paths between two vertices, they are close to each other. Disconnected vertices or vertices with just a few paths between them imply their distance is far. Here, we use the random walk model [4] to measure the distance between vertices.

The transition probability matrix P for a graph is defined as an N-by-N matrix, where N is the number of vertices. The entry P_{ql} indicates the probability of traveling from vertex v_q to vertex v_l. The transition probability matrix P that we can travel between two vertices in s steps is defined as: $P^{(1)} = P, P^{(s)} = P^{(s-1)} * P = P^s$. The neighborhood random walk distance is defined as:

$$d(v_q, v_l) = 1 - \sum_{s=1}^{S} P^{(s)}(v_q, v_l) \quad (2)$$

where s is step size of random walk. The transition probability measures reaching v_l in $1, ..., S$ steps starting at v_q. The higher probability means the shorter distance between two vertices.

Next, we give the definition of transition probability among *D_node* and *A_node*. The transition probability from a vertex v_q in *D_node* to another vertex v_l in *D_node* is defined by:

$$P^{(1)}(v_q, v_l) = \begin{cases} \frac{1}{|\Omega(v_q)|+r} & \text{if } (v_q, v_l) \in E \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $|\Omega(v_q)|$ represents the number of neighbors of vertex v_q. It means v_q can reach v_l in one step with the probability $\frac{1}{|\Omega(v_q)|+r}$ if v_q and v_l are connected by a edge. This is intuitive since v_q has $\Omega(v_q) + r$ edges to other vertices. Here, the distance is defined based on the number of connected vertices. An alternative is to utilize some weights attached to the edges, such as the likelihood of the connection. Similarly, the transition probability from a vertex in *D_node* to a vertex in *A_node* is given by:

$$P^{(1)}(v_q, v_j^{a_i}) = \begin{cases} \frac{1}{|\Omega(v_q)|+r} & \text{if } (v_q, v_j^{a_i}) \in E_A \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The transition probability from a vertex in *A_node* to a vertex in *D_node* is given by:

$$P^{(1)}(v_j^{a_i}, v_q) = \begin{cases} \frac{1}{|\Omega(v_j^{a_i})|} & \text{if } (v_j^{a_i}, v_q) \in E_A \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Since there is no edge between any two *A_node*, the transition probability is zero between two vertices in *A_node*:

$$P^{(1)}(v_j^{a_i}, v_t^{a_s}) = 0, \forall v_j^{a_i}, v_t^{a_s} \in V_A \quad (6)$$

From the transition probabilities defined by Eq. 3 to Eq. 6, we are able to compute the transition probability matrix of graph G_a. In Figure 3, we use only the pose attribute for example. There are totally eleven vertices. An example of the transition probability matrix P for nine vertices in *D_node* and two vertices in *A_node* is given below:

$$p^{(1)} = \begin{bmatrix} 0 & 1/5 & .. & 0 & 0 & 0 & 1/5 \\ 1/3 & 0 & .. & 0 & 0 & 0 & 1/3 \\ : & : & : & : & : & : & : \\ 0 & 0 & .. & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & .. & 1/3 & 0 & 1/3 & 0 \\ 0 & 0 & .. & 1/6 & 1/6 & 0 & 0 \\ 1/3 & 1/3 & .. & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

where the order of rows and columns corresponds to vertices $(v_1, v_2, ..., v_8, v_9, v_1^{a_1}, v_2^{a_1})$. By defining the transition probability between vertices in the augmented graph, we can note that for two vertices v_q and v_l with the same connectivity in graph G_a, if a vertex v_t shares more attributes with v_q than v_l, the distance $d(v_t, v_q)$ is less than $d(v_t, v_l)$. A simple example is given in Figure 4. Vertices v_2 and v_3 have three edges connecting to other vertices in *D_node*. Vertex v_1 shares the same attributes (90°, angry) with v_3 and shares only one attribute 90° with v_2. Intuitively, there are only two paths from v_1 to v_2 whereas three paths exist between v_1 and v_3. In other words, the transition probability from v_1 to v_3 is higher than that between v_1 and v_2. Therefore, $d(v_1, v_3)$ is less than $d(v_1, v_2)$.

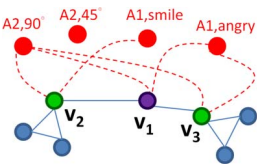


Figure 4. Example of closer vertices when sharing more attributes.

4. Multi-Attributed Dictionary Learning

Base on the unified distance measure of data and attributes, we propose a novel Multi-Attributed Dictionary Learning (MADL) scheme. Instead of learning a dictionary for the entire dataset, we learn K category-dependent sub-dictionary $D^{(1)}, ..., D^{(K)}$. Denote $D = [D^{(1)}, ..., D^{(K)}]$. Dictionary learning aims to be compact (closeness of dictionary atoms based on data distance and attribute similarity), reconstructive (low reconstruction error with correct dictionary), and label-consistent (encouraging labels of dictionary atoms to be similar). In the following, we formulate the multi-attributed dictionary learning problem and describe the novel objective function for the optimization.

4.1. Compact Term

We use the compact term to constrain the dictionary atoms to be selected under closer data distance or with more shared attributes to the centroid. The pair-wise distance matrix computed from Eq. 2 is utilized. Denote $\bar{v}^{(k)}$ the centroid of atoms in dictionary $D^{(k)}$. To minimize the intra-class distance over the data and attributes, the compact term is defined as:

$$C(D) = \sum_{k=1}^{K} \sum_{\forall v_q \in D^{(k)}} d(v_q, \bar{v}^{(k)}) \quad (8)$$

In the dictionary selection process, an atom v_q is assigned to dictionary $D^{(k)}$ if it satisfies:

$$k^* = \arg\min_k d(v_q, \bar{v}^{(k)}) \quad (9)$$

4.2. Reconstruction Term

It is critical to learn a dictionary which is representative, i.e. with low reconstruction error, since the discrimination

power relies on low reconstruction error for representing a data sample using the correct dictionary. A reconstruction term is introduced to encourage dictionary selection with minimal reconstruction error during training process. Therefore, the reconstruction term can be defined as:

$$R(D) = \sum_{k=1}^{K} \sum_{\forall v_q \in D^{(k)}} \|v_q - D^{(k)}\alpha_q^{(k)}\|_2 \qquad (10)$$

An atom v_q is assigned to dictionary $D^{(k^*)}$ if it satisfies:

$$k^* = \arg\min_k \|v_q - D^{(k)}\alpha_q^{(k)}\|_2 \qquad (11)$$

In Eq.10, the sparse coefficients can be solved by a set of initial dictionaries. Then, dictionaries are updated iteratively. Each time we solve Eq.10 by using the learned dictionaries from the previous iteration. The details are given in Section 5.

4.3. Label Term

In the dictionary learning based on multiple data attributes, the attribute labels within a sub-dictionary are encouraged to be consistent, as suggested by [5, 6]. Denote $N_{i,j}^k$ to be the number of labels with type j of attribute a_i in dictionary k. The label consistency can be evaluated by counting the maximal number of types in each attribute across all classes. The label term is given as:

$$L\,(\mathrm{D}) = \sum_{\mathrm{k}=1}^{\mathrm{K}} \left(\frac{1}{r} \sum_{i=1}^{r} \frac{N_{i,j^*}^k}{\sum_{j=1}^{n_i} N_{i,j}^k} \right), \text{where} \quad j^* = \arg\max_j N_{i,j}^k \qquad (12)$$

where j^* is the label type for attribute i, in dictionary k with the maximal number. The summation is normalized by the total attribute number r.

We calculate the increment of label number when adding a sample into the sub-dictionary. Let $\hat{N}_{i,j}^k$ be the number of labels with type j of attribute a_i in dictionary k after adding one sample v_q into this dictionary. A training sample v_q is assigned to dictionary k if it satisfies:

$$k^* = \arg\max_k \frac{1}{r} \sum_{i=1}^{r} ((\hat{N}_{i,j^*}^k - N_{i,j^*}^k) - \sum_{j=1, j\neq j^*}^{n_i} (\hat{N}_{i,j}^k - N)_{i,j}^k) \quad \textit{where} \quad j^* = \arg\max_j \hat{N}_{i,j}^k \qquad (13)$$

The above formulation includes two parts. The first part calculates the difference of the current maximal number of type after adding one sample into this dictionary. The second part accumulates the difference in the other types. Since any sample contains one label type in each attribute, if the label falls in the first part, the function is incremented by 1. Otherwise, it is -1. One can expect if all attributes of a sample fall in the categories with the maximal number, the function returns the value 1 (after normalization by r). For other cases, smaller values or negative values are given.

5. Optimization of MADL

The objective function of multi-attributed dictionary learning combines the compact term, reconstruction term and label term. The solution is obtained by minimizing the objective function given as follows:

$$\min_D C(D) + R(D) - L(D) \qquad (14)$$

Directly minimizing the object function is a non-convex problem. A *K-Medoids* clustering method [7] is exploited to find the solution iteratively: the most centrally located data sample in a cluster is selected as a centroid according to the learned distance matrix. Then, assign the rest of the samples to their closest centroids. In each iteration, centroids and clusters are updated according to the above objective function. The process is repeated until convergence. Details for each step are given as follows:

Initialization: We use the class configuration of the training set and adopt the number *K* to initialize the number of centroids. This avoids the major obstacle of *K-Means* based methods to predict the number of clusters *k* in prior. In our method, $X^{(k)}, k = 1, \ldots, K$ are used as initial dictionaries.

Distance between a pair of samples: We can note that both compact term and label term range from 0 to 1 whereas the reconstruction term does not. Here, a reconstruction error vector is defined to make its range from 0 to 1 as well so that these three terms can be combined easily. Define a reconstruction error vector associated with a data sample v_i as $e(v_i) = [e_1(v_i), \ldots, e_k(v_i)]$, where $e_k(v_i)$ is the reconstruction error of sparse coding using dictionary $D^{(k)}$. Denote $d_{recon}(v_q, v_l)$ the distance of reconstruction error of two samples. It can be constructed by the cosine distance of a pair between $e(v_q)$ and $e(v_l)$.

Find a new centroid: The centroid is supposed to have the minimal distance to all members within this cluster. The unified distance matrix learned in Section 3.2 is used to find centroids of clusters.

Cluster update: After new centroids are decided for all clusters, data points are assigned to new clusters according to their nearest centroids based on the summed-up distance of three terms.

Termination: The iteration is repeated to update the centroids and cluster. The process terminates when there is no change for clusters. The algorithm is given in Algorithm 1.

Algorithm 1 The MADL algorithm.

Initialization: Calculate the unified distance matrix. Set initial dictionary $D^{(k)}$ by $X^{(k)}$. Calculate centroid of $\bar{v}^{(k)}$ of $D^{(k)}$, $k = 1, \ldots, K$.
repeat
for each data point v_q
Calculate compact and label term by Eq.9 & Eq.13,
Solve sparse coefficients for v_q using $D^{(1)}, \ldots, D^{(k)}$,
Calculate $d_{recon}(v_q, \bar{v}^{(k)})$ using $e(v_q)$,
Compute the distance of v_q to each centroid $\bar{v}^{(k)}$,
Switch v_q to a new cluster if the distance decreases,
end
Update centroid $\bar{v}^{(k)}$ of $D^{(k)}, k = 1, \ldots, K$.
until no new assignment occurs
Output: $D^{(1)}, \ldots, D^{(k)}$.

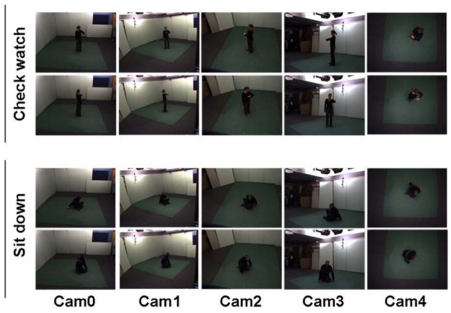


Figure 5. Sample images in the IXMAS Action Database.

It is proven that the convergence of the K-medoid algorithm is fast [8]. It occurs usually within six iterations.

Classification: After the iteration terminates, the category-dependent dictionaries are learned. For classification, a test sample is sparse coded by each sub-dictionary. The class label of the test sample is decided by counting the label with maximal number from non-zero coefficients using the dictionary with minimal reconstruction error.

6. Experimental Results

We apply the proposed Multi-Attributed Dictionary Learning (MADL) to the task of action recognition and face recognition in our experiments. Our method was evaluated in the following datasets: IXMAS Human Action Dataset [17], AR Face Database [13], and CMU PIE database [15]. Since our method is nearly parameter-free except for the sparsity factor when solving the sparse coefficients, there is no complicated parameter setting. Following the work [5], the sparsity factor was set to 30 in our experiments. We used the tool package *cvx* [3] for solving the optimization problem with sparse coding.

We compared our results with K-Means, SRC [18] and other dictionary learning algorithms: SPAMS [11], FDDL [19], K-SVD [1] and LCSVD [6]. K-Means is a traditional clutering method, which is used as a baseline method in our comparison. SRC achieved good performance for classification. Among the dictionary learning methods, SPAMS learns the dictionary by matrix factorization in an online learning manner. FDDL adopts the Fisher discrimination criterion into the dictionary learning, which also learns class-specified dictionaries. We also compare our method with LCSVD which uses class labels (single attribute only) in their formulation to learn dictionaries. We ran the codes of the previous works and our own program for the proposed algorithm on our training/testing datasets to compare the experimental results.

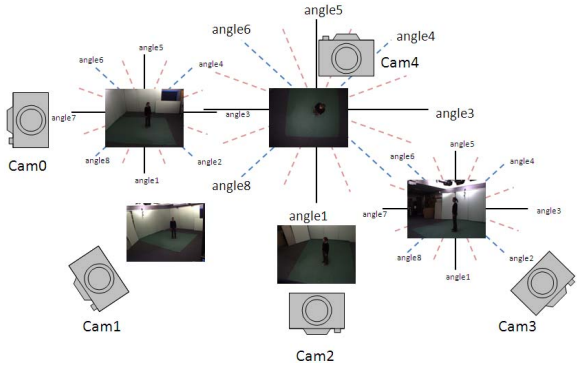


Figure 6. Angles of actions in the IXMAS Action Database.

Table 1. Recognition accuracy (%) on the IXMAS Human Action Database.

#Tr / #Te	495 / 1485	990 / 990	1485 / 495
SRC [18]	61.41	66.87	69.29
SPAMS [11]	59.53	67.27	68.08
FDDL [19]	42.97	47.98	45.25
K-SVD [1]	54.93	59.37	58.44
LCSVD1 [6]	38.04	61.36	63.60
LCSVD2 [6]	51.92	60.67	63.69
Proposed	**64.27**	**68.34**	**70.86**

6.1. Evaluation on IXMAS Action Database

IXMAS Human Action Database [17] contains eleven actions (check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, pick up). Each is performed three times by twelve actors. Each action is captured from five cameras observing the subjects with very different angles. Thus, it is also a multi-view dataset. Figure 5 shows some sample images of actions from IXMAS multi-view human action dataset. In this dataset, each actor performs actions at varying orientations and positions.

It is a challenging task to recognize human actions in this dataset because the same action recorded from the same

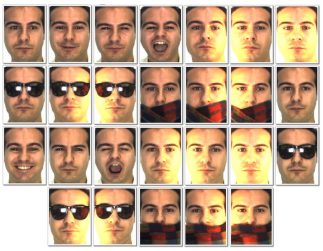


Figure 7. Sample images of AR Face Database

Table 2. Recognition accuracy (%) on the AR Face Database.

#Tr / #Te	8 / 18	13 / 13	21 / 5
K-Means	23.83	19.38	23.40
SPAMS [11]	76.72	86.85	94.80
FDDL [19]	87.28	94.38	98.40
LCSVD1 [6]	92.67	96.54	98.00
LCSVD2 [6]	**92.72**	96.46	98.40
Proposed	90.92	**97.15**	**100.00**

camera in different time exhibits large variations in appearance. Images are cropped with the height of human was scaled to M (M = 120). The motion-history image (MHI) [2] is extracted from the image sequences for spatio-temporal representation.

Two attributes are exploited in this dataset: action and angle. In the attribute of action, there are eleven types of actions. For the angle attribute, the angles were quantized into eight sections. Since the camera of top view (Cam4) is very different from the others. Eight angle sections are separated from those from Cam0 to Cam4, making it totally 16 angle sections, as depicted in Figure 6. We manually labeled the angle of each image and eliminated angles with no image. There are totally eleven angles as our second attribute. We extracted 1980 MHI images from the dataset for eleven actions. In this experiment, we used three data splits (SP): 495, 990 and 1485 MHI images were randomly selected for training while the rest of the images were used for testing. PCA was applied to reduce the data dimension to 400. The quantitative comparison of the experimental results is shown in Table 1. We can note that SRC shows good recognition results. By adopting multiple attributes, the proposed MADL outperforms LCSVD which uses only single attributes. In the meanwhile, MADL also provides the highest recognition accuracy in three different train and testing configurations.

6.2. Evaluation on AR Face Database

The AR database consists of over 4000 frontal images from 126 individuals. For each individual, 26 images were taken in two separate sessions. Since there are many kinds of variations in this dataset, we exploited five attributes: 1. *identity*, 2. *facial expression*, 3. *lighting*, 4. *sun glasses* and 5. *scarf*, as shown in Figure 7. A subset that contains 50 males and 50 females was chosen from the AR dataset. So, there are 100 types in the attribute of identity. The database contains four types of facial expressions: neutral, smile, anger and scream. In the lighting conditions, there are left light on, right light on and all side lights on. For sun glasses and scarf, some wear them and some do not. In the case that we only have 26 images for each person, we cannot have very fine types in each attribute. Otherwise, the image samples will not be enough to split the training and testing set. In such case, we defined binary types for each attribute: *with/without* expressions, light on, sun glasses and scarf.

The images were converted to grayscale in our implementation. In this experiment, we used three data splits: 8, 13 and 21 images from each person were selected for training and the rest of the images were used for testing. The numbers of training images randomly chosen for SP1 to SP3 from (expression, lighting, sun glasses, scarf) are (2, 2, 2, 2), (4, 3, 3, 3) and (6, 5, 5, 5), respectively. PCA was applied to reduce the data dimension to 1039. The quantitative comparison of the experimental results is shown in Table 3. We can note that the pure clustering for dictionary learning based only on data distance gave the worst results than all the other methods. Our MADL method achieved 100% accuracy using 21 training samples from one class. Compared to the results of LCSVD reported in [6], there is only slight difference by using our training/testing configuration (93.7% using 5 training samples and 97.8 using 20 training samples). The proposed MADL algorithm only achieved 90.92% accuracy in SP1. This is mainly caused by the problem of very few training samples. Initially, each class has eight dictionary atoms. After the termination of our optimization process, we found that some classes might have only 2 to 3 samples as their dictionary atoms. This limits the representation capability of those classes. Since we did not constrain the number of dictionary atoms, we will address this issue in our future work.

6.3. Evaluation on CMU PIE Face Database

The CMU PIE dataset is a relatively large face dataset. It contains over 40,000 facial images of 68 people. The face images were taken from 13 fixed cameras of different poses, under 24 illumination conditions and with 3 expressions for each person in separate sets, as shown in Figure 8. For the illumination category, there are 24 images of different illumination conditions under 13 different poses for each person. Therefore, there are totally 312 images for one person. For facial expression, three expressions, namely blink, smile and neutral expression, were obtained under 13 different poses. Therefore, we adopted two attributes in this dataset: *identity* and *pose*. The total number of types for

Figure 8. Sample images of CMU PIE Face Database

Table 3. Recognition accuracy (%) on the CMU PIE Human Action Database.

Split #	SP1	SP2	SP3
SRC [18]	84.85	86.76	86.63
SPAMS [11]	34.68	49.72	60.83
LCSVD1 [6]	60.88	59.73	60.29
LCSVD2 [6]	60.14	60.35	60.77
Proposed	**88.36**	**90.21**	**91.58**

identity and pose attributes are 68 and 13, respectively.

There are totally 39 images for one person. We set the training set by the following configurations: 7, 13, 21 images randomly chosen from one pose of one person for (expression, illumination) were (1, 6), (1, 12) and (2, 18), respectively. In total, 91, 169, 260 images were used for training for one person. The remaining images were used for testing. The face region of all training and testing images were cropped to 75 by 90. We processed all face images with PCA to reduce the dimension to 942. The recognition accuracies of the proposed MADL and other dictionary learning methods are listed in Table 4. We can see that the accuracies for different methods vary largely in the table. The SPAMS method produced quite low accuracies for all splits and the LCSVD method renders similar accuracies even though more training images were provided. The proposed MADL algorithm consistently provided the best recognition accuracies in all the experiments with three different splits.

7. Conclusion

We presented a novel multi-attributed dictionary learning algorithm for sparse coding in this paper. In order to take both data and the associated multiple attributes into consideration, we first proposed a joint distance matrix. An objective function is presented to learn compact, representative and attribute-consistent dictionaries. Experimental results have shown improved performance by using the proposed algorithm over the previous dictionary learning methods through the action classification and face recognition experiments.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Processing*, 54(11):4311 – 4322, nov. 2006.

[2] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, mar 2001.

[3] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, Apr. 2011.

[4] P. Hoel, S. Port, and C. Stone. *Introduction to Stochastic Processes*. Waveland Pr Inc, 1986.

[5] Z.-L. Jiang, Z. Lin, and L. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR*, 2011.

[6] Z.-L. Jiang, G.-X. Zhang, and L. Davis. Submodular dictionary learning for sparse coding. In *CVPR*, 2012.

[7] L. Kaufman and P. J. Rousseeuw. *Clustering by means of medoids*. Fac., Univ., 1987.

[8] M. Kirsten and S. Wrobel. Extending k-means clustering to first-order representations. In *ILP*, volume 1866, pages 112–129, 2000.

[9] A. Krause and V. Cevher. Submodular dictionary selection for sparse representation. In *ICML*, 2010.

[10] Y. Liu, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *ICCV*, june 2008.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19 – 60, 2010.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, june 2008.

[13] A. Martinez and R. Benavente. The ar face database. In *CVC Technical Report*, June 1998.

[14] I. Ramírez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, pages 3501–3508, 2010.

[15] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. PAMI*, 25(12):1615 – 1618, dec. 2003.

[16] F. Siyahjani and G. Doretto. Learning a context aware dictionary for sparse representation. In *ACCV*, 2012.

[17] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2):249 – 257, 2006.

[18] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. PAMI*, 31(2):210 –227, feb. 2009.

[19] M. Yang, L. Zhang, X.-C. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 2011.

[20] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2010.