LEARNING PATTERN TRANSFORMATION MANIFOLDS FOR CLASSIFICATION

Elif Vural and Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Laboratory - LTS4
CH-1015 Lausanne, Switzerland

ABSTRACT

Manifold models provide low-dimensional representations that are useful for analyzing and classifying data in a transformation-invariant way. In this paper we study the problem of jointly building multiple pattern transformation manifolds from a collection of image sets, where each set consists of observations from a class of geometrically transformed signals. We build the manifolds such that each manifold approximates a different signal class. Each manifold is characterized by a representative pattern that consists of a linear combination of analytic atoms selected from a continuous dictionary manifold. We propose an iterative algorithm for jointly building multiple manifolds such that the classification accuracy is promoted in the learning of the representative patterns. We present a DC (Difference-of-Convex) optimization scheme that is applicable to a wide range of transformation and dictionary models, and demonstrate its application to transformation manifolds generated by the rotation, translation and scaling of a reference image. Experimental results suggest that the proposed method yields a high classification accuracy compared to reference methods based on individual manifold building or locally linear manifold approximations.

***Index Terms*—** Manifold learning, pattern transformation manifolds, pattern classification, transformation-invariance, sparse approximations

1. INTRODUCTION

Manifold models provide low-dimensional representations that facilitate the treatment of signals. In a setting where different signal classes are represented with different manifolds, the class label of a query signal can be estimated by comparing its distance to the candidate manifolds. In this study, we focus on pattern transformation manifolds (PTMs). A PTM represents images generated from a reference pattern that undergoes a set of geometric transformations. For instance, the images obtained by the rotation and scaling of a reference pattern form a PTM. A set of PTMs representing different classes can be used for the classification of geometrically transformed images. In this work, we address the problem of constructing PTMs that are good for transformation-invariant classification.

We consider multiple sets of geometrically transformed observations, where each set consists of a different class of images. We build on our previous work [1], where we propose a method for building a PTM for signal approximation. Here, we extend this study to the joint construction of multiple PTMs such that each PTM represents one image class, and the images can be accurately classified with respect to their distances to the constructed PTMs. Each PTM is formed by building a representative pattern as a linear combination of some parametric atoms selected from a continuous dictionary manifold. The dictionary is formed by smooth geometric transformations of an analytic mother function. We propose an iterative method that jointly selects atoms for all classes using the training data of each class. We define an objective function that is a weighted combination of a classification and a data approximation error term. The atom selection is then formulated as an optimization problem with respect to the parameters and the coefficients of the atoms. We propose a two-stage solution for atom selection. We first obtain an initial solution by finding an approximation of the objective function in a DC form [2] and minimizing it with a DC solver. Then, we improve this solution, by approximating the manifold distance with the tangent distance in the objective function and minimizing the objective function with gradient descent.

Our study is linked to two main topics; manifold learning and sparse signal representations. Firstly, our PTM building approach can be seen as a special instance of manifold learning with prior information on the data model. Many methods have been recently proposed in this field, such as the ISOMAP [3] and LLE [4] algorithms. However, such generic approaches do not use any information about the model generating the data, and they have the restriction that the generation of novel manifold points and the parametrization and classification of new data is not straightforward. Our method is particularly designed for manifolds generated by 2-D geometric transformations, and also provides natural solutions for these issues as the manifold is constructed through a parametric mapping.

We further remark the following about the relation between this work and the field of sparse signal approximations. The main contributions of this work in comparison with sparse approximation algorithms such as Matching Pursuit (MP) [5] or Simultaneous Orthogonal Matching Pursuit (SOMP) [6] are that we achieve transformation-invariance due to the transformation manifold model, set classification objectives as well as signal approximation, and learn atoms from a dictionary manifold rather than a predefined discrete dictionary. Hence, it is also possible to link our work to transformation-invariant dictionary learning.

2. PROBLEM FORMULATION

Consider a visual pattern $p \in L^2(\mathbb{R}^2)$. Let $\Lambda \subset \mathbb{R}^d$ be a closed parameter domain, and $\lambda \in \Lambda$ be a parameter vector. We define $A_\lambda(p) \in L^2(\mathbb{R}^2)$ as the pattern that is generated by applying a smooth geometric transformation specified by λ to p. Then, we define the transformation manifold of p as

$$\mathcal{M}(p) = \{U_\lambda(p) : \lambda \in \Lambda\} \subset \mathbb{R}^n, \tag{1}$$

where $U_\lambda(p) \in \mathbb{R}^n$ is an n-dimensional discretization of $A_\lambda(p)$.

This work has been partly funded by the Swiss National Science Foundation under Grant 200020_132772.

Now let $\mathcal{U} = \bigcup_{m=1}^{M} \mathcal{U}^m \subset \mathbb{R}^n$ be a collection of images consisting of M classes, where each subset $\mathcal{U}^m = \{u_i^m\}_{i=1}^{N_m}$ consists of N_m geometrically transformed observations of a visual signal of class m. Our objective is to represent each set $\mathcal{U}^m$ by a transformation manifold $\mathcal{M}(p^m)$ that is generated by the geometric transformations of a representative pattern p^m. Let us denote

$$\mathcal{M}^m = \mathcal{M}(p^m) = \{U_\lambda(p^m), \lambda \in \Lambda\} \subset \mathbb{R}^n. \tag{2}$$

We would like to build the manifolds $\{\mathcal{M}^m\}$ such that they provide a good representation of the images in $\mathcal{U}$ and also permit to classify them accurately by manifold distance computation.[1] Hence, in the construction of the manifolds, we formulate the objective function as a weighted combination of two terms E_a and E_c, which respectively represent approximation and classification errors. The approximation error E_a is given by the sum of the squared distances of images to the manifold of the same class.

$$E_a = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|e_i^m\|^2 = \sum_{m=1}^{M} \sum_{i=1}^{N_m} d^2(u_i^m, \mathcal{M}^m) \tag{3}$$

Here, $d(u, \mathcal{M}(p)) = \min_{\lambda \in \Lambda} \|u - U_\lambda(p)\|$ denotes the distance between the image u and the manifold $\mathcal{M}(p)$, and the notation $\|.\|$ stands for the ℓ_2-norm in $\mathbb{R}^n$. We assume that the class label of an image is determined with respect to its distance to the manifolds, i.e., an image is assigned the label of the manifold with smallest distance to it. We define a misclassification indicator function I such that

$$I(u_i^m) = \begin{cases} 0, & \text{if } d(u_i^m, \mathcal{M}^m) < \min_{r \neq m} d\big(u_i^m, \mathcal{M}^r\big) \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

Then, the classification error

$$E_c = \sum_{m=1}^{M} \sum_{i=1}^{N_m} I(u_i^m) \tag{5}$$

is the total number of misclassified images. We would like to compute $\{\mathcal{M}^m\}_{m=1}^{M}$ such that the weighted error $E = E_a + \alpha\, E_c$ is minimized, where $\alpha > 0$ is a coefficient adjusting the weight between the approximation and classification terms.

In the construction of the manifolds $\{\mathcal{M}^m\}$, we propose to build each p^m as a sparse linear combination of some parametric atoms from a dictionary manifold

$$\mathcal{D} = \{B_\gamma(\phi) : \gamma \in \Gamma\} \subset L^2(\mathbb{R}^2). \tag{6}$$

Each atom $B_\gamma(\phi) \in L^2(\mathbb{R}^2)$ is derived from an analytic mother function $\phi \in L^2(\mathbb{R}^2)$ through a smooth geometric transformation specified by a parameter vector γ. The parameter domain Γ is assumed to be a closed and convex subset of $\mathbb{R}^s$, where s is the number of transformation parameters generating $\mathcal{D}$. Let us write $\phi_\gamma = B_\gamma(\phi)$ for simplicity. Then, would like to obtain each representative pattern in the form $p^m = \sum_{j=1}^{K_m} c_j^m \phi_{\gamma_j^m}$ as a combination of K_m atoms from $\mathcal{D}$. Having set our notation, now we formulate the multiple manifold learning problem as follows.

Problem 1. *Given image sets $\{\mathcal{U}^m\}$ of different classes, an analytic mother function ϕ and sparsity constraints $\{K_m\}$; compute a set of atom parameter vectors $\{\gamma_j^m\} \subset \Gamma$, coefficients $\{c_j^m\} \subset \mathbb{R}$, and transformation parameter vectors $\{\lambda_i^m\} \subset \Lambda$ for $m = 1, \ldots, M$, $j = 1, \ldots, K_m$ and $i = 1, \ldots, N_m$ by minimizing*

$$E = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \left(\|u_i^m - U_{\lambda_i^m}\big(\sum_{j=1}^{K_m} c_j^m \phi_{\gamma_j^m}\big)\|^2 + \alpha\, I(u_i^m) \right). \tag{7}$$

3. JOINT PTM BUILDING ALGORITHM

The cost function E in Problem 1 has a complex dependence on the atom parameters and the initially unknown transformation parameters. Therefore, it is hard to solve Problem 1 optimally. We present a constructive solution based on our previous work [1], and build the patterns $\{p^m\}$ iteratively with joint atom selection.

Since the atom selection involves an approximation of the manifolds around each image, the algorithm initially needs a tentative assignment of transformation parameters. In (7) each vector λ_i^m corresponds to the projection of u_i^m onto $\mathcal{M}^m$. We assign $\{\lambda_i^m\}$ by picking a reference pattern Ψ^m (that can be selected within the data set) for each class and then projecting each $\mathcal{U}^m$ onto $\mathcal{M}(\Psi^m)$. Similarly, we also initialize the cross-projection vectors $\lambda_i^{m,r} = \arg\min_{\lambda \in \Lambda} \|u_i^m - U_\lambda(p^r)\|$ by projecting u_i^m onto $\mathcal{M}(\Psi^r)$.

Then, we construct $\{p^m\}$ gradually by simultaneously adding them new atoms. Now we consider the j-th iteration of the algorithm. Let p_{j-1}^m denote the already computed patterns, and γ_j^m and c_j^m be the parameters and the coefficients of the new atoms, such that $p_j^m = p_{j-1}^m + c_j^m \phi_{\gamma_j^m}$ is the new pattern of each class. In the j-th iteration, we would like to optimize γ_j^m and c_j^m such that E is minimized. Let us simply write $\gamma^m = \gamma_j^m$, $c^m = c_j^m$. Then $\gamma = [\gamma^1\, \gamma^2\, \ldots\, \gamma^M]$ and $c = [c^1\, c^2\, \ldots\, c^M]$ are the optimization variables of the j-th iteration. We consider E as a function of γ and c, and minimize it through a two-stage optimization.

In the first stage, we obtain an approximation $\tilde{E}$ of E in a DC form. A DC function is a function that can be written as the difference of two convex functions [2]. The global minimum of DC functions can be found with DC solvers [2]. We obtain $\tilde{E}$ in the form $\tilde{E} = \tilde{E}_a + \alpha\, \tilde{E}_c$, where $\tilde{E}_a$ and $\tilde{E}_c$ are approximations of E_a and E_c. The term $\tilde{E}_a$ is the total data approximation error

$$\tilde{E}_a = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|\tilde{e}_i^m\|^2 = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|v_i^m - c^m\, U_{\lambda_i^m}(\phi_{\gamma^m})\|^2$$

where the parameters λ_i^m are the ones computed at the end of iteration $j-1$, and $v_i^m = u_i^m - U_{\lambda_i^m}(p_{j-1}^m)$. Note that $\tilde{E}_a \neq E_a$ in general since the parameter vectors λ_i^m change when a new atom is added to the pattern. The approximation $\tilde{E}_a \approx E_a$ is useful since λ_i^m is not known in a closed form as a function of c^m and γ^m.

Then, we derive $\tilde{E}_c$ in the following way. The discontinuous misclassification indicator function in (4) corresponds to the unit step function of $d^2(u_i^m, \mathcal{M}^m) - d^2\big(u_i^m, \mathcal{M}^{r^*}\big)$, where $r^* = \arg\min_{r \neq m} d\big(u_i^m, \mathcal{M}^r\big)$. We replace the unit step function with the analytic sigmoid function $S(x) = (1 + e^{-\beta x})^{-1}$, and then use a linear approximation of the sigmoid function for each u_i. Then, $\tilde{E}_c$ can be derived in the form (see Section IV.B of [7])

$$\begin{aligned}\tilde{E}_c = &\sum_{m=1}^{M} \sum_{i=1}^{N_m} \eta_i^m\, \|v_i^m - c^m\, U_{\lambda_i^m}(\phi_{\gamma^m})\|^2 \\ &- \sum_{m=1}^{M} \sum_{(i,k) \in R^m} \eta_i^k\, \|v_i^{k,m} - c^m\, U_{\lambda_i^{k,m}}(\phi_{\gamma^m})\|^2\end{aligned}$$

[1] Although the images $\{u_i^m\}$ of a particular class m do not have to be exactly on a PTM, our method is based on the assumption that one can find a pattern p^m such that $\{u_i^m\}$ lie "around" its PTM. In the case that the images of the same class deviate significantly from the representative PTM, the method can be adapted to do a preliminary clustering and learn more than one representative manifold per class. Our method is designed for applications where images can be well-approximated with 2-D transformations. Some examples can be microscopic images, satellite images, handwritten digits and letters, and images generated by orthographic projections.

where η_i^m are some constants, $v_i^{k,m} = u_i^k - U_{\lambda_i^{k,m}}(p_{j-1}^m)$, and $R^m = \{(i,k)\}$ consists of the pairs of data i and class k indices of images that do not belong to class m but have $\mathcal{M}^m$ as their closest manifold among all manifolds except their own manifold $\mathcal{M}^k$. It can be shown that $\tilde{E} = \tilde{E}_a + \alpha\,\tilde{E}_c$ is a DC function of γ and c, which follows from the smoothness of $\tilde{E}$; and that a DC decomposition of $\tilde{E}$ is computable if a DC decomposition for the components of $U_\lambda(\phi_\gamma)$ is known (see [7], Proposition 3). We minimize $\tilde{E}$ using the cutting plane algorithm [2] and estimate a coarse solution, which is used as an initial solution in the second stage.

In the second stage of the optimization, we define a refined approximation $\hat{E}$ of E based on the tangent distances of images to the manifolds. We minimize $\hat{E}$ with a gradient-descent algorithm. Let us use the notation $p^m = p_j^m$ for now. We derive $\hat{E}$ by replacing the manifold distances by tangent distances, i.e., we use the approximation $d^2(u_i^k, \mathcal{M}^m) \approx d^2(u_i^k, \mathcal{S}_i^k(p^m))$, where $\mathcal{S}_i^k(p^m)$ is the first-order approximation of $\mathcal{M}^m$ around the point $U_{\lambda_i^{k,m}}(p^m)$. Let $w_i^m = u_i^m - U_{\lambda_i^m}(p^m)$ and $w_i^{k,m} = u_i^k - U_{\lambda_i^{k,m}}(p^m)$. Then the function E_a in (3) is approximated by ([7], Section IV.B)

$$\hat{E}_a = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \| w_i^m - T_i^m \left((T_i^m)^{\mathrm{T}} T_i^m \right)^{-1} (T_i^m)^{\mathrm{T}} w_i^m \|^2 .$$

Similarly, the classification error function in (5) is approximated by

$$\hat{E}_c = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \eta_i^m \| w_i^m - T_i^m \left((T_i^m)^{\mathrm{T}} T_i^m \right)^{-1} (T_i^m)^{\mathrm{T}} w_i^m \|^2 -$$
$$\sum_{m=1}^{M} \sum_{(i,k)\in R^m} \eta_i^k \| w_i^{k,m} - T_i^{k,m} \left((T_i^{k,m})^{\mathrm{T}} T_i^{k,m} \right)^{-1} (T_i^{k,m})^{\mathrm{T}} w_i^{k,m} \|^2 .$$

Here T_i^m and $T_i^{k,m}$ denote the $n \times d$ matrices whose columns are the tangent vectors to the manifold $\mathcal{M}^m$ at respectively the points $U_{\lambda_i^m}(p^m)$ and $U_{\lambda_i^{k,m}}(p^m)$. Finally, we define $\hat{E} = \hat{E}_a + \alpha\,\hat{E}_c$.

The minimization of $\tilde{E}$ and $\hat{E}$ determines a solution for γ and c. We update the patterns of each class such that $p_j^m = p_{j-1}^m + c^m \phi_{\gamma^m}$. Then, we recompute the transformation parameter vectors $\{\lambda_i^m\}$ and $\{\lambda_i^{m,r}\}$ by projecting the images onto the new manifolds. Since the target application area of this method is image classification, we check if the updates on the manifolds reduce the classification error E_c in (5). If E_c is not reduced we reject the updates and pass to the next iteration. We update the values of the algorithm parameters β and α at the end of each iteration as explained in Section IV.C of [7]. Therefore, even if the optimization fails in one iteration, it may succeed in the next one. We continue the iterations until the classification error E_c converges. The termination of the algorithm is guaranteed by constraining E_c to be non-increasing during the iterations, which in return stabilizes the objective function E. We call this method Joint Parameterized Atom Selection (JPATS) and give a sketch of it in Algorithm 1.

Algorithm 1 Joint Parameterized Atom Selection (JPATS)

```
1:  Input:
    $\mathcal{U} = \bigcup_{m=1}^{M} \mathcal{U}^m$: Set of observations for $M$ signal classes
2:  Initialization:
3:  Determine tentative parameter vectors $\{\lambda_i^{m,r}\}$ by projecting $\{u_i^m\}$ on
    the transformation manifolds $\{\mathcal{M}(\Psi^m)\}$ of reference patterns $\{\Psi^m\}$.
4:  $p_0^m = 0$ for $m = 1, \ldots, M$.
5:  $j = 0$.
6:  Initialize the sigmoid parameter $\beta$ and the weight parameter $\alpha$.
7:  repeat
8:      $j = j + 1$.
9:      Optimize the joint atom parameters $\gamma = [\gamma^1\ \gamma^2\ \ldots\ \gamma^M]$ and coeffi-
        cients $c = [c^1\ c^2\ \ldots\ c^M]$ with DC programming such that the error
        $\tilde{E}$ is minimized.
10:     Further optimize $\gamma$ and $c$ with gradient descent such that the refined
        error $\hat{E}$ is minimized.
11:     Update $p_j^m = p_{j-1}^m + c^m\,\phi_{\gamma^m}$ for $m = 1, \ldots, M$ if $c^m$ is signif-
        icant.
12:     Update the parameter vectors $\{\lambda_i^{m,r}\}$.
13:     Update $\beta$ and $\alpha$.
14:     Check if the new manifolds reduce the classification error $E_c$. If not,
        reject the updates on $p^m$ and $\{\lambda_i^{m,r}\}$, and go back to 9.
15: until the classification error $E_c$ converges
16: Output:
    $\{p^m\} = \{p_j^m\}$: A set of patterns whose transformation manifolds
    $\{\mathcal{M}^m\}$ represent the data classes $\mathcal{U}^m$
```

4. EXPERIMENTAL RESULTS

We now evaluate the JPATS method with experiments on transformation invariant image classification. We use a manifold model

$$\mathcal{M}(p) = \{U_\lambda(p) : \lambda = (\theta, t_x, t_y, s) \in \Lambda\} \subset \mathbb{R}^n \qquad (8)$$

where the parameters θ, t_x, t_y and s denote respectively the rotation, horizontal and vertical translations, and isotropic scale change. The dictionary $\mathcal{D}$ is derived from the Gaussian mother function $\phi(x,y) = \sqrt{2/\pi}\, e^{-(x^2+y^2)}$ with respect to the model

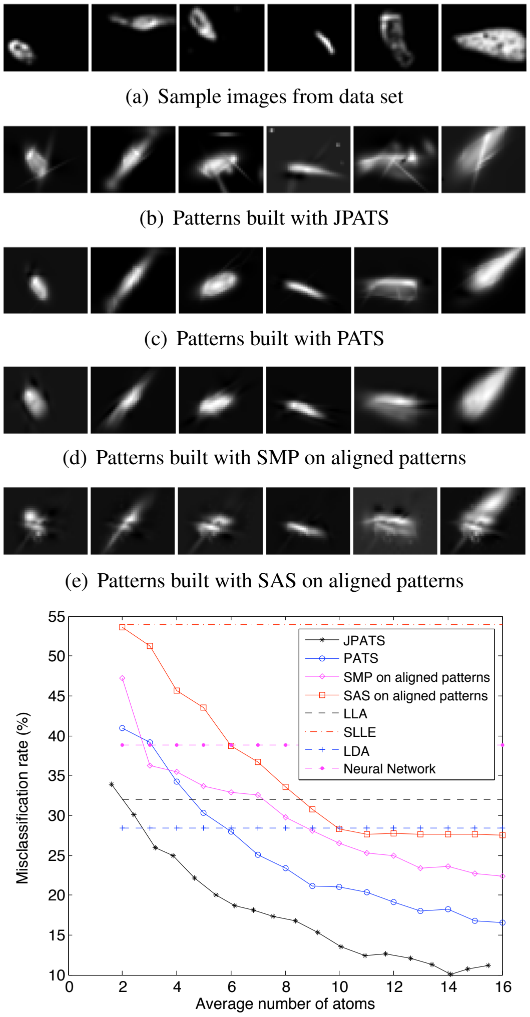
$$\mathcal{D} = \{B_\gamma(\phi) : \gamma = (\psi, \tau_x, \tau_y, \sigma_x, \sigma_y) \in \Gamma\} \subset L^2(\mathbb{R}^2) \qquad (9)$$

where ψ is a rotation parameter, τ_x and τ_y denote translations in x and y directions, and σ_x and σ_y represent anisotropic scalings in x and y directions. The computation of the DC decomposition of $U_\lambda(\phi_\gamma)$ and $\tilde{E}$ for this setup is explained in Appendix C of [7].

The experiment is done on some sequences from the microbiology video collection of the Natural History Museum [8], which contains short video clips of living protists. We run the experiment on 6 different species (Discocephalus sp., Epiclintes ambiguus, Oxytricha sp., Scyphidia sp., Stentor roeseli, Stylonychia sp.), and we use three sample videos for each one. Each species is considered as a different class. The transformation manifold in (8) provides a suitable model for the frames, as the rotation and translation parameters describe well the movements of the protists, and the isotropic scale parameter compensates for zoom changes. However, there is still some deviation from the manifold model, as a result of noise, small nonrigid protist articulations and occasional recording of different individuals of the same species in different sample videos. For each species, we experiment on a subset of frames from all three video sequences. We preprocess the frames by conversion to greyscale, smoothing and thresholding. Then for each class, we randomly select 70 training and 35 test images among the preprocessed frames.

In the experiments we compare JPATS with the following methods. Firstly, we test three methods where a PTM is built individually for each class by forming a representative pattern. The PATS method proposed in [1] is a pure approximation-based method that corresponds to a special case of JPATS with $\alpha = 0$. In the methods "SMP on aligned patterns" and "SAS on aligned patterns" we use the aligned versions of data set images and select a set of Gaussian atoms using respectively the SMP [9] and SAS[2] [10] algorithms.

[2] The SAS algorithm involves a class separability cost function in addition to an approximation error function. We set the weight factor to $\lambda = 2$ in [10].



(a) Sample images from data set

(b) Patterns built with JPATS

(c) Patterns built with PATS

(d) Patterns built with SMP on aligned patterns

(e) Patterns built with SAS on aligned patterns

(f) Classification errors of test images

Fig. 1. Performance of the classification-driven learning algorithms on the microbiological images data set

Then, we gradually construct representative patterns by adding the selected atoms weighted by their average coefficients. In all of these methods we apply the algorithms on the training images in order to build PTMs. Then we compute the misclassification rate of the test images. The class label of a test image is estimated by identifying the smallest distance between the test image and the class-representative PTMs. Then, in the LLA (Locally Linear Approximation) method, for each test image, we identify its $(d+1)$-nearest neighbors among the training images of each class and compute the distance between the test image and the plane passing through the nearest neighbors. Then we classify the test image according to its distance to these planes. In the SLLE method we compute a low-dimensional embedding of the training images with the SLLE algorithm [11] and assign the class labels of the test images via nearest-neighbor classification in the embedded domain. Finally, we compare our method to LDA and Neural Networks on aligned images.

In Figure 1(a), a sample data set image is shown for each class. Some typical representative patterns computed with the JPATS and PATS algorithms and the reference methods are shown in Figures 1(b)-1(e). Figure 1(f) shows the misclassification rates of test images (in percentage) vs. the number of atoms per class in the progressive construction of the representative patterns. The graph is obtained by averaging the results of 5 repetitions of the experiment with different training and test sets. The results suggest that the PATS algorithm computes a visually more meaningful approximation of the input images than JPATS, however, the features discriminating different classes from each other are better highlighted in the patterns computed with JPATS. Figure 1(f) shows that the JPATS method yields the best classification performance in general. Moreover, due to the sparse sampling and the noisy structure of the data set, the methods based on computing PTMs generally perform better than learning methods relying on local linearity assumptions. Similar results are obtained on handwritten digit images in [7], which we do not present here due to lack of space.

5. CONCLUSIONS

We have studied the problem of jointly building multiple pattern transformation manifolds for the transformation-invariant approximation and classification of sets of visual signals. The manifold learning problem is cast as the simultaneous construction of a representative pattern for each class as a linear combination of smooth parametric atoms. The manifolds are then created by geometric transformations of these patterns. The smoothness of the manifolds is ensured by the smoothness of the atoms. Experimental results show that the proposed method provides a good classification accuracy compared to reference methods. The presented method is suitable for the representation and classification of unregistered data that approximately conforms to a 2-D pattern transformation model with a known transformation type.

6. REFERENCES

[1] E. Vural and P. Frossard, “Learning pattern transformation manifolds with parametric atom selection,” in *Proc. Sampling Theory and Applications*, May 2011.

[2] R. Horst and N. V. Thoai, “DC programming: overview,” *J. Optim. Theory Appl.*, vol. 103, pp. 1–43, October 1999.

[3] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction.,” *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[4] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, Dec. 2000.

[5] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[6] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, “Algorithms for simultaneous sparse approximation part I: Greedy pursuit,” *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[7] E. Vural and P. Frossard, “Learning smooth pattern transformation manifolds,” available at http://arxiv.org/abs/1112.5640.

[8] Natural History Museum, “Microbiology video collection,” 2011.

[9] R. Sala Llonch, E. Kokiopoulou, I. Tošić, and P. Frossard, “3d face recognition with sparse spherical representations,” *Pattern Recogn.*, vol. 43, no. 3, pp. 824–834, Mar. 2010.

[10] E. Kokiopoulou and P. Frossard, “Semantic coding by supervised dimensionality reduction,” *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 806–818, 2008.

[11] D. De Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R. Duin, “Supervised locally linear embedding.,” in *Proc. Int. Conf. Art. Neur. Networks*, 2003, pp. 333–341.