IEICE TRANS. INF. & SYST., VOL.E96–D, NO.12 DECEMBER 2013

PAPER

Nonlinear Metric Learning with Deep Independent Subspace Analysis Network for Face Verification

Xinyuan CAI[†a)], **Chunheng WANG**[†], **Baihua XIAO**[†], *Nonmembers*, *and* **Yunxue SHAO**[†], *Student Member*

SUMMARY Face verification is the task of determining whether two given face images represent the same person or not. It is a very challenging task, as the face images, captured in the uncontrolled environments, may have large variations in illumination, expression, pose, background, *etc*. The crucial problem is how to compute the similarity of two face images. Metric learning has provided a viable solution to this problem. Until now, many metric learning algorithms have been proposed, but they are usually limited to learning a linear transformation. In this paper, we propose a nonlinear metric learning method, which learns an explicit mapping from the original space to an optimal subspace using deep Independent Subspace Analysis (ISA) network. Compared to the linear or kernel based metric learning methods, the proposed deep ISA network is a deep and local learning architecture, and therefore exhibits more powerful ability to learn the nature of highly variable dataset. We evaluate our method on the Labeled Faces in the Wild dataset, and results show superior performance over some state-of-the-art methods.

key words: *metric learning, independent subspace analysis, deep learning architecture, face verification*

1. Introduction

Face recognition, as one of the major biometric technologies, has attracted much attention in both industrial and research communities. It has become increasingly important owing to the availability of huge amounts of face images on the web, and increasing demands for higher security. There has been a lot of progress made in this area, and many face recognition systems have been developed. The operation of face recognition systems can be divided into two modes: identification and verification [1]. In the identification mode, the system compares the given probe image with all the gallery images and finds its closest match. The assumption is that the probe image and its closest image in the gallery belong to the same person. A more complicated version of this problem would be to include the possibility that the person of the probe image may not be present in the gallery. So the system has to decide whether the person with the highest rank is a correct match or not. In the verification mode, someone claims that he or she is a particular person. The system verifies this assertion by matching the probe against the gallery entry corresponding to the claimed identity. The system accepts the claim if the matching score lies above a predetermined operating threshold; otherwise the claim is rejected. More generally, face verification refers to deciding whether two images depict the same person or not.

In this paper, we focus on the face verification problem. In the constrained situations, where lighting, pose, facial wear and expressions can be controlled, automated face recognition can achieve satisfactory performance. While in the unconstrained environment, the variation caused by the changes in illumination, pose or others, could be larger than that caused by the identity changes. Therefore, the performance degrades significantly. Metric learning has provided a viable solution for the unconstrained face verification problem by comparing the image pairs based on the learned metric, which could suppress the variations in the unconstrained environment [2]. Most metric learning methods attempt to learn an appropriate similarity measure from the labeled side information, which are often available in the form of pairwise constraints, *i.e.* pairs of similar or dissimilar data points [3]. A common theme in metric learning is to learn a distance metric such that the distance between similar examples should be relatively smaller than that between dissimilar examples. Although the distance metric can be a general function, the most prevalent one is the Mahalanobis metric. It is equivalent to first applying a linear transformation, then computing Euclidean distance in the new subspace. Nevertheless, in many situations, a linear transformation is not powerful enough to capture the underlying data manifold and often fails to give desired performance in high dimensional space. Therefore, we need to resort to more powerful non-linear transformations. The kernel-based approach can achieve this goal. They implicitly map the original space to a high dimensional space by kernel-tricks. However, they have to compute the kernel similarity between each testing sample with training samples. Therefore, they behave almost like template-based approaches, and often have difficulty in handling large datasets [4]. Moreover, if the chosen kernel cannot well reflect the true class-related structure of data, the performance will be unsatisfactory.

We propose an explicit nonlinear metric learning method by using deep Independent Subspace Analysis (ISA) network. ISA [5] is a variant of Independent Component Analysis (ICA), and it can be described as a two-layered network (as shown in Fig. 1 (b)). An advantage of ISA is that it can learn receptive fields similar to the V1 area of visual cortex when applied to static images [6]. However, a disadvantage of ISA is that it can be very slow to train

Manuscript received March 5, 2013.
Manuscript revised June 27, 2013.

[†]The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
a) E-mail: xinyuan.cai@ia.ac.cn

DOI: 10.1587/transinf.E96.D.2830

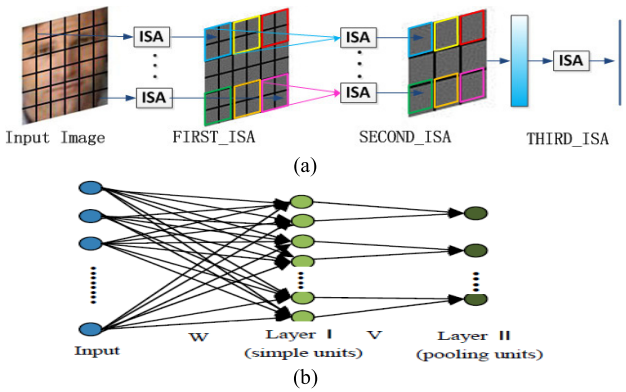

Fig. 1 (a) the architecture of the proposed Deep Independent Subspace Analysis networks; (b) the neural network architecture of an ISA network.

when the dimension of the input data is large. Therefore, we employ the ISA as a local network, and stack the local networks in a deep architecture for the high-resolution images (as shown in Fig. 1 (a)). The deep ISA networks can be regarded as an explicit nonlinear mapping function, which transforms the features from the original space to another subspace. The original ISA algorithm is unsupervised, so the learned features may not be suitable for some task-related objectives. In order to get discriminative features, we combine the side information constraints of metric learning with ISA, and formulate it as an appropriate optimization problem. Furthermore, we employ the greedy layer-wise pre-training and fine-tuning schemes to get the optimal solution. The proposed method is evaluated on the Labeled Faces in the Wild (LFW) [8] benchmark. And the result demonstrates superior performance over some state-of-the-art methods.

Some contents of this paper have been reported in our conference paper [9]. However, this paper extends it in several ways: (i) more technical details are given; (ii) more experiments are done. The remainder of this paper is organized as follows. In Sect. 2, we review some related works of distance metric learning and deep learning. In Sect. 3, we describe the details of the proposed Nonlinear Metric Learning with Deep Independent Subspace Analysis network (NLML_DISA). Experimental results and analysis are provided in Sect. 4. Finally, we summarized the contributions in Sect. 5.

2. Related Works

2.1 Distance Metric Learning

Distance metrics are fundamental concepts in machine learning, and are crucial in many real-world applications (e.g. [2], [10]). There has been considerable research on metric learning over the past few years [3]. The literature in metric learning can be split into two main categories: manifold learning and supervised or semi-supervised metric learning. Manifold learning is a kind of unsupervised metric learning. Its key idea is to learn an underlying low-dimensional manifold preserving the distance or structure between observed data points (such as ISOMAP [11], LLE [12]). The supervised or semi supervised approaches try to learn metrics by keeping points of the same class close while separating points from different classes. This paper relates to the latter.

One of the representative works of supervised metric learning is [13], which formulates the distance metric learning under side information constraints as a constrained convex programming problem. Let $C = \{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\}$ be a collection of data points, where T is the number of samples and each $x^{(i)} \in R^n$ is a data vector. The set of equivalent constraints is denoted by:

$$S = \{(x^{(i)}, x^{(j)}) \,|\, x^{(i)} \text{ and } x^{(j)} \text{ belong to the same class}\}$$

and the set of inequivalent constraints is denoted by

$$D = \{(x^{(i)}, x^{(j)}) \,|\, x^{(i)} \text{ and } x^{(j)} \text{ belong to different classes}\}$$

Under the distance metric $A \in R^{n \times n}$, the distance between any two data points x and y can be expressed as:

$$d_A^2(x, y) = \|x - y\|_A^2 = (x - y)^T A (x - y) \tag{1}$$

Given the constraints in S and D, Xing *et al.* [13] formulates the problem of metric learning into the following convex programming problem:

$$\min_A \sum_{(x^{(i)}, x^{(j)}) \in S} d_A^2\left(x^{(i)}, x^{(j)}\right) \tag{2}$$

$$s.t \;\; A \geq 0, \quad \sum_{(x^{(i)}, x^{(j)}) \in D} d_A^2\left(x^{(i)}, x^{(j)}\right) \geq 1 \tag{3}$$

The objective term is to make the distance between similar pairs as small as possible. The positive semi-definite constraint ($A \geq 0$) is needed to ensure the nonnegative distance between any two data points and the triangle inequality. The third term is to make the distance between dissimilar pairs at least larger than one. A is symmetric positive semi-definite, and it can be decomposed as $A = W^T W$. In the learned metric, the distance between any two points can be written as $d_A^2(x, y) = (x - y)^T A (x - y) = (Wx - Wy)^T (Wx - Wy)$. Thus, the traditional metric learning is equivalent to learn a linear transformation matrix W, and then compute Euclidean distance in the transformed subspace.

Following the above general approach, several methods are proposed to specifically address k-Nearest Neighbor classification. They either introduce constraints on absolute distance between pairs, such as Neighborhood Component Analysis [14], Maximally Collapsing Component Analysis [15], or constraints on relative distance such as Large Margin Nearest Neighbor [16] or Large Margin Component Analysis [17]. In these approaches, the k nearest neighbors of each point are explicitly selected, and the distance metric is learned in a way that for each training point, the neighbors from other classes are always farther than the neighbors from the same class up to a margin. However, these approaches require the class labels of all the training points, and are thus not adapted to the problems for which only side-information or pair-wise constraints are available.

The recently proposed Information Theoretic Metric Learning (ITML) [18] and Logistic Discriminate Metric Learning (LDML) [2] are designed to deal with general pair-wise constraints. Furthermore, ITML considers not only the pair-wise constraints, but also a prior knowledge on the learned metric A. This is done by regularizing the matrix A such that it is as close as possible to a known prior matrix A_0. Moreover, the closeness is measured in the Kullback-Leibleer divergence criterion. LDML uses a robust probabilistic model to estimate the similarity between two data points in the learned metric, and applies maximum log-likelihood to learn the optimal metrics. The number of parameters increases with the square of the dimensionality of the input space, so for the high dimensional input space, both ITML and LDML must be preceded by a step of prior dimensionality reduction, which may result in loss of information.

Most of the above distance metric learning approaches learn a Mahalanobis matrix, which is equivalent to learn a linear transformation matrix for the original space. But the linear transformation has limited power to describe the underlying data manifold. Kwok *et al.* [19] shows that the general framework of distance metric learning can be extended to non-linear problems using the kernel trick, which maps the data vectors into a high dimensional space implicitly. However, the kernel-based methods need to compute the kernel similarity between the testing sample and each training sample. Therefore, they require high computational cost for testing, especially when there are a large number of training samples. Our proposed method tries to deal with the limitation of traditional metric learning methods, and learn an explicit nonlinear transformation, which could exhibit good generalization properties.

2.2 Deep Learning

Deep multi-layer neural networks have many levels of nonlinearity, which allows them to potentially represent nonlinear and highly varying functions. However, until recently it was not clear how to train such deep networks, since gradient-based optimization starting from random initialization appears to often get stuck in poor solutions. Recently, Hinton *et al.* [7] proposed a deep learning architecture called Deep Belief Network (DBN). DBN is a generative graphical model consisting of a layer of visible units and multiple layers of hidden units, where each layer encodes correlations of the units in the layer below. The training of DBN consists of three phase: pre-training, unrolling, and fine-tuning. DBN first trains a sequence of Restricted Boltzmann Machine (RBM), and then unrolls the sequence of RBMs to form a deep auto-encoder neural network. Finally, in the fine-tuning phase, the deep auto-encoder neural network is trained using back propagation algorithm to optimize some task related objectives. The greedy layer-wise pre-training serves as a better initialization than random initialization for supervised training of the whole network.

While DBN has been successful in controlled domains, scaling them to realistic-sized (e.g. 200×200 pixels) images remains challenging. One way to address this issue is to subsample the large image to small size, such as [20]. However, sub-sampling may loss much useful information. Another way is the convolutional learning approach, which learns feature detectors that are shared among all locations in an image. They assume that features, which capture useful information in one part of an image, can pick up the same information elsewhere. Unlike categorizing generic object images, face verification focuses on a much more restricted subset of images (i.e., faces), requiring a fine granularity of discrimination solely between images within this restricted class. Therefore, in contrast to the convolutional neural network, we divide the image into a number of overlapping blocks, and use a separate set of weights for each block (see details in Sect. 3.3). Thus, we are able to train the models directly on the large size images. Huang *et al.* [21] take a similar deep learning approach for face verification. They develop local convolutional Restricted Boltzmann Machines (RBMs), an extension of convolutional RBMs, to learn high-level representations from low-level features, such as pixel intensity or Local Binary Pattern. Then they apply a metric-learning approach to learn an appropriate metric for the high-level representations. While in our approach, we combine the metric learning with the basic ISA network to learn the optimal connection weights, which can get more suitable representations for face verification than [21]. And it is evaluated in the experimental part.

3. Proposed Framework: NonLinear Metric Learning with Deep Independent Subspace Analysis Network

Because of the powerful approximate ability of the deep learning architecture to learn functions or distributions, and the virtues brought by the deep architecture, deep learning methods theoretically exhibit powerful learning ability to discover the nature of the dataset. Deep learning architecture has been successfully employed to enhance the learning ability of existing algorithms [22].

In this work, we also employ the deep learning architecture, and use the ISA network as the basic network. In the following subsections, we will first introduce the ISA network, then describe our discriminative training algorithm—NonLinear Metric Learning with ISA networks (NLML_ISA), and finally we stack the pre-trained ISA networks for deep metric learning.

3.1 Independent Subspace Analysis

Given the unlabeled data $\{x^{(t)}\}_{t=1}^{T}$, regular Independent Component Analysis (ICA) [23] is traditionally defined as the following optimization problem:

$$\min_W \sum_{t=1}^{T} \sum_{j=1}^{k} g(W_j x^{(t)}) \tag{4}$$

$$s.t.\ \ WW^T = I \tag{5}$$

where g(x) is a nonlinear convex function, e.g., smooth

L_1 penalty: $g(x) = \log(\cosh(x))$. W is the weight matrix $W \in R^{k\times n}$, k is number of components, and W_j is one row in W. The orthonormality constraint $WW^T = I$ is used to prevent the bases in W from becoming degenerate. According to [5], ISA is a variant of ICA, and it can be described as a two-layered network (as shown in Fig. 1 (b)). The active functions of the first and second layer are square and square-root respectively. The connection weight W of the first layer is learned, and the weight V of the second layer is fixed, which represents the subspace structure of the neurons in the first layer. Specifically, each of the second layer hidden units pools over a small neighborhood of adjacent units in the first layer. The first and second layer units are called simple and pooling units respectively.

More precisely, for an input pattern x, the output of each second layer unit is:

$$\mathrm{f_i}(x, \mathrm{W}, \mathrm{V}) = \sqrt{\sum_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{V_{ij}} \Big(\sum_{\mathrm{p}=1}^{\mathrm{n}} \mathrm{W_{jp}} x_{\mathrm{p}} \Big)^2}. \tag{6}$$

ISA learns the network parameters through finding sparse feature representations in the second layer, by solving:

$$\min_W \sum_{t=1}^{T} \sum_{i=1}^{m} f_i(x^{(t)}, W, V) \tag{7}$$

$$s.t.\ \ WW^T = I \tag{8}$$

where $\{x^{(\mathrm{t})}\}_{\mathrm{t}=1}^{\mathrm{T}}$ are the input samples. $W \in R^{k\times n}$, $\mathrm{V} \in R^{m\times k}$ are the connection weights of the first and second layer. n, k, m are the input dimension, number of simple units and pooling units respectively. One property of the ISA pooling units is that they are invariant and thus suitable for recognition task.

3.2 Nonlinear Metric Learning with ISA

The original ISA algorithm is unsupervised, so the learned features might not be suitable for some task specific objectives. We regard the ISA network as an explicit nonlinear transformation function $\mathrm{f}(x, \mathrm{W}, \mathrm{V})$: $\mathrm{R^n} \rightarrow \mathrm{R^m}$, and use the side information constraints to get the optimal parameters of the ISA network.

Similar to [2], we assume a logistic regression model to estimate the probability that two data points $x^{(i)}$ and $x^{(j)}$ share the same class or be semantically dissimilar, i.e.,

$$\Pr(l_{i,j}|\mathrm{x^{(i)}}, \mathrm{x^{(j)}}) = 1/(1 + \exp(l_{i,j}(\mathrm{d}(\hat{\mathrm{x}}^{(\mathrm{i})}, \hat{\mathrm{x}}^{(\mathrm{j})}) - \mu))) \tag{9}$$

where $l_{i,j} = 1$ if $(\mathrm{x^{(i)}}, \mathrm{x^{(j)}}) \in \mathrm{S}$, and $l_{i,j} = -1$ if $(\mathrm{x^{(i)}}, \mathrm{x^{(j)}}) \in \mathrm{D}$; $\hat{\mathrm{x}}^{(i)} = \mathrm{f}(\mathrm{x^{(i)}}, \mathrm{W}, \mathrm{V})$, $\hat{\mathrm{x}}^{(j)} = \mathrm{f}(\mathrm{x^{(j)}}, \mathrm{W}, \mathrm{V})$. The parameter μ is a threshold. Two data points $x^{(i)}$ and $x^{(j)}$ will have the same class label only when their distance $\mathrm{d}(\hat{x}^{(i)}, \hat{x}^{(j)})$ is less than the threshold μ. We use two simple distance measure: the Euclidean distance and Chi_Square distance to compute the distance between two feature vectors. Then the overall log likelihood for all the equivalent constraints S and the inequivalent constraints D can be written as:

$$\begin{aligned} \mathrm{L_g}(\mathrm{W}, \mu) &= \log(\Pr(\mathrm{S})) + \log(\Pr(\mathrm{D})) \\ &= -\sum_{(\mathrm{x^{(i)}}, \mathrm{x^{(j)}}) \in \mathrm{S}} \log(1 + \exp(\mathrm{d}(\hat{\mathrm{x}}^{(\mathrm{i})}, \hat{\mathrm{x}}^{(\mathrm{j})}) - \mu)) \\ &\quad - \sum_{(\mathrm{x^{(i)}}, \mathrm{x^{(j)}}) \in \mathrm{D}} \log(1 + \exp(\mu - \mathrm{d}(\hat{\mathrm{x}}^{(\mathrm{i})}, \hat{\mathrm{x}}^{(\mathrm{j})}))) \end{aligned} \tag{10}$$

Using the maximum likelihood estimation, we will cast the problem of distance metric learning into the following optimization problem:

$$\min_{\mathrm{W},\mu} E = -\mathrm{L_g}(\mathrm{W}, \mu) + \lambda \sum_{\mathrm{t}=1}^{\mathrm{T}} \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{f_i}(\mathrm{x^{(t)}}, \mathrm{W}, \mathrm{V}) \tag{11}$$

$$s.t.\ \ \mathrm{WW^T} = \mathrm{I} \tag{12}$$

The first term $L_g(W, \mu)$ is the log likelihood of side information constraints, which encourages the margin between positive and negative samples to be large. The second term is the mapping function of ISA, which encourages the sparsity of the transformed features. The hard orthonormality constraints ($WW^T = I$) is used to prevent degenerated solution of W. The standard optimization procedure, such as projected gradient descent, can be used to solve the above problem, and W is orthonormalized at each iteration by solving $W := (WW^T)^{-0.5}W$. This symmetric orthonormalization procedure requires Eigen decomposition, which is very challenging and time consuming, especially for the high dimensional data. However, the side information constraints can also prevent W from becoming degenerate. So in order to reduce the computational time, we ignore the hard orthonormality constraints.

We adopt gradient descend scheme with line search to solve the objective function optimization. The key issue is to compute the gradient of E with respect to W and μ. We write $z_i^{(l)}$, $a_i^{(l)}$ to denote the total weighted sum of inputs and the activation of unit i in layer l respectively. And we denote $f^{(1)}(x) = x^2$, and $f^{(2)}(x) = \sqrt{x}$ as the active function of the first and second layer of ISA network. Specifically, the computation is given by

$$z_i^{(1)}(x) = W_i x, \qquad a_i^{(1)}(x) = f^{(1)}(z_i^{(1)}(x)), \tag{13}$$

$$z_i^{(2)}(x) = V_i a^{(1)}(x), \qquad a_i^{(2)}(x) = f^{(2)}(z_i^{(2)}(x)), \tag{14}$$

where W_i and V_i is the i-th row vector of W and V. In the following, we will show the computation of $\partial E/\partial W_{jq}$ and $\partial E/\partial \mu$. (W_{jq} is the element of W at the j-th row and q-th column). Due to space limitation, we just show the gradient computation for Euclidean distance. The gradient computation for Chi_Square distance is similar.

The Euclidean distance between two vectors x and y in the transformed space is denoted as:

$$\mathrm{d}(\hat{\mathrm{x}}, \hat{\mathrm{y}}) = \sum_{\mathrm{i}=1}^{\mathrm{m}} (\hat{\mathrm{x}}_{\mathrm{i}} - \hat{\mathrm{y}}_{\mathrm{i}})^2 = \sum_{\mathrm{i}=1}^{\mathrm{m}} (a_i^{(2)}(x) - a_i^{(2)}(y))^2 \tag{15}$$

Then the gradient $\partial E/\partial W_{jq}$ and $\partial E/\partial \mu$ can be computed as:

$$\begin{aligned} \frac{\partial E}{\partial W_{jq}} &= -\sum_{(\mathrm{x},\mathrm{y}) \in \mathrm{S}} (\Pr(\mathrm{x}, \mathrm{y}) - 1)(\delta_{\mathrm{j}}^{(1)}(\mathrm{x})\mathrm{x_q} - \delta_{\mathrm{j}}^{(1)}(\mathrm{y})\mathrm{y_q}) \\ &\quad - \sum_{(\mathrm{x},\mathrm{y}) \in \mathrm{D}} (\Pr(\mathrm{x}, \mathrm{y}) - 0)(\delta_{\mathrm{j}}^{(1)}(\mathrm{x})\mathrm{x_q} - \delta_{\mathrm{j}}^{(1)}(\mathrm{y})\mathrm{y_q}) \end{aligned}$$

$$+\lambda\sum_{t=1}^{T}\sum_{i=1}^{m}1/a_i^{(2)}(x^{(t)})V_{ij}a_j^{(1)}(x^{(t)})x_q^{(t)} \quad (16)$$

$$\frac{\partial E}{\partial \mu}=-\left(\sum_{(x,y)\in S}(1-Pr(x,y))+\sum_{(x,y)\in D}(0-Pr(x,y))\right) \quad (17)$$

$$\text{where}\ \ \delta_i^2(x)=2(a_i^{(2)}(x)-a_i^{(2)}(y))\,(z_i^{(2)}(x))^{-0.5}, \quad (18)$$

$$\delta_i^2(y)=2(a_i^{(2)}(x)-a_i^{(2)}(y))\,(z_i^{(2)}(y))^{-0.5}, \quad (19)$$

$$\delta_j^1(x)=\sum_{i=1}^{m}\delta_i^2(x)V_{ij}z_j^{(1)}(x), \quad (20)$$

$$\delta_j^1(y)=\sum_{i=1}^{m}\delta_i^2(y)V_{ij}z_j^{(1)}(y). \quad (21)$$

After obtaining the gradient, the parameter W and μ can be updated by

$$W_{t+1}(\alpha_t)=W_t-\alpha_t\,\partial E/\partial W_t, \quad (22)$$

$$\mu_{t+1}=\mu_t-\alpha_t\,\partial E/\partial \mu_t. \quad (23)$$

The dynamic parameter α_t is an appropriate step size to enable effective gradient descent at step t. It can be selected from β^z ($0<\beta<1$, $z=0, 1, 2,\ldots$), such that the Wolfe condition prescribed below holds:

$$E(W_{t+1}(\beta^z))-E(W_t) \le \eta\sum_{l=1}^{L}\mathrm{tr}\left(\left(\frac{\partial E}{\partial W_t}\right)^T(W_{t+1}(\beta^z)-W_t)\right) \quad (24)$$

α_t is simply chosen as β^z where z is the smallest nonnegative integer satisfying the Wolfe condition. $0<\eta<1$ is a constant. Through making use of the gradient descent algorithm, we can achieve the optimal connection weight W of the local ISA networks under the side information constraints.

3.3 Stacked Local ISA for Deep Metric Learning

Traditionally, the convolutional neural network architecture is designed to scale up the algorithm for high resolution images (e.g. 150×150 pixel images in the LFW dataset). The key idea is that they first train the local filters on small input patches, and then take these learned filters to convolve with the large input images. It is based on the assumption that the distribution over features is stationary in an image with respect to position. However, for images belonging to a specific object class, such as faces, this assumption is no longer reasonable.

One strategy for removing this stationary assumption is to use a different set of filters for each region. Therefore, in our experiment, as shown in Fig. 1 (a), we divide the image into a number of overlapping blocks, and connect each ISA network to only one block, which we call local or block-wise ISA network. We regard all the local ISA networks as the FIRST_ISA network. Then we combine the responses of the spatial neighbored local ISA networks in the FIRST_ISA network, and treat them as inputs of the next layer of ISA network, which is regarded as the SECOND_ISA network. This procedure continues for the next layers. At the last layer, we combine all the response of previous layer of local ISA networks as input for one ISA network, and finally the output is the transformed feature vector of the original input image features. Figure 1 (a) shows the architecture of stacking three layers of local ISA networks.

The whole model can be regarded as a stacked ISA network. Similar to other algorithms proposed in the deep learning literature [7], [21], our stacked ISA model is trained greedy layer-wise in the pre-training phase, but we use the discriminative pre-training algorithm (NLML_ISA). In the fine-tuning phase, the objective function is similar to Eq. (11), but the mapping function is a stacked ISA network. We also adopt the gradient descent method for objective optimization, and the gradient computing steps are similar to those in Sect. 3.2.

4. Experiments

In this section, we will evaluate the effectiveness of the proposed method, and compare against state-of-the-art methods on the Labeled Faces in the Wild (LFW) [8]. We implement the method in MATLAB, and the source code is available upon request.

4.1 The LFW Dataset and Experiment Settings

The LFW was recently introduced as a benchmark dataset for face verification in the unconstraint environments. It is very challenging and difficult due to large variations in pose, age, expression, race and illumination. Figure 2 shows some example image pairs. The database contains 13,233 target images of 5749 persons. In addition, it is divided into ten independent folds that can be used for cross validation, where the subject identities are mutually exclusive. Each fold contains between 527 to 609 different people, and between 1016 to 1783 faces. This database is aimed at studying the problem of face verification. There are two evaluation settings provided by the authors of the database: the restricted and unrestricted setting. Under the restricted setting, only binary information is given for each pair of images, as we only know whether a pair of images belongs to the same

Fig. 2 Some example images of the LFW database. Each pair is from the same person but with variations in: First row: pose and expression; Second row: lighting; Third row: expression; Last row: occlusion

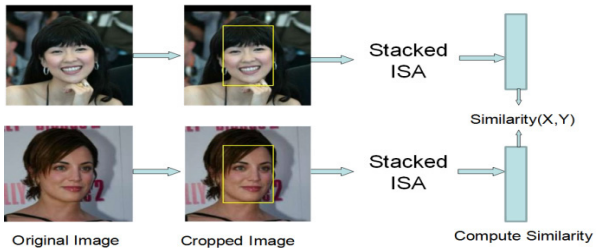

Fig. 3 The pipeline of our face verification system.

person or not, while the identity information of each face is unknown. Under the unrestricted setting, the identity information of each image is available. Therefore, it allows us to generate more image pairs for training.

In this experiment, we do not use any outside training data, and the performance is measured by ten-fold cross validation. We use the aligned version of the database, lfw_a [24]. And the face images are cropped to 150×80 pixels just by simply cutting out the center of the image. To combine the complementariness of different types of low-level features, we carry out the experiments on four descriptors: SIFT [25], Local Binary Pattern (LBP) [26], and two LBP's variations: Hierarchical Local Binary Pattern (HLBP) [27], and Patterns of Oriented Magnitudes (POEM) [28]. The architecture of the stacked ISA networks used in our experiment is shown in Fig. 1 (a). For LBP and its variations, the facial image is divided into half-size overlapping 36 (6×6) blocks, and then each block is further partitioned into 6 (3×2) patches. Histogram features are extracted in each patch. Then we concatenate the patch descriptor in one block to form the block-wise descriptors, which are the input for the FIRST_ISA network. And the spatial neighborhood in the SECOND_ISA network is defined as 2×2 neighbors. For SIFT, we use the Nine-Points SIFT features provided by [2], which are computed at the fixed facial points (e.g. corners of eyes, nose and mouth). For each facial point, we use multiscale SIFT descriptors to describe the patches. Setting scale $\sigma = 1$ to represent a 16×16 patch in the 250×250 face images, we extract SIFT features at multiple scale for $\sigma \in \{1, 2, 3\}$. We regard the concatenated 384D SIFT-based descriptor at each facial point as one block, and in the SECOND_ISA network, the spatial neighbors are the blocks in one component (e.g. eye, nose, mouth). For each local ISA network, the number of simple units and pooling units are set as 500 and 250 respectively.

The verification pipeline of our algorithm is illustrated in Fig. 3. For two given facial images, we first crop the original image into an appropriate size and extract some low-level features for each image; then these features are given as input to the proposed stacked ISA networks, and the output is the feature vector in the transformed subspace. We then compute the similarity of two feature vectors using Eq. (9). If the similarity score is larger than the predefined threshold (*e.g.* 0.5), the two facial images are classified as the same person, otherwise they depict different persons.

Table 1 Performance of our method in different phases for four descriptors.

Descriptor	Pre-training			Fine-tuning
	FIRST_ISA	SECOND_ISA	THIRD_ISA	
SIFT	79.12±0.61	80.23±0.52	82.53±0.40	**84.31±0.42**
LBP	78.00±0.52	79.85±0.43	82.10±0.50	83.21±0.32
HLBP	77.17±0.48	79.50±0.36	81.33±0.35	82.80±0.41
POEM	76.10±0.35	78.05±0.20	79.24±0.45	80.40±0.50

Table 2 Performance comparison of our method and other metric learning methods with SIFT descriptor under restricted setting.

Method	Accuracy
ITML[2]	76.18±1.25
LDML[2]	76.60±0.70
DML-eig[29]	80.55±1.71
PCCA[30]	82.20±0.40
KPCCA[30]	83.8-±0.40
NLML_DISA	**84.31±0.42**

Table 3 Performance comparison of our method and other state-of-art methods under restricted setting.

Method	Accuracy
LDML+Combined, funneled[2]	79.27±0.60
CSML+SVM, Combined[31]	88.00±0.37
High-Throughput Brain-inspired Feature[32]	88.13±0.58
DML-eig+Combined [29]	85.65±0.56
Local CRBM+Combined [21]	87.77±0.62
Our Method+Combined	**88.30±0.40**

4.2 Effectiveness of the Proposed NLML_DISA

In this experiment, we evaluate the effectiveness of our proposed NLML_DISA. Under restricted setting, each fold contains 300 matched pairs and 300 mismatched pairs. We use nine of the ten folds for training, and the left one for testing. Table 1 shows the performance of our proposed NLML_DISA with Euclidean distance measure in the pre-training and fine-tuning phase for different descriptors. From these results, we can see that the performance improves significantly as the number of layers increase, and the fine-tuning can further improve the performance near 2% over the THIRD_ISA. In addition, the average improvement of four descriptors is near 5% over the single layer ISA (FIRST_ISA). Thus, it proves the effectiveness of the deep learning architecture.

Table 2 gives the fair comparison between the proposed NLML_DISA and some metric learning methods with the same SIFT features under the image restricted setting. Our method obtains 84.31%, which is significantly better than the linear metric learning methods LDML [2] (76.60%), ITML [2] (76.18%), DML-eig [29] (80.55%), PCCA [30] (82.2%) and the kernel based metric learning method (KPCCA [30] (83.8%)). In order to compare with state-of-the-art results on LFW, we combine the similarity scores of four descriptors under two distance measures by a linear Support Vector Machine. The comparisons are presented in Table 3. Our method achieves a new state-of-the-art accuracy (88.30%). We believe that this is because the

Table 4 Performance of our method when varying the number of training pairs per fold.

		2000		3000		6000	
		Euclidean	Chi-Square	Euclidean	Chi-Square	Euclidean	Chi-Square
SIFT	Original	85.43±0.40	83.53±0.39	85.83±0.45	85.21±0.32	87.84±0.46	86.34±0.38
	Sqrt Root	86.10±0.38	85.32±0.42	87.13±0.40	86.58±0.35	**88.41±0.42**	88.10±0.56
POEM	Original	87.50±0.41	86.30±0.35	88.95±0.47	87.73±0.42	89.50±0.51	88.42±0.39
	Sqrt Root	**87.75±0.36**	**87.90±0.36**	88.94±0.37	**88.85±0.40**	89.90±0.35	89.21±0.36
LBP	Original	85.30±0.44	85.43±0.41	86.15±0.42	86.51±0.39	87.80±0.41	87.53±0.42
	Sqrt Root	87.43±0.36	86.50±0.32	88.53±0.40	87.62±0.40	**90.17±0.35**	89.40±0.34
HLBP	Original	85.75±0.45	84.55±0.47	85.94±0.49	86.73±0.49	87.54±0.47	87.80±0.51
	Sqrt Root	87.56±0.40	86.65±0.35	**89.65±0.43**	88.34±0.45	**90.51±0.41**	**89.58±0.35**
Feature Combined		90.10±0.32		92.40±0.35		**92.81±0.42**	

Table 5 Performance comparisons with other method on the LFW dataset, under unrestricted setting.

Method	Accuracy
LDML-MkNN,funneled[2]	87.50±0.40
LBP multishot aligned[35]	85.17±0.61
Combined multishot, aligned[35]	89.50±0.51
LBP PLDA, aligned[36]	87.33±0.55
Combined PLDA, funneled&aligned[36]	90.07±0.51
Face.com [37]	91.30±0.30
CMD, aligned[38]	91.70±1.10
SLBP,aligned[38]	90.00±1.33
CMD+SLBP, aligned[38]	92.58±1.36
NLML_DISA+Combined	**92.81±0.42**

proposed NLML_DISA method can suppress the variations of the facial appearance, and obtain a compact representation after transformation. Note that Pinto *et al.* [32] performs sophisticated large-scale feature search, and use multiple complimentary representations, which are combined by using kernel techniques. The local CRBM [21] approach combines two low-level features (pixels and local binary pattern), and achieves comparable results to ours. However, they use outside sources of data for training. Complete benchmark results can be found on the LFW website [33].

4.3 Performance Comparison under Unrestricted Setting

Under the unrestricted setting, the identity information of each training image is available. Therefore, we can generate as many pairs of matched and mismatched images as desired. This reduces the risk of over-fitting. Table 4 shows the performance of our method for four different descriptors, when using an increasing number of training pairs of each fold: 2000, 3000, and 6000. For each descriptor, we evaluate the original feature and the element-wise square-root feature. As shown by [34], taking square root of the original feature can improve the performance. Table 5 shows the comparisons of our method with current state-of-the-art methods under unrestricted setting.

From these results, we can observe that:
(1) As expected, our method benefits from an increasing number of training pairs. With 2000 training pairs per fold, the combined performance of our method is 90.10%. While with 6000 training pairs per fold, our method with only HLBP descriptor gives a mean accuracy of 90.51%; by combining multiple descriptors, the accuracy is further improved to 92.81%.
(2) As reported in [2], with 10000 training pairs per fold, LDML with the SIFT descriptor gives a mean accuracy of 83.2%. By combining LDML with the Marginalized k-nearest neighbor classifier [2], the performance can be improved to 87.50%, which is still slightly worse than our method (88.41%). With LBP descriptor, our method outperforms [35], [36] with a large margin.
(3) The mean accuracy of our method by ten-fold cross validation is 92.81%, which is remarkably well with other state-of-the-art results on LFW. In addition, the standard error (0.42%) is much lower than most of the other methods. Note that the result of [37] is obtained by a commercial system. It utilizes a proprietary 3D face reconstruction engine to produce an accurate 3D model from a single face image, while our method does not apply any preprocessing methods, and uses the original images. Huang *et al.* [38] proposes an ensemble metric learning approach, which first selects effective feature groups, and then further exploits correlations between selected feature groups. In addition, they use all the possible matched and mismatched pairs, while our method uses 6000 pairs per fold.

4.4 Computational Complexity Analysis

Finally, we analyze the computational complexity. Assume there are N pairs of facial images for training, and each image is divided into M blocks. The dimension of the low-level features in one block is d. The number of input units, simple units and pooling units in one ISA network is d_1, d_2, d_3 respectively. The spatial neighborhood is P, and the number of layers of the whole network is L. In the pretraining phase, we train the network layer by layer. The computational complexity of feed-forward computing will scale in about $O((\hat{d}_1 d_2 + d_2 + d_3)\hat{M}N)$, and the back-propagation will scale in about $O((d_3 d_2 \hat{d}_1)\hat{M}N)$, where $\hat{d}_1 = d$, $\hat{M} = M$ for the FIRST_ISA network, $\hat{d}_1 = Pd_3$, $\hat{M} = M/P^{l-1}$ for the l-th network ($l = 2, \ldots, L-1$), and $\hat{d}_1 = d_3 M/P^{L-2}$, $\hat{M} = 1$ for the last layer of network. So the computation complexity of training one layer of ISA network will scale in about $O((\hat{d}_1 d_2 + d_2 + d_3 + d_3 d_2 \hat{d}_1)\hat{M}NT_l)$, where T_l is the number of iterations for the l-th layer network. In the finetuning phase, the computational complexity will scale in about $O((M(d_1 d_2 + d_2 + d_3) + P^L M(d_2 d_3 + d_2 + d_3))T)$, where T is the number of iterations.

Running time: In our experiment, without feature

extraction, it takes about 6 hours to train the FIRST_ISA network on 108000 pairs of samples, 4 hours to train the SECOND_ISA network, 1 hour to train the THIRD_ISA network, and 1 hour to finetuning the whole network on a 3.0GHz, 8GB RAM PC. In addition, it takes about 4 minutes to test on 12000 pairs of samples.

5. Conclusions

In this paper, we proposed a nonlinear metric learning method by using Deep ISA network (NLML-DISA). Unlike traditional linear or kernel based metric learning methods, NLML-DISA learns an explicit nonlinear transformation and can handle high dimensional input spaces without prior dimension reduction. More specifically, for the high-resolution facial image, we use ISA as the basic network to connect to one block of the image and stack the block-wise ISA networks in a deep architecture. We combine the side-information constraints with ISA to get the optimal network connections. With the stacked ISA networks, every instance can be transformed nonlinearly to a compact vector for efficient verification. We evaluate the proposed method on LFW benchmark, and achieve better results than the state-of-the-art methods. Although initially the proposed NLML-DISA is designed for face verification, it has a wide range of applications, which we plan to explore in future works.

References

[1] A.K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," IEEE Trans Circuits Syst. Video Technol., vol.14, no.1, pp.4–20, 2004.
[2] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," Int. Conf. on Computer Vision, pp.498–505, 2009.
[3] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Technical report, Department of Computer Science and Engineering, Michigan State University, 2007.
[4] M.R. Min, D.A. Stanley, Z. Yuan, A. Bonner, and Z.L. Zhang, "Large-margin KNN classification using a deep encoder network," arXiv preprint arXiv:0906.1814, 2009.
[5] Q.V. Le, W. Zou, S.Y. Yeung, and A.Y. Ng, "Learning hierarchical spatial temporal features for action recognition with independent subspace analysis," Int. Conf. on Computer Vision and Pattern Recognition, pp.3361–3368, 2011.
[6] A. Hyvarinen, J. Hurri, and P. Hoyer, Natural image Statistics, Springer, 2009.
[7] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol.313, no.5786, pp.504–507, 2006.
[8] G.B. Huang, M. Ramesh, T. Berg, and E.L. Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environment," Technical Report, 09-49, University of Massachusetts, 2007.
[9] X.Y. Cai, C.H. Wang, B.H. Xiao, X. Chen, and J. Zhou, "Deep nonlinear metric learning with independent subspace analysis for face verification," Proc. ACM Conf. on Multi-Media, 2012.
[10] Y. Ijiri, S.H. Lao, and H. Murase, "Human re-identification by non-linear distance metric learning," IEICE Techical Report, PRMU2011-25, May 2011.
[11] J.B. Tenenbaum, V. Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," Science, vol.290, no.5500, pp.2319–2323, 2000.
[12] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science, vol.290, no.5500, pp.2323–2326, 2000.
[13] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," Proc. Neural Information Processing System, pp.505–512, 2003.
[14] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighborhood components analysis," Proc. Neural Information Processing System, pp.513–520, 2004.
[15] A. Globerson and S. Roweis, "Metric learning by collapsing classes," Proc. Neural Information Processing System, pp.451–458, 2006.
[16] K.Q. Weinberger and L.K. Saul, "Distance metric learning for large margin nearest neighbor classification," J. Machine Learning Research, vol.10, pp.207–244, 2009.
[17] L. Torresani and K.C. Lee, "Large margin component analysis," Proc. Neural Information Processing System, pp.1385–1392, 2007.
[18] J.V. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon, "Information-theoretic metric learning," Int. Conf. on Machine Learning, pp.209–216, 2007.
[19] J.T. Kwok and I.W. Tsang, "Learning with idealized kernels," Int. Conf. on Machine Learning, pp.400–407, 2003.
[20] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On deep generative models with applications to recognition," Int. Conf. on Computer Vision and Pattern Recognition, pp.2857–2864, 2011.
[21] G.B. Huang, H. Lee, and E.L. Miller, "Learning hierarchical representations for face verification with convolutionb deep belief networks," Conf. Computer Vision and Pattern Recognition, pp.2518–2525, 2012.
[22] W.K. Wong and M. Sun, "Deep learning regularized fisher mappings," IEEE Trans. Neural Netw., vol.22, no.10, pp.1668–1675, 2011.
[23] A. Hyvarinen, J. Karhunen, and E. Oja, Independent Component Analysis, Wiley Interscience, 2001.
[24] L. Wolf, T. Hassner, and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.10, pp.1978–1990, 2011.
[25] D.G. Lowe, "Object recognition from local scale-invariant features," Int. Conf. on Computer Vis., pp.1150–1157, 1999.
[26] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," European Conf. On Computer Vision, pp.469–481, 2004.
[27] Z. Guo, L. Zhang, D. Zhang, and X. Mou, "Hierarchical multiscale LBP for face and palmprint recognition," Int. Conf. Image Process., pp.4521–4524, 2010.
[28] N. Vu and A. Caplier, "Enhanced pattern of oriented edge magnitudes for face recognition and image matching," IEEE Trans. Image Process., vol.21, no.3, pp.1352–1365, 2012.
[29] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," J. Machine Learning Research (Special Topics on Kernel and Metric Learning), vol.13, pp.1–26, 2012.
[30] A. Mignon and F. Jurie, "PCCA: A new approach for distance learning from sparse pairwise constraints," Int. Conf. on Computer Vision and Pattern Recognition, pp.2666–2672, 2012.
[31] H. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," Asian Conf. on Computer Vision, pp.709–720, 2010.
[32] N. Pinto and D. Cox, "Beyond simple features: A large-scale feature search approach to unconstrained face recognition," Int. Conf. Automatic Face and Gesture Recognition, pp.8–15, 2011.
[33] http://vis-www.cs.umass.edu/lfw/index.html
[34] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," IEEE Trans. Pattern Anal. Mach. Intell., vol.34, no.3, pp.480–492, 2011.
[35] Y. Taigman, L. Wolf, and T. Hassner, "Multiple one-shots for utilizing class label information," British Machine Vision Conference,

pp.1–12, 2009.
[36] P. Li, Y. Fu, U. Mohammed, J.H. Elder, and S.J.D. Prince, "Probabilistic models for inference about identity," IEEE Trans Pattern Anal. Mach. Intelligence, vol.34, no.1, pp.144–157, 2012.
[37] Y. Taigman and L. Wolf, "Leveraging billions of faces to overcome performance barriers in unconstrained face recognition," ArXiv e-prints, ArXiv preprint arXiv: 1108-1122, 2011.
[38] C. Huang, S.H. Zhu, and K. Yu, "Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval," NEC Technical Report, TR115, 2011.

Xinyuan Cai received the B.S degree in the department of computer science and technology from Nanjing University, Nanjing, China. in 2008. He is currently pursuing the PH.D. degree in the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include face recognition, image processing, machine learning, and computer vision.

Chunheng Wang received the B.Eng and M.Eng degree from the Dalian University of Technology, and the Ph.D degree from the Institute of Automation, Chinese Academy of Sciences. He is currently a professor of State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Science. He is a member of IEEE and INCOSE. His research interest includes pattern recognition and intelligent system, Image processing, character recognition, and artificial intelligence.

Baihua Xiao received the B.Eng degree from the automatic control department of the Northwestern Polytechnic University, Shanxi, China. And then he received the PhD degree from the Institute of Automation, Chinese Academy of Sciences. He is currently a professor of State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Science. He is a member of IEEE and INCOSE. His research interest includes pattern recognition and intelligent system, computer vision, information retrieval.

Yunxue Shao received the B.S degree in the department of computer science and technology from HoHai University, Nanjing, China. in 2008. He is currently pursuing the PH.D. degree in the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include character recognition, image processing, and machine learning. He is a student member of IEICE.