Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Face recognition using linear representation ensembles

Hanxi Li [a,b], Fumin Shen [c,*], Chunhua Shen [d], Yang Yang [c], Yongsheng Gao [b]

[a] School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, China
[b] School of Engineering, Griffth University, QLD 4111, Australia
[c] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China
[d] School of Computer Science, The University of Adelaide, SA 5005, Australia

ARTICLE INFO

Article history:
Received 30 July 2015
Received in revised form
23 November 2015
Accepted 11 December 2015

Keywords:
Face recognition
Linear representation
Ensemble learning

ABSTRACT

In the past decade, linear representation based face recognition has become a very popular research subject in computer vision. This method assumes that faces belonging to one individual reside in a low-dimensional linear subspace. In real-world applications, however, face images usually are of degraded quality due to expression variations, disguises, and partial occlusions. These problems undermine the validity of the subspace assumption and thus the recognition performance deteriorates significantly. In this work, we propose a simple yet effective framework to address the problem. Observing that the linear subspace assumption is more reliable on certain face patches rather than on the holistic face, Probabilistic Patch Representations (PPRs) are randomly generated, according to the Bayesian theory. We then train an ensemble model over the patch-representations by minimizing the empirical risk w.r.t. the "leave-one-out margins", which we term Linear Representation Ensemble (LRE). In the test stage, to handle the non-facial or novel face patterns, we design a simple inference method to dynamically tune the ensemble weights according to the proposed Generic Face Confidence (GFC). Furthermore, to accommodate immense PPR sets, a boosting-like algorithm is also derived. In addition, we theoretically prove two desirable property of the proposed learning methods. We extensively evaluate the proposed methods on four public face dataset, i.e., Yale-B, AR, FRGC and LFW, and the results demonstrate the superiority of both our two methods over many other state-of-the art algorithms, in terms of both recognition accuracy and computational efficiency.

1. Introduction

Face Recognition is a long-standing problem in computer vision and pattern recognition. In the past decade, much effort has been devoted to the *Linear Representation* (LR) based algorithms such as Nearest Feature Line (NFL) [1], Nearest Feature Subspace (NFS) [2], Sparse Representation Classification (SRC) [3], Linear Representation Classification (LRC) [4] and the mostly recently proposed variations of SRC [5,6] and the variations of LRC [7–11]. Compared with traditional face recognition approaches, higher accuracies have been reported. The underlying assumption for these LR classifiers is that the faces of one individual reside in a low-dimensional linear manifold. This assumption, however, is only valid when the cropped faces are considered as rigid Lambertian surfaces and without any occlusion [12,13]. In practice, the linear-subspace model is sometimes too rudimentary to cope with expressions, disguises and random occlusions, which usually occur in local regions. For example, expressions influence the mouth and eyes more greatly than the nose; scarves typically have the impact on lower-half faces. The problematic face parts violate the assumptions required by linear representations and thus deteriorates face recognition accuracy. On the other hand, there are almost always some face parts that are less problematic, *i.e.*, more reliable. But, how can we effectively evaluate the reliability of one face part? Given the reliabilities of all the parts, how do we make the final decision in order to achieve better recognition accuracy?

In the literature, some heuristic methods were introduced to address this problem. In particular, the modular approach is used for eliminating the adverse impact of continuous occlusions [3,4,14]. Significant improvement in accuracy was observed from the partition-and-vote [3] or the partition-and-compete [4] strategy. The drawbacks of these heuristics are also obvious. First, one must roughly know *a priori* the shape and location of the occlusion. Otherwise one still cannot obtain satisfactory performance. It is desirable to design more flexible "models" to handle occlusions

* Corresponding author.
E-mail addresses: lihanxi2001@gmail.com (H. Li), fumin.shen@gmail.com (F. Shen), chhshen@gmail.com (C. Shen), dlyyang@gmail.com (Y. Yang), yongsheng.gao@griffith.edu.au (Y. Gao).

http://dx.doi.org/10.1016/j.patcog.2015.12.011
0031-3203/

with arbitrary locations and shapes. Furthermore, the existing heuristics discard much useful information, such as the representation residuals in [3] or the classification results of the unselected blocks in [4]. Higher efficiency is expected when all the information is simultaneously analyzed. Third, there is great potential to increase the performance by employing a sophisticated fusion method, rather than the primitive rules used in [3] and [4]. Finally, most existing methods neglect the fact that LR methods can also be used to distinguish human faces from non-facial images, or partly-non-facial images. By harnessing this power, one could achieve higher robustness to occlusions and noises.

In this work, we propose a simple yet effective framework to learn and recognize faces. The novel framework generates, interprets and aggregates the partial representations in a Bayesian manner. First of all, LRs are performed on randomly generated face patches. Second, we view each patch representation as a probability vector, with each element corresponding to a certain individual. The interpretation is obtained by applying Bayes theorem on a basic distribution assumption, and thus is referred to as Probabilistic Patch Representation (PPR). We then learn a linear combination of the obtained PPRs to gain much higher classification ability. The combination coefficients, *i.e.*, the weights associated to different PPRs, are achieved via minimizing the exponential loss w.r.t sample margins [15]. Thus, most given face-related patterns are learned via assigning different "importances" to different patches. The learned model is termed *Linear Representation Ensemble* (LRE) and it is obtained via minimizing a predefined empirical risk. To cope with unseen patterns in the test stage, the *Generic Face Confidence* (GFC) is derived under the Bayesian framework by taking account of the non-facial category. This confidence indicates how a test patch is contaminated by unknown patterns. The learned LRE model is then adaptively updated according to GFCs, which leads to improved robustness.

Essentially, the PPRs are instance-based. Due to the requirement of "gallery" samples, one cannot simply apply those off-the-shelf ensemble learning methods such as AdaBoost to combine them. To accommodate the instance-based predictors and optimally exploit the given information, we propose the *leave-one-out margin* for replacing the conventional margin concept. The leave-one-out margin also makes the LRE-Learning procedure more resistant to the overfitting, as we will theoretically verify. One therefore can choose the model parameter merely depending on the training errors. This merit leads to a remarkable drop in the validation complexity. In addition, to tailor the proposed method to immense PPR sets, a boosting-like algorithm is designed to obtain the LRE in an iterative fashion. The boosted model, Boosted-LRE, could be learned very efficiently as we prove that the training procedure is unrelated to data weights. *The high-level idea of this work is that, we offer an effective framework for training a discriminative ensemble of instance-based classifiers*. Differing from other ensemble learning methods for face recognition [16–20], our method focuses on the ensemble of the spatially local regions of the face. This strategy has been extensively used for image classification [21–23].

The experiments demonstrate the high accuracy of proposed algorithms. In particular, LRE achieves the 99.9% for Yale-B dataset, 99.5% for AR dataset and around 60.0% for the LFW identification dataset [8,7], for the faces with extreme illumination changes, expressions/disguises and uncontrolled environment, respectively. Boosted-LRE also shows similar recognition capability. Equipped with the GFC, LRE outperforms other modular heuristics under all the tested circumstances. Moreover, the LRE model also shows the highest efficiency (less than 20 ms per face with Matlab and single CPU core) among all the compared LR methods.

In summary, the contributions of this work include the following:

1. An effective face recognition algorithm is proposed. Based on simple and fast linear representations, our algorithm achieves comparable or better recognition accuracies compared with the-state-of-the-art methods on various face datasets. Meanwhile, our method performs even faster than the ordinary Linear Representation Classification algorithm.
2. A novel inference strategy is introduced to adapt the learned LRE model to the current test image, which may contain non-facial parts or unseen facial behaviors. The inference strategy is also build upon linear representations and improves the algorithm robustness significantly.
3. To properly accommodate the instance-based weak learners (*i.e.*, the PPRs in this work) in the ensemble learning framework. A tailored learning method based on the "leave-one-out margin" is proposed and illustrates its high robustness to overfittings. For immense PPRs, we design a variation of the learning method and theoretically prove its high training speed.

The rest of this paper is organized as follows. In Section 2, we briefly overview the family of LR classifiers and the modular heuristics. PPR is proposed in the following section. The learning algorithm for obtaining LRE is derived in Section 3 where we also prove the validity of the training-determined model-selection. The derivations of the boosting-like algorithm, *a.k.a.*, Boosted-LRE, and its desirable feature in terms of fast training are given in Section 3.4. Section 4 introduces the inference method for updating the LRE model and the learning-inference-based strategy. Experimental results are shown in Section 5. We then conclude the paper.

2. Background

2.1. The family of linear representation classifiers

In a face recognition problem, one is usually given N vectorized face images $\mathbf{X} \in \mathbb{R}^{D\times N}$ belonging to K different individuals, where D is the feature dimension and N is the face number. Suppose that their labels are $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}, l_i \in \{1, 2, \ldots, K\} \forall i$, when a probe face $\mathbf{y} \in \mathbb{R}^D$ is provided, we need to identify it as one individual in the training set, *i.e.*, $\gamma_{\mathbf{y}} = H(\mathbf{y}) \in \{1, 2, \ldots, K\}$, where $H(\cdot)$ is the face recognizer that generates the predicted label γ. Without loss of generality, we assume that all the classes share the same sample number $M = N/K$. For the k-th face category, let $\mathbf{x}_i^k \in \mathbb{R}^D$ denote the i-th face image and $\mathbf{X}_k = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M] \in \mathbb{R}^{D\times M}$ indicates the image collection of the k-th class.[1]

Nearest Neighbor (NN) can be considered as the most primitive LR method. It uses only one training face, the nearest neighbor, to represent the test face. However, without a powerful feature extraction approach, NN usually performs unsatisfactorily. Therefore, more advanced methods like NFL [1], NFS [2], SRC [3] and LRC [4] are proposed. Most of their formulations ([1,2,4]) can be unified. For class $k \in \{1, 2, \ldots, K\}$, a typical LR classifiers firstly solve the following problem to get the representation coefficients $\boldsymbol{\beta}_k^*$

$$\min_{\boldsymbol{\beta}_k} \|\mathbf{y} - \tilde{\mathbf{X}}_k \boldsymbol{\beta}_k\|_2 \quad \forall k \in \{1, 2, \ldots, K\}, \tag{1}$$

where $\|\cdot\|_p$ stands for the ℓ_p norm and $\tilde{\mathbf{X}}_k$ is a subset of $\mathbf{X}_k$, selected under certain rules. The above least square problem has a

[1] For simplicity, we slightly abuse the notation here: the symbol of a matrix is also used to represent the set comprised of all the columns of this matrix.

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

closed-form solution given by

$$\boldsymbol{\beta}_k^* = (\tilde{\mathbf{X}}_k^\top \tilde{\mathbf{X}}_k)^{-1} \tilde{\mathbf{X}}_k^\top \mathbf{y}. \tag{2}$$

The identity of test face $\mathbf{y}$ is then retrieved as

$$\gamma_{\mathbf{y}} = \text{argmin}_{k \in \{1,\ldots,K\}} r_k, \tag{3}$$

where r_k is the reconstruction residual associated with class k, *i.e.*,

$$r_k = \|\mathbf{y} - \tilde{\mathbf{X}}_k \boldsymbol{\beta}_k^*\|_2. \tag{4}$$

Different rules for selecting $\tilde{\mathbf{X}}_k$ actually specify different members of the LR family. NN merely uses one nearest neighbor from $\mathbf{X}_k$ as the representation basis; NFL exhaustively searches two faces which form a nearest line to the test face; NFS conducts a similar search for the nearest subspace with a specific dimensionality. Finally, at the other end of the spectrum, LRC directly employs the whole $\mathbf{X}_k$ to represent $\mathbf{y}$. Note that although the solution of problem (1) is closed-form, most LR methods require a brute-force search to obtain $\tilde{\mathbf{X}}_k$. The only exception exists in LRC where $\tilde{\mathbf{X}}_k = \mathbf{X}_k$, thus LRC is usually faster than the other members.

The SRC algorithm, on the other hand, solves a second-order cone problem over the entire training set $\mathbf{X}$. The optimization problem writes:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \quad \text{s.t. } \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2 \le \varepsilon. \tag{5}$$

Then the representation coefficients for class k are calculated as:

$$\boldsymbol{\beta}_k^* = \delta_k(\boldsymbol{\beta}) \forall k \in \{1, 2, \ldots, K\}, \tag{6}$$

where function $\delta_k(\boldsymbol{\beta})$ sets all the coefficients of $\boldsymbol{\beta}$ to 0 except those corresponding to the k-th class [3]. By treating the occlusion as a "noisy" part, Wright et al. [3] also proposed a robust version of SRC, which conducts the optimization as follows:

$$\min_{\boldsymbol{u}} \|\boldsymbol{u}\|_1 \quad \text{s.t. } \|\mathbf{y} - [\mathbf{X}, \mathbf{I}]\boldsymbol{u}\|_2 \le \varepsilon, \tag{7}$$

where $\mathbf{I}$ is an identity matrix, $\boldsymbol{u} = [\boldsymbol{\beta}, \mathbf{e}]^\top$ and $\mathbf{e}$ is the representation coefficients corresponding to the non-facial part. SRC is slow [24] due to the second-order cone programming. Moreover, recent advances show that the sparsity is not so important for face recognition [25,9].

LR-based methods have achieved impressive performances because the underlying linear-subspace theory [12,13] keeps approximately valid no matter how the illumination changes. Unfortunately, for the face with extreme expressions, disguises or random contaminations, the theory does not hold anymore and poor recognition accuracies are usually observed.

2.2. Random face patches

The main motivation of this work is that, when the linear-subspace assumption is invalid for a holistic face, one can usually find some local facial regions that satisfy the assumption better. By localizing and weighting those regions, we can combine those local recognizers into a much better face recognition model. In this paper, we generate 500 small patches randomly distributing over the entire face image. Those patches are already sufficient to cover most of the reliable face parts. Different weights are assigned to these patches to indicate their contributions to a specific recognition task. We expect that a certain combination of these patches could deliver similar classification capacity to the direct use of all the reliable regions.

Fig. 1(a) shows an example of the weighted patches. 500 random face patches are generated with different shapes (here only rectangles). The higher its weight is assigned, the redder and wider a patch is shown. The weights are obtained by using the proposed LRE algorithm on AR dataset [26]. Note that most patches are purely blue which implies their weights are too small to influence the classification. These patches are negligible for the final learned classifier.

Compared with the deterministic blocks used in [3,4,7,8], random patches are more flexible (could be any shape as a combination) and more efficient (only a small number of patches are important), as we empirically proved later.

2.3. Probabilistic patch representation

Given that the linear-subspace assumption is more reliable on certain face patches, it is intuitive to firstly perform the LR method on each of them. In principle, we could employ either member of the LR family to perform the linear representation on the patches. According to the theoretical analysis [12,13], however, it seems no need to specifically select a certain subset from $\mathbf{X}_k$. We thus employ the whole $\mathbf{X}_k$ to form the representation basis, just as what [4] and [12] did. In particular, for class k and patch t, we denote the patch set as $\mathbf{X}_k^t = [\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_M^t] \in \mathbb{R}^{d \times M}$, with each column obtained via vectorizing a image patch. The representation coefficients $\boldsymbol{\beta}_{t,k}^*$, for the k-th class and t-th patch is then given by

$$\boldsymbol{\beta}_{t,k}^* = (\mathbf{X}_k^t \top \mathbf{X}_k^t)^{-1} \mathbf{X}_k^t \top \mathbf{y}. \tag{8}$$

Then the residual $r_{t,k}$ can be obtained as $r_{t,k} = \|\mathbf{y}_t - \mathbf{X}_k^t \boldsymbol{\beta}_{t,k}^*\|_2$, where $\mathbf{y}_t$ is the cropped test image according to the patch location. In this paper, all the patches are normalized so that their ℓ_2 norms are equal to 1. As a result, $r_{t,k} \in [0, 1], \forall t, k$.

Ordinary LR methods only focus on the smallest residual or the corresponding class label. Differing from that, we interpret every patch representation as a probability vector $\mathbf{b}_t$. The k-th element of $\mathbf{b}_t$, namely $b_{t,k}$, is the probability that current test patch $\mathbf{y}_t$ belongs to individual k *i.e.*

$$b_{t,k} = P\left(\gamma_{\mathbf{y}} = k | \mathbf{y}_t\right). \tag{9}$$

We obtain the above posteriors by applying the Bayesian theorem. First of all, it is common that all the classes share the same prior probability, *i.e.*, $P\left(\gamma_{\mathbf{y}} = k\right) = 1/K, \forall k$. The linear-subspace assumption states that, if one test face belongs to class k, the test patch $\mathbf{y}_t$ should distribute around the linear subspace spanned by $\mathbf{X}_k^t$. The probability of a remote $\mathbf{y}_t$ is smaller than the one close to the subspace. In this sense, when the category is known, we can assume the random variable $\mathbf{y}_t$ belongs to a distribution with the probability density function

$$P\left(\mathbf{y}_t | \gamma_{\mathbf{y}} = k\right) = C \cdot \exp(-r_{t,k}^2/\delta), \tag{10}$$

where δ is a assumed variance and the C is the normalization factor. This distribution, in essence, is a *singular normal distribution* as its covariance matrix is singular. This model can be considered as the original linear subspace model of faces plus the Gaussian noise that represents the uncertainty of the assumption.

According to the Bayes' rule, the posterior probability is then derived as

$$b_{t,k} = \frac{P\left(\mathbf{y}_t | \gamma_{\mathbf{y}} = k\right) \cdot P\left(\gamma_{\mathbf{y}} = k\right)}{\sum_{j=1}^K P\left(\mathbf{y}_t | \gamma_{\mathbf{y}} = j\right) \cdot P\left(\gamma_{\mathbf{y}} = j\right)} = \frac{C/K \cdot \exp(-r_{t,k}^2/\delta)}{\sum_{j=1}^K C/K \cdot \exp(-r_{t,j}^2/\delta)} = \frac{\exp(-r_{t,k}^2/\delta)}{\sum_{j=1}^K \exp(-r_{t,j}^2/\delta)}. \tag{11}$$

As an example, Fig. 2 shows the distribution of the posterior $b_{t,1}$ when there are only 2 orthogonal linear subspaces (1 and 2 and dimensionality $D=2$).

We finally aggregate all the posteriors into a vector $\mathbf{b}_t = [b_{t,1}, b_{t,2}, \ldots, b_{t,K}]^\top$. The probabilistic interpretation $\mathbf{b}_t$, termed Probabilistic Patch Representation (PPR), keeps most information related to the representation and thus could lead to a more

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

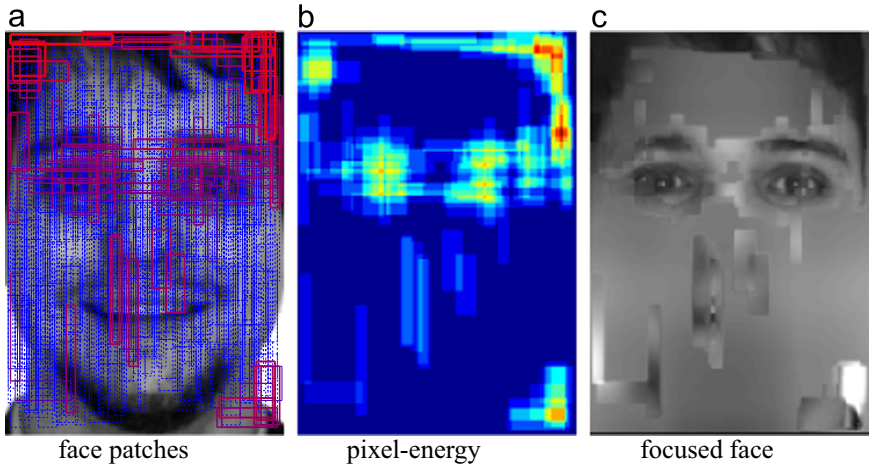

Fig. 1. Demonstration of random face patches. (a) 500 random face patches with different weights. (b) The corresponding pixel-energy map. (c) The simulated focusing behavior. The weights are obtained by using the proposed LRE-Learning algorithm on AR dataset [26]. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

accurate recognition result. In practice, it makes little sense to impose a constant δ for all the patches and faces. We thus perform normalization for every patch and set $\delta_t = 0.1 \cdot \min_k(r_{t,k}^2)$.

3. Learning the linear representation ensemble

3.1. Learning a PPR ensemble via empirical risk minimization

Besides the interpretation, the aggregation method is also crucial for the final classification. Some classical fusion rules, such as voting [3] or competition [4], are rudimentary and non-parameterized thus hard to optimize. In the machine learning community, classifier ensembles learned via an empirical risk minimization process are considered to be more powerful than the simple methods [27,28].

In this work, we linearly combine the PPRs to generate a prediction vector $\boldsymbol{\xi}(\mathbf{y}) = [\xi_1(\mathbf{y}), \xi_2(\mathbf{y}), \ldots, \xi_K(\mathbf{y})]^\top \in \mathbb{R}^K$,

$$\boldsymbol{\xi}(\mathbf{y}) = \sum_{t=1}^{T} \alpha_t \mathbf{b}_t(\mathbf{y}) = \mathbf{B}(\mathbf{y})\boldsymbol{\alpha}, \tag{12}$$

with $\xi_k(\mathbf{y})$ indicating the confidence that $\mathbf{y}$ belongs to the k-th class, T represents the number of test patches and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_T]^\top \succcurlyeq \mathbf{0}$. The identity of test face $\mathbf{y}$ is given by

$$\gamma_{\mathbf{y}} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmin}} \xi_k(\mathbf{y}) \tag{13}$$

This kind of linear model dominates the supervised learning literature [29,28] as it is flexible and easy to learn. The parameter vector $\boldsymbol{\alpha}$ is optimized via minimizing the following empirical risk:

$$\mathrm{ER} = \sum_{i}^{N} \mathrm{Loss}(z_i) + \lambda \cdot \mathrm{Reg}(\boldsymbol{\alpha}), \tag{14}$$

where $\mathrm{Loss}(\cdot)$ is a certain loss function (usually convex), $\mathrm{Reg}(\cdot)$ is the regularization term and λ is the trade-off parameter. The margin $z_i = \mathcal{Z}(l_i, \boldsymbol{\xi}(\mathbf{x}_i))$ reflects the confidence that $\boldsymbol{\xi}$ select the correct label for $\mathbf{x}_i$. Specifically, for binary classifications,

$$z_i = \xi_{l_i}(\mathbf{x}_i) - \xi_{l'}(\mathbf{x}_i), \quad l' \neq l_i. \tag{15}$$

For multiple-class problems, however, it is less intuitive to define z_i. We can define it as

$$z_i = \frac{1}{K-1} \sum_{j \neq l_i}^{K} \left(\xi_{l_i}(\mathbf{x}_i) - \xi_j(\mathbf{x}_i) \right), \tag{16}$$

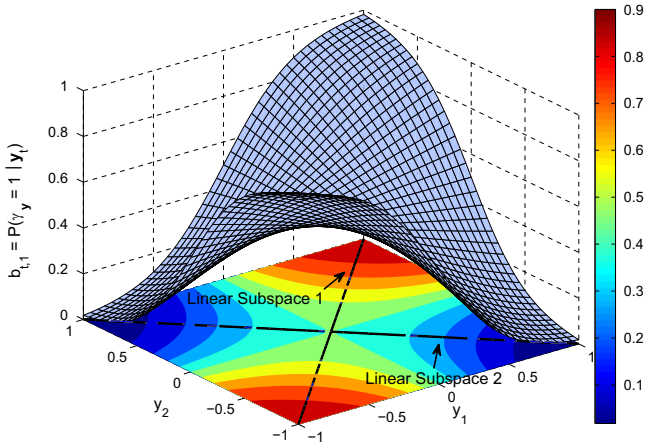

Fig. 2. Given two subjects, this figure demonstrates the posterior distribution $b_{t,1} = P(\gamma_{\mathbf{y}} = 1 | \mathbf{y}_t)$, i.e., the probability that $\mathbf{y}_t$ belongs to the first subject. The dimensionality of the feature space is 2. In this example, two linear subspaces, *i.e.*, the two black lines, are orthogonal to each other and represent two subjects, respectively.

which is the mean of all the "bi-class margins". Recall that $\sum_{j=1}^{K} b_{t,j}(\mathbf{x}_i) = 1$, we then arrive at a simpler definition of z_i,

$$z_i = \frac{K}{K-1} \sum_{t=1}^{T} \alpha_t \left(b_{t,l_i}(\mathbf{x}_i) - \frac{1}{K} \right). \tag{17}$$

By absorbing the constant $K/(K-1)$ into each α_t, we have

$$z_i = \sum_{t=1}^{T} \alpha_t \left(b_{t,l_i}(\mathbf{x}_i) - \frac{1}{K} \right). \tag{18}$$

The term $b_{t,l_i}(\mathbf{x}_i) - 1/K$ can be thought of as the confidence gap between using the t-th PPR and using a *random guess*. The larger the gap, the more powerful this PPR is. Consequently, z_i is the weighted sum of all the gaps, which measures the predicting capability of $\boldsymbol{\xi}(\mathbf{x}_i)$.

The selection for the loss function and the regularization function has been extensively studied in machine learning [30]. We choose the exponential loss $\mathrm{Loss}(z_i) = \exp(-z_i)$, motivated by its success in AdaBoost [28,31]. The ℓ_1 norm is adopted as our regularization method since it encourages the sparsity of $\boldsymbol{\alpha}$, which is desirable when we want an efficient ensemble. Finally, the

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

optimization problem is given by:

$$\min_{\boldsymbol{\alpha}} \quad \sum_{i}^{N} \exp\left(-\sum_{t=1}^{T} \alpha_t \left(b_{t,l_i}(\mathbf{x}_i) - \frac{1}{K}\right)\right)$$
$$\text{s.t.} \quad \boldsymbol{\alpha} \geqslant \mathbf{0}, \ \|\boldsymbol{\alpha}\|_1 \leq \lambda \tag{19}$$

Note that for easing the optimization, we convert the regularization term to a constraint. With an appropriate λ, this conversion does not change the optimization result [32]. The optimization problem is convex and can be solved by using off-the-shelf optimization tools such as Mosek [33] or CVX [34]. The learned model is termed *Linear Representation Ensemble* (LRE) as it guarantees the global optimality of $\boldsymbol{\alpha}$ from the perspective of empirical risk minimization. The learning algorithm for achieving the LRE is referred to as LRE-Learning.

3.2. Leave-one-out margin

It would be simple to calculate the margin z_i if the PPR were model-based, *i.e.*, $\mathbf{b}_t(\cdot)$ was a set of explicit functions. In fact, that is the situation for most ensemble learning approaches. Unfortunately, that is not the case in this paper where $\mathbf{b}_t(\cdot)$ is actually instance-based.

For a PPR, we always need a *gallery*, *i.e.*, the representation basis, to calculate $\mathbf{b}_t(\cdot)$. Ideally, the gallery should be the same for both training and test, otherwise the learned model is only optimal for the training gallery. Nonetheless, we cannot directly use the training set, which is the test gallery, as the training gallery. Any training sample $\mathbf{x}_i$ will be perfectly represented by the whole training set because $\mathbf{x}_i$ itself is in the basis. Consequently, all PPRs will generate identical outputs and the learned weights $\alpha_t, \forall t$ will also be the same. To further divide the training set into one basis and one validation set, of course, is a feasible solution. However, it might reduce the classification power of LRE as the larger basis usually implies higher accuracies.

To avoid this problem, we employ a leave-one-out scheme to utilize as many training instances as possible for representations. For every training sample $\mathbf{x}_i$, its gallery is $\mathbf{x}_i^C = \mathbf{X} \backslash \mathbf{x}_i$: the complement of $\mathbf{x}_i$ w.r.t the universe $\mathbf{X}$. The leave-$\mathbf{x}_i$-out PPRs, referred to as $\mathbf{b}_t^{\mathbf{x}_i^C}(\mathbf{x}_i)$ $(\forall t)$, are yielded based on the gallery $\mathbf{x}_i^C$. The leave-one-out margin z_i is then calculated as

$$z_i = \sum_{t=1}^{T} \alpha_t \left(b_{t,l_i}^{\mathbf{x}_i^C}(\mathbf{x}_i) - \frac{1}{K}\right). \tag{20}$$

So, the size of the training gallery is always $N-1$, we can approximately consider the learned $\boldsymbol{\alpha}^*$ as optimal for the test gallery $\mathbf{X}$ with the size of N.

After $\boldsymbol{\alpha}^*$ is obtained, we also calculate the leave-one-out predicting vector as

$$\boldsymbol{\xi}^{\mathbf{x}_i^C}(\mathbf{x}_i) = \mathbf{B}^{\mathbf{x}_i^C}(\mathbf{x}_i)\boldsymbol{\alpha}^*, \tag{21}$$

where $\mathbf{B}^{\mathbf{x}_i^C}(\mathbf{x}_i)$ is the collection of the leave-one-out PPRs. The training error of the LRE-Learning is given by

$$e_{trn} = \frac{1}{N}\sum_{i=1}^{N} [\![\operatorname{argmax}_k \xi_k^{\mathbf{x}_i^C}(\mathbf{x}_i) \neq l_i]\!], \tag{22}$$

where $[\![\cdot]\!]$ denote the boolean operator. This training error, as illustrated below, plays a crucial role in the model-selection procedure.

3.3. Training-determined model-selection

Another issue arising here is how to select a proper parameter λ for the LRE-Learning. Usually, a validation method such as the *n-fold cross-validation* is performed to select the optimal parameter among candidates. The validation method, however, is expensive in terms of computation, because one needs to repeat the extra "subset training" for n times and usually $n \geq 5$. From the instance-based perspective, a cross-validation is also unacceptable. In every "fold" of a n-fold cross-validation, we only use a part of training samples as the gallery. The setting contradicts the principle that one needs to keep the representation basis similar over all the stages.

Fortunately, the leave-one-out margin provides the LRE-Learning an advantage: The training error of the LRE-Learning serves as a good estimate for its leave-one-out error. *We can directly use the training error to select the model-parameter* λ. To understand this, let us firstly recall the definition of the leave-one-out error.

Definition 3.1 (*Leave-one-out error [15]*). Suppose that $\mathcal{X}^N$ denotes a training set space comprised of the training sets with N samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. Given an algorithm $\mathcal{A}: \bigcup_{N=1}^{\infty} \mathcal{X}^N \rightarrow \mathcal{F}$, where $\mathcal{F}$ is the functional space of classifiers. The leave-one-out error is defined by

$$e_{loo} \triangleq \frac{1}{N}\sum_{i=1}^{N} [\![F_{\mathbf{x}_i^C}^{\mathcal{A}}(\mathbf{x}_i) \neq l_i]\!], \tag{23}$$

where $F_{\mathbf{x}_i^C}^{\mathcal{A}} = \mathcal{A}(\mathcal{X}^N \backslash \mathbf{x}_i)$, *i.e.*, the classifier learned using $\mathcal{A}$ based on the set $\mathcal{X}^N \backslash \mathbf{x}_i$.

The leave-one-out error is known as an unbiased estimate for the generalization error [15,35]. Suppose that all the training faces are non-disguised, which is the common situation, then let us make the following basic assumption.

Assumption: One patch-location t on the human face could be affected by Q_t different expressions. Every expression leads to a distinct and convex Lambertian surface.

According to the theory in [12] and [13], different appearances of one patch surface, caused by illumination changes, span a linear subspace with a small dimensionality Φ. Given that M training patches from the patch-location t is collected in $\mathbf{X}_k^t$, its arbitrary subset $\mathbf{X}_P^t$ contains P $(P < \Phi \ll M)$ samples. With the assumption, we can verify the following lemma.

Observation (*The stability of PPRs*): If the training subset $\mathbf{X}_k^t$ contains at least $(\Phi Q_t + P)$ i.i.d.patch samples, set $\mathbf{X}_k^t$ and set $\mathbf{X}_k^t \backslash \mathbf{X}_P^t$ share the same representation basis.

The proof can be found in the supplementary. In contrast, other classifiers, such as decision trees or LDA classifiers, do not have this desirable stability. They always depend on the exact data, rather than the extracted space-basis.

Note that Φ is usually very small [12,13]. The value of Q_t is determined by the types of expressions that can affect patch t. It is also very limited if we only consider the common ones. That is to say, *with a reasonable number of training samples, the PPRs is stable* w.r.t *the data fluctuation.* Specifically, when $\mathbf{x}_i$ is left out $(P=1)$, all the PPRs' values on samples $\{\mathbf{x}_j \in \mathbf{X} | j \neq i\}$ won't change, *i.e.*

$$\mathbf{b}_t^{\mathbf{x}_j^C}(\mathbf{x}_j) = \mathbf{b}_t^{\mathbf{x}_{ij}^C}(\mathbf{x}_j), \forall i \neq j, i, j \in \{1, 2, \ldots, N\}, \tag{24}$$

where $\mathbf{x}_{ij}^C$ stands for the complement of set $\{\mathbf{x}_i, \mathbf{x}_j\}$. From the perspective of ensemble learning, the original LRE-Learning problem $\mathcal{A}(\mathcal{X}^N)$ and the leave-$\mathbf{x}_i$-out problem $\mathcal{A}(\mathcal{X}^N \backslash \mathbf{x}_i)$ share the same "basic hypotheses" $\mathbf{b}_t(\mathbf{x}), \forall t \in \{1, 2, \ldots, T\}$, and constraints $\boldsymbol{\alpha} \geqslant \mathbf{0} \& \|\boldsymbol{\alpha}\|_1 \leq \lambda$. The only difference is that the former problem involves one more training sample, $\mathbf{x}_i$. We know that usually $N \gg 1$, thus one can

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

approximately consider their solutions are the same, *i.e.*

$$\boldsymbol{\alpha}^*_{\mathbf{x}_i^c} = \boldsymbol{\alpha}^*, \ \forall i, \tag{25}$$

where $\boldsymbol{\alpha}^*_{\mathbf{x}_i^c}$ is the optimal solution for problem $\mathcal{A}(\mathcal{X}^N \backslash \mathbf{x}_i)$. Finally, we arrive at the following theorem

Theorem 3.1. *With Eq. (25) holding, the training error of the LRE-Learning exactly equals to its leave-one-out error.*

Proof of the above theorem is proved in the supplementary. In practice, the above observation and Eq. (25) could be only considered as approximately true. However, we still can treat the training error as a good estimate to the leave-one-out error. Recall that n-fold cross-validation is also an approximation to the leave-one-out validation. We thus can directly employ the training error of LRE to choose the model-parameter, without an extra validation procedure. The fast model-selection, termed "training-determined model-selection" is justified empirically in the experiment. We tune the λ for both LRE and Boosted-LRE, which is introduced below. No significant overfitting is observed.

In this sense, one can directly set the parameter λ to a very small value, *e.g.* $\lambda = 1e-5$, to achieve the LRE model with a good generalization capability. A simple theoretical proof and a empirical evidence is given in the supplementary.

3.4. LRE-boosting for immense PPR sets

In principle, the convex optimization for the LRE-Learning could be solved easily. Nonetheless, sometimes the patch number T is enormous or even infinite. In those scenarios, to solve problem (19) via standard convex solvers is intractable. Recall that boosting-like algorithms can exploit the infinite functional space effectively [27,28]. We therefore can solve the immense problem in a stage-wise fashion, *i.e.*, the PPRs are added into the LRE model one by one, based upon certain criteria.

3.4.1. Solving the immense optimization problem via column-generation

The conventional boosting algorithms [27,28] conduct the optimization in a coordinate-descend manner. However, it is slow and cannot guarantee the global optimality at every step. Recently, several boosting algorithms based on the column-generation [36,31,37] were proposed and showed higher training efficiencies. We thus follow their principle to solve our problem.

To arrive at the boosting-style LRE-Learning, the dual problem of (19) need to be derived firstly.

Theorem 3.2. *The Lagrange dual problem of (19) writes*

$$\begin{aligned} \min_{\mathbf{u},r} \quad & r + \frac{1}{\lambda}\sum_i^N (u_i \log\, u_i - u_i) \\ \text{s.t.} \quad & \sum_{i=1}^N u_i \left(b_{t,l_i}(\mathbf{x}_i) - \frac{1}{K} \right) \le r, \ \forall t, \mathbf{u} \succcurlyeq 0. \end{aligned} \tag{26}$$

Refer to the supplementary for the proof. In Theorem 3.2, $\mathbf{u} = [u_1, u_2, \ldots, u_N]$ is usually viewed as the weighted data distribution. Considering that PPR is instanced-based and thus depends on $\mathbf{u}$, we then use $\mathbf{b}_t^{\mathbf{u}}$ to represent the t-th PPR under the data distribution $\mathbf{u}$. With the column-generation scheme employed in [36,31,37] and Theorem 3.2, we design a boosting-style LRE-Learning algorithm. The algorithm, termed LRE-Boosting, is summarized in Algorithm 1

Algorithm 1. LRE-Boosting.

Input:

- A set of training data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$.
- A set of patch-locations, indexed by $1, 2, \ldots, T$.
- A termination threshold $\epsilon > 0$.
- A maximum training step S.
- A primitive dual problem:

$$\min_{\mathbf{u},r} r + \frac{1}{\lambda}\sum_i^N (u_i \log\, u_i - u_i), \ \text{s.t.} \ \mathbf{u} \succcurlyeq 0.$$

begin

· Initialize $\boldsymbol{\alpha} = 0, \ t = 0, \ u_i = 1/N, \ \forall i$;

for $s \leftarrow 1$ **to** S **do**

· Find a new PPR, $\mathbf{b}_{t^*}^{\mathbf{u}}$, such that

$$t^* = \text{argmax}_{t \in \{1,2,\ldots,T\}} \sum_{i=1}^N u_i \left(b_{t,l_i}^{\mathbf{u}}(\mathbf{x}_i) - 1/K \right); \tag{27}$$

· **if** $\sum_{i=1}^N u_i \left(b_{t^*,l_i}^{\mathbf{u}}(\mathbf{x}_i) - 1/K \right) < r + \epsilon$, **break**;

· Assign the inequality

$$\sum_{i=1}^N u_i \left(b_{t^*,l_i}^{\mathbf{u}}(\mathbf{x}_i) - 1/K \right) \le r$$

into the dual problem as its $s-$th constraint;

· Solve the updated problem;

· Calculate the primal variable $\boldsymbol{\alpha}$ according to the dual solutions and KKT conditions;

end

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

3.4.2. Significantly faster – the data-weight-free training

For the conventional weak hypotheses used in boosting, such as decision trees, decision stumps and the LDA classifiers, one needs to re-train them after the training samples' weights $\mathbf{u}$ are updated. Usually, the re-training procedure dominates the computational complexity [36,31].

Clearly, we need to follow this computationally expensive scheme since PPRs are totally data-dependent. It is easy to see the computation complexity of each PPR is

$$C_L = \mathcal{O}(M^3) + \mathcal{O}(M^2 d), \tag{28}$$

then the complexity of the training procedure is given by

$$C_{train} = T \cdot S \cdot C_L = \mathcal{O}(TSM^3) + \mathcal{O}(TSM^2 d), \tag{29}$$

The entire training procedure can be very slow when T and S are both large.

However, we argue that: *the LRE-Boosting can be performed much faster*. To explain this, let us firstly rewrite the constraint $\mathbf{u} \geqslant \mathbf{0}$ in (26) as $\mathbf{u} \succ \mathbf{0}$. This change won't influence the interior-point based optimization method [32]. Then we can prove the following theorem.

Theorem 3.3. *Given that* $\mathbf{u} \succ 0$, *The PPRs are independent of the weight vector* $\mathbf{u}$. *In other words, for LRE-Boosting, all the PPRs need to be trained only once.*

The final part of the supplementary gives the proof. According to the theorem, one needs to calculate the PPRs only once, thus the training cost is reduced by S times to

$$\tilde{C}_{train} = T \cdot C_L = \mathcal{O}(TM^3) + \mathcal{O}(TM^2 d). \tag{30}$$

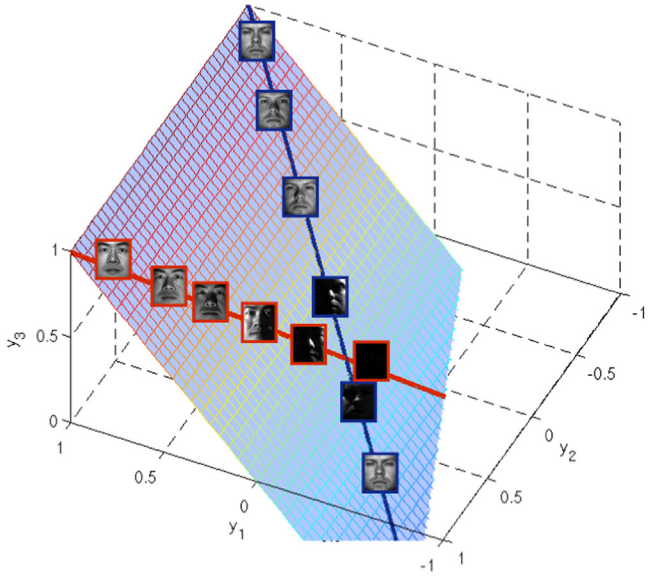

Fig. 3. Demonstration of the generic-face subspace in the original 3-D feature space. Faces from all the categories (K=2 here) form a 2-D linear subspace, *i.e.*, a plane shown in light blue. Two linear subspaces, *i.e.*, the lines shown in blue and red, respectively, correspond to two different subjects. In this work, however, we are only interested in the face patches and consequently the "generic-face-patch" subspace are considered instead. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

4. Adapting the linear representation ensemble to the test face

The learning algorithms for LRE models has been proposed in Sections 3 and 3.4. Now we design the inference approach for the LRE model.

4.1. Generic face confidence

When a facial patch is clean, the standard PPR is informative enough. However, in the test phase, some unknown patterns, which usually present as non-facial patches, might occur. Most LR methods, including the standard PPR, only pay attention to

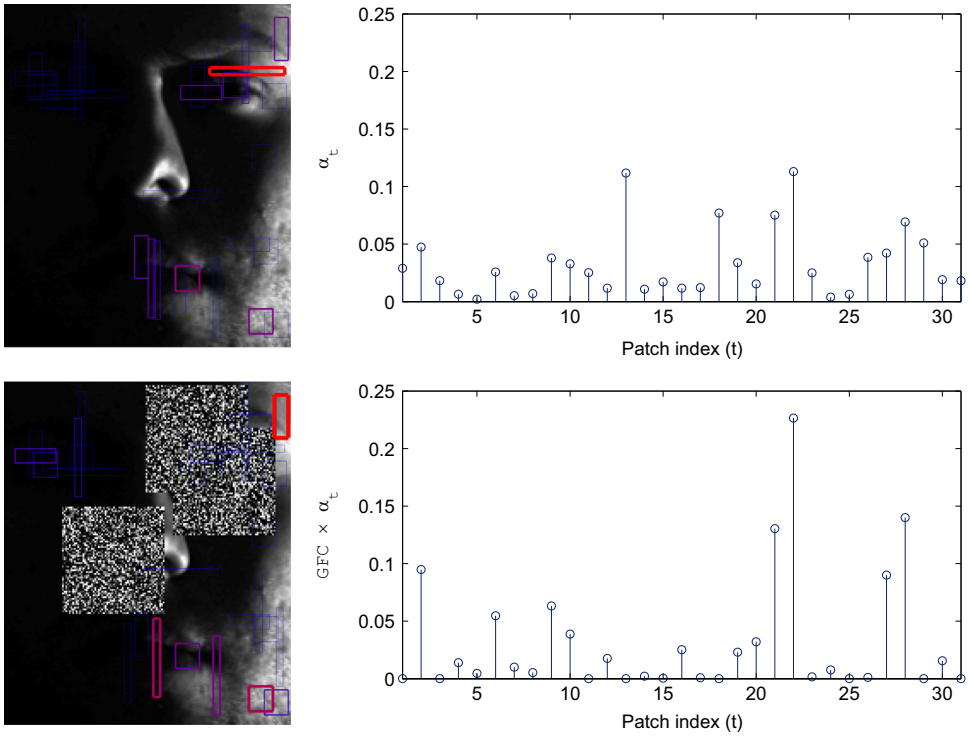

Fig. 4. The demonstration for the patch-weight adaption method. Upper row: the selected patches by LRE-Learning. Their weights are shown as stems in the left chart. Bottom row: one test face contaminated by three noisy blocks. The patches' weights are modified by using GFC.

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

distinguish the face between different individuals thus can hardly handle these kind of patterns. On the other hand, several evidences [38] suggests that *generic faces, including all the categories, also form a linear subspace*. The linear subspace is sufficiently compact comparing with the general image space. Furthermore, some visual tracking algorithms have already employed LR approaches (SRC or its variations) to distinguish the foreground from the background [39,40].

Inspired by the successful implementations, we propose to employ the linear representation for distinguishing face patches from face-unrelated or partly-face-related patches. Specifically, a badly contaminated face patch is supposed to be distant from the linear subspace spanned by the clean face patches in the same position.

Fig. 3 illustrates the assumption about the linear subspace of generic-faces. Note that the faces are merely for demonstration, in this paper, we focus on the face patches. According to this assumption, one test patch will be considered as a face part only when it is close enough to the corresponding "generic-face-patch" subspace.

Now we formulate this idea in the Bayesian framework. Given that all the training face patches $\mathbf{X}^t=[\mathbf{X}_1^t,\mathbf{X}_2^t,\ldots,\mathbf{X}_K^t]\in\mathbb{R}^{d\times N}$ are clean and forming the representation basis, for a test patch $\mathbf{y}_t$, the reconstruction residual $\tilde{r}_t^2$ is given by:

$$\tilde{r}_t^2=\|\mathbf{y}_t-\mathbf{X}^t({\mathbf{X}^t}^\top\mathbf{X}^t)^{-1}{\mathbf{X}^t}^\top\mathbf{y}_t\|_2. \tag{31}$$

Let us use the notation $u_t=1$ to indicate that $\mathbf{y}_t$ is a face patch while $u_t=0$ indicates the opposite. After taking the non-facial category into consideration, the original posterior in (9) is equivalent to $P\left(\gamma_{\mathbf{y}}=k\mid u_t=1,\mathbf{y}_t\right)$. The new target posterior becomes

$$\begin{aligned}\tilde{b}_{t,k}&=P\left(\gamma_{\mathbf{y}}=k,u_t=1|\mathbf{y}_t\right)=P\left(\gamma_{\mathbf{y}}=k|u_t=1,\mathbf{y}_t\right)\\&\quad\cdot P(u_t=1|\mathbf{y}_t)=b_{t,k}\cdot P(u_t=1|\mathbf{y}_t).\end{aligned} \tag{32}$$

Following the principle of linear subspace, we can assume that

$$P(\mathbf{y}_t|u_t=0)=C_0,P(\mathbf{y}_t|u_t=1)=C_1\cdot\exp(-\tilde{r}_t^2/\tilde{\delta}),$$

where C_1, C_0 is the normalization constant. The subspace for the non-facial category is the universe space $\mathbb{R}^d$, which leads to the uniform distribution $P(\mathbf{y}_t|u_t=0)=C_0$. Recall that all the patches are normalized, thus the domain of $\mathbf{y}_t$ is bounded. One can calculate both C_1 and C_0 with a specific $\tilde{\delta}$. For simplicity, let us define

$$\tilde{C}=\frac{C_0\cdot P(u_t=0)}{C_1\cdot P(u_t=1)}=\frac{C_0}{C_1}, \tag{33}$$

because without any specific prior we usually consider $P(u_t=0)=P(u_t=1)$. We then arrive at the new posterior, which is given by

$$\tilde{b}_{t,k}=b_{t,k}\cdot P(u_t=1|\mathbf{y}_t)=\frac{b_{t,k}\cdot P(\mathbf{y}_t|u_t=1)\cdot P(u_t=1)}{\sum_{j\in\{0,1\}}P(\mathbf{y}_t|u_t=j)\cdot P(u_t=j)}=\frac{b_{t,k}}{1+\tilde{C}\exp(\tilde{r}_t^2/\tilde{\delta})}. \tag{34}$$

In practice, we replace the original $\tilde{b}_{t,k}$ with its upper bound

$$1/\tilde{C}\cdot\exp(-\tilde{r}_t^2/\tilde{\delta})\cdot b_{t,k} \tag{35}$$

Note that the constant $\tilde{C}$ won't influence the final classification result as all the PPRs are linear combined. As a result, we can discard the term $1/\tilde{C}$ and avoid the complex integral operation for calculating it.

We call the term $\exp(-\tilde{r}_t^2/\tilde{\delta})$ the *Generic Face Confidence* (GFC) as it peaks when the patch is perfectly represented by generic face patches. With this confidence, we can easily estimate how an image patch is face related, or in other words, how is it contaminated by occlusions or noises. Note that the variance $\tilde{\delta}$ is usually data-dependent, we thus set $\tilde{\delta}=0.05\cdot(1/T\cdot\sum_t^T\tilde{r}_t)^2$ for all the faces.

4.2. The inference approach equipped with generic face confidence

With the unknown patterns, the learned patch-weights $\alpha_t\forall t$, could not guarantee their optimality anymore. A highly weighted patch-location could be corrupted badly on the test image. In this scenario, it should merely play a trivial role in the test phase. As derived above, the posterior $b_{t,k}$ is replaced by $GFC_t\cdot b_{t,k}$ when taking the non-facial category into consideration. Consequently, the original LRE is updated as

$$\boldsymbol{\xi}=\sum_{t=1}^{T'}\alpha_t\mathbf{b}_t\longrightarrow\tilde{\boldsymbol{\xi}}=\sum_{t=1}^{T'}\alpha_t^q\mathrm{GFC}_t\mathbf{b}_t \tag{36}$$

where $\alpha_t^q, q<1$ is the "faded" α. As patches are contaminated by unknown patterns, the learned weights are less trusted. The smaller the q is, the less we take account of the previously learned weights. Note that T' is the number of selected patches via the previous LRE-Learning and usually $T'\ll T$ as we impose a ℓ_1 regularization on the loss function. Fig. 4 gives us a explicit illustration of the mechanism of the patch-weight adaption procedure. In

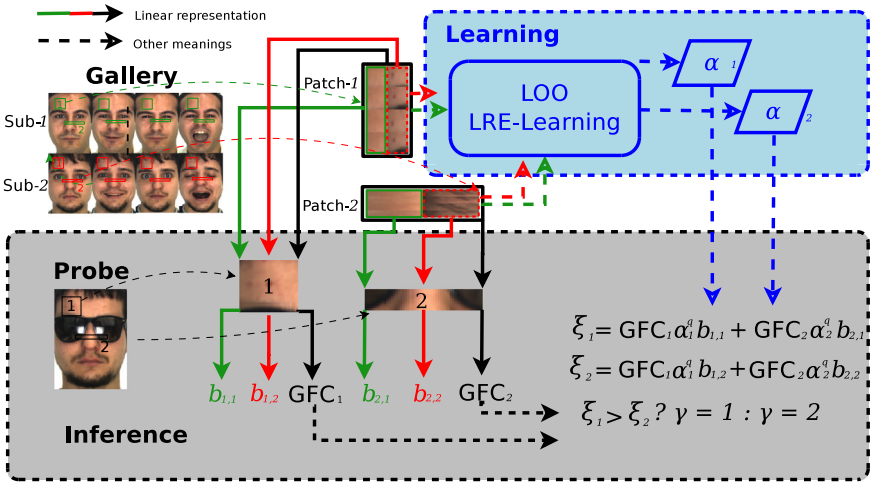

Fig. 5. Demonstration of LRE-Learning and the inference procedure. The simplified problem only contains two subjects and two patch candidates. All the green items are related to Sub-1 while the red ones are related to Sub-2. The solid arrows indicate linear representation approaches, with different colors standing for different representation basis. The black solid arrows represent the representations based on all the patches from a certain position while the green and red ones stand for those corresponding to Sub-1 and Sub-2, respectively. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

Fig. 6. The demonstration of Yale-B dataset with extreme illumination conditions.

the upper row, 31 patches are selected by LRE-Learning. Their weights are also shown as stems in the left chart. When a test face is badly contaminated by noisy occlusions, as shown in the bottom row, its importance is not reliable anymore. After modification by the proposed methods, all the large weights are assigned to the clean locations. Consequently, the following classification can hardly be influenced by the occlusions. *From a bionic angle, the weight-adaption is analogue to a focus-changing procedure, as the previously emphasized parts look "unfamiliar" and are not reliable anymore.*

The patch-weight adaption plays a essential role in the inference part of the LRE algorithm. To distinguish the inference-facilitated LRE model from the original ones, we refer to it as Adaptive-LRE. Some prior work has been done to reject the non-face part in the image [41,42]. However, our GFC is derived based on the theoretical work in [13] and [12] and does not require any feature extraction process. Compared with the anti-noise method proposed in [3] (see optimization problem (7)), our Adaptive-LRE does not impose a sparse assumption on the corrupted part thus we can handle much larger occlusions. Furthermore, our method is much faster than the robust SRC while maintaining its high robustness, as shown in the experiment. Most recently, Zhou et al. [43] proposed an advanced version of Eq. (7) via imposing a spatially continuous prior to the error vector e. The algorithm, admittedly, performed very well, especially on the face with single occlusion. However, we argue that the performance gain is due to the extra spatial prior knowledge. In this paper, none of the spatial relation is considered.

4.3. Face recognition using LRE — weighting the local patches for both training and test faces

Fig. 5 summarizes the LRE algorithm with a simplified setting where only two subjects (Sub-1 and Sub-2) and two patches (patch-1 on the right forehead and patch-2 on the middle face) are involved.

From the flow chart, we can see that the LRE algorithm is, in essence, a mixture of inference and learning. First of all, the patches are cropped and collected according to their locations and identities (different columns in one collection). Secondly, the leave-one-out margins are generated based on the leave-one-out PPRs. Then the existing face patterns are learned via the LRE-Learning or LRE-Boosting procedure. The learned results, α_1 and α_2, indicate the importances of the two patches. When a probe image is given, one perform 3 different linear representations for each test patch. The LRs with the patches from Sub-1 and Sub-2 generate the PPRs $b_{t,1}$ and $b_{t,2}$ ($t \in \{1,2\}$), respectively. In addition, we also use all the patches from one location to represent the corresponding test patch. In this way, the Generic Face Confidence (GFC $_t, \forall t$) is calculated for each location. When calculating the LRE output $\xi_i, i \in \{1,2\}$, we multiply the term $\alpha_t^q b_{t,k}$ with the corresponding GFC $_t$. In this sense, one reduces the influence of unknown patterns (like the sunglasses in the example) arise in the test image. This is, typically, an inference manner based on the learned information (α_1 and α_2) and the prior assumption (the linear subspaces corresponding to different individuals and the generic face patches). Finally, the identity γ is obtained via a simple comparison operation.

The superiority of this learning-inference-based strategy is demonstrated in the next experiment section.

5. Experiments

We design a series of experiments to evaluate different properties of the proposed algorithm on four well-known datasets, *a.k.a.* Yale-B [12], AR [26], FRGC [44] and LFW [45].

5.1. Experiment setting

5.1.1. Comparing methods

For the faces captured in laboratories, *i.e.*, those from Yale-B and AR datasets, we compare the LRE algorithm with some relatively classical LR-based methods, *i.e.* Nearest Feature Line (NFL) [1], Sparse Representation Classification (SRC) [3], Linear Regression Classification (LRC) [4] and the two modular heuristics: DEF and Block-SRC. As a baseline, the Nearest Neighbor (NN) algorithm is also performed. We choose those "classical methods" because for Yale-B and AR, they can already achieve results with high accuracies. For FRGC and LFW, in which the recognition tasks are more difficulty, we compare the results of LRE with some basic methods including Eigenface [46], Fisherface [47], OTF-based [48] and OEOTFbased [49] CFA , SRC [3], and the state-of-the-art local-based FE algorithms including Block-FLD [50], Cascaded LDA (C-LDA) [51], Hierarchical Ensemble Classifier (HEC) [52], Block-based Bag-Of-Words (BBOW) [53], Patch-based Collaborative Representation based Classification (PCRC) [8], and the most recently proposed MS-CEB method [7].

5.1.2. Algorithm parameters

In general, the LRE algorithm is performed with 500 patches. For Yale-B and AR, each patch comprised of 225 pixels. The widths of those patches are randomly selected from the set $\{5, 9, 15, 25, 45\}$ and consequently we generate the patches with 5 different aspect-ratios. The patch features are then randomly mapped into a lower dimensional space in which the linear regressions are conduct. In this work

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

Table 1
The comparison of accuracy on Yale-B. The highest recognition rates are shown in bold. Note that we only perform algorithms with the Fisherface (LDA) on the 25-D feature space. The original patch has 225 pixels, thus we cannot conduct LRE algorithms with 400-D features.

Algorithm		25-D	50-D	100-D	200-D	400-D
LDA	NN	93.4 ± 1.3	–	–	–	–
	NFL	89.4 ± 1.0	–	–	–	–
	SRC	92.5 ± 1.2	–	–	–	–
	LRC	58.0 ± 1.9	-	-	-	-
Rand	NN	42.6 ± 4.0	51.4 ± 1.5	54.2 ± 3.0	54.8 ± 1.7	56.6 ± 1.5
	NFL	83.2 ± 1.7	88.2 ± 1.0	89.5 ± 0.6	90.7 ± 0.5	90.9 ± 0.4
	SRC	80.1 ± 1.6	90.7 ± 1.0	94.7 ± 0.5	96.6 ± 0.7	97.1 ± 0.5
	LRC	25.9 ± 4.1	88.1 ± 0.6	93.1 ± 1.2	94.5 ± 0.4	94.7 ± 0.4
PCA	NN	22.3 ± 1.8	30.4 ± 1.7	34.4 ± 0.5	36.6 ± 1.2	37.0 ± 1.0
	NFL	69.5 ± 1.4	77.4 ± 1.2	81.4 ± 1.0	83.0 ± 0.5	83.5 ± 0.5
	SRC	80.4 ± 1.6	89.1 ± 0.9	92.8 ± 0.8	94.2 ± 0.7	95.1 ± 0.7
	LRC	74.7 ± 1.9	88.1 ± 0.4	89.8 ± 0.3	90.7 ± 0.5	90.8 ± 0.6
LRE		96.5 ± 0.5	99.6 ± 0.2	99.7 ± 0.1	$\mathbf{99.9} \pm 0.1$	–
Boosted-LRE		95.6 ± 1.2	99.6 ± 0.2	99.8 ± 0.1	$\mathbf{99.9} \pm 0.1$	–
Adaptive-LRE		$\mathbf{98.3} \pm 0.3$	$\mathbf{99.8} \pm 0.2$	$\mathbf{99.9} \pm 0.1$	$\mathbf{99.9} \pm 0.1$	–

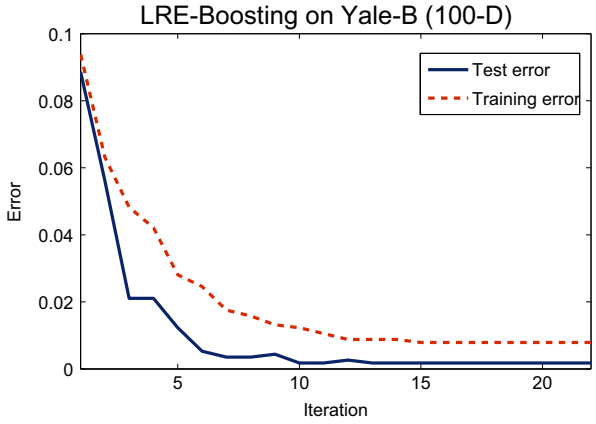

Fig. 7. Demonstration of the boosting procedure of LRE-Boosting with 100-D features on Yale-B.

we perform the LRE method with dimensions $\{25, 50, 100, 200\}$.[2] For more difficult datasets FRGC and LFW, we follow the setting in [8] thus that each face is 32×32 and we randomly generate 500 face patches with different pixel numbers which are randomly selected from the set $\{36, 64, 144, 196, 256\}$ and random aspect ratios (AR) which are pre-bounded so that $0.25 \leq AR \leq 4$. For the faces patches in FRGC and LFW, no further dimension reduction is performed. The inverse value of the trade-off parameter, *i.e.* $1/\lambda$, is selected from candidates $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, via the training-determined model-selection procedure. The variance δ and $\tilde{\delta}$ are set according to the rule described in Section 2.3 and Section 4.1 respectively. We let $q = 0.2$ for the LRE update. As to LRE-Boosting, we set the convergence precision $\epsilon = 1e-5$ and the maximum iteration number $S = 100$. For the comparing LR-based methods, random projection (Randomfaces) [3], PCA (Eigenfaces) [41] and LDA (Fisherfaces) [54] are used to reduce the dimension to 25, 50, 100, 200, 400. Note that the dimension of Fisherfaces are constrained by the number of classes. When carrying out the modular methods, we partition all the faces into 8 (4×2) blocks and downsample each block to smaller ones in the size of 12×9, as recommended by the authors [4]. Other involved methods which are not mentioned above are set following the experiment part of [7].

[2] We treat the LRE's results with original patches (225-D, no dimension reduction) as its 200-D performance.

5.1.3. Other settings

For Yale-B and AR, the test is repeated 5 times with each individual experimental setting. We then report the average results and the corresponding standard deviations. Every training and test sample, *e.g.* faces, patches and blocks, are normalized so that they have unit ℓ_2-norm. For FRGC and LFW, the experiment is conducted for 20 times and the patches are normalized using the ZCA whitening method. All the algorithms are conducted in Matlab-R2014*a*, on the Laptop PC with a 2.5GHz quad-core CPU and 16 GB RAM. When testing the running speed, we only enable one CPU-core. All the optimization, including the ones for LRE-Learning, LRE-Boosting and SRC, are performed by using Mosek [33].

5.2. Face recognition with illumination changes

Yale-B contains 2414 well-aligned face images, belonging to 38 individuals, captured under various lighting conditions, as illustrated in Fig. 6. For each subject, we randomly choose 30 images to compose the training set and other 30 images for testing. The Fisherfaces are only generated with dimension 25 as LDA requiring that the reduced dimension is smaller than the class number. When performing LRC and LRE with 25-D data, we only randomly chose 20 training faces since the least-square-based approaches need an over-determined linear system.

The experimental results are reported in Table 1. As can be seen, the LRE-based algorithms *consistently outperform all the competitors*. Moreover, all the proposed methods achieve the accuracy of 99.9% on 200-D (225-D in fact) feature space. To our knowledge, *this is the highest recognition rate ever reported for Yale-B under similar circumstances*. Given 1140 faces are involved as test samples and the recognition rate 99.9%, *only 1 faces are incorrectly classified in average*. In particular, Adaptive-LRE, *i.e.* LRE equipped with GFCs, shows the highest recognition ability. Its recognition rates are always above 99.8% when $d \geq 50$. The boosting-like variation of the LRE algorithm performs similarly to its prototype and also superior to the performances of other compared methods.

Fig. 7 shows the boosting procedure, *i.e.*, the training and test error curves, for the LRE-Boosting algorithm with 100-D features. We observe fast decreases for both curves. That justifies the efficacy of the proposed boosting approach. Furthermore, no overfitting is illustrated even though the optimal model parameter λ is selected according to the training errors. It empirically supports our theoretical analysis in Section 3.3.

On Yale-B, both LRE-Learning and its boosting-like cousin only select a very limited part (usually around 5%) of all the candidate patches, thanks to the ℓ_1-norm regularization. To illustrate this, Fig. 8 shows all the candidates (Fig. 8(a)), the selected patches by LRE-Learning (Fig. 8(b)), and those selected by LRE-Boosting (Fig. 8(c)). We can see that two algorithms make similar selections: in terms of patch positions and patch numbers (32 for LRE-Boosting *vs.* 31 for LRE-Learning). Nonetheless, minor differences is shown w.r.t the weight assignment, *i.e.*, assigning values to the coefficients $\alpha_i, \forall i$. The LRE-Boosting aggressively assigns dominant weights to a few patches. In contrast, LRE-Learning distributes the weights more uniformly. The more conservative strategy often leads to a higher robustness.

5.3. Face recognition with synthetic occlusions

Sometimes the faces are contaminated by occlusions and most state-of-the-arts may fail on some of them. The most occlusions occur on face images could be divided into two categories: noisy occlusions and disguises. Let us consider the noisy ones first.

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

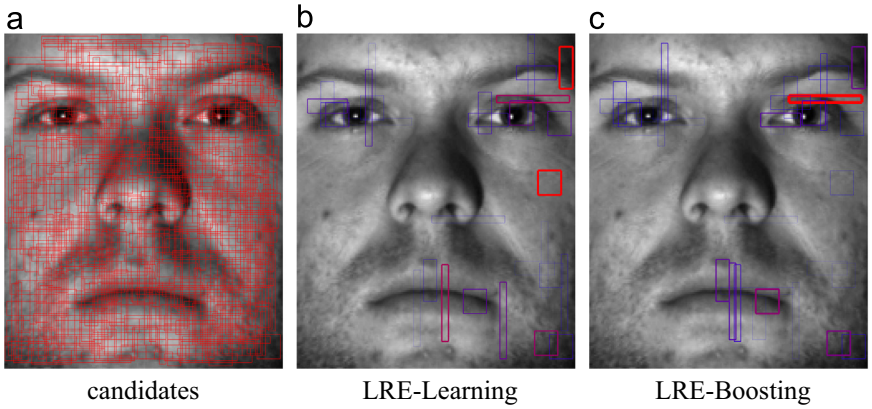

Fig. 8. The patch candidates (a) and those selected by LRE-Learning (b) and LRE-Boosting (c). All the patches are shown as blocks. Their widths and colors indicate the associated weights $\alpha_i, \forall i$. A thicker and redder edge stands for a larger α_i, *i.e.* a more important patch. The LRE algorithms are conducted on a 100-D feature space.

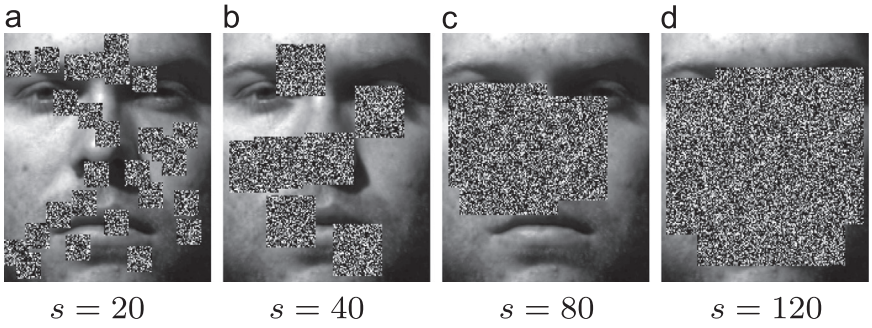

Fig. 9. The Yale-B faces with Gaussian noise occlusions. The block size is increased from 20 to 120. We can see that when $s=120$, more than 70% of the face image are totally contaminated. (a) $s=20$. (b) $s=40$. (c) $s=80$. (d) $s=120$.

Table 2
The comparison of accuracy on the occluded Yale-B. The highest recognition rates are shown in bold. Adaptive-LRE represents the LRE-model with the updating procedure. Note that the original LRC is performed with 400-D Randomfaces.

Algorithm	$s=20$	$s=40$	$s=60$	$s=80$	$s=100$	$s=120$
LRC (400-D)	74.1 ± 1.4	69.7 ± 1.3	68.4 ± 1.5	45.5 ± 1.4	30.4 ± 0.7	16.7 ± 0.2
DEF	42.9 ± 0.3	80.1 ± 0.4	88.8 ± 1.0	72.3 ± 0.6	48.0 ± 1.4	26.6 ± 1.3
Block-SRC	94.1 ± 0.5	93.3 ± 0.5	94.1 ± 0.5	85.7 ± 0.8	78.3 ± 0.4	56.8 ± 0.6
LRE	93.9 ± 2.6	98.2 ± 1.0	98.8 ± 0.6	97.5 ± 1.7	94.2 ± 3.6	86.1 ± 8.9
Adaptive-LRE	$\mathbf{98.5} \pm 0.7$	$\mathbf{99.6} \pm 0.2$	$\mathbf{99.7} \pm 0.1$	$\mathbf{99.4} \pm 0.5$	$\mathbf{98.3} \pm 1.0$	$\mathbf{93.8} \pm 4.6$

The noisy occlusions are the ones not supposed to arise on a human face, or in other words, not face-related. They are unpredictable, and thus hard to learn. We then design a experiment to verify the inference capability of LRE-based methods. Considering that LRE-Learning and LRE-Boosting select similar patches, we only perform the former one in this test. To generate the corrupted samples for testing, we impose several Gaussian noise blocks on the Yale-B faces. The blocks are square and in the size of $s \times s$, $s \in \{20, 40, 60, 80, 100, 120\}$. The number of the blocks are defined by

$$N_o = \max\{\text{round}(0.4\sigma_f/s^2), 3\}, \tag{37}$$

where σ_f represents the area of the whole face image. That is to say, the occluded parts won't cover more than 40% area of the original face, unless the number requirement $N_o \geq 3$ is not met. The yielded faces are shown in Fig. 9. We can see that when $s=120$, the contaminated parts dominate the face image.

Before testing, we train our LRE models on the clean faces (30 faces for each individual). Then, on the contaminated faces (also 30 faces for each individual), we test the learned models, with or without GFCs, comparing to the modular heuristics. In this way, we guarantee that no occlusion information is given in the training phase. As a reference, we also perform the standard LRC to illustrate different difficulty levels. The experiment is repeated 5 times with the training and test faces selected randomly[3]. The results are shown in Table 2.

As we can see, again, the proposed LRE models achieve overwhelming performances. In particular, the original LRE-models are nearly (except for the case where $s=20$) consistently better than all the state-of-the-arts. Furthermore, the Adaptive-LRE models illustrate a very high robustness to the noisy occlusions. It is always ranked first in all the conditions and achieves the recognition rates above 98% when $s<120$. Recall that the performance obtained by LRE models on clean test sets is 99.9%. *The severe occlusions merely reduce the performance of LRE model by around two percent.* When the face is dominated by continuous occlusions ($s=120$), the accuracies of modular methods drop sharply to the ones below 70% while that

[3] We guarantee that a clean face and its contaminated version won't be selected simultaneously in each test.

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

Fig. 10. Images with occlusions and expressions in AR dataset. Note that we use only gray-scale faces in the experiment.

Table 3
The comparison of accuracy on the AR dataset. The highest recognition rates are shown in bold. Adaptive-LRE represents the LRE-model with the updating procedure. Note that the original LRC is performed with 400-D Randomfaces. The results of (rCIL2 and CIL2) [55] is actually not comparable to other results in the table as the author employed a different experiment setting.

Algorithm	Expressions	Sunglasses	Scarves
LRC (400-D)	81.0	54.5	10.7
DEF	88.2	91.2	85.2
Block-SRC	87.5	95.7	86.0
rCIL2	NA	85.9*	79.8*
CIL2	NA	83.3*	77.2*
LRE	82.0 ± 1.2	85.0 ± 3.9	86.5 ± 0.7
Adaptive-LRE	**92.8** ± 0.9	**96.1** ± 1.8	**95.8** ± 1.2

Table 4
The comparison of accuracy on AR. The highest recognition rates are shown in bold. Note that we only perform algorithms with the Fisherface (LDA) on the 25-D and 50-D feature spaces. The original patch has 225 pixels, thus we can't conduct LRE algorithms with 400-D features.

Algorithm		25-D	50-D	100-D	200-D	400-D
LDA	NN	95.3 ± 0.3	97.4 ± 0.6	97.9 ± 0.5	–	–
	NFL	92.5 ± 0.8	96.8 ± 0.4	97.9 ± 0.2	–	–
	SRC	94.6 ± 0.5	97.4 ± 0.5	97.9 ± 0.4	–	–
	LRC	72.8 ± 1.6	94.5 ± 0.4	97.1 ± 0.3	–	–
Rand	NN	17.0 ± 0.9	19.8 ± 1.6	22.8 ± 2.3	22.2 ± 1.5	23.6 ± 1.1
	NFL	44.9 ± 3.0	55.2 ± 1.9	60.9 ± 1.7	63.1 ± 1.2	65.1 ± 1.2
	SRC	45.4 ± 0.5	71.3 ± 1.7	85.8 ± 1.1	91.5 ± 0.7	93.9 ± 0.6
	LRC	43.0 ± 2.2	71.6 ± 1.8	78.9 ± 1.3	82.1 ± 1.2	83.5 ± 0.7
PCA	NN	19.4 ± 1.3	20.4 ± 1.1	21.7 ± 1.3	21.8 ± 1.2	22.0 ± 1.0
	NFL	41.9 ± 1.6	48.2 ± 1.2	52.1 ± 1.6	54.3 ± 1.3	55.4 ± 1.3
	SRC	52.7 ± 0.8	72.1 ± 1.5	80.8 ± 1.0	83.6 ± 0.5	83.9 ± 0.7
	LRC	60.3 ± 0.8	75.3 ± 1.0	80.3 ± 0.7	82.1 ± 0.8	82.7 ± 0.8
LRE		97.0 ± 0.5	98.7 ± 0.5	99.0 ± 0.3	99.1 ± 0.1	–
Boosted-LRE		96.8 ± 0.3	98.6 ± 0.3	98.9 ± 0.3	99.0 ± 0.4	–
Adaptive-LRE		**98.4** ± 0.5	**99.1** ± 0.4	**99.4** ± 0.2	**99.5** ± 0.2	–

of Adaptive-LRE is still above 90%. This success justifies our assumption about the generic-face-patch linear-subspace.

5.4. Face recognition with expressions and disguises

Another kind of common occlusions are functional disguises such as sunglasses and scarves. They are, generally speaking, face-related and intentionally put onto the faces. This kind of occlusions are unavoidable in real life. Besides this difficulty, expression is another important influential factor. Expressions invalidate the rigidity of the face surface, which is one foundation of the linear-subspace assumption. To verify the efficacy of our algorithms on the disguises and expressions, we employ the AR dataset. There are 100 individuals in the AR (cropped version) dataset. Each subject consists of 26 face images which come with different expressions and considerable disguises such as scarf and sunglasses (see Fig. 10).

First of all, following the conventional scheme, we use all the clean and inexpressive faces (8 faces for each individual) as training samples and test the algorithms on those with expressions (6 faces per individual), sunglasses (6 faces per individual) and scarves (6 faces per individual), respectively. The test results can be found in Table 3. Note that the tests for Block-SRC, DEF and LRC are conducted once as the data split is deterministic. Consequently, no standard deviation is reported for those algorithms. The LRE-based methods are still run for 5 times, with different random patches and random projections. We also compare the method rCIL2 and CIL2 proposed in [55], which is designed for recognizing the "noisy faces", i.e., the occluded faces in this dataset.

According to the table, Adaptive-LRE beats other methods in all the scenarios. In particular, for the faces with scarves, both of LRE and Adaptive-LRE are superior to other methods. The performance gap between Adaptive-LRE and the involved state-of-the-arts is around 10%.

5.5. Learn the patterns of disguises and expressions

The expressions and disguises share one desirable property: they can be characterized by typical and limited patterns. One thus can learn those patterns within our ensemble learning framework. To verify the learning power of the proposed method, we re-split the data: for each individual, 13 images are randomly selected for training while the remaining ones are test images. In this way, the LRE-Learning or LRE-Boosting algorithm is given the information on disguise patterns. The experiment on AR is rerun in the new setting. Table 4 shows the recognition accuracies. Note that the results for the 100-D Fisherface are actually obtained by using 95-D features since here (100 categories) the dimension limit for LDA is 99.

Similar to the previous test, our methods once again show overwhelming superiority. *The Adaptive-LRE algorithm achieves a recognition rate of 99.5% which is also the best reported result on AR in the similar experimental setting.* In this sense, we can conclude that *the LRE algorithms can effectively learn the patterns of disguises*. The boosting-like variation of LRE-Learning obtains remarkable performances as well, but is slightly worse than the original

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

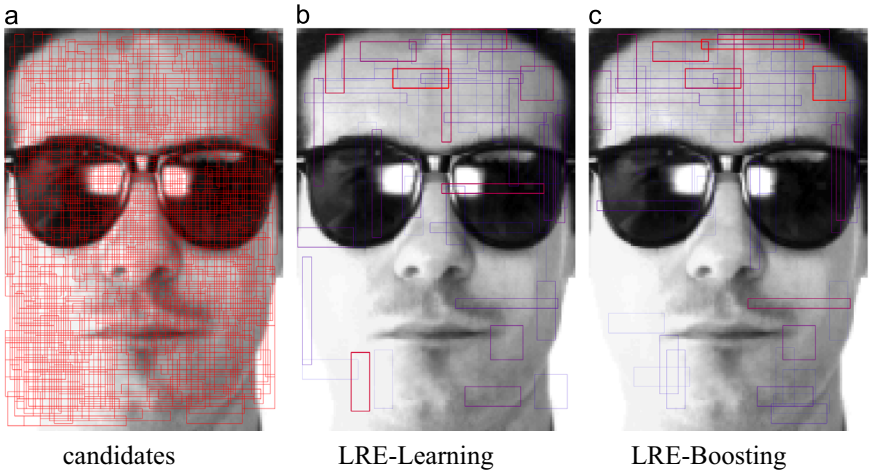

Fig. 11. The patch candidates (a) and those selected by LRE-Learning (b) and LRE-Boosting (c). All the patches are shown as blocks. Their widths and colors indicate the associated weights $\alpha_i, \forall i$. A thicker and redder edge stands for a larger α_i, *i.e.* a more important patch. The LRE algorithms are conducted on a 100-D feature space.

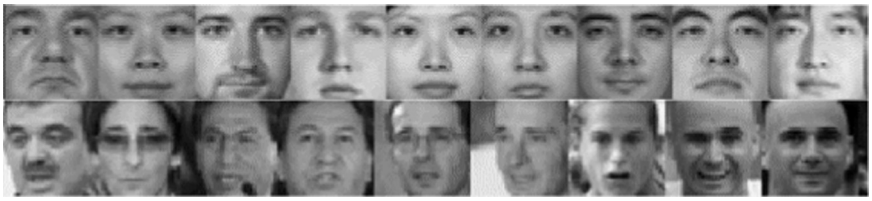

Fig. 12. Face images in FRGC (upper row) and LFW (bottom row). In each row, the most left image is the target image while the others are the query ones with uncontrolled illuminations and poor quality. The FRGC faces are aligned according to the eyes' locations as what has been done in [7] and [8]. The LFW faces are roughly aligned via the funneling method [59], which are more challenging than the manually labeled ones used in [7] and [8].

Table 5
The recognition performance comparison on FRGC. The t in the first line stands for the number of training samples per class. The highest score for each t is shown in bold font.

Algorithm	$t=2$	$t=3$	$t=4$	$t=5$
Eigenface	45.38 ± 1.3	53.10 ± 1.2	64.35 ± 1.1	70.26 ± 1.5
Fisherface	48.17 ± 1.1	55.42 ± 1.3	66.78 ± 1.5	69.06 ± 1.7
CFA (OTF)	54.35 ± 0.8	62.17 ± 0.8	65.99 ± 0.9	73.81 ± 1.0
CFA (OEOTF)	59.80 ± 0.7	70.05 ± 0.9	78.31 ± 0.7	85.04 ± 0.6
SRC	57.72 ± 1.1	65.14 ± 1.2	72.28 ± 0.9	81.18 ± 0.9
Block-FLD	53.14 ± 0.8	62.28 ± 1.3	66.77 ± 0.9	70.20 ± 1.0
C-LDA	55.72 ± 1.1	66.11 ± 0.8	72.24 ± 1.1	76.89 ± 1.2
HEC	57.28 ± 1.3	66.24 ± 1.2	71.17 ± 1.3	75.25 ± 1.5
BBOW	58.57 ± 1.4	71.90 ± 1.2	73.10 ± 0.7	78.43 ± 0.9
PCRC	59.02 ± 1.0	70.02 ± 1.0	75.65 ± 0.6	80.11 ± 0.5
MS-CFB-max	59.86 ± 1.2	70.66 ± 1.3	78.31 ± 1.2	85.53 ± 1.2
MS-CFB-cos	63.99 ± 0.8	75.24 ± 0.9	82.21 ± 0.5	**88.58** ± 0.6
LRE	**65.14** ± 0.8	**76.11** ± 1.3	82.02 ± 1.3	86.56 ± 1.0
Adaptive-LRE	63.44 ± 1.2	75.42 ± 1.4	**82.43** ± 1.1	86.88 ± 0.9

Table 6
The recognition performance comparison on LFW. The t in the first line stands for the number of training samples per class. The highest score for each t is shown in bold font.

Algorithm	$t=2$	$t=3$	$t=4$	$t=5$
Eigenface	24.15 ± 3.2	28.10 ± 3.8	32.23 ± 3.5	37.00 ± 3.7
Fisherface	27.89 ± 2.8	33.42 ± 2.7	38.42 ± 2.4	44.25 ± 2.3
CFA (OTF)	25.27 ± 3.5	30.17 ± 3.9	32.17 ± 4.0	35.24 ± 3.5
CFA (OEOTF)	30.11 ± 2.1	35.39 ± 1.8	39.95 ± 1.6	42.13 ± 1.5
SRC	30.25 ± 2.5	35.24 ± 2.3	39.97 ± 2.8	45.13 ± 2.0
Block-FLD	32.53 ± 2.3	36.78 ± 2.4	40.12 ± 1.9	45.24 ± 1.5
C-LDA	31.10 ± 2.2	35.41 ± 2.1	38.82 ± 1.5	44.99 ± 1.3
HEC	33.24 ± 2.3	41.78 ± 2.2	45.80 ± 1.5	49.72 ± 1.9
BBOW	31.27 ± 2.2	33.41 ± 1.9	41.17 ± 1.5	48.21 ± 1.5
PCRC	**38.20** ± 2.0	42.17 ± 1.4	48.58 ± 1.3	50.72 ± 1.3
MS-CFB (max)	31.10 ± 2.4	35.22 ± 2.1	42.32 ± 2.0	46.00 ± 1.8
MS-CFB (cos)	37.17 ± 1.8	43.10 ± 1.5	47.15 ± 1.4	52.20 ± 1.2
LRE	33.33 ± 2.8	**45.92** ± 1.9	53.86 ± 4.8	**59.80** ± 2.7
Adaptive-LRE	34.60 ± 2.4	45.38 ± 2.5	**54.27** ± 3.9	59.7 ± 2.5

version. Besides the LRE algorithms, the Fisherface approach also shows a high learning capacity. With Fisherfaces, the simplest Nearest Neighbor algorithm already achieves the recognition rate of 97.9%. This empirical evidence implies that discriminative face recognition methods usually benefit from learning certain face-related patterns.

Fig. 11 shows the patch candidates (Fig. 11(a)) and the selected ones for LRE-Learning (Fig. 11(b)) and LRE-Boosting (Fig. 11(c)). As illustrated in the figure, the 500 patch candidates redundantly samples the face image. Both LRE-Learning and LRE-Boosting choose 54 patches and LRE-Learning still employs a more conservative strategy of weight assignment. Differing from Fig. 8, the LRE algorithms now focus on the forehead more than eyes and the mouth. Considering that sunglasses and scarves are usually located in those two places, the disguises' patterns are learned and the corresponding patch positions are less trusted during the test.

5.6. Face recognition in uncontrolled environments

In real-world applications, the faces are usually captured in uncontrolled environments. It is important to verify a practical face recognizer in these scenarios. As the last part of our experiment, we test the LRE method on the faces in the wild. Two well-known face datasets, namely, FRGC [44] and LFW [45] are used in this test. Some sample faces of these two datasets are illustrated in Fig. 12. For a fair comparison, we follow the experimental setting used in [7] and the performances are shown in Tables 5 for FRGC and 6 for LFW. Note that both this work and [7] only focus on the

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

face identification task, rather than the binary face-pair verification problem addressed in other papers [56–58]. As the LRE-Boosting algorithm usually achieve the similar performance as the standard LRE method, here we only report the results of LRE and Adaptive-LRE algorithms.

From Tables 5 and 6, we have two observations. First of all, the LRE and its adaptive variation performs well in uncontrolled the environments. Among 8 comparisons with different conditions, our methods achieve 6 best scores and for the other 2 comparisons LRE and Adaptive-LRE also obtain the performances close to the highest ones. Secondly, as there is no significant non-facial part or unknown facial behavior in those two datasets, the Adaptive-LRE performs very similar to the original LRE method.

5.7. Efficiency

For a practical computer vision algorithm, the running speed is usually crucial. Here we show the high efficiency of the proposed algorithms, in both terms of training and test.

5.7.1. Improvement on the training speed

Here we evaluate the improvement on the training speed. Fig. 13 depicts the difference on the time consumptions for training a Boosted-LRE model, between the methods with and without updating PPRs at every iteration. The test is conducted with the increasing number (from 10 to 2000) of PPRs and trade-off parameter $\lambda = 0.02$ in the 100-D feature space. As illustrated, the efficiency gap is huge. Without the PPR-recalculation, one could save the training time by from 700 s (10 PPRs) to more than 10 days (2000 PPRs).

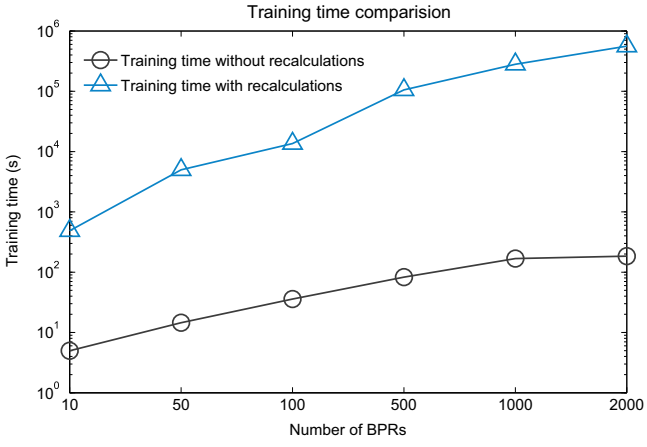

Fig. 13. The training times consumed by the LRE-Boosting methods with the PPR recalculations and without them. Note that the y-axis is shown in the logarithmic scale. The results are obtained on the AR dataset with 100-D features.

5.7.2. Recognition efficiency

At last, let us verify the most important efficiency property – execution speed. The test face (or face patch) is randomly mapped to a lower-dimensional space. Given a reduced dimension, all the face recognition algorithms are performed 100 times on faces from Yale-B. We record the elapsed times (in ms) for each method and show the average values in Fig. 14. Note that for LRC and LRE-based methods, there is no need to perform LRs when testing as all the representation bases are deterministic. Before test, one can pre-calculate and store all the matrices $\mathbf{E} = (\hat{\mathbf{X}}^\top \hat{\mathbf{X}})^{-1} \hat{\mathbf{X}}^\top$, where $\hat{\mathbf{X}}$ represents different basis for different algorithms. Then the representation coefficients $\boldsymbol{\beta}$ for the test face (or patch) $\mathbf{y}$ can be obtained via $\boldsymbol{\beta} = \mathbf{E}\mathbf{y}$.

As demonstrated, the SRC-based algorithms are the slowest two. The original SRC needs up to 31 s (400-D) to process one test face. The Block-SRC approach, which shows relatively high robustness in the literature, shows even worse efficiency. For 400-D features, one need to wait more than 4 min for one prediction yielded by Block-SRC. NFL also performs slowly. It requires 9–1113 ms to handle one test image. In contrast, the LRE-based methods consistently outperform others in terms of efficiency. In particular, on the 200-D (225-D in fact) feature space, one only needs 16 ms to identify a probe face by using either LRE algorithm. This speed not only overwhelms those of SRC and NFL, but is also 2-time higher than those of LRC and NN.

Such a high efficiency, however, seems not reliable. Intuitively, the time consumed by LRC might be always shorter than that for LRE because LRE performs multiple LRs (here actually matrix multiplications) while LRC only performs one. We then track the execution time of the Matlab code via the "profile" facility. We found that, with high-dimensional features and efficient classifiers, it is the dimension reduction which dominates the time usage. The NN and LRC algorithms both perform the linear projection over all the pixels while LRE only select a small part (the pixels in the patches) of them to do the dimension-reduction. As a result, if not too many patches are selected, the LRE algorithms usually illustrate even higher efficiencies than LRC and NN.

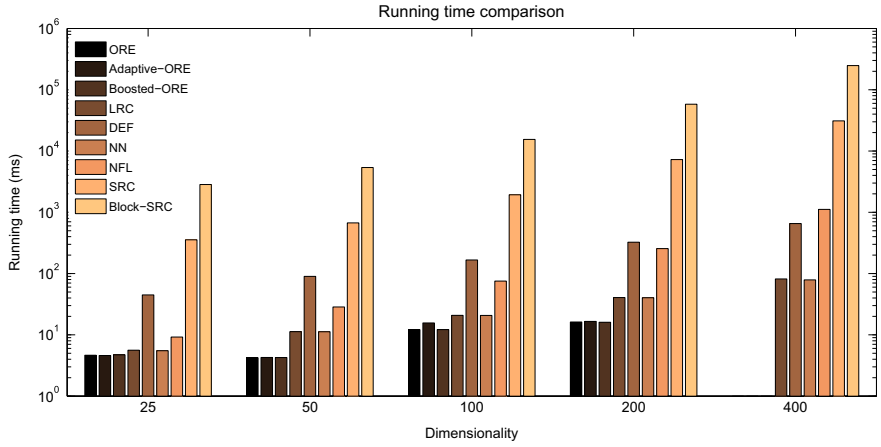

Fig. 14. Comparison of the running time. Note that the y-axis is in the logarithmic scale. We do not perform LRE algorithms in the 400-D feature space as each patch only has 225 pixels. The 200-D results for the proposed methods are actually obtained in the original 225-D space.

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

6. Conclusion

In this paper, a simple yet effective framework is proposed for face recognition. By observing that, in practice, only partial face is reliable for the linear-subspace assumption. We generate random face patches and conduct LRs on each of them. The patch-based linear representations are interpreted by using the Bayesian theory and linearly aggregated via minimizing the empirical risks. The resulting combination, Linear Representation Ensemble, shows high capability of learning face-related patterns and outperforms state-of-the-arts on both accuracy and efficiency. With LRE models, one can almost perfectly recognize the faces in Yale-B (with the accuracy 99.9%), AR (with the accuracy 99.5%), and obtain around 60.0% for the LFW identification dataset [8,7], at a remarkable speed (below 20 ms per face with the unoptimized Matlab code and a single CPU core).

To cope with the foreign patterns arising in test faces, the Generic Face Confidence is derived by taking the non-facial patch into consideration. Facilitated by GFCs, the LRE model shows a high robustness to noisy occlusions, expresses and disguises. It beats the modular heuristics under nearly all the circumstances. In particular, for Gaussian noise blocks, the recognition rate of our method is always above 93% and fluctuates around 99% when the blocks are not too large. For real-life disguises and facial expressions, Adaptive-LRE also outperforms the competitors consistently.

In addition, to accommodate the instance-based PPRs, a novel ensemble learning algorithm is designed based on the proposed leave-one-out margins. The learning algorithm, LRE-Learning, is theoretically and empirically proved to be resistant to overfittings. This desirable property leads to a training-determined model-selection, which is much faster than conventional n-fold cross-validations. For immense PPR sets, we propose the LRE-Boosting algorithm to exploit the vast functional spaces. Furthermore, we also increase the training speed a lot by proving that the LRE-Boosting is actually data-weight-free.

Conflict of Interest

None declared.

Acknowledgements

This work was supported in part the National Natural Science Foundation of China under Project 61503168 and Project 61502081.

References

[1] S. Li, J. Lu, Face recognition using the nearest feature line method, IEEE Trans. Neural Netw. 10 (1999) 439–443.
[2] J.T. Chien, C.C. Wu, Discriminant waveletfaces and nearest feature classifiers for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1644–1649.
[3] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell 31 (2009) 210–227.
[4] N. Imran, T. Roberto, B. Mohammed, Linear regression for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 2106–2112.
[5] X. Shi, Y. Yang, Z. Guo, Z. Lai, Face recognition by sparse discriminant analysis via joint l 2, 1-norm minimization, Pattern Recognit. 47 (7) (2014) 2447–2453.
[6] S. Gao, K. Jia, L. Zhuang, Y. Ma, Neither global nor local: regularized patch-based representation for single sample per person face recognition, Int. J. Comput. Vis. 111 (3) (2015) 365–383.
[7] Y. Yan, H. Wang, D. Suter, Multi-subregion based correlation filter bank for robust face recognition, Pattern Recognit. 47 (11) (2014) 3487–3501.
[8] P. Zhu, L. Zhang, Q. Hu, S.C. Shiu, Multi-scale patch based collaborative representation for face recognition with margin distribution optimization, Proc. Eur. Conf. Comp. Vis., 2012, pp. 822–835.
[9] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition? in: Proc. IEEE Int. Conf. Comp. Vis., 2011, pp. 471–478.
[10] F. Shen, C. Shen, A. van den Hengel, Z. Tang, Approximate least trimmed sum of squares fitting and applications in image analysis, IEEE Trans. Image Proc. 22 (5) (2013) 1836–1847.
[11] F. Shen, C. Shen, R. Hill, A. van den Hengel, Z. Tang, Fast approximate l_∞ minimization: speeding up robust regression, Comput. Stat. Data Anal. 77 (0) (2014) 25–37.
[12] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[13] R. Basri, D. Jacobs, Lambertian reflectance and linear subspaces, IEEE Trans. Pattern Anal. Mach. Intell. (2003) 218–233.
[14] F. Shen, Z. Tang, J. Xu, Locality constrained representation based classification with spatial pyramid patches, Neurocomputing 101 (0) (2013) 104–115.
[15] R. Herbrich, Learning Kernel Classifiers: Theory and Algorithms, The MIT Press, Cambridge, MA, USA, 2002.
[16] X. Wang, X. Tang, Random sampling LDA for face recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, vol. 2, 2004, pp. II–259.
[17] N. Chawla, K. Bowyer, Random subspaces and subsampling for 2-d face recognition, Proc. IEEE Conf. Comp. Vis. Patt. Recognit., vol. 2, 2005, pp. 582–589.
[18] J. Lu, K. Plataniotis, A. Venetsanopoulos, S. Li, Ensemble-based discriminant learning with boosting for face recognition, IEEE Trans. Neural Netw. 17 (1) (2006) 166–178.
[19] R. Xiao, X. Tang, Joint boosting feature selection for robust face recognition, Proc. IEEE Conf. Comp. Vis. Patt. Recognit., vol. 2, 2006, pp. 1415–1422.
[20] X. Wang, X. Tang, Random sampling for subspace face recognition, Int. J. Comp. Vis. 70 (1) (2006) 91–104.
[21] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3360–3367.
[22] L. Lin, P. Luo, X. Chen, K. Zeng, Representing and recognizing objects with massive local image patches, Pattern Recognit. 45 (1) (2012) 231–240.
[23] C. Zhang, S. Wang, Q. Huang, J. Liu, C. Liang, Q. Tian, Image classification using spatial pyramid robust sparse coding, Pattern Recognit. Lett. 34 (9) (2013) 1046–1052.
[24] Q. Shi, H. Li, C. Shen, Rapid face recognition using hashing, in: Proc. IEEE Conf. Comp. Vis. Patt. Recognit., 2010.
[25] Q. Shi, A. Eriksson, A. van den Hengel, C. Shen, Is face recognition really a compressive sensing problem? in: Proc. IEEE Conf. Comp. Vis. Patt. Recognit., IEEE, 2011.
[26] A. Martinez, R. Benavente, The AR face database, Technical Report, CVC, Technical report (1998).
[27] R. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn. 37 (3) (1999) 297–336.
[28] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Statist. (2000) 337–374.
[29] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[30] T. Hastie, R. Tibshirani, J. Friedman, J. Franklin, The elements of statistical learning: data mining, inference and prediction, Math. Intell. 27 (2) (2005) 83–85.
[31] C. Shen, H. Li, On the dual formulation of boosting algorithms, IEEE Trans. Pattern Anal. Mach. Intell. (2010) 2216–2231.
[32] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, United Kingdom, 2004.
[33] A. Mosek, The mosek optimization software, Online at ⟨http://www.mosek.com⟩.
[34] M. Grant, S. Boyd, Cvx: Matlab software for disciplined convex programming, Online at ⟨http://www.stanford.edu/~boyd/cvx/⟩.
[35] T. Evgeniou, M. Pontil, A. Elisseeff, Leave one out error, stability, and generalization of voting combinations of classifiers, Mach. Learn. 55 (1) (2004) 71–97.
[36] A. Demiriz, K. Bennett, J. Shawe-Taylor, Linear programming boosting via column generation, Mach. Learn. 46 (1–3) (2002) 225–254.
[37] Z. Hao, C. Shen, N. Barnes, B. Wang, Totally-corrective multi-class boosting, Proc. Asia. Conf. Comp. Vis., 2011, pp. 269–280.
[38] M. Meytlis, L. Sirovich, On the dimensionality of face space, IEEE Trans. Pattern Anal. Mach. Intell. 29 (7) (2007) 1262–1267.
[39] X. Mei, H. Ling, Robust visual tracking using ℓ_1 minimization, Proc. IEEE Int. Conf. Comp. Vis., 2009, pp. 1436–1443.
[40] H. Li, C. Shen, Q. Shi, Real-time visual tracking with compressed sensing, Proc. IEEE Conf. Comp. Vis. Pattern Recognit., IEEE, 2011.
[41] M. Turk, A. Pentland, Face recognition using eigenfaces, Proc. IEEE Conf. Comp. Vis. Pattern. Recognit., 1991, pp. 586–591.
[42] A. Martinez, Recognition of partially occluded and/or imprecisely localized faces using a probabilistic approach, Proc. IEEE Conf. Comp. Vis. Pattern. Recognit., vol. 1, 2000, pp. 712–717.
[43] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, Y. Ma, Face recognition with contiguous occlusion using Markov random fields, Proc. IEEE Int. Conf. Comp. Vis., IEEE, 2009, pp. 1050–1057.
[44] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, Proc. IEEE Conf. Comp. Vis. Pattern Recognit., vol. 1, 2005, pp. 947–954.

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011

[45] B. Gary, E. Huang, M. Learned, Labeled faces in the wild: updates and new reporting procedures, Technical Report. UM-CS-2014-003, University of Massachusetts, Amherst (May 2014).
[46] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognit. Neurosci. 3 (1) (1991) 71–86.
[47] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[48] B.V. Kumar, M. Savvides, C. Xie, Correlation pattern recognition for face recognition, Proc. IEEE 94 (11) (2006) 1963–1976.
[49] Y. Yan, Y. Zhang, 1d correlation filter based class-dependence feature analysis for face recognition, Pattern Recognit. 41 (12) (2008) 3834–3841.
[50] S. Chen, J. Liu, Z. Zhou, Making flda applicable to face recognition with one sample per person, Pattern Recognit. 37 (7) (2004) 1553–1555.
[51] T. Kim, H. Kim, W. Hwang, J. Kittler, Component-based lda face description for image retrieval and mpeg-7 standardisation, Image Vis. Comput. 23 (7) (2005) 631–642.
[52] Y. Su, S. Shan, X. Chen, W. Gao, Hierarchical ensemble of global and local classifiers for face recognition, IEEE Trans. Image Process. 18 (8) (2009) 1885–1896.
[53] Z. Li, J. Imai, M. Kaneko, Robust face recognition using block-based bag of words, Proc. IEEE Int. Conf. Pattern Recognit., IEEE, 2010, pp. 1285–1288.
[54] C. Liu, H. Wechsler, Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition, IEEE Trans. Image Process. 11 (4) (2002) 467–476.
[55] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, Z. Lin, Correntropy induced l2 graph for robust subspace clustering, in: 2013 IEEE International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 1801–1808.
[56] J. Hu, J. Lu, Y. Tan, Discriminative deep metric learning for face verification in the wild, in: Proc. IEEE Conf. Comp. Vis. Patt. Recognit., IEEE, 2014, pp. 1875–1882.
[57] Z. Lei, M. Pietikainen, S. Li, Learning discriminant face descriptor, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 289–302.
[58] G.B. Huang, H. Lee, E. Learned-Miller, Learning hierarchical representations for face verification with convolutional deep belief networks, in: Proc. IEEE Conf. Comp. Vis. Pattern Recognit., IEEE, 2012, pp. 2518–2525.
[59] G.B. Huang, V. Jain, E. Learned-Miller, Unsupervised joint alignment of complex images, Proc. IEEE Int. Conf. Comput. Vis., 2007.

Hanxi Li received a B.Sc. and a Master's degree from Beihang University, China in 2004 and 2007 respectively. He obtained his Ph.D. degree in computer science (2012) from the Australian National University. From 2011 to 2014, he worked as a researcher in the Computer Vision group of National ICT Australia (NICTA). Since September 2014, he is a special-term professor in the School of Computer and Information Engineering, Jiangxi Normal University, China. His research interests include Boosting algorithms, object tracking, object detection and face recognition.

Fumin Shen received his B.S. and Ph.D. degree from Shandong University and Nanjing University of Science and Technology, China, in 2007 and 2014, respectively. Currently he is a lecturer in school of Computer Science and Engineering, University of Electronic of Science and Technology of China, China. His major research interests include computer vision and machine learning, including face recognition, image analysis, hashing methods, and robust statistics with its applications in computer vision.

Chunhua Shen received the Ph.D. degree from the University of Adelaide, Adelaide, Australia, in 2006. He is currently a Professor with the School of Computer Science, University of Adelaide. He was with the Computer Vision Program, National ICT Australia, Canberra Research Laboratory, Canberra, Australia. His current research interests include intersection of computer vision and statistical machine learning. Dr. Shen received the Australian Research Council Future Fellowship from 2012 to 2016.

Yang Yang is currently with University of Electronic Science and Technology of China. He was a Research Fellow in National University of Singapore during 2012–2014. He was conferred his Ph.D. Degree (2012) from The University of Queensland, Australia. During the Ph.D. study, Yang Yang was supervised by Prof. Heng Tao Shen and Prof. Xiaofang Zhou. He obtained Master Degree (2009) and Bachelor Degree (2006) from Peking University and Jilin University, respectively.

Yongsheng Gao received the B.Sc. and M.Sc. degrees in electronic engineering from Zhejiang University, China, in 1985 and 1988, respectively, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore. Currently, he is a Professor with the School of Engineering, Griffith University, Australia. His research interests include face recognition, biometrics, biosecurity, image retrieval, computer vision, and pattern recognition.

Please cite this article as: H. Li, et al., Face recognition using linear representation ensembles, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2015.12.011