Fast Feature Extraction Approach for Multi-Dimension Feature Space Problems

Sagheer, Alaa
Department of Intelligent Systems, Kyushu University

Tsuruta, Naoyuki
Department of Electronics Engineering and Computer Science, Fukuoka University

Taniguchi, Rin-ichiro
Department of Intelligent Systems, Kyushu University

Arita, Daisaku
Department of Intelligent Systems, Kyushu University

他

出版情報：Proceedings of the 18th International Conference on Pattern Recognition, pp.417-420, 2006-08
バージョン：
権利関係：

Fast Feature Extraction Approach for Multi-Dimension Feature Space Problems

Alaa Sagheer[1], Naoyuki Tsuruta[2], Rin-Ichiro Taniguchi[1], Daisaku Arita[1], Sakashi Maeda[2]
[1] *Department of Intelligent Systems, Kyushu University, Japan*
[2]*Department of Electronics Engineering and Computer Science, Fukuoka University, Japan*
{alaa,rin, arita@ limu.is.kyushu-u.ac.jp} {tsuruta, maeda@tl.fukuoka-u.ac.jp}

Abstract

Recently, we proposed a fast feature extraction approach denoted FSOM utilizes Self Organizing Map (SOM). FSOM [1] overcomes the slowness of traditional SOM search algorithm. We investigated the superiority of the new approach using two lip reading data sets which require a limited feature space as the experiments in [1] showed. In this paper, we continue FSOM investigation but using an RGB face recognition database across different poses and different lighting conditions. We believe that such data sets require multi-dimensional feature space to extract the information included in the original data in an effective way especially if you have a big number of classes. Again, we show here how is FSOM reduces the feature extraction time of traditional SOM drastically while preserving same SOM's qualities.

1. Introduction

In the field of data analysis, it is important to reduce the dimensionality of original data space in order to understand data and reduces computational cost. If the original space dimensionality is very high, then even the commonly used techniques fail to reduce the dimensionality in a costless way. For instance, the multidimensional scaling techniques, such as principal component analysis, are computationally costly by their own, and if the dimensionality of input data is so high it is infeasible for them to reduce dimensionality [2].

The problem of dimensionality reduction is closely related to feature extraction. Feature extraction refers to identifying the salient aspects of data to facilitate its use in subsequent tasks such as classification or regression. As the amount of data grows larger these days, then a non-linear feature extraction approach is desired in order to reduce the data dimensions down to the number of classes in order to achieve effective feature extraction.

Though Self Organizing Map (SOM) [3] is a widely used technique achieves feature extraction in a good manner its computation cost is high when it is used in multivariate data applications. This is due to that its search algorithm usually seeks about the best matching unit (BMU) among "all" the map units which tuned to "each" input sample [4]. Therefore a new and rapid search algorithm within the context of the self-organizing map is required for situations where it is impossible to use the input vectors as such.

In this paper, a rapid search approach is proposed to extract features of large data sets. The key idea is based on the fact that the greatest variance of the data distribution comes to lie on the low-order axes; or principal components *PC*s, of the feature space. It is demonstrated that such low-order components often contain the most important aspects of the data set [5-6]. Then FSOM operates by extracting the subspaces spanned by the *PC*s of its feature space. Later on, FSOM seeks about BMU only through these subspaces instead of all units. The result is a new competition algorithm that is much faster than traditional SOM algorithm and, in the same time, performing better.

Recently, the authors investigated the proposed approach in an artificial data set and two different lip-reading data sets as well [1]. It is known that a lip based image problem requires a somehow limited feature space due to fewer classes which are usually used. Here in this paper, we try to investigate the new approach but using large scale face data set instead. Definitely, the impact of FSOM is demonstrated via experiments conducted on the large sized PIE-CMU [7] data set for face recognition across different poses and different lightening conditions. It is known that such data sets require a higher or multi dimensions feature space [8], which represents a big obstacle for traditional SOM [9].

2. Self organizing map (SOM)

The SOM algorithm [5] is an unsupervised learning algorithm and usually consists of a two-dimensional grid of units (or neurons). Each unit has a weight (reference) vector, $\boldsymbol{w}_j$, that will resemble different kind of input patterns after the learning procedure is over. The learning algorithm for SOM will accomplish two important tasks [2]:

a. Clustering the input data;
b. Topological ordering of the grid in the sense that similar input patterns tend to produce response in units that are close to each other in the grid.

Consider the input data $\mathcal{X} = \{\boldsymbol{x}_i, 1 < i < M\}$ belongs to a high dimensional space, i.e. $\boldsymbol{x}_i = (\boldsymbol{x}^{(l)}{}_i)_{1<l<m} \in R^m$. However, the traditional SOM feature map is extended through two-dimensions; a multi-dimension feature space has been successfully exploited for nowadays applications. The N-dimensional grid can be given as:

$$\mathcal{U} = \left\{ \boldsymbol{u}_{d_1, d_2, \ldots, d_j} \mid d_j = 1, 2, .., D_j, j = 1, .., N \right\} \quad (1)$$

where $\boldsymbol{u}_{d_1, d_2, \ldots, d_N}$ terms to the neuron $\boldsymbol{u}$ spanned through the N-dimensions $d_1, d_2, .., d_N$. In each dimension; like $d_j = 1, 2, .., D_j$, D_j refers to the maximum number of neurons distributed through the dimension j. Each neuron $\boldsymbol{u}_{d_1, d_2, \ldots, d_N}$ has a codebook vector $\boldsymbol{w}_{d_1, d_2, \ldots, d_N}$.

In each training step, the following two steps are repeated for each input sample $\boldsymbol{x}_i$.

1. For each dimension d_j: find the best matching unit BMU, or *winner*, c over this dimension using a similarity measure between the input and all the grid's units according to the following winner-take-all (WTA) rule:

$$\left\| \boldsymbol{x}_i - \boldsymbol{w}_{c_1, c_2, \ldots c_N} \right\| = \min_{d_j} \left(\left\| \boldsymbol{x}_i - \boldsymbol{w}_{d_1, d_2, \ldots d_N} \right\| \right) \quad (2)$$

2. Update the weigh vector of each *winner c* for each dimension and also all its topological neighborhood in the grid towards the prevailing input using the rule:

$$\boldsymbol{w}_j(t+1) = \boldsymbol{w}_j(t) + h_{cj}(t)[\boldsymbol{x}_i(t) - \boldsymbol{w}_j(t)] \quad (3)$$

$$h_{cj}(t) = \alpha(t).\exp\left(\left\| r_c - r_j \right\| / 2\sigma^2(t) \right) \quad (4)$$

where $h_{cj}(t)$ is the neighborhood kernel function around the *winner* $\boldsymbol{c}$ at time t, $\alpha(t)$ is the learning rate and is decreased gradually toward zero and $\sigma^2(t)$ is a factor used to control the width of the neighborhood kernel. The term $\| r_c - r_j \|$ refers to the distance between the *winner* neuron $\boldsymbol{c}$ and neuron j. After the training data is exhausted, the grid is automatically organized, without external supervision, into a meaningful N-dimensional order denoted by feature map (or codebook). It is clear that, to get the winner list $C = (c_1, c_2, .., c_N)$ included in (2) you need $\boldsymbol{O(D_1 \times D_2 \times \ldots \times D_N)}$ steps.

3. Fast self organizing map (FSOM)

It is demonstrated that, the greatest variance of the data distribution comes to lie on the low-order principal components *PCs* of the manifold. Accordingly, such low-order *PCs* often contain the most important aspects of the data set [5-6]. FSOM operates by extracting the subspaces spanned by the *PC*s of the feature space. The features in such subspaces provide more salient and richer information for recognition than the rest of the feature space [8]. In that sense, FSOM finds the BMU among the units distributed through these subspaces "only" not all the feature map units as the traditional SOM requires in (2); *for related works please refer to* [1].

In the context of FSOM, the traditional N-dimensions feature map of SOM can be viewed as "N" "one-dimension SOM", such that each one-dimension SOM matches a subspace spanned by a *PC*. In another terminology, the FSOM structure consists of the following N-series of "one-dimension SOM":

$$\boldsymbol{u}_1 = \left\{ u_{1,d_1} \mid d_1 = 1, 2, .., D_1 \right\} \quad (5)$$

$$\boldsymbol{u}_2 = \left\{ u_{2,d_1,d_2} \mid d_2 = 1, 2, .., D_2 \right\} \quad (6)$$

$$\ldots$$

$$\boldsymbol{u}_N = \left\{ u_{N,d_1,\ldots,d_N} \mid d_N = 1, 2, .., D_N \right\} \quad (7)$$

where the term u_{1,d_1} refers to the neurons of the first "one-dimension SOM" d_1 (or first *PC*), u_{2,d_1,d_2} refers to the neurons distributed through the second "one dimension SOM" d_2 (or second *PC*) and the former dimension d_1 and so on.

3.1. Learning phase

The learning process is described as a recursive call for the function $\text{Learn}\left(1, u_{1,d_1}, c_1\right)$, where 1 is the order n of the extracted component, u_{1,d_1} refers to the neurons of this component and c_1 is the winner neuron. Therefore the function $\text{Learn}\left(n, u_{n,d_1,\ldots,d_n}, C\right)$ can be generated as given in Table 1 below. In step 2 in the table, i.e. the "if" part, it is worthy to explain that we decide each *PC* by using the central column of the current map and copy it to first *PC* (or u_1), then for second *PC* (or u_2) and so on until extracting all *PCs*.

Table 1. Fast SOM learning algorithm

{// start of algorithm
For each n, do: If ($n \neq N$)
{1- Train the following (N-n+1)-dimensions SOM
$$u = \left\{ u_{d_n, d_{n+1}, \ldots, d_N} \mid d_j = 1, 2, .., D_j,\ j = n, n+1, .., N \right\}$$
using (WTA) rule in (2) and get the winner list.
2- For each dimension d_n, regard to the central column of the current codebook units;
$$u_n = \left\{ u_{d_n, D_{n+1}/2, \ldots, D_N/2} \mid d_n = 1, 2, .., D_n \right\},$$
as the n^{th} *PC* and "copy" it onto u_{n,d_n} .
3- For each d_n, train u_{n,d_n} using (WTA) rule in (2-3) and get the winner neuron through dimension d_n.
4- For each $n+1$, do: $\text{Learn}\left(n+1, u_{n+1,d_{n+1}}, C\right)$ }
else
{ 5- For each input sample x_i, train each neuron $u_{N,d_1,..,d_N}$. That is, for each $\left(d_N = 1, 2, .., D_N\right)$ apply the following (WTA) rule
$$\left\| \boldsymbol{x_i} - \boldsymbol{w}_{c_1, c_2, \ldots c_N} \right\| = \min_{d_N} \left(\left\| \boldsymbol{x_i} - \boldsymbol{w}_{c_1, c_2, \ldots, c_{N-1}, d_N} \right\| \right)$$
} //end of else
} //end of algorithm

The simplicity of the proposed approach is obvious as this recursive function can easily translate to one "for-loop" statement. In this "for-loop", the function $\text{Learn}\left(n, u_{n,d_n}, C\right)$ calls the function $\text{Learn}\left(n+1, u_{n+1,d_{n+1}}, C\right)$ D_n times. Now, as the size of $u_{n+1,d_{n+1}}$ is $1/D_n$ of u_{n,d_n}, therefore, the computation to train u_{n+1} is $1/D_n$ of that to train u_n. In another terminology, if the computational complexity to train u_n (or traditional N-dimension SOM) is $\boldsymbol{C}$, then, the overall computation complexity to train FSOM is:

$$\boldsymbol{C}\left(1 + 1/D_1 + 1/D_1 D_2 + \cdots + 1/\prod_i D_i\right) \approx \boldsymbol{C} \tag{8}$$

This means that learning's FSOM consumes "approximately" the same time of that to learn SOM.

3.2. Recognition (competition) phase

After getting a well ordered feature map during learning phase, we try to get the winner list through recognition phase using the following N-steps in turn.

- First, (WTA) rule is applied to select first *winner* $\boldsymbol{c}_1$ from first *PC*, or u_1 given in (5), using:

$$\left\| x_i - w_{1,\boldsymbol{c}_1} \right\| = \min_{d_1}\left(\left\| x_i - w_{1,d_1} \right\| \right) \tag{9}$$

- Second *winner* $\boldsymbol{c}_2$ is picked from second *PC*; u_2 in (6):

$$\left\| x_i - w_{2,c_1,c_2} \right\| = \min_{d_2}\left(\left\| x_i - w_{2,c_1,d_2} \right\| \right) \tag{10}$$

- Finally, the *N-winner* $\boldsymbol{c}_N$ is picked from the N^{th} *PC*; u_N given in (7):

$$\left\| x_i - w_{N,c_1,c_2,\ldots,c_N} \right\| = \min_{d_N}\left(\left\| x_i - w_{N,c_1,c_2,\ldots,c_{N-1},d_N} \right\| \right) \tag{11}$$

Obviously, computation efforts (or steps) during FSOM recognition phase is $\boldsymbol{O(D_1 + D_2 + \ldots + D_N)}$. Of course, this amount is much less than that of the traditional N-dimension SOM in (2); which is $\boldsymbol{O(D_1 \times D_2 \times \ldots \times D_N)}$. According to the above scenario, FSOM consumes less computation time than traditional SOM during "recognition" phase; *for a simple case of the above algorithm please refer to* [1].

4. Experimental results

The author have been investigated the proposed algorithm using an artificial data set and two different lip-reading data sets as well [1]. Here we continue investigation by exploiting a large scale data set for face recognition across different poses and different lighting conditions.

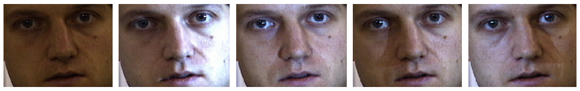
4.1 Input image overview

Face recognition evaluation reports indicate that the performance of many state-of-the-art face recognition methods deteriorates with changes in lighting, pose and so on. An input image including such constraints, especially if it is RGB type, requires a wider, or multi-dimension, feature space. It is demonstrated that, multi dimension feature space has been successfully exploited for complex applications such as face recognition across changing in pose and illumination conditions [8]. It is worthy to say that the added dimensionality provides more rooms to characterize more information.

Here, we utilized the PIE-CMU database which includes 41.386 colored images for 68 different subjects such that all images are across 13 different poses, 43 different illumination conditions and 3 facial expressions [7]. In the current stage of experiments we chose two different poses, one for training phase and the other for testing phase, such that all images of both phases are across 24 different lighting conditions. Figure 1 shows samples for the utilized images.

4.2 Time & accuracies

Here, we conduct a comparison between traditional SOM and FSOM regarding to feature extraction time and overall recognition accuracy.



(a) Samples for one subject across different lighting conditions

(b) Same subject across different pose and lightening conditions

Figure 1. (a) Training samples (b) Test samples for same subject: across 2 pose and 24 different illumination conditions.

Regarding feature extraction time, we tested more than one structure for the feature map of each approach up to two-dimensions. We found best accuracies are given if the map is in 3 dimensions, especially if it includes 16, 12 and 10 neurons, respectively, for each approach. Table 2 shows the time consumed by both approaches to achieve feature extraction in same computer. We started to process a set of 960 images and doubling the size gradually such that each time we calculate the feature extraction time consumed by each approach.

Table 2. Feature extraction time “second”

No. of images	**SOM**	**FSOM**
960	691.4	110.6
1920	1411.9	215

In Table 2, it is easy to remark how FSOM is faster than SOM in extracting features. Speedup ratio is around 6 times. As we increase the number of images or/and the number of dimensions of the feature map as the speedup ratio also increases [1]. Of course, computation time should not treat isolated from accuracy. Therefore, utilizing support vector machine [10] as a recognizer and under same experimental conditions, FSOM gives accuracy as 63.9% whereas SOM gives 51%.

The question now is: If the reduction in computation time, given in Table 2, is understood, how can one imagine this improvement occurs in accuracy?. According to the algorithm given in Table 1, it is easy to notice that FSOM’s learning phase accomplishes:

1. Training the traditional *N*-dimension SOM.
2. Then FSOM concentrates the most important information (or features) of the feature map and copy them to the neurons of the subspaces spanned by principal components.
3. Later on, another (or extra) training, for these subspaces only, is done.

Given that, the use of subspace modeling scenarios has significantly advanced face recognition performance [8]. In that sense, the last two steps granted the superiority of FSOM feature map over this of traditional SOM.

5. Conclusion

Continuing to our approach, Fast SOM, presented in [1], in this paper we investigated it using the PIE-CMU face recognition database. This database is devised across different poses and illumination conditions which require a wider feature space. Experimental results showed that FSOM consumes less feature extraction time than traditional SOM while preserving other qualities of SOM. Not only this but also FSOM showing better recognition accuracy than SOM, same as shown in [1] too. We are planning to continue development face recognition experiments by increasing the number of subjects and number of constraints on images.

6. References

[1] ____, “Fast Competition Approach using Self Organizing Map for Lip-Reading Applications,” Proc. of the IEEE Int. J Conf. on Neural Networks, IJCNN06, July 2006.

[2] Samuel Kaski, “Dimensionality Reduction by Random Mapping Fast Similarity Computation for Clustering,” Proc. of the IEEE Int. J Conf. on Neural Networks, IJCNN98, Anchorage, Alaska, May 1998.

[3] T. Kohonen, Self Organizing Maps, Springer series in information sciences, 3rd edition, 2001.

[4] S. Lawrence, C. Giles, A. Tsoi and A Back, “Face Recognition: A Convolutional Neural Network Approach,” The IEEE Trans. on Neural Networks, Special Issue on Neural Networks and Pattern Recog., 8(1), 1997, pp 98–113.

[5] I. Jolliffe, Principal Component Analysis. 2nd Ed, Springer, 2002.

[6] B. Lu and W. W. Hsieh, “Simplified nonlinear principal component analysis”, in Proc. Int. J. Conf. on Neural Networks IJCNN03, vol 1, July 2003, pp. 759 – 763.

[7] T. Sim, S. Baker and M. Bsat, “The CMU pose, illumination, and expression database,” The IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(12), 2003, pp1615–1618.

[8] Z. Stan and A. K. Jain (Ed), Handbook of Face Recognition, Springer; 1st edition, March 2005.

[9] L. Cole, D. Austin and L. Cole, “Visual Object Recognition using Template Matching,” Proc. of the 2004 Australasian Conference on Robotics & Automation, Canberra, Australia, Dec 2004.

[10] C. C. Chang and C. J. Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm