LETTER

Group Sparse Reduced Rank Tensor Regression for Micro-Expression Recognition*

Sunan LI[†,††], **Yuan ZONG**[†a)], **Cheng LU**[†,††], ***Nonmembers***, **Chuangan TANG**[†], ***Member***, ***and*** **Yan ZHAO**[††], ***Nonmember***

SUMMARY To overcome the challenge in micro-expression recognition that it only emerge in several small facial regions with low intensity, some researchers proposed facial region partition mechanisms and introduced group sparse learning methods for feature selection. However, such methods have some shortcomings, including the complexity of region division and insufficient utilization of critical facial regions. To address these problems, we propose a novel Group Sparse Reduced Rank Tensor Regression (GSRRTR) to transform the fearure matrix into a tensor by laying blocks and features in different dimensions. So we can process grids and texture features separately and avoid interference between grids and features. Furthermore, with the use of Tucker decomposition, the feature tensor can be decomposed into a product of core tensor and a set of matrix so that the number of parameters and the computational complexity of the scheme will decreased. To evaluate the performance of the proposed micro-expression recognition method, extensive experiments are conducted on two micro expression databases: CASME2 and SMIC. The experimental results show that the proposed method achieves comparable recognition rate with less parameters than state-of-the-art methods.

key words: *micro expression recognition, tensor, Tucker decomposotion, group sparse learning*

1. Introduction

In recent years, micro-expression recognition has been an active topic in the fields of computer vision and human-machine interaction. However, the micro-expression is hard to detect and classify due to it only emerges in several small facial regions and only lasts fewer than 0.2 second. Due to these difficulties, many researchers had investigated the region selection methods to suppress the influence of other facial regions. Inspired by Facial Action Coding System (FACS), Wang et al. designed Regions of Interests (ROIs) that crop the face based on the action units [1]. Furthermore, in [2] Zong et al. proposed a kernelized group sparse learning (KGSL) method to build the relationship between descriptors that extracted from hierarchical divided facial image. However, these methods laying texture features by turns in one dimension to construct feature matrices, which mixed information from different facial blocks and texture features of these blocks.

Based on the fact that different kind feature matrices can be stacked in different dimensions to make feature tensors, tensor is very useful in excavateing underlying structure of high dimensional feature [3]. In [4], Vasilescu et al. constructed tensor with two dimensions corresponding the views and illuminations respectively to excavate the multi-factor structure information of the feature tensor. Decomposition and sparse coding based method was proposed in [5] that extracted useful information from the origin feature tensor to reduce the number of parameters needed in the optimization step.

In this letter, we propose a novel Group Sparse Reduced Rank Tensor Regression (GSRRTR) model to construct feature tensor by laying grids and features in different dimensions. Benefit from the separation of selecting grids and features, these two different type features can be projected into the better feature spaces without the interference of another. Since the complexity of computing quadratic increased with the size of feature and number of blocks, we use Tucker decomposition to reduce the paramters. In this case, the GSRRTR model learns projection matrix from different dimension of feature tensor respectively and reduces the compution complexity based on the Tucker decomposition.

2. GSRRTR

2.1 GSLSR

To begin with, we will give a brief review of the Group Sparse Least-Squares Regression (GSLSR). Given R micro-expression video clip's hierarchical spatiotemporal descriptors that extracted according [6] as $X = [x_1, \cdots, x_R] \in R^{d\times R}$, where d represents the size of hierarchical spatio-temporal descriptors. And we use $Y = [y_1, \cdots, y_R] \in R^{c\times R}$ represents X's label, and the c is the number of classes in used database and y_R is a vector that depends on the sample R's true micro-expression. Intuitively, there could have a projection matrix to project the feature matrix into the label space. For this reason, we can construct the difference between label matrix and projected feature matrix by Eq. (1):

$$\min_U \left\|Y - U^T X\right\|_F^2, \tag{1}$$

Manuscript received September 5, 2022.
Manuscript revised December 5, 2022.
Manuscript publicized January 5, 2023.

[†]The authors are with the Key Laboratory of Child Development and Learning Science of Ministry of Education, Southeast University, Nanjing 210096, China.

[††]The authors are with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China.

*This work was supported in part by the NSFC under Grants U2003207 and 61921004 and in part by the Zhishan Young Scholarship of Southeast University.

a) E-mail: xhzongyuan@seu.edu.cn

DOI: 10.1587/transinf.2022EDL8073

where U^T is the proposed projection matrix. By partitioning the ordinary face into blocks, matrix $U^T X$ should be transformed into $\sum_{i=1}^{N} U_i^T X_i$ where X_i is the used facial block's texture descriptor matrix and U_i is the corresponded projection matrix of block X_i. Then the equation of (1) can be rewritten as:

$$\min_{U_i} \left\| Y - \sum_{i=1}^{N} U_i^T X_i \right\|_F^2. \tag{2}$$

Then, for selecting facial blocks that exactly associated with micro-expressions, the parameter β_i is introduced to quantify different facial blocks based on their importance. Since their contributions are probably different, we can impose l_1 norm regularization onto the objective function to eliminate the influence of most blocks during optimization. Therefore, the final GSLSR model can be written as follows:

$$\min_{U_i,\beta} \left\| Y - \sum_{i=1}^{N} \beta_i U_i^T X_i \right\|_F^2 + \mu \sum_{i=1}^{N} \beta_i \tag{3}$$

2.2 GSRRTR

Due to the QR decompose was used in the original GSLSR, the projection matrix U was decomposed into the product of two matrices G and U. Then the parameter β_i had been product with U to construct U_i. During the optimization procedure of U_i, the information of blocks and part of texture feature are mixed while the other part of texture feature G is not affected. This issue may influence the performance of learned projection matrix.Tensor-based algorithms are preferred in tickling multi-view features. This is the main motivation of extending ordinary GSLSR to a tensor version for tackling hierarchical block and texture descriptors respectively. For using tensor based method, the matrix X_i and U_i in formulation (3) can be written in a high-order fomulation $\mathcal{X} = [X_1, \cdots, X_R] \in R^{d' \times N \times R}$ and $\mathcal{U} \in R^{c \times d' \times N}$ where $X_R \in R^{d' \times N}$ represent the feature of sample R, d' is the dimension of hierarchical spatiotemporal descriptors and N is the number of facial clip's patchs that satisfied $d = d' \times N$. Then, we can rewritten the GSLSR formulation of (3) as:

$$\min_{U_i,\beta} \left\| Y - \sum_{i=1}^{N} \beta_i \mathcal{U}_{(1)i}^T \mathcal{X}_{(3)i} \right\|_F^2 + \mu \sum_{i=1}^{N} \beta_i. \tag{4}$$

The Tucker decomposition can reduce the rank of feature tensor and projection tensor. The tensor $\mathcal{X}$ and $\mathcal{U}$ in formulation (4) can be rewritten as the tensor product of core tensor and some projection matrix:

$$\mathcal{X} = CE_X \times_1 V_{X1} \times_2 V_{X2} \times_3 V_{X3};$$
$$\mathcal{U} = CE_U \times_1 V_{U1} \times_2 V_{U2} \times_3 V_{U3},$$

where $\times_n$ represent the product between a tensor and a matrix on nth dimension, $CE_X \in R^{M1 \times M2 \times M3}$ and $CE_U \in R^{M1 \times M2 \times M3}$ are the core tensor of the decomposition, $V_{(X)N}$ and $V_{(U)N}$ represent the projection matrix on the n-order of the feautre tensor and original projection tensor. The final optimization formulation is getted by substituting $\mathcal{X}$ and $\mathcal{U}$ into the original GSLSR instead of X_i and U_i in (3):

$$\min_{U_i,\beta} \left\| Y - \sum_{i=1}^{N} \beta_i V_{U1} \times CE_{U(1)} \times A_1 \times A_2 \times CE_{X(3)}^T \times V_{X3}^T \right\|_F^2 + \mu \sum_{i=1}^{N} \beta_i. \tag{5}$$

Where $A_1 = V_{U2} \otimes V_{U3}$, $A_2 = V_{X1} \otimes V_{X2}$, then we can translate the formultion (5) into:

$$\min_{U_i,\beta} \left\| Y - \sum_{i=1}^{N} \beta_i V_{U1} \times CE_{U(1)} \times A_3 \times A_4 \times CE_{X(3)}^T \times V_{X3}^T \right\|_F^2 + \mu \sum_{i=1}^{N} \beta_i. \tag{6}$$

Where $A_3 = V_{U2} \otimes I$, $A_4 = V_{X1} \otimes V_{U3} \times V_{X2}$, so we can translate the loss function of GSRRTR into the format of original GSLSR problem. To achieve this goal, we follow the definition of parameters in formulation (3), let:

$$G = V_{U1} \times CE_{U(1)};$$
$$P = V_{U2} \otimes I;$$
$$Z = (V_{X1} \otimes V_{U3} \times V_{X2}) \times CE_{X(3)}^T \times V_{X3}^T.$$

Based on the construction of Eq. (4) and the definition of Tucker decomposition in different dimensions we can get different loss functions due to the Tucker decomposition in different dimensions of feature tensor, then the final loss function can be rewritten as the sum of two losses:

$$\begin{aligned} L = &\min_{U_i,P_i} \left\| Y - \sum_{i=1}^{d} G_1 \times P_1 \times Z_1 \right\|_F^2 + \lambda_1 \|U_1\|_1 \\ &+ \left\| Y - \sum_{i=1}^{N} G_2 \times P_2 \times Z_2 \right\|_F^2 + \lambda_2 \|U_2\|_1 \\ &\text{s.t. } P_1 = U_1;\ P_2 = U_2. \end{aligned} \tag{7}$$

Then, based on the optimization of GSLSR, the final optimization problem can be obtained, which aims to minimize the Lagrangian function using ALM method:

$$\begin{aligned} &\mathcal{L}(P_1, P_2, U_1, U_2, \lambda_1, \lambda_2, u_1, u_2) = L + Tr[T_1(P_1 - U_1)] \\ &+ \frac{u_1}{2} \|P_1 - U_1\|_F^2 + Tr[T_2(P_2 - U_2)] + \frac{u_2}{2} \|P_2 - U_2\|_F^2. \end{aligned} \tag{8}$$

Where T_1 and T_2 are Lagrange multipliers, u_1 and u_2 are regularized parameters. Finally the optimal P_1, P_2 and U_1, U_2 can be learned by optimizing the Lagrangian function (8) iteratively until all convergence conditions are satisfied. The procedures for P_1's optimization are given in Algorithm 1 as

Algorithm 1 Procedure for learning the optimal parameter U_n of GSRRTR.

Input: feature matrix of training set X, label matrix of training set L, trade-off parameter λ_n u_n, order of Tucker decomposition M_n

Output: Ensemble of classifiers on the current batch, E_n;

1: Fix T_1 and U_1 and update P_1:

$$P_1 = \left(\frac{2ZZ^T}{\mu_1} + I\right)^{-1}\left(\frac{2ZX^TG_1 - T_1}{\mu_1} + U_1\right)$$

in the formulation matrix I represent the identity matrix

2: Fix T_1 and P_1 and update U_1 [7]:

$$U_1 = \frac{\left\|P_1 + \frac{T_1}{\mu_1}\right\|_F - \frac{\lambda_1}{\mu_1}}{\left\|P_1 + \frac{T_1}{\mu_1}\right\|_F}\left(P_1 + \frac{T_1}{\mu_1}\right)$$

3: Update T_1:

$$T_1 = T_1 + \rho_1(P_1 - U_1)$$

Where ρ_1 is a scaled parameter

4: Update the parameter u_1;

$$u_1 = \min(\rho_1 u_1, u_{\max})$$

5: Repeat step 1–4 for feature tensor and projection tensor in dimension 2 to optimize P_2, U_2

6: Check whether convergence condition is satisfied;

$$\|P_1 - U_1\|_\infty + \|P_2 - U_2\|_\infty \le \varepsilon$$

7: **return** P_1, P_2;

a example.

2.3 Complexity Analysis

In ordinary GSLSR, the computational complexity of computing the product of projection matrix and feature matrix is $O(NM)$ where N is the number of grids and M is the size of features from every grids. While the computational complexity of the GSRRTR is $O(M_1 + M_2 + N_1 + N_2)$. Since the dimension M_n and N_n used in the Tucker decomposition is usually much smaller than the dimension of ordinary matrix M and N, the overall computational cost of GSRRTR is much smaller than the GSLSR based method.

3. Experiment

3.1 Experimental Setup

In this section, extensive experiments are conducted to evaluate the performance of the proposed GSRRTR. Two widely-used micro-expression databases are adopted, i.e. CASME II [8] and SMIC [9]. CASME II collected 247 micro-expression video clips from 26 subjects. Each of these video clips is labeled by one of five micro-expressions classes, i.e., Happy, Surprise, Disgust, Repression, and Others. SMIC consists of 16 subjects and 164 facial video clips involving three different micro expressions classes, i.e. Positive, Negative, and Surprise. In the experiments, we choose the STLBP-IP [6] to describe the facial video clips. And we choose unweighted average recall (UAR), which can be calculated by [10], as the main performance metric due to its robustness to the sample class imbalance. Besides, the weighted average recall (WAR) is also adopted to serve as the secondary performance metric. For the experiments of using both micro-expression databases, leave-one-subject-out (LOSO) protocol is used to calculate the performance metric. In addition, the trade-off parameters corresponding to the best UAR in the fold of the first subject are selected in the experiments on each database, which means that the trade-off parameters are fixed throughout the following folds.

Table 1 WAR and UAR results on the CASME2 database, where the best results are highlight in bold.

Comparison Methods	Accuracy (%)	
	WAR	UAR
LBP-SIP+SVM [2]	40.89	/
STLBP-IP+SVM [6]	55.47	/
CNN + LSTM Model [11]	60.98	/
LGCcon [12]	62.14	**59.00**
GSLSR [13]	57.09	49.77
GSRRTR (ours)	**63.56**	57.24

Table 2 WAR and UAR results on the SMIC database, where the best results are highlight in bold.

Comparison Methods	Accuracy (%)	
	WAR	UAR
LBP-SIP+SVM [2]	47.56	/
STLBP-IP+SVM [6]	52.44	/
CNN + LSTM Model [11]	62.97	/
LGCcon [12]	**63.41**	62.00
GSLSR [13]	53.05	53.87
GSRRTR (ours)	60.37	**64.10**

3.2 Results and Analysis

The experimental results are reported in Tables 1 and 2 respectively. From the results, it is clear that GSRRTR obtains the comparable performance among all the methods. Compared with other methods (including CNN based and subspace based methods), we can see that GSRRTR achieves significant improvement over the baseline methods in most cases, especially in SMIC. In SMIC, our method obtains 64.10% (UAR) and 60.37% (WAR) recognition accuracies, which have the increases of 2.10% (UAR) over LGCcon and 7.30% (WAR) / 10.23% (UAR) over original GSLSR method respectively. The best result was obtained when the λ_n was fixed at 1 and u_n was fixed at 2 in CASME II, λ_n was fixed at 1 and u_n fixed at 3 in SMIC. The parameters was select from preset paramters grids, i.e. M_1 range from 100 to 230 with step size 10: $M_1 \in [100 : 10 : 230]$ and M_2 range from 30 to 80 with step size 10: $M_2 \in [30 : 10 : 80]$.

It is interesting to see that, in contrast to the result in

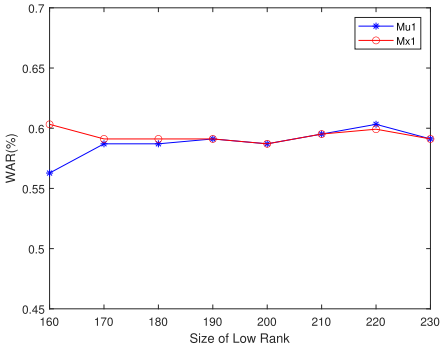

Fig. 1 Result of different rank for GSRRTR.

SMIC, the UAR and WAR of GSRRTR in CASME II are much worse than compared methods. This is most likely due to the unbalance problem of the labeled data samples in each class of database. In SMIC, we can find that GSRRTR outperform other algotithm in UAR rather than WAR, that, GSRRTR perform well with smaller size samples and class balanced database. While the unbalnce problem between different calsses is much serious in CASME II. In previous works in [14] and [15], we can find that LSR, which is the fundamental part of GSRRTR, could be viewed as a simplified version of linear discriminant analysis (LDA) under the assumption that the numbers of data samples in each class are approximately equal. If the numbers of the labeled data samples in each class are largely different, the LSR model will deviate from the LDA model and hence the discriminant ability of LSR would degraded. In addition, it is clear to see that, compared with SMIC, where the classes are balanced, the performance of all LSR based methods including KGSL and GSLSR decreases clearly in CASME II.

And we also investigate the parameter sensitivity of the proposed GSRRTR method. To see whether GSRRTR model with different ranks of Tucker decomposition is robust, we conduct extensive experiments on CASME II. In the experiment we fix the rank of dimension in number of blocks in GSRRTR while changing the rank of texture descriptor. More specifically, the value range of M_1 are set as $[100 : 10 : 230]$. Experimental results are given in Fig. 1, where the M_U1 corresponds to the M_1 of project tensor U and M_X1 corresponds to the M_1 of feature tensor X. From the results, we can see that the performance of GSRRTR changes slightly with different value of M_U1 and M_X1. The result demonstartes that the proposed GSRRTR method is robust to its change in low rank parameters.

4. Conclusion

In this letter, we have designed a tensor based group sparse learning scheme called GSRRTR to better describe micro-expressions and reduce the interference between different facial blocks. By the extensive experiments, the results show that our method can effectively deal with micro-expression classification tasks with fewer parameters.

References

[1] S.-J. Wang, W.-J. Yan, X. Li, G. Zhao, C.-G. Zhou, X. Fu, M. Yang, and J. Tao, "Micro-expression recognition using color spaces," IEEE Trans. Image Process., vol.24, no.12, pp.6034–6047, 2015.
[2] Y. Zong, X. Huang, W. Zheng, Z. Cui, and G. Zhao, "Learning from hierarchical spatiotemporal descriptors for micro-expression recognition," IEEE Trans. Multimedia, vol.20, no.11, pp.3160–3172, 2018.
[3] N.D. Sidiropoulos and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," IEEE Signal Process. Lett., vol.19, no.11, pp.757–760, 2012.
[4] M.A.O. Vasilescu and D. Terzopoulos, "TensorTextures: Multilinear image-based rendering," ACM Trans. Graph., vol.23, no.3, pp.336–342, 2004.
[5] N. Qi, Y. Shi, X. Sun, and B. Yin, "TenSR: Multi-dimensional tensor sparse representation," IEEE Conference on Computer Vision and Pattern Recognition, pp.5916–5925, 2016.
[6] X. Huang, S.-J. Wang, G. Zhao, and M. Piteikäinen, "Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection," 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), pp.1–9, 2015.
[7] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," SIAM Journal on Imaging Sciences, vol.2, no.2, pp.569–592, 2009.
[8] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, and X. Fu, "CASME II: An improved spontaneous micro-expression database and the baseline evaluation," Plos One, vol.9, no.1, e86041, 2014.
[9] X. Li, T. Pfister, X. Huang, G. Zhao, and M. Pietikäinen, "A spontaneous micro-expression database: Inducement, collection and baseline," 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2013.
[10] B. Schuller, B. Vlasenko, F. Eyben, M. Wöllmer, A. Stuhlsatz, A. Wendemuth, and G. Rigoll, "Cross-corpus acoustic emotion recognition: Variances and strategies," IEEE Trans. Affect. Comput. vol.1, no.2, pp.119–131, 2010.
[11] D.H. Kim, W.J. Baddar, and Y.M. Ro, "Micro-expression recognition with expression-state constrained spatio-temporal feature representations," MM '16, Proc. 24th ACM International Conference on Multimedia, pp.382–386, 2016.
[12] Y. Li, X. Huang, and G. Zhao, "Joint local and global information learning with single apex frame detection for micro-expression recognition," IEEE Trans. Image Process., vol.30, pp.249–263, 2021.
[13] W. Zheng, "Multi-view facial expression recognition based on group sparse reduced-rank regression," IEEE Trans. Affect Comput., vol.5, no.1, pp.71–85, 2014.
[14] W. Zheng and X. Zhou, "Cross-pose color facial expression recognition using transductive transfer linear discriminat analysis," IEEE International Conference on Image Processing, 2015.
[15] Y. Zong, W. Zheng, X. Huang, J. Yan, and T. Zhang, "Transductive transfer LDA with Riesz-based volume LBP for emotion recognition in the wild," ICMI '15, Proc. 2015 ACM on International Conference on Multimodal Interaction, pp.491–496, 2015.