Supplementary Material for Bilevel Model Based Discriminative Dictionary Learning for Recognition

Pan Zhou, Chao Zhang, *Member, IEEE*, and Zhouchen Lin, *Senior Member, IEEE*

I. Details of the closed-form solutions in Algorithm 1

In this section, we will present how to mathematically deduce the closed-form solutions for updating W, Z, M, X, and S in Algorithm 1. Actually, by minimizing $\mathcal{L}_2$ we can update these variables in the following way.

$$W = \arg\min_{W} \|H - WPZ\|_F^2 + \lambda\|W\|_F^2 = HZ^TP^T(PZZ^TP^T + \lambda I)^{-1}. \tag{1}$$

$$\begin{aligned} Z =& \arg\min_{Z} \|H - WPZ\|_F^2 + \left\langle R_1, P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\right\rangle + \left\langle R_3, P^TPZ - X\right\rangle + \langle R_4, Z - S\rangle \\ &+ \frac{\mu}{2}\|P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\|_F^2 + \frac{\mu}{2}\left(\|P^TPZ - X\|_F^2 + \|Z - S\|_F^2\right) \\ =& \left(2P^TW^TWP + 2\mu P^TD^TDD^TDP + 2\mu P^TP + \mu I\right)^{-1}\left[2P^TWH - \mu P^TD^TDP(-P^TD^TY + \alpha E + \beta XL \right. \\ & \left. + M + R_1/\mu) - P^TP(R_3 - \mu X) - R_4 + \mu S\right]. \end{aligned} \tag{2}$$

$$\begin{aligned} M =& \arg\min_{M\leq 0} \langle R_2, M \odot S\rangle + \frac{\mu}{2}\|M \odot S\|_F^2 + \left\langle R_1, P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\right\rangle \\ &+ \frac{\mu}{2}\|P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\|_F^2 \\ =& -\Theta((S \odot R_2/\mu + P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + R_1/\mu) \oslash (S \odot S + E)), \end{aligned} \tag{3}$$

where $\Theta(\cdot)$ is an operator that projects a matrix onto the nonnegative cone, which can be defined as follows:

$$\Theta(X_{ij}) = \begin{cases} X_{ij}, & \text{if } X_{ij} \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$\begin{aligned} X =& \arg\min_{X} \left\langle R_3, P^TPZ - X\right\rangle + \frac{\mu}{2}\|P^TPZ - X\|_F^2 + \left\langle R_1, P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\right\rangle \\ &+ \frac{\mu}{2}\|P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\|_F^2 \\ =& \left[P^TPZ + R_3/\mu - \beta(P^TD^TDPZ - P^TD^TY + \alpha E + M + R_1/\mu)L^T\right]\left(\beta^2LL^T + I\right)^{-1}. \end{aligned} \tag{5}$$

$$S = \arg\min_{S\geq 0} \langle R_2, M \odot S\rangle + \langle R_4, Z - S\rangle + \frac{\mu}{2}\|M \odot S\|_F^2 + \frac{\mu}{2}\|Z - S\|_F^2 = \Theta\left((Z + R_4/\mu - M \odot R_2/\mu) \oslash (M \odot M + E)\right). \tag{6}$$

$$\begin{aligned} D =& \arg\min_{D\in\Omega} \left\langle R_1, P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\right\rangle + \frac{\mu}{2}\|P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M\|_F^2 \\ =& \arg\min_{D\in\Omega} \psi(D), \end{aligned} \tag{7}$$

where $\Omega = \{D \mid \|D_i\|_2^2 \leq 1, i = 1, \cdots, k\}$ and

$$\psi(D) = \|P^TD^TDPZ - P^TD^TY + \alpha E + \beta XL + M + R_1/\mu\|_F^2. \tag{8}$$

P. Zhou, C. Zhang, and Z. Lin are with Key Lab. of Machine Perception (MoE), School of EECS, Peking University, P. R. China. C. Zhang and Z. Lin are also with Cooperative Medianet Innovation Center, Shanghai, China. C. Zhang is the corresponding author. Emails: pzhou@pku.edu.cn, chzhang@cis.pku.edu.cn, and zlin@pku.edu.cn.

II. Details of the closed-form solutions in Algorithm 2

In this section, we will present the mathematically deduction for updating the variables A, J, and G in Algorithm 2. Also, we can update these three variables in turn by minimizing the corresponding Lagrangian function $\mathcal{L}_3$.

$$A = \arg\min_{A} \alpha\|A\|_1 + \langle A - J, R_5\rangle + \frac{\mu}{2}\|A - J\|_F^2 + \langle A - G, R_6\rangle + \frac{\mu}{2}\|A - G\|_F^2 = \mathcal{S}_{\alpha/\mu}\left(\frac{1}{2}(J + G - (R_5 + R_6)/\mu)\right), \tag{9}$$

where $\mathcal{S}_\epsilon(x) = \text{sgn}(x)\text{max}(|x| - \epsilon, 0)$ is the hard thresholding operator [1],

$$J = \arg\min_{J} \frac{1}{2}\|Y - DJ\|_F^2 + \langle A - J, R_5\rangle + \frac{\mu}{2}\|A - J\|_F^2 = V_D\left(\Sigma_D^T\Sigma_D + \mu I\right)^{-1} V_D^T\left(D^TY + \mu A + R_5\right), \tag{10}$$

$$G = \arg\min_{G} \frac{\beta}{2}\text{tr}(GLG^T) + \langle A - G, R_6\rangle + \frac{\mu}{2}\|A - G\|_F^2 = (\mu A + R_6)V_L\left(\Sigma_L + \mu I\right)^{-1} V_L^T, \tag{11}$$

where $U_D\Sigma_D V_D^T$ and $V_L\Sigma_L V_L^T$ are the full SVD of D and $\beta(L + L^T)/2$, respectively.

III. Connections with Neural Networks

There are connections between BMDDL and supervised neural networks [2], [3], [4]. Both BMDDL and supervised neural networks are task-driven feature learning schemes. In recognition tasks, minimizing the classification loss is the final task. So BMDDL and neural networks adopt it as their optimization goal. They can be formulated as a general multi-level model:

$$\begin{aligned}
&\min_{\{W^i\},\{A^i\}} \sum_{i=1}^{n} \Phi(h_i, f(A_i^{m-1}, W^m)),\\
&\text{s.t. } A^{m-1} = \arg\min_{A} G^{m-1}(A^{m-2}, W^{m-1}, A),\\
&\qquad\quad \text{s.t. } A^{m-2} = \arg\min_{A} G^{m-2}(A^{m-3}, W^{m-2}, A),\\
&\qquad\qquad\qquad \cdots\cdots\\
&\qquad\qquad\qquad\quad \text{s.t. } A^{1} = \arg\min_{A} G^{1}(A^{0}, W^{1}, A),
\end{aligned} \tag{12}$$

where $G^i(A^{i-1}, W^i, A)$ is a feature extractor, W^i are its parameters, and A^i is feature of the ith level extracted by $G^i(A^{i-1}, W^i, A)$ (A^0 is the input). $\Phi(h_i, f(A_i^{n-1}, W^m))$ is a classification loss function. $f(A_i^{m-1}, W^m)$ is a classifier, such as multinomial logistic regression or a linear classifier. A_i^{m-1} is the finally extracted feature for the ith sample. $H = [h_1, \cdots, h_n]$ is the label of A^{m-1}, in which n is the number of training samples.

In BMDDL, the lower level optimization problem (5) in our paper can be regarded as the first level, which extracts group sparse feature from training samples and feeds them into the second level, i.e., a classification loss function $\Phi(\cdot)$. In this way, BMDDL is only a two-level feature learning network. The reason why BMDDL can only stack two levels is that its feature extractor $G^i(A^{i-1}, W^i)$ is too complex. This is the very difference between BMDDL and neural networks. In BMDDL, $A^i = \arg\min_A G^i(A^{i-1}, W^i, A)$ has no closed-form solution and we have to solve it to obtain A^i by iterative algorithms. If we stack K ($K \geq 3$) levels, the optimization problem will be too difficult to solve. In contrast, in neural networks the new feature can be directly obtained, since we usually set $A^i = \arg\min_A G^i(A^{i-1}, W^i, A) = \arg\min_A \|A - \Psi(W^iA^{i-1})\|_F^2 = \Psi(W^iA^{i-1})$, where $\Psi(\cdot)$ is an activation function. Then we can use the backpropagation algorithm (based on the chain rule) to update the parameters W^i in turn. Thus, both BMDDL and neural networks are task-driven feature learning methods. But, due to the optimization difficulty, BMDDL can only be a network with two levels.

IV. Effectiveness of the Laplacian Term

In this section, we conduct experiments to verify the advantages of the Laplacian term. In our paper, we have compared our method with TDDL [5]. As we have mentioned, TDDL is also a bilevel model based dictionary learning method, but it does not consider the intrinsic data structure. Actually, TDDL [5] replaces the Laplacian term tr(ALA^T) in problem (4) with a regularization $\|A\|_F^2$, which results in a subproblem that is easier to solve for the subgradient with respect to D via implicit differentiation. From Tables 3∼8 in the paper, by comparison, we can see that our method outperforms TDDL on the six testing datasets.

To further demonstrate the contribution of the group sparsity constraint, we discard the Laplacian term in our model (setting $\beta = 0$) and add a regularization $\|A\|_F^2$ to enhance the convexity of the lower level problem. To accommodate this change, we only need to set $L = I$ in problem (4) in our paper, where I is the identity matrix. Now, our model is the same as the model in TDDL [5]. Then we first replace the lower level with its KKT conditions and then apply ADM to solve the new model. We also use the same initialization strategy in our paper. Namely, we first use KSVD [6] to initialize D, then solve

TABLE I
THE EFFECTS TO RECOGNITION RATES (%) OF THE LAPLACIAN TERM ON THE THREE DATABASES.

Method	Extended YaleB [7]	15 Scene Categories [8]	Caltech 101 [9]
TDDL [5]	94.6	92.1	71.5
Our method without the Laplacian term	94.8	92.9	71.8
Our method with the Laplacian term	**95.5**	**96.9**	**75.5**

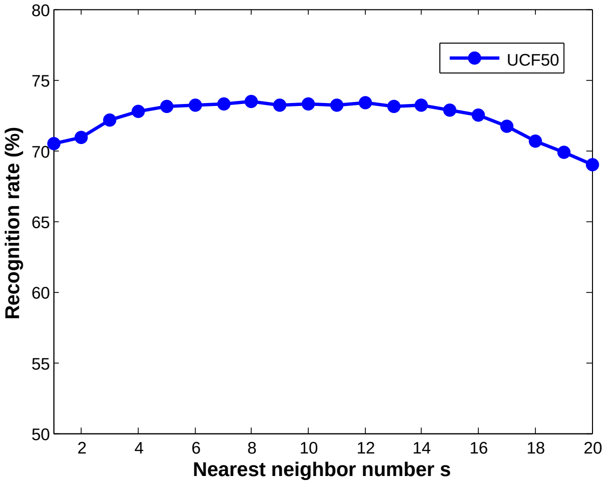


Fig. 1. Effects of the nearest neighbor number s of a testing sample on UCF50.

the lower level problem to initialize other variables. We report the experimental results in Table I. The experimental settings in this section are as described in corresponding subsections in the paper, respectively. We can see that our original BMDDL can achieve better recognition performance than the BMDDL without the Laplacian term, which demonstrates the benefits of the Laplacian term. From Table I, we can also see that when the models are the same, our method without the Laplacian term only outperforms TDDL slightly. Thus, the improvements of recognition rates are mainly achieved by the Laplacian term. But from Table 9 in the paper, our method is much faster than TDDL, which employs the stochastic subgradient descent algorithm to solve its model. Thus, our optimization method is more efficient in speed than the stochastic subgradient descent algorithm.

V. Effects of the nearest neighbor number of a testing sample

In this section, we conduct experiments on UCF50 to verify the effects of the nearest neighbor number s of a testing sample. We run our method with the nearest neighbor number s varying from 1 to 20 with increment 1. The experimental settings except s in this section are as described in Sections V.D in the paper. Fig. 1 displays the experimental results. We can see that the nearest neighbor number s can effect the recognition rate. But when s is in a range, such as from 4 to 15, the recognition rates are relatively stable. Thus, we only need to select s from this range. The reason why recognition rate drops when s increases from 15 to 20 is that the found s nearest neighbors may not be correct, i.e., not all nearest neighbors are from the class of the testing sample, leading to performance degradation. Actually, for fairness, we follow the experimental setting in [10] and set $s = 5$ in all our experiments, which works well. So it is unnecessary to tune s very carefully.

VI. Confusion matrices of our method on 15 Scene Categories, UCF50 and HMDB51

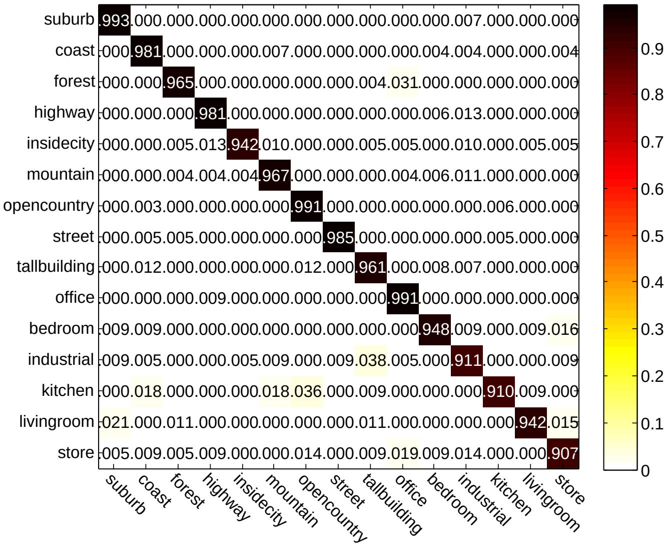
In this section, we report the average recognition rates for each class on 15 Scene Categories, UCF50 and HMDB51 datasets by using confusion matrices. The experimental settings in this section are as described in Sections V.B and V.D in the paper, respectively.

The confusion matrix of our method on the 15 Scene Categories dataset can be seen in Fig. 2, where the average recognition rates for each class are along the diagonal. There is no class that are classified badly and the worst recognition rate is as high as 90.7% on 15 Scene Categories.

The confusion matrices for UCF50 and HMDB51 are shown in Fig. 3 and 4. In Fig. 3, there are no stand-out confusion in the diagonal position. Only two categories (Pizza Tossing and Walking with Dog) obtain relatively low classification rates. However, due to its richer movement types and more complex illumination conditions, HMDB51 is much more difficult than UCF50. So the recognition rates on HMDB51 are lower than those on UCF50.

References

[1] S. Foucart, “Hard thresholding pursuit: An algorithm for compressive sensing,” *SIAM J. on Numerical Analysis*, vol. 49, no. 6, pp. 2543–2563, 2011.
[2] Y. LeCun, L. Bottou, G. Orr, and K. Müller, “Efficient backprop,” in *Neural Networks: Tricks of the Trade*. Springer, 1998, pp. 9–48.



	suburb	coast	forest	highway	insidecity	mountain	opencountry	street	tallbuilding	office	bedroom	industrial	kitchen	livingroom	store
suburb	.993	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.007	.000	.000	.000
coast	.000	.981	.000	.000	.000	.007	.000	.000	.000	.000	.004	.004	.000	.000	.004
forest	.000	.000	.965	.000	.000	.000	.000	.000	.004	.031	.000	.000	.000	.000	.000
highway	.000	.000	.000	.981	.000	.000	.000	.000	.000	.000	.006	.013	.000	.000	.000
insidecity	.000	.000	.005	.013	.942	.010	.000	.000	.005	.005	.000	.010	.000	.005	.005
mountain	.000	.000	.004	.004	.004	.967	.000	.000	.000	.004	.006	.011	.000	.000	.000
opencountry	.000	.003	.000	.000	.000	.000	.991	.000	.000	.000	.000	.000	.006	.000	.000
street	.000	.005	.005	.000	.000	.000	.000	.985	.000	.000	.000	.000	.005	.000	.000
tallbuilding	.000	.012	.000	.000	.000	.000	.012	.000	.961	.000	.008	.007	.000	.000	.000
office	.000	.000	.000	.009	.000	.000	.000	.000	.000	.991	.000	.000	.000	.000	.000
bedroom	.009	.009	.000	.000	.000	.000	.000	.000	.000	.000	.948	.009	.000	.009	.016
industrial	.009	.005	.000	.000	.005	.009	.000	.009	.038	.005	.000	.911	.000	.000	.009
kitchen	.000	.018	.000	.000	.000	.018	.036	.000	.009	.000	.000	.000	.910	.009	.000
livingroom	.021	.000	.011	.000	.000	.000	.000	.000	.011	.000	.000	.000	.000	.942	.015
store	.005	.009	.005	.009	.000	.000	.014	.000	.009	.019	.009	.014	.000	.000	.907

Fig. 2. The confusion matrix of our method on the Fifteen Scene Categories database. The average classification rates of each class are along the diagonal. The entry in the ith row and the jth column is the ratio of images from class i that are misidentified as class j.

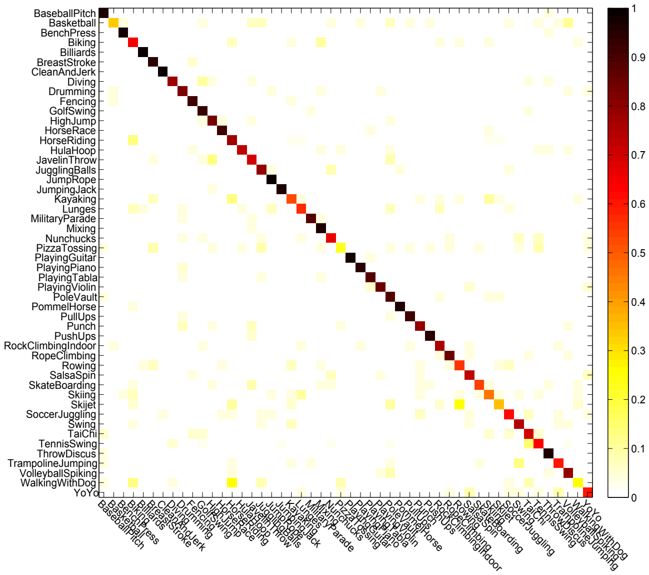


Fig. 3. The confusion matrix of our method on the UCF50 database. The classification rates are not shown. The color legend is drawn on the right, **best viewed in color**.

[3] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[5] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[7] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[8] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[9] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[10] S. Gao, I. Tsang, and L. Chia, "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 92–104, 2013.

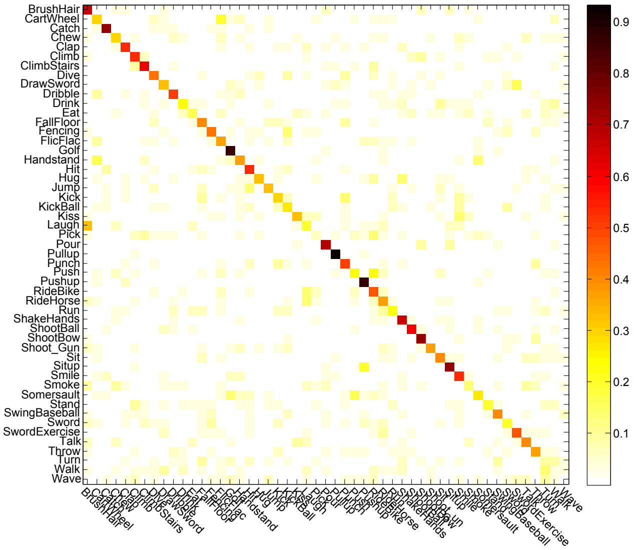


Fig. 4. The confusion matrix of our method on the HMDB51 database. The classification rates are not shown. The color legend is drawn on the right, **best viewed in color**.