Supplementary Materials

1. Proof of Proposition 3.1

Without loss of generality, we may assume $\|D_i\| = 1$ for all i. From the definition of Φ, we know

$$\|\Phi(D_i)\|_2^2 = \langle \Phi(D_i), \Phi(D_i) \rangle = \psi(0),\ \forall i,$$

and

$$\langle \Phi(D_i), \Phi(D_j) \rangle = \psi(2 - 2\mu_0),\ \forall i \neq j.$$

We complete the proof by noting $c_0 = \sqrt{\psi(0)}$ and $\eta = \psi(2 - 2\mu_0)$.

2. Proof of Proposition 3.5

Since $H(C, D) = \frac{1}{2}\mathrm{Tr}(C^\top QC - 2K(D, Y)^\top C)$ and $k(x, y) = \exp(-\|x - y\|_2^2/2\sigma^2)$, we have

$$\begin{aligned} &\nabla_C H(C, D) = QC - K(D, Y), \\ &\nabla_{D_\ell} H(C, D) = \sum_{i=1}^{n} a_{\ell i}(D_\ell - Y_i),\ \forall \ell, \end{aligned} \tag{1}$$

where $a_{\ell i} = -\frac{1}{\sigma^2} C_{\ell i} \exp\big(-\frac{\|D_\ell - Y_i\|_2^2}{2\sigma^2}\big)$.

As $\nabla_C^2 H(C, D) = Q$ implies that $\nabla_C H(C, D)$ is Lipschitz with modulus $\lambda_{max}(Q)$, where $\lambda_{max}(Q)$ is the maximal eigenvalue of Q. Moreover, the Hessian matrix $\nabla_{D_\ell}^2 H(C, D)$ is given by

$$\sum_{i=1}^{n} a_{\ell i}\big(I - \frac{1}{\sigma^2}(D_\ell - Y_i)(D_\ell - Y_i)^\top\big).$$

By the fact $(1 - \|y\|_2^2)^2 \leq \|d - y\|^2 \leq (1 + \|y\|_2^2)^2$ for any $\|d\|_2 = 1$, we have $|a_{\ell i}| \leq \frac{1}{\sigma^2}|C_{\ell i}| \exp(-\frac{(1 - \|Y_i\|_2^2)}{2\sigma^2})$ and the maximal eigenvalue is bounded by $1 + \frac{1}{\sigma^2}\|D_\ell - Y_i\|_2^2 \leq 1 + \frac{1}{\sigma^2}(1 + \|Y_i\|_2^2)^2$. Thus, the maximal eigenvalue of $\nabla_{D_\ell}^2 H(C, D)$ is bounded by $L(C_\ell)$ which is defined as

$$\sum_{i=1}^{n} \tfrac{1}{\sigma^2}|C_{\ell i}| \exp(-\tfrac{1 + \|Y_i\|_2^2}{2\sigma^2})(1 + \tfrac{1}{\sigma^2}(1 + \|Y_i\|_2^2)^2). \tag{2}$$

3. Numerical Algorithm for The Supervised Equiangular Kernel Sparse Coding Problem (16)

Recall that the supervised extension of our equiangular kernel dictionary learning method is formulated as the following minimization model:

$$\begin{aligned} \min_{D \in \mathcal{D}, C \in \mathcal{C}, W} &\tfrac{1}{2}\mathrm{Tr}(C^\top QC - 2K(D, Y)^\top C) \\ &+ \tfrac{\beta}{2}\|L - WC\|_F^2 + \tfrac{\alpha}{2}\|W\|_F^2, \end{aligned} \tag{3}$$

where $\mathcal{C} = \{C : \|C\|_\infty \leq M,\ \|C_z\|_0 \leq T, \forall z\}$ and $\mathcal{D} = \{D : D^\top D = DD^\top = I\}$. We give the detailed algorithm for solving (3) as follows. Define

$$\begin{aligned} &H(C, D, W) = \tfrac{1}{2}\mathrm{Tr}(C^\top QC - 2K(Y, D)^\top C) + \tfrac{\beta}{2}\|L - WC\|_F^2, \\ &F(C) = \delta_{\mathcal{C}}(C),\ G(D) = \delta_{\mathcal{D}}(C), E(W) = \tfrac{\alpha}{2}\|W\|_F^2. \end{aligned}$$

Then the sparse code C, dictionary D and classifier W are updated by the following proximal alternating scheme.

1. Kernel sparse coding. When the dictionary D and the classifier W are fixed, we update the sparse code C via solving:

$$C^{j+1} \in \operatorname*{argmin}_C F(C) + \tfrac{s^j}{2}\|C - U^j\|_F^2, \tag{4}$$

where $U^j = C^j - \nabla_C H(C^j, D^j, W^j)/s^j$ and s^j is some positive step size. This subproblem has a closed-form solution given by

$$C^{j+1} = \mathrm{sign}(U^j) \odot \mathrm{argmin}(H_T(|U^j|), M), \tag{5}$$

where $H_T(X)$ keeps the largest T entries in each column of X and sets others to zero.

2. Dictionary update. When the sparse code C and the classifier W are fixed, the update of dictionary D is the same as that in the unsupervised version, *i.e.* we update the dictionary D by solving

$$D^{j+1} \in \operatorname*{argmin}_D G(D) + \tfrac{t^j}{2}\|D - V^j\|_F^2, \tag{6}$$

where $V^j = D^j - \nabla_D H(C^{j+1}, D^j, W^j)/t^j$ and t^j is some positive step size. This problem (6) has a closed-form solution given by the Proposition. 3.4 in our paper.

3. Classifier update. When the dictionary D and sparse code C are fixed, we update W via solving the following minimization:

$$\operatorname*{argmin}_W \tfrac{\beta}{2}\|L - WC^j\|_F^2 + \tfrac{\alpha}{2}\|W\|_F^2 + \tfrac{p^j}{2}\|W - W^j\|_F^2, \tag{7}$$

where $p^j > 0$. The solution of (7) is given by

$$W^{j+1} = (\beta L C^{j\top} + p^j W^j)(\beta C^j C^{j\top} + (\alpha + p^j) I)^{-1}. \quad (8)$$

Setting of step size. The three step sizes p^j, s^j, t^j are set as follows. Since $\|W\|_F^2$ has coercive property, we know W^j is a bounded sequence and the maximal eigenvalue of $Q + W^{j\top} W^j$ is defined by λ_{max}^j and $\lambda_{max} = \max_j(\lambda_{max}^j)$. Given $\gamma_j > 1$, $0 < a < b$ and $0 < c < d$ such that $b > \lambda_{max}$ for all j and $d > L_{max}$, where $L_{max} = \max\{L(C_\ell) : \ell = 1, 2, \ldots, m, C \in \mathcal{C}\}$ and $L(C_\ell)$ is defined in (2).

$$s^j = \max(\min(\gamma_j \lambda_{max}^j, b), a), \quad (9a)$$

$$t^j = \max(\min(\gamma_j L(C^{j+1}), d), c), \quad (9b)$$

$$p^j \in [p_{min}, p_{max}], \quad (9c)$$

where $L(C^{j+1}) = \max(\{L(C_\ell^{j+1}), \ell = 1, 2, \ldots, m\})$ and p_{min}, p_{max} are two positive numbers.

Convergence analysis. We can easily extend the convergence result of Alg. 1 to the supervised version by checking the conditions in the proof of Theorem 3.7. The proof is omitted here.

4. Algorithm for Solving Problem (17)

The minimization problem (17) is equivalent to

$$\min_X \operatorname{Tr}(X^\top A X - B^\top X), \quad (10)$$

subject to $\|X\|_0 \leq T$, where $A = K(D, D)$ and $B = K(D, Y)$. We use proximal gradient descent method to solve (10). More specifically, we update X via

$$X^{j+1} = \operatorname{sign}(\hat{\mathrm{X}}^{\mathrm{j}}) \odot H_T(|\hat{X}^j|), \quad (11)$$

where $\hat{X}^j = X^j - (AX^j - B)/v$ and H_T is defined in (5). The step size v is set as $v > \lambda(A)$ where $\lambda(A)$ is the maximal eigenvalue of A.

5. Details of The Global Feature Extraction

Given the sparse code $C \in \mathcal{R}^{m \times n \times t \times k}$ of a DT sequence $g \in \mathcal{R}^{m \times n \times t}$, we use $C_{(i)} = C(:,:,:,i) \in \mathcal{R}^{m \times n \times t}$ to denote the sparse code that corresponds to the ith dictionary atom D_i. As the sparse code is extracted by a sliding window, $C_{(i)}$ can be viewed as a sequence whose size is the same as the original DT sequence. Then, we extract a histogram $h_{(i)}^{\mathrm{A}} \in \mathcal{R}^{l_0 \times 1}$ on $C_{(i)}$ w.r.t. code value. Moreover, we extract three mean histograms along X, Y, and T axes, which are denoted by $h_{(i)}^{\mathrm{X}}, h_{(i)}^{\mathrm{Y}}, h_{(i)}^{\mathrm{T}} \in R^{l_1 \times 1}$ respectively. Take the X-axis case for example. We cut $C_{(i)}$ into slices along the X axis, and compute a histogram w.r.t. code value on each slice. These histograms are averaged to be the mean histogram for the X axis. See Fig. 1 for an illustration of such a process. Define $h_{(i)} = [h_{(i)}^{\mathrm{A}}; h_{(i)}^{\mathrm{X}}; h_{(i)}^{\mathrm{Y}}; h_{(i)}^{\mathrm{T}}]$. The final feature vector for g is the concatenation of $h_{(i)}$ over i.

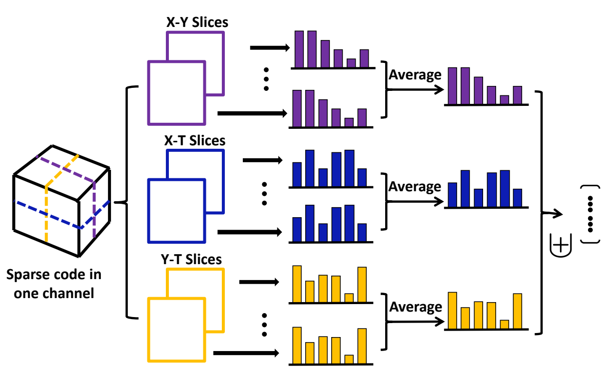

Figure 1. Calculation of space-time histograms in one coding channel.