Information-theoretic Dictionary Learning

Qiang Qiu, Vishal Patel, and Rama Chellappa, "Information-theoretic Dictionary Learning for Image Classification", IEEE Transaction on Pattern Analysis and Machine Intelligence, (under review)
Two-stage Dictionary Learning

- **Input:** an initial dictionary $D^0$, input signals $Y$, labels $C$

- **Stage I:** Greedy Atom Selection
  
  Select a set of atoms $D^*$ from $D^0$ via
  
  $\arg \max_D \lambda_1 I(D; D^0 \setminus D) + \lambda_2 I(X_D; C) + \lambda_3 I(Y; D)$
  
  Compactness  Discriminability  Representation

- **Stage II:** Gradient Ascent Atom Update
  
  Update the selected set of atoms to further maximize
  
  $\lambda_2 I(X_D; C) + \lambda_3 I(Y; D)$

- **Output:** a compact, discriminative and reconstructive dictionary
Dictionary Discriminability $I(X_{D^*}; C)$

- An upper bound on the Bayes error over $X_{D^*}$
  \[ \frac{1}{2} (H(C) - I(X_{D^*}; C)) \]  
  [1]

- A discriminative dictionary $D^*$ is obtained via
  \[ \arg \max_{D^*} I(X_{D^*}; C) \]

- A greedy atom selection

Start with $D^* = \emptyset$

- Untill $|D^*| = k$, iteratively choose $d^*$ from $D^0 \setminus D^*$,
  \[ \arg \max_{d^* \in D^0 \setminus D^*} I(X_{D^* \cup d^*}; C) - I(X_{D^*}; C) \]

Evaluate \( I(X_{D^*}; C) \)

- \( I(X_{D^*}; C) \) is evaluated as follows

\[
I(X_{D^*}; C) = H(X_{D^*}) - H(X_{D^*} | C)
\]

\[
= H(X_{D^*}) - \sum_{c=1}^{p} p(c)H(X_{D^*} | c)
\]

- Compute \( p(X_{D^*} | c) \) with kernel density estimation

\[
p(x | c) = \frac{1}{N_c} \sum_{j=1}^{N_c} K_G(x - x_j^c, \sigma^2 I)
\]

with a Gaussian kernel

\[
K_G(x, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} x^T \Sigma^{-1} x \right)
\]

- Compute \( p(X_{D^*}) \) as

\[
p(x) = \sum_c p(x | c)p(c)
\]

where \( p(c) = \frac{N_c}{N} \)
A reconstructive dictionary $D^*$ maximizes the information between $D^*$ and signals $Y$, i.e.,

$$\arg \max_{D^*} I(Y; D^*)$$

A greedy atom selection

- Start with $D^* = \emptyset$
- Untill $|D^*| = k$, iteratively choose $d^*$ from $D^\circ \setminus D^*$,

$$\arg \max_{d^* \in D^\circ \setminus D^*} I(Y; D^* \cup d^*) - I(Y; D^*)$$
Evaluate $I(Y; D^*)$

- $I(Y; D^*)$ is evaluated as follows
  
  \[
  I(Y; D^*) = H(Y) - H(Y|D^*)
  \]

  \[
  = H(Y) - \sum_{i=1}^{N} H(y_i|D^*)
  \]

- Compute $p(y_i|D^*)$ with the following relation

  \[
  y_i = D^*x_i + r_i \quad \text{where} \quad r_i: \text{a Gaussian residual vector}
  \]

- Write such relation in a probabilistic form, we have

  \[
P(y_i|D^*) \propto \exp\left(-\frac{1}{2\sigma_r^2}||y_i - D^*x_i||^2\right)
  \]
Selection of \(\{\lambda_1, \lambda_2, \lambda_3\}\)

- Each term is maximized in a unified greedy manner.

\[
\arg\max_D \lambda_1 I(D; D^o \setminus D) + \lambda_2 I(X_D; C) + \lambda_3 I(Y; D)
\]

- \(\{\lambda_1, \lambda_2, \lambda_3\}\) are estimated as follows,

\[
\lambda_1 = 1
\]

\[
\lambda_2 = \frac{\max_i I(X_{d_i}; C)}{\max_i I(d_i; D^o \setminus d_i)}
\]

\[
\lambda_3 = \frac{\max_i I(Y; d_i)}{\max_i I(d_i; D^o \setminus d_i)}
\]

- Only the first greedily selected atoms are involved.
Dictionary Update

- **The objective is to further maximize** \( \lambda_2 I(X_D; C) + \lambda_3 I(Y; D) \)

- **Assumption:**

  The number of selected atoms \( D << \) The signal feature space dimension.

- **The sparse representation of signals \( Y \) can be obtained**

  \[ X = D^\dagger Y \]

  where \( D^\dagger = (D^T D)^{-1} D^T \)

  to minimize the representation error \( \| Y - DX \|^2 \)

**maximize** \( \lambda_2 I(X_D; C) + \lambda_3 I(Y; D) \)

**Finding** \( D^\dagger \) that maximize \( I(D^\dagger Y; C) \)
A Differentiable Objective Function

- **Revise** $I(\mathbf{D}^\dagger \mathbf{Y} ; \mathbf{C})$ as a differentiable function.
- $I(\mathbf{X}; \mathbf{C})$ is the $D(p\|q)$ between $p(\mathbf{X}, \mathbf{C})$ and $p(\mathbf{X})p(\mathbf{C})$
- **Approximate** $D(p\|q)$ with
  \[
  Q(p\|q) = \int_t (p(t) - q(t))^2 \, dt \quad (D(p\|q) \geq \frac{1}{2} Q(p\|q))
  \]
- $I(\mathbf{X}; \mathbf{C})$ is revised as
  \[
  I_Q(\mathbf{X}; \mathbf{C}) = \sum_c \int_x p(\mathbf{x}, c)^2 \, d\mathbf{x} - 2 \sum_c \int_x p(\mathbf{x}, c)p(\mathbf{x})p(c) \, d\mathbf{x} + \sum_c \int_x p(\mathbf{x})^2 p(c)^2 \, d\mathbf{x}.
  \]
- **Using** $p(\mathbf{x}/c), p(\mathbf{x}), p(c)$, $I_Q(\mathbf{X}; \mathbf{C}) = \frac{1}{N^2} \sum_{c=1}^p \sum_{k=1}^{N_c} \sum_{l=1}^{N_c} K_G(\mathbf{x}_k^c - \mathbf{x}_l^c, 2\sigma^2 \mathbf{I})$
  \[
  - \frac{2}{N^2} \sum_{c=1}^p \frac{N_c}{N} \sum_j \sum_{k=1}^{N_c} K_G(\mathbf{x}_j^c - \mathbf{x}_k, 2\sigma^2 \mathbf{I})
  \]
  \[
  + \frac{1}{N^2} \left( \sum_{c=1}^p \left( \frac{N_c}{N} \right)^2 \right) \sum_{k=1}^{N} \sum_{l=1}^{N} K_G(\mathbf{x}_k - \mathbf{x}_l, 2\sigma^2 \mathbf{I}).
  \]
Gradient Ascent Dictionary Update

- Use gradient ascent to find $D^\dagger$ that maximize $I_Q(D^\dagger Y; C)$
- For simplicity, $\Phi \triangleq (D^\dagger)^T$

**Update step:**

$$\Phi_{k+1} = \Phi_k + \nu \frac{\partial I_Q}{\partial \Phi} \bigg|_{\Phi=\Phi_k}$$

where

$$\frac{\partial I_Q}{\partial \Phi} = \sum_{c=1}^{p} \sum_{i=1}^{N_c} \frac{\partial I_Q}{\partial x_i^c} \frac{\partial x_i^c}{\partial \Phi}$$

**Partial derivatives:**

$$\frac{\partial}{\partial x_i^c} I_Q = \frac{1}{N^2 \sigma^2} \sum_{k=1}^{N_c} \mathcal{K}_G(x_k^c - x_i^c, 2\sigma^2 I)(x_k^c - x_i^c)$$

$$- \frac{2}{N^2 \sigma^2} \left( \sum_{c=1}^{p} \left( \frac{N_c}{N} \right)^2 \right) \sum_{k=1}^{N} \mathcal{K}_G(x_k - x_i^c, 2\sigma^2 I)(x_k - x_i^c)$$

$$+ \frac{1}{N^2 \sigma^2} \sum_{k=1}^{p} \frac{N_k}{2N} \sum_{j=1}^{N_c} \mathcal{K}_G(x_j^k - x_i^c, 2\sigma^2 I)(x_j^k - x_i^c).$$

**Result:**

$$x_i^c = \Phi^T y_i^c$$

$$\Rightarrow \frac{\partial x_i^c}{\partial \Phi} = (y_i^c)^T.$$
Supervised Sparse Coding

**Input:** Dictionary \( \mathbf{D}^0 \), signals \( \mathbf{Y} \), class labels \( \mathcal{C} \), sparsity level \( T \)

**Output:** sparse coefficients \( \mathbf{X} \), reconstruction \( \hat{\mathbf{Y}} \)

\[
\text{begin}
\begin{align*}
\text{Initialization stage:} & \quad 1. \text{ Initialize } \mathbf{X} \text{ with any pursuit algorithm, } \\
& \quad \quad \quad \quad i = 1, \ldots, N \min_{\mathbf{x}_i} \| \mathbf{y}_i - \mathbf{D}^0 \mathbf{x}_i \|_2 \quad \text{s.t. } \| \mathbf{x}_i \|_0 \leq T.
\end{align*}
\]

**IDS stage (shared atoms):**
2. Estimate \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) from \( \mathbf{Y}, \mathbf{X} \) and \( \mathcal{C} \);
3. Find \( T \) most compact, discriminative and reconstructive atoms:
\[
\mathbf{D}^* \leftarrow \emptyset; \quad \Gamma \leftarrow \emptyset;
\]
4. For \( i = 1 \) to \( T \) do
\[
\begin{align*}
\mathbf{d}^* & \leftarrow \arg \max_{\mathbf{d} \in \mathbf{D}^0 \setminus \mathbf{D}^*} \lambda_1 (I(\mathbf{D}^* \cup \mathbf{d}; \mathbf{D}^0 \setminus (\mathbf{D}^* \cup \mathbf{d}))) - \\
& \quad - I(\mathbf{D}^*; \mathbf{D}^0 \setminus \mathbf{D}^*) + \lambda_2 (I(\mathbf{X}_{\mathcal{D}^*} \cup \mathbf{C}) - I(\mathbf{X}_{\mathcal{D}^*}; \mathbf{C})) + \\
& \quad + \lambda_3 (I(\mathbf{Y}; \mathbf{D}^* \cup \mathbf{d}) - I(\mathbf{Y}; \mathbf{D}^*));
\end{align*}
\]
\[
\mathbf{D}^* \leftarrow \mathbf{D}^* \cup \mathbf{d}^*;
\]
\[
\Gamma \leftarrow \Gamma \cup \gamma^*; \quad \gamma^* \text{ is the index of } \mathbf{d}^* \text{ in } \mathbf{D}^0;
\]
end

4. Compute sparse codes and reconstructions:
\[
\mathbf{X} \leftarrow \text{pinv}(\mathbf{D}^*) \mathbf{Y};
\]
\[
\hat{\mathbf{Y}} \leftarrow \mathbf{D}^* \mathbf{X};
\]
5. return \( \mathbf{X}, \hat{\mathbf{Y}}, \mathbf{D}^*, \Gamma \);
\[
\text{end}
\]

**Algorithm 1:** Sparse coding with global atoms

**Input:** Dictionary \( \mathbf{D}^0 \), signals \( \mathbf{Y} = \{ \mathbf{Y}_c \}_{c=1}^p \), sparsity level \( T \)

**Output:** sparse coefficients \( \{ \mathbf{X}_c \}_{c=1}^p \), reconstruction \( \{ \hat{\mathbf{Y}}_c \}_{c=1}^p \)

\[
\text{begin}
\begin{align*}
\text{Initialization stage:} & \quad 1. \text{ Initialize } \mathbf{X} \text{ with any pursuit algorithm, } \\
& \quad \quad \quad \quad i = 1, \ldots, N \min_{\mathbf{x}_i} \| \mathbf{y}_i - \mathbf{D}^0 \mathbf{x}_i \|_2 \quad \text{s.t. } \| \mathbf{x}_i \|_0 \leq T.
\end{align*}
\]

**IDS stage (dedicated atoms):**
2. For \( c = 1 \) to \( p \) do
\[
\begin{align*}
\mathcal{C}_c & \leftarrow \{ c_i | c_i = 1 \text{ if } y_i \in \mathbf{Y}_c, 0 \text{ otherwise} \};
\end{align*}
\]
3. Estimate \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) from \( \mathbf{Y}_c, \mathbf{X} \) and \( \mathcal{C}_c \);
4. For \( i = 1 \) to \( T \) do
\[
\begin{align*}
\mathbf{d}^* & \leftarrow \arg \max_{\mathbf{d} \in \mathbf{D}^0 \setminus \mathbf{D}_c^*} \lambda_1 (I(\mathbf{D}_c^* \cup \mathbf{d}; \mathbf{D}^0 \setminus (\mathbf{D}_c^* \cup \mathbf{d}))) - \\
& \quad - I(\mathbf{D}_c^*; \mathbf{D}^0 \setminus \mathbf{D}_c^*) + \lambda_2 (I(\mathbf{X}_{\mathcal{D}_c^*} \cup \mathbf{C}_c) - I(\mathbf{X}_{\mathcal{D}_c^*}; \mathbf{C}_c)) + \\
& \quad + \lambda_3 (I(\mathbf{Y}_c; \mathbf{D}_c^* \cup \mathbf{d}) - I(\mathbf{Y}_c; \mathbf{D}_c^*));
\end{align*}
\]
\[
\mathbf{D}_c^* \leftarrow \mathbf{D}_c^* \cup \mathbf{d}^*;
\]
\[
\Gamma_c \leftarrow \Gamma_c \cup \gamma^*; \quad \gamma^* \text{ is the index of } \mathbf{d}^* \text{ in } \mathbf{D}_c^0;
\]
end

5. Compute sparse codes and reconstructions:
\[
\mathbf{X}_c \leftarrow \text{pinv}(\mathbf{D}_c^*) \mathbf{Y}_c;
\]
\[
\hat{\mathbf{Y}}_c \leftarrow \mathbf{D}_c^* \mathbf{X}_c;
\]
5. return \( \{ \mathbf{X}_c \}_{c=1}^p, \{ \hat{\mathbf{Y}}_c \}_{c=1}^p \), \( \{ \mathbf{D}_c^* \}_{c=1}^p, \{ \Gamma_c \}_{c=1}^p \);
\[
\text{end}
\]

**Algorithm 2:** Sparse coding with atoms per class
Sparse Coding with Atom Updates

Algorithm 3: Sparse coding with Atom Updates

Input: Dictionary $D^o$, signals $Y = \{Y_c\}_{c=1}^P$, class labels $C$, sparsity level $T$, update step $\nu$

Output: Learned dictionary $D$, sparse coefficients $X$, reconstruction $\hat{Y}$

begin

Sparse coding stage:
Use supervised sparse coding to obtain $\{D^*_c\}_{c=1}^P$.

IDU stage:
foreach class $c$ do

[In the shared atom case, use the global label $C$ instead of $C_c$, and one iteration is required as the same $D^*_c$ is used for all classes.]

$C_c \leftarrow \{c_i | c_i = 1 \text{ if } y_i \in Y_c, 0 \text{ otherwise } \}$;

$\Phi_1 \leftarrow \text{pinv}(D^*_c)^T$;
$X \leftarrow \text{pinv}(D^*_c)Y$;
repeat

$\Phi_{k+1} = \Phi_k + \nu \frac{\partial I_Q(X, C_c)}{\partial \Phi} |_{\Phi = \Phi_k}$;
$D^* \leftarrow \text{pinv}(\Phi_{k+1}^T)$;
$X \leftarrow \text{pinv}(D^*)Y$;
until convergence;
$D^*_c \leftarrow D^*$;

end

foreach class $c$ do

$X_c \leftarrow \text{pinv}(D^*_c)Y_c$;
$\hat{Y}_c \leftarrow D^*_c X_c$;
end

5. return $\{X_c\}_{c=1}^P$, $\{\hat{Y}_c\}_{c=1}^P$, $\{D^*_c\}_{c=1}^P$;
end
Sparse Representation Examples (sparsity = 3)

(a) Examples of four subjects

(b) Sparse representation for images of four subjects (Sparsity = 3)

(c) Examples of four digits

(d) Sparse representation for images of four handwritten digits (Sparsity = 3)
Dictionary Update Examples
(sparsity = 2)

(a) Before dictionary update (Acc. = 30.63%)
(b) After 100 update iterations (Acc. = 42.34%)
(c) Converge after 489 update iterations (Acc. = 51.35%)

(d) Before dictionary update (Acc. = 73.54%)
(e) After 50 update iterations (Acc. = 84.45%)
(f) Converge after 171 update iterations (Acc. = 87.75%)
Image Reconstruction Example
(sparsity = 2)

Dictionary Atoms

(a) before update  (b) 30 iterations  (c) 57 iterations

Reconstruction

(d) original images  (e) before update  (f) 30 iterations  (g) 57 iterations

Acc: 85.17%  Acc: 89.11%  Acc: 90.47%

Reconstruction to noisy images (60% missing pixels)

(h) noisy images  (i) before update  (j) 30 iterations  (k) 57 iterations

Acc: 76.87%  Acc: 85.03%  Acc: 85.71%