Learning Low-rank Transformations: Algorithms and Applications

Qiang Qiu
Guillermo Sapiro
Motivation
Outline

- Low-rank transform - algorithms and theories
- Applications
  - Subspace clustering
  - Classification
  - Hashing/indexing
Algorithms and Theories
A Toy Formulation

- A toy formulation

\[
\arg\min_T \sum_{c=1}^{C} \text{rank}(TY_c) - \text{rank}(TY), \quad \text{s.t. } ||T||_2 = 1
\]

- Notation
  - \( Y_c \) denotes \( d \)-dim points in the \( c \)-th class (arranged as columns).
  - \( Y = [Y_1, Y_2, \ldots, Y_C] \), points from all \( C \) classes.
  - \( T \) is a learned \( d \times d \) transformation matrix.

- Theorem
  - \( \text{rank}([A, B]) \leq \text{rank}(A) + \text{rank}(B) \)
  - Non-negative.
  - But zero for independent matrices.
Low-rank Transformation

- Basic formulation
  \[
  \arg\min_T \sum_{c=1}^C \|TY_c\|_* - \|TY\|_*, \quad \text{s.t.} \|T\|_2 = 1.
  \]

- \( |A|_* \) denotes the nuclear norm of the matrix \( A \):
  - The sum of the singular values of \( A \).
  - A good approximation to the matrix rank.

- Theorem:
  - Non-negative
  - **Zero for orthogonal subspaces**
    - Not true for rank and other popular norms

- Works on-line.
- Works with compressing transform matrix.
\( \theta_{AB} = \frac{\pi}{2} = 1.57 \).  

\[ T = \begin{bmatrix} 1.00 & 0 \\ 0 & 1.00 \end{bmatrix}; \quad \theta_{AB} = 1.57. \]

\( \theta_{AB} = \frac{\pi}{4} = 0.79, \quad |A|_* = 1, |B|_* = 1, \quad |[A, B]|_* = 1.41 \)

\[ T = \begin{bmatrix} 0.50 & -0.21 \\ -0.21 & 0.91 \end{bmatrix}; \quad \theta_{AB} = 1.57, \quad |A|_* = 1, |B|_* = 1, \quad |[A, B]|_* = 1.95 \]

\( \left[ \theta_{AB} = 0.79, \quad \theta_{AC} = 0.79, \quad \theta_{BC} = 1.05 \right], \quad \left[ \epsilon_A = 0.0141, \quad \epsilon_B = 0.0131, \quad \epsilon_C = 0.0148 \right], \quad |A|_* = 4.06, \quad |B|_* = 4.08, \quad |C|_* = 4.16. \]

\( \left[ \theta_{AB} = 1.51, \quad \theta_{AC} = 1.49, \quad \theta_{BC} = 1.57 \right], \quad \left[ \epsilon_A = 0.0091, \quad \epsilon_B = 0.0085, \quad \epsilon_C = 0.0114 \right], \quad |A|_* = 1.93, \quad |B|_* = 2.37, \quad |C|_* = 1.20. \)
Low-rank Transformation

(a) Two classes \( \{Y_+, Y_-\} \),
\[
Y_+ = \{ A(\text{blue}), B(\text{cyan}) \},
\]
\[
Y_- = \{ C(\text{yellow}), D(\text{red}) \},
\]
\[
\theta_{AB} = 1.1, \theta_{AC} = 1.1, \theta_{AD} = 1.1,
\]
\[
\theta_{BC} = 1.3, \theta_{BD} = 1.4, \theta_{CD} = 0.5,
\]
\[
|Y_+|^* = 1.58, \quad |Y_-|^* = 1.27.
\]

(b) \( T = \begin{bmatrix} -3.64 & -1.95 & 5.98 \\ 0.19 & 3.87 & 3.35 \end{bmatrix} \);
\[
\theta_{AB} = 0.7, \theta_{AC} = 0.78, \theta_{AD} = 0.2,
\]
\[
\theta_{BC} = 1.5, \theta_{BD} = 0.9, \theta_{CD} = 0.57,
\]
\[
|Y_+|^* = 1.35, \quad |Y_-|^* = 1.27.
\]

(c) \( T = \begin{bmatrix} 1.47 & 0.26 & -0.73 \\ 0.07 & 0.06 & -1.62 \end{bmatrix} \);
\[
\theta_{AB} = 0.04, \theta_{AC} = 1.54, \theta_{AD} = 1.5,
\]
\[
\theta_{BC} = 1.55, \theta_{BD} = 1.56, \theta_{CD} = 0.0,
\]
\[
|Y_+|^* = 1.02, \quad |Y_-|^* = 1.00.
\]

(d) Two classes \( \{A(\text{blue}), B(\text{red})\} \),
\[
\theta_{AB} = 0.31,
\]
\[
|A|^* = 1.91, |B|^* = 1.88.
\]

(e) \( T = \begin{bmatrix} -0.54 & 2.60 & -9.51 \\ 0.56 & -3.21 & -1.02 \end{bmatrix} \);
\[
\theta_{AB} = 0,
\]
\[
|A|^* = 1.52, \quad |B|^* = 1.69.
\]

(f) \( T = \begin{bmatrix} 0.49 & -0.11 & 1.27 \\ -0.09 & 0.29 & -0.59 \end{bmatrix} \);
\[
\theta_{AB} = 1.57,
\]
\[
|A|^* = 1.08, \quad |B|^* = 1.03.
\]
Theorem 1  Let $A$ and $B$ be matrices of the same row dimensions, and $[A, B]$ be the concatenation of $A$ and $B$, we have

$$||[A, B]||_\ast \leq ||A||_\ast + ||B||_\ast.$$

Proof:

$$||A||_\ast + ||B||_\ast = ||[A \ 0]||_\ast + ||[0 \ B]||_\ast \geq ||[A \ 0] + [0 \ B]||_\ast = ||[A, B]||_\ast$$
Theorem 2 Let $A$ and $B$ be matrices of the same row dimensions, and $[A, B]$ be the concatenation of $A$ and $B$, we have

$$||[A, B]||_* = ||A||_* + ||B||_*.$$  

when the column spaces of $A$ and $B$ are orthogonal.

Proof: We perform the singular value decomposition of $A$ and $B$ as

$$A = [U_A U_A] \begin{bmatrix} \Sigma^A & 0 \\ 0 & 0 \end{bmatrix} [V_A V_A]', \quad B = [U_B U_B] \begin{bmatrix} \Sigma^B & 0 \\ 0 & 0 \end{bmatrix} [V_B V_B]',$$

where the diagonal entries of $\Sigma^A$ and $\Sigma^B$ contain non-zero singular values. We have

$$AA' = [U_A U_A] \begin{bmatrix} \Sigma^A & 0 \\ 0 & 0 \end{bmatrix} [U_A U_A]', \quad BB' = [U_B U_B] \begin{bmatrix} \Sigma^B & 0 \\ 0 & 0 \end{bmatrix} [U_B U_B]' .$$

The column spaces of $A$ and $B$ are considered to be orthogonal, i.e., $U_A'U_B = 0$. The above can be written as

$$AA' = [U_A U_B] \begin{bmatrix} \Sigma^A & 0 \\ 0 & 0 \end{bmatrix} [U_A U_B]', \quad BB' = [U_A U_B] \begin{bmatrix} 0 & 0 \\ \Sigma^B & 0 \end{bmatrix} [U_A U_B]' .$$

Then, we have

$$[A, B][A, B]' = AA' + BB' = [U_A U_B] \begin{bmatrix} \Sigma^A & 0 \\ 0 & \Sigma^B \end{bmatrix} [U_A U_B]' .$$

The nuclear norm $||A||_*$ is the sum of the square root of the singular values of $AA'$. Thus,

$$||[A, B]||_* = ||A||_* + ||B||_*.$$
Proposition 3 Let \( A \) and \( B \) be matrices of the same row dimensions, and \([A, B]\) be the concatenation of \( A \) and \( B \), we have

\[
\|[A, B]\|_2 \leq \|A\|_2 + \|B\|_2,
\]

with equality if at least one of the two matrices is zero.

Proposition 4 Let \( A \) and \( B \) be matrices of the same row dimensions, and \([A, B]\) be the concatenation of \( A \) and \( B \), we have

\[
\|[A, B]\|_F \leq \|A\|_F + \|B\|_F,
\]

with equality if and only if at least one of the two matrices is zero.
Kernelized Transform

\[
\min_T \sum_{c=1}^{C} \|T\mathcal{K}(Y_c)\|_* - \|T\mathcal{K}(Y)\|_* , \quad \text{s.t. } \|T\|_2 = 1.
\]

\[
\mathcal{K}(y) = (\kappa(y, y_1); \ldots; \kappa(y, y_n))
\]
Transform-based Dimension Reduction

Extended YaleB face dataset
Subspace Clustering using Low-rank Transform
Subspace clustering

Input: A set of data points $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N} \subseteq \mathbb{R}^d$ in a union of $C$ subspaces.
Output: A partition of $\mathbf{Y}$ into $C$ disjoint clusters $\{\mathbf{Y}_c\}_{c=1}^{C}$ based on underlying subspaces.

begin

1. Initial a transformation matrix $\mathbf{T}$ as the identity matrix;

repeat

   Assignment stage:
   2. Assign points in $\mathbf{TY}$ to clusters with any subspace clustering methods, e.g., the proposed R-SSC;

   Update stage:
   3. Obtain transformation $\mathbf{T}$ by minimizing (6) based on the current clustering result;

until assignment convergence;

4. Return the current clustering result $\{\mathbf{Y}_c\}_{c=1}^{C}$;

end

Algorithm 1: Learning a robust subspace clustering (LRSC) framework.
For the transformed points, we first recover their low-rank representation $L$:

$$\arg \min_{L,S} \|L\|_* + \beta \|S\|_1 \quad \text{s.t.} \quad TY = L + S.$$ 

Each transformed point $Ty_i$ is then represented using its KNN in $L$, denoted as $L_i$:

$$\arg \min_{x_i} \|Ty_i - L_ix_i\|_2^2 \quad \text{s.t.} \quad 1'x_i = 1.$$ 

Let $\bar{L}_i = L_i - 1Ty_i^T$, $x_i = \bar{L}_i\bar{L}_i^T \backslash 1$

Perform spectral clustering on sparse representation matrix $(|X| + |X'|)$
(a) Example illumination conditions.

(b) Example subjects.

(a) Ground truth.  
(b) SSC, $e = 71.25\%$,  
$t = 714.99$ sec.  
(c) LBF, $e = 76.37\%$,  
$t = 460.76$ sec.

(d) LSA, $e = 71.96\%$,  
$t = 22.57$ sec.  
(e) R-SSC, $e = 67.37\%$,  
$t = 1.83$ sec.
Misclassification rate (e%) on clustering different subjects.

<table>
<thead>
<tr>
<th>Subsets</th>
<th>[1:10]</th>
<th>[1:15]</th>
<th>[1:20]</th>
<th>[1:25]</th>
<th>[1:30]</th>
<th>[1:38]</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>38</td>
</tr>
<tr>
<td>LSA</td>
<td>78.25</td>
<td>82.11</td>
<td>84.92</td>
<td>82.98</td>
<td>82.32</td>
<td>84.79</td>
</tr>
<tr>
<td>LBF</td>
<td>78.88</td>
<td>74.92</td>
<td>77.14</td>
<td>78.09</td>
<td>78.73</td>
<td>79.53</td>
</tr>
<tr>
<td>LRSC</td>
<td><strong>5.39</strong></td>
<td><strong>4.76</strong></td>
<td><strong>9.36</strong></td>
<td><strong>8.44</strong></td>
<td><strong>8.14</strong></td>
<td><strong>11.02</strong></td>
</tr>
</tbody>
</table>
Classification using Low-rank Transform
Basic Scheme

- For the c-th class, we first recover its low-rank representation $L_c$

$$\arg \min_{L_c, S_c} \|L_c\|_* + \beta \|S_c\|_1 \quad \text{s.t.} \quad TY_c = L_c + S_c$$

- Each testing point $y$ is assigned to $L_c$ that gives the minimal reconstruction error

$$\arg \min_{x} \|Ty - L_c x\|_2^2 \quad \text{s.t.} \quad \|x\|_0 \leq T$$
Face recognition across illumination

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-KSVD Zhang and Li (2010)</td>
<td>94.10</td>
</tr>
<tr>
<td>LC-KSVD Jiang et al. (2011)</td>
<td>96.70</td>
</tr>
<tr>
<td>SRC Wright et al. (2009)</td>
<td>97.20</td>
</tr>
<tr>
<td>Original+NN</td>
<td>91.77</td>
</tr>
<tr>
<td>Class LRT+NN</td>
<td>97.86</td>
</tr>
<tr>
<td>Class LRT+OMP</td>
<td>92.43</td>
</tr>
<tr>
<td>Global LRT+NN</td>
<td>99.10</td>
</tr>
<tr>
<td>Global LRT+OMP</td>
<td><strong>99.51</strong></td>
</tr>
</tbody>
</table>
Face recognition across pose and illumination

<table>
<thead>
<tr>
<th>Method</th>
<th>Frontal (c27)</th>
<th>Side (c05)</th>
<th>Profile (c22)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMD Castillo and Jacobs (2009)</td>
<td>83</td>
<td>82</td>
<td>57</td>
</tr>
<tr>
<td>Original+NN</td>
<td>39.85</td>
<td>37.65</td>
<td>17.06</td>
</tr>
<tr>
<td>Original(crop+flip)+NN</td>
<td>44.12</td>
<td>45.88</td>
<td>22.94</td>
</tr>
<tr>
<td>Class LRT+NN</td>
<td>98.97</td>
<td>96.91</td>
<td>67.65</td>
</tr>
<tr>
<td>Class LRT+OMP</td>
<td>100</td>
<td>100</td>
<td>67.65</td>
</tr>
<tr>
<td>Global LRT+NN</td>
<td>97.06</td>
<td>95.58</td>
<td>50</td>
</tr>
<tr>
<td>Global LRT+OMP</td>
<td>100</td>
<td>98.53</td>
<td>57.35</td>
</tr>
</tbody>
</table>
Classification using Transform Forest
Transform Forest

The ensemble model

Forest output probability \( p(c|v) = \frac{1}{T} \sum_{t=1}^{T} p_t(c|v) \)
Transform learner

- Learn $\mathbf{T}$ at each split node

$$\arg\min_{\mathbf{T}} \|\mathbf{T}\mathbf{Y}_+\|_* + \|\mathbf{T}\mathbf{Y}_-\|_* - \|\mathbf{T}[\mathbf{Y}_+, \mathbf{Y}_-]\|_*,$$

$$s.t. \|\mathbf{T}\|_2 = 1,$$

- Kernelized version

$$\min_{\mathbf{T}} \|\mathbf{T}\mathcal{K}(\mathbf{Y}^+)\|_* + \|\mathbf{T}\mathcal{K}(\mathbf{Y}^-)\|_* - \|\mathbf{T}[\mathcal{K}(\mathbf{Y}^+), \mathcal{K}(\mathbf{Y}^-)]\|_*,$$

$$s.t. \|\mathbf{T}\|_2 = 1.$$ (2)
Transform Learner

- **Random Grouping**: Randomly partition training classes arriving at each split node into two groups.

- Learn a pair of dictionaries $D^\pm$, for each of the two groups by minimizing

\[
\min_{D^\pm, z^\pm} \|X^\pm - D^\pm z^\pm\| \quad \text{s.t.} \quad \|z_i^\pm\|_0 \leq l,
\]

- The split function is evaluated using the reconstruction error,

\[
e^\pm(x) = \|x - P^\pm x\|_2
\]

where

\[
P^\pm = D^\pm (D^{\pm T} D^\pm)^{-1} D^{\pm T}
\]
Results
## Quantitative Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Testing time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-tree based methods</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-KSVD (Zhang &amp; Li, 2010)</td>
<td>94.10</td>
<td>-</td>
</tr>
<tr>
<td>LC-KSVD (Jiang et al., 2011)</td>
<td>96.70</td>
<td>-</td>
</tr>
<tr>
<td>SRC (Wright et al., 2009)</td>
<td>97.20</td>
<td>-</td>
</tr>
<tr>
<td><strong>Classification trees</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision stump (1 tree)</td>
<td>28.37</td>
<td>0.09</td>
</tr>
<tr>
<td>Decision stump (100 trees)</td>
<td>91.77</td>
<td>13.62</td>
</tr>
<tr>
<td>Conic section (1 tree)</td>
<td>8.55</td>
<td>0.05</td>
</tr>
<tr>
<td>Conic section (100 trees)</td>
<td>78.20</td>
<td>5.04</td>
</tr>
<tr>
<td>C4.5 (1 tree) (Quinlan, 1993)</td>
<td>39.14</td>
<td>0.21</td>
</tr>
<tr>
<td>LDA (1 tree)</td>
<td>38.32</td>
<td>0.12</td>
</tr>
<tr>
<td>LDA (100 trees)</td>
<td>94.98</td>
<td>7.01</td>
</tr>
<tr>
<td>SVM (1 tree)</td>
<td>95.23</td>
<td>1.62</td>
</tr>
<tr>
<td>Identity learner (1 tree)</td>
<td>84.95</td>
<td>0.29</td>
</tr>
<tr>
<td>Transformation learner (1 tree)</td>
<td><strong>98.77</strong></td>
<td>0.15</td>
</tr>
</tbody>
</table>

Extended YaleB face dataset
On the Number of Trees

(a) MNIST.
(b) 15-Scenes.
(c) Kinect.
Hashing using Transform Forest
Motivation

- Hash/binary codes are needed to deal with big data
  - Storage
  - Retrieval
ForestHash

- Challenge 1: Create consistent hash codes in each tree.
  - Low-rank transform.
- Challenge 2: Merging trees for unique codes per class.
  - A mutual information based technique for near-optimal code aggregation.

We simply set ‘1’ for the visited nodes, and ‘0’ for the rest, obtaining a \((2^d - 2)\)-bit hash code.
Challenge 1: Consistent Codes

- Transform learner: learn \( T \) at each split node

\[
\arg\min_T ||TY_+||_* + ||TY_-||_* - ||T[Y_+, Y_-]||_*,
\]
\[
s.t. ||T||_2 = 1,
\]

- **Random Grouping**: Randomly partition training classes arriving at each split node into two groups.
  - Each tree enforces consistent but non-unique codes for a class.
  - But each class shares codes with different classes in different trees.
Challenge 2: Code Aggregation

- Hash codes from a random forest consisting of $M$ trees for $N$ training samples $\mathcal{B} = \{\mathbf{B}_i\}_{i=1}^M$
- Our objective is to select $k$ code blocks $\mathbf{B}^*$, $k \leq L/(2^d - 2)$
- Unsupervised code aggregation,
  $$\mathbf{B}^* = \arg \max_{\mathbf{B}:|\mathbf{B}|=k} I(\mathbf{B};\mathcal{B}\backslash\mathbf{B}).$$
- Supervised code aggregation (class labels $C$),
  $$\arg \max_{\mathbf{B}:|\mathbf{B}|=k} I(\mathbf{B};C).$$
- Semi-supervised code aggregation,
  $$\arg \max_{\mathbf{B}:|\mathbf{B}|=k} I(\mathbf{B};\mathcal{B}\backslash\mathbf{B}) + \lambda I(\mathbf{B};C).$$
**Example 1: Image Retrieval**

<table>
<thead>
<tr>
<th></th>
<th>6,000 samples per class</th>
<th>100 samples per class</th>
<th>30 samples per class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train time (s)</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>HDML [23]</td>
<td>10</td>
<td>93.780</td>
<td>92.94</td>
</tr>
<tr>
<td>DeepNet [30]</td>
<td>52</td>
<td>34.07</td>
<td>79.12</td>
</tr>
<tr>
<td>FastHash [19]</td>
<td>115</td>
<td>86.5</td>
<td>84.70</td>
</tr>
<tr>
<td>TSH [20]</td>
<td>411</td>
<td>164.325</td>
<td>86.30</td>
</tr>
<tr>
<td>FaceHash-base</td>
<td>1.2</td>
<td>0.02</td>
<td>20.92</td>
</tr>
<tr>
<td>FaceHash-aggr</td>
<td>1.2</td>
<td>0.02</td>
<td>20.28</td>
</tr>
</tbody>
</table>

HDML and DeepNet are deep learning based hashing methods.
Example 1: Image Retrieval

Cifar-10 dataset
Example 2: Cross-modality

Query 1: (Biology) The Kakapo is the only species of flightless parrot in the world, and the only flightless bird that has a lek breeding system. "Collins Field Guide to New Zealand Wildlife

Answer:
- ForestHash
- CM

Query 2: (Sport) Wales won two matches in each Five Nations championship between 1980 and 1984, and in 1983 were nearly upset by Japan; winning by 24-29 at Cardiff...

Answer:
- ForestHash
- CM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>50.8</td>
<td>45.5</td>
<td>18.4</td>
<td>19.6</td>
<td>22.3</td>
<td>22.6</td>
<td>27.4</td>
<td>25.8</td>
</tr>
</tbody>
</table>

The Wikipedia dataset
Example 3: Document Retrieval

Reuters21578 dataset
Example 4: Faces

Each face is indexed by a 48-bit hash code.

~30 microseconds to index a face.
~20 milliseconds to scan one million faces.
The blacklist/whitelist scenario

- Enrolling in a list 200 subjects.
- 5,992 face queries

Query over a 73K database containing 37,007 unseen faces from 200 known subjects and 35,902 unseen faces from 27,859 unknown subjects.

<table>
<thead>
<tr>
<th>Method</th>
<th>radius = 0</th>
<th></th>
<th>radius ≤ 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>SH [31]</td>
<td>6.56</td>
<td>0.15</td>
<td>37.18</td>
<td>1.98</td>
</tr>
<tr>
<td>AGH1 [21]</td>
<td>31.74</td>
<td>56.12</td>
<td>17.17</td>
<td>82.30</td>
</tr>
<tr>
<td>AGH2 [21]</td>
<td>22.44</td>
<td>57.48</td>
<td>12.17</td>
<td>89.52</td>
</tr>
<tr>
<td>LDAHash [27]</td>
<td>23.42</td>
<td>0.65</td>
<td>45.30</td>
<td>10.25</td>
</tr>
<tr>
<td>TSH [20]</td>
<td>6.75</td>
<td>0.22</td>
<td>9.96</td>
<td>0.35</td>
</tr>
<tr>
<td>FaceHash</td>
<td>95.91</td>
<td>82.29</td>
<td>88.05</td>
<td>89.38</td>
</tr>
<tr>
<td>FaceHash (48-bit)</td>
<td>96.54</td>
<td>80.42</td>
<td>96.45</td>
<td>87.41</td>
</tr>
</tbody>
</table>

Query over a 0.7M database containing 37,007 unseen faces (200 known subjects) and 0.7M unseen faces (29,392 unknown subjects).

<table>
<thead>
<tr>
<th>Method</th>
<th>radius = 0</th>
<th></th>
<th>radius ≤ 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>AGH1 [21]</td>
<td>18.38</td>
<td>56.12</td>
<td>7.75</td>
<td>82.30</td>
</tr>
<tr>
<td>AGH2 [21]</td>
<td>13.56</td>
<td>57.48</td>
<td>5.53</td>
<td>89.52</td>
</tr>
<tr>
<td>LDAHash [27]</td>
<td>23.42</td>
<td>0.65</td>
<td>45.11</td>
<td>10.25</td>
</tr>
<tr>
<td>FaceHash</td>
<td>82.17</td>
<td>82.29</td>
<td>47.58</td>
<td>89.38</td>
</tr>
<tr>
<td>FaceHash (48-bit)</td>
<td>90.74</td>
<td>80.42</td>
<td>81.74</td>
<td>87.41</td>
</tr>
</tbody>
</table>

~24 seconds to index all 0.7M faces.
~16 milliseconds to query 0.7M faces.
Example 4: Faces
Example 5: Cross-modality Faces

Face attributes

1 Male          19 Frowning          37 Narrow Eyes          55 Posed Photo
2 Asian         20 Chubby           38 Eyes Open           56 Attractive Man
3 White         21 Blurry           39 Big Nose           57 Attractive Woman
4 Black         22 Harsh           40 Pointy Nose        58 Indian
5 Baby          23 Lighting Flash   41 Big Lips           59 Gray Hair
6 Child         24 Soft Lighting    42 Mouth Closed       60 Bags Under Eyes
7 Youth         25 Outdoor          43 Mouth Slightly Open 61 Heavy Makeup
8 Middle Aged   26 Curly Hair       44 Mouth Wide Open    62 Rosy Cheeks
9 Senior        27 Wavy Hair        45 Teeth Not Visible  63 Shiny Skin
10 Black Hair    28 Straight Hair    46 No Beard           64 Pale Skin
11 Blond Hair    29 Receding Hairline 47 Goatee            65 5 o' Clock Shadow
12 Brown Hair    30 Bangs            48 Round Jaw          66 Strong Nose-Mouth Lines
13 Bald          31 Sideburns         49 Double Chin        67 Wearing Lipstick
14 No Eyewear   32 Fully Visible Forehead 50 Wearing Hat    68 Flushed Face
15 Eyeglasses   33 Partially Visible Forehead 51 Oval Face    69 High Cheekbones
16 Sunglasses   34 Obstructed Forehead    52 Square Face     70 Brown Eyes
17 Mustache      35 Bushy Eyebrows    53 Round Face         71 Wearing Earrings
18 Smiling       36 Arched Eyebrows    54 Color Photo       72 Wearing Necktie
                                 73 Wearing Necklace
Example 5: Cross-modality Faces

Table 5. FaceHash cross-representation face retrieval performance(%) using attribute queries on large scale datasets (36-bit).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Precision</th>
<th>Recall</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pubfig dataset</td>
<td>93.12</td>
<td>77.24</td>
<td>95.56</td>
<td>88.01</td>
</tr>
<tr>
<td>73K dataset</td>
<td>91.49</td>
<td>77.24</td>
<td>87.25</td>
<td>88.01</td>
</tr>
<tr>
<td>0.7M dataset</td>
<td>76.93</td>
<td>77.24</td>
<td>46.09</td>
<td>88.01</td>
</tr>
</tbody>
</table>
Thank you!
Reference

The Concave-Convex Procedure

- Difference of convex function

\[ J(T) = J_{vex}(T) + J_{cav}(T) \]

\[ = \left[ \sum_{c=1}^{C} \|TY_c\|_* \right] + [-\|TY\|_*] \]

- A simple projected subgradient method

\[ \sum_{c=1}^{C} \partial\|TY_c\|_*Y'_c - \partial\|T^{(t)}Y\|_*Y'. \]
A subgradient of matrix nuclear norm

Input: An $m \times n$ matrix $A$, a small threshold value $\delta$
Output: A subgradient of the nuclear norm $\partial \|A\|_*$.

begin
1. Perform singular value decomposition:
   $A = U \Sigma V$ ;

2. $s \leftarrow$ the number of singular values smaller than $\delta$ ,
3. Partition $U$ and $V$ as
   $U = [U^{(1)}, U^{(2)}]$, $V = [V^{(1)}, V^{(2)}]$ ;
   where $U^{(1)}$ and $V^{(1)}$ have $(n - s)$ columns.

4. Generate a random matrix $B$ of the size $(m - n + s) \times s$,
   $B \leftarrow \frac{B}{\|B\|}$ ;

5. $\partial \|A\|_* \leftarrow U^{(1)} V^{(1)'} + U^{(2)} B V^{(2)'}$ ;

6. Return $\partial \|A\|_*$ ;

end