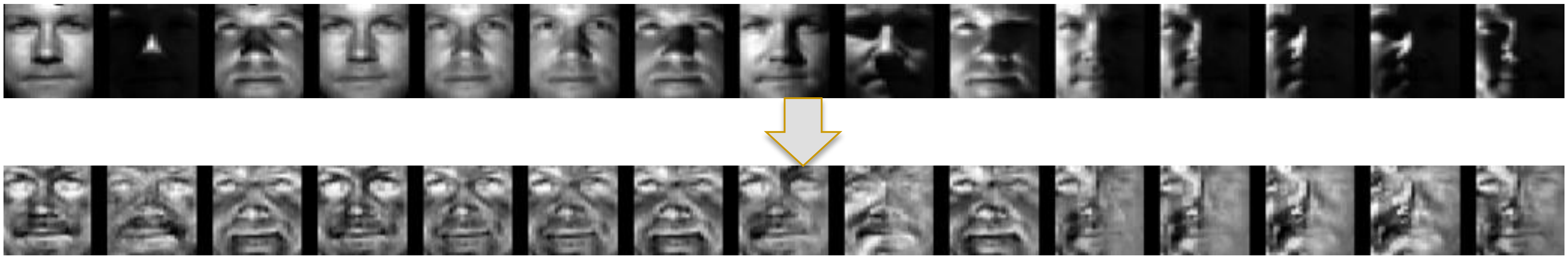
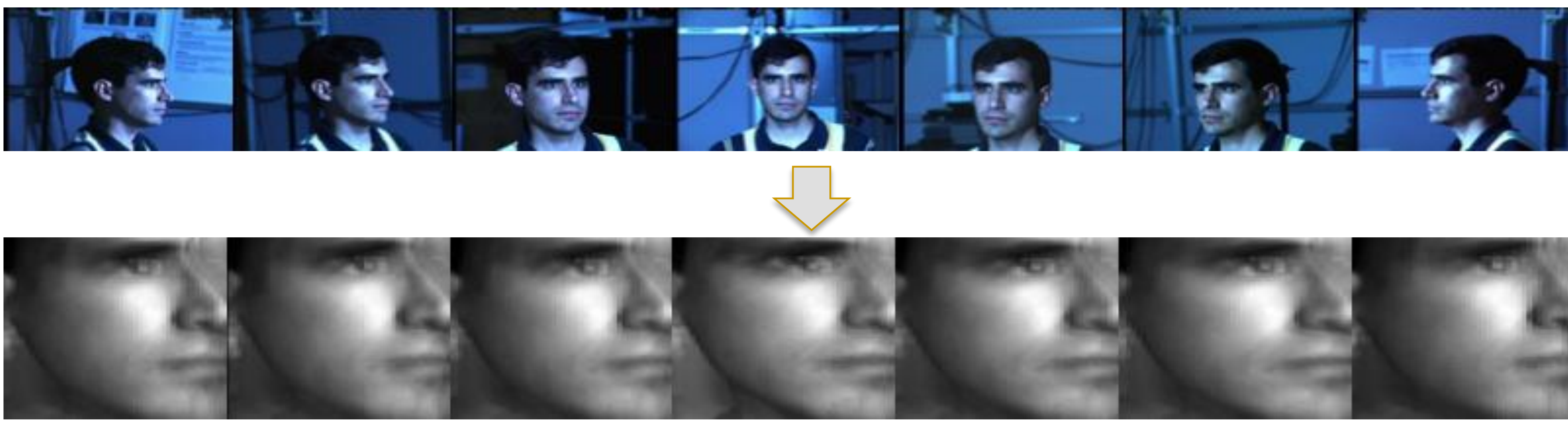
Learning Low-rank Transformations: Algorithms and Applications

Qiang Qiu
Guillermo Sapiro

Motivation





Outline

Duke UNIVERSITY

- Low-rank transform - algorithms and theories
- Applications
  - Subspace clustering
  - Classification
  - Hashing/indexing

Algorithms and Theories

A Toy Formulation

- A toy formulation

$$\arg\min_{\mathbf{T}} \sum_{c=1}^{C} \underbrace{rank(\mathbf{T}\mathbf{Y}_c)}_{\textbf{Intra-class}} - \underbrace{rank(\mathbf{T}\mathbf{Y})}_{\textbf{Inter-class}}, \quad \underbrace{\text{s.t.}||\mathbf{T}||_2 = 1}_{\textbf{Non-trivial solution}}$$

- 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×d transformation matrix.

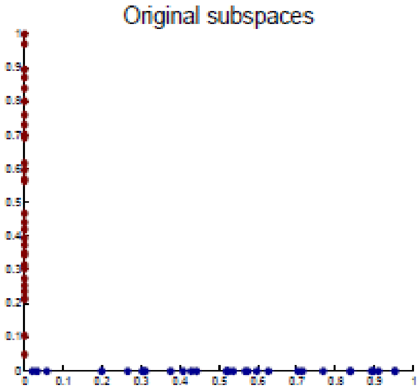
- Theorem $rank([\mathbf{A}, \mathbf{B}]) \leq rank(\mathbf{A}) + rank(\mathbf{B})$
  - Non-negative.
  - But zero for independent matrices.

Low-rank Transformation

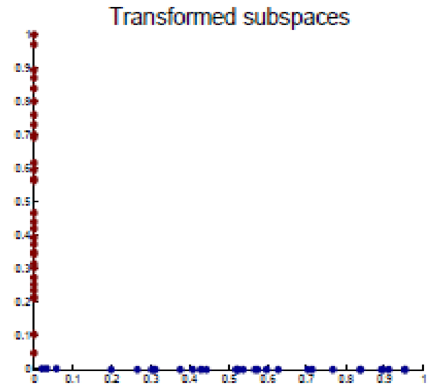
- Basic formulation

$$\arg\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_*, \quad \text{s.t.} ||\mathbf{T}||_2 = 1.$$

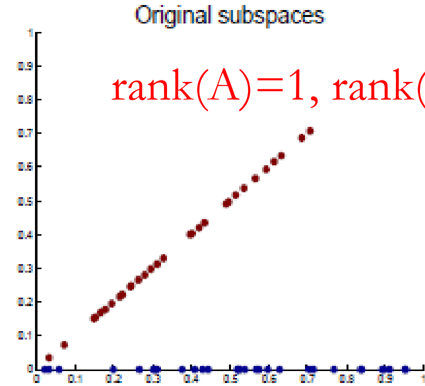
- $|\mathbf{A}|_*$ denotes the nuclear norm of the matrix **A**:
  - The sum of the singular values of **A**.
  - A good approximation to the matrix rank.
- Theorem:
  - Non-negative
  - **Zero for orthogonal subspaces**
    - Not true for rank and other popular norms
- Works on-line.
- Works with compressing transform matrix.



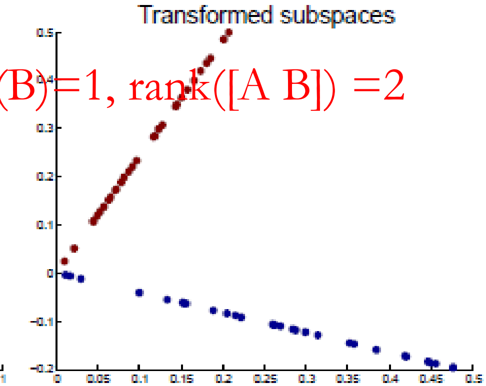
(a) $\theta_{AB} = \frac{\pi}{2} = 1.57.$



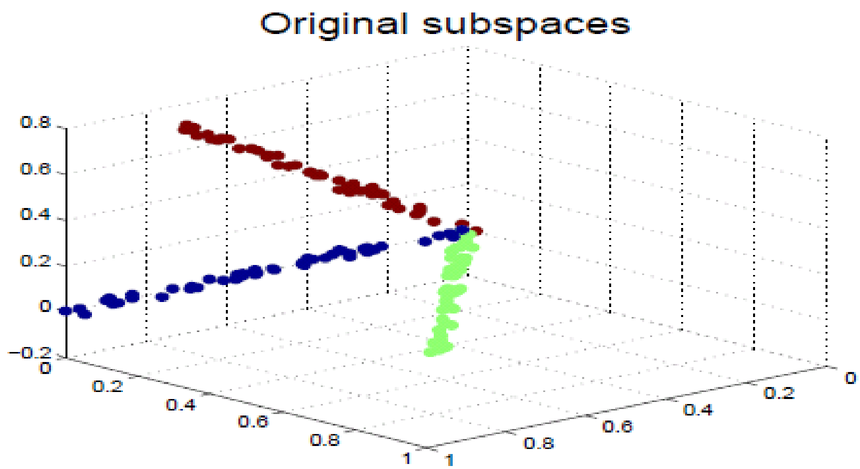
(b) $\mathbf{T} = \begin{bmatrix} 1.00 & 0 \\ 0 & 1.00 \end{bmatrix}$; $\theta_{AB} = 1.57.$



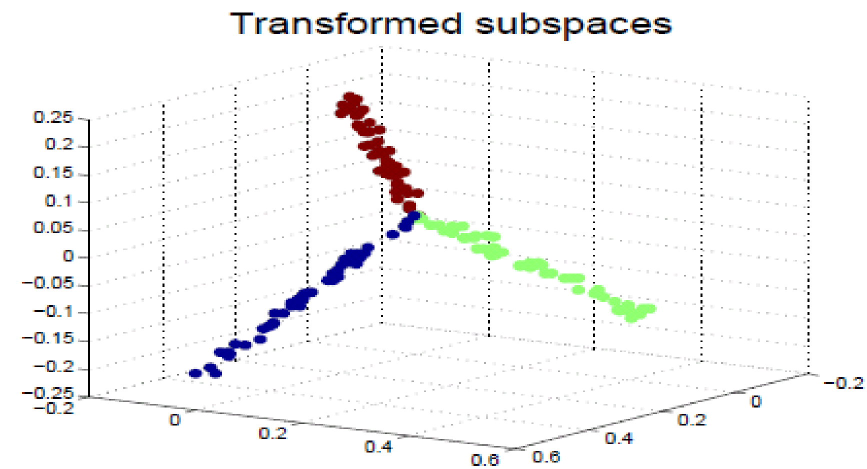
(c) $\theta_{AB} = \frac{\pi}{4} = 0.79,$ $|\mathbf{A}|_* = 1, |\mathbf{B}|_* = 1,$ $|[\mathbf{A}, \mathbf{B}]|_* = 1.41$



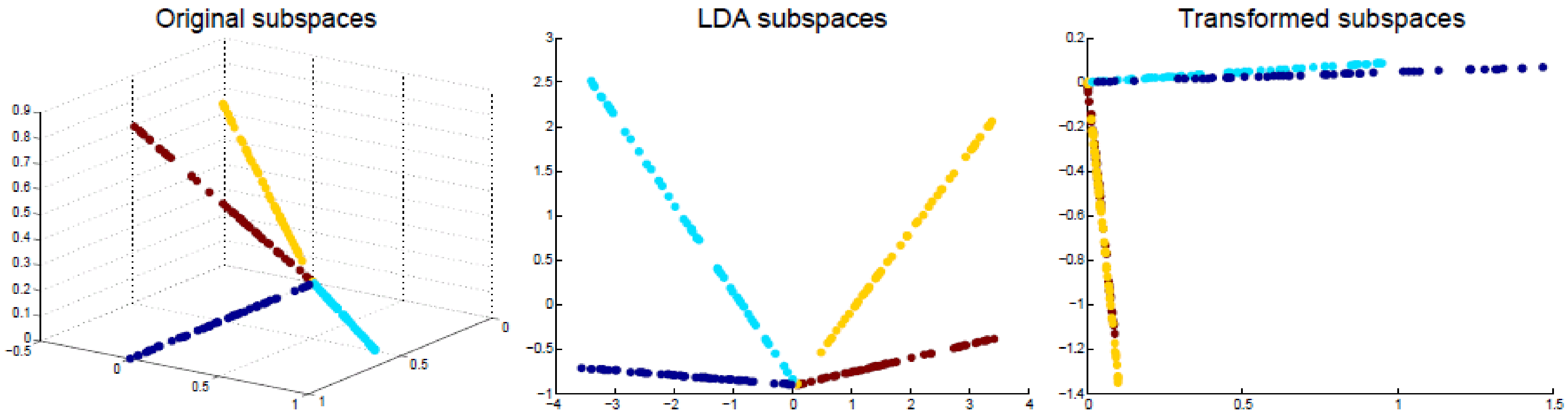
(d) $\mathbf{T} = \begin{bmatrix} 0.50 & -0.21 \\ -0.21 & 0.91 \end{bmatrix}$; $\theta_{AB} = 1.57,$ $|\mathbf{A}|_* = 1, |\mathbf{B}|_* = 1,$ $|[\mathbf{A}, \mathbf{B}]|_* = 1.95$



(e) $\begin{bmatrix} \theta_{AB} = 0.79, & \theta_{AC} = 0.79, & \theta_{BC} = 1.05 \\ \epsilon_A = 0.0141, & \epsilon_B = 0.0131, & \epsilon_C = 0.0148 \end{bmatrix},$ $|\mathbf{A}|_* = 4.06, \quad |\mathbf{B}|_* = 4.08, \quad |\mathbf{C}|_* = 4.16.$



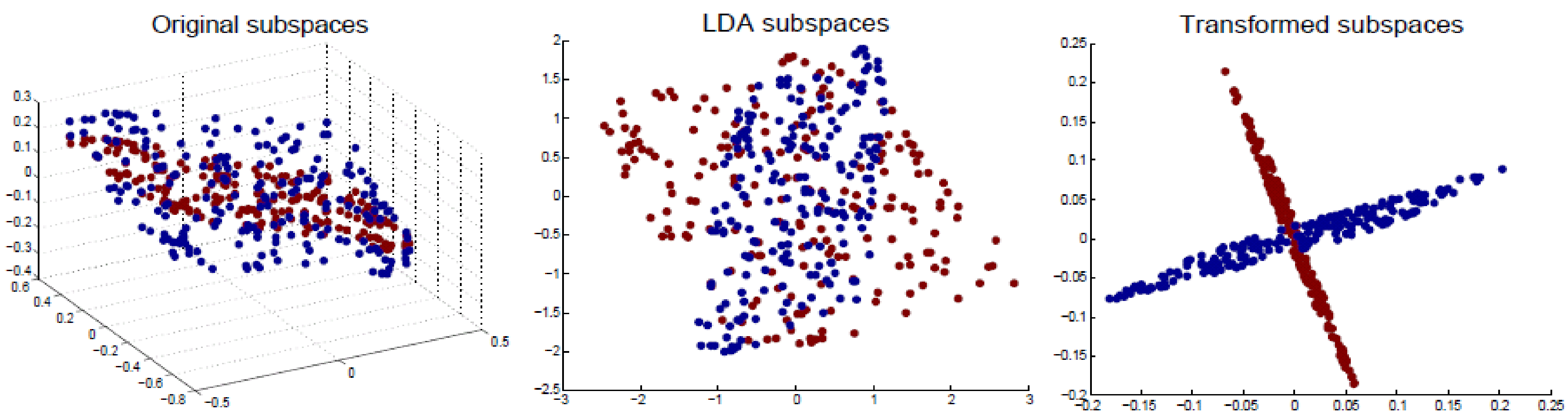
(f) $\mathbf{T} = \begin{bmatrix} 0.39 & -0.16 & -0.16 \\ -0.13 & 0.90 & 0.11 \\ -0.23 & 0.11 & 0.57 \end{bmatrix}$; $\begin{bmatrix} \theta_{AB} = 1.51, & \theta_{AC} = 1.49, & \theta_{BC} = 1.57 \\ \epsilon_A = 0.0091, & \epsilon_B = 0.0085, & \epsilon_C = 0.0114 \end{bmatrix},$ $|\mathbf{A}|_* = 1.93, \quad |\mathbf{B}|_* = 2.37, \quad |\mathbf{C}|_* = 1.20.$



(a) Two classes $\{\mathbf{Y}_+, \mathbf{Y}_-\}$, $\mathbf{Y}_+ = \{\mathbf{A}(blue), \mathbf{B}(cyan)\}$, $\mathbf{Y}_- = \{\mathbf{C}(yellow), \mathbf{D}(red)\}$, $\begin{bmatrix} \theta_{AB} = 1.1, \theta_{AC} = 1.1, \theta_{AD} = 1.1, \\ \theta_{BC} = 1.3, \theta_{BD} = 1.4, \theta_{CD} = 0.5 \end{bmatrix}$, $|\mathbf{Y}_+|_* = 1.58, \quad |\mathbf{Y}_-|_* = 1.27.$

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

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



(d) Two classes $\{\mathbf{A}(\text{blue}), \mathbf{B}(\text{red})\}$, $\theta_{AB} = 0.31$, $|\mathbf{A}|_* = 1.91, |\mathbf{B}|_* = 1.88.$

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

(f) $\mathbf{T} = \begin{bmatrix} 0.49 & -0.11 & 1.27 \\ -0.09 & 0.29 & -0.59 \end{bmatrix}$; $\theta_{AB} = 1.57$, $|\mathbf{A}|_* = 1.08, \quad |\mathbf{B}|_* = 1.03.$

Theorem 1

Theorem 1 *Let* $\mathbf{A}$ *and* B *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathrm{B}]$ *be the concatenation of* $\mathbf{A}$ *and* B*, we have*

$$||[\mathbf{A}, \mathrm{B}]||_* \leq ||\mathbf{A}||_* + ||\mathrm{B}||_*.$$

Proof:

$$||\mathbf{A}||_* + ||\mathrm{B}||_* = ||[\mathbf{A}\ \mathbf{0}]||_* + ||[\mathbf{0}\ \mathrm{B}]||_* \geq ||[\mathbf{A}\ \mathbf{0}] + [\mathbf{0}\ \mathrm{B}]||_* = ||[\mathbf{A}, \mathrm{B}]||_*$$

■

Theorem 2

Theorem 2 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$, *we have*

$$||[\mathbf{A}, \mathbf{B}]||_* = ||\mathbf{A}||_* + ||\mathbf{B}||_*.$$

when the column spaces of $\mathbf{A}$ *and* $\mathbf{B}$ *are orthogonal.*

Proof: We perform the singular value decomposition of $\mathbf{A}$ and $\mathbf{B}$ as

$$\mathbf{A} = [\mathbf{U_{A1}}\mathbf{U_{A2}}]\begin{bmatrix}\mathbf{\Sigma_A} & 0 \\ 0 & 0\end{bmatrix}[\mathbf{V_{A1}}\mathbf{V_{A2}}]', \quad \mathbf{B} = [\mathbf{U_{B1}}\mathbf{U_{B2}}]\begin{bmatrix}\mathbf{\Sigma_B} & 0 \\ 0 & 0\end{bmatrix}[\mathbf{V_{B1}}\mathbf{V_{B2}}]',$$

where the diagonal entries of $\mathbf{\Sigma_A}$ and $\mathbf{\Sigma_B}$ contain non-zero singular values. We have

$$\mathbf{AA}' = [\mathbf{U_{A1}}\mathbf{U_{A2}}]\begin{bmatrix}\mathbf{\Sigma_A}^2 & 0 \\ 0 & 0\end{bmatrix}[\mathbf{U_{A1}}\mathbf{U_{A2}}]', \quad \mathbf{BB}' = [\mathbf{U_{B1}}\mathbf{U_{B2}}]\begin{bmatrix}\mathbf{\Sigma_B}^2 & 0 \\ 0 & 0\end{bmatrix}[\mathbf{U_{B1}}\mathbf{U_{B2}}]'.$$

The column spaces of $\mathbf{A}$ and $\mathbf{B}$ are considered to be orthogonal, i.e., $\mathbf{U_{A1}}'\mathbf{U_{B1}} = 0$. The above can be written as

$$\mathbf{AA}' = [\mathbf{U_{A1}}\mathbf{U_{B1}}]\begin{bmatrix}\mathbf{\Sigma_A}^2 & 0 \\ 0 & 0\end{bmatrix}[\mathbf{U_{A1}}\mathbf{U_{B1}}]', \quad \mathbf{BB}' = [\mathbf{U_{A1}}\mathbf{U_{B1}}]\begin{bmatrix}0 & 0 \\ 0 & \mathbf{\Sigma_B}^2\end{bmatrix}[\mathbf{U_{A1}}\mathbf{U_{B1}}]'.$$

Then, we have

$$[\mathbf{A}, \mathbf{B}][\mathbf{A}, \mathbf{B}]' = \mathbf{AA}' + \mathbf{BB}' = [\mathbf{U_{A1}}\mathbf{U_{B1}}]\begin{bmatrix}\mathbf{\Sigma_A}^2 & 0 \\ 0 & \mathbf{\Sigma_B}^2\end{bmatrix}[\mathbf{U_{A1}}\mathbf{U_{B1}}]'.$$

The nuclear norm $||\mathbf{A}||_*$ is the sum of the square root of the singular values of $\mathbf{AA}'$. Thus, $||[\mathbf{A}, \mathbf{B}]||_* = ||\mathbf{A}||_* + ||\mathbf{B}||_*$. ■

Other Popular Norms?

Proposition 3 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

$$||[\mathbf{A}, \mathbf{B}]||_2 \leq ||\mathbf{A}||_2 + ||\mathbf{B}||_2,$$

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

Proposition 4 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices of the same row dimensions, and* $[\mathbf{A}, \mathbf{B}]$ *be the concatenation of* $\mathbf{A}$ *and* $\mathbf{B}$*, we have*

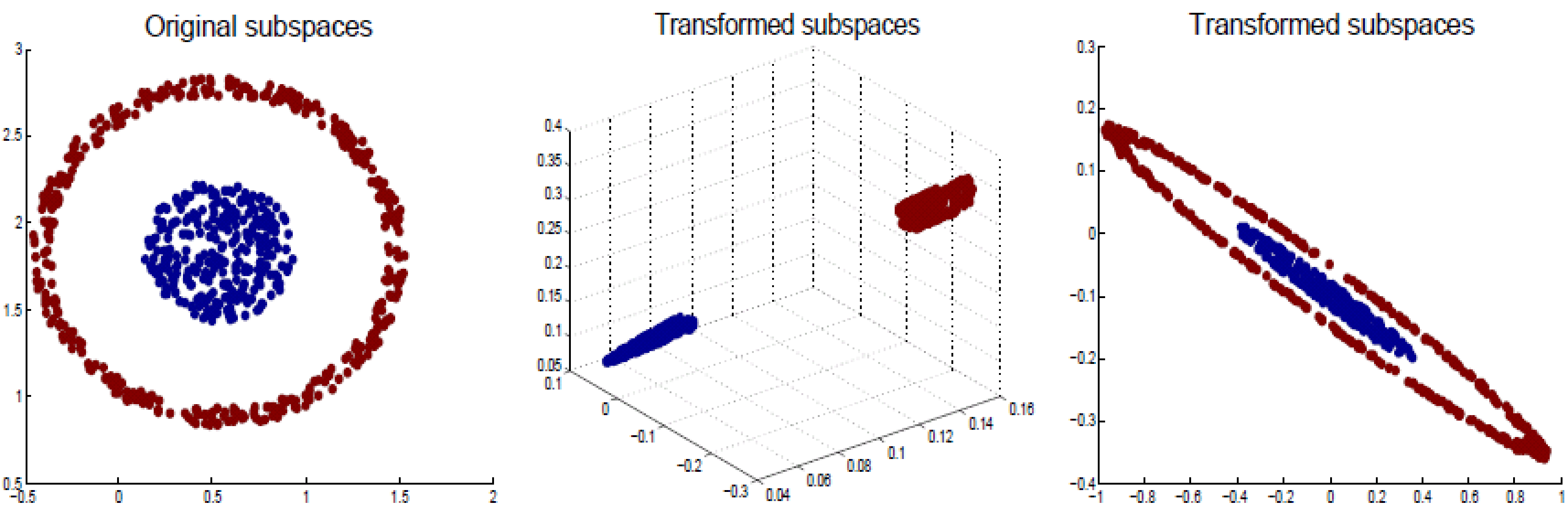
$$||[\mathbf{A}, \mathbf{B}]||_F \leq ||\mathbf{A}||_F + ||\mathbf{B}||_F,$$

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

Kernelized Transform

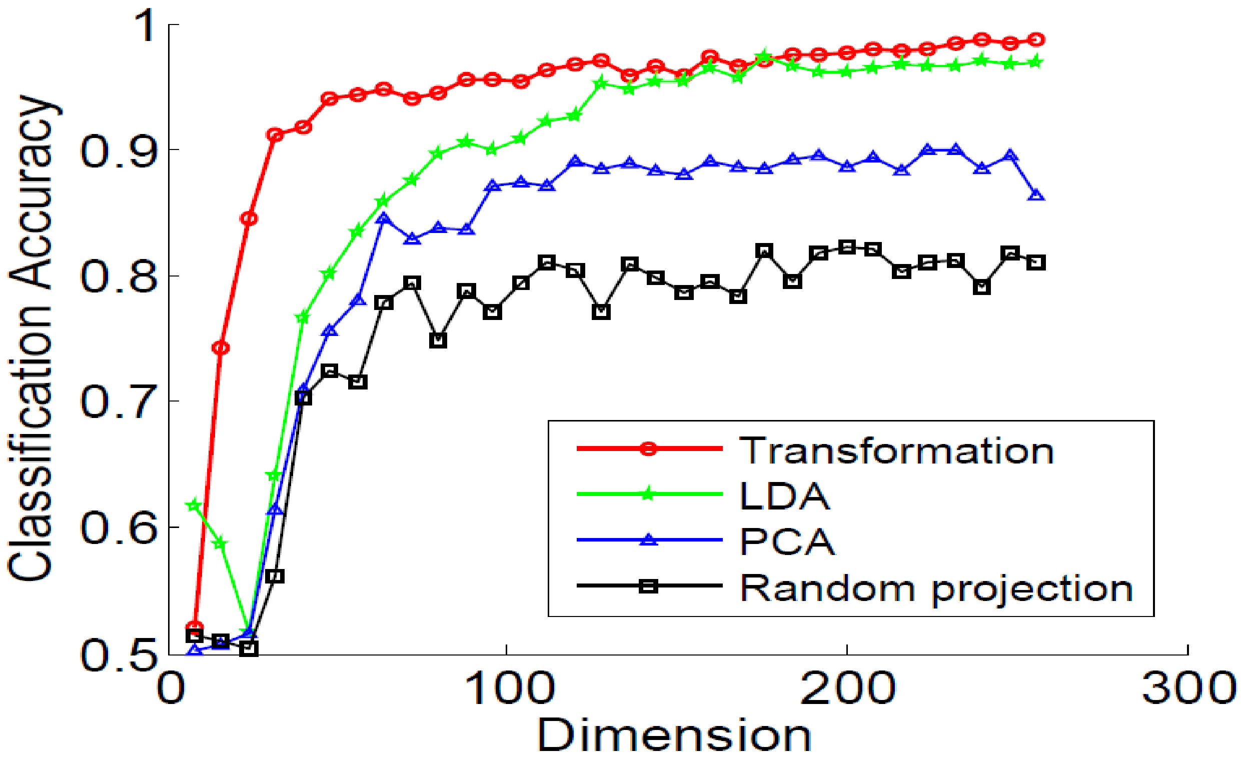
$$\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathcal{K}(\mathbf{Y}_c)||_* - ||\mathbf{T}\mathcal{K}(\mathbf{Y})||_*, \quad \text{s.t. } ||\mathbf{T}||_2 = 1.$$

$$\mathcal{K}(\mathbf{y}) = (\kappa(\mathbf{y}, \mathbf{y}_1); \ldots; \kappa(\mathbf{y}, \mathbf{y}_n))$$



(a) (b) (c)

Transform-based Dimension Reduction



Extended YaleB face dataset

Subspace Clustering using Low-rank Transform

Subspace clustering

Duke University

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.

Robust Sparse Subspace Clustering (R-SSC)

- For the transformed points, we first recover their low-rank representation **L**

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

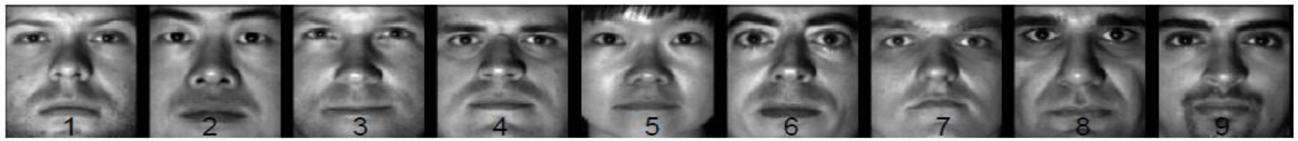
- Each transformed point $\mathbf{Ty}_i$ is then represented using its KNN in **L**, denoted as $\mathbf{L}_i$

$$\arg\min_{\mathbf{x}_i} ||\mathbf{Ty}_i - \mathbf{L}_i\mathbf{x}_i||_2^2 \quad \text{s.t. } \mathbf{1}'\mathbf{x}_i = 1.$$

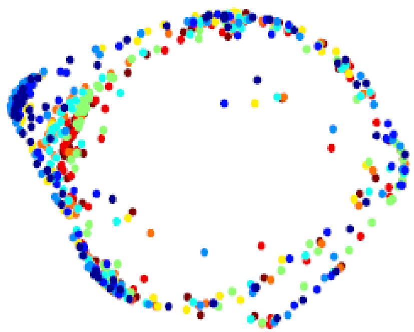
Let $\bar{\mathbf{L}}_i = \mathbf{L}_i - \mathbf{1}\mathbf{Ty}_i^T$, $\quad \mathbf{x}_i = \bar{\mathbf{L}}_i\bar{\mathbf{L}}_i^T \setminus \mathbf{1}$

- Perform spectral clustering on sparse representation matrix (|**X**|+|**X'**|)

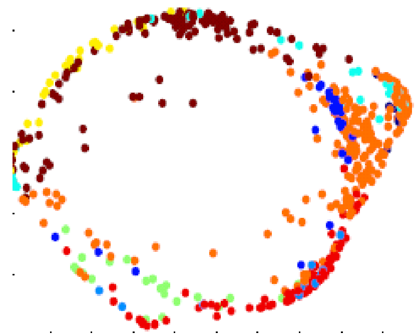
(a) Example illumination conditions.



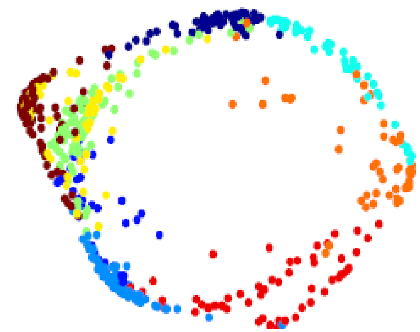
(b) Example subjects.



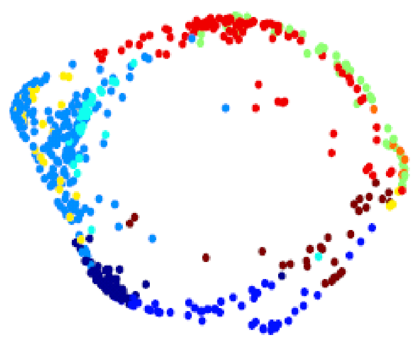
(a) Ground truth.



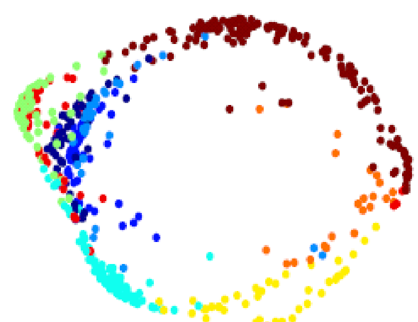
(b) SSC, $e = 71.25\%$, $t = 714.99$ *sec.*



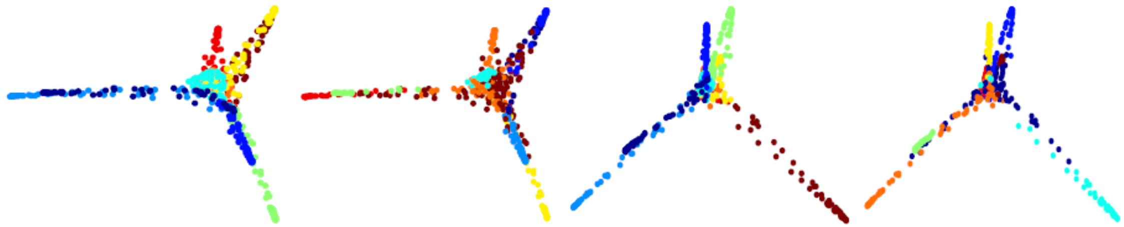
(c) LBF, $e = 76.37\%$, $t = 460.76$ *sec.*



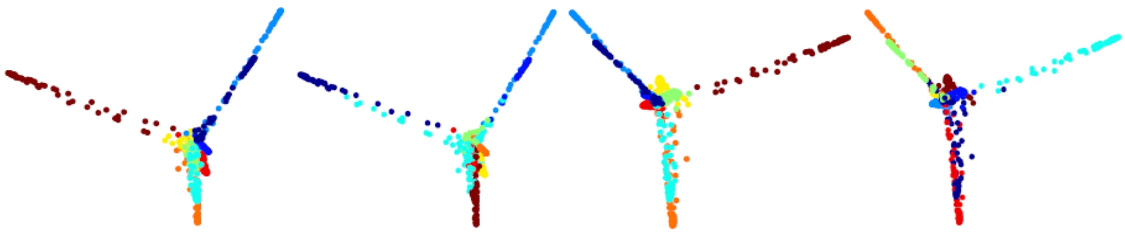
(d) LSA, $e = 71.96\%$, $t = 22.57$ *sec.*



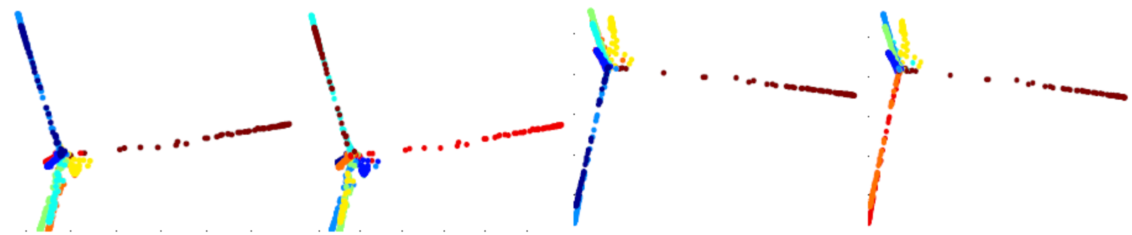
(e) R-SSC, $e = \mathbf{67.37}\%$, $t = \mathbf{1.83}$ *sec.*



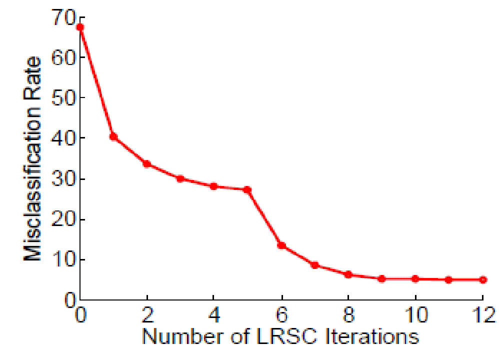
(a) Ground truth (iter=1). (b) $e = 40.39\%$ (iter=1). (c) Ground truth (iter=2). (d) $e = 33.51\%$ (iter=2).



(e) Ground truth (iter=3). (f) $e = 29.98\%$ (iter=3). (g) Ground truth (iter=6). (h) $e = 13.40\%$ (iter=6).



(i) Ground truth (iter=8). (j) $e = 6.17\%$ (iter=8). (k) Ground truth (iter=12). (l) $e = 4.94\%$ (iter=12).



Subsets	[1:10]	[1:15]	[1:20]	[1:25]	[1:30]	[1:38]
C	10	15	20	25	30	38
LSA	78.25	82.11	84.92	82.98	82.32	84.79
LBF	78.88	74.92	77.14	78.09	78.73	79.53
LRSC	**5.39**	**4.76**	**9.36**	**8.44**	**8.14**	**11.02**

Misclassification rate (e%) on clustering different subjects.

Classification using Low-rank Transform

Basic Scheme

- For the c-th class, we first recover its low-rank representation $\mathbf{L}_c$

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

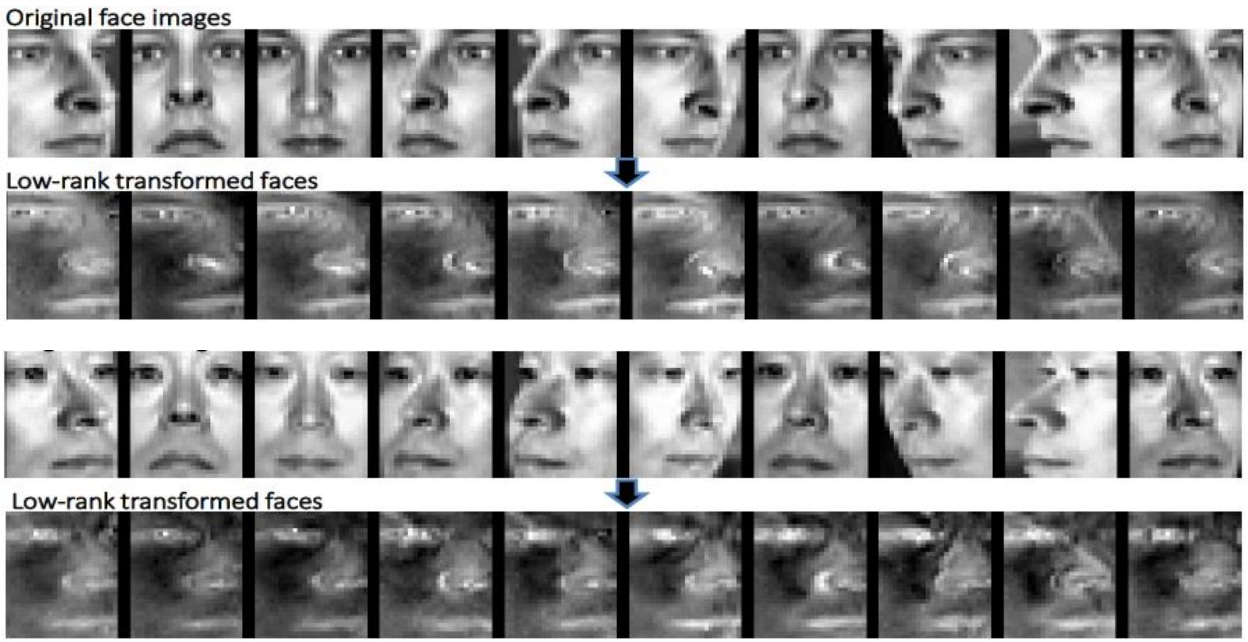
- Each testing point $\mathbf{y}$ is assigned to $\mathbf{L}_c$ that gives the minimal reconstruction error

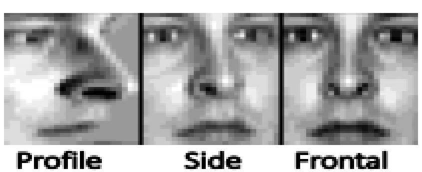
$$\arg\min_{\mathbf{x}} \|\mathbf{Ty} - \mathbf{L}_i\mathbf{x}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_0 \leq T$$

Face recognition across illumination

Method	Accuracy (%)
D-KSVD Zhang and Li (2010)	94.10
LC-KSVD Jiang et al. (2011)	96.70
SRC Wright et al. (2009)	97.20
Original+NN	91.77
Class LRT+NN	97.86
Class LRT+OMP	92.43
Global LRT+NN	99.10
Global LRT+OMP	**99.51**

Face recognition across pose and illumination

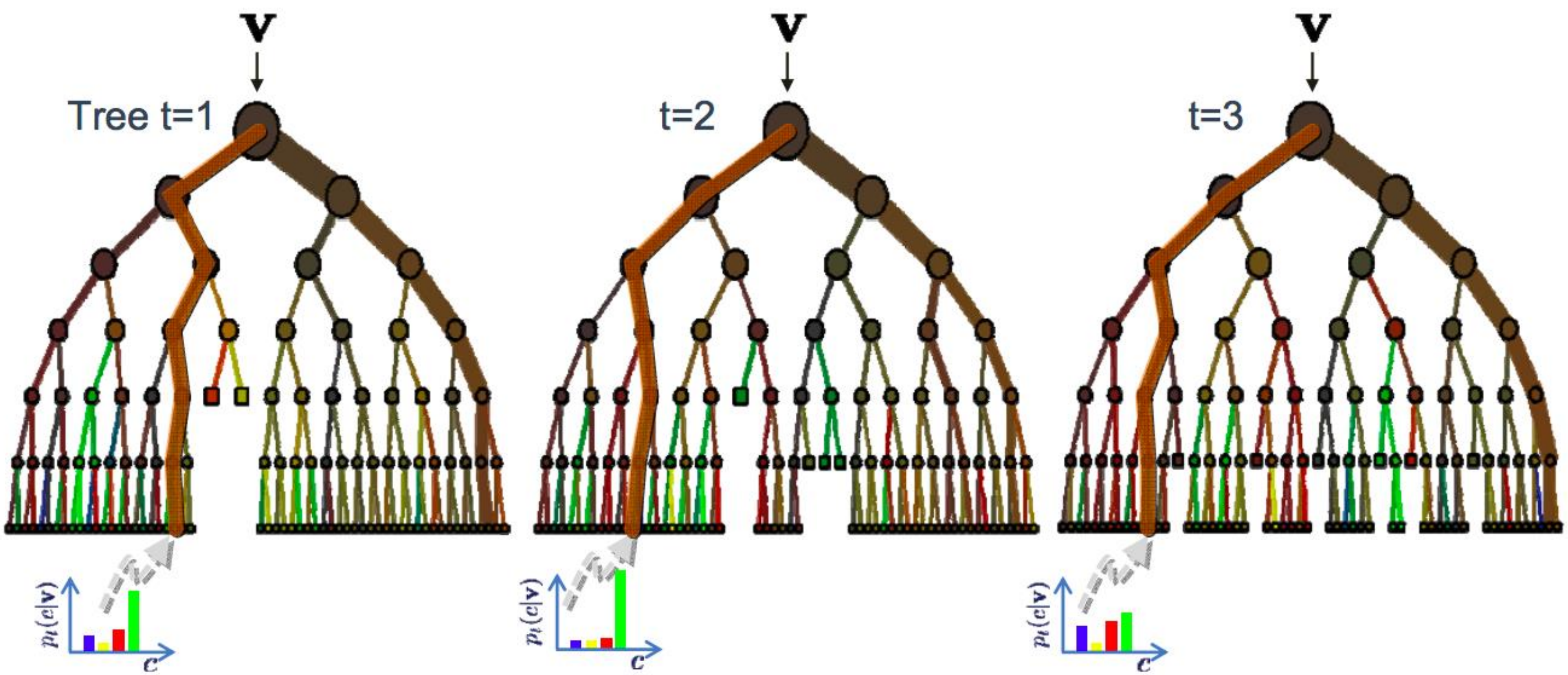




Method	Frontal (c27)	Side (c05)	Profile (c22)
SMD Castillo and Jacobs (2009)	83	82	57
Original+NN	39.85	37.65	17.06
Original(crop+flip)+NN	44.12	45.88	22.94
Class LRT+NN	98.97	96.91	67.65
Class LRT+OMP	**100**	**100**	**67.65**
Global LRT+NN	97.06	95.58	50
Global LRT+OMP	100	98.53	57.35

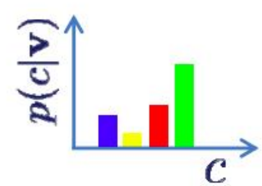
Classification using Transform Forest

Transform Forest



The ensemble model

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



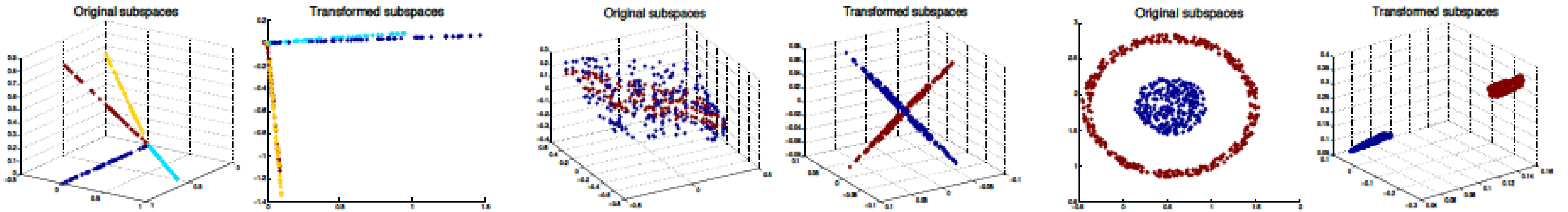
Transform learner

- 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

$$\min_{\mathbf{T}} ||\mathbf{T}\mathcal{K}(\mathbf{Y}^+)||_* + ||\mathbf{T}\mathcal{K}(\mathbf{Y}^-)||_* - ||\mathbf{T}[\mathcal{K}(\mathbf{Y}^+), \mathcal{K}(\mathbf{Y}^-)]||_*,$$
$$\text{s.t. } ||\mathbf{T}||_2 = 1. \qquad (2)$$



(a) (b) (c) (d) (e) (f)

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

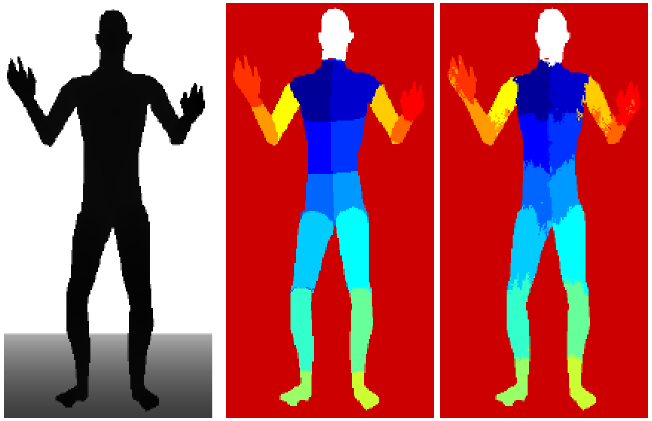
$$\min_{\mathbf{D}^{\pm},\mathbf{Z}^{\pm}} \|\mathbf{X}^{\pm} - \mathbf{D}^{\pm}\mathbf{Z}^{\pm}\| \text{ s.t. } \|\mathbf{z}_i^{\pm}\|_0 \leq l,$$

- The split function is evaluated using the reconstruction error,

$$e^{\pm}(\mathbf{x}) = \|\mathbf{x} - \mathbf{P}^{\pm}\mathbf{x}\|_2$$

where $\mathbf{P}^{\pm} = \mathbf{D}^{\pm}(\mathbf{D}^{\pm T}\mathbf{D}^{\pm})^{-1}\mathbf{D}^{\pm T}$

Results



(a) Depth. (b) Groundtruth. (c) Prediction.

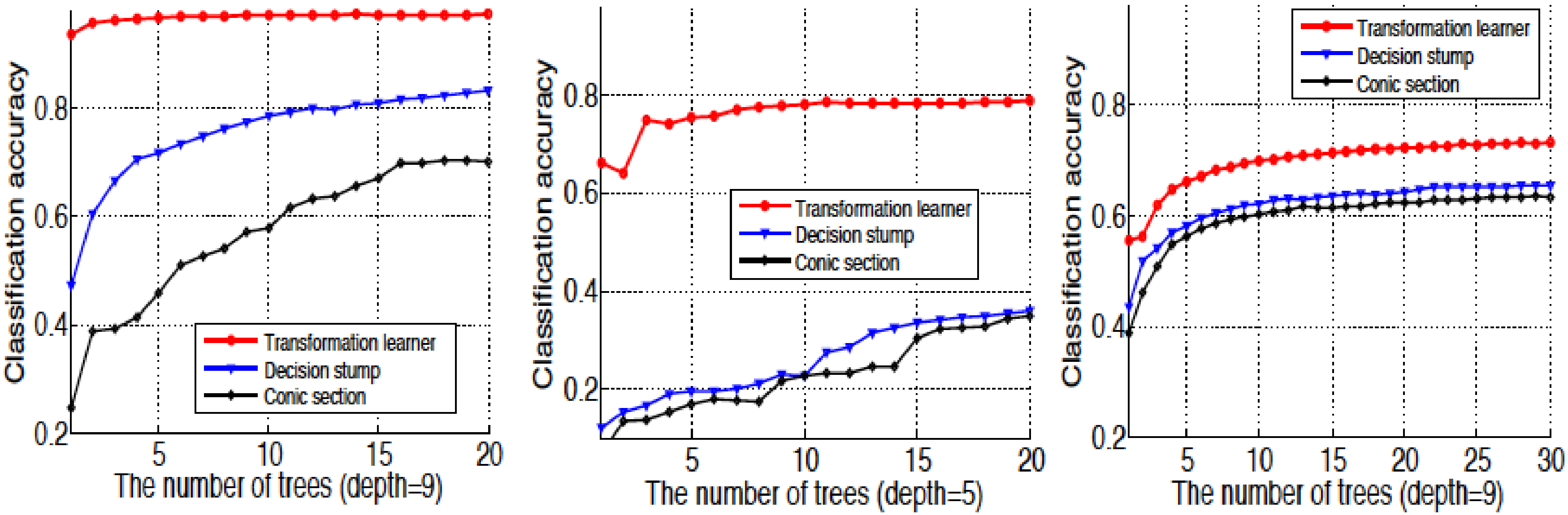
Quantitative Results

Duke UNIVERSITY

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

Extended YaleB face dataset

On the Number of Trees



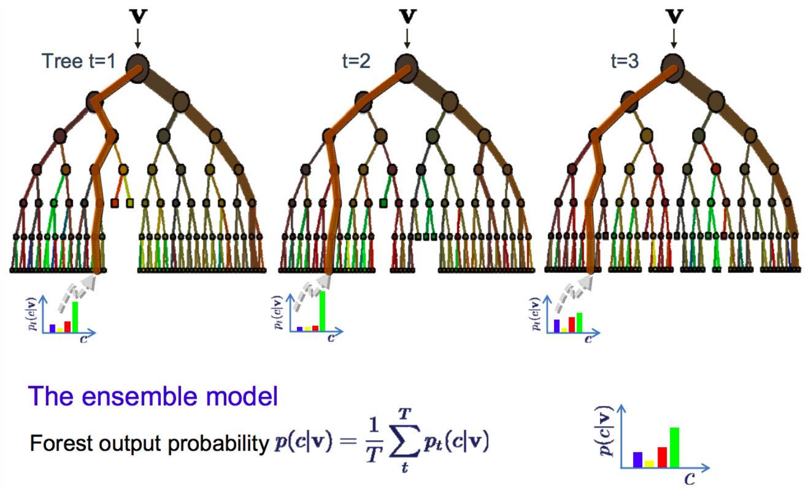
(a) MNIST. (b) 15-Scenes. (c) Kinect.

Hashing using Transform Forest

Motivation

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

ForestHash



We simply set '1' for the visited nodes, and '0' for the rest, obtaining a (2^d-2)-bit hash code.

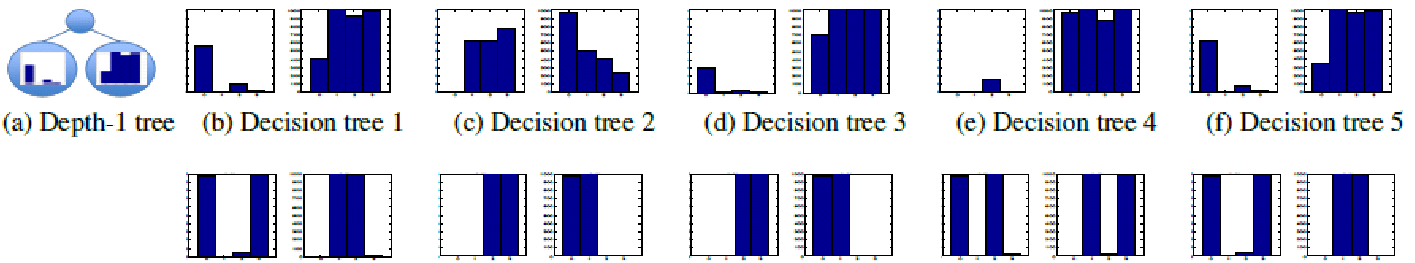
- Challenge 1: Create consistent hash codes in each tree.
  - Low-rank transform.
- Challenge 2: Merging trees for unique codes per class.
  - A mutual information based technique for near-optimal code aggregation.

Challenge 1: Consistent Codes

- Transform learner: learn $\mathbf{T}$ at each split node

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

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



(a) Depth-1 tree (b) Decision tree 1 (c) Decision tree 2 (d) Decision tree 3 (e) Decision tree 4 (f) Decision tree 5

(g) Transform tree 1 (h) Transform tree 2 (i) Transform tree 3 (j) Transform tree 4 (k) Transform tree 5

- *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

Duke UNIVERSITY

- Hash codes from a random forest consisting 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

Duke UNIVERSITY

Table 1. 36-bit retrieval performance (%) on MNIST (rejection hamming radius 0) using different training set sizes. Test time is the average binary encoding time in microseconds (μs).

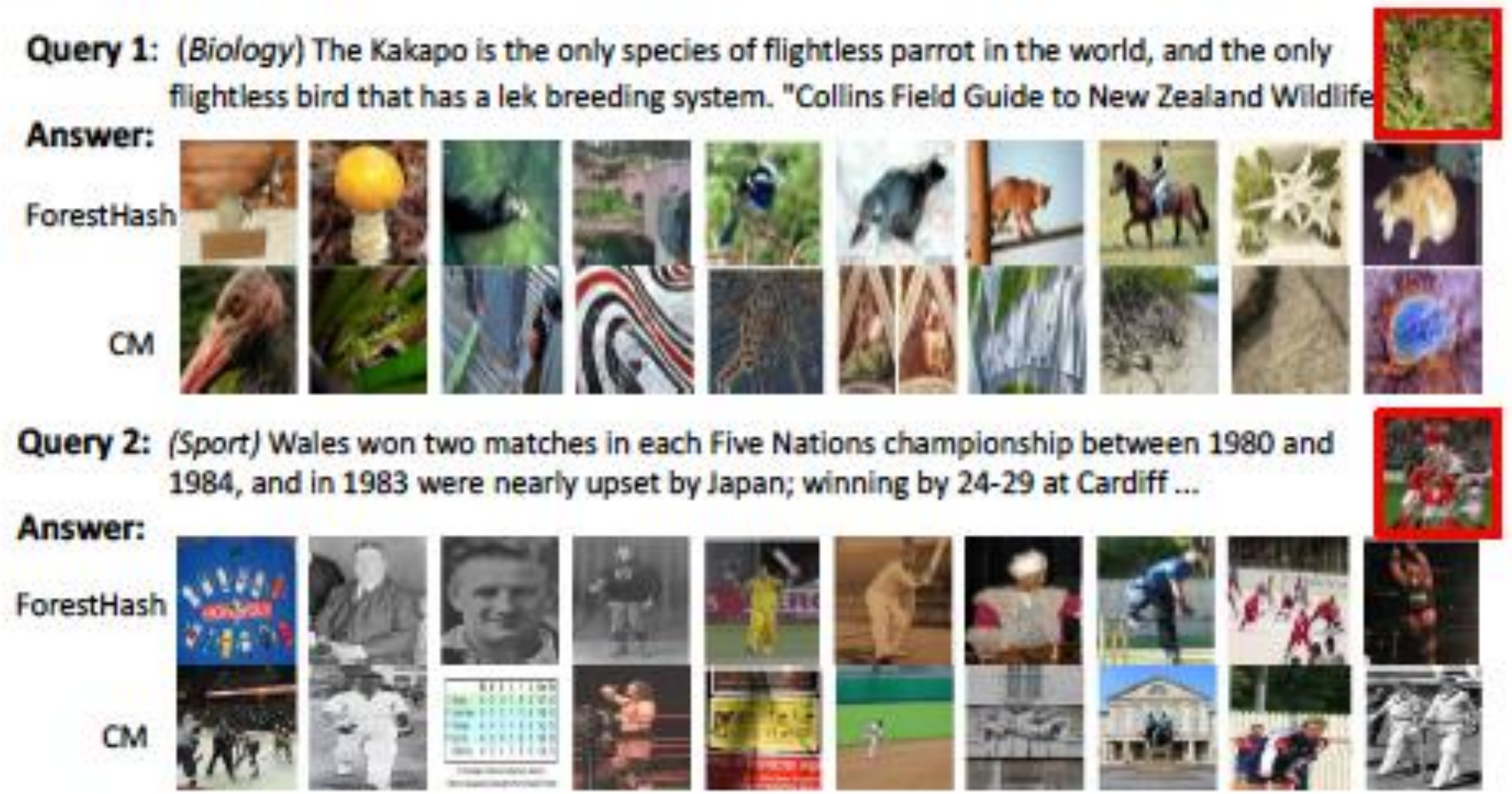
	Test	6,000 samples per class				100 samples per class				30 samples per class			
	time (μs)	Train time (s)	Precision	Recall	NN	Train time (s)	Precision	Recall	NN	Train time (s)	Precision	Recall	NN
HDML [23]	10	93780	**92.94**	60.44	**98.33**	1505	62.52	2.19	83.23	458	24.28	0.21	75.12
DeepNet [30]	52	3407	79.12	0.92	97.52	62.07	63.91	0.39	88.49	23.57	63.71	0.73	81.18
FastHash [19]	115	865	84.70	**76.60**	97.42	213	73.32	33.04	90.95	151	57.08	11.77	82.57
TSH [20]	411	164325	86.30	3.17	97.09	21.08	74.00	5.19	**91.79**	2.83	56.86	3.94	81.84
FaceHash-base	1.2	0.02	20.92	16.24	17.88	0.003	20.10	13.63	14.90	0.002	22.35	13.87	26.11
FaceHash-aggr	**1.2**	**0.02**	20.28	30.09	20.04	**0.003**	21.03	36.57	13.27	**0.002**	17.07	**44.68**	17.86
FaceHash	57	24.20	86.53	46.30	90.78	4.19	**84.98**	**45.00**	90.14	1.43	**79.38**	42.27	**84.49**

HDML and DeepNet are deep learning based hashing methods.

Example 1: Image Retrieval

Cifar-10 dataset

Example 2: Cross-modality

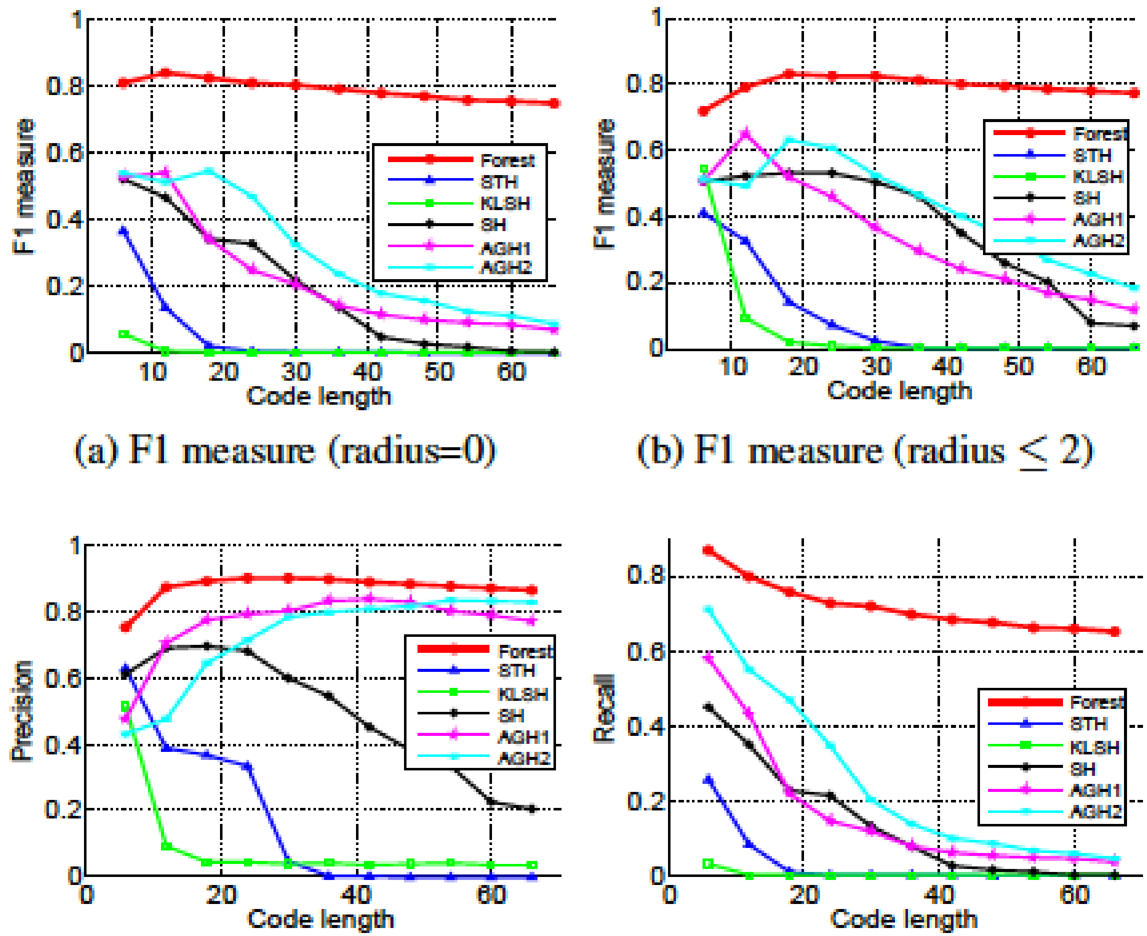


ForestHash (66-bit)	ForestHash (36-bit)	CM-SSH [4]	CM [27]	SM [27]	SCM[27]	MM-NN[18]	CM-NN [18]
50.8	45.5	18.4	19.6	22.3	22.6	27.4	25.8

The Wikipedia dataset

Example 3: Document Retrieval

Duke UNIVERSITY



(a) F1 measure (radius=0)

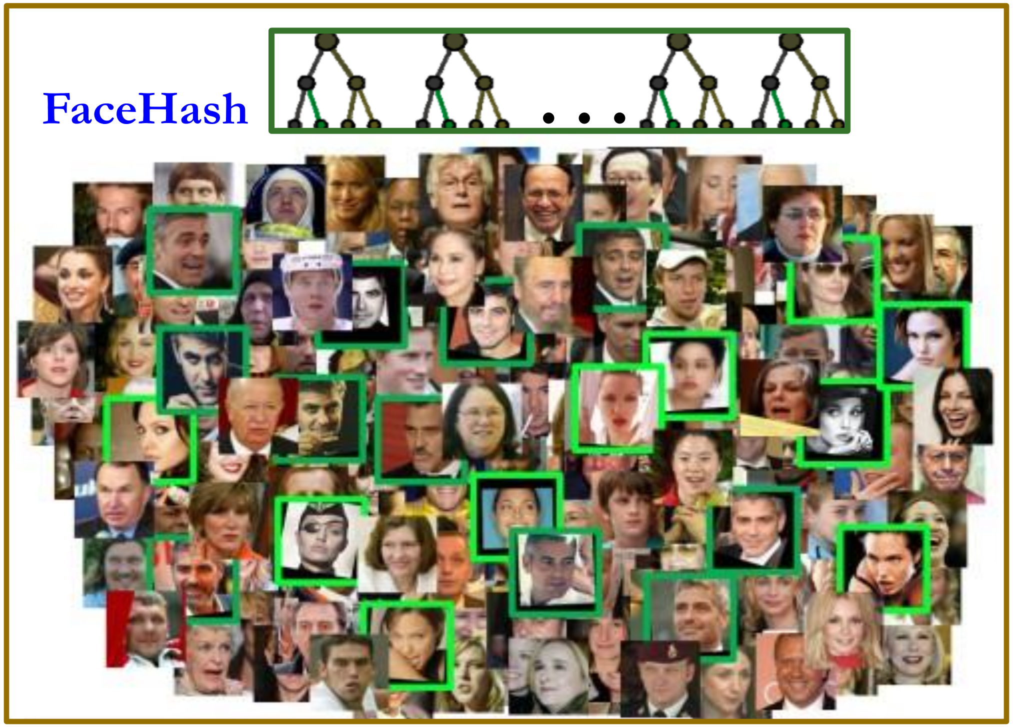
(b) F1 measure (radius $\leq$ 2)

(c) Precision (radius = 0)

(d) Recall (radius = 0)

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

Duke UNIVERSITY

- 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.

Method	radius = 0		radius ≤ 2	
	Precision	Recall	Precision	Recall
SH [31]	6.56	0.15	37.18	1.98
KLSH [15]	16.97	3.73	31.93	8.38
AGH1 [21]	31.74	56.12	17.17	82.30
AGH2 [21]	22.44	57.48	12.17	**89.52**
LDAHash [27]	23.42	0.65	45.30	10.25
FastHash [19]	16.33	3.47	27.87	20.62
TSH [20]	6.75	0.22	9.96	0.35
FaceHash	95.91	**82.29**	88.05	89.38
FaceHash (48-bit)	**96.54**	80.42	**96.45**	87.41

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

Method	radius = 0		radius ≤ 2	
	Precision	Recall	Precision	Recall
KLSH [15]	16.97	3.73	31.93	8.38
AGH1 [21]	18.38	56.12	7.75	82.30
AGH2 [21]	13.56	57.48	5.53	**89.52**
LDAHash [27]	23.42	0.65	45.11	10.25
FastHash [19]	16.33	3.47	27.82	20.62
FaceHash	82.17	**82.29**	47.58	89.38
FaceHash (48-bit)	**90.74**	80.42	**81.74**	87.41

~24 seconds to index all 0.7M faces.
~16 milliseconds to query 0.7M faces.

Example 4: Faces

Example 5: Cross-modality Faces

Duke UNIVERSITY

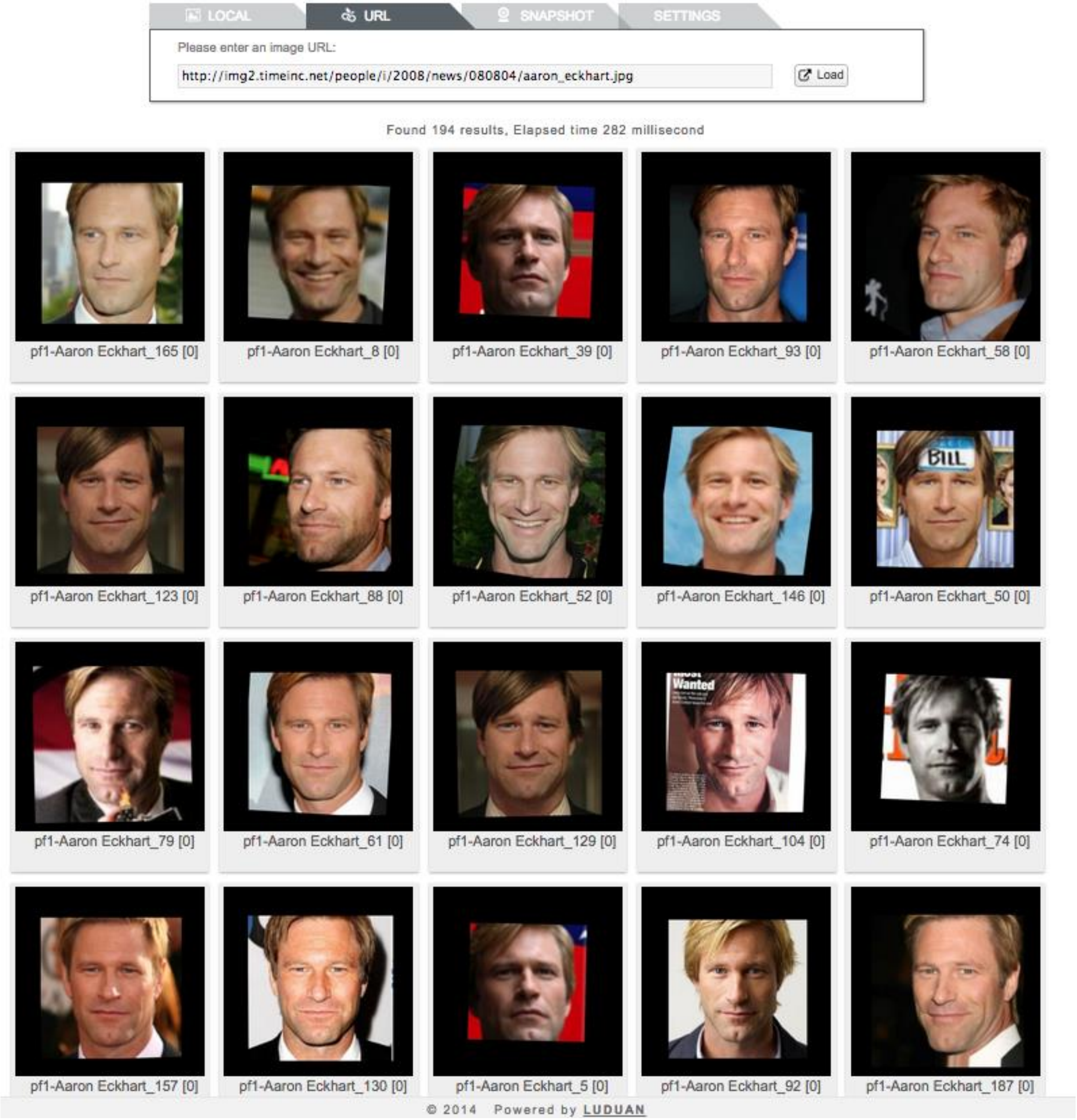
Face attributes

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

	radius = 0		radius ≤ 2	
	Precision	Recall	Precision	Recall
Pubfig dataset	93.12	77.24	95.56	88.01
73K dataset	91.49	77.24	87.25	88.01
0.7M dataset	76.93	77.24	46.09	88.01



Duke UNIVERSITY

Thank you!

Reference

- Qiang Qiu, Guillermo Sapiro, "Learning Transformations for Clustering and Classification", Journal of Machine Learning Research (JMLR), 16(Feb):187−225, 2015
- Qiang Qiu, Guillermo Sapiro, Alex Bronstein, "Random Forests Can Hash", International Conference on Learning Representations (ICLR) Workshop, 2015
- Qiang Qiu and Guillermo Sapiro, "Learning Transformations for Classification Forests", International Conference on Learning Representations (ICLR), 2014
- Qiang Qiu and Guillermo Sapiro, "Learning Compressed Image Classification Features", International Conference on Image Processing, 2014

The Concave-Convex Procedure

- Difference of convex function

$$\mathcal{J}(\mathbf{T}) = \mathcal{J}_{vex}(\mathbf{T}) + \mathcal{J}_{cav}(\mathbf{T})$$
$$= [\sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_*] + [-||\mathbf{T}\mathbf{Y}||_*]$$

- A simple projected subgradient method

$$\sum_{c=1}^{C} \partial||\mathbf{T}\mathbf{Y}_c||_* \mathbf{Y}_c' - \partial||\mathbf{T}^{(t)}\mathbf{Y}||_* \mathbf{Y}'.$$

A subgradient of matrix nuclear norm

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

1. Perform singular value decomposition:
   $\mathbf{A} = \mathbf{U\Sigma V}$;
2. $s \leftarrow$ the number of singular values smaller than δ ,
3. Partition $\mathbf{U}$ and $\mathbf{V}$ as
   $\mathbf{U} = [\mathbf{U}^{(1)}, \mathbf{U}^{(2)}]$, $\mathbf{V} = [\mathbf{V}^{(1)}, \mathbf{V}^{(2)}]$;
   where $\mathbf{U}^{(1)}$ and $\mathbf{V}^{(1)}$ have $(n - s)$ columns.
4. Generate a random matrix $\mathbf{B}$ of the size $(m - n + s) \times s$,
   $\mathbf{B} \leftarrow \frac{\mathbf{B}}{||\mathbf{B}||}$;
5. $\partial||\mathbf{A}||_* \leftarrow \mathbf{U}^{(1)}\mathbf{V}^{(1)'} + \mathbf{U}^{(2)}\mathbf{B}\mathbf{V}^{(2)'}$;
6. Return $\partial||\mathbf{A}||_*$;

end