Riemannian Metric Learning and its Vision Applications

Ruiping Wang
Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS)
Nov. 2, 2019
Outline

- Background
  - Visual feature aggregation
  - Metric learning

- Literature review
  - Image set classification
  - Image recognition (fine-grained)

- Summary
Video classification

Searching criminal suspects

Smart TV-Series Character Shots Retrieval

Automating facial expression analysis

Gesture analysis

Action analysis
Video as Image Set

- Image set classification
  - Unconstrained acquisition conditions
  - Complex appearance variations

Image/frame aggregation
Image recognition

- General recognition
- Fine-grained recognition

Figures courtesy: Wei et al. arXiv1907.03069
Image recognition

- Highly complicated features extraction
  - Deep and multi-channel
  - Aggregation for classifiers

Feature maps

Aggregation
Common representation paradigm

- **Aggregation** to obtain **informative** feature
  - Covariance matrix
  - Gaussian distribution
  - …

Video representation

Image representation

- Feature maps
  - Aggregation
  - $c_1$
  - $c_2$
  - $c_3$
  - $c_n$
Second-order and higher

What is second-order?

Feature vector $x \in \mathbb{R}^d$

- 1st order: $R = x$
- 2nd order: $R = \frac{1}{|c|} \sum_{x \in C} x \otimes x$
- 3rd order: $R = \frac{1}{|c|} \sum_{x \in C} x \otimes x \otimes x$
Metric learning

Sample similarity

- **Euclidean distance**
  
  \[ d(x_1, x_2) = \| x_1 - x_2 \|_2 = \sqrt{(x_1 - x_2)^T (x_1 - x_2)} \]

- **Mahalanobis distance**
  
  \[ d_M(x_1, x_2) = \sqrt{(x_1 - x_2)^T \Sigma^{-1} (x_1 - x_2)} \]

where

\[ \Sigma = \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T \]

\[ \mu = \frac{1}{n} \sum_{i=1}^{n} x_i \]
Metric learning

- Sample similarity
  - Euclidean distance
  - Mahalanobis distance
Metric learning

Basic formulation

- Applying Mahalanobis distance to learn a semi-positive semi-definite (PSD) matrix

\[ d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)} \]

- Relationship with subspace learning

\[ d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)} = \sqrt{(x_i - x_j)^T W^T W (x_i - x_j)} = \|Wx_i - Wx_j\|_2 \]

where \( M = W^T W \)
Metric learning

- Large Margin Nearest Neighbor

  □ Cost function

  \[
  \epsilon(L) = \sum_{ij} \eta_{ij} \|L(x_i - x_j)\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) [1 + \|L(x_i - x_j)\|^2 - \|L(x_i - x_l)\|^2]_+
  \]

  □ Objective function: semidefinite programming (SDP)

  Minimize \[
  \sum_{ij} \eta_{ij} (x_i - x_j)^T M(x_i - x_j) + c \sum_{ijl} \eta_{ij}(1 - y_{il})\xi_{ijl}
  \]

  subject to

  \[
  (1) (x_i - x_l)^T M(x_i - x_l) - (x_i - x_j)^T M(x_i - x_j) \geq 1 - \xi_{ijl}
  \]

  \[
  (2) \xi_{ijl} \geq 0
  \]

  \[
  (3) M \succeq 0
  \]

Information-Theoretic Metric Learning (ITML)

- **Distance metric learning problem**

  \[
  \min_A \quad \text{KL} \left( p(x; A_0) \parallel p(x; A) \right) \\
  \text{subject to} \quad d_A(x_i, x_j) \leq u \quad (i, j) \in S, \\
  d_A(x_i, x_j) \geq l \quad (i, j) \in D.
  \]

  where \( \text{KL}(p(x; A_0) \parallel p(x; A)) = \int p(x; A_0) \log \frac{p(x; A_0)}{p(x; A)} \, dx \)

- **Optimization problem can be reformulated as**

  \[
  \min_{A \succeq 0} \quad D_{ld}(A, A_0) \\
  \text{s.t.} \quad \text{tr}(A(x_i - x_j)(x_i - x_j)^T) \leq u \quad (i, j) \in S, \\
  \text{tr}(A(x_i - x_j)(x_i - x_j)^T) \geq l \quad (i, j) \in D.
  \]

  where \( D_{ld}(A, A_0) = \text{tr}(AA_0^{-1}) - \log \det(AA_0^{-1}) - n \)

Metric learning: linear vs. nonlinear

**Linear**

- \( f: \mathbf{x} \rightarrow \mathbf{y}, \mathbf{y} = W \mathbf{x} \) (\( \mathbf{x} \in \mathbb{R}^D, \mathbf{y} \in \mathbb{R}^d, W \in \mathbb{R}^{d \times D} \))

**Nonlinear**

- \( f(.) \) is nonlinear mapping, or \( \mathbf{x}, \mathbf{y} \) is in non-Euclidean space
- **Riemannian metric learning**
  - \( \mathbf{x} \in M \) is element on some Riemannian manifold \( M \)

**Kernel-based**

**Manifold-to-manifold**
Metric learning: linear vs. nonlinear

Linear

- \( f: x \rightarrow y, y = Wx \) (\( x \in \mathbb{R}^D, y \in \mathbb{R}^d, W \in \mathbb{R}^{d \times D} \))

Nonlinear

- \( f(.) \) is nonlinear mapping, or \( x, y \) is in non-Euclidean space
- \textbf{Riemannian metric learning}
  - \( x \in M \) is element on some Riemannian manifold \( M \)
- \textbf{Hash learning} (\textit{a.k.a.} binary code learning)
  - \( y \in \{0,1\}^K \) is element in \( K \)-dimensional Hamming space
Outline

- Background
- Literature review
- Evaluations
- Summary
Task 1: Image set classification

From the view of set modeling

- Linear subspace
  - Yamaguchi, FG'98
  - Kim, PAMI'07
  - Hamm, ICML'08
  - Harandi, CVPR'11
  - Huang, CVPR'15

- Nonlinear manifold
  - Hadid, FG'04
  - Kim, BMVC'05
  - Wang, CVPR'08/09
  - Chen, CVPR'13
  - Lu, CVPR'15

- Affine/Convex hull
  - Cevikalp, CVPR'10
  - Hu, CVPR'11
  - Yang, FG'13
  - Zhu, ICCV'13
  - Wang, ACCV'16

- Statistics
  - Shakhnarovich, ECCV'02
  - Arandjelović, CVPR'05
  - Wang, CVPR'12
  - Harandi, ECCV'14/ICCV'15
  - Wang, CVPR'15/CVPR'17
Task 2: Image recognition

- Second order representation learning pipeline
Task 2: Image recognition

From the view of channel-set modeling

- Bilinear
  [Lin, ICCV’15]
  [Lin, BMVC’17]

- Covariance
  [Li, ICCV’17]
  [Li, CVPR’18]

- Gaussian
  [Wang, ICCV’15]

- Tensor sketch
  [Gao, CVPR’16]
Task 1: Image set classification

From the view of set modeling

- Linear subspace
  - [Yamaguchi, FG’98]
  - [Kim, PAMI’07]
  - [Hamm, ICML’08]
  - [Harandi, CVPR’11]
  - [Huang, CVPR’15]

- Nonlinear manifold
  - [Hadid, FG’04]
  - [Kim, BMVC’05]
  - [Wang, CVPR’08/09]
  - [Chen, CVPR’13]

- Affine/Convex hull
  - [Cevikalp, CVPR’10]
  - [Hu, CVPR’11]
  - [Yang, FG’13]
  - [Zhu, ICCV’13]
  - [Wang, ACCV’16]

- Statistics
  - [Shakhnarovich, ECCV’02]
  - [Arandjelović, CVPR’05]
  - [Wang, CVPR’12]
  - [Harandi, ECCV’14/ICCV’15]
  - [Wang, CVPR’15/CVPR’17]
Overview of previous works

- **Set modeling**
  - Linear subspace $\rightarrow$ Nonlinear manifold
  - Affine/Convex Hull (affine subspace)
  - Parametric PDFs $\rightarrow$ high-order statistics

- **Set matching—basic distance**
  - Principal angles-based measure
  - Nearest neighbor (NN) matching approach
  - K-L divergence $\rightarrow$ SPD Riemannian metric...

- **Set matching—metric learning**
  - Learning in Euclidean space
  - Learning on Riemannian manifold
Set model I: linear subspace

Properties
- PCA on the set of image samples to get subspace
- Loose characterization of the set distribution region
- Principal angles-based measure discards the varying importance of different variance directions

Methods
- MSM [FG’98]
- DCC [PAMI’07]
- GDA [ICML’08]
- GGDA [CVPR’11]
- PML [CVPR’15]
- LieNet [CVPR’17]
Set model I: linear subspace

- MSM (Mutual Subspace Method) [FG’98]
  - Pioneering work on image set classification
  - First exploit principal angles as subspace distance
  - Metric learning: N/A

\[
\cos^2 \theta = \sup_{d \in D, g \in G, \|d\| \neq 0, \|g\| \neq 0} \frac{\|(d, g)\|^2}{\|d\|^2 \|g\|^2}
\]

Set model I: linear subspace

- DCC (Discriminant Canonical Correlations) [PAMI’07]
  - Metric learning: in Euclidean space

Set 1: $X_1$

Set 2: $X_2$

Linear subspace by:
orthonormal basis matrix

$X_iX_i^T \approx P_i\Lambda_iP_i^T$

DCC

- Canonical Correlations/Principal Angles
  - Canonical vectors \(\rightarrow\) common variation modes

Set 1: \(X_1\)

\[
P_1^T P_2 = Q_{12} \Lambda Q_{21}^T
\]

\[
U = P_1 Q_{12} = [u_1, \ldots, u_2]
\]

\[
V = P_2 Q_{21} = [v_1, \ldots, v_2]
\]

\[
\Lambda = diag(\cos \theta_1, \ldots, \cos \theta_d)
\]

Canonical Correlation: \(\cos \theta_i\)

Principal Angles: \(\theta_i\)
DCC

- Discriminative learning
  - Linear transformation
    - $T: X_i \rightarrow Y_i = T^T X_i$
  - Representation
    - $Y_i Y_i^T = (T^T X_i) (T^T X_i)^T \\ \approx (T^T P_i) \Lambda_i (T^T P_i)^T$
  - Set similarity
    - $F_{ij} = \max_{Q_{ij}, Q_{ji}} tr(M_{ij})$
      - $M_{ij} = Q_{ij}^T P_i^T T T^T P_j^T Q_{ji}$
  - Discriminant function
    - $T = \max_{\arg T} tr(T^T S_b T) / tr(T^T S_w T)$
Set model I: linear subspace

- GDA [ICML’08] / GGDA [CVPR’11]
  - Treat subspaces as points on Grassmann manifold
  - Metric learning: on Riemannian manifold


GDA/GGDA

- Projection metric

\[ d_P(Y_1, Y_2) = \left( \sum_i \sin^2(\theta_i) \right)^{1/2} = 2^{-1/2} \left\| Y_1 Y_1^T - Y_2 Y_2^T \right\|_F \]

\[ \theta_i: \text{Principal angles} \]

Geodesic distance: (Wong, 1967; Edelman et al., 1999)

\[ d_G^2(Y_1, Y_2) = \sum_i \theta_i^2 \]
GDA/GGDA

- Projection kernel
  - Projection embedding (isometric)
    - $\Psi_P : G(m, D) \rightarrow \mathbb{R}^{D \times D}$, $\text{span}(Y) \rightarrow YY^T$
  - The inner-product of $\mathbb{R}^{D \times D}$
    - $\text{tr}((Y_1Y_1^T)(Y_2Y_2^T)) = \|Y_1^TY_2\|_F^2$
- Grassmann kernel (positive definite kernel)
  - $k_P(Y_1, Y_2) = \|Y_1^TY_2\|_F^2$
GDA/GGDA

- Discriminative learning
  - Classical kernel methods using the Grassmann kernel
    - e.g., Kernel LDA / kernel Graph embedding

  \[ \alpha^* = \arg \max_{\alpha} \frac{\alpha^T KWK \alpha}{\alpha^T KK \alpha} \]

  Grassmann kernel
Set model I: linear subspace

- PML (Projection Metric Learning) [CVPR’15]
  - Metric learning: on Riemannian manifold

Explicit manifold to manifold mapping

- \( f(Y) = W^T Y \in \mathcal{G}(q, d), Y \in \mathcal{G}(q, D), \ d \leq D \)

Projection metric on target Grassmann manifold \( \mathcal{G}(q, d) \)

- \( d_p^2(f(Y_i), f(Y_j)) = 2^{-1/2} \| (W^T Y'_i)(W^T Y'_i)^T - (W^T Y'_j)(W^T Y'_j)^T \|_F^2 = 2^{-1/2} \text{tr}(P^T A_{ij} A_{ij} P) \)

- \( A_{ij} = (Y'_i Y'_i^T - Y'_j Y'_j)^T \), \( P = WW^T \) is a rank-\( d \) symmetric positive semidefinite (PSD) matrix of size \( D \times D \) (similar form as Mahalanobis matrix)

- \( Y_i \) needs to be normalized to \( Y'_i \) so that the columns of \( W^T Y_i \) are orthonormal
Discriminative learning

- Discriminant function
  - Minimize/Maximize the projection distances of any within-class/between-class subspace pairs
  - \( J = \min \sum_{l_i=l_j} tr(P^T A_{ij} A_{ij} P) - \lambda \sum_{l_i \neq l_j} tr(P^T A_{ij} A_{ij} P) \)

- Optimization algorithm
  - Iterative solution for one of \( Y' \) and \( P \) by fixing the other
  - Normalization of \( Y \) by QR-decomposition
  - Computation of \( P \) by Riemannian Conjugate Gradient (RCG) algorithm on the manifold of PSD matrices
Set model I: linear subspace

- LieNet [CVPR’17]
  - Metric learning: Lie group nonlinear learning in deep networks

LieNet

Basic idea: **respect manifold property** of Lie group structure in nonlinear transformation of deep networks

- Rotation mapping (**RotMap**) layer
  - Fully connected convolution-like
  - Transform and align the rotation matrices

- Rotation pooling (**RotPooling**) layer
  - Reduce Lie group dimension
  - Both spatial and temporal pooling

- Logarithm mapping (**LogMap**) layer
  - Manifold to Euclidean space
Evaluations

- **G3D-Gaming** [Bloom, CVPR workshop’12]
  - 20 motions, 663 sequences
  - 3D coordinates of 20 joints

- **HDM05** [Muller, Tech. Rep.’07]
  - 130 actions, 2,337 sequences
  - 3D coordinates of 31 joints

- **NTU RGB+D** [Shahroudy, CVPR’16]
  - 60 actions, 56,000 sequences
  - 3D coordinates of 25 joints
## Evaluations

- **Performance comparisons**

<table>
<thead>
<tr>
<th>Method</th>
<th>G3D-Gaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM+HMM [32]</td>
<td>86.40%</td>
</tr>
<tr>
<td>SE [41]</td>
<td>87.23%</td>
</tr>
<tr>
<td>SO [42]</td>
<td>87.95%</td>
</tr>
<tr>
<td>LieNet-0Block</td>
<td>84.55%</td>
</tr>
<tr>
<td>LieNet-1Block</td>
<td>85.16%</td>
</tr>
<tr>
<td>LieNet-2Blocks</td>
<td>86.67%</td>
</tr>
<tr>
<td>LieNet-3Blocks</td>
<td><strong>89.10%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>RGB+D-subject</th>
<th>RGB+D-view</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBRNN [13]</td>
<td>59.07%</td>
<td>63.97%</td>
</tr>
<tr>
<td>Deep RNN [37]</td>
<td>56.29%</td>
<td>64.09%</td>
</tr>
<tr>
<td>Deep LSTM [37]</td>
<td>60.69%</td>
<td>67.29%</td>
</tr>
<tr>
<td>PA-LSTM [37]</td>
<td>62.93%</td>
<td>70.27%</td>
</tr>
<tr>
<td>ST-LSTM [26]</td>
<td><strong>69.2%</strong></td>
<td><strong>77.7%</strong></td>
</tr>
<tr>
<td>SE [41]</td>
<td>50.08%</td>
<td>52.76%</td>
</tr>
<tr>
<td>SO [42]</td>
<td>52.13%</td>
<td>53.42%</td>
</tr>
<tr>
<td>LieNet-0Block</td>
<td>53.54%</td>
<td>54.78%</td>
</tr>
<tr>
<td>LieNet-1Block</td>
<td>56.35%</td>
<td>60.14%</td>
</tr>
<tr>
<td>LieNet-2Blocks</td>
<td>58.02%</td>
<td>62.52%</td>
</tr>
<tr>
<td>LieNet-3Blocks</td>
<td>61.37%</td>
<td>66.95%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>HDM05</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPDNet [18]</td>
<td>61.45%±1.12</td>
</tr>
<tr>
<td>SE [41]</td>
<td>70.26%±2.89</td>
</tr>
<tr>
<td>SO [42]</td>
<td>71.31%±3.21</td>
</tr>
<tr>
<td>LieNet-0Block</td>
<td>71.26%±2.12</td>
</tr>
<tr>
<td>LieNet-1Block</td>
<td>73.35%±1.14</td>
</tr>
<tr>
<td>LieNet-2Blocks</td>
<td><strong>75.78%±2.26</strong></td>
</tr>
</tbody>
</table>
Set model IV: statistics (COV+)

- Properties
  - The natural raw statistics of a sample set
  - Flexible model of multiple-order statistical information

- Methods
  - CDL [CVPR’12]
  - LMKML [ICCV’13]
  - DARG [CVPR’15]
  - B. Gauss [ICCV’15]
  - SPD-ML [ECCV’14]
  - LEML [ICML’15]
  - DCRL [CVPR’17]
  - SPDNet [AAAI’17]
  - DHH [TIP’19]
Set model IV: statistics (COV+)

- CDL (Covariance Discriminative Learning) [CVPR’12]
  - Set modeling by Covariance Matrix (COV)
    - The 2nd order statistics characterizing set data variations
    - Robust to noisy set data, scalable to varying set size
  - Metric learning: on the SPD manifold

Set modeling by Covariance Matrix

- Image set: $N$ samples with $d$-dimension image feature

$$X = [x_1, x_2, \ldots, x_N]_{d \times N}$$

- COV: $d*d$ symmetric positive definite (SPD) matrix

$$C = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

*: use regularization to tackle singularity problem
Set matching on COV manifold

- **Riemannian metrics** on the SPD manifold
  - Affine-invariant distance (AID) \[1\]
    \[
d^2(C_1, C_2) = \sum_{i=1}^{d} \ln^2 \lambda_i(C_1, C_2)
    \]
or
    \[
d^2(C_1, C_2) = \left\| \log_I(C_1^{-1/2}C_2C_1^{-1/2}) \right\|_F^2
    \]
  - Log-Euclidean distance (LED) \[2\]
    \[
d(C_1, C_2) = \left\| \log_I(C_1) - \log_I(C_2) \right\|_F
    \]

---


Set matching on COV manifold (cont.)

- Explicit Riemannian kernel feature mapping with LED

\[ \Psi_{log} : C \rightarrow \log_I(C), \quad (M \in \mathbb{R}^{d \times d}) \]

\[ k_{log}(C_1, C_2) = trace[\log_I(C_1) \cdot \log_I(C_2)] \]

Mercer's theorem

Tangent space at Identity matrix \( I \)

Riemannian manifold of non-singular COV
Discriminative learning on COV manifold

- **Partial Least Squares (PLS)** regression
- **Goal**: Maximize the covariance between observations and class labels

**Linear Projection**

- \( X = T \cdot P' \)
- \( Y = T \cdot B \cdot C' = X \cdot B_{pls} \)
- \( T \) is the common latent representation
CDL vs. GDA

COV $\rightarrow$ SPD manifold

- **Model**
  
  $$C = \frac{1}{N - 1} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

- **Metric**
  
  $$d(C_1, C_2) = \|\log_I(C_1) - \log_I(C_2)\|_F$$

- **Kernel**
  
  $$\Psi_{log} : C \rightarrow \log_I(C), \ (M a R^{d\times d})$$

Subspace $\rightarrow$ Grassmannian

- **Model**
  
  $$C = U \Lambda U^T$$

  $$\Rightarrow U = [u_1, u_2, \ldots, u_m]_{D\times m}^*$$

- **Metric**
  
  $$d_{proj}(U_1, U_2) = 2^{-1/2} \left\| U_1 U_1^T - U_2 U_2^T \right\|_F$$

- **Kernel**
  
  $$\Psi_{proj} : U \rightarrow UU^T, \ G(m, D) \rightarrow R^{d\times d}$$
Set model IV: statistics (COV+)

- LMKML (Localized Multi-Kernel Metric Learning) [ICCV’13]
  - Exploring multiple order statistics
    - Data-adaptive weights for different types of features
    - Ignoring the geometric structure of 2\textsuperscript{nd}/3\textsuperscript{rd}-order statistics
  - Metric learning: in Euclidean space

\[ m = \frac{1}{n} \sum_{i=1}^{n} x_i \quad C = \frac{1}{n-1} \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - m)(x_j - m)^T \]

\[ T = C \otimes m \]

Objective function

\[ d(S_i, S_j) = \sum_{p=1}^{P} \eta_p (\phi_i^p - \phi_j^p)^T M (\phi_i^p - \phi_j^p) \eta_p (\phi_j^p) \]

\[ \max_M J = \sum_{i,j=1}^{N} \frac{d(S_i, S_j)}{N_{C^-}} - \sum_{i,j=1}^{N} \frac{d(S_i, S_j)}{N_{C^+}} \]

Complementary information (mean vs. covariance)

Set model IV: statistics (COV+)

- **DARG** (Discriminant Analysis on Riemannian manifold of Gaussian distributions) [CVPR’15]
  - Set modeling by mixture of Gaussian distribution (**GMM**)
    - Naturally encode the 1st order and 2nd order statistics
  - Metric learning: on Riemannian manifold

[Shakhnarovich, ECCV’02]
[Arandjelović, CVPR’05]

\[ \mathcal{M} \]: Riemannian manifold of Gaussian distributions

DARG

Framework

\( \mathcal{M} \): Riemannian manifold of Gaussian distributions
\( \mathcal{H} \): high-dimensional reproducing kernel Hilbert space (RKHS)
\( \mathbb{R}^d \): target lower-dimensional discriminant Euclidean subspace
Kernels on the Gaussian distribution manifold

- Kernel based on Lie Group
- Distance based on Lie Group (LGD)

\[ \text{LGD}(P_i, P_j) = \| \log(P_i) - \log(P_j) \|_F, \]

\[ g \sim N(x|\mu, \Sigma) \leftrightarrow P = |\Sigma|^{-\frac{1}{d+1}} \begin{pmatrix} \Sigma + \mu \mu^T & \mu \\ \mu^T & 1 \end{pmatrix} \]

- Kernel function

\[ K_{\text{LGD}}(g_i, g_j) = \exp \left( - \frac{\text{LGD}^2(P_i, P_j)}{2t^2} \right) \]
DARG

- Kernels on the Gaussian distribution manifold
  - kernel based on Lie Group
  - kernel based on MD and LED
  - Mahalanobis Distance (MD) between mean
    \[ MD(\mu_i, \mu_j) = \sqrt{(\mu_i - \mu_j)^T (\Sigma_i^{-1} + \Sigma_j^{-1})(\mu_i - \mu_j)} \]
  - LED between covariance matrix
    \[ LED(\Sigma_i, \Sigma_j) = \| \log(\Sigma_i) - \log(\Sigma_j) \|_F \]
  - Kernel function
    \[ K_{MD+LED}(g_i, g_j) = \gamma_1 K_{MD}(\mu_i, \mu_j) + \gamma_2 K_{LED}(\Sigma_i, \Sigma_j) \]
    \[ K_{MD}(\mu_i, \mu_j) = \exp \left( - \frac{MD^2(\mu_i, \mu_j)}{2t^2} \right) \]
    \[ K_{LED}(\Sigma_i, \Sigma_j) = \exp \left( - \frac{LED^2(\Sigma_i, \Sigma_j)}{2t^2} \right) \]
    \[ \gamma_1, \gamma_2 \] are the combination coefficients
DARG

- Discriminative learning
  - Weighted KDA (kernel discriminant analysis)
    - incorporating the weights of Gaussian components

\[
J(\alpha) = \frac{|\alpha^T B \alpha|}{|\alpha^T W \alpha|}
\]

\[
W = \sum_{i=1}^{C} \frac{1}{\omega_i} \sum_{j=1}^{N_i} w_j^{(i)} (k_j^i - m_i)(k_j^i - m_i)^T
\]

\[
B = \sum_{i=1}^{C} N_i (m_i - m)(m_i - m)^T
\]

\[
m_i = \frac{1}{N_i \omega_i} \sum_{j=1}^{N_i} w_j^i k_j^i, m = \frac{1}{N_i} \sum_{i=1}^{C} \frac{1}{\omega_i} \sum_{j=1}^{N_i} w_j^i k_j^i
\]
Set model IV: statistics (COV+)

- Beyond Gauss [ICCV’15]
  - Set modeling by probability distribution functions (PDFs)
    - More general than Gaussian assumption
    - Non-parametric, data-driven kernel density estimator (KDE)
  - Metric learning: on Riemannian manifold

PDFs form a Riemannian manifold, i.e., the statistical manifold.

Csiszár f-divergences are exploited to measure the geodesic distance.

Beyond Gauss

- Set modeling with PDFs
  - **Kernel Density Estimation (KDE)**
    \[
    \hat{p}(x) = \frac{1}{n \sqrt{\det(2\pi\Sigma)}} \sum_{i=1}^{n} \exp \left( -\frac{1}{2} (x - x_i)^T \Sigma^{-1} (x - x_i) \right)
    \]
  - Given two image sets \( \{x_i^p\}_{i=1}^{n_p} \) and \( \{x_i^q\}_{i=1}^{n_p} \) with estimated PDFs \( p(x) \) and \( q(x) \), how to compare two PDFs \( p(x) \) and \( q(x) \)?

- Empirical estimation of \(f\)-Divergences
  - **Hellinger distance**
    \[
    \delta_H^2(p||q) = \frac{1}{n_p} \sum_{i}^{n_p} \left( \sqrt{T(x_i^p)} - \sqrt{1 - T(x_i^p)} \right)^2 + \frac{1}{n_q} \sum_{i}^{n_q} \left( \sqrt{T(x_i^q)} - \sqrt{1 - T(x_i^q)} \right)^2
    \]
  - **Jeffrey divergence**
    \[
    \delta_H^2(p||q) = \frac{1}{n_p} \sum_{i}^{n_p} \left( 2T(x_i^p) - 1 \right) \ln \frac{T(x_i^p)}{1 - T(x_i^p)} + \frac{1}{n_q} \sum_{i}^{n_q} \left( 2T(x_i^q) - 1 \right) \ln \frac{T(x_i^q)}{1 - T(x_i^q)}
    \]
    \[
    T(x) = \frac{p(x)}{p(x) + q(x)}
    \]
Beyond Gauss

- Kernels on the Statistical Manifold
  - Hellinger Kernel
    \[ K_H(p, q) = \exp\left(-\sigma \delta_H^2(p, q)\right) \]
  - Laplace Kernel
    \[ K_L(p, q) = \exp(-\sigma \delta_H(p, q)) \]
  - Jeffrey Kernel
    \[ K_J(p, q) = \exp(-\sigma \delta_J(p||q)) \]

- Dimensionality Reduction
  \[ W^* = \arg\min_W L(W), \text{ s.t. } W^T W = I_d \]
  \[ L(W) = \sum_{i,j} a(X_i, X_j) \cdot \delta(W^T X_i, W^T X_j) \]

- High affinity \( a(X_i, X_j) \leftrightarrow \text{small distance after mapping} \)
- Low/negative affinity \( a(X_i, X_j) \leftrightarrow \text{large distance after mapping} \)

- Optimization by conjugate gradient on a Grassmann manifold.
Set model IV: statistics (COV+)

- **Properties**
  - The natural raw statistics of a sample set
  - Flexible model of multiple-order statistical information

- **Methods**
  - CDL [CVPR’12]
  - LMKML [ICCV’13]
  - DARG [CVPR’15]
  - B. Gauss [ICCV’15]
  - SPD-ML [ECCV’14]
  - LEML [ICML’15]
  - DCRL [CVPR’17]
  - SPDNet [AAAI’17]
  - DHH [TIP’19]

[Shakhnarovich, ECCV’02]
[Arandjelović, CVPR’05]
Set model IV: statistics (COV+)

- SPD-ML (SPD Manifold Learning) [ECCV’14]
  - Pioneering work on explicit manifold-to-manifold dimensionality reduction
  - Metric learning: on Riemannian manifold

SPD-ML

- SPD manifold dimensionality reduction
  - Mapping function: \( f : S_{++}^n \times \mathbb{R}^{n \times m} \rightarrow S_{++}^m \)
    - \( f(X, \tilde{W}) = \tilde{W}^T X \tilde{W} \in S_{++}^m > 0, \) \( X \in S_{++}^n, \) \( \tilde{W} \in \mathbb{R}^{n \times m} \) (full rank)

- Affine invariant metrics: AIRM / Stein divergence on target SPD manifold \( S_{++}^m \)
  - \( \delta^2 (\tilde{W}^T X_i \tilde{W}, \tilde{W}^T X_i \tilde{W}) = \delta^2 (W^T X_i W, W^T X_j W) \)
    - \( \tilde{W} = MW, \) \( M \in GL(n), \) \( W \in \mathbb{R}^{n \times m}, \) \( W^T W = I_m \)
Discriminative learning

- Discriminant function
  - Graph Embedding formalism with an affinity matrix that encodes intra-class and inter-class SPD distances
  - \[ \min L(W) = \min \sum_{ij} A_{ij} \delta^2 (W^T X_i W, W^T X_j W) \]
  - s. t. \[ W^T W = I_m \] (orthogonality constraint)

- Optimization
  - Optimization problems on Stiefel manifold, solved by nonlinear Conjugate Gradient (CG) method
Set model IV: statistics (COV+)

- LEML (Log-Euclidean Metric Learning) [ICML’15]
  - Learning tangent map by preserving matrix symmetric structure
  - Metric learning: on Riemannian manifold

CDL [CVPR’12]

LEML

- SPD tangent map learning
  - Mapping function: $DF: f(\log(S)) = W^T \log(S)W$
    - $W$ is column full rank
  - Log-Euclidean distance in the target tangent space
    - $d_{LED}(f(T_i), f(T_j)) = ||W^T T_i W - W^T T_j W||_F$
      
      $$= tr(Q(T_i - T_j)(T_i - T_j))$$

      $Q = (WW^T)^2$

      $T = \log(S)$

      analogy to 2DPCA
LEML

- Discriminative learning
  - Objective function
    \[ \arg \min_{Q, \xi} D_{ld}(Q, Q_0) + \eta D_{ld}(\xi, \xi_0) \]
    \[ \text{s. t. } \text{tr}(QA_{ij}^T A_{ij}) \leq \xi_{c(i,j)}, (i, j) \in S \]
    \[ \text{tr}(QA_{ij}^T A_{ij}) \geq \xi_{c(i,j)}, (i, j) \in D \]
    \[ A_{ij} = \log(C_i) - \log(C_j), \text{ } D_{ld}: \text{LogDet divergence} \]
  - Optimization
    - Cyclic Bregman projection algorithm [Bregman’1967]
Set model IV: statistics (COV+)

- Properties
  - The natural raw statistics of a sample set
  - Flexible model of multiple-order statistical information

- Methods
  - CDL [CVPR’12]
  - LMKML [ICCV’13]
  - DARG [CVPR’15]
  - B. Gauss [ICCV’15]
  - SPD-ML [ECCV’14]
  - LEML [ICML’15]
  - DCRL [CVPR’17]
  - SPDNet [AAAI’17]
  - DHH [TIP’19]

- [Shakhnarovich, ECCV’02]
- [Arandjelović, CVPR’05]
Set model IV: statistics (COV+)

- **DCRL (Discriminative Covariance Oriented Representation Learning) [CVPR’17]**
  - **Image feature learning** that facilitates image set modeling and classification
    - Image feature learning: Deep learning networks, e.g., CNN
    - Image set modeling: Set covariance matrices
  - **Metric learning:** on set covariance matrices with image feature space learned jointly

Formulation

- Given $n$ training image sets $\{X_i\}_{i=1}^{n}$, where $X_i$ contains original feature vectors of $N_i$ images
- Image feature learning
  - $X_i \mapsto h_i = \phi_\Theta(X_i)$
- Image set modeling
  - $C_i = \hat{h}_i^T \hat{h}_i$, where $\hat{h}_i$ is the centered $h_i$
- Network optimization
  - Formulate the discrimination of set covariance matrices by some loss function
  - Optimize the feature learning network to minimize such loss function
Graph Embedding Scheme

Loss function

\[ J(\Theta) = \frac{1}{4} \sum_{i,j} A_{ij} LEM^2(C_i, C_j) \]

where

\[ LEM(C_i, C_j) = \| \log_I(C_i) - \log_I(C_j) \|_F \]

is the Log-Euclidean Metric (LEM)*

---

Graph Embedding Scheme

Loss function

- **Adjacency Graph**: Encode the data structure and semantic relationship of set covariance matrices

\[
A_{ij} = \begin{cases} 
  d_{ij} & \text{if } X_i \in N_w(X_j) \text{ or } X_j \in N_w(X_i) \\
  -d_{ij}, & \text{if } X_i \in N_b(X_j) \text{ or } X_j \in N_b(X_i) \\
  0, & \text{otherwise}
\end{cases}
\]

\[
d_{ij} = \exp(-LEM^2(C_i, C_j)/\sigma^2)
\]
Softmax Regression Scheme

- Loss function
  - Softmax regression

\[
J(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} 1\{y_i = j\} \log(o_{ij})
\]

1\{true\} = 1, 1\{false\} = 0;

\[o_{ij} = P(y_i = j|v_i; W, b)\]

- log-covariance vector

\[v_i = vec(\log(C_i))\]
Softmax Regression Scheme

Loss function

Train a Softmax classifier to discriminate the set covariance matrices on a flat tangent space.
SPDNet (SPD matrix network) [AAAI’17]

- Metric learning: SPD matrices nonlinear learning in deep networks

\[ X_0 \xrightarrow{f_b^{(1)}} X_1 = W_1 X_0 W_1^T \xrightarrow{f_r^{(2)}} X_2 = U_1 \max(\epsilon I, \Sigma_1) U_1^T \]

\[ X_1 = U_1 \Sigma_1 U_1^T \]

\[ X_{l-1} = U_{l-2} \log(\Sigma_{l-2}) U_{l-2}^T \]

\[ X_{l-2} = U_{l-2} \Sigma_{l-2} U_{l-2}^T \]

Input SPD matrix

BiMap Layer

ReEig Layer

LogEig Layer

Output Layers

SPDNet

Basic idea: respect Riemannian geometry of SPD manifold in nonlinear transformation of deep networks

- Bilinear mapping (BiMAP) Layer
  - Fully connected convolution-like

- Eigenvalue rectification (ReEig) Layer
  - Rectified linear units (ReLU)-like

- Eigenvalue logarithm (LogEig) Layer
  - Manifold to Euclidean space

Formulation analogous to LieNet
DHH (Deep Heterogeneous Hashing) [TIP’19]
- Application scenario: image-video face retrieval
- Metric learning: hamming distance learning across deep heter. spaces

Face Video Database

Search Results

Image Query

DHH

- Heterogeneous Hash Learning

- Cov.

- Manifold

- Tangent Space

- Euclidean Space

- Hashing

- Hamming Space

- Vectors

- Euclidean Space

- Hashing
DHH

- Framework
  - Jointly learn deep hashing in homogeneous and across heterogeneous spaces
DHH

- Framework
  - Jointly learn deep hashing in **homogeneous** and across **heterogeneous** spaces

**Triplet Loss Function**

\[ J_{i,j,k} = \max\{0, \alpha + d_h(b_i, b_j) - d_h(b_i, b_k)\} \]

s.t. \( b_i, b_j, b_k \in \{0,1\}^K \),

\( (i,j) \in \text{positive}, (i,k) \in \text{negative} \)

**Objective Function**

\[ J = J_{er} + \lambda_1 J_e + \lambda_2 J_r \]

- Across heter. spaces
- Within homo. spaces
Riemannian matrix backpropagation
Riemannian matrix backpropagation

Structured gradients between two adjacent layers

\[
\frac{\partial L^{(k)}}{\partial X_{k-1}} : dX_{k-1} = \frac{\partial L^{(k+1)}}{\partial X_k} : dX_k
\]

Riemannian matrix gradients of video second-order modeling

Input: \(X_{k-1}, SVD (X_{k-1}) = U \Sigma V^T\), \hspace{1cm} Output: \(X_k = V \log m(\Sigma^T \Sigma + \epsilon I)V^T\)

\[D = \Sigma^{-1}_m \left( \frac{\partial L^{(k')}}{\partial V} \right)_1 - \Sigma^{-1}_m V_1^T \left( \frac{\partial L^{(k')}}{\partial V} \right)_2 V_2^T, K_{i,j} = \begin{cases} 
0 & i = j \\
1 & i \neq j \\
\sigma_j^{-2} - \sigma_i^{-2} & i \neq j
\end{cases} \]

\[
\frac{\partial L^{(k)}}{\partial X_{k-1}} = UD + U \left( \frac{\partial L^{(k')}}{\partial \Sigma} - DV \right)_{\text{diag}} V^T + 2U(K \circ (-DV \Sigma^T))_{\text{sym}} \Sigma V^T
\]

\[
\frac{\partial L^{(k')}}{\partial V} = 2\left( \frac{\partial L^{(k+1)}}{\partial X_k} \right)_{\text{sym}} V \log m(\Sigma^T \Sigma + \epsilon I),
\]

\[
\frac{\partial L^{(k')}}{\partial \Sigma} = 2\Sigma (\Sigma^T \Sigma + \epsilon I)^{-1} V^T \left( \frac{\partial L^{(k+1)}}{\partial X_k} \right)_{\text{sym}} V
\]
Outline

- Background
- Literature review
- Evaluations
- Summary
Evaluations: video face recognition

- Two YouTube datasets
  - YouTube Celebrities (YTC) [Kim, CVPR’08]
    - 47 subjects, 1910 videos from YouTube
  - YouTube FaceDB (YTF) [Wolf, CVPR’11]
    - 3425 videos, 1595 different people
Evaluations: video face recognition

- **COX Face** [Huang, ACCV’12/TIP’15]
  - 1,000 subjects
  - each has 1 high quality images, 3 unconstrained video sequences

Images

Videos

http://vipl.ict.ac.cn/resources/datasets/cox-face-dataset/cox-face
Evaluations: video face recognition

- **PaSC** [Beveridge, BTAS’13]
  - **Control videos**
    - 1 mounted video camera
    - 1920*1080 resolution
  - **Handheld videos**
    - 5 handheld video cameras
    - 640*480~1280*720 resolution

Table 2. Summary of Video PaSC Data.

<table>
<thead>
<tr>
<th>Number of Subjects</th>
<th>265</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Videos</td>
<td>2,802</td>
</tr>
<tr>
<td>Total Control Videos</td>
<td>1,401</td>
</tr>
<tr>
<td>Total Handheld Videos</td>
<td>1,401</td>
</tr>
<tr>
<td>Control Videos per Subject</td>
<td>4 to 7</td>
</tr>
</tbody>
</table>
Evaluations

Results (**reported in our DARG paper***)

<table>
<thead>
<tr>
<th>Method</th>
<th>YTC</th>
<th>COX-11</th>
<th>COX-12</th>
<th>COX-23</th>
<th>COX-21</th>
<th>COX-31</th>
<th>COX-32</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHISD [CVPR’10]</td>
<td>66.46</td>
<td>56.87</td>
<td>30.10</td>
<td>14.80</td>
<td>44.37</td>
<td>26.44</td>
<td>13.68</td>
</tr>
<tr>
<td>GDA [CVPR’08]</td>
<td>65.91</td>
<td>72.26</td>
<td>80.70</td>
<td>74.36</td>
<td>71.44</td>
<td>81.99</td>
<td>77.57</td>
</tr>
<tr>
<td>GGDA [CVPR’11]</td>
<td>66.83</td>
<td>76.73</td>
<td>83.80</td>
<td>76.59</td>
<td>72.56</td>
<td>82.84</td>
<td>79.99</td>
</tr>
<tr>
<td>MMD [CVPR’08]</td>
<td>65.30</td>
<td>38.29</td>
<td>30.34</td>
<td>15.24</td>
<td>34.86</td>
<td>22.21</td>
<td>11.44</td>
</tr>
<tr>
<td>MDA [CVPR’09]</td>
<td>66.98</td>
<td>65.82</td>
<td>63.01</td>
<td>36.17</td>
<td>55.46</td>
<td>43.23</td>
<td>29.70</td>
</tr>
<tr>
<td>SGM [ECCV’02]</td>
<td>52.00</td>
<td>26.74</td>
<td>14.32</td>
<td>12.39</td>
<td>26.03</td>
<td>19.21</td>
<td>10.50</td>
</tr>
<tr>
<td>CDL [CVPR’12]</td>
<td>69.70</td>
<td>78.37</td>
<td>85.25</td>
<td>79.74</td>
<td>75.59</td>
<td>85.83</td>
<td>81.87</td>
</tr>
<tr>
<td>DARG-KLD</td>
<td>72.21</td>
<td>71.93</td>
<td>80.11</td>
<td>73.65</td>
<td>70.87</td>
<td>81.03</td>
<td>76.99</td>
</tr>
<tr>
<td>DARG-LGD</td>
<td>68.72</td>
<td>76.74</td>
<td>84.99</td>
<td>78.02</td>
<td>72.93</td>
<td>83.88</td>
<td>81.54</td>
</tr>
<tr>
<td>DARG-MD+LED</td>
<td>77.09</td>
<td>83.71</td>
<td>90.13</td>
<td>85.08</td>
<td>81.96</td>
<td>89.99</td>
<td>88.35</td>
</tr>
</tbody>
</table>

Evaluations

- Results (*reported in our DARG paper*)

VR@FAR = 0.01 on PaSC

AUC on YTF
Evaluations

- **Results** *(reported in our DCRL paper*)
  - Verification task (on YTF dataset)


GE: Graph embedding scheme
SR: Softmax regression scheme
FN: Baseline Deep ID net with single CNN
Evaluations

- Results (*reported in our DCRL paper*)
- Verification task (on PaSC dataset)
Evaluations

- Results *(reported in our DCRL paper*)
  - Identification task (on YTC dataset)
Evaluations: video face retrieval

- **Datasets**
  - YouTube Celebrities (YTC) [Kim, CVPR’08]
  - Prison Break (PB) [Li, FG.’15]
  - UMDFaces-200 [Bansal, IJCB’17]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>YTC</th>
<th>PB</th>
<th>UMDFaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Video #</td>
<td>7,190</td>
<td>2,415</td>
<td>6,614</td>
</tr>
<tr>
<td>Test Video #</td>
<td>3,101</td>
<td>10,495</td>
<td>3,422</td>
</tr>
<tr>
<td>Training Image #</td>
<td>21,570</td>
<td>7,245</td>
<td>19,842</td>
</tr>
<tr>
<td>Test Image #</td>
<td>3,101</td>
<td>10,495</td>
<td>3,422</td>
</tr>
<tr>
<td>Video # per subject</td>
<td>219.0±114.6</td>
<td>679.5±710.6</td>
<td>50.2±19.5</td>
</tr>
</tbody>
</table>
## Evaluations: video face retrieval

### Retrieval mAP results

<table>
<thead>
<tr>
<th>Method</th>
<th>YouTube Celebrities</th>
<th>the Prison Break</th>
<th>UMDFaces</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12-bit</td>
<td>24-bit</td>
<td>48-bit</td>
</tr>
<tr>
<td>LSH [4]</td>
<td>0.1105</td>
<td>0.1504</td>
<td>0.2042</td>
</tr>
<tr>
<td>SH [5]</td>
<td>0.2262</td>
<td>0.2726</td>
<td>0.2814</td>
</tr>
<tr>
<td>SSH [33]</td>
<td>0.2811</td>
<td>0.3324</td>
<td>0.3068</td>
</tr>
<tr>
<td>ITQ [6]</td>
<td>0.3461</td>
<td>0.4424</td>
<td>0.4596</td>
</tr>
<tr>
<td>DBC [35]</td>
<td>0.4244</td>
<td>0.5017</td>
<td>0.5478</td>
</tr>
<tr>
<td>KSH [7]</td>
<td>0.3973</td>
<td>0.4917</td>
<td>0.5709</td>
</tr>
<tr>
<td>DNNH [8]</td>
<td>0.4868</td>
<td>0.5467</td>
<td>0.5701</td>
</tr>
<tr>
<td>DSH [46]</td>
<td>0.4657</td>
<td>0.5305</td>
<td>0.5432</td>
</tr>
<tr>
<td>HashNet [50]</td>
<td>0.3965</td>
<td>0.5302</td>
<td>0.5865</td>
</tr>
<tr>
<td>CMSSH [52]</td>
<td>0.1082</td>
<td>0.1703</td>
<td>0.2005</td>
</tr>
<tr>
<td>CVH [53]</td>
<td>0.2081</td>
<td>0.2371</td>
<td>0.2693</td>
</tr>
<tr>
<td>PLMH [55]</td>
<td>0.1755</td>
<td>0.1959</td>
<td>0.2065</td>
</tr>
<tr>
<td>PDH [10]</td>
<td>0.2719</td>
<td>0.3809</td>
<td>0.4190</td>
</tr>
<tr>
<td>MLBE [54]</td>
<td>0.4641</td>
<td>0.4438</td>
<td>0.5287</td>
</tr>
<tr>
<td>MM-NN [56]</td>
<td>0.2791</td>
<td>0.5218</td>
<td>0.5595</td>
</tr>
<tr>
<td>HER [58]</td>
<td>0.3600</td>
<td>0.5045</td>
<td>0.5756</td>
</tr>
<tr>
<td>DHH</td>
<td><strong>0.5406</strong></td>
<td><strong>0.5802</strong></td>
<td><strong>0.6120</strong></td>
</tr>
</tbody>
</table>
Evaluations: video face retrieval

- Retrieval using still images from internet

Web Image Query vs. Video Database

- DNNH
- DSH
- HashNet
- MM-NN
- HER
- DHH

(a) YTC
(b) PB
(c) UMDFaces
(d) UMDFaces*
Outline

- Background
- Literature review
- Evaluations
- Summary
Summary

What we learn from current studies
- Set modeling
  - Linear(/affine) subspace $\rightarrow$ Manifold $\rightarrow$ Statistics
- Set matching
  - Non-discriminative $\rightarrow$ Discriminative
- Metric learning
  - Euclidean space $\rightarrow$ Riemannian manifold

Future directions
- More flexible set modeling for different scenarios
- Multi-model combination
- Learning method should be more efficient
- Set-based vs. sample-based?
Additional references (not listed above)

Thanks, Q & A

Lab of Visual Information Processing and Learning (VIPL) @ICT@CAS

Codes of our methods available at: http://vipl.ict.ac.cn/resources/codes