Learning Euclidean-to-Riemannian Metric for Point-to-Set Classification

Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xilin Chen
Institute of Computing Technology, Chinese Academy of Sciences

June 25, 2014
Problem (1/2)

- Video surveillance

Seeking missing children

Search criminal suspects

http://www.youtube.com/watch?v=M80DXI932OE
http://www.youtube.com/watch?v=RfJsGeq0xRA#t=22
Problem (2/2)

- Video surveillance
  - Still-to-Video face recognition

- Watch list
- Surveillance video
- Point-to-Set Classification

Watch list screening
Motivation (1/3)

- **Point-to-Set Classification**
  - Match points against sets
    - Compare point with all points in set
      - 1-N Point to Point (P2P) matching
  - Compare point with set model
    - 1-1 Point to Set (P2S) matching

- **Hausdorff dist.**

- **Time / space consuming**

- **More robust**

- **Point**

- **Set model**

- **Linear subspace**
  - [Yamaguchi, FG'98]
  - [Chien, PAMI'02]
  - [Kim, PAMI'07]

- **Affine hull**
  - [Vincent, NIPS'01]
  - [Cevikalp, CVPR'10]
  - [Zhu, ICCV'13]

- **Covariance matrix**
  - [Wang, CVPR'12]
  - [Vemulapalli, CVPR'13]
  - [Lu, ICCV'13]
Motivation (2/3)

- **Point-to-Set Classification**
  - Euclidean points vs. Riemannian points

Euclidean space (E) vs. Corresponding manifold: 1. Grassmann (G) 2. Affine Grassmann (A) 3. SPD (S)

- Heterogeneous linear subspace
- Affine hull
- Covariance matrix

- Point
- Set model

Reference:
- [Hamm, ICML’08]
- [Harandi, CVPR’11]
- [Hamm, NIPS’08]
- [Pennec, IJCV’06]
- [Arsigny, SIAM’07]
Learn Euclidean-to-Riemannian metric

- Corresponding manifold:
  1. Grassmann (G)
  2. Affine Grassmann (A)
  3. SPD (S)

Heterogeneous
Our Method(1/11)

- Basic idea
  - Reduce Euclidean-to-Riemannian metric to classical Euclidean metric
  - Seek maps $F, \Phi$ to a common Euclidean subspace

$$d(x_i, y_j) = \sqrt{(F(x_i) - \Phi(y_j))^T(F(x_i) - \Phi(y_j))}$$
Our Method(2/11)

- Basic idea
  - Bridge Euclidean-to-Riemannian gap
    - Hilbert space embedding
      - Adhere to Euclidean geometry
      - Globally encode the geometry of manifold
Our Method (3/11)

- **Three Mapping Modes**

  1. **Mapping Mode 1**
     - Single Hilbert space
     - Further reduce the gap
     - Further explore correlations

  2. **Mapping Mode 2**
     - Double Hilbert spaces

  3. **Mapping Mode 3**
     - Cross-space kernel

**Different:**
- Final maps \( F, \Phi \)

**Same:**
- Objective fun. \( E(F, \varphi) \)
Our Method (6/11)

- Mapping Mode 1

**Single Hilbert space**

**Final maps:**
\[
F = f_x = W^T_x X \\
\Phi = f_y \circ \phi_y = W^T_y K_y \\
K_y(i, j) = \exp(-d^2(y_i, y_j)/2\sigma^2)
\]

**Distance metric:**
\[
d(x_i, y_j) = \sqrt{(F(x_i) - \Phi(y_j))^T(F(x_i) - \Phi(y_j))}
\]

**Objective function:**
\[
\min_{F, \Phi} \{ D(F, \Phi) + \lambda_1 G(F, \Phi) + \lambda_2 T(F, \Phi) \}
\]

- Distance
- Geometry
- Transformation

Riemannian metrics [ICML’08, NIPS’08, SIAM’06]
Our Method (7/11)

Three Mapping Modes

- **Mapping Mode 1**: Single Hilbert space
  - Same: Objective fun. $E(F, \varphi)$
  - Different: Final maps $F, \Phi$

- **Mapping Mode 2**: Double Hilbert spaces
  - Further explore correlations

- **Mapping Mode 3**: Double Hilbert spaces + Cross-space kernel
  - Further reduce the gap
Our Method (8/11)

- Mapping Mode 2

Further reduce the E-R gap with double Hilbert spaces

Final maps:

- $F = f_x \circ \varphi_x = W_x^T K_x$
- $\Phi = f_y \circ \varphi_y = W_y^T K_y$

- $K_x(i, j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$

Classical Euclidean distance

- $K_y(i, j) = \exp(-d^2(y_i, y_j) / 2\sigma^2)$

Distance metric:

- $d(x_i, y_j) = \sqrt{(F(x_i) - \Phi(y_j))^T (F(x_i) - \Phi(y_j))}$

Objective function: $E(F, \varphi)$

- $\min_{F, \Phi} \{ D(F, \Phi) + \lambda_1 G(F, \Phi) + \lambda_2 T(F, \Phi) \}$

- Distance
- Geometry
- Transformation
Our Method(9/11)

- Three Mapping Modes

- Mapping Mode 1
  Single Hilbert space

- Mapping Mode 2
  Double Hilbert spaces

- Mapping Mode 3
  +Cross-space kernel

Different:
Final maps $F, \Phi$

Same:
Objective fun. $E(F, \phi)$

Further reduce the gap

Further explore correlations
Our Method (10/11)

- **Mapping Mode 3**

**Final Maps:**

\[
F = f_x \circ \varphi'_x = W_x^T [K_x, K_{xy}]
\]

\[
\Phi = f_y \circ \varphi'_y = W_y^T [K_y, K_{xy}^T]
\]

\[
K_x(i, j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)
\]

\[
K_y(i, j) = \exp\left(-d^2(y_i, y_j)/2\sigma^2\right)
\]

\[
K_{xy}(i, j) = \exp\left(-d^2(x_i, y_j)/2\sigma^2\right)
\]

**Distance metric:**

\[
d(x_i, y_j) = \sqrt{(F(x_i) - \Phi(y_j))^T (F(x_i) - \Phi(y_j))}
\]

**Objective function:** \(E(F, \Phi)\)

\[
\min_{F, \Phi} \left\{ \sum D(F, \Phi) + \lambda_1 G(F, \Phi) + \lambda_2 T(F, \Phi) \right\}
\]

- Distance
- Geometry
- Transformation

Further correlate the double Hilbert spaces with cross-space kernel

Point-to-Set (E-R) distance

[PAMI’02, NIPS’01, Mahalanobis]
Our Method (11/11)

- Optimization of the objective functions
  - Iterative optimization
    - CCA-like Initialization
      - \[ \max_{W_x, W_y} \{ D^b(W_x, W_y) + \lambda_1 G^b(W_x, W_y) \} \]
      - s. t. \[ D^w(W_x, W_y) + \lambda_1 G^w(W_x, W_y) = 1 \]
  - Alternately updating (gradient descent)
    - Fix \( W_y \) to update \( W_x \)
      - \( W_x =? \)
    - Fix \( W_x \) to update \( W_y \)
      - \( W_y =? \)

\[
F = f_x \circ \varphi_x = W_x^T K_x \\
\Phi = f_y \circ \varphi_y = W_y^T K_y
\]
Experiments(1/7)

- **Video face datasets**
  - **YouTube Celebrities** [Kim, CVPR’08]
    - 47 subjects
    - 1,910 videos from YouTube, images selected from videos

Images

Videos
Experiments (2/7)

- Video face datasets
  - COX Face [Huang, ACCV’12]
    - 1,000 subjects
      - each has 1 high quality images, 3 unconstrained video sequences

Images

Videos
Experiments (3/7)

- Comparative Methods
  - Point-to-Point (P2P) metric learning methods
    - NCA [Goldberger, NIPS’04]
    - ITML [Davis, ICML’07]
    - LMNN [Weinberger, JMLR’09]
  - Point-to-Set (P2S) matching methods
    - NFS [Chien, PAMI’02]
    - HKNN [Vincent, NIPS’01]
    - PSDML [Zhu, ICCV’13]
  - Kernelized Multiview Learning (KML) methods
    - KPLS [Sharma, CVPR’11]
    - KCCA [Hardoon, Neural Comp.’04]
    - KGMLDA [Sharma, CVPR’12]
Experiments(4/7)

- Still-to-Video face recognition
  - YouTube dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>Gallery-Probe; Average rank-1 recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Still-Video</td>
</tr>
<tr>
<td><strong>P2P</strong></td>
<td></td>
</tr>
<tr>
<td>NCA [NIPS'04]</td>
<td>51.74±3.11</td>
</tr>
<tr>
<td>ITML [ICML'07]</td>
<td>47.62±1.73</td>
</tr>
<tr>
<td>LMNN [JMLR'09]</td>
<td>55.02±2.71</td>
</tr>
<tr>
<td><strong>P2S</strong></td>
<td></td>
</tr>
<tr>
<td>NFS [PAMI'02]</td>
<td>53.27±2.75</td>
</tr>
<tr>
<td>HKNN [NIPS'01]</td>
<td>36.94±2.71</td>
</tr>
<tr>
<td>PSDML [ICCV'13]</td>
<td>55.30±1.90</td>
</tr>
<tr>
<td><strong>KML</strong></td>
<td></td>
</tr>
<tr>
<td>KPLS [CVPR'11]</td>
<td>54.02±2.61</td>
</tr>
<tr>
<td>KCCA [Neural Comp.'04]</td>
<td>55.80±3.12</td>
</tr>
<tr>
<td>KGMLDA [CVPR'12]</td>
<td>58.19±4.00</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td></td>
</tr>
<tr>
<td>LERM</td>
<td>69.11±2.99</td>
</tr>
</tbody>
</table>
Experiments (5/7)

- Still-to-Video face recognition
  - COX Face dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>Gallery-Probe; Average rank-1 recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S-V1</td>
</tr>
<tr>
<td>P2P</td>
<td></td>
</tr>
<tr>
<td>NCA [NIPS’04]</td>
<td>39.14</td>
</tr>
<tr>
<td>ITML [ICML’07]</td>
<td>19.83</td>
</tr>
<tr>
<td>LMNN [JMLR’09]</td>
<td>34.44</td>
</tr>
<tr>
<td>P2S</td>
<td></td>
</tr>
<tr>
<td>NFS [PAMI’02]</td>
<td>9.99</td>
</tr>
<tr>
<td>HKNN [NIPS’01]</td>
<td>4.70</td>
</tr>
<tr>
<td>KML</td>
<td></td>
</tr>
<tr>
<td>KCCA [Neural Comp.’04]</td>
<td>38.60</td>
</tr>
<tr>
<td>KGMLDA [CVPR’12]</td>
<td>41.89</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
</tr>
<tr>
<td>LERM</td>
<td>45.71</td>
</tr>
</tbody>
</table>
Experiments (6/7)

Different mapping modes

Set models:
- EG: Euclidean-to-Grassmannian (linear subspace)
- EA: Euclidean-to-AffineGrassmannian (affine subspace)
- ES: Euclidean-to-SPD (covariance matrix)
Experiments (7/7)

Running time (seconds) on YouTube

<table>
<thead>
<tr>
<th>Methods</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P2P</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCA [NIPS’04]</td>
<td>7761</td>
<td>0.165</td>
</tr>
<tr>
<td>ITML [ICML’07]</td>
<td>523.2</td>
<td>0.394</td>
</tr>
<tr>
<td>LMNN [JMLR’09]</td>
<td>282.4</td>
<td>0.162</td>
</tr>
<tr>
<td><strong>P2S</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSDML [ICCV’13]</td>
<td>55.78</td>
<td>0.016</td>
</tr>
<tr>
<td>LERM</td>
<td>3.615</td>
<td>0.032</td>
</tr>
</tbody>
</table>

Note: The running time in test is average per testing video
Conclusions

- Point-to-set classification
  - Euclidean-to-Riemannian matching

- Conventional metric learning
  - Euclidean-to-Riemannian Metric Learning

- Impressively improvement and much higher efficiency
Future work

- This framework may be also applied in many other CV tasks, e.g., image-to-set object categorization, image-to-video face retrieval.
- This framework can be extended for metrics between sets or even different kinds of sets.
Take home message

- Learning Euclidean-to-Riemannian Metric (LERM) for point-to-set classification

Get a copy of COX face dataset for Still and video face recognition!

Data and code are available online!

http://vipl.ict.ac.cn/resources

Poster ID: O-2E-1
References

References

Appendix
Background (1/3)

- **Riemannian metric**
  - **Grassmannian metric**
    - Manifold $\mathcal{G}(D', D)$
      - The space of linear subspaces
        - Each is represented by an orthonormal matrix $U \in \mathbb{R}^{D \times D'}$
    - Projection Metric [Hamm et al., ICML’08]
      - $d^2(U_i, U_j) = 2^{-1/2} \| U_i U_i^T - U_j U_j^T \|_F$
  - **Affine Grassmannian metric**
    - Manifold $\mathcal{AG}(D', D)$
      - The space of affine subspaces
        - Each is represented by $U \in \mathbb{R}^{D \times D'}$ with mean value $u$
    - Affine Projection Metric [Hamm et al., NIPS’08]
      - $d^2(U_i, U_j) = 2^{-1/2} (\| U_i U_i^T - U_j U_j^T \|_F + \| (I - U_i U_i^T)u_i - (I - U_j U_j^T)u_j \|_F)$
Riemannian metric

- Symmetric Positive Define (SPD) metric
  - Manifold $\text{Sym}_D^+$
    - The space of SPD matrices $C \in \mathbb{R}^{D \times D}$
  - Log-Euclidean distance Metric [Arsigny et al., SIAM’07]
    - $d^2(C_i, C_j) = \| \log(C_i) - \log(C_j) \|_F$
Background (3/3)

- **Euclidean-to-Riemannian metric**
  - **Euclidean-to-Grassmannian metric**
    - NFS classifier [Chien et al., PAMI’02]
      - \(d(x_i, u_j) = \|x_i - U_j U_j^T x_i\|_F\)
  - **Euclidean-to-AffGrassmannian metric**
    - HKNN classifier [Vincent et al., NIPS’01]
      - \(d(x_i, A_j) = \min_{\alpha} \|U_j \alpha + u_j - x_i\|_F\), \(\alpha\) is a vector of free parameters that provides coordinates for points within the subspace

- **Euclidean-to-SPD metric**
  - Mahalanobis distance
  - \(d^2(x_i, C_j) = \sqrt{(x_i - u_j)^T C_j^{-1} (x_i - u_j)}\)
**Obj. Fun. of Proposed Method**

**Distance metric:** 
\[ d(x_i, y_j) = \sqrt{(F(x_i) - \Phi(y_j))^T (F(x_i) - \Phi(y_j))} \]

**Objective fun.:** 
\[ E(F, \Phi) = \min_{F,\Phi} \{ \begin{array}{ll} D(F, \Phi) + \lambda_1 G(F, \Phi) + \lambda_2 T(F, \Phi) \end{array} \} \]

\[ D = \frac{1}{2} \sum \sum \ sgn(l_i^x, l_j^y) d^2(x_i, y_j) \]

\[ G_x = \frac{1}{2} \sum \sum e \frac{d^2(x_i, x_j)}{\sigma_i^2} \]

\[ T = \frac{1}{2} \left( \|F(X)\|_F^2 + \|\Phi(Y)\|_F^2 \right) \]