A Survey on Riemannian Geometry of SPD Matrices and Its Applications

Yunpeng Shi

December 13, 2017

Content

Why we care about SPD matrices?

- Symmetric positive definite matrices

Why we care about SPD matrices?

- Symmetric positive definite matrices

Applications

- Covariance region descriptors for detection and recognition
- Texture classification
- Object tracking, face recognition
- Diffusion tensor images (DTI)
- Motion segmentation

The Issue of Euclidean Geometry

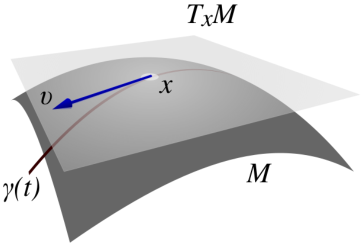
- The space of $d \times d$ SPD matrices Sym_d^+ is not a vector space
- The Euclidean distance does not correctly measure the similarities among SPD matrices

The Issue of Euclidean Geometry

- The space of $d \times d$ SPD matrices Sym_d^+ is not a vector space
- The Euclidean distance does not correctly measure the similarities among SPD matrices
- What is the true underlying geometry?

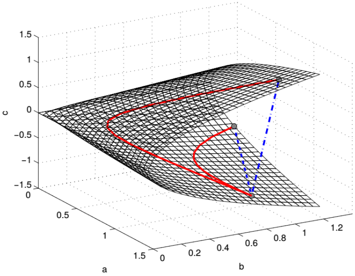
Riemannian Geometry

- Riemannian manifold is a differentiable manifold with a family smoothly varying inner products (Riemannian metric) on tangent spaces



Riemannian Geometry

- The space of $d \times d$ SPD matrices Sym_d^+ is a Riemannian manifold



Metrics for Measuring Similarities

Metric Name	Formula	Geodesic Distance	Positive Definite Gaussian Kernel $\forall \sigma > 0$
Log-Euclidean	$\|\log(\mathbf{S}_1) - \log(\mathbf{S}_2)\|_F$	**Yes**	**Yes**
Affine-Invariant	$\|\log(\mathbf{S}_1^{-1/2}\mathbf{S}_2\mathbf{S}_1^{-1/2})\|_F$	Yes	No
Cholesky	$\|\mathrm{chol}(\mathbf{S}_1) - \mathrm{chol}(\mathbf{S}_2)\|_F$	No	Yes
Power-Euclidean	$\frac{1}{\alpha}\|\mathbf{S}_1^\alpha - \mathbf{S}_2^\alpha\|_F$	No	Yes
Root Stein Divergence	$\left[\log\det\left(\frac{1}{2}\mathbf{S}_1 + \frac{1}{2}\mathbf{S}_2\right) - \frac{1}{2}\log\det(\mathbf{S}_1\mathbf{S}_2)\right]^{1/2}$	No	No

For $\boldsymbol{S}_1, \boldsymbol{S}_2 \in Sym_d^+$, define multiplication

$$\boldsymbol{S}_1 \odot \boldsymbol{S}_2 = \exp(\log \boldsymbol{S}_1 + \log \boldsymbol{S}_2),$$

then Sym_d^+ becomes a Lie group. Here both $\exp$ and $\log$ are matrix exponential and matrix logarithm.

Dictionary Learning in face recognition setting

- Dictionary: bases that spans the space of human face, where each face image can be expressed as linear combination of those bases
- Representation: the coefficient of the linear combination

$$\min \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{R}\|_F^2 \text{ subject to } \|\boldsymbol{r}_i\|_0 \leq T_0.$$

- $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]$, $\boldsymbol{D} = [\boldsymbol{d}_1, ..., \boldsymbol{d}_m]$
- Representation: the coefficient of the linear combination

Riemannian Dictionary Learning

- Stein Divergence: $\boldsymbol{S}(\boldsymbol{X}, \boldsymbol{Y}) = \log(\det \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}) - \frac{1}{2}\log(\det(\boldsymbol{X}\boldsymbol{Y}))$
  - induces kernel $k(\boldsymbol{X}, \boldsymbol{Y}) = e^{-\sigma \boldsymbol{S}(\boldsymbol{X}, \boldsymbol{Y})}$ for some $\sigma > 0$
  - (Harandi et al. ECCV 2012)
    $\min \sum_j \|\Phi(\boldsymbol{X}_j) - \sum_i r_{ji}\Phi(\boldsymbol{D}_i)\|_2^2 + \lambda \|\boldsymbol{r}_j\|_1$

Riemannian Dictionary Learning

- Stein Divergence: $\boldsymbol{S}(\boldsymbol{X},\boldsymbol{Y}) = \log(\det \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}) - \frac{1}{2}\log(\det(\boldsymbol{X}\boldsymbol{Y}))$
  - induces kernel $k(\boldsymbol{X},\boldsymbol{Y}) = e^{-\sigma \boldsymbol{S}(\boldsymbol{X},\boldsymbol{Y})}$ for some $\sigma > 0$
  - (Harandi et al. ECCV 2012)
    $\min \sum_j \|\Phi(\boldsymbol{X}_j) - \sum_i r_{ji}\Phi(\boldsymbol{D}_i)\|_2^2 + \lambda\|\boldsymbol{r}_j\|_1$

- Log-Euclidean distance: $\boldsymbol{d}_{le}(\boldsymbol{X},\boldsymbol{Y}) = \|\log \boldsymbol{X} - \log \boldsymbol{Y}\|_F$
  - induces kernel $k(\boldsymbol{X},\boldsymbol{Y}) = e^{-\sigma \boldsymbol{d}_{le}(\boldsymbol{X},\boldsymbol{Y})}$ for all $\sigma > 0$
  - (Li et al. ICCV 2013)
    $\min \sum_j \|\Phi(\boldsymbol{X}_j) - \sum_i r_{ji}\Phi(\boldsymbol{D}_i)\|_2^2 + \lambda\|\boldsymbol{r}_j\|_1$

Riemannian Dictionary Learning

- Stein Divergence: $\boldsymbol{S}(\boldsymbol{X},\boldsymbol{Y}) = \log(\det \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}) - \frac{1}{2}\log(\det(\boldsymbol{X}\boldsymbol{Y}))$
  - induces kernel $k(\boldsymbol{X},\boldsymbol{Y}) = e^{-\sigma \boldsymbol{S}(\boldsymbol{X},\boldsymbol{Y})}$ for some $\sigma > 0$
  - (Harandi et al. ECCV 2012)
    $\min \sum_j \|\Phi(\boldsymbol{X}_j) - \sum_i r_{ji}\Phi(\boldsymbol{D}_i)\|_2^2 + \lambda\|\boldsymbol{r}_j\|_1$

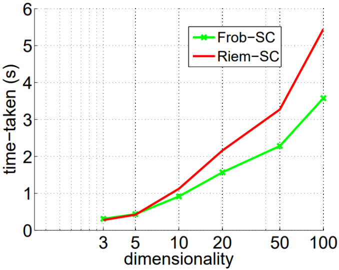
- Log-Euclidean distance: $\boldsymbol{d}_{le}(\boldsymbol{X},\boldsymbol{Y}) = \|\log \boldsymbol{X} - \log \boldsymbol{Y}\|_F$
  - induces kernel $k(\boldsymbol{X},\boldsymbol{Y}) = e^{-\sigma \boldsymbol{d}_{le}(\boldsymbol{X},\boldsymbol{Y})}$ for all $\sigma > 0$
  - (Li et al. ICCV 2013)
    $\min \sum_j \|\Phi(\boldsymbol{X}_j) - \sum_i r_{ji}\Phi(\boldsymbol{D}_i)\|_2^2 + \lambda\|\boldsymbol{r}_j\|_1$

- AIRM: $\boldsymbol{d}_R(\boldsymbol{X},\boldsymbol{Y}) = \|\log(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{X}^{-\frac{1}{2}})\|_F$
  - Geodesic distance, most faithful to Riemannian geometry
  - Hard to kernelize
  - (Cherian and Sra 2017) $\min \frac{1}{2}\sum_{j=1}^N \boldsymbol{d}_R^2(\boldsymbol{X}_j, \boldsymbol{D}\boldsymbol{r}_j) + sp(\boldsymbol{r}_j) + \Omega(\boldsymbol{D})$

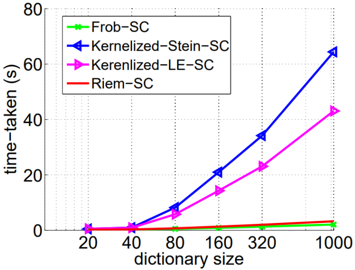
Riemannian Dictionary Learning

- Stein Divergence: $\boldsymbol{S}(\boldsymbol{X}, \boldsymbol{Y}) = \log(\det \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}) - \frac{1}{2}\log(\det(\boldsymbol{X}\boldsymbol{Y}))$
  - induces kernel $k(\boldsymbol{X}, \boldsymbol{Y}) = e^{-\sigma \boldsymbol{S}(\boldsymbol{X}, \boldsymbol{Y})}$ for some $\sigma > 0$
  - (Harandi et al. ECCV 2012)
    $\min \sum_j \|\Phi(\boldsymbol{X}_j) - \sum_i r_{ji}\Phi(\boldsymbol{D}_i)\|_2^2 + \lambda\|\boldsymbol{r}_j\|_1$
- Log-Euclidean distance: $\boldsymbol{d}_{le}(\boldsymbol{X}, \boldsymbol{Y}) = \|\log \boldsymbol{X} - \log \boldsymbol{Y}\|_F$
  - induces kernel $k(\boldsymbol{X}, \boldsymbol{Y}) = e^{-\sigma \boldsymbol{d}_{le}(\boldsymbol{X}, \boldsymbol{Y})}$ for all $\sigma > 0$
  - (Li et al. ICCV 2013)
    $\min \sum_j \|\Phi(\boldsymbol{X}_j) - \sum_i r_{ji}\Phi(\boldsymbol{D}_i)\|_2^2 + \lambda\|\boldsymbol{r}_j\|_1$
- AIRM: $\boldsymbol{d}_R(\boldsymbol{X}, \boldsymbol{Y}) = \|\log(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{X}^{-\frac{1}{2}})\|_F$
  - Geodesic distance, most faithful to Riemannian geometry
  - Hard to kernelize
  - (Cherian and Sra 2017) $\min \frac{1}{2}\sum_{j=1}^N \boldsymbol{d}_R^2(\boldsymbol{X}_j, \boldsymbol{D}\boldsymbol{r}_j) + sp(\boldsymbol{r}_j) + \Omega(\boldsymbol{D})$

Performance Comparison



(a)



(b)

Performance Comparison

Method	Accuracy (%)
Riem-DL + Riem-SC	74.9
LE-DL + LE-SC	73.4
Frob-DL + Frob SC	23.5
Kernelized LE DLSC [13]+ [34]	47.9
Kernelized Stein DLSC [12]+ [34]	**76.7**
Tensor DL+SC [52]	37.1
GDL [15]	47.7
Random DL + Riem-SC	70.3

TABLE I
BRODATZ TEXTURE DATASET

Method	Accuracy (%)
Riem-DL + Riem-SC	80.0
LE-DL + LE-SC	80.5
Frob-DL + Frob SC	54.2
Kernelized LE DLSC [13]+ [34]	**86.0**
Kernelized Stein DLSC [12]+ [34]	85.7
Tensor DL+SC [52]	68.1
GDL [15]	43.0
Random DL + Riem-SC	62

TABLE II
RGBD OBJECTS

Method	Accuracy (%)
Riem-DL + Riem-SC	80.5
LE-DL + LE-SC	80.0
Frob-DL + Frob SC	77.6
Kernelized LE DLSC [13]+ [34]	**86.6**
Kernelized Stein DLSC [12] [34]	71.6
Tensor DL+SC [52]	67.4
GDL [15]	71.0
Random DL + Riem-SC	54.6

TABLE III
ETHZ PEOPLE DATASET

Method	Accuracy (%)
Riem-DL + Riem-SC	92.4
LE-DL + LE-SC	82.6
Frob-DL + Frob SC	82.9
Kernelized LE DSC [13]+ [34]	**93.1**
Kernelized Stein DLSC [12] + [34]	70.1
GDL [15]	92.0
Random DL + Riem-SC	83.9

TABLE IV
YOUTUBE FACES DATASET

Kernel Methods on Sym_d^+

- Log-Euclidean distance is proved to be kernelizable
- Map Sym_d^+ to infinite dimensional feature space and apply Euclidean geometry
- (Jayasumana et al. CVPR 2013)

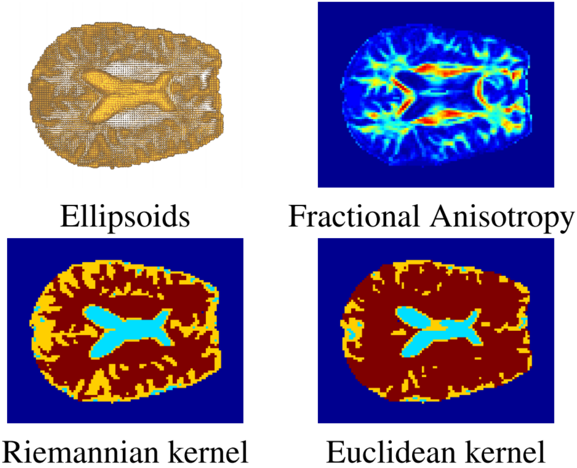
Algorithms	Input	Output
kSVM	$(\boldsymbol{X}_i, y_i)$	hyperplane in feature space
kPCA	$\boldsymbol{X}_i \in Sym_d^+$	PCA for kernel matrix
kernel k-means	$\boldsymbol{X}_i \in Sym_d^+$	k-means in feature space

Image Categorization

Nb. of classes	Euclidean		Cholesky		Power-Euclidean		Log-Euclidean	
	KM	KKM	KM	KKM	KM	KKM	KM	KKM
3	72.50	79.00	73.17	82.67	71.33	84.33	75.00	**94.83**
4	64.88	73.75	69.50	84.62	69.50	83.50	73.00	**87.50**
5	54.80	70.30	70.80	82.40	70.20	82.40	74.60	**85.90**
6	50.42	69.00	59.83	73.58	59.42	73.17	66.50	**74.50**
7	42.57	68.86	50.36	69.79	50.14	69.71	59.64	**73.14**
8	40.19	68.00	53.81	69.44	54.62	68.44	58.31	**71.44**

Applications of Kernel Methods

Image Segmentation



Summary

- AIRM
  - Pros: geodesic distance
  - Cons: hard to kernelize

Summary

- AIRM
  - Pros: geodesic distance
  - Cons: hard to kernelize
- Stein Divergence
  - Pros: kernelizable, but only for certain parameters; efficient
  - Cons: not geodesic distance

Summary

- AIRM
  - Pros: geodesic distance
  - Cons: hard to kernelize
- Stein Divergence
  - Pros: kernelizable, but only for certain parameters; efficient
  - Cons: not geodesic distance
- Log-Euclidean
  - Pros: kernelizable; geodesic

Summary

- AIRM
  - Pros: geodesic distance
  - Cons: hard to kernelize
- Stein Divergence
  - Pros: kernelizable, but only for certain parameters; efficient
  - Cons: not geodesic distance
- Log-Euclidean
  - Pros: kernelizable; geodesic
- Geodesic: AIRM > Log-Euclidean > Stein

Summary

- AIRM
  - Pros: geodesic distance
  - Cons: hard to kernelize
- Stein Divergence
  - Pros: kernelizable, but only for certain parameters; efficient
  - Cons: not geodesic distance
- Log-Euclidean
  - Pros: kernelizable; geodesic
- Geodesic: AIRM > Log-Euclidean > Stein
- Efficiency: Stein $\approx$ Log-Euclidean > AIRM

Summary

- AIRM
  - Pros: geodesic distance
  - Cons: hard to kernelize
- Stein Divergence
  - Pros: kernelizable, but only for certain parameters; efficient
  - Cons: not geodesic distance
- Log-Euclidean
  - Pros: kernelizable; geodesic
- Geodesic: AIRM > Log-Euclidean > Stein
- Efficiency: Stein $\approx$ Log-Euclidean > AIRM
- Kernelization: Log-Euclidean > Stein > AIRM

Future Directions

- Algorithm designed for large SPD matrices (Harandi et al. 2017)
- Riemannian Networks (Huang and Gool 2017)
- Theory of kernel methods on AIRM

Conclusion

- Certain non-Euclidean metrics are more faithful to the Sym_d^+
- Selecting correct metric is important
- Geometry-aware methods are emerging in the past 5 years
- Geometry-aware methods for other manifolds are also being studied