

Apprendre des distributions sur les variétés riemanniennes

Auto encodeur pour matrices SPD

Charlotte BOUCHERIE, Florian YGER, Thibault DE SURREL

LITIS

2025

Table des matières

Table des matières

A Riemannian Network for SPD Matrix Learning

- Introduction d'une architecture de réseau qui conserve les propriétés des matrices définies positives pour le Deep Learning
- 3 couches différentes : BiMap, ReEig, LogEig
- On va se baser sur ces couches pour notre auto-encodeur

DreamNet: A Deep Riemannian Manifold Network for SPD Matrix Learning

- Méthodologie pour créer des réseaux profonds

Riemannian Multinomial Logistics Regression for SPD Neural Networks

- Introduction de la régression logistique pour les matrices SPD
- Nouvelle couche pour la classification

- Auto-encodeur pour matrices SPD
- Couche pour faire les opérations inverses de l'auto encodeur
- Comparaison de différents modèles

- Aplatir l'espace vers l'espace tangent
- Approximation de la distance géodésique : distance de Frobenius

$$L = \|A - B\|_F = \sqrt{\sum_{i,j} (A_{ij} - B_{ij})^2}$$

- Ne préserve pas la structure géométrique
- Effet de gonflement (*swelling effect*) : les déterminants de l'interpollation des matrices aplattis

On veut donc prendre en compte la courbure et la non-linéarité de l'espace des matrices SPD

Métrique de Riemann (*AIRM*) : $\delta_R^2(X, Y) = \|\log(X^{-1/2} Y X^{-1/2})\|_F^2$

- Permet de mesurer la similarité entre deux matrices SPD tout en respectant la structure
- On l'utilisera dans notre AE dans le modèle, dans le coût et dans la confiance.

Table des matières

Principes de base de l'auto-encodeur

- Réseau de neurones
- Apprentissage non supervisé
- Réduction de dimensions
- Apprends les patrons sous-jacents
- Utilisés pour les modèles génératifs

$\phi : \mathcal{X} \rightarrow \mathcal{F}$, l'encodeur

$\psi : \mathcal{F} \rightarrow \mathcal{X}$, le décodeur

$$\phi, \psi = \arg \min_{\phi, \psi} ||X - (\psi \circ \phi)X||^2$$

- La fonction de cette couche est de générer des matrices SPD plus compactes et plus discriminantes.
- On a donc une couche qui effectue une application bilinéaire f_b pour transformer les matrices initiales en nouvelles matrices de dimension inférieure.

$$X_k = f_b^{(k)}(X_{k-1}; W_k) = W_k X_{k-1} W_k^T$$

W_k est de rang plein pour garantir que X_k reste SPD.

Paramètres dans le réseau

Nombre de filtres/canaux d'entrée h_i , nombre de filtres/canaux de sorties h_o , taille de la matrice d'entrée n_i , taille de la matrice de sortie n_o

- La fonction de cette couche est d'améliorer les performances discriminantes en introduisant une non linéarité, de la même manière que ReLU.
- On introduit donc une fonction non-linéaire f_r qui corrige les matrices en posant un seuil aux valeurs propres faibles.

$$X_k = f_r^{(k)}(X_{k-1}) = U_{k-1} \max(\epsilon I, \Sigma_{k-1}) U_{k-1}^T$$

LogEig

La fonction de cette couche est de pouvoir appliquer la géométrie de Riemann au matrice sortante.

$$X_k = f_l^{(k)}(X_{k-1}) = \log(X_{k-1}) = U_{k-1} \log(\Sigma_{k-1}) U_{k-1}^T$$

ExpEig

La fonction de cette couche est d'appliquer la fonction inverse de la couche LogEig

$$X_k = f_e^{(k)}(X_{k-1}) = \exp(X_{k-1})$$

- On a une seule couche BiMap pour l'encodeur qui va de $ni \rightarrow no$ et de $hi \rightarrow ho$.
- On regarde l'influence de la dimension de sortie et de la couche de sortie.
- Le décodeur fait l'opération inverse.

- On a deux couches BiMap.
 - $ni \rightarrow ni/2$ et de $hi \rightarrow ho$.
 - $ni/2 \rightarrow no$ et de $ho \rightarrow hi$.
- On regarde l'influence du nombre de filtres intermediaires et de la dimension de sortie.
- Le décodeur fait l'opération inverse.

Modèle : plusieurs couches régulières

Modèle : taille de couches réduite de moitié

- Pour chaque matrice, on prend ses k matrices les plus proches dans l'espace de sortie et ses matrices les plus proches dans l'espace d'arrivée.
- La distance est la même utilisée que pour calculer notre fonction de coût.
- On pénalise proportionnellement au rang différent dans l'espace d'entrée.
- On ne pénalise pas les rapprochements.

$$T(k) = 1 - \frac{2}{nk(2n-3k-1)} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i^k} \max(0, (r(i, j) - k))$$

- On utilise la MDM (Minimum Distance to Mean) pour connaître la précision avant la reconstitution de nos matrices.
- Pour chaque classe, un centroïde est estimé selon notre distance.
- On compare la précision initiale avec la précision finale.

Table des matières

Ajout de bruits

Bruit gaussien

Bruit poivre et sel

Bruit de masquage

- On prédit moins bien les données après le passage dans l'auto-encodeur
- On conserve les voisinages
- On compresse avec perte