Towards Adaptive Classification using Riemannian Geometry approaches in Brain-Computer Interfaces

Satyam Kumar[1,2] Florian Yger[3] Fabien Lotte[2]

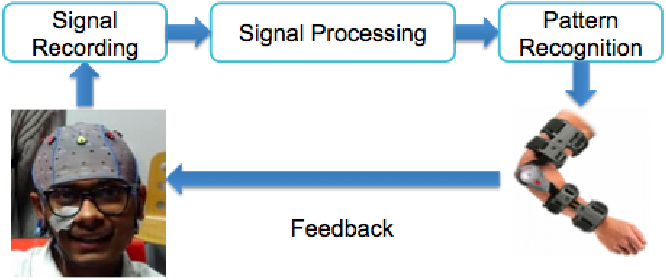
[1]Department of Electrical Engineering
IIT Kanpur, India

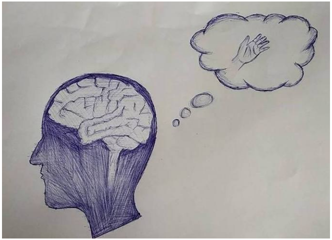
[2]Inria Bordeaux - Sud-Ouest, France

[3]LAMSADE, Paris-Dauphine University, France

International Winter Conference on Brain-Computer Interface

Brain-computer interface (BCI)

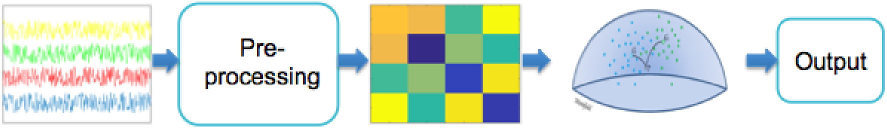




- Motor Imagery (MI) BCI
- Event related synchronisation/de-synchronisation

Motor Imagery classification

- Riemannian geometry [1] framework [2] to efficient
  1. DecMEG2014
  2. BCI challenge 2015



[1] Yger, F., Berar, M. and Lotte, F., 2017. Riemannian approaches in brain-computer interfaces: a review.

[2] Lotte F, Bougrain L, Cichocki A, Clerc M, Congedo M, Rakotomamonjy A, Yger F. A review of classification algorithms for EEG-based braincomputer interfaces: a 10 year update

Challenges in BCI

- Presence of non-stationarity

[3] Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P. and Muller, K.R., 2006. Towards adaptive classification for BCI.

Challenges in BCI

- Presence of non-stationarity
- Shift in data distribution

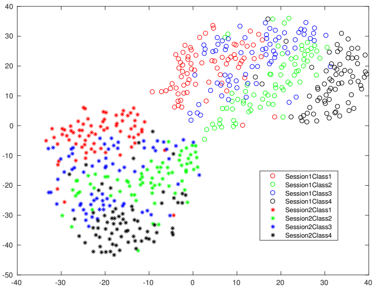


Figure: 2D visualisation of covariance matrices

[3] Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P. and Muller, K.R., 2006. Towards adaptive classification for BCI.

Challenges in BCI

- Presence of non-stationarity
- Shift in data distribution

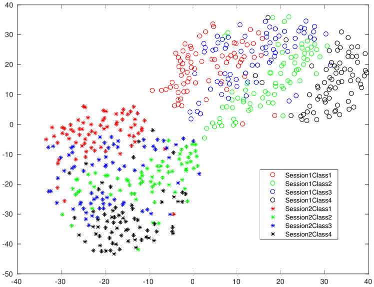


Figure: 2D visualisation of covariance matrices

- Need for adaptive techniques [3]

[3] Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P. and Muller, K.R., 2006. Towards adaptive classification for BCI.

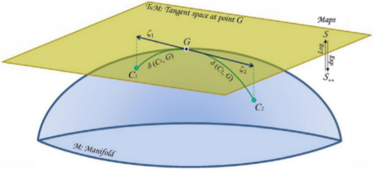
Objective

- Motor imagery classification
- Formulation of Adaptive Riemannian geometry frameworks

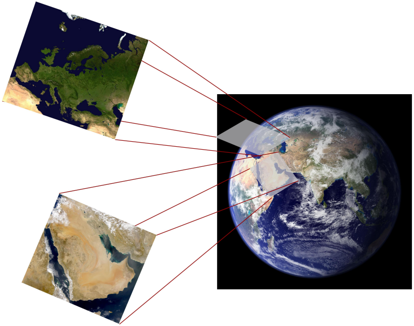
Adaptive methods

- REBIAS [a] - Unsupervised adaptation
- RETRAIN - Supervised | Unsupervised
- Hybrid REBIAS-RETRAIN

[a] Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P. and Muller, K.R., 2006. Towards adaptive classification for BCI.



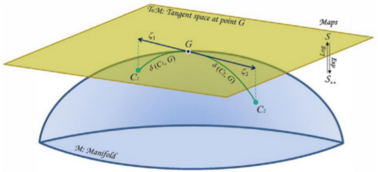
[Barachant et al. 2017]



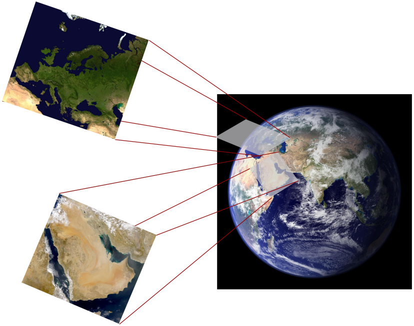
[Yger et al. 2016]

- Smooth differentiable manifold

Riemannian geometry framework



[Barachant et al. 2017]



[Yger et al. 2016]

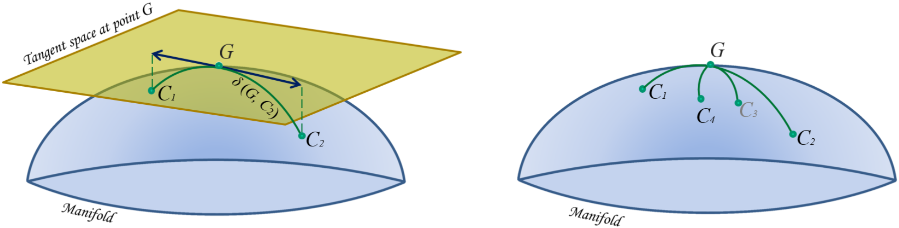
- Smooth differentiable manifold
- Finite dimensional tangent space at each point on manifold
- Tangent space vectors are euclidean
- Logarithmic and Exponential maps to transport the points from manifold to tangent space and vice versa

Riemannian Geometry tools

- Equation of Geodesic and Riemannian distance

$$\gamma(C_1, C_2, t) = C_1^{\frac{1}{2}} \left(C_1^{-\frac{1}{2}} C_2 C_1^{-\frac{1}{2}} \right)^t C_1^{\frac{1}{2}} \quad t \in [0, 1]$$

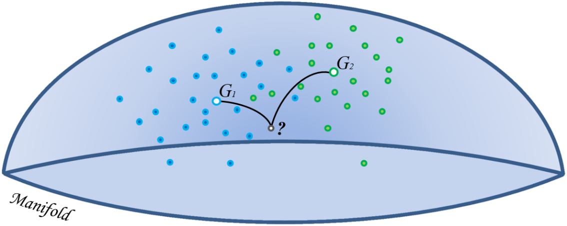
- Karcher mean



[Lotte et al. 2017]

Minimum Distance to Mean classifier (MDM)

- Minimum distance to mean classifier : A distance based classifier
- Uses Karcher mean and riemannian distance on manifold of SPD matrices for classification



[Lotte et al. 2017]

- **Parameters**: class specific means $\{\bar{C}_1, \bar{C}_2, ..., \bar{C}_K\}$
- Simple and elegant, highly efficient classification

Filter geodesic MDM (FgMDM) classifier

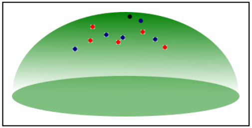


Figure: Matrices on manifold

Filter geodesic MDM (FgMDM) classifier

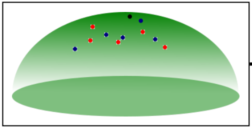


Figure: Matrices on manifold

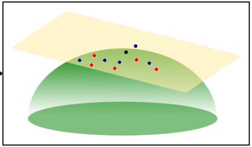


Figure: Tangent space mapping at reference

- **Parameters**: C_{ref}, $\{\bar{C}_1, \bar{C}_2, ..., \bar{C}_K\}$, Discriminant filters (W)

Filter geodesic MDM (FgMDM) classifier

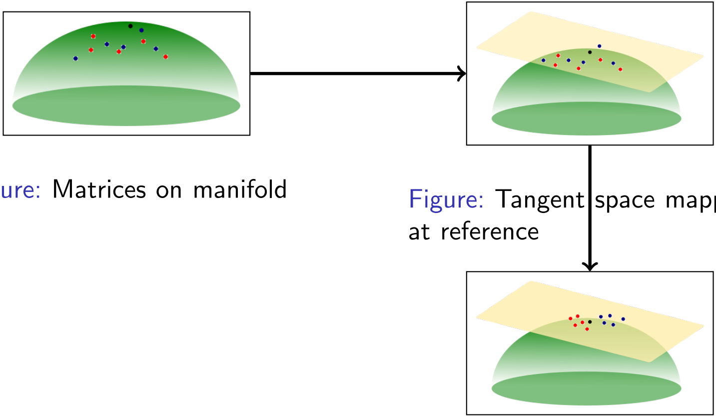


Figure: Matrices on manifold

Figure: Tangent space mapping at reference

Figure: Discriminant filtering

- **Parameters**: C_{ref}, $\{\bar{C}_1, \bar{C}_2, ..., \bar{C}_K\}$, Discriminant filters (W)

Filter geodesic MDM (FgMDM) classifier

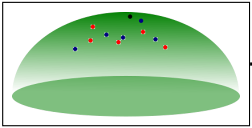


Figure: Matrices on manifold

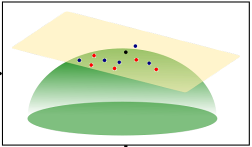


Figure: Tangent space mapping at reference

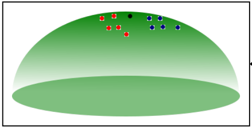


Figure: Back projection on manifold

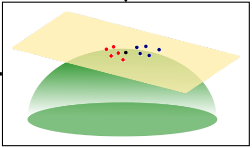


Figure: Discriminant filtering

- **Parameters**: C_{ref}, $\{\bar{C}_1, \bar{C}_2, ..., \bar{C}_K\}$, Discriminant filters (W)

Supervised Adaptation in RGC classifier

MDM

- Geodesic update of class prototypes

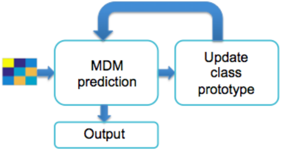
$$\bar{C}_i^k = \begin{cases} \gamma\left(\bar{C}_{i-1}^k, C^k, \frac{1}{N_{\bar{C}_{i-1}^k}+1}\right) & i > 0 \\ \bar{C}_{train}^k & i = 0 \end{cases}$$

Supervised Adaptation in RGC classifier

MDM

- Geodesic update of class prototypes

$$\bar{C}_i^k = \begin{cases} \gamma\left(\bar{C}_{i-1}^k, C^k, \frac{1}{N_{\bar{C}_{i-1}^k}+1}\right) & i > 0 \\ \bar{C}_{train}^k & i = 0 \end{cases}$$

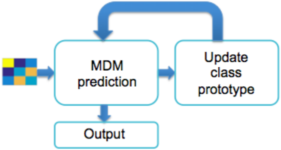


Supervised Adaptation in RGC classifier

MDM

- Geodesic update of class prototypes

$$\bar{C}_i^k = \begin{cases} \gamma\left(\bar{C}_{i-1}^k, C^k, \frac{1}{N_{\bar{C}_{i-1}^k}+1}\right) & i > 0 \\ \bar{C}_{train}^k & i = 0 \end{cases}$$



FgMDM

- Geodesic update of reference covariance matrix

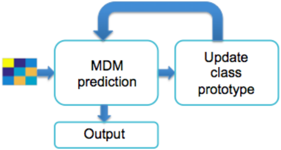
$$\bar{C}_i = \begin{cases} \gamma\left(\bar{C}_{i-1}, C, \frac{1}{N_{\bar{C}_{i-1}}+1}\right) & i \geq 1 \\ \bar{C}_{train} & i = 0 \end{cases}$$

Supervised Adaptation in RGC classifier

MDM

- Geodesic update of class prototypes

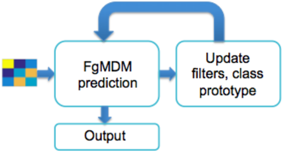
$$\bar{C}_i^k = \begin{cases} \gamma\left(\bar{C}_{i-1}^k, C^k, \frac{1}{N_{\bar{C}_{i-1}^k}+1}\right) & i > 0 \\ \bar{C}_{train}^k & i = 0 \end{cases}$$



FgMDM

- Geodesic update of reference covariance matrix

$$\bar{C}_i = \begin{cases} \gamma\left(\bar{C}_{i-1}, C, \frac{1}{N_{\bar{C}_{i-1}}+1}\right) & i \geq 1 \\ \bar{C}_{train} & i = 0 \end{cases}$$



Rebias Adaptation in RGC

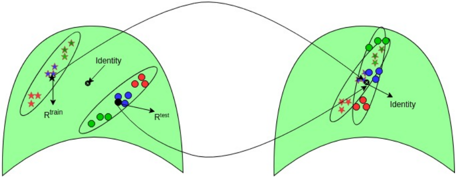
- Covariance matrices are moved to a common reference [Zanini et al. 2018]

$$C_i^{Rebiased} = R^{-\frac{1}{2}} C_i R^{-\frac{1}{2}} \quad R \in \{R^{test}, R^{train}\}$$

Rebias Adaptation in RGC

- Covariance matrices are moved to a common reference [Zanini et al. 2018]

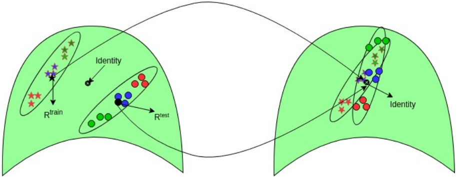
$$C_i^{Rebiased} = R^{-\frac{1}{2}} C_i R^{-\frac{1}{2}} \quad R \in \{R^{test}, R^{train}\}$$

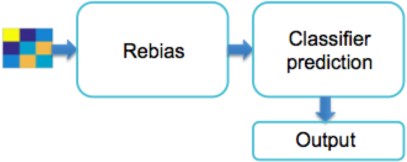


Rebias Adaptation in RGC

- Covariance matrices are moved to a common reference [Zanini et al. 2018]

$$C_i^{Rebiased} = R^{-\frac{1}{2}} C_i R^{-\frac{1}{2}} \quad R \in \{R^{test}, R^{train}\}$$



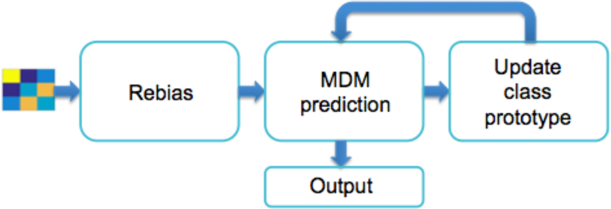


- Incremental geodesic adaptation of rebiasing matrix

$$R_i^{test} = \begin{cases} R^{train} & i = 1 \\ C_{i-1} & i = 2 \\ \gamma\left(R_{i-1}^{test}, C_i, \frac{1}{i-1}\right) & i \geq 3 \end{cases}$$

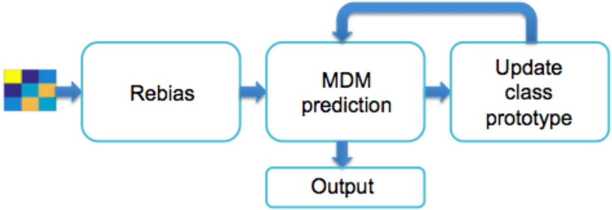
Supervised Rebias Adaptation in RGC

- MDM classifier

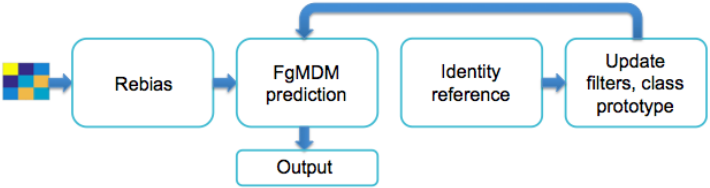


Supervised Rebias Adaptation in RGC

- MDM classifier



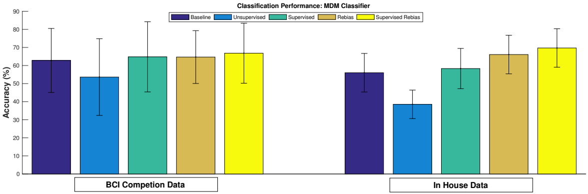
- FgMDM classifier



Dataset

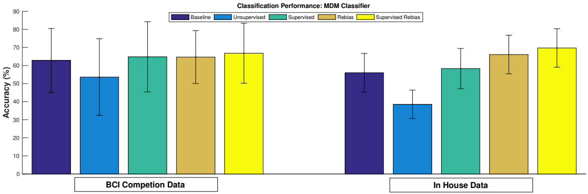
1. BCI competition dataset
   - 9 subjects, two session
   - Four class motor imagery (left hand, right hand, foot ,tongue)
   - 72 trial per class per session
   - Performance averaged across subjects on session 2
2. In house dataset
   - 18 subjects, 6 session
   - Three class Mental imagery (left hand motor imagery, mental rotation, mental subtraction)
   - 75 trial per class per session
   - Performance averaged across subjects on session 2 to 6

Results: MDM classifier



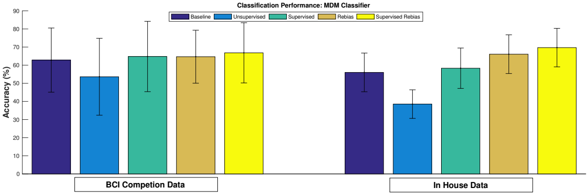
- Unsupervised Adaptation results in performance drop

Results: MDM classifier



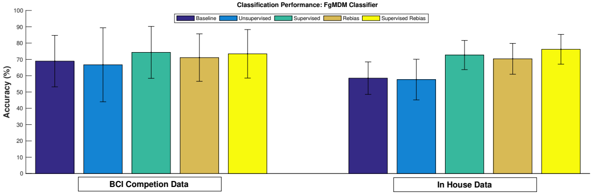
- Unsupervised Adaptation results in performance drop
- Supervised and Rebias MDM adaptation increases the performance

Results: MDM classifier



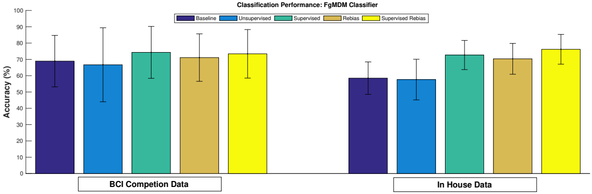
- Unsupervised Adaptation results in performance drop
- Supervised and Rebias MDM adaptation increases the performance
- Supervised Rebias MDM adaptation outperforms all the adaptation schemes

Results: FgMDM classifier



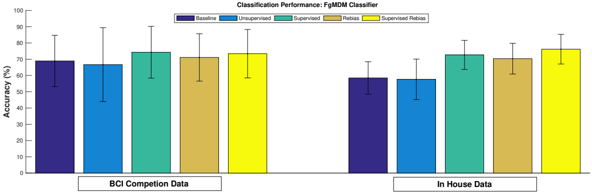
- Performance drop for unsupervised adaptation is smaller compared to MDM

Results: FgMDM classifier



- Performance drop for unsupervised adaptation is smaller compared to MDM
- Supervised, Rebias and Supervised Rebias results in performance increase

Results: FgMDM classifier



- Performance drop for unsupervised adaptation is smaller compared to MDM
- Supervised, Rebias and Supervised Rebias results in performance increase
- Supervised and Supervised Rebias higher computational time
- Rebias adaptation for both RGC's is efficient and fast!

Conclusion

Contributions

- Adaptation framework for Riemannian geometry classifiers (RGC)
- Robust method for adaptation of riemannian mean in BCI framework
- Efficient adaptation schemes and most of them are fast
- Demonstrated that adaptation bolsters the performance over static RGC

Conclusion

Contributions

- Adaptation framework for Riemannian geometry classifiers (RGC)
- Robust method for adaptation of riemannian mean in BCI framework
- Efficient adaptation schemes and most of them are fast
- Demonstrated that adaptation bolsters the performance over static RGC

Future works

- Determining optimal rate of adaptation
- Implementation for real time motor imagery
- Extension of proposed approaches for P300 and SSVEP

Thank you for your attention!

Riemannian Geometry tools

- Geodesic between two SPD matrices P_1 and P_2

$$\gamma(C_1, C_2, t) = C_1^{\frac{1}{2}} \left(C_1^{-\frac{1}{2}} C_2 C_1^{-\frac{1}{2}} \right)^t C_1^{\frac{1}{2}} \quad t \in [0,1]$$

- Riemannian distance between two SPD matrices P_1 and P_2

$$\delta_r(C_1, C_2) = \int_0^1 \gamma(C_1, C_2, t) dt = \left\| \log(C_1^{-\frac{1}{2}} C_2) \right\|_{\mathcal{F}}$$

- Affine invariance of of riemannian distance

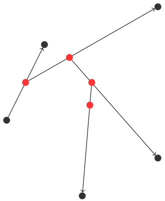
$$\delta_r(W^T C_1 W, W^T C_2 W) = \delta_r(C_1, C_2) \quad W \in GL(n)$$

- Karcher mean of SPD matrices

$$\bar{C} = \underset{C \in \mathcal{C}_n}{\arg\min} \sum_{i=1}^{N} \delta_r^2(C_i, C)$$

Adaptive estimation of Riemannian Mean

- Equation of line :
  $y(C_1, C_2, t) = tC_1 + (1-t)C_2 \quad t \in [0,1]$



$$m1 = C1, m2 = \frac{1}{2}(m1 + C2), m3 = \frac{2}{3}m1 + \frac{1}{3}C2$$
$$m4 = \frac{3}{4}m3 + \frac{1}{4}C3 \quad \cdots$$

- Equation of Geodesic

$$\gamma(C_1, C_2, t) = C_1^{\frac{1}{2}}\left(C_1^{-\frac{1}{2}} C_2 C_1^{-\frac{1}{2}}\right)^t C_1^{\frac{1}{2}} \quad t \in [0,1]$$