Geometric transfer learning with PyRiemann

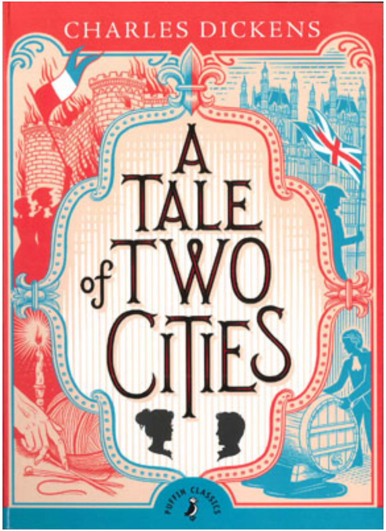
Sylvain Chevallier
Inria TAU team, LISN, Université Paris-Saclay, France
sylvain.chevallier@universite-paris-saclay.fr

W6: Learning from small datasets

7 June 2023

A tale of two journeys

- Revolution of transfer learning
- Different paths, common goal
- Meet within Pyriemann



Two approaches on transfer learning

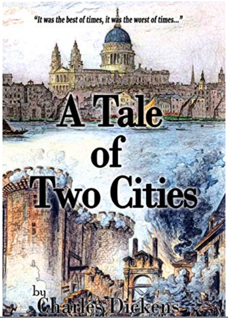
Definitions

Source domain $D_{\mathcal{S}}$ and task $\mathcal{T}_{\mathcal{S}}$: data from different subjects or datasets
Target domain D_t and task $\mathcal{T}_t$: data to analyze

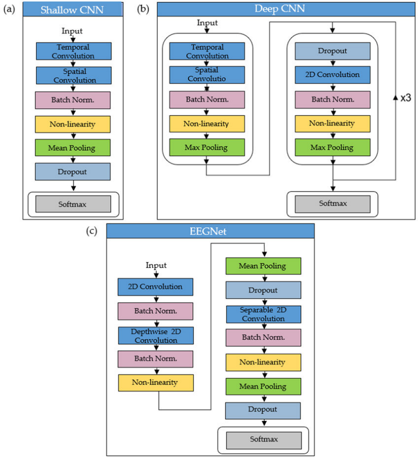
Transfer learning overcomes $D_{\mathcal{S}} \neq D_t$ and $\mathcal{Y}_{\mathcal{S}} \neq \mathcal{Y}_t$ (labels of $\mathcal{T}_{\mathcal{S}}$ and $\mathcal{T}_t$)

Domain adaptation: adapt trials from source subject to be most useful for a target subject

Domain generalisation: transform all data available, make it useful for any subject



Deep transfer learning



[Cooney, 2020]

- raw processing
- or EEG embedding
- encode sujbect information
- train on auxilliary unsupervised task
- use transformers/attention

Learn latent space for good discriminability

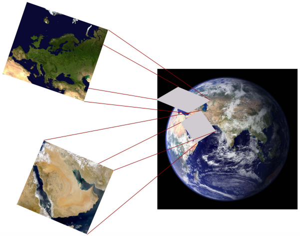
⇒ Require huge dataset

Geometric transfer learning

- Covariance estimation $\Sigma^{(y)}$, for class $y \in \mathcal{Y}$
- Or functional connectivity representations
- Classifier on manifold of SPD matrices
- Or on tangent space

Good generalization properties of affine-invariant metrics

$\Rightarrow$ Limitation in high-dimensional regime



[Yger, 2016]

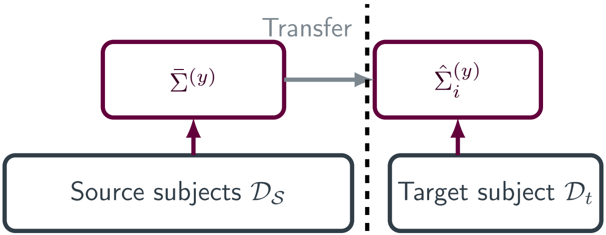
Domain adaptation: blend in the mix

Modify target data with source knowledge

$$\text{Euclidean composite:} \quad \hat{\Sigma}_i^{(y)} = \lambda\bar{\Sigma}^{(y)} + (1-\lambda)\Sigma_i^{(y)}$$

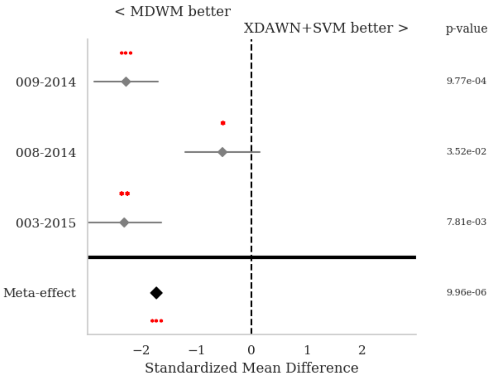
$$\text{Riemannian composite:} \quad \hat{\Sigma}_i^{(y)} = \text{geodesic}_\lambda\left(\bar{\Sigma}^{(y)}, \Sigma_i^{(y)}\right)$$

$\Rightarrow$ Works without calibration data ($\lambda = 1$) or with few samples ($0 \leqslant \lambda < 1$)

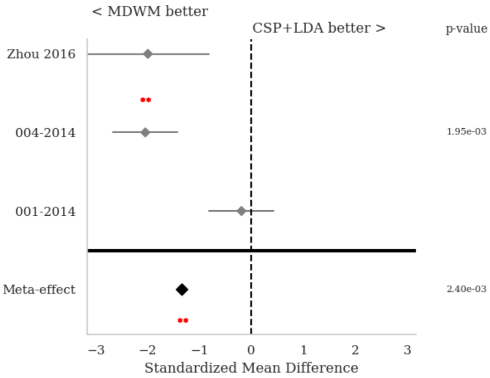


Good cross-subject performances

P300 datasets



Multiclass MI datasets



[Khazem, 2021]

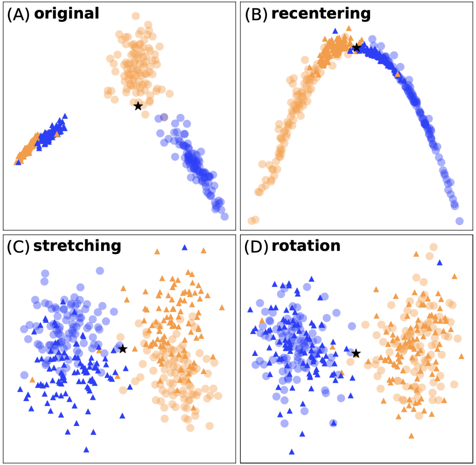
Domain generalization: all samples, unite!

Euclidean or Riemannian alignment

Riemannian Procrustes Analysis

1. Recenter
2. Rescale
3. Rotate

Train classifier on aligned data



[Rodrigues, 2018]

pyRiemann

`pip install pyriemann`

Documentation: pyriemann.readthedocs.io

Github: github.com/pyRiemann/pyRiemann

universite PARIS-SACLAY

Transfer learning in pyRiemann

```
from pyriemann.embedding import SpectralEmbedding
from pyriemann.datasets.simulated import make_classification_transfer
from pyriemann.transfer import (
    decode_domains,
    TLCenter,
    TLRotate,
)
```

```
# create source and target datasets
n_matrices = 50
X_enc, y_enc = make_classification_transfer(
    n_matrices=n_matrices,
    class_sep=2.0,
    class_disp=0.25,
    domain_sep=2.0,
    theta=np.pi/4,
    random_state=seed
)

# generate dataset
X_org, y, domain = decode_domains(X_enc, y_enc)
```

Hierarchical label representation

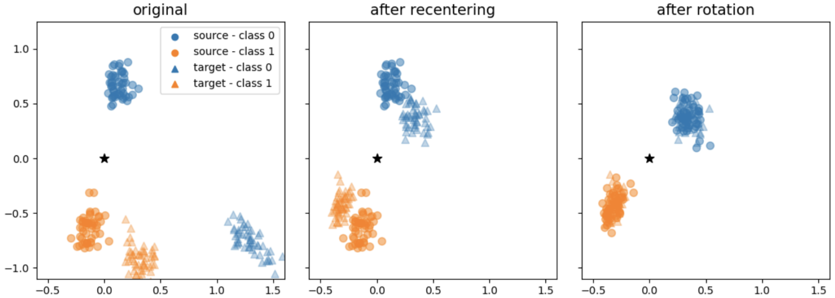
label is dataset/session/subject/class
eg. `BNCI2014001/sess1/subj6/lefthand`

universite PARIS-SACLAY

Transfer learning in pyRiemann

```
# embed the source and target datasets after recentering
rct = TLCenter(target_domain='target_domain')
X_rct = rct.fit_transform(X_org, y_enc)
points = np.concatenate([X_rct, np.eye(2)[None, :, :]])  # stack the identity
embedded_points['rct'] = emb.fit_transform(points)

# embed the source and target datasets after recentering
rot = TLRotate(target_domain='target_domain', metric='riemann')
X_rot = rot.fit_transform(X_rct, y_enc)
```



Thank you !