Sparse Wavelet Representations and Classification

Challenge Kaggle in class

DREEM

March 5th 2015

Introduction

- Context: challenge kaggle in class proposed by the start-up Dreem. Binary classification problem.

Introduction

- Context: challenge kaggle in class proposed by the start-up Dreem. Binary classification problem.
- Data: EEG signals, 3 sec sampled at 200 Hz on 20 subjects.

Introduction

- Context: challenge kaggle in class proposed by the start-up Dreem. Binary classification problem.
- Data: EEG signals, 3 sec sampled at 200 Hz on 20 subjects.
- Labels: 1 if a Slow Oscillation occurs during the next 0.5 sec, 0 else.

Introduction

- Context: challenge kaggle in class proposed by the start-up Dreem. Binary classification problem.
- Data: EEG signals, 3 sec sampled at 200 Hz on 20 subjects.
- Labels: 1 if a Slow Oscillation occurs during the next 0.5 sec, 0 else.
- Skewed data set: only 7% of positive exs.

Introduction

- Context: challenge kaggle in class proposed by the start-up Dreem. Binary classification problem.
- Data: EEG signals, 3 sec sampled at 200 Hz on 20 subjects.
- Labels: 1 if a Slow Oscillation occurs during the next 0.5 sec, 0 else.
- Skewed data set: only 7% of positive exs.
- AUC metric.

The data

The AUC metric

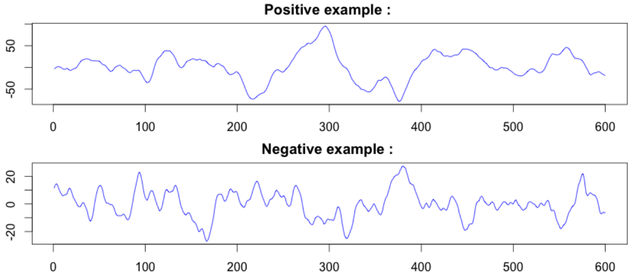
Classification algorithms we tried

- ScatNet + AUC gradient descent (Herschtal, Raskutti, ICML 2004
- ScatNet + Mixture of Probabilistic Principal Component Analysis (MPPCA) (Tipping, Bishop, 1999)
- Neural Networks
- Idea (cheat ?) use subjects IDs

Results and Conclusion

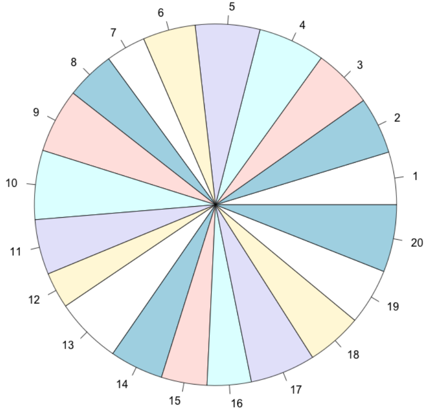
The data

- Times series



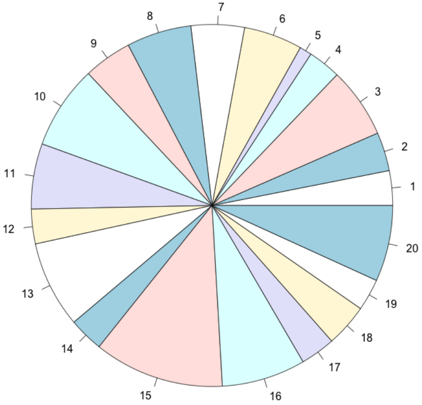
The data

- Times series
- Proportion of each subject in the training set



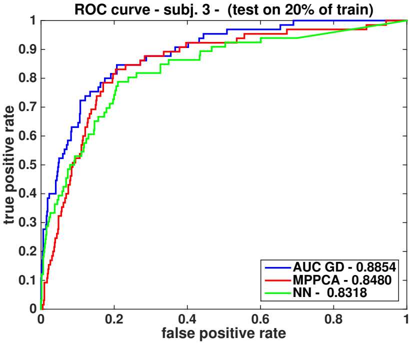
The data

- Times series
- Proportion of each subject in the training set
- Positive rate for each subject in the training set



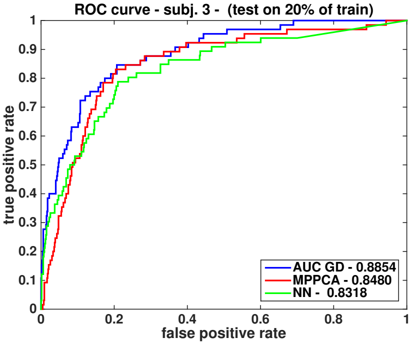
The AUC metric

- Area under ROC curve



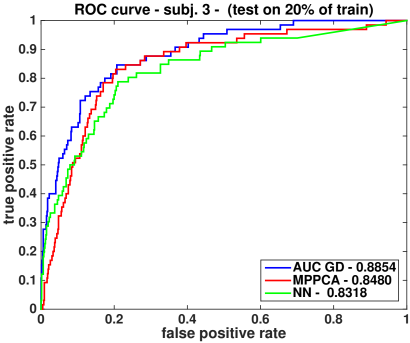
The AUC metric

- Area under ROC curve
- Estimator (Wilcoxon-Man-Whitney) : $\mathbb{P}(f(x^+) > f(x^-))$



The AUC metric

- Area under ROC curve
- Estimator (Wilcoxon-Man-Whitney) : $\mathbb{P}(f(x^+) > f(x^-))$
- Adapted to skewed data



AUC gradient descent

- linear classifier :

$$\hat{AUC}(\vec{\beta}) = \frac{1}{PQ} \sum_{j=1}^{P} \sum_{k=1}^{Q} g(\vec{\beta}.(x_j^+ - x_k^-)) \qquad (1)$$

AUC gradient descent

- linear classifier :

$$\hat{AUC}(\vec{\beta}) = \frac{1}{PQ} \sum_{j=1}^{P} \sum_{k=1}^{Q} g(\vec{\beta}.(x_j^+ - x_k^-)) \qquad (1)$$

- softening (for differentiability)

$$R(\vec{\beta}) = \frac{1}{PQ} \sum_{j=1}^{P} \sum_{k=1}^{Q} sigmoid(\vec{\beta}.(x_j^+ - x_k^-)) \qquad (2)$$

AUC gradient descent

- linear classifier :

$$\hat{AUC}(\vec{\beta}) = \frac{1}{PQ} \sum_{j=1}^{P} \sum_{k=1}^{Q} g(\vec{\beta}.(x_j^+ - x_k^-)) \qquad (1)$$

- softening (for differentiability)

$$R(\vec{\beta}) = \frac{1}{PQ} \sum_{j=1}^{P} \sum_{k=1}^{Q} sigmoid(\vec{\beta}.(x_j^+ - x_k^-)) \qquad (2)$$

- computationally feasible estimator :

$$R_l(\vec{\beta}) = \frac{1}{Q} \sum_{k=1}^{Q} sigmoid(\vec{\beta}.(x_{kmodP}^+ - x_k^-)) \qquad (3)$$

AUC gradient descent

- linear classifier :

$$\hat{AUC}(\vec{\beta}) = \frac{1}{PQ}\sum_{j=1}^{P}\sum_{k=1}^{Q} g(\vec{\beta}.(x_j^+ - x_k^-)) \qquad (1)$$

- softening (for differentiability)

$$R(\vec{\beta}) = \frac{1}{PQ}\sum_{j=1}^{P}\sum_{k=1}^{Q} sigmoid(\vec{\beta}.(x_j^+ - x_k^-)) \qquad (2)$$

- computationally feasible estimator :

$$R_l(\vec{\beta}) = \frac{1}{Q}\sum_{k=1}^{Q} sigmoid(\vec{\beta}.(x_{kmodP}^+ - x_k^-)) \qquad (3)$$

- progressively increase norm of $\beta \implies$ close to real AUC + no local maximum

MPPCA

- Generative model

$$t|x \sim \mathcal{N}(Wx + \mu; \sigma^2 I_d) \quad , \qquad x \sim \mathcal{N}(0; I_q)$$

MPPCA

- Generative model

$$t|x \sim \mathcal{N}(Wx + \mu; \sigma^2 I_d) \quad , \qquad x \sim \mathcal{N}(0; I_q)$$

- Idea : train a model for each class

MPPCA

- Generative model

$$t|x \sim \mathcal{N}(Wx + \mu; \sigma^2 I_d) \quad , \qquad x \sim \mathcal{N}(0; I_q)$$

- Idea : train a model for each class
- Probabilistic model $\Rightarrow$ independent of ratio $+/-$

MPPCA

- Generative model

$$t|x \sim \mathcal{N}(Wx + \mu; \sigma^2 I_d) \quad , \qquad x \sim \mathcal{N}(0; I_q)$$

- Idea : train a model for each class
- Probabilistic model $\Rightarrow$ independent of ratio $+/-$
- Compare probabilities of belonging to each class

Neural Networks

- Apply on raw data (and not on top of ScatNet) not to overfit.

Neural Networks

- Apply on raw data (and not on top of ScatNet) not to overfit.
- Cross-validations in the hyper-parameters space.

Neural Networks

- Apply on raw data (and not on top of ScatNet) not to overfit.
- Cross-validations in the hyper-parameters space.
- Used as a black box.

Neural Networks

- Apply on raw data (and not on top of ScatNet) not to overfit.
- Cross-validations in the hyper-parameters space.
- Used as a black box.
- Best architecture : 4 hidden layers with 14,13,10,9 neurons, and 400 iterations.

Use subjects IDs

- Idea 1 : different behaviors among subjects $\Rightarrow$ train 1 classifier/subject

Use subjects IDs

- Idea 1 : different behaviors among subjects $\Rightarrow$ train 1 classifier/subject
- ... lead to overfitting
- and not useful for “ready trained” device

Conclusion

MPPCA	NN	AUC GD
0.81578	0.84976	0.86933

Conclusion

MPPCA	NN	AUC GD
0.81578	0.84976	0.86933

- Best score : linear classifier on scattering transforms

Conclusion

MPPCA	NN	AUC GD
0.81578	0.84976	0.86933

- Best score : linear classifier on scattering transforms
- NN : blind but good

Conclusion

MPPCA	NN	AUC GD
0.81578	0.84976	0.86933

- Best score : linear classifier on scattering transforms
- NN : blind but good
- All algorithms : less than 15 minutes on a laptop for training

Conclusion

MPPCA	NN	AUC GD
0.81578	0.84976	0.86933

- Best score : linear classifier on scattering transforms
- NN : blind but good
- All algorithms : less than 15 minutes on a laptop for training
- Possible improvements : ScatNet + AUCSVM (Brefeld, Scheffer) ? ScatNet+SVM-RBF cross-validated ?

Conclusion

MPPCA	NN	AUC GD
0.81578	0.84976	0.86933

- Best score : linear classifier on scattering transforms
- NN : blind but good
- All algorithms : less than 15 minutes on a laptop for training
- Possible improvements : ScatNet + AUCSVM (Brefeld, Scheffer) ? ScatNet+SVM-RBF cross-validated ?
- Reinforcement learning to adapt the algorithm to the subject using the device.