Predicting change points in time series data

Haidar Khan

Overview

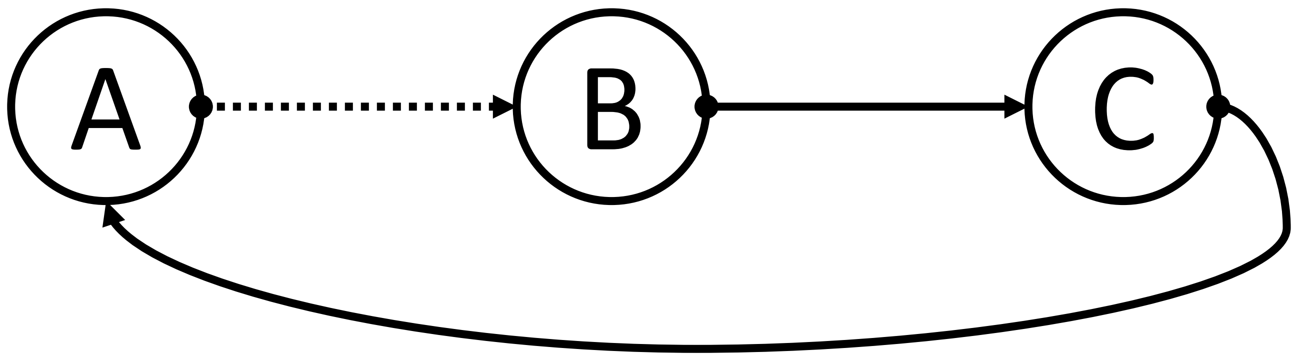
- Problem
- Literature review
- Completed Work
    - Theory
    - Approach and Evaluation
    - Application: Seizure Prediction
    - Results and Challenges
- Proposed Work

Setting and Motivation

- We observe a system that generates a time series signal while transitioning between states
- With a dataset of time series with labeled states, we can train a discriminative model
- Use model to predict next states given previous data.

- Predicting pre-seizure transition in patients with epilepsy*
- Computer network intrusion detection
- Climate change detection

High level problem



- Problem: Only some states labeled.
  - $A \rightarrow B$ transition unknown, only that it occurs some time before $B \rightarrow C$
- To learn a good discriminative model, need to assign labels to the time series.
  - unsupervised/semi-supervised learning

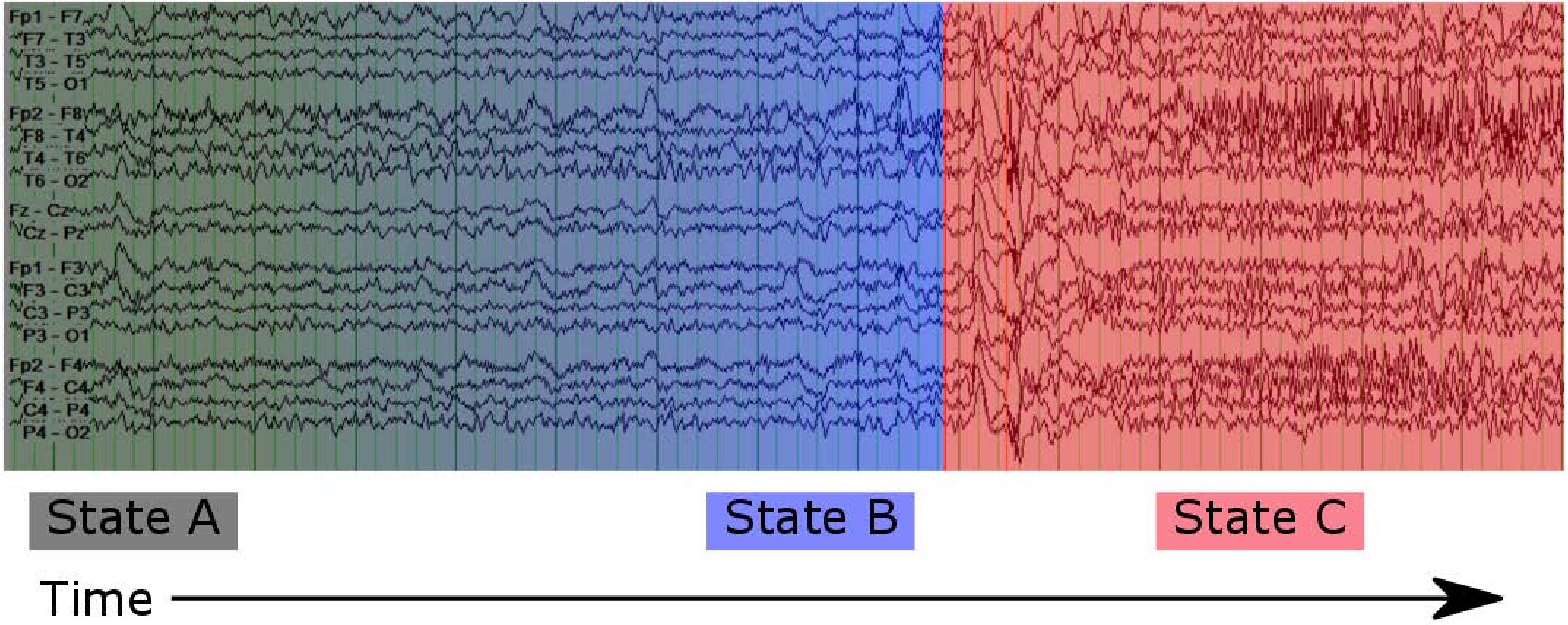


Figure 1. Example of a system generating a multichannel signal transitioning between states. The transition from State B to State C can be easily marked, but the transition from State A to State B cannot be marked. This results in a region of uncertainty about the state of the system.

Change point detection (CPD)

- The system generating a time-series undergoes a transition from one state to another state.
- Change point detection is determining **when** the transition occurs.
- Related to: transition detection/concept drift/covariate shift

Problem definition

- Given a time series X where $\mathrm{X} = \{x_1, x_2, \dots x_T\}$
- Assume X is generated by a process which undergoes a transition from state A to state B,
  - with probability distributions P_A and P_B respectively and $P_A \neq P_B$.
- A time t is the change point if:

$$\{x_1, x_2, \dots x_t\} \sim P_A$$
$$\{x_{t+1}, x_{t+2}, \dots x_T\} \sim P_B$$

Related work

- Hypothesis testing: (Kuncheva, 2013)
  - H_0 - x_t and x_{t-1} drawn from the same multivariate Gaussian distribution
- CUSUM (Jeske et al., 2009)
  - monitor cumulative sum which measures accrued deviations
- Bayesian change-point detection (Adams and MacKay, 2007)
  - Estimate posterior probability of the “run-time” distribution
  - “run-time”: length of time since last change point

Unsupervised

- One class SVM (Mika et al., 1999)
- KLIEP (Sugiyama et al., 2007; Kawahara et al. 2012)
  - approximate density ratio to measure change in distribution
- **Virtual classifiers** (Desobry et al., 2005; Hido et al, 2008, Yamada et al., 2013)
  - **measure likelihood of change point using classification accuracy**

Semi - supervised

Completed work

Virtual classifiers and convolutional networks for seizure prediction

Virtual classifiers (VC) - Theory

- If we consider the change point detection problem as an optimization problem of the form:

$$\max_t D(P_t(x|A), P_t(x|B))$$

- where $D(\cdot,\cdot)$ is a divergence measure between the two distributions.
- Idea is to approximate $D(P_t(x|A), P_t(x|B))$ with classification accuracy

Time series of feature vectors $\{x_k\}_{k=1}^{T}$ with state space $\mathcal{X} = \mathbb{R}^d$.
Time t defines a split of the time series into disjoint sets $A = \{x_1, x_2, \dots, x_t\}$ and $B = \{x_{t+1}, x_{t+2}, \dots, x_T\}$

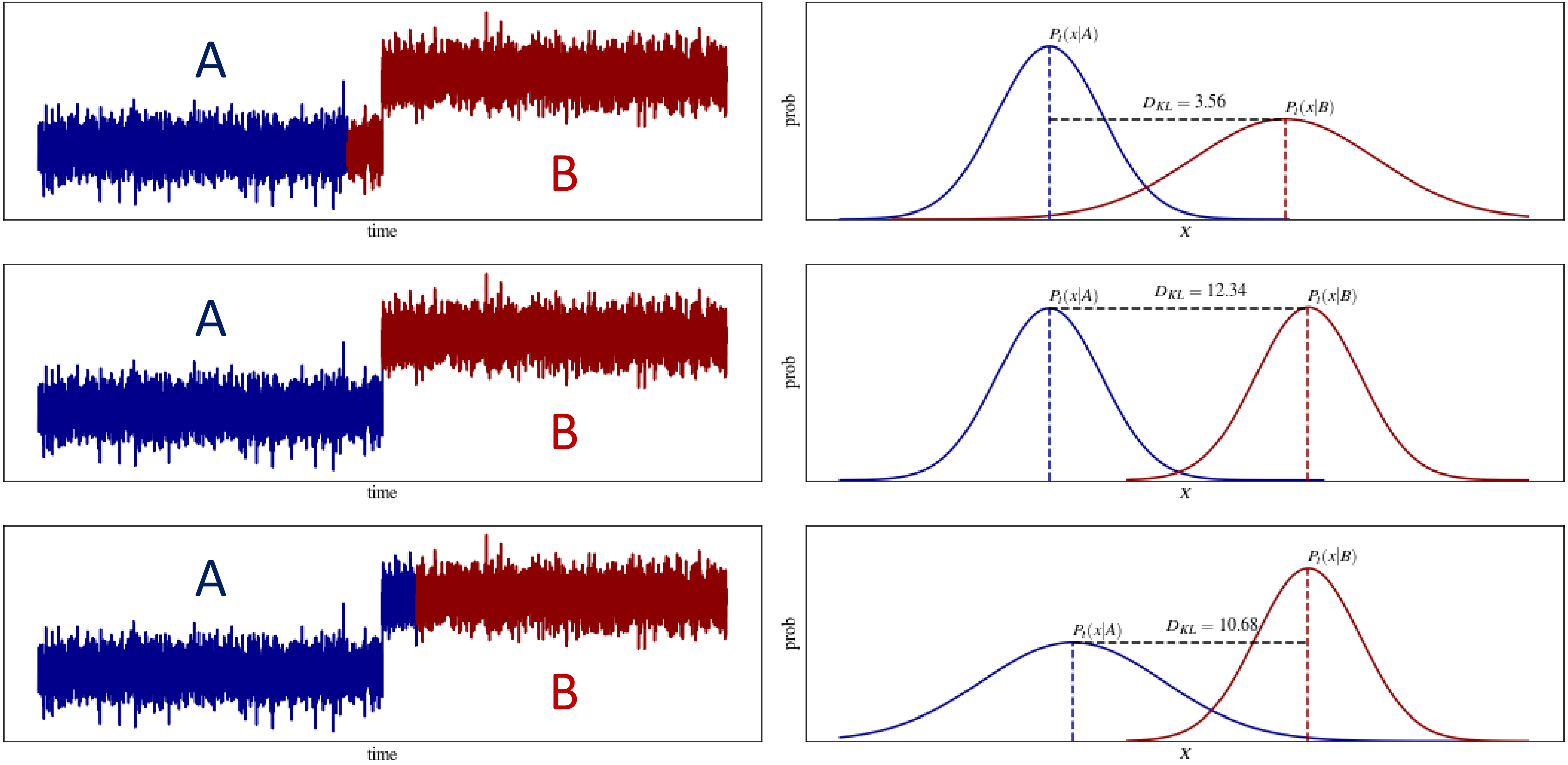


Figure 2. Example of a Gaussian noise signal undergoing a mean shift. By splitting the signal into segments A and B at different time points and approximating the conditional probability distributions with Gaussians, we see the KL-divergence is maximal when the split matches the change point.

Approximating KL-divergence with VC

- Using the KL-divergence for $D(\cdot,\cdot)$ yields:

$$\max_t \sum_{x \in \mathcal{X}} P_t(x|A) \log\left(\frac{P_t(x|A)}{P_t(x|B)}\right)$$

$$\max_t \sum_{x \in \mathcal{X}} P_t(x|A) \log P_t(x|A) - \sum_{x \in \mathcal{X}} P_t(x|A) \log P_t(x|B)$$

- Assuming the entropy of $P_t(x|A)$ is fixed with respect to t:

$$\max_t - \sum_{x \in \mathcal{X}} P_t(x|A) \log P_t(x|B)$$

Bayes rule to isolate posterior distribution

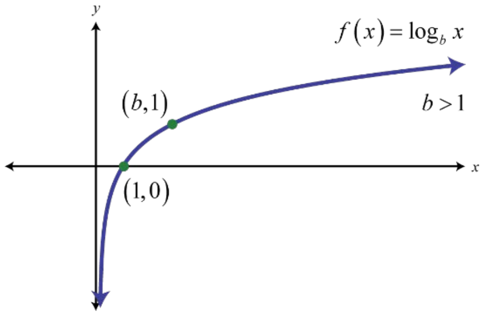
- Applying Bayes rule to $P_t(x|B)$ yields:

$$\max_t - \sum_{x \in \mathcal{X}} P_t(x|A) \log P_t(B|x) - \sum_{x \in \mathcal{X}} P_t(x|A) \log P(x) + \sum_{x \in \mathcal{X}} P_t(x|A) \log P(B)$$

- Simplifying:

$$\max_t - \sum_{x \in \mathcal{X}} P_t(x|A) \log P_t(B|x)$$

- Model posterior $P_t(B|x)$ as a classifier and $P_t(x|A)$ as an assumed set of labels



$$- \sum_{x \in \mathcal{X}} P_t(x|A) \log P_t(B|x) \approx \min - \frac{1}{T} \sum_{i=1}^{T} y_i \log(z_i)$$

Virtual classifiers summary

- Given:
  - a set of candidate change points $\{\tau_1, \tau_2, \dots \tau_m\}$
  - a set of time series $\{X_i\}_{i=1}^{n}$
- Construct a set of binary labels $\left\{Y_j\right\}_{j=1}^{m}$
- Each Y_j is a vector of length T with:

$$Y_{jk} = \begin{cases} -1 \; if \; k \leq \; \tau_j \\ \;\; 1 \; if \; k > \; \tau_j \end{cases} \; for \; k = 1,2, \dots T$$

- Copies of each of these label vectors Y_j are paired with every time series in $\{X_i\}_{i=1}^{n}$ forming the pseudo-labeled dataset $D_j = \left\{\left(X_i, Y_j\right)\right\}_{i=1}^{n}$.
- A classifier is trained on each dataset D_j, resulting in m classifiers each trained on a different labeling of the data.
- Accuracy on a validation set of each of the classifiers is measured as $p_1, p_2, \dots p_m$.

Learn a predictor

1. Determine when the change point occurs in each time series X_i of the dataset $\{X_i\}_{i=1}^{n}$
2. Train a predictor, using the result of step 1, to predict the current state of the system given a sample from a time series
3. On a previously unseen time series X' generated by the same system, predict the change point prospectively.

Evaluating prediction systems

- Sensitivity: percentage of events predicted within prediction horizon
- Specificity: false prediction rate
- Comparison to random predictor (Schelter et al., 2006)

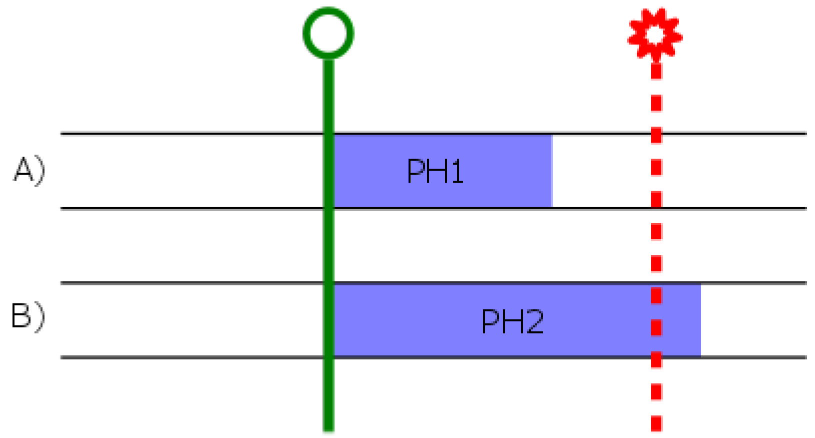


Figure 3. The prediction horizon is a critical parameter for a prediction system as it can be increased arbitrarily to achieve perfect sensitivity.

Is a system better than random?

- Analytical model given by (Schelter et al., 2006)
- Predictions are generated with probability $P \approx \mathrm{FPr} * \mathrm{PH}$
- To perform better than random, sensitivity must be greater than:

$$\sigma > \frac{\max_k \left\{ \left(1 - \left(\sum_{j<k} \binom{K}{j} P^j (1-P)^{K-j} \right)^d \right) > \alpha \right\}}{K}$$

- Where K is the number of analyzed events, d is the dimension of the feature space, and α is a significance level.

Application – Seizure prediction

- Changes occur in the brain prior to seizure onset that make the seizure inevitable.
  - **Seizure prediction horizon (SPH), preictal state/period**
- Central question: When do the pre-seizure changes occur?

Literature review – seizure prediction

- Seizure prediction horizon (SPH)
  - Previous studies assume SPH in the range 2 minutes to 262.5 minutes (Mormann et al., 2016)
  - SPH reported varies based on features extracted
- Features:
  - Time/frequency domain features (Karoly et al., 2016)
  - Multivariate features (Cho et al., 2017; Dhulekar et al., 2016)
  - Model based features (Arabi and He, 2014)

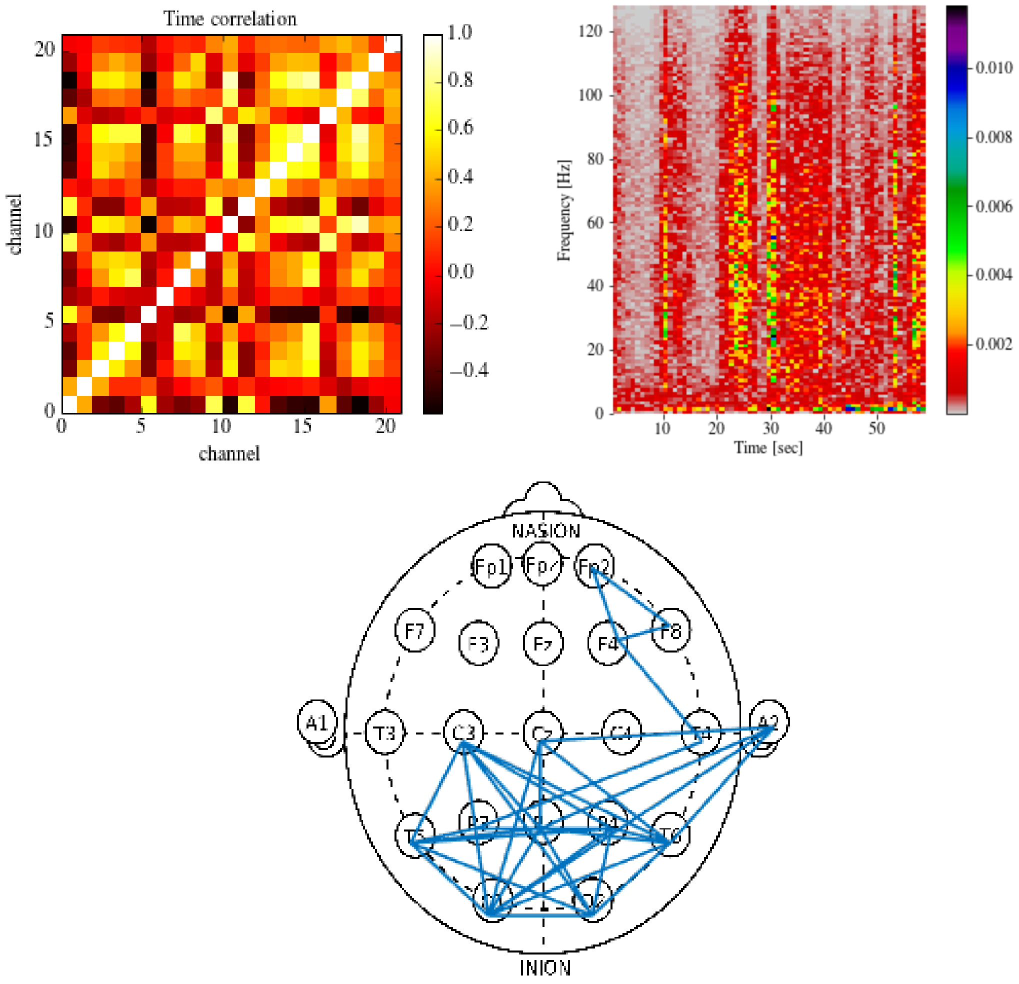


Figure 4. Examples of features extracted for seizure prediction.

Literature Review – state of the art

- “Crowdsourcing reproducible seizure forecasting in human and canine epilepsy” (Brinkman et al., 2016)
  - Results of Kaggle competition on seizure prediction
  - Winning submissions used time/frequency domain features extracted from intracranial human and dog EEG
  - SPH – 60 mins
- “Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant epilepsy: A first-in-man study” (Cook et al., 2013)
  - Implanted seizure prediction device
  - Three energy measures in filtered intracranial EEG as features
  - SPH – 6 – 30 minutes (optimized per patient)
- “On the proper selection of preictal period for seizure prediction” (Bandarabadi et al., 2015)
  - Measure common area (C) between preictal and interictal feature histograms
  - Define optimal preictal period for a single feature as minimum C

Learning the preictal period

- Our Contributions
  - Use Change Point Detection (CPD) to determine preictal period
  - Combine CPD with automatic feature extraction

CNN for feature extraction

- Use CNN (LeCun et al., 1998) as virtual classifiers to detect change point and learn features from EEG
- Convolutions over time and frequency domain via wavelets

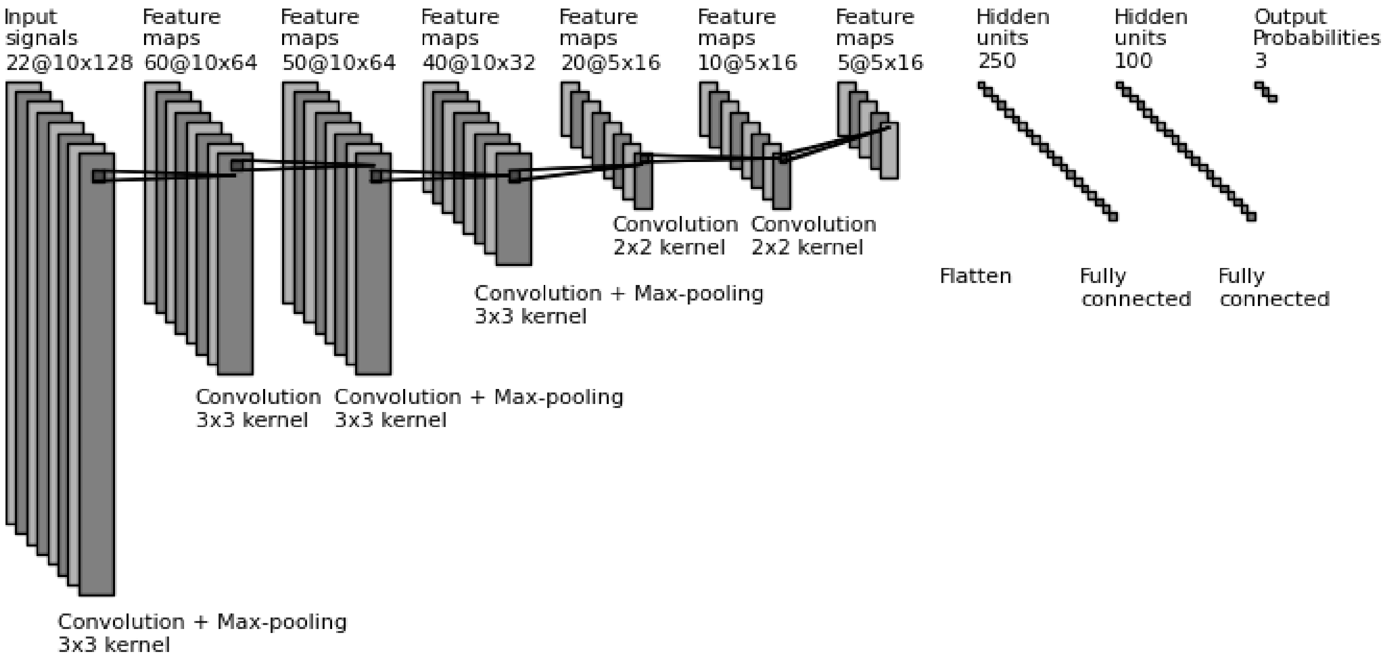
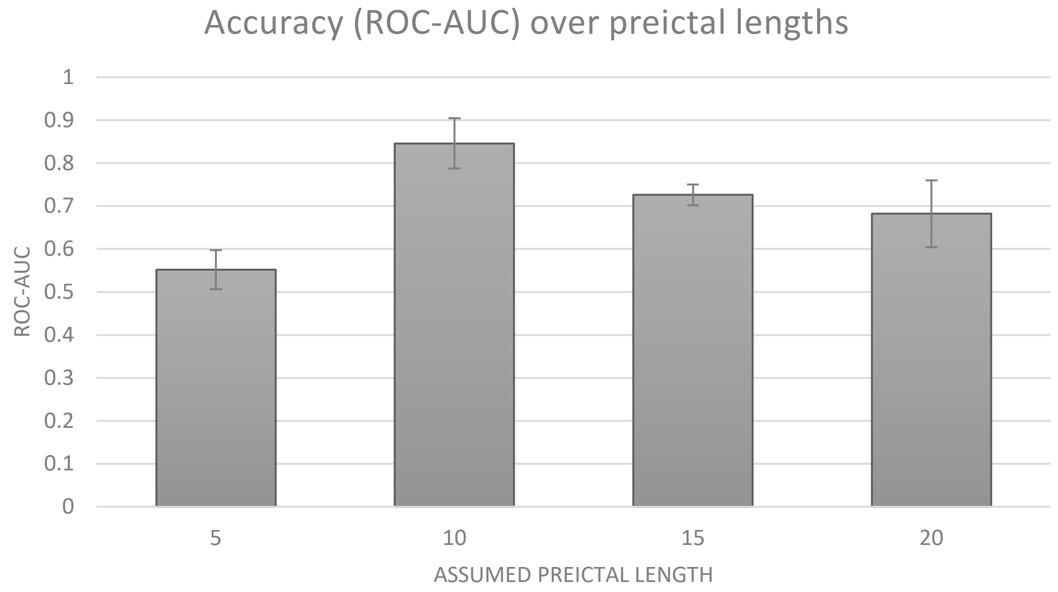


Figure 5. Convolutional neural network trained on EEG to predict brain states from wavelet transformed EEG.

VC for preictal period length

- Candidate preictal lengths were 5, 10, 15, and 20 mins
- A CNN was trained for each labelling of the data
- Preictal length of 10 mins was chosen based on significant improvement in accuracy

Table 1. ROC-AUC between interictal and preictal classes for different assumed preictal lengths. Averaged over 10-folds of validation data, error bar shows 1 standard deviation.



Results and Comparison

- We compared our results to:
  - 2 top performing algorithms from Kaggle (Brinkmann et al., 2016)
  - Cook group's algorithm (Cook et al., 2013)

Table 2. Seizure prediction results

Method	SPH (mins)	Sensitivity	FPr (FP/h)	Random pred. $\sigma_{low} - \sigma_{high}$
Kaggle1	60	72.7%	0.285	15.1% - 27.2%
Kaggle 2	60	75.8%	0.230	12.1% - 24.2%
Cook et al.	PS*	66.7%	0.186	12.1% - 21.2%
This work	10	**87.8%**	**0.142**	9.1% - 15.1%

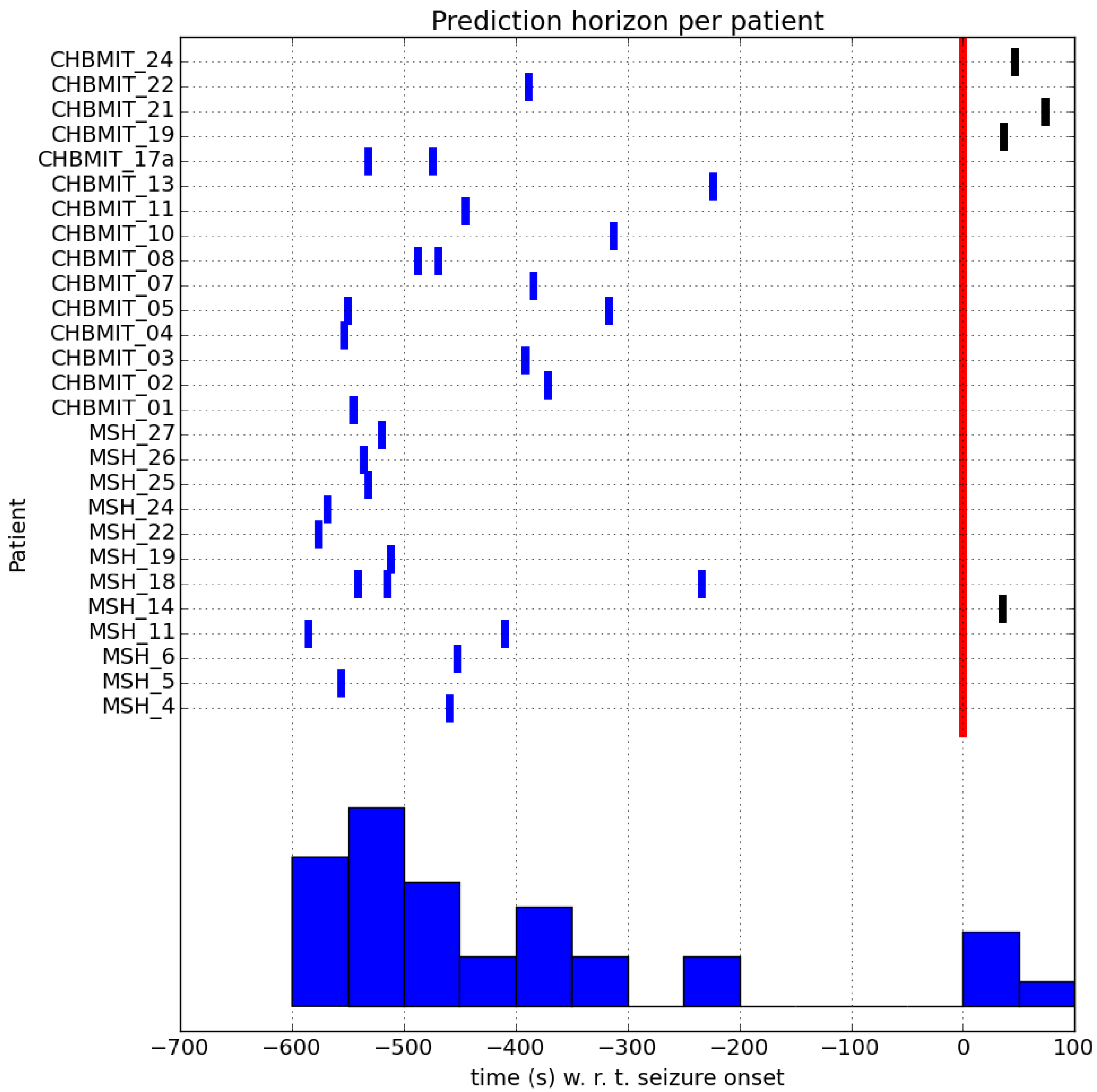


Figure 6. Prediction times generated by the CNN for all test set recordings with seizures grouped by patient. The spread of the prediction times is large indicating a non-uniform transition time within patients and between patients.

Limitations

- VC requires a uniform transition time over all time series
  - Otherwise combinatorial explosion of m^n occurs
- VC is also very computationally expensive
  - Requires training m neural nets multiple times.
- Want a method that allows variable transition time between time series, while preserving benefits of VC.
  - CPD with Density Ratio estimation and DNN's

Proposed Work

Estimating density ratio with deep neural networks

Density ratio/importance

- Density ratio or importance arises in many contexts
  - Monte Carlo importance sampling
  - Covariate shift detection
- Can use density ratio for CPD when:

$$\beta_t = \frac{P(x_t|B)}{P(x_t|A)}$$

- β_t measures likelihood x_t comes from the distribution of state B vs. the distribution of state A
- Can view as a weight for each sample

Density ratio estimation

- Since density estimation is difficult, this is avoided by attempting to estimate the density ratio directly.

$$f(x_t;\theta) \approx \frac{P(x_t|B)}{P(x_t|A)}$$

  - Note that $P(x_t|A)f(x_t;\theta)$ should equal $P(x_t|B)$

- Parameterize density ratio approximator with a set of kernels and minimize KL-divergence to the true density ratio. (Sugiyama et al., 2007)

$$\min_\theta \sum_x P(x|B)\log\frac{P(x|B)}{P(x|A)\sum_{l=1}^{b}\theta_l K(x,x_l)}$$

$$\min_\theta \sum_x P(x|B)\log P(x|B) - \sum_x P(x|B)\log\sum_{l=1}^{b}\theta_l K(x,x_l)$$

$$\min_\theta -\sum_x P(x|B)\log\sum_{l=1}^{b}\theta_l K(x,x_l)$$

Empirical estimates and constraints

- Split data into two sets:
  - Reference: known to be in state A
  - Test: unknown state

$$\min_{\theta} -\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \log \sum_{l=1}^{b} \theta_l K(x_i, x_l)$$

$$\sum_{l=1}^{b} \theta_l K\left(x_j, x_l\right) > 0 \; \forall \, j = 1 \dots n_{ref}$$

$$\frac{1}{n_{ref}} \sum_{j=1}^{n_{ref}} \sum_{l=1}^{b} \theta_l K\left(x_j, x_l\right) = 1$$

Neural network estimator

- We propose using a neural network estimator to the density ratio instead to learn features.
- Challenges:
  - Original optimization problem becomes non-convex
  - Equality constraints difficult to satisfy: $\frac{1}{n_{ref}} \sum_{j=1}^{n_{ref}} f(x_j; \theta) = 1$
- Two proposed solutions:
  - Use Lagrange multipliers/Barrier method
  - Use double sided KL-divergence

Density ratio estimation with DNN

- We need formulations that can be optimized with neural networks (SGD)
- Lagrange multipliers/Barrier method:

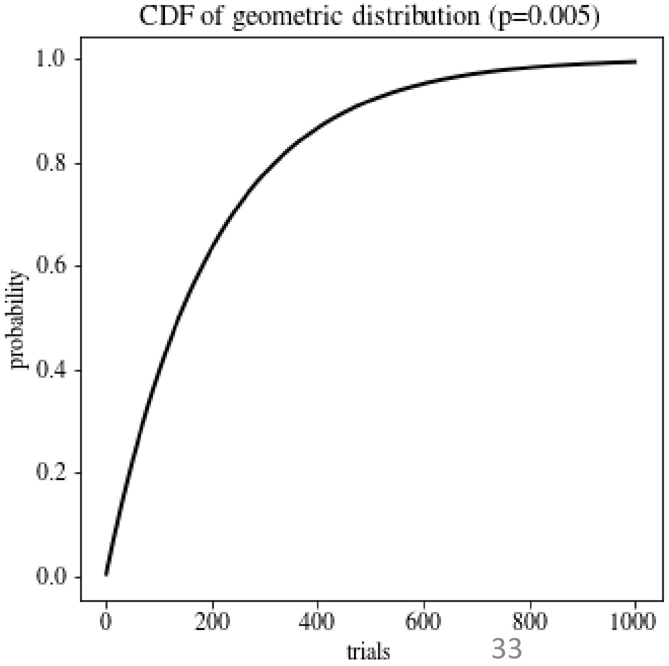
$$\min_{\theta} -\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \log f(x_i;\theta) + \lambda \left(\left| \frac{1}{n_{ref}} \sum_{j=1}^{n_{ref}} f(x_j;\theta) - 1 \right| \right)$$
$$f(x_j;\theta) > 0 \; \forall \, j = 1 \ldots n_{ref}$$

- Double sided KL-divergence: (Note that $P(x_t|A)$ should equal $P(x_t|B)/f(x_t;\theta)$)

$$\min_{\theta} -\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \log f(x_i;\theta) + \frac{1}{n_{ref}} \sum_{j=1}^{n_{ref}} \log f(x_j;\theta)$$
$$f(x_j;\theta) > 0 \; \forall \, j = 1 \ldots n_{ref}$$

Incorporating temporal information

- An additional challenge with density ratio estimation for change point detection is that the ratio does not include temporal information
- We can add temporal information by weighting test samples by temporal distance from the known transition point.



Potential problems and remedies

- Lagrange multipliers only guarantees that the optimal feasible solution is a stationary point
- Batch vs Minibatch gradient descent

Contributions

- Completed
  - Incorporate automatic feature extraction into change point detection (VC+DNN)
  - New automatic feature extraction methods for time series (Wavelet Deconvolutions)
  - Application to epileptic seizure prediction
- Proposed
  - Approximate density ratio with DNN
  - Incorporate temporal information in density ratio estimation for CPD

Tasks and Timeline

1. Establish baseline results for density ratio estimation (March-April)
2. Compare each proposed method to baseline results (April-May)
3. Adjust methods based on results (May-June)
4. Apply proposed methods to data from previous seizure prediction study and new data collected at MSH (August-October)
5. Analyze results using clinical factors (November-December)
6. Compare performance of system using proposed methods to previous work (January-February)

Acknowledgements

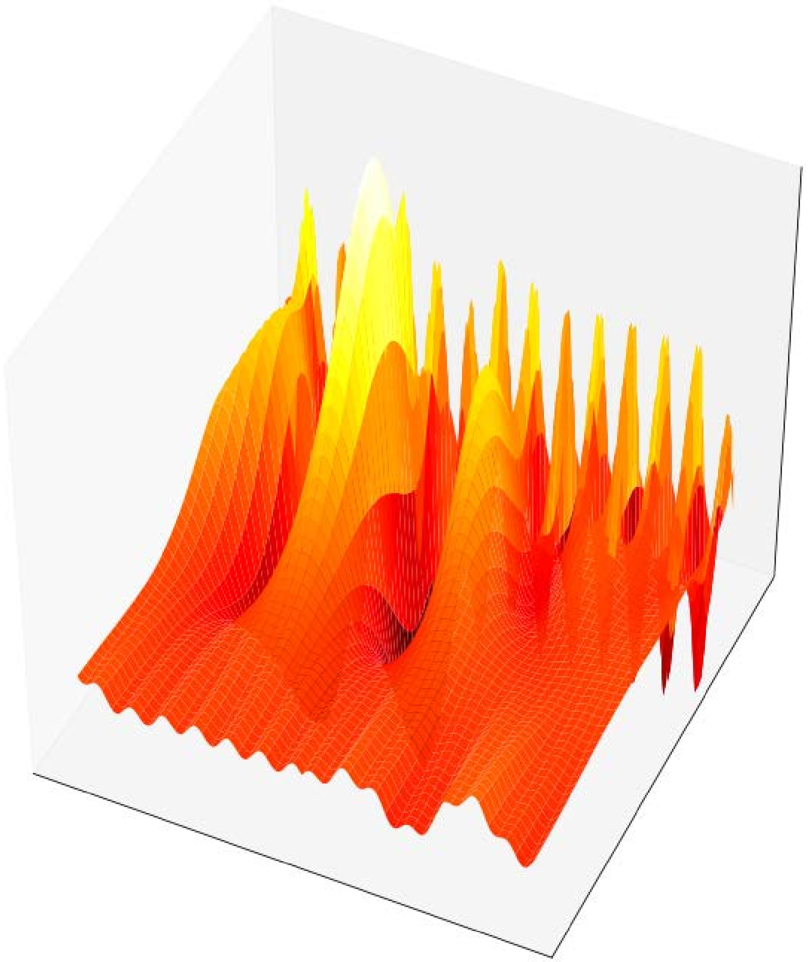
- Advisor:
  - Prof. Bülent Yener
- Committee:
  - Prof. Malik Magdon-Ismail
  - Dr. Lara Marcuse
  - Prof. Mohammed Zaki

BACKUP SLIDES

Wavelet Deconvolutions

Spectral decompositions

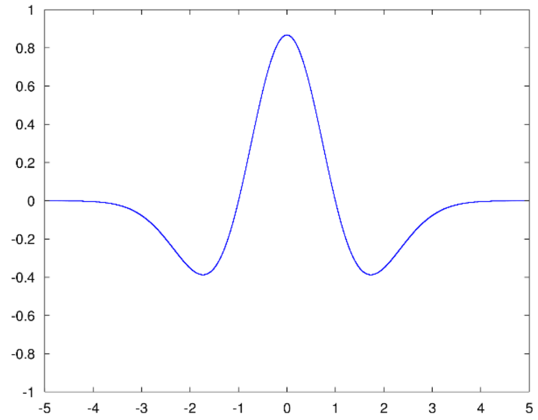
- Many models for time series use a spectral decomposition of the signals as input
  - Many parameters to pick
- Typically use cross validation to pick parameters
  - Can be time consuming and data hungry
- Used in applications such as automatic speech recognition (Hinton et al., 2012), biological signal analysis (Andreao et al., 2006), and financial time series (Cao et al., 2003)



Wavelet transform

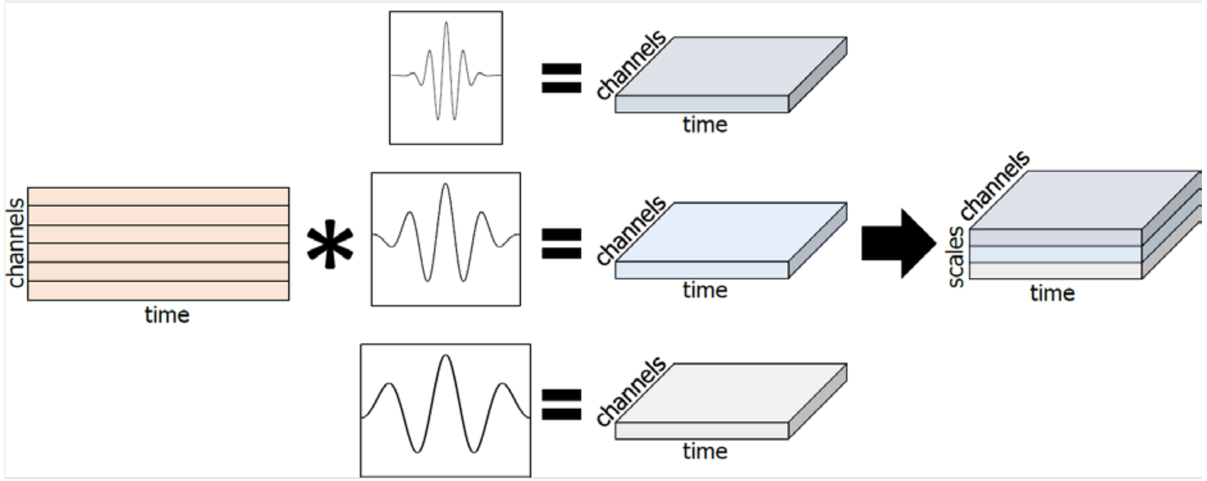
- The wavelet transform reveals spatial and spectral information (Daubechies, 1990)
- Scale the mother wavelet and convolve with the signal
- Reveals frequency content of the signal at that scale and each time point.

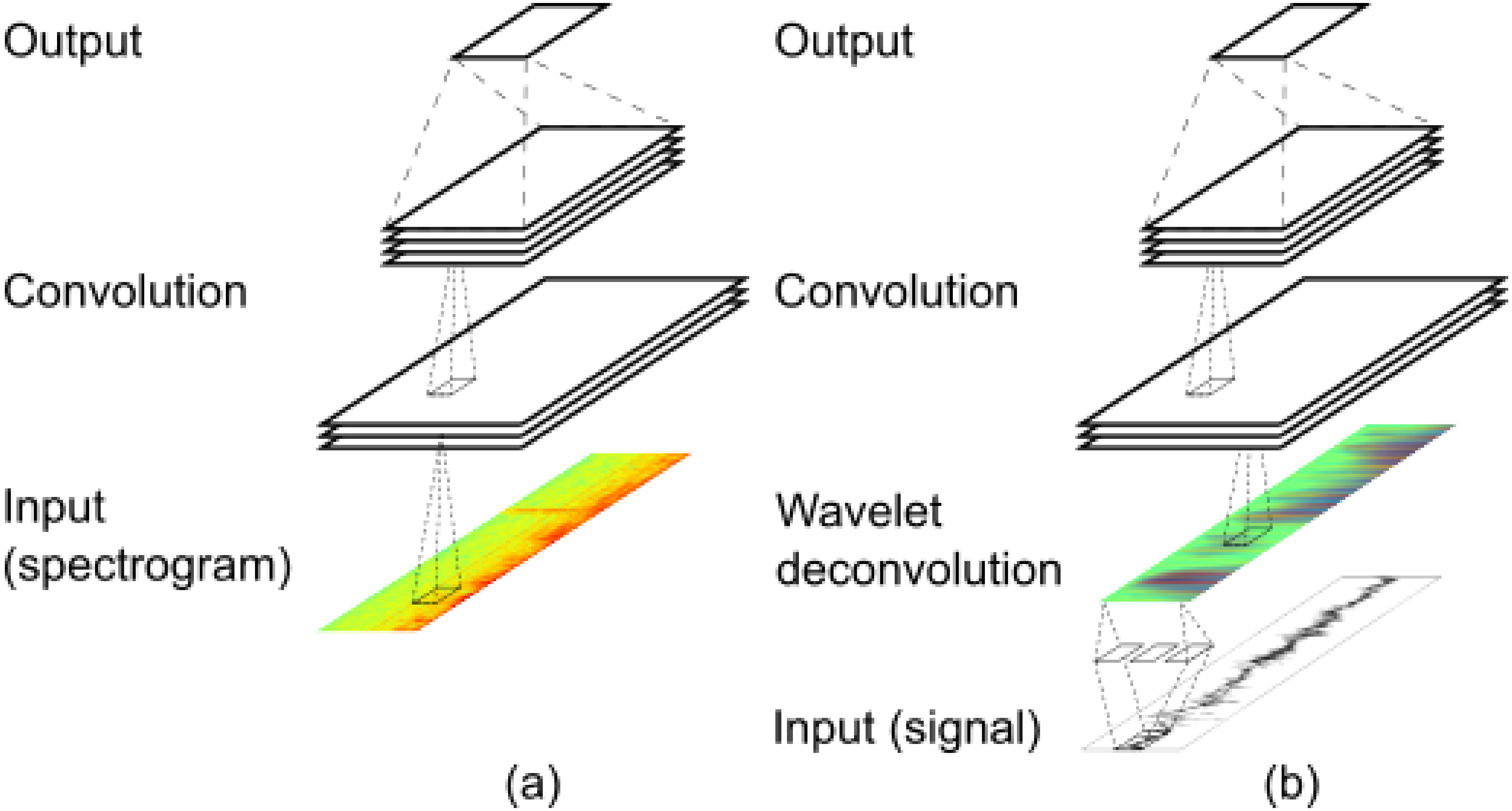
$$\Psi_w(t) = \frac{2}{\sqrt{3w\pi^{\frac{1}{4}}}}\left(1 - \frac{t^2}{w^2}\right)e^{-\frac{t^2}{2w^2}}$$



Automatically extracting time/freq domain features

- Combine the Wavelet Transform and CNN
- Use backpropagation to learn the scale parameters
- This enables learning the “width” of the kernel with gradient descent
  - CNN’s have fixed kernel sizes
- Also a reduction in the number of parameters





Output
Convolution
Input
(spectrogram)
(a)
Output
Convolution
Wavelet
deconvolution
Input (signal)
(b)

Learn scales with backpropagation

- Wavelet transform with learnable scales

The output of the wavelet layer is given by:

$$y_{ij} = \sum_{a=1}^{P} s_{ia}\, x_{j+a} \quad \forall i = 1 \dots M$$

Where the wavelet filter $s_i \in \mathbb{R}^{1xP}$ is the discretized wavelet function over the grid $k = \left\{-\frac{P-1}{2} \dots \frac{P-1}{2}\right\}$:

$$s_{ia} = \frac{2}{\sqrt{3w_i}\pi^{\frac{1}{4}}}\left(1 - \frac{k_a^2}{w_i^2}\right) e^{-\frac{k_a^2}{2w_i^2}} \quad \forall a = 1 \dots P$$

For backpropagation, we want $\frac{\delta E}{\delta w_i}$ where E is some error function:

$$\frac{\delta E}{\delta w_i} = \sum_{a=1}^{P} \frac{\delta E}{\delta s_{ia}} \frac{\delta s_{ia}}{\delta w_i} = \sum_{a=1}^{P} \frac{\delta E}{\delta s_{ia}} \left[A\left(M\frac{\delta G}{\delta w_i} + G\frac{\delta M}{\delta w_i}\right) + MG\frac{\delta A}{\delta w_i}\right]$$

$$\frac{\delta E}{\delta s_{ia}} = \sum_{j=1}^{N} \frac{\delta E}{\delta y_{ij}} \frac{\delta y_{ij}}{\delta s_{ia}} = \sum_{j=1}^{N} \frac{\delta E}{\delta y_{ij}} x_{j+a}$$

$$A = \frac{2}{\pi^{\frac{1}{4}}} \boxed{(3w_i)^{-\frac{1}{3}}}$$

$$M = 1 - \frac{k_a^2}{w_i^2}$$

$$G = e^{-\frac{k_a^2}{2w_i^2}}$$

$$\frac{\delta A}{\delta w_i} = -\frac{6}{\pi^{\frac{1}{4}}} \boxed{(3w_i)^{-\frac{3}{2}}}$$

$$\frac{\delta M}{\delta w_i} = \frac{2k_a^2}{w_i^3}$$

$$\frac{\delta G}{\delta w_i} = \frac{k_a^2}{w_i^3} e^{-\frac{k_a^2}{w_i^2}}$$

$w_i > 0$

Results

- TIMIT Phone recognition dataset
- UCR Haptics dataset

Table 3. Best reported PER on the Timit dataset without context dependence

Method	PER (Phone Error Rate)
DNN with ReLU units [96]	20.8
DNN + RNN [110]	18.8
CNN [97]	18.9
WD + CNN (this work)	18.1
LSTM RNN [111]	17.7
Hierarchical CNN [97]	**16.5**

Table 4. Test error on the Haptics dataset

Method	Test Error
DTW [113]	0.623
BOSS [114]	0.536
ResNet [105]	0.495
COTE [115]	0.488
FCN [105]	0.449
WD + CNN (this work)	**0.425**