Identifying Brain Activity from EEG Recordings

CS229: Machine Learning

Ray Iyer - `rri@stanford.edu`, Trenton Chang - `tchang97@stanford.edu`, Caroline Ho - `cho19@stanford.edu`
Department of Computer Science, Stanford University

Prediction Task

- **Motivation:** classify EEG recordings as different types of brain activity
- **Application:** automate detection of disorders such as epilepsy in healthcare setting

Data

Epileptic Seizure Recognition Dataset [1]

- Size: 11500 rows (500 people x 23 recording segments per person), 178 columns (1/178 s)
- Input: 1-second-long EEG recording
- Label: one of 5 classes of brain activity
  1. Epileptic seizure (epileptic subject)
  2. From tumor (epileptic subject)
  3. From healthy area (epileptic subject)
  4. Eyes closed (non-epileptic subject)
  5. Eyes open (non-epileptic subject)

Feature Extraction

Preprocessing

- Normalize data

Feature Sets

- **Summary statistics**: min and std over recording
- **Spectral entropy**: treats normalized power distribution in the frequency domain as a probability distribution, and calculates the Shannon entropy

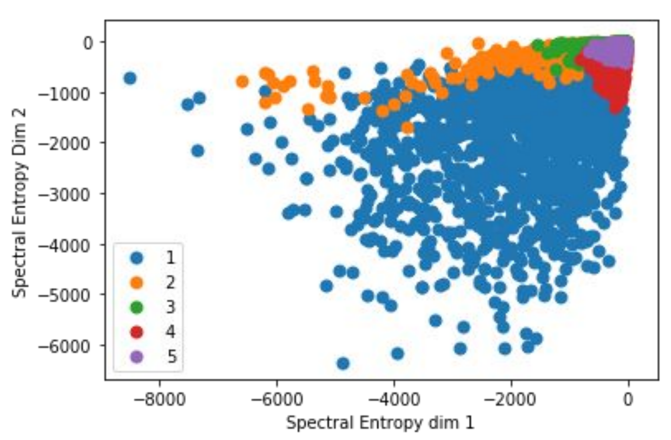

Fig. 1: Spectral Entropy separation in 2 dimensions

- **Raw data**: raw numeric EEG data
- **Fourier transform**: transforms raw EEG data from time to frequency domain

References

[1] Andrzejak RG, Lehnertz K, Rieke C, Mormann F, David P, Elger CE (2001). Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E, 64, 061907

Models

Softmax Regression

- Hypothesis:

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^{K} \exp(\theta_j^T x)}$$

- Loss: multiclass cross entropy

$$\mathcal{L}(\hat{y}, y) = -\sum_{j=1}^{k} \mathbf{1}\{y = j\} \log \hat{y}_j$$

Evaluation

- Nested CV: 10 inner + outer folds
- Hyperparameters: C = 0.1, 0.5, 1, 5

K-Nearest Neighbors

- Predict class based on *k* closest EEGs
- Euclidean distance: $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

Evaluation

- Nested CV: 10 inner + outer folds
- Hyperparameters: k = 3, 5, 7, 9

Hidden Markov Model (HMM)

- Use Baum-Welch algorithm (similar to EM) to generate one HMM per class
- Assumption: EEG data is a noisy stream modelable by transitions between n Gaussians
- Markov assumption: $P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-1})$
- Emission probabilities: $P(Y_t \mid X_{0:t}, Y_{1:t-1}) = P(Y_t \mid X_t)$
- Note: these are **very** strong independence assumptions!
- Note: not guaranteed to converge globally

Evaluation

- Test time: which HMM yields the highest probability path through states?

$$\hat{y} = \underset{j}{argmax} \sum_{i=0} P(Y_t \mid X_t^j) P(X^j)$$

- 70/30 train-test split
- Trials with n=[1, 2, … 8] latent variables over 178 timesteps

Convolutional Neural Network (CNN)

- Shift invariance good for picking up sequence patterns
- Loss: multiclass cross entropy
- Trained for 25 epochs with batch size 32

Architecture

- 2 1D convolutional layers w/64 filters (ReLU)
- 1D max pooling layer
- 2 1D convolutional layers w/128 filters (ReLU)
- 1D global average pooling layer
- Dropout layer
- Dense layer (softmax)

Evaluation

- 10-fold CV, 80/20 split (nested CV too costly)
- Hyperparameters
  - Dropout = 0.1, 0.2, 0.5
  - Window = 3, 7, 11 (Fourier) or 3, 11, 15, 19 (raw)

Experimental Results

Features	Model	Train Acc	Test Acc	Test Macro F1	Test Seizure F1
Summary stats	Softmax reg	0.465	0.434	0.399	0.873
	kNN	0.591	0.472	0.466	0.902
Spectral entropy	**kNN**	**0.768**	**0.681**	**0.653**	**0.919**
Raw data	Softmax reg	0.303	0.195	0.186	0.323
	kNN	0.655	0.423	0.421	0.799
	HMM	0.432	0.434	0.420	0.873
	CNN	**0.927**	**0.780**	**0.773**	**0.974**
Fourier transform	Softmax reg	0.284	0.187	0.177	0.295
	kNN	0.656	0.443	0.448	0.805
	CNN	**0.959**	**0.740**	**0.737**	**0.941**

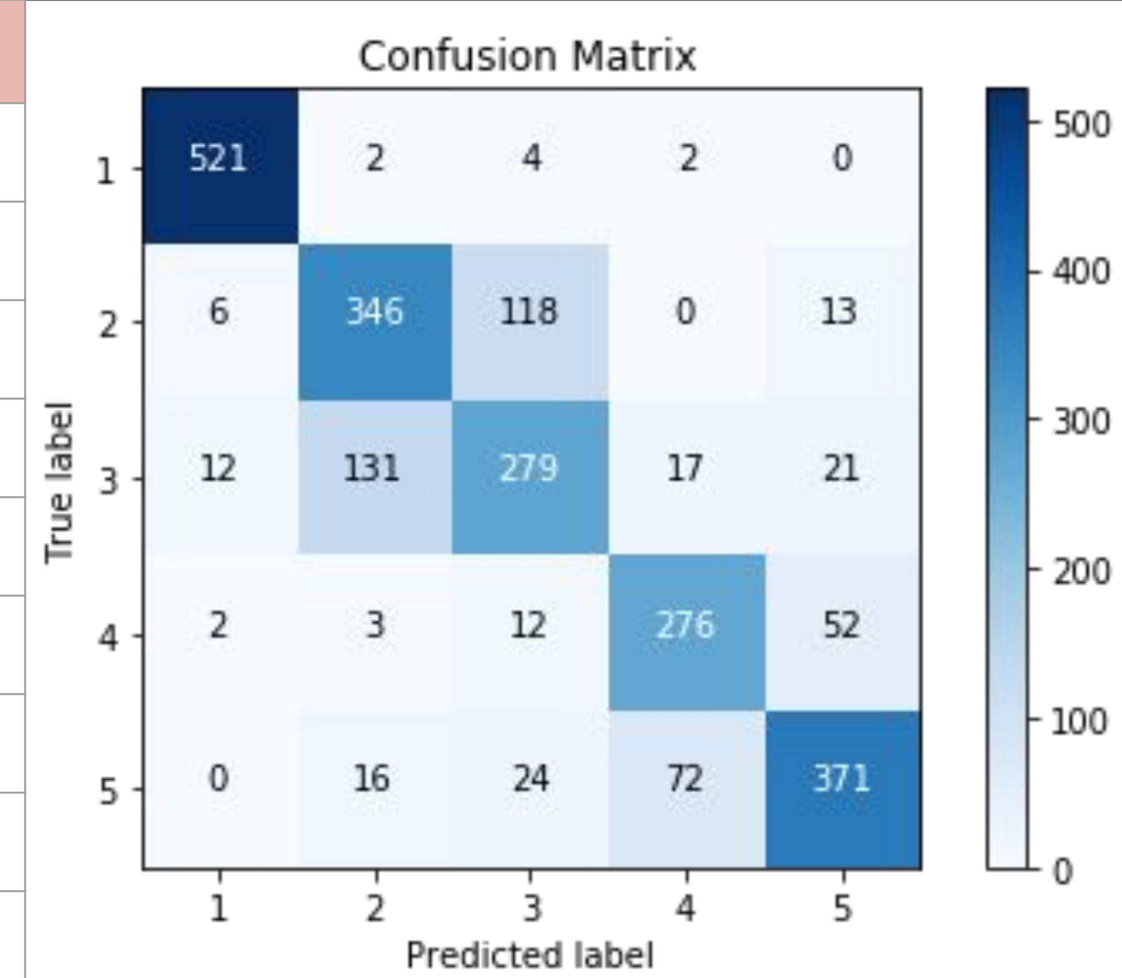

Fig. 2: Raw CNN confusion matrix

Discussion

- CNN performs well on raw sequence data, though model overfits
  - Likely picks up on recurring patterns in sequence
  - Most errors within super-class of epileptic or non-epileptic subject
- Spectral entropy kNN performs fairly well with far fewer features
  - Achieves good separability of data

Future Directions

- Try different CNN hyperparameters and architectures to reduce overfitting, such as regularization techniques like weight decay and dropout
- Experiment with different time-distributed architectures (RNNs)
- Refine signal processing techniques with domain knowledge to improve separability in feature space