Machine Learning for Time Pulse Analysis

Feature Extraction (sig/bkg model)

Ensemble Learning-Based Pulse Signal Recognition: Classification Model Development Study

- uses an SVM for classification and a CNN to "extract local features of unstructured data [pulses]"
- time domain feature extraction
    - ratio between different amplitudes was added as a feature "in order to better reflect the waveform characteristics ... because of the large differences between different pulse waveforms"

Bayesian Spectrum Estimation of Harmonic Signals

- spectrum estimation: estimation of average power of signals as a function of frequency
- derived Bayesian spectrum estimator of multiple sinusoids in Gaussian noise
- shows spectrum estimates of sinusoidal signals whose frequencies are separated by a much smaller value than Rayleigh resolution
    - could search for inverse: assume signal -> noise and vice versa
    - in this case, could decompose noise via FFT for superimposed sinusoids
        - anything else (ie signal) is the "noise" in this framework
    - able to incorporate domain specific knowledge about the signal/noise parameters in prior
- in this paper, they assume the only unknown parameter is the variance on the Gaussian noise (coefficients and frequencies of sinusoidal signal are known)
- it might be difficult to use a Bayesian analysis like this for the background model since white noise has a constant power spectral density
    - uniformly spread across all frequencies
    - Gaussian white noise has flat spectrum

Evidence for observation of virtual radio Cherenkov fields

Alice Bean, John P. Ralston, James Snow

- implements "noise filters" that matches to the patterns/correlations in the noise
- I'm pretty sure I did an analysis similar to the procedure described in 2.2.1 for Ralston's class
- procedure defines how to filter noise from the data by analyzing the data in the "pattern subspace"
- procedure from paper:
    1. construct density matrix ρ_{ij} of patterns of data, p_i, by taking the outer product of these vectors (which means these pattern vectors will be eigenvectors of ρ_{ij})

2. classify pattern importance quantitatively
3. expand data in pattern basis
4. throw away components not desired (ie small/rare elements)
5. revert expansion back to original basis

Notes

- using matched wavelets (or honestly even a Fourier transform) to transform the signal for time resolution extraction
    - then can work in frequency space
    - give neural network frequency information instead of time information?
- use recurrent NN
    - sort of same idea behind CNN
        - CNN gets at relational information between data points (convolves)
        - sort of similarly, RNNs have a "memory" that can make connections between data points it's processing and points it's already seen
        - this might be better for time series data since the relationships between data points are temporal (not static)
- white noise can be modeled as a moving average process in time series analysis
    - current value of dependent vairable depends on current + past values of a sequential white noise process
    - could use this model, or model of it's power spectrum,
      $S(\omega) = N_0/2$ for some constant N_0 over all frequencies ω
      to essentially create a background hypothesis over some set of parameters so we could hypothesis test
- maybe we could model the background as a random walk?
    - or like semi-random walk?
- random walk
    - steps are defined by iid random variables

Regression to time resolution

Notes

- wavelet regression to find σ_t (or even just $\Delta t = t - t_{pred}$)
- various, scattered probability distribution thoughts