Estimation of Channel Distribution Functions using a Neural Network

Peter Hartig

April 27, 2020

Outline

The Channel State

- Observations are made of some channel in a point-to-point communication system.
- For each observation, this channel takes on a state $s[k] \in \mathcal{S}$.
- The true state $s[k]$ is hidden by the addition of noise to an observation $y[k]$.

Sampling Channel State

Over many observations, the sequence $\mathbf{y}$ corresponds to a sequence of channel states $\mathbf{s} \in S^N$

Sampling Channel State

Over many observations, the sequence $\mathbf{y}$ corresponds to a sequence of channel states $\mathbf{s} \in S^N$

For a channel represented by an LTI system, the state is determined entirely by the transmitted information $\mathbf{x}$.

Estimating the True Channel State

Goal:

We attempt to estimate the true, hidden, sequence of channel states, $\mathbf{s}$, based the sequence of samples $\mathbf{y}$.

Note

Assume we known the number of channel states $|S|$

Outline

MAP Sequence Detection

$$\underset{\mathbf{s}\in S^N}{\text{maximize}}\ p(\mathbf{s}|\mathbf{y}).$$

MAP Sequence Detection

$$\underset{\mathbf{s}\in S^N}{\text{maximize}}\ p(\mathbf{s}|\mathbf{y}).$$

Using Bayes' theorem

$$p(\mathbf{s}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{s})p(\mathbf{s})}{p(\mathbf{y})}$$

Noting that $p(\mathbf{y})$ can be ignored

MAP Sequence Detection

$$\underset{\mathbf{s}\in S^N}{\text{maximize}}\ p(\mathbf{s}|\mathbf{y}).$$

Using Bayes' theorem

$$p(\mathbf{s}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{s})p(\mathbf{s})}{p(\mathbf{y})}$$

Noting that $p(\mathbf{y})$ can be ignored

$$\underset{\mathbf{s}\in S^N}{\text{maximize}}\ p(\mathbf{y}|\mathbf{s})p(\mathbf{s}) \tag{1}$$

MAP Sequence Detection

$$\underset{\mathbf{s} \in S^N}{\text{maximize}}\; p(\mathbf{y}|\mathbf{s})p(\mathbf{s})$$

MAP Sequence Detection

$$\underset{\mathbf{s}\in S^N}{\text{maximize}}\ p(\mathbf{y}|\mathbf{s})p(\mathbf{s})$$

Assuming

$$p(\mathbf{y}|\mathbf{s}) = \prod_{k=0}^{N-1} p(y[k]|\mathbf{s})$$

MAP Sequence Detection

$$\underset{\mathbf{s} \in S^N}{\text{maximize}} \; p(\mathbf{y}|\mathbf{s})p(\mathbf{s})$$

Assuming

$$p(\mathbf{y}|\mathbf{s}) = \prod_{k=0}^{N-1} p(y[k]|\mathbf{s})$$

and

$$p(y[k]|\mathbf{s}) = p(y[k]|s[k])$$

MAP Sequence Detection

$$\underset{\mathbf{s}\in S^N}{\text{maximize}}\; p(\mathbf{y}|\mathbf{s})p(\mathbf{s})$$

Assuming

$$p(\mathbf{y}|\mathbf{s}) = \prod_{k=0}^{N-1} p(y[k]|\mathbf{s})$$

and

$$p(y[k]|\mathbf{s}) = p(y[k]|s[k])$$

this simplifies to

$$\underset{\mathbf{s}\in S^N}{\text{maximize}}\; \prod_{k=0}^{N-1} p(y[k]|s[k])p(\mathbf{s}).$$

MAP Sequence Detection

For the LTI channel

$$p(\mathbf{s})$$
$$=$$
$$p(s[N]|s[N-1] \ldots s[0])p(s[N-1]|s[N-2] \ldots s[0]) \ldots p(s[1]|s[0])p(s[0])$$

describes the consistency of transmitted symbols implied by the state sequence. The channel states of the LTI channel satisfy the Markov property

$$p(s[N]|s[N-1] \ldots s[0]) = p(s[N]|s[N-1]]).$$

Example with LTI channel - Continued

With these assumptions,

$$\underset{\mathbf{s}\in S^N}{\text{maximize}} \prod_{k=0}^{N-1} p(y[k]|s[k])p(\mathbf{s})$$

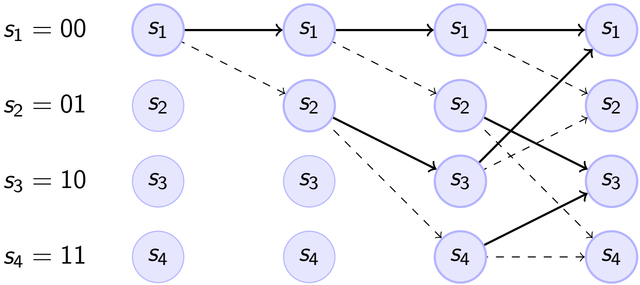
is equivalent to

$$\underset{\mathbf{s}\in S^N}{\text{minimize}} \sum_{k=0}^{N-1} -log(p(y[k]|s[k])p(s[k]|s[k-1])).$$

For the LTI channel $p(s[k]|s[k-1])$ is 0 if states contradict transmission sequence, otherwise this term is constant.

Viterbi Algorithm

$$\underset{\mathbf{s}\in S^N}{\text{minimize}} \sum_{k=0}^{N-1} -log(p(y[k]|s[k])p(s[k]|s[k-1])).$$



Example with channel impulse response length 2 and constellation size 2.

Outline

Decomposing Terms in the Viterbi Algorithm

The individual terms in

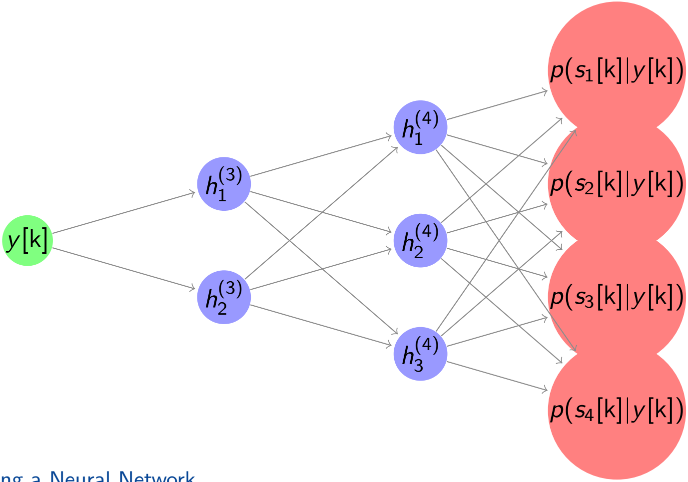
$$\underset{\mathbf{s} \in S^N}{\text{minimize}} \sum_{k=0}^{N-1} -log(p(y[k]|s[k])p(s[k]|s[k-1])).$$

can be rewritten

$$p(y[k]|s[k])p(s[k]|s[k-1]) = \frac{p(s[k]|y[k])p(y[k])}{p(s[k])} p(s[k]|s[k-1]).$$

Decomposing Terms in the Viterbi Algorithm

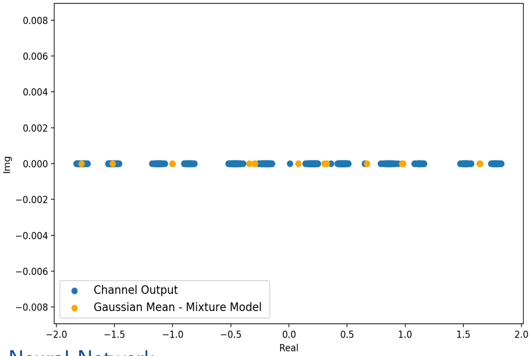
$$\frac{p(s[k]|y[k])p(y[k])}{p(s[k])}$$



Decomposing Terms in the Viterbi Algorithm

$$\frac{p(s[k]|y[k])\textcolor{red}{p(y[k])}}{p(s[k])}$$

$$p(y[k]) = \sum_{s_i \in \mathcal{S}} p(y[k], s_i)$$



Decomposing Terms in the Viterbi Algorithm

$$\frac{p(s[k]|y[k])p(y[k])}{p(s[k])}$$

Outline

State Redundancy

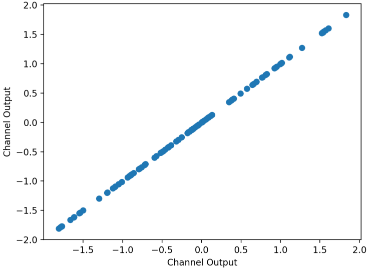


Figure: AWGN channel output for CIR $= [1, .7, 0.4, 0.1]$ with SNR $= 15$ dB

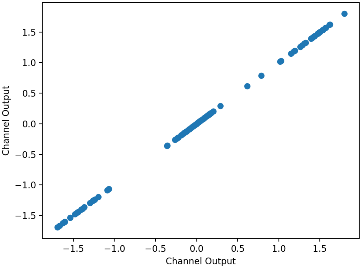


Figure: AWGN channel output for CIR $= [1, 1, 0.15, 0.1]$ with SNR $= 15$ dB

Exploiting State Redundancy

1. Cluster some set of observed channel output into desired number of states

Exploiting State Redundancy

1. Cluster some set of observed channel output into desired number of states
2. For states with ambiguous channel input, use majority decision

Exploiting State Redundancy

1. Cluster some set of observed channel output into desired number of states
2. For states with ambiguous channel input, use majority decision
3. *Choosing too few states will degrade performance

Outline

Simulation System

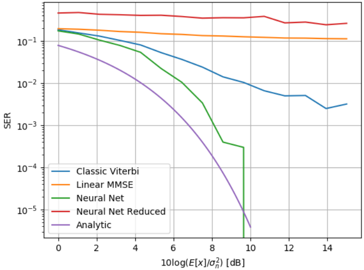
- BPSK with AWGN at receiver with SNR given by

$$\mathrm{SNR} = \frac{E\{|x[k]|^2\}}{E\{|n[k]|^2\}}$$

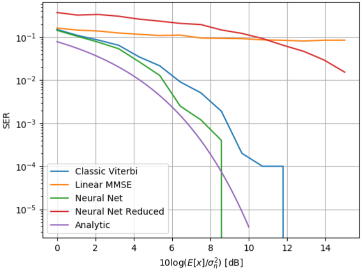
- Impulse response is normalized.

Detection Performance: LTI Channel

$\text{CIR} = [0.9, 0.7, 0.3, 0.5, 0.1]$



$\text{CIR} = [.9, 0, .0, .4, .7]$



*Reduced states uses 8 states in both figures above

Detection Performance: Quantizer

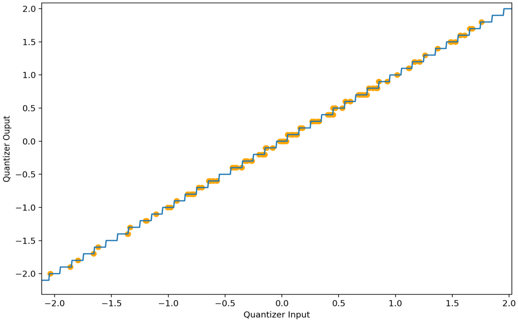
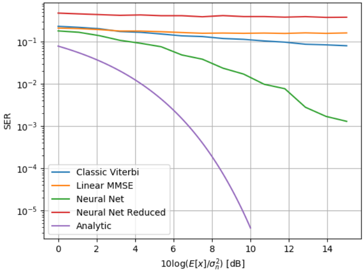


Figure: Quantizer input/output for AWGN channel with CIR = $[0.9, 0.7, 0.3, 0.5, 0.1]$ (32 states) with SNR = 10 dB

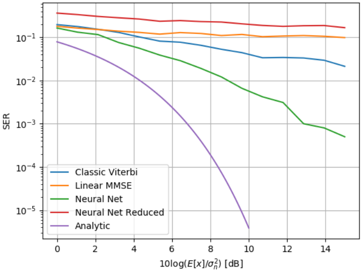
Implementated by rounding down to chosen number of decimal places *after* adding noise at receiver.

Detection Performance: LTI Channel with Quantization

$\text{CIR} = [0.9, 0.7, 0.3, 0.5, 0.1]$



$\text{CIR} = [.9, 0, .0, .4, .7]$



*Reduced states uses 8 states in both figures above

*Note that the "Classic" Viterbi is no longer using ideal metric here

Outline

Other Notes

- Can be applied to other algorithms (BCJR).
- Generate training data for molecular communications channel and test on real data.

Thank You.

Questions or Comments?