ECE 175B: Probabilistic Reasoning and Graphical Models
Lecture 16: Inference in Convolutional Wireless Channels

Florian Meyer & Ken Kreutz-Delgado

Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Convolutional Wireless Channel Model

Convolutional (FIR) Cellular Communication Channel Model with AWGN

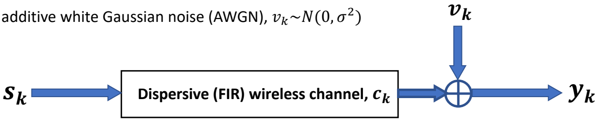
y_k = received continuous signal

s_k = transmitted discrete symbol $\in \{0,1\}$

v_k = stationary additive white Gaussian noise (AWGN), $v_k \sim N(0, \sigma^2)$

c_0 = channel fade parameter

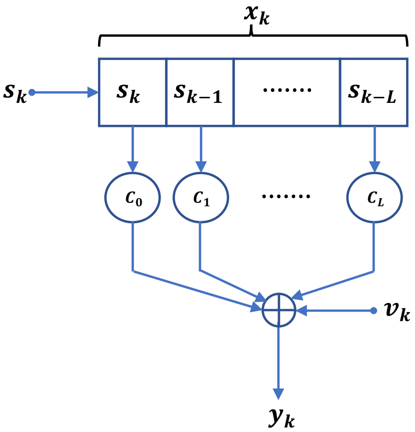
c_j = intersymbol interference parameter, $1 \leq j \leq L$



$$\boldsymbol{y_k = c_0 s_k + c_1 s_{k-1} + \cdots + c_L s_{k-L} + v_k}$$

This model is referred to as a *dispersive channel model* because the received signal $\boldsymbol{y_k}$ is a combination of sent symbols $\boldsymbol{s_k}$ that have been "dispersed" over time. Much of the dispersion, which is captured in the convolution structure of the model, is due to multipath reflection of signals bouncing off buildings, trees, and other large objects in the channel environment. To estimated the sent symbols from the received signals requires "undoing" the dispersive convolution; for this reason symbol estimation is called "channel deconvolution". Of course, this procedure should also ameliorate the corruption due to the additive white gaussian channel noise.

Equivalent Shift-Register-State HMM Model



Let the symbols s_k be iid with known distribution $P(s_k)$. E.g., $P(s_k = 1) = p$ and $P(s_k = 0) = q = 1 - p$. The "shift-register state" x_k **is the state of a Markov Chain** with **known transition probabilities** $P(x_{k+1} | x_k)$ determined from $P(s_k)$. When the symbols are unknown, then the shift-register model defines a **Hidden Markov Model (HMM).**

With $y_k = c^T x_k + v_k$ and $v_k \sim N(0, \sigma^2)$, the probability of the observation y_k given the HMM state x_k is Gaussian,

$$P(y_k | x_k) \sim N(c^T x_k, \sigma^2)$$

For a given FIR lag L, the model is completely specified once the **FIR channel coefficients** $\mathbf{c}^T = (c_0, c_1, \cdots, c_L)$ and the AWGN **channel noise variance** σ^2 are known. Given **pilot symbols** as known channel symbols and **associated channel output observations**, these can be **learned via MLE**.

Once the channel model is learned, message symbols can be estimated using the Forward-Backward or Viterbi Algorithms.

Setup - I

Binary symbols $s_k \in \{0, 1\}$ are transmitted wirelessly to your cell phone in *blocks* (aka *frames*) of length $B = P + N$, where P is the number of a priori known "pilot" symbols used to estimate ("sound") the channel and the remaining N symbols in the block contain the actual transmitted information.

For fixed block-length B there is a trade off between the size of P, with larger sizes allowing for better channel estimation (and hence a lower symbol detection error rate), and N, with larger N corresponding to a longer sent message.

One needs to optimize between the number of symbols correctly detected by the receiver and the length of the message, and this impacts the choice of the relative size of P versus N.

Setup - II

A simple, but very effective wireless communication channel model that models dispersion effects and noise is the $(L+1)$-tap FIR (finite impulse response) plus AWGN ("additive white Gaussian noise") model,

FIR + AWGN Channel Model:

$$y_k = c_k * x_k + v_k = c_0 s_k + c_1 s_{k-1} + \cdots + c_L s_{k-L} + v_k = \mathbf{c}^T \mathbf{x}_k + v_k$$

with

$$\mathbf{c} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_L \end{pmatrix} \in \mathbb{R}^{L+1}, \quad \mathbf{x}_k = \begin{pmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-L} \end{pmatrix} \in \mathbb{R}^{L+1} \quad \text{and} \quad v_k \sim \mathrm{N}(0, \sigma^2)\,,$$

where it is assumed that $c_0 \neq 0$. L is known as the *delay* or *lag* of the FIR filter. The noise is assumed to be independent ("white") and identically distributed (iid), and Gaussian. It is also assumed that the binary symbols, s_k, are iid and that $s_j \perp\!\!\!\perp v_k$ for all j and k, which implies that $\mathbf{x}_j \perp\!\!\!\perp v_k$ for all j, k. The unknown quantities that define the channel model are the FIR filter tap weights $\mathbf{c}$ and the scalar noise variance σ^2. Thus there are $L+2$ model parameters that need to be learned. The scalar measurement $y_k \in \mathbb{R}$ is the signal received at the cell phone. The parameter c_0 is called the *channel fade* and the term $c_i s_{k-i}$, $i = 1, \cdots, L$, is *intersymbol interference*.

Setup - III

As we shall see below, under the assumptions of our model the vector x_k is the state of a Markov Chain. Thus if the model parameters are known, then for any block of received data one can set up a finite-state HMM from which one find a Bayes-optimal ("minimum symbol error rate") MAP estimate of the corresponding sent message symbols for each time k within the block, via the Forward-Backward MPA, or of the entire string of message symbols, via the Viterbi Algorithm.

Although our assumptions are made for convenience, it is rational to determine symbol MAP estimates using a learned model under the supposition that despite the fall-off in performance from the ideal situation, the performance in practice will still be very good for an appropriately chosen value of the lag L and the number of pilot symbols P.

The signal model assumes the channel coefficients are *convolved* with the input signals,

$$y_k = c_k * x_k + v_k.$$

For this reason we have a *convolutional channel model* and determining estimates of the individual symbols s_k is called *deconvolution* (because in essence we are undoing the channel convolution). Therefore to obtain optimal estimations of the symbols s_k is called *optimal deconvolution*.

Setup - IV

A frame (aka block) of $(P+N)$ transmitted symbols is laid out as

$$\Big[\ \underbrace{\bar{s}_{1-L}\ \cdots\ \bar{s}_{-1}\ \ \bar{s}_0\ \ \bar{s}_1\ \cdots\ \bar{s}_\ell\ \cdots\ \bar{s}_{M-L+1}\ \cdots\ \bar{s}_M}_{P\ \text{ Pilot Symbols}}\ \Big|\ \underbrace{s_1\ \ s_2\ \cdots\ s_k\ \cdots\ s_{N-1}\ \ s_N}_{N\ \text{ Message Symbols}}\ \Big]$$

The **barred symbols** $\bar{s}_\ell$ denote the $P = M + L + 1$ **pilot symbols** used to estimate the $L+2$ parameters defining the FIR dispersive channel model. The **unbarred symbols** s_k denote the **actual message** sent through the channel that must be estimated ("deconvolved") once the channel has been learned. Note that there is no message symbol s_0. Because there are $L+2$ symbols to learn by sounding the channel with pilot signals, *ideally* we want the number of pilot signals P to satisfy $P \gg L+2$, a constraint that limits the size, L, of the FIR filter in practice.

Note that **within the pilot symbols subblock** we let the parameter $\boldsymbol{\ell}$ index the barred preamble pilot symbols $\bar{s}_\ell$, $\ell = 1-L, \cdots, M$, and **within the message symbols subblock** we let the parameter $\boldsymbol{k}$ index the unbarred message symbols s_k, $k = 1, \cdots N$. Thus the parameters ℓ and k serve different roles in indexing symbols in the frame.

We have put the pilot symbols at the beginning of the frame and thus they are known as the **preamble** of the block of sent data. If the pilot symbols are tucked in the middle of the block, they are then the *midamble*, and if at the end of the block, the *postamble*.

Setup - V

Let $\bar{\mathbf{x}}_\ell$ and $\bar{y}_\ell$, for $\ell = 1, \cdots, M$, be associated purely with the pilot ("barred") data, $\bar{s}_\ell$, $\ell = 1 - L, \cdots, M$ in the sent block of symbols. **Note:** because of the delay in the FIR channel, the indices for $\bar{\mathbf{x}}_\ell$ and $\bar{y}_\ell$ begin at $\ell = 1$ whereas that of $\bar{s}_\ell$ begins at $\ell = 1 - L$.

Let $\mathbf{x}_k$ and y_k, for $k = 1, \cdots, N$, be associated with the sent message symbols s_k, which are to be estimated. We also set $\mathbf{x}_0 = \bar{\mathbf{x}}_M$, with initialization probability $p(\mathbf{x}_0) = p(\bar{\mathbf{x}}_M) = 1$ (because $\bar{\mathbf{x}}_M$ is *a priori* known). **Note:** because of the delay in the FIR channel, some of the sent symbols which comprise the elements of the vector $\mathbf{x}_k$ for $k = 1, \cdots, L - 1$ are pilot symbols. For this reason, for convenience, we define $s_j = \bar{s}_{j+M}$ for $j = 0, -1, \cdots, -L + 1$. For example,

$$\mathbf{x}_1 = \begin{pmatrix} s_1 \\ s_0 \\ \vdots \\ s_{1-L} \end{pmatrix} = \begin{pmatrix} s_1 \\ \bar{s}_M \\ \vdots \\ \bar{s}_{M-L+1} \end{pmatrix}.$$

Because s_k takes on a finite set of possible values (two here), the vector $\mathbf{x}_k$ takes its values in a finite set. By the use of clever coding techniques, communications engineers can ensure that *the transmitted symbols, s_k, are independent and identically distributed* (iid). Note that the vectors $\mathbf{x}_k$ do *not* form an independent sequence. However, they can be modeled as the finite-state values of a serial Markov chain by exploiting the independence of the symbols s_k. Further exploiting the mutual independence of s_j and v_k (and hence between $\mathbf{x}_j$ and v_k) for all j, k allows us to construct a finite-state, continuous output hidden Markov model (HMM).

Three Crucial Steps to Perform

1. **Learn the Channel Model Parameters**

   Given the known pilot data $\bar{\mathbf{x}}_\ell$, $\ell = 1, \cdots, M$, and the corresponding received data $\bar{y}_\ell$, find the conditional (conditioned on the pilot data) MLE of the parameters $\mathbf{c}$ and σ^2.

2. **Perform Optimal Channel Deconvolution**

   Specify the HMM and implement an algorithm to determine estimates of the unknown sent message symbols. This can be the MPA (aka Forward-Backward algorithm) to obtain the minimum symbol error rate ("Bayes optimal") estimate for each message symbol s_k, $k = 1, \cdots, N$, given the received data $Y_1^N = \{y_1, \cdots, y_N\}$. Or this can be the Viterbi Algorithm (aka, Min-Sum Algorithm) to determine the most probable entire sequence of message symbols.

Conditional MLE of the Model Parameters

Note that the pilot data for $\ell = 1, \cdots, M$ is modeled by $\bar{y}_\ell = c^T \bar{x}_\ell + \bar{v}_\ell$. Define

$$\bar{y} = \begin{pmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_M \end{pmatrix} \in \mathbb{R}^M, \quad \bar{v} = \begin{pmatrix} \bar{v}_1 \\ \vdots \\ \bar{v}_M \end{pmatrix} \in \mathbb{R}^M \quad \text{and} \quad \bar{\mathbb{X}} = \begin{pmatrix} \bar{x}_1^T \\ \vdots \\ \bar{x}_M^T \end{pmatrix} \in \mathbb{R}^{M \times (L+1)}.$$

This gives

$$\bar{y} = \bar{\mathbb{X}}\, c + \bar{v} \iff \bar{y} \,\big|\, \bar{\mathbb{X}} \sim \mathrm{N}(\bar{\mathbb{X}}\, c,\, \sigma^2 I) \iff \bar{y}_\ell \,\big|\, \bar{x}_\ell \sim \mathrm{N}(\bar{x}_\ell^T c,\, \sigma^2) \text{ for } \ell = 1, \cdots, M$$

where $y \,\big|\, \bar{\mathbb{X}}$ indicates that we are conditioning on the pilot data matrix $\bar{\mathbb{X}}$ and I is the $M \times M$ identity matrix. The negative pilot-data-conditional loglikelihood is therefore

$$-\ell(c, \sigma^2) \propto M \ln \sigma^2 + \frac{1}{\sigma^2} \|\bar{y} - \bar{\mathbb{X}}\, c\|^2.$$

Minimizing this with respect to c gives

$$\hat{c} = \arg\min_c \|\bar{y} - \bar{\mathbb{X}}\, c\|^2 = \bar{\mathbb{X}}^+ \bar{y} = \left(\bar{\mathbb{X}}^T \bar{\mathbb{X}}\right)^{-1} \bar{\mathbb{X}}^T \bar{y} = \left(\frac{1}{M} \sum_\ell \bar{x}_\ell \bar{x}_\ell^T\right)^{-1} \left(\frac{1}{M} \sum_\ell \bar{x}_\ell \bar{y}_\ell\right) = \hat{R}_{\bar{x}\bar{x}}^{-1} \hat{R}_{\bar{x}\bar{y}}$$

where $\hat{R}$ denotes sample correlation matrices. Minimizing $-\ell(\hat{c}, \sigma^2)$ with respect to σ^2 gives

$$\widehat{\sigma^2} = \frac{1}{M} \|\bar{y} - \bar{\mathbb{X}}\, \hat{c}\|^2 = \frac{1}{M} \sum_\ell (\bar{y}_\ell - \bar{x}_\ell^T \hat{c})^2 = \text{ sample variance }.$$

Hidden Markov Model - I

To make things concrete, take the FIR lag to be $L = 1$ and the symbols to be binary $s_k \in \{0, 1\}$. This gives the simple channel model

$$y_k = c_0 s_k + c_1 s_{k-1} + v_k = \begin{pmatrix} c_0 & c_1 \end{pmatrix} \begin{pmatrix} s_k \\ s_{k-1} \end{pmatrix} + v_k = c^T x_k + v_k$$

where x_k is categorical, taking on the four possible values $x_k \in \{\xi_1, \xi_2, \xi_3, \xi_4\}$,

$$\xi_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \xi_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \xi_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \xi_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Take $P(s_k = 1) = p$ and $P(s_k = 0) = q = 1 - p$. Typically $p = q = 0.5$. Note that

$$P(x_{k+1}|x_k, \underbrace{x_{k-1}, x_{k-2} \cdots}_{\text{past}}) = P(s_{k+1}, s_k|s_k, s_{k-1}, s_{k-2}, s_{k-3}, \cdots) = P(s_{k+1}, s_k|s_k, s_{k-1}) = P(x_{k+1}|x_k)$$

because s_k is iid. **Thus x_k is a Markov Chain.** Define $p_{ij} = P(x_{k+1} = \xi_i | x_k = \xi_j)$. It is straightforward to determine that the transition probabilities p_{ij} are given by

$$\begin{array}{llll} p_{11} = q & p_{12} = q & p_{13} = 0 & p_{14} = 0 \\ p_{21} = 0 & p_{22} = 0 & p_{23} = q & p_{24} = q \\ p_{31} = p & p_{32} = p & p_{33} = 0 & p_{34} = 0 \\ p_{41} = 0 & p_{42} = 0 & p_{43} = p & p_{44} = p \end{array}$$

Hidden Markov Model - II

With $y_k = c^T x_k + v_k$ where v_k are iid, Gaussian $v_k \sim N(0, \sigma^2)$ and independent of s_k, we have $y_k \big| x_k \sim \mathrm{N}(c^T x_k, \sigma^2)$ so that the state-conditional observational probabilities are

$$p(y_k | x_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{y_k - c^T x_k}{\sigma}\right)^2}$$

for $x_k \in \left\{\xi_1,\, \xi_2,\, \xi_3,\, \xi_4\right\}$. Furthermore, because the symbols s_k are iid, it is straightforward to ascertain that $y_k \perp\!\!\!\perp \{x_t, y_t\} \,\big|\, x_k$ for all $t \neq k$. With the fact that x_k is the state of a Markov Chain, this completes the proof that we have a correct Hidden Markov Model with hidden state x_k and state-dependent observations y_k.

Optimal Channel Deconvolution I

Given a completely specified HMM, in previous lectures we have shown that an HMM is Markov equivalent to a Markov Network with observations-dependent potentials given by

$$\psi(x_k, x_{k+1}) = \psi_{k,k+1}(x_k, x_{k+1}; y_{k+1}) = P(y_{k+1}|x_{k+1})P(x_{k+1}|x_k).$$

In Lecture 14 we give the MPA/Forward-Backward and Belief Propagation (BP) algorithm for computing the Bayes optimal single time estimate of an HMM state x_k given the observational data $Y_1^N = \{y_1, \cdots, y_N\}$ associated with the unknown symbols in a frame, while in Lecture 15 we give the Viterbi Algorithm (VA), aka the Min-Sum Algorithm in the negative-log-probability domain, for computing the Bayes optimal entire sequence of unknown states given the observational data.

We can implement these algorithms once we have estimated the parameter models because we can construct the needed potentials in the manner described above. For the case we considered with $L = 1$ and $s_k \in \{0, 1\}$ we determine this to be

$$\psi(x_k = \xi_j, x_{k+1} = \xi_i) = P(y_{k+1}|x_{k+1} = \xi_i)P(x_{k+1} = \xi_i|x_k = \xi_j) = \frac{p_{ij}}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{\boldsymbol{y}_k - \boldsymbol{c}^T\xi_i}{\boldsymbol{\sigma}}\right)^2}$$

Optimal Channel Deconvolution - II

Although we have obtained estimates of the unknown Markov state values x_k given the data $Y_1^N = \{y_1, \cdots, y_N\}$, we actually want estimates of the *symbols* s_k which are the components of x_k. Note, furthermore, that s_k can be the component of more than one state vector x_k.

For the Viterbi Algorithm, this is not an issue because a single, and therefore symbol-consistent, path is found through the trellis of possible state-paths. An inconsistent path is a probability zero path which the Viterbi Algorithm will ignore in its search for the maximum probability path.

For the Forward-Backward algorithm, which computes $P(x_k|Y_1^N)$ (or its unnormalized equivalent) we are not yet done because we need to first compute $P(s_k|Y_1^N)$ (or its unnormalized equivalent), i.e.,

$$P(s_k|Y_1^N) = \sum_{x_k \setminus s_k} P(x_k|Y_1^N)$$

after which one computes,

$$\hat{s}_k^{(\mathrm{map})} = \arg\max_{s_k} P(s_k|Y_1^N).$$