A.I. in health informatics

lecture 8 structured learning

kevin small &
byron wallace
today

• models for structured learning: HMMs and CRFs

  – structured learning is particularly useful in biomedical applications: parsing (clinical) text; genetic data, etc.

  – we’ll cover this in more detail; but need the basics first
unstructured learning

assumptions:

• we’re given a set of i.i.d. instances \( \{x_1, x_2, \ldots, x_N\} \)
  and their (univariate) labels \( \{y_1, y_2, \ldots, y_N\} \)

• no order or *sequence* to the data
unstructured learning: graphically

$y_1 \rightarrow x_1$

$y_2 \rightarrow x_2$

$y_3 \rightarrow x_3$

...
structured learning

assumptions:

- there is some correlation between a label $y_i$ and the preceding labels, in other words,

- $y_i$ is **structured**, i.e., $y_{i+1}$ is affected by the previous labels $y_i, y_{i-1} ...$
structured learning

• consider the task of part-of-speech (POS) tagging sentences
  – *nouns* tend to follow *verbs*

image from: http://www.cnts.ua.ac.be/pages/MBSP
structured learning: graphically
probability & structured learning

\[
p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_1, \ldots, x_{i-1})
\]

intractable!
(first-order) HMM

* usually assume states \((y_s)\) are *latent*; but not always (see weather example, upcoming)
(first-order) MM

\[ p(x_1, x_2, \ldots, x_n) = p(x_1) \prod_{i=2}^{n} p(x_i \mid x_{i-1}) \]

tractable!
markov model

from Bishop PMLR
Markov model: unfolded

$k = 1$

$k = 2$

$k = 3$

\[
\begin{align*}
A_{11} & \\
A_{33} & \\
A_{11} & \\
A_{33} & \\
A_{11} & \\
\end{align*}
\]

from Bishop PRML
markov model

• $a_{ij}$ – probability of transitioning from state $i$ to $j$
  - $\sum_j a_{ij} = 1$ (we have to go somewhere!)
  - $a_{ij} \geq 0$
let’s talk about the weather

• the world has three states:
  1 rainy, 2 cloudy, 3 sunny

• (in the weather case the markov model is not hidden)

• our transition matrix is:

$$A = \{a_{ij}\} = \begin{bmatrix}
0.4 & 0.3 & 0.3 \\
0.2 & 0.6 & 0.2 \\
0.1 & 0.1 & 0.8
\end{bmatrix}$$

note that this system likes to stay where it is!
the weather

• probability that of observing the weather \( \{sunny, sunny, rainy\} \)?

\[
A = \{a_{ij}\} = \begin{bmatrix}
0.4 & 0.3 & 0.3 \\
0.2 & 0.6 & 0.2 \\
0.1 & 0.1 & 0.8 \\
\end{bmatrix}
\]

\[
= 1.0 \times P(S|S) \times P(R|S) = .8 \times .1
\]
• it’s sunny today. what’s the probability that it remains so for $k$ days?

\[
A = \{a_{ij}\} = \begin{bmatrix}
0.4 & 0.3 & 0.3 \\
0.2 & 0.6 & 0.2 \\
0.1 & 0.1 & 0.8
\end{bmatrix} = (.8)^k
\]
**hidden** markov models

- in weather example, we only care about *transitions*
  - the (weather) states were the observations

- often we need to model data in which an observation is generated conditional on some latent, underlying state
  - e.g., bias coin example; urn-genie example

- enter the HMM
the urn example

A genie is in the room choosing urns to draw balls from – each earn contains different proportions of the various colored balls.

The actual urn the genie draws from is unobserved – we only know the sequence of draws.

\[
\begin{align*}
\text{URN 1} & : & \begin{cases}
P(\text{RED}) = b_1(1) \\
P(\text{BLUE}) = b_1(2) \\
P(\text{GREEN}) = b_1(3) \\
P(\text{YELLOW}) = b_1(4) \\
\vdots \\
P(\text{ORANGE}) = b_1(M)
\end{cases} \\
\text{URN 2} & : & \begin{cases}
P(\text{RED}) = b_2(1) \\
P(\text{BLUE}) = b_2(2) \\
P(\text{GREEN}) = b_2(3) \\
P(\text{YELLOW}) = b_2(4) \\
\vdots \\
P(\text{ORANGE}) = b_2(M)
\end{cases} \\
\text{URN N} & : & \begin{cases}
P(\text{RED}) = b_N(1) \\
P(\text{BLUE}) = b_N(2) \\
P(\text{GREEN}) = b_N(3) \\
P(\text{YELLOW}) = b_N(4) \\
\vdots \\
P(\text{ORANGE}) = b_N(M)
\end{cases}
\end{align*}
\]

O = \{GREEN, GREEN, BLUE, RED, \ldots\}
**hidden** markov models: sufficient parameters

$N$ states (latent), $M$ symbols (observed): symbols are observed conditioned on the current state

- $A$ - transition probabilities (from urn to urn)
- $B$ – symbol emission probabilities (color proportions in each urn)
- $\pi$ – initial state distribution (initial urn likelihood)

$\lambda = (A, B, \pi)$ specifies our model
hidden markov models: the three problems

1 given a set of observations \( o = \{o_1, o_2 \ldots o_T\} \) and a model \( \lambda \), compute \( P(o|\lambda) \)

2 given \( o, \lambda \), calculate most likely latent states (usually thought of as labels) \( q = \{q_1, q_2 \ldots q_T\} \)

3 given \( o \), calculate \( \lambda \)
given a set of observations $o = \{o_1, o_2 \ldots o_T\}$ and a model $\lambda$, compute $P(o|\lambda)$

$$
P(o | \lambda) = \sum_{all\ Q} P(o | q, \lambda)P(q | \lambda) =
$$

... cool. so we’re done?

not quite. this will require $O(N^T)$ calculations
dynamic programming to the rescue!

\[ O_1 \quad \cdots \quad O_k \quad O_{k+1} \quad \cdots \quad O_{K} = \text{Observations} \]

- \( O_1 \) to \( O_{K} \) represent observations at different time steps.
- \( S_1 \) to \( S_N \) represent states at each time step:
  - \( S_1 \) to \( S_k \) at time step \( k \), etc.
- \( a_{ij} \) denote transitions between states.

At time \( T \):
- The graph shows how states transition and how observations are influenced by these transitions.
- Dynamic programming is used to find the optimal sequence of actions or decisions that results in the best outcome over time.
dynamic programming to the rescue!

1) Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N.$$ 

2) Induction:

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^{N} \alpha_t(i)a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T - 1$$

$$\quad 1 \leq j \leq N.$$ 

3) Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$
about that runtime

N states to (N-1) other states
T times
can save a bit using the
backward direction, too (see
the paper); this will be
referred to with a β and is
analogous to α
hidden markov models: the three problems

1 given a set of observations $\mathbf{o} = \{o_1, o_2, \ldots, o_T\}$ and a model $\lambda$, compute $P(\mathbf{o} | \lambda)$

2 given $\mathbf{o}$, $\lambda$, calculate most likely latent states (usually thought of as labels) $\mathbf{q} = \{q_1, q_2, \ldots, q_T\}$

3 given $\mathbf{o}$, calculate $\lambda$
hidden markov models: problem 2

2 given a set of observations $o = \{o_1, o_2 \ldots o_T\}$ and a model $\lambda$, find most likely latent states (urns, say) $q^* = \{q_1, q_2 \ldots q_T\}$

• a unique solution need not exist! what are we even optimizing here?
  – individual most likely states $q_i$?
  – joint probability $q$?
hidden markov models: problem 2

**problem** optimizing for the most likely state at *any given time* can lead to *impossible* sequences under $\lambda$

so let’s consider the joint instead
solving problem 2;
the joint solution

want to solve for $q^*$:

\[ q^* \leftarrow \text{argmax}_{q} P(q,0 | \lambda) \]

• obviously enumerating all possible $q$ sequences is infeasible

• dynamic programming to the rescue, again!
the viterbi algorithm

define:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1, q_2, \ldots, q_t = i, O_1, O_2, \ldots, O_t | \lambda]$$

(most likely sequence up to time t-1). the inductive step:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1})$$

i.e., from the best path so far, we find the most likely transition/emission probability
hidden markov models: the three problems

1 given a set of observations \( o = \{o_1, o_2, \ldots, o_T\} \) and a model \( \lambda \), compute \( P(o|\lambda) \)

2 given \( o \), \( \lambda \), calculate most likely latent states (usually thought of as labels) \( q = \{q_1, q_2, \ldots, q_T\} \)

3 given \( o \), calculate \( \lambda \)
EM for $\lambda$

Define

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}$$

(from $i$ to $j$)

emit observation in state $j$

forward to state $i$

backward to state $j$

(the probability of being in state $i$ at time $t$ and state $j$ at time $t+1$)
EM for $\lambda$

probability of being in state $i$ is:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j).$$

(we have to transition somewhere)

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from } S_i \text{ to } S_j.$$
EM for $\lambda$

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } S_i \text{ at time } (t = 1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_i(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and other states}}{\text{expected number of times in state } j}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \overset{\text{s.t. } O_t = v_k}{=} \frac{1}{T}$$

all of these estimates can be computed from the observed data! (by counting!)
EM for $\lambda$

at a given time $t$ we have

$$\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$$

**E-step** calculate the likelihood of our observations $\mathbf{o}$ using the current estimates

**M-step** re-estimate the parameters (left-hand sides of equations on previous slide) using current estimates
shortcomings of HMMs

• HMMs model the joint probability of the observations \((x)\) and \((y)\)
  – ie., it’s a generative model

• but what we really care about is the conditional probability of a label sequence \(y\) given \(x\)

• enter conditional markov models
conditional models

conditional models don’t waste time modeling the observed data
MEMMs (2000, McCallum et al)

- MEMM: Maximum Entropy Markov Model

- output probability vector of transitioning from one state to all other states, given an input observation $\mathbf{x}$
  - *conditional* in that we estimate $p(\text{state})$ at time $t$ given an observation and the preceding state

\[
\begin{align*}
    & S_{t-1} \rightarrow S_t \\
    & S_t \rightarrow O_t \\
\end{align*}
\]

HMM

\[
\begin{align*}
    & S_{t-1} \rightarrow S_t \\
    & S_t \rightarrow O_t \\
\end{align*}
\]

MEMM
MEMMs v. HMM

• HMM prob. of emitting $o$ and being in state $s$ at time $t$

\[
\alpha_{t+1}(s) = \sum_{s' \in S} \alpha_t(s') P(s|s') P(o_{t+1}|s)
\]

• MEMM

\[
\alpha_{t+1}(s) = \sum_{s' \in S} \alpha_t(s') P_{s'}(s|o_{t+1})
\]
**CRFs** (2001, Lafferty et al)

- CRF: Conditional Random Field

- single model for the joint probability of the sequence of labels given the observations

- mitigates the *label bias* problem in which states with low-entropy (almost certain) transition probability vectors effectively ignore the observation

---

<table>
<thead>
<tr>
<th></th>
<th>HMM</th>
<th>MEMM</th>
<th>CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_{i-1}$</td>
<td>$Y_i$</td>
<td>$Y_{i+1}$</td>
<td>$Y_{i-1}$</td>
</tr>
<tr>
<td>$X_{i-1}$</td>
<td>$X_i$</td>
<td>$X_{i+1}$</td>
<td>$X_{i-1}$</td>
</tr>
<tr>
<td>$Y_{i-1}$</td>
<td>$Y_i$</td>
<td>$Y_{i+1}$</td>
<td>$Y_{i-1}$</td>
</tr>
<tr>
<td>$X_{i-1}$</td>
<td>$X_i$</td>
<td>$X_{i+1}$</td>
<td>$X_{i-1}$</td>
</tr>
<tr>
<td>$Y_{i-1}$</td>
<td>$Y_i$</td>
<td>$Y_{i+1}$</td>
<td>$Y_{i-1}$</td>
</tr>
<tr>
<td>$X_{i-1}$</td>
<td>$X_i$</td>
<td>$X_{i+1}$</td>
<td>$X_{i-1}$</td>
</tr>
</tbody>
</table>