Tools & Models for Data Science

Sequential Models

Chris Jermaine & Risa Myers

Rice University

So Far, Talked About “iid” Data

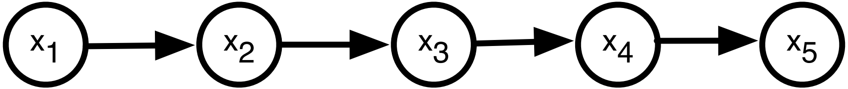
- “iid” = independent and identically distributed
- Each observation independent
- Not always realistic!
- Often, data are sequential (and therefore, not iid)
  - Temperature readings
  - Words in a sentence
  - Parts of speech: noun followed by verb $\cdots$
  - Stock prices
  - Many others!
- How can we make predictions in sequences?
- How can we solve a labeling problem in sequences?

"Markov Models"

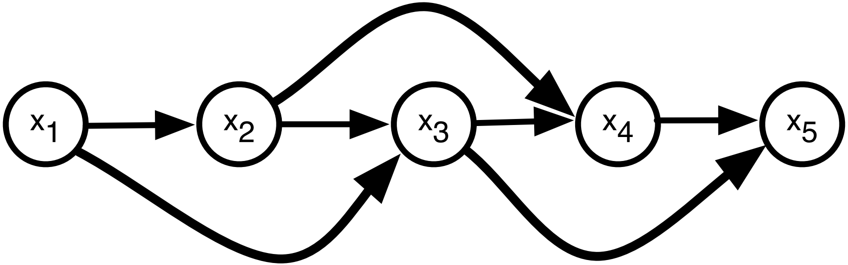
- Ubiquitous in data science
- Basic idea:
  - Data observed at a sequence of "time ticks"
  - Data at time tick t is x_t
  - **Markov Assumption**: x_t depends only on x_{t-1}
  - (Or on $x_{t-d}, x_{t-d+1}, x_{t-d+2}, \ldots, x_{t-1}$ for order-d model)
  - d is the number of time ticks in the past that contribute to the current time tick
  - Order d and Order 1 are not all that different
  - Asset: Going back 5 time ticks is not really more powerful, since the intermediary states can carry forward the information

Markov Model Order

- $d = 1$



- $d = 2$



Classic Sequential Model From Stats

- The "Autoregressive" Model
- Simple extension of linear regression
  - We are doing something like linear regression on the last d observations
  - Basically, compute the expected value of the next point, using a linear model of the last d points
  - Order-d model is called an AR(d) model
- As d increases, the plots get smoother

Classic Sequential Model From Stats

- The "Autoregressive" Model
- Simple extension of linear regression
  - Order-d model is called an AR(d) model
  - Have d regression coefs for an order-d model
  - $r_1, r_2, \ldots, r_d$
- Generative process is:

```
1 For t = 1 to d do:
2     x_t ~ Normal(μ, σ^2)

3 For t = d+1 to n do:
4     θ = Σ_{i=0}^{d-1} r_{i+1} × x_{t-d+i}

5     x_t ~ Normal(θ, σ^2)
```

$$x_t \sim \text{Normal}(\mu, \sigma^2)$$

$$\theta = \sum_{i=0}^{d-1} r_{i+1} \times x_{t-d+i}$$

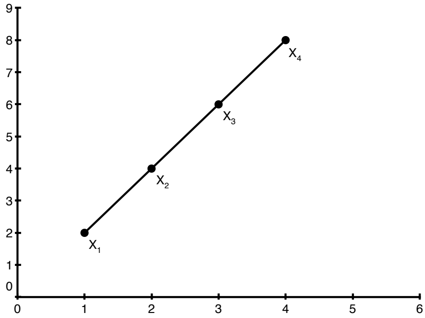
$$x_t \sim \text{Normal}(\theta, \sigma^2)$$

1. Initialize model generate first d data points

4. Dot product of regression coefficients with d observations gives the Expected value

5. Sample from a Normal distribution with that mean

Example



- To continue the trajectory, we need at least AR(2)
- Assume the step same step size to X_4 as was to X_3
- Assume each time tick is uniform
- Here, r = $< 2, -1 >$

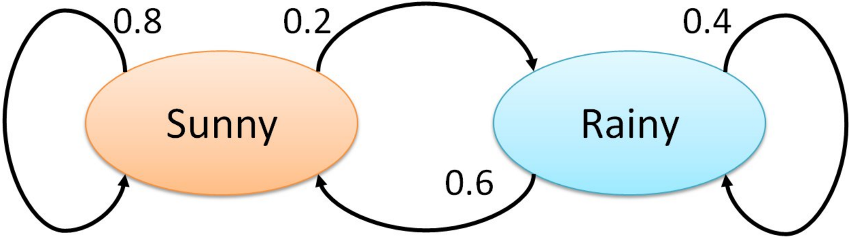
$$x_n = r_0 x_{n-1} + r_1 x_{n-2}$$

Definitions & Properties

- Markov Property
  - Future state depends only on the current state
- Markov Chain
  - Stochastic process with the Markov Property
  - May have an infinite number of states
- Markov Process
  - Stochastic process that transitions between states using provided probabilities
- Markov Model
  - Commonly viewed as any sequential model with a finite dependency back in time
  - Really means a finite state Markov Chain

Classic Sequential Model from CS

- Markov Model
  - Begins with a Markov chain
  - Assume that there are m states
  - We stochastically jump around between the states



Classic Sequential Model from CS

- Markov Model
  - Begins with a Markov chain
  - Assume that there are m states
  - We stochastically jump around between the states
  - Let π_0 be start probabilities
  - Let π_i be transition probabilities out of state i
  - Let s_1 be the start state selected from the start probabilities

$s_1 \sim \text{Categorical}(\pi_0)$
For t = 2 to n **do**:
    $s_t \sim \text{Categorical}(\pi_{s_{t-1}})$

Categorical Distribution

- Bernoulli distribution generalized for more than 2 choices
- Outcomes are discrete
- Each outcome has a probability
- Probabilities sum to 1
- Example
  - 10 balls into 5 baskets
  - Multinomial distribution tells you how the balls are distributed within the baskets
  - Categorical distribution tells you will basket a single ball landed in
- Another Example
  - Throw a weighted die
  - The side facing up is the selected category

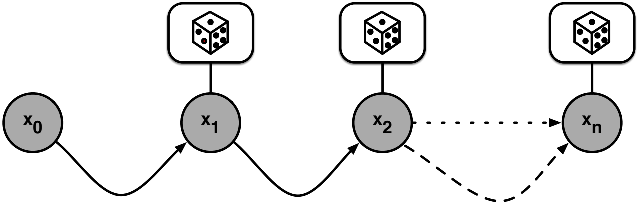
HMM

- Hidden Markov Model
- Called “hidden” because we typically don’t observe states
- We just see emitted values
- Most common type of Markov model

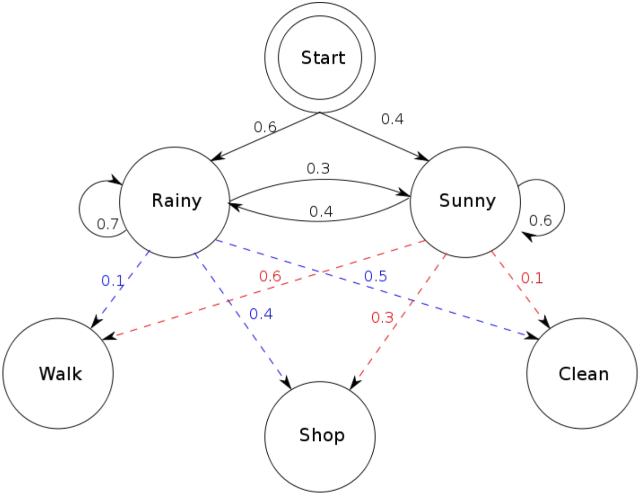
HMM

- Then we add the observed data
  - Often Categorical
  - Though sometimes not (Normal, Gamma, Poisson are common)
  - Let θ_s be parameter set associated with state s
- Called “hidden” because we typically don’t observe states

$s_1 \sim \text{Categorical}(\pi_0)$
$x_1 \sim f(\theta_{s_1})$
For t = 2 to n **do:**
  $s_t \sim \text{Categorical}(\pi_{s_{t-1}})$
  $x_t \sim f(\theta_{s_t})$



Example HMM



- $\pi_0 = [0.6 \;\; 0.4]$
- π matrix

$$\begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

- Categorical output. Different probabilities based on current state

Making Predictions using an HMM

- Problem: Predict the observation at x_{n+1}
- Given an HMM
- and a sequence $S = \langle x_1, x_2, ..., x_n \rangle$
- How to do it?

Making Predictions using an HMM

- Problem: Predict the observation at x_{n+1}
- Basic idea:
  - First, for each state s, find $p_s^{(n)}$
  - This is probability of being in state s at time tick n
  - Then, compute $p_s^{(n+1)} = \sum_{s'} p_{s'}^{(n)} \pi_{s',s}$
  - $\pi_{s',s}$ is probability of transitioning from state s' from s
  - Since we sum over all ways to get to s from tick t, $p_s^{(n+1)}$ is probability of state s at tick $n+1$
  - And choose $x_{n+1} = \mathrm{argmax}_{x_{n+1}} \sum_s p_s^{(n+1)} f(x_{n+1}|\theta_s)$
  - Now we have our prediction!
- BUT, still need to find $p_s^{(n)}$ for each s. How?

Making Predictions using an HMM

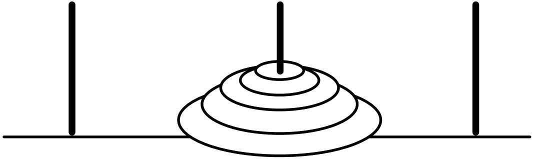
- Problem: Predict the observation at x_{n+1}
- In other words:
  - Figure out the most likely combination of where you are
  - ... and where you are going
  - Based on where you know you are, the transition probabilities, and the emission probabilities
- For example
  - If you stayed home and cleaned, what are you likely to do tomorrow?
- BUT, still need to find $p_s^{(n)}$ for each s. How?

Finding a Path Thru an HMM

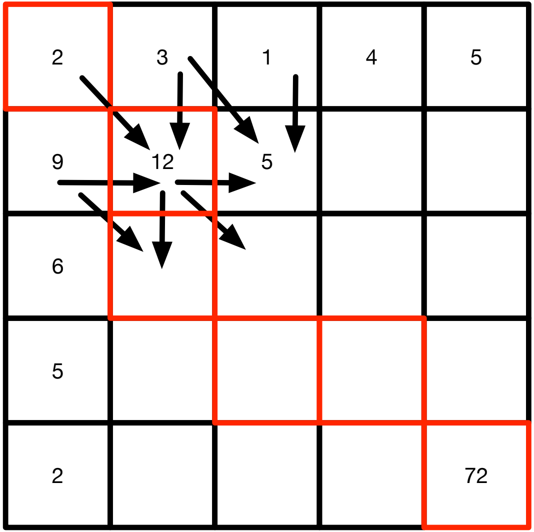
- Problem: Predict the observation at x_{n+1}
- How to compute the (posterior) probability we were in state s at each time tick?
- To do this, let $A[i,j]$ denote
  - Probability of us having $s_i = j$
  - Given that we have observed $\langle x_1, x_2, ..., x_i \rangle$
- Fill out A using dynamic programming

What is Dynamic Programming?

- Approach to solving problems with overlapping sub-problems
- The optimal solution must use the optimal sub-problem
- Basically, save your solutions
- Solve the base case
- Solve the recursive relationship
- Example: Tower of Hanoi
- Contrast with “divide and conquer”
  - Break a problem with non-overlappying sub-problems into pieces
  - Solve the sub-problems
  - Combine results



Common Application of Dynamic Programming?



- Populate top row and left column via base cases
- Populate internal cells from rule based on neighbors above and to the left
- Begin in upper left
- End in lower right

How to Compute the DP Matrix?

- Base case:
  - $A[1,j] \propto \pi_{0,j} \times f(x_1|\theta_j)$
  - Then normalize so $A[1,j] = \frac{A[1,j]}{\sum_{j'} A[1,j']}$
  - Note, normalization comes out of Bayes' rule: Pr[$s_1 = j$ given x_1] is...
  - Pr[$s_1 = j$ and x_1] / Pr[x_1]

- $\pi_{0,j}$ = Probability of being in state j at tick 1
- $f(x_1|\theta_j) =$ LLH of emitting x_1

How to Compute the DP Matrix?

- Recurrence:
  - $A[i,j] \propto \sum_{j'} A[i-1,j'] \times \pi_{j',j} \times f(x_1|\theta_j)$
  - Then normalize so $A[i,j] = \frac{A[i,j]}{\sum_{j'} A[i,j']}$
  - Note, normalization again comes out of Bayes' rule
- Now we can do our prediction!
- Use DP to compute A matrix
- Then use $p_s^{(n)} = A[n,s]$ to make prediction

- $A[i,j] =$ Pr(in state j at tick i | $x_1, x_2, \cdots, x_n$)
- $A[i-1,j']$ = Pr(I was in the last state)
- multiply $A[i-1,j']$ by
- $\pi_{j',j}$ the transition probability
- and $f(x_1|\theta_j)$ the emission probability

Now We Know How To Compute the Probability of a State

- Similar methods can be used to learn an HMM
- What do we mean by "learn an HMM"?
- Given a number of states, m
- And a set of sequential observations $\langle x_1, x_2, \ldots, x_n \rangle$
- Learn
  - π_0, the start probabilities
  - π, the transition probabilities between states
  - The parameters, θ_s of the emission distribution for each state s

Now We Know How To Compute the Probability of a State

- Similar methods can be used to learn an HMM
- Relies on an EM algorithm
- Why EM?
  - Missing data: we don't know the state at each time tick
  - EM is meant to solve MLE given missing data
  - EM for HMM aka "Baum-Welch algorithm"
- We won't derive the EM algorithm from the EM Q function
- We begin with the E-step
  - We need to be able to compute the probability that we are in state j at tick i, given a model
  - DP algorithm to do this is often called "forward-backward algorithm"

EM For Learning a HMM

- Let $C[i,j]$ be the probability that we are in state j at tick i
  - Given ALL of $\langle x_1, x_2, ..., x_n \rangle$
- How to compute? DP! Two other matrices will help...
- This is where the name comes from
- Forward: Let $\alpha[i,j]$ denote the probability
  - Of observing $\langle x_1, x_2, ..., x_i \rangle$
  - AND ending in state j
  - Takes into account everything before this time tick
- Backward: Let $\beta[i,j]$ denote the probability
  - Of observing $\langle x_{i+1}, ..., x_n \rangle$
  - Given we start in state j
  - takes into account everything after this time tick
- We combine these matrices to compute C

Combining the Two Probabilities

- Why do these help?
- Note that $C[i,j]$ is probability we are in state j
  - Given $\langle x_1, x_2, ..., x_i \rangle$
  - AND given $\langle x_{i+1}, x_2, ..., x_n \rangle$
- From Bayes' rule

  Pr[in state j | sequence until i, sequence after i] =

  $$C[i,j] = \frac{\text{Pr[in state } j \text{ with sequence until } i \text{ and sequence after } i\text{]}}{\text{Pr[whole sequence]}}$$
- So C can be expressed in terms of α and β

$$C[i,j] = \frac{\alpha[i,j]\beta[i,j]}{\sum_{j'} \alpha[i,j']\beta[i,j']}$$

- Still need to compute α, β

The Forward Pass

- Recall $\alpha[i,j]$ denotes the probability
  - Of observing $\langle x_1, x_2, ..., x_i \rangle$
  - AND ending in state j
- Compute with DP! Base case: probability of being in state j and observing the first output
  - $\alpha[1,j] \propto \pi_{0,j} \times f(x_1|\theta_j)$
  - Recall: π_0 is the vector of start probabilities
  - $f(x_1|\theta_j)$ is the probability of emitting x_1
- Recurrence
  - $\alpha[i,j] \propto f(x_i|\theta_j) \times \sum_{j'} \left(\pi_{j',j} \times \alpha[i-1,j']\right)$
  - entry $\propto$ LLH of observation from state $j \times$ sum over all possible ways to get to state j

The Backward Pass

- Recall $\beta[i,j]$ denotes the likelihood
  - Of observing $\langle x_{i+1},...,x_n\rangle$
  - Given we start in state j
- Again, compute with DP! Base case
  - $\beta[n,j] = 1$
  - We want the probability I see everything in the future if I'm in state j
  - But, I'm at tick n, so there is no future
  - The probability I observe nothing when I'm done is 1
- Recurrence
  - $\beta[i,j] \propto \sum_{j'} \left(\pi_{j,j'} \times f(x_{i+1}|\theta_{j'}) \times \beta[i+1,j']\right)$
  - The recursion happens backwards
  - Start in state j
  - Consider all possible next states in the next time tick, taking into account $\pi_{j,j'}$
  - $\beta[i+1,j']$ = how well does j' explain everything in the future $(i+2, i+3, \cdots)$

That's The E-Step!

- Let's relate this back to the coin flip EM
  - ? E-step: What was missing?

That's The E-Step!

- Let's relate this back to the coin flip EM
  - E-step: What was missing?
  - The identity of the coin
  - We computed the probability of the coin identity each time we reached into the bag
  - Given the current parameters, what's the probability that the current coin is coin 1? coin 2?
  - Say we see HHHTHH
  - and we estimate the probability of HEADS for the coins as $\langle 0.8, .03\rangle$
  - ? Which coin was more likely to generate that sequence?

  - $C[i,j]$ gives us the probability I'm in state j given the entire sequence
- How about the M-Step?

First: Estimate the Distributional Params

- Need to update each parameter θ_j
  - Set each θ_j to

$$\text{argmax}_{\theta_j} \sum_i \log C[i,j] f(x_i|\theta_j)$$

  - Note: If $C[i,j]$ is large, then that observation is more tightly coupled with that state
- What's going on here?
  - We are doing a MLE
  - Weighted on $C[i,j]$
  - Which is the probability that we were in state j at time i
  - Given the current model

Estimating the Transition Probs

- Consider D[2, sunny, rainy] = certainty I was in the sunny state at tick 2 and rainy state at tick 3
- Define $D[i,j,k]$ to be
  - The probability of being in state j at time i
  - AND being in state k at time $i+1$
  - AND seeing the entire sequence
  - Can be computed as

$$\frac{\alpha[i,j]\pi_{j,k}\beta[i+1,k]f(x_{i+1}|\theta_k)}{\sum_{j',k'}\alpha[i,j']\pi_{j',k'}\beta[i+1,k']f(x_{i+1}|\theta_{k'})}$$

- Why? Recall: $\alpha[i,j]$ is probability of $\langle x_1,...,x_i\rangle$ and ending in state j
- $\beta[i+1,k]$ is probability of $\langle x_{i+2},...,x_n\rangle$ starting in state k
- $\pi_{j,k}$ is prob of transition from state j to state k
- $f(x_{i+1}|\theta_k)$ is probability of emitting x_{i+1} in state k
- Put them together and normalize... exactly the probability we want!

More about the D Matrix

$$\frac{\alpha[i,j]\pi_{j,k}\beta[i+1,k]f(x_{i+1}|\theta_k)}{\sum_{j',k'}\alpha[i,j']\pi_{j',k'}\beta[i+1,k']f(x_{i+1}|\theta_{k'})}$$

- D is about both states: state j at tick i and state k at tick $i+1$
- How does π relate to D?
- π is a model parameter, we read it off the state transition diagram
- It DOESN'T describe a particular run, but rather the expected results
- D comes from estimating the parameters based on the emissions seen
- D says how certain I am that I was in the sunny state instead of the rainy state

Estimating the Transition Probs

- Then $\pi_{j,k}$ is estimated as:

$$\pi_{j,k} = \frac{\sum_i D[i,j,k]}{\sum_i C[i,j]}$$

- What's going on here?
  - $D[i,j,k]$ is the probability that we transitioned from j to k at tick i
  - So we are setting $\pi_{j,k}$ to be fraction of time we transitioned from j to k
  - Out of the total time that we were in j
- And start probs:

$$\pi_{0,j} = C[1,j]$$

  - This is simply the probability that we were in j at tick 0

Review of EM Steps

- Compute α and β to get C
  - The probability we are in state j at time i
- Use C to get D
  - The probability we are in state j at time i AND in state k at time $i+1$
- Use these to get f, θ
  - The parameters of the emission function
- Estimate π from these
  - The transition probabilities from state to state

This Allow Us to Learn HMM for One Big Sequence

- How to handle many sequences?
  - Have a special symbol ε at the end of each sequence
  - Have a special state e (for "end")
  - Set $f(\varepsilon|\theta_e) = 1, f(x \neq \varepsilon|\theta_e) = 0, f(\varepsilon|\theta_{s \neq e}) = 0$
  - And just concatenate all of the sequences, learn as a single sequence
- Example: A large number of sentences
- The emissions are wordso
- Use an extra state that you transition to every time you get a new sequence / sentence

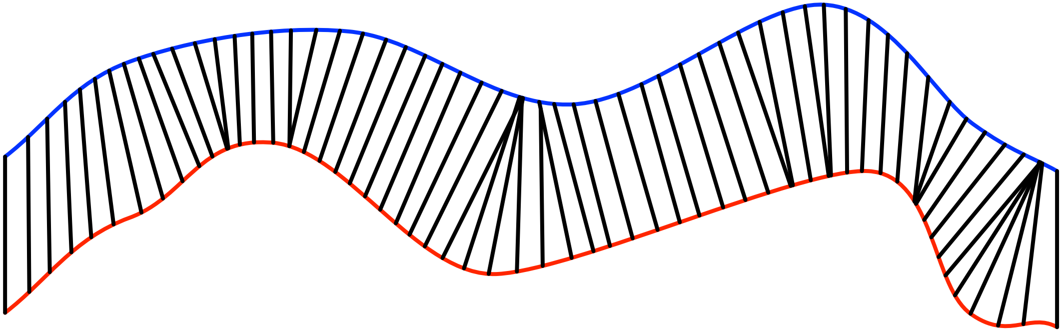
Advantages, Disadvantages and Other Algorithms

- HMMs are interpretable
- Recurrent Neural Networks can have higher accuracy

- Viterbi Algorithm
  - Computes the most likely sequence of hidden states given the emissions
- Dynamic Time Warping

Dynamic Time Warping

- Pairwise comparison of time series for classification
- Allows for distortion in the time dimension
- Works well for pattern matching
- Used in conjunction with k-Nearest Neighbors
- Can be used with different distance measures
- Implemented via dynamic programming



Questions?

- What do we know now that we didn't know before?
  - We understand some of the complexities of sequential data
  - We know some ways of making predictions for sequences
- How can we use what we learned today?
  - We can build models to classify sequences or predict from sequences