10-301/601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Hidden Markov Models

+

Exam 2 Review

Matt Gormley & Henry Chai
Lecture 18
Oct. 27, 2021

Reminders

- Lecture on Friday!
- Homework 6: Learning Theory / Generative Models
  - Out: Thu, Oct. 21
  - Due: Thu, Oct. 28 at 11:59pm
  - Same collaboration policy as Homework 3
    - Opt-in to homework groups on Piazza
  - IMPORTANT: you may only use 2 grace days on Homework 6
    - Last posible moment to submit HW6: Sat, Oct. 30 at 11:59pm
- Midterm Exam 2
  - Tue, Nov. 2, 6:30pm – 8:30pm
- Practice for Exam 2
  - Practice problems released on course website
    - (Tentatively) Out: Thu, Oct. 21
  - Mock Exam 2
    - (Tentatively) Out: Thu, Oct. 28
    - Due Sun, Oct. 31 at 11:59pm

MIDTERM EXAM LOGISTICS

Midterm Exam

- **Time / Location**
  - **Time: Tue, Nov. 2, 6:30pm – 8:30pm**
  - **Location & Seats:** You have all been split across multiple rooms. Everyone has an assigned seat in one of these room. Please watch Piazza carefully for announcements.
- **Logistics**
  - Covered material: Lecture 9 – Lecture 17
  - Format of questions:
    - Multiple choice
    - True / False (with justification)
    - Derivations
    - Short answers
    - Interpreting figures
    - Implementing algorithms on paper
  - No electronic devices
  - You are allowed to **bring** one 8½ x 11 sheet of notes (front and back)

Midterm Exam

- **How to Prepare**
  - Attend the midterm review lecture (right now!)
  - Review prior year's exam and solutions (we'll post them)
  - Review this year's homework problems
  - Consider whether you have achieved the "learning objectives" for each lecture / section
  - crowdsource exam questions

Midterm Exam

- **Advice (for during the exam)**
  - Solve the easy problems first
    (e.g. multiple choice before derivations)
    - if a problem seems extremely complicated you're likely missing something
  - Don't leave any answer blank!
  - If you make an assumption, write it down
  - If you look at a question and don't know the answer:
    - we probably haven't told you the answer
    - but we've told you enough to work it out
    - imagine arguing for some answer and see if you like it

Topics for Midterm 1

- Foundations
  - Probability, Linear Algebra, Geometry, Calculus
  - Optimization
- Important Concepts
  - Overfitting
  - Experimental Design
- Classification
  - Decision Tree
  - KNN
  - Perceptron
- Regression
  - Linear Regression

Topics for Midterm 2

- Classification
  - Binary Logistic Regression
- Important Concepts
  - Stochastic Gradient Descent
  - Regularization
  - Feature Engineering
- Feature Learning
  - Neural Networks
  - Basic NN Architectures
  - Backpropagation
- Learning Theory
  - PAC Learning
- Generative Models
  - Generative vs. Discriminative
  - MLE / MAP
  - Naïve Bayes

SAMPLE QUESTIONS

Sample Questions

3.2 Logistic regression

Given a training set $\{(x_i, y_i), i = 1, \ldots, n\}$ where $x_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0, 1\}$ is a binary label, we want to find the parameters $\hat{w}$ that maximize the likelihood for the training set, assuming a parametric model of the form

$$p(y = 1|x; w) = \frac{1}{1 + \exp(-w^T x)}.$$

The conditional log likelihood of the training set is

$$\ell(w) = \sum_{i=1}^{n} y_i \log p(y_i, |x_i; w) + (1 - y_i) \log(1 - p(y_i, |x_i; w)),$$

and the gradient is

$$\nabla \ell(w) = \sum_{i=1}^{n} (y_i - p(y_i|x_i; w)) x_i.$$

(b) [5 pts.] What is the form of the classifier output by logistic regression?

$h(\vec{x}) = \arg\max_y p(y|\vec{x}) = \begin{cases} 1 & \text{if } p(y|\vec{x}) > 0.5 \\ 0 & \text{otherwise} \end{cases}$

linear decision boundary

(c) [2 pts.] **Extra Credit:** Consider the case with binary features, i.e, $x \in \{0, 1\}^d \subset \mathbb{R}^d$, where feature x_1 is rare and happens to appear in the training set with only label 1. What is $\hat{w}_1$? Is the gradient ever zero for any finite w? Why is it important to include a regularization term to control the norm of $\hat{w}$?

Samples Questions

Q1

2.1 Train and test errors

In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $\mathcal{D}^{\text{train}}$, and tested on a separate test set $\mathcal{D}^{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

1. **[4 pts]** Which of the following is expected to help? Select all that apply.

   ✓ (a) Increase the training data size.

   (b) Decrease the training data size.

   (c) Increase model complexity (For example, if your classifier is an SVM, use a more complex kernel. Or if it is a decision tree, increase the depth).

   ✓ (d) Decrease model complexity.

   (e) Train on a combination of $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ and test on $\mathcal{D}^{\text{test}}$

   (f) ~~Conclude that Machine Learning does not work.~~

   toxic

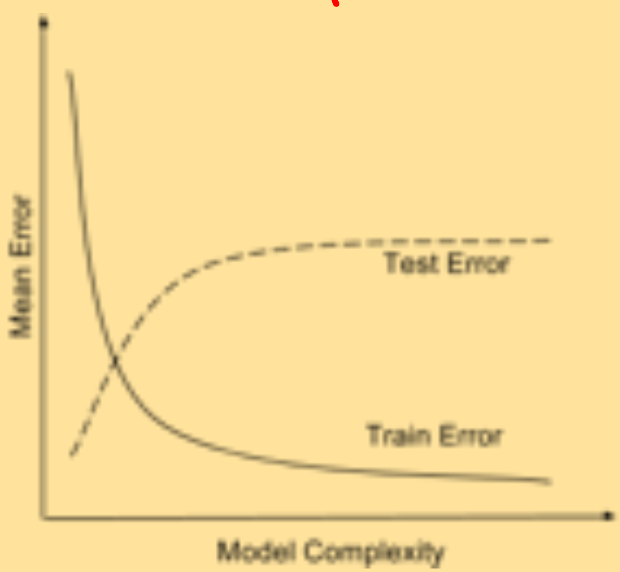
Samples Questions

2.1 Train and test errors

In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $\mathcal{D}^{\text{train}}$, and tested on a separate test set $\mathcal{D}^{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

4. **[1 pts]** Say you plot the train and test errors as a function of the model complexity. Which of the following two plots is your plot expected to look like?

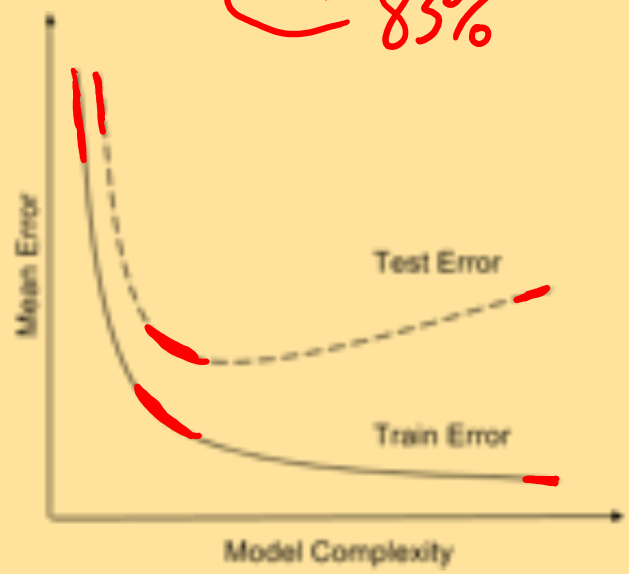
A



(a)

B 83%

C = toxic



(b)

Sample Questions

5 Learning Theory [20 pts.]

(a) [3 pts.] **T or F:** It is possible to label 4 points in $\mathbb{R}^2$ in all possible 2^4 ways via linear separators in $\mathbb{R}^2$.

(d) [3 pts.] **T or F:** The VC dimension of a concept class with infinite size is also infinite.

(f) [3 pts.] **T or F**: Given a realizable concept class and a set of training instances, a consistent learner will output a concept that achieves 0 error on the training instances.

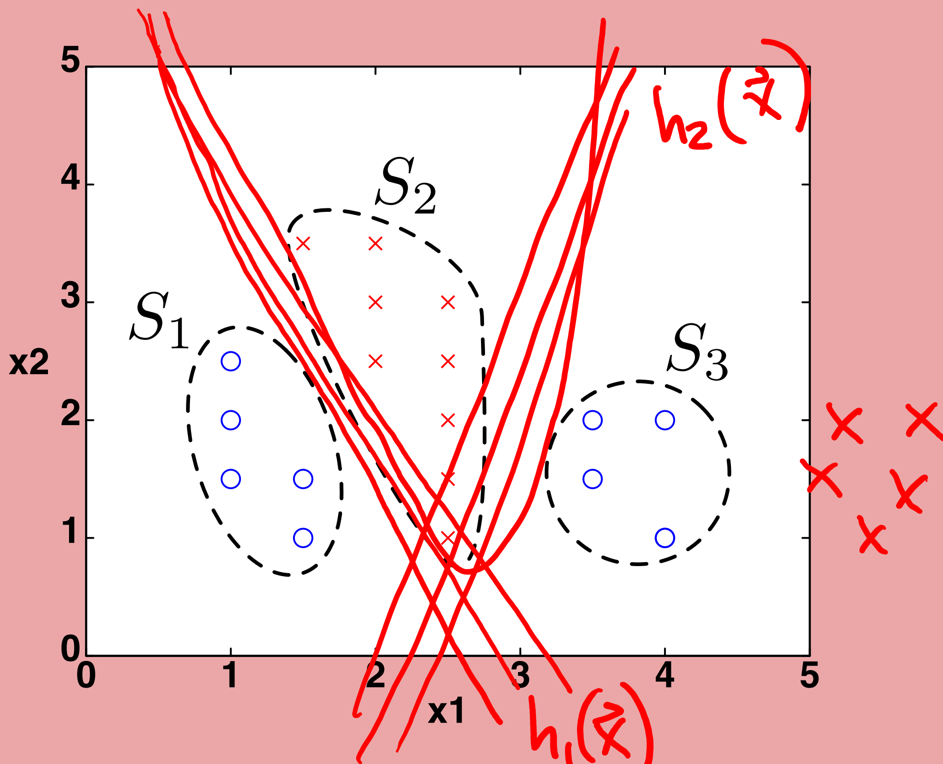
Sample Questions

Q3

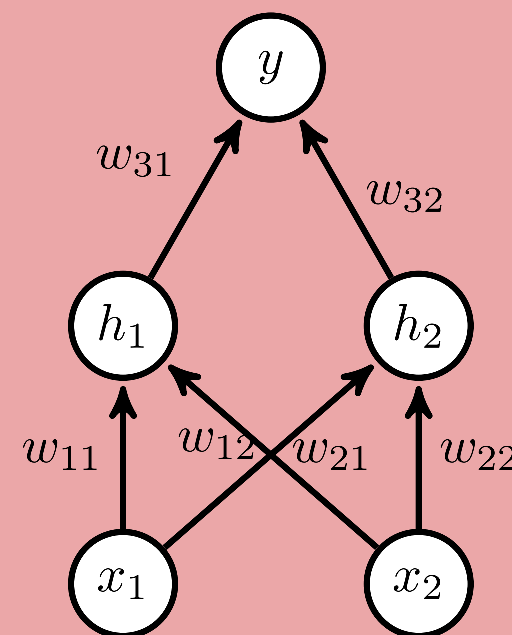
Neural Networks

A = toxic B = Yes C = NO
82%

Can the neural network in Figure (b) correctly classify the dataset given in Figure (a)?



(a) The dataset with groups S_1, S_2, and S_3.

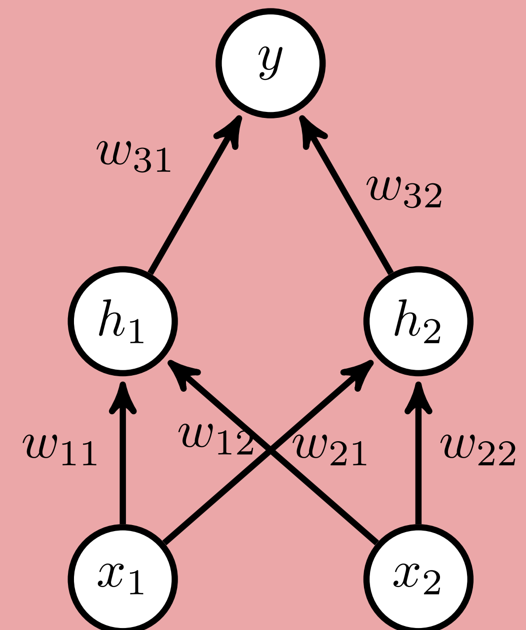


(b) The neural network architecture

Sample Questions

Neural Networks

Apply the backpropagation algorithm to obtain the partial derivative of the mean-squared error of y with the true value y^* with respect to the weight w_{22} assuming a sigmoid nonlinear activation function for the hidden layer.



(b) The neural network architecture

Sample Questions

1.2 Maximum Likelihood Estimation (MLE)

Assume we have a random sample that is Bernoulli distributed $X_1, \ldots, X_n \sim \text{Bernoulli}(\theta)$. We are going to derive the MLE for θ. Recall that a Bernoulli random variable X takes values in $\{0, 1\}$ and has probability mass function given by

$$P(X; \theta) = \theta^X (1 - \theta)^{1-X}.$$

(a) [2 pts.] Derive the likelihood, $L(\theta; X_1, \ldots, X_n)$.

(c) **Extra Credit:** [2 pts.] Derive the following formula for the MLE: $\hat{\theta} = \frac{1}{n} \left(\sum_{i=1}^{n} X_i\right)$.

Sample Questions

1.3 MAP vs MLE

Answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [2 pts.] **T or F:** In the limit, as n (the number of samples) increases, the MAP and MLE estimates become the same.

Sample Questions

1.1 Naive Bayes

You are given a data set of 10,000 students with their sex, height, and hair color. You are trying to build a classifier to predict the sex of a student, so you randomly split the data into a training set and a testing set. Here are the specifications of the data set:

- sex $\in$ {male,female}
- height $\in$ [0,300] centimeters
- hair $\in$ {brown, black, blond, red, green}
- 3240 men in the data set
- 6760 women in the data set

Under the assumptions necessary for Naive Bayes (not the distributional assumptions you might naturally or intuitively make about the dataset) answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [2 pts.] **T or F:** As height is a continuous valued variable, Naive Bayes is not appropriate since it cannot handle continuous valued variables.

(c) [2 pts.] **T or F:** $P(\texttt{height}|\texttt{sex}, \texttt{hair}) = P(\texttt{height}|\texttt{sex})$.

Naïve Bayes vs. Logistic Regression

Q4:

Question:

You just started working at a new company that manufactures comically large pennies. Your manager asks you to build a binary classifier that takes an image of a penny (on the factory assembly line) and predicts whether or not it has a defect.

What follow-up questions would you pose to your manager in order to decide between using a Naïve Bayes classifier and a Logistic Regression classifier?

Answer:

Question 4

Join by Web

1. Go to **PollEv.com**
2. Enter **10301601POLLS**
3. Respond to activity

Instructions not active. **Log in** to activate

MOTIVATION: STRUCTURED PREDICTION

Structured Prediction

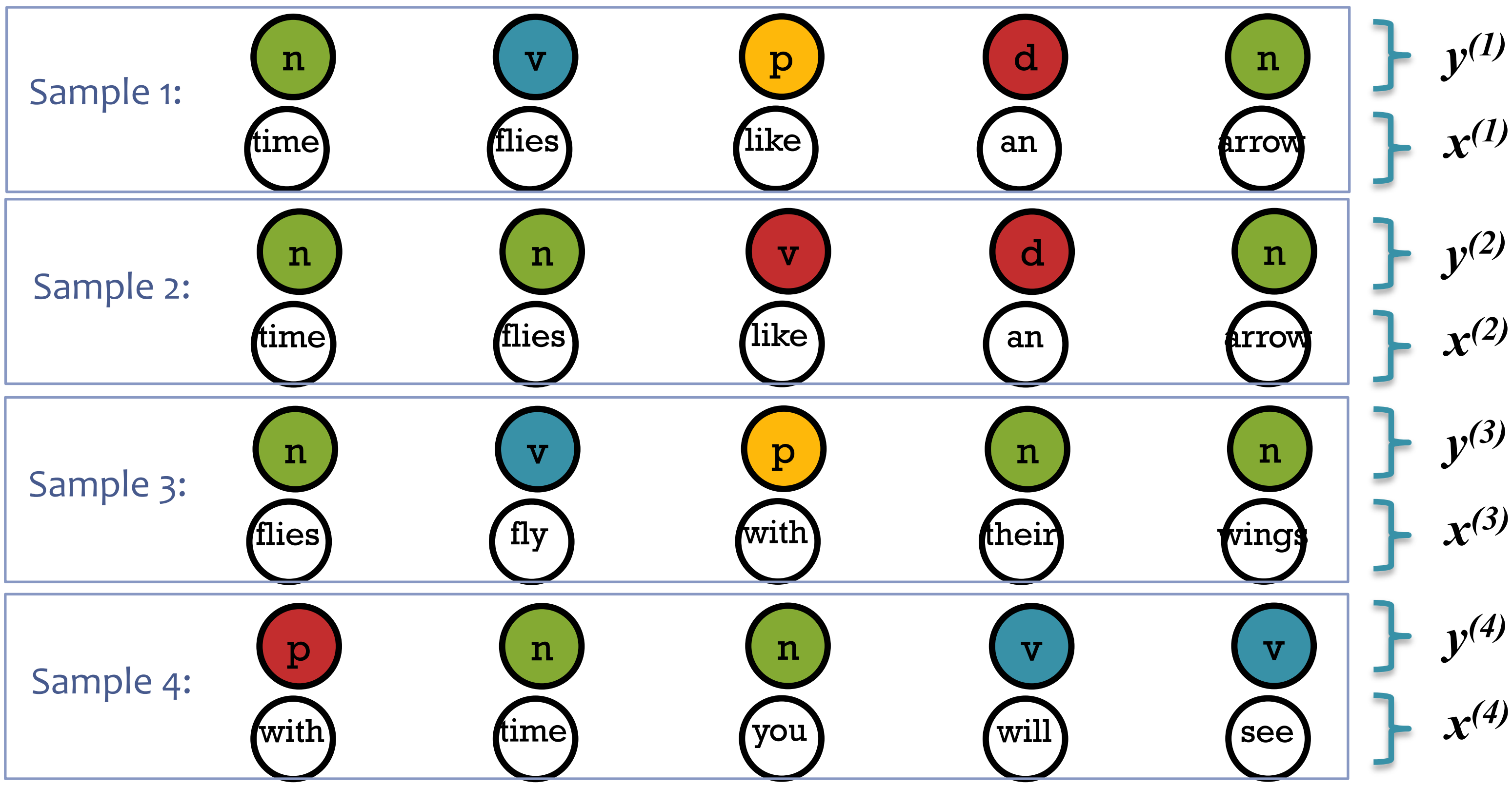
- Most of the models we've seen so far were for **classification**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  - Predict a (binary) **label:** y
- Many real-world problems require **structured prediction**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  - Predict a **structure:** $\boldsymbol{y} = (y_1, y_2, \ldots, y_J)$
- Some *classification* problems benefit from **latent structure**

Structured Prediction Examples

- **Examples of structured prediction**
  - Part-of-speech (POS) tagging
  - Handwriting recognition
  - Speech recognition
  - Word alignment
  - Congressional voting
- **Examples of latent structure**
  - Object recognition

Dataset for Supervised Part-of-Speech (POS) Tagging

Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$



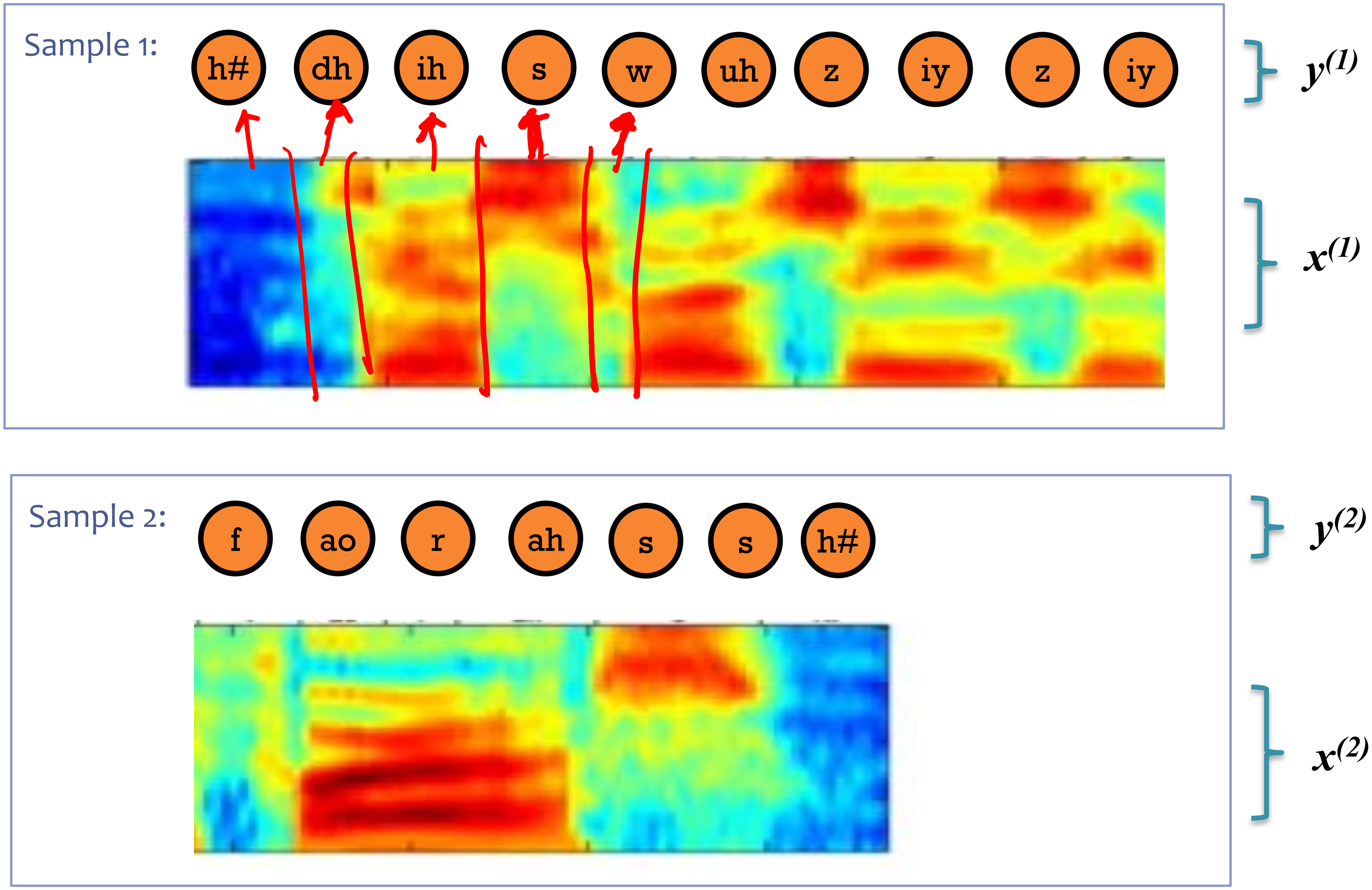
Dataset for Supervised Handwriting Recognition

Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Figures from (Chatzis & Demiris, 2013)

Dataset for Supervised Phoneme (Speech) Recognition

Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

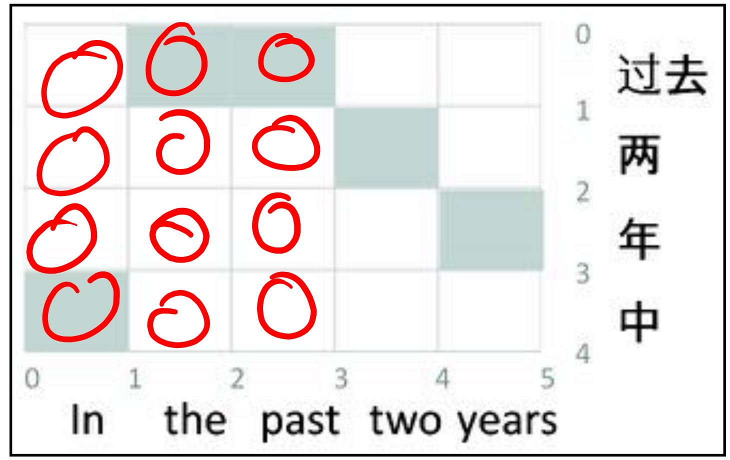


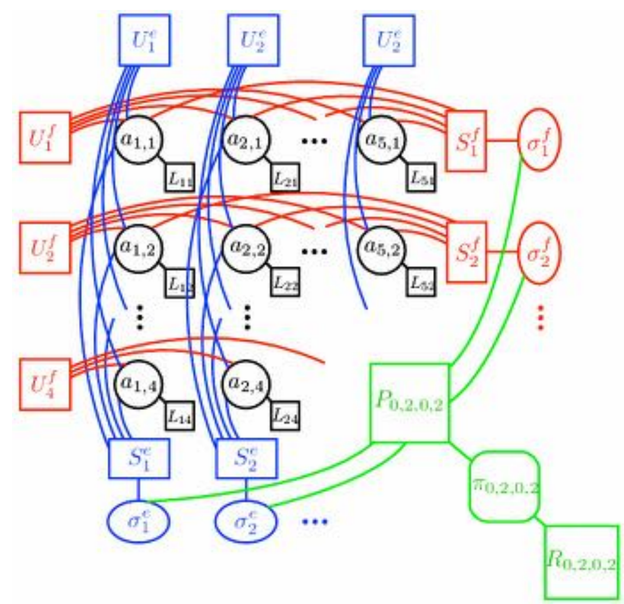
Figures from (Jansen & Niyogi, 2013)

Application:

Word Alignment / Phrase Extraction

- **Variables (boolean):**
  - For each (Chinese phrase, English phrase) pair, are they linked?
- **Interactions:**
  - Word fertilities
  - Few “jumps” (discontinuities)
  - Syntactic reorderings
  - “ITG contraint” on alignment
  - Phrases are disjoint (?)

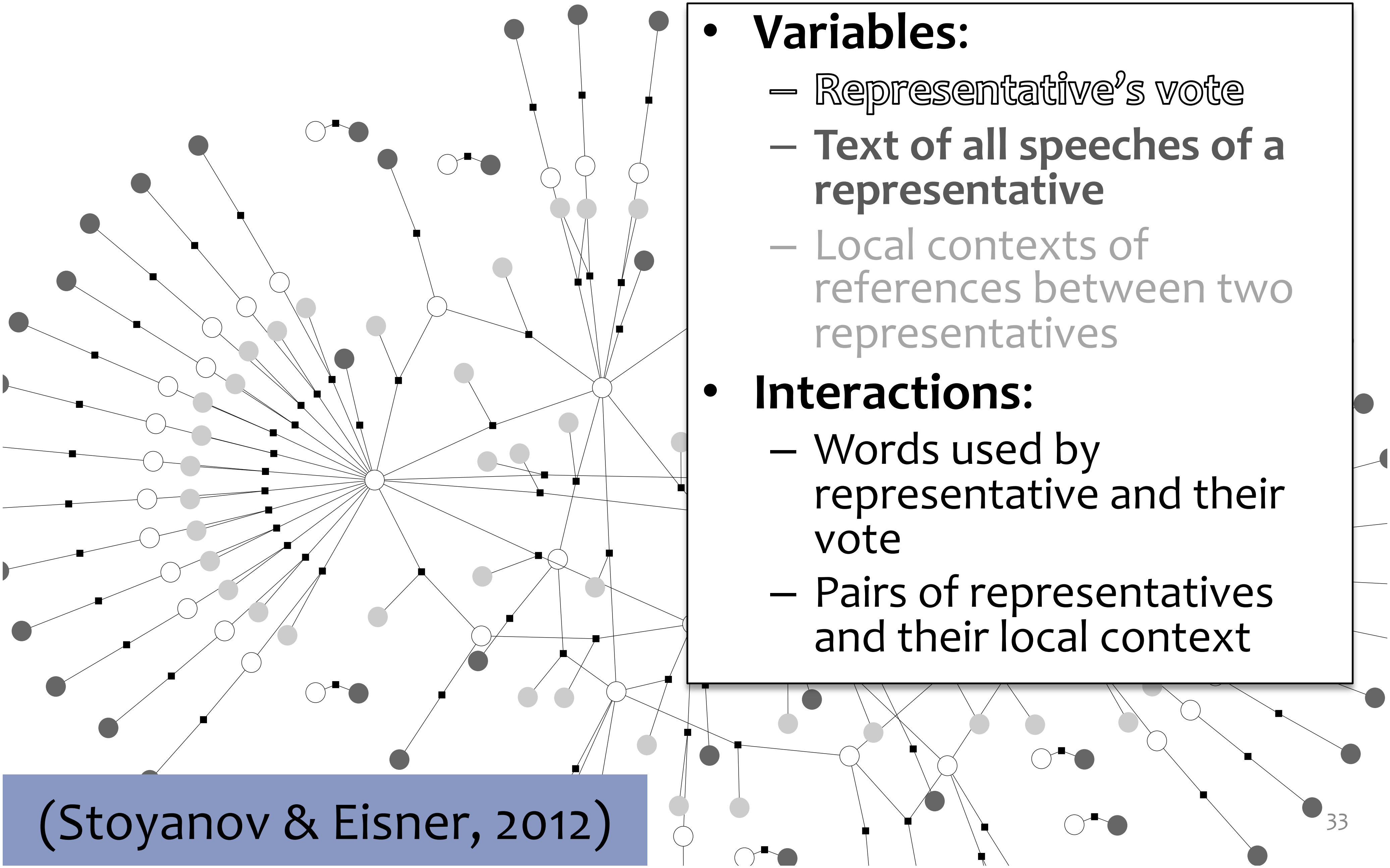




(Burkett & Klein, 2012)

Application:

Congressional Voting



- **Variables:**
  - Representative's vote
  - **Text of all speeches of a representative**
  - Local contexts of references between two representatives
- **Interactions:**
  - Words used by representative and their vote
  - Pairs of representatives and their local context

(Stoyanov & Eisner, 2012)

Structured Prediction Examples

- **Examples of structured prediction**
  - Part-of-speech (POS) tagging
  - Handwriting recognition
  - Speech recognition
  - Word alignment
  - Congressional voting
- **Examples of latent structure**
  - Object recognition

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

$x^{(1)}$

pigeon $y^{(1)}$

$x^{(2)}$

rhinoceros $y^{(2)}$

$x^{(3)}$

leopard $y^{(3)}$

$x^{(4)}$

llama $y^{(4)}$

Case Study: Object Recognition

Data consists of images x and labels y.

- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)

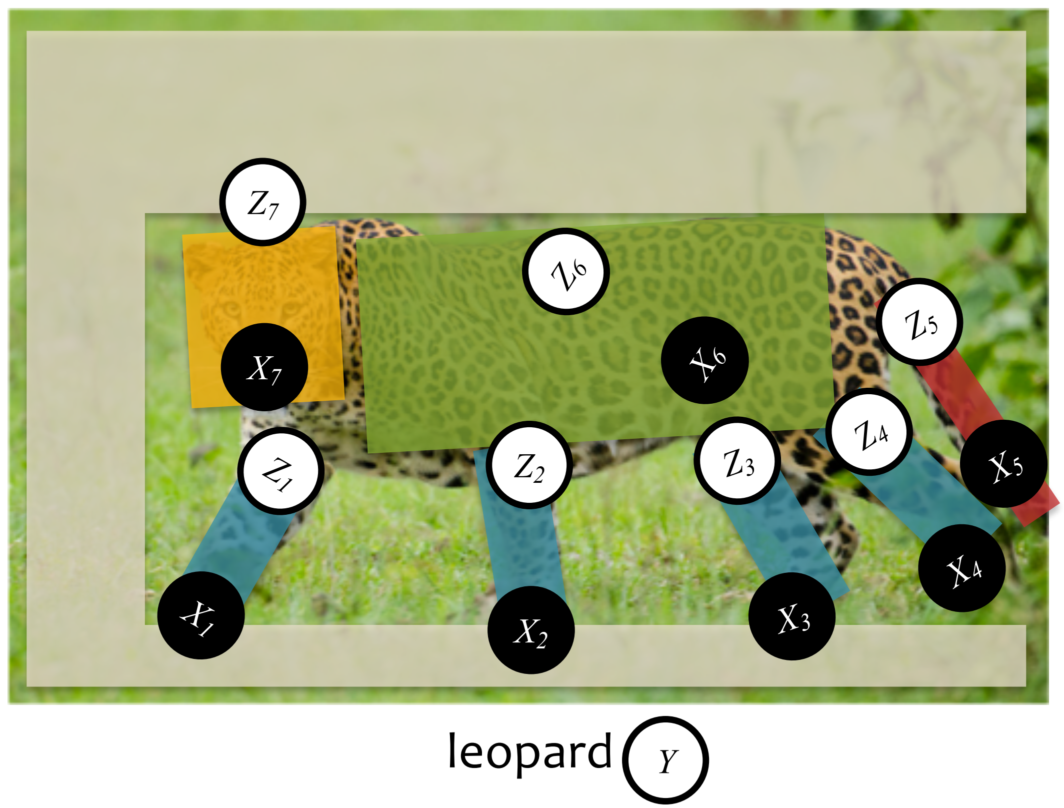
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

leopard

Case Study: Object Recognition

Data consists of images x and labels y.

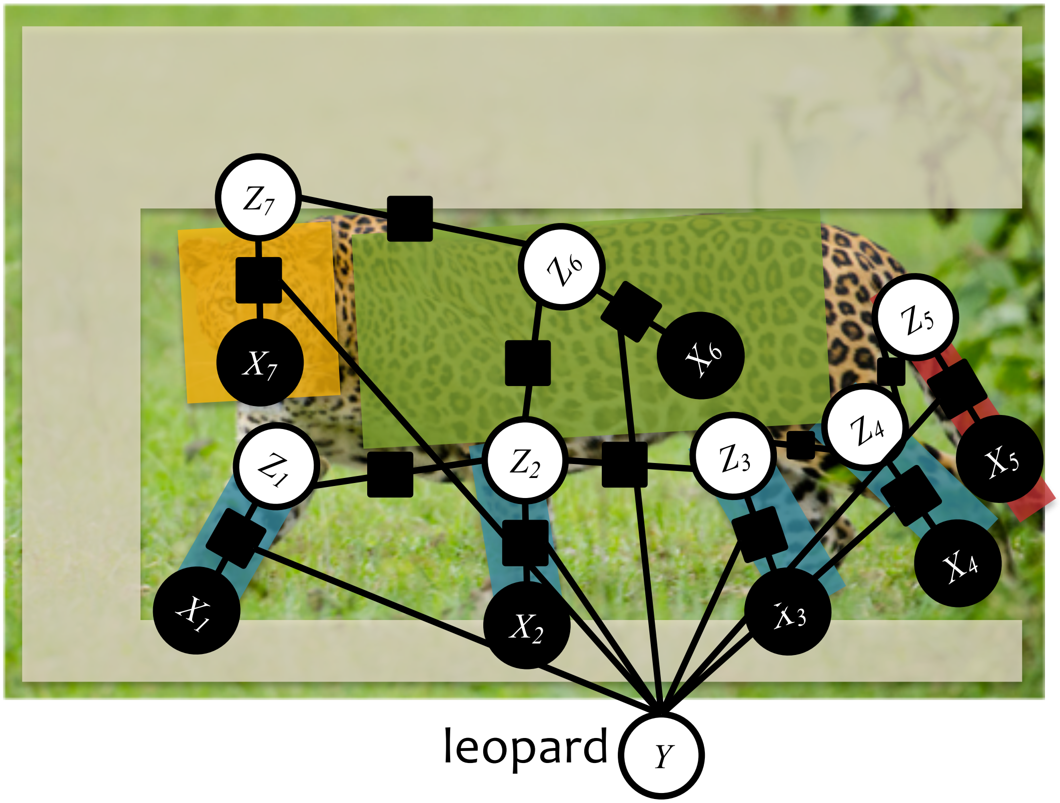
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Case Study: Object Recognition

Data consists of images x and labels y.

- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Structured Prediction

Preview of challenges to come…

- Consider the task of finding the **most probable assignment** to the output

$p(y=+1|x)$
$p(y=-1|x)$

for $y \in \mathcal{Y}$:
$p(y|x)$

Classification

$$\hat{y} = \operatorname*{argmax}_{y} p(y|\mathbf{x})$$

where $y \in \{+1, -1\}$

Structured Prediction

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y}|\mathbf{x})$$

where $\mathbf{y} \in \mathcal{Y}$

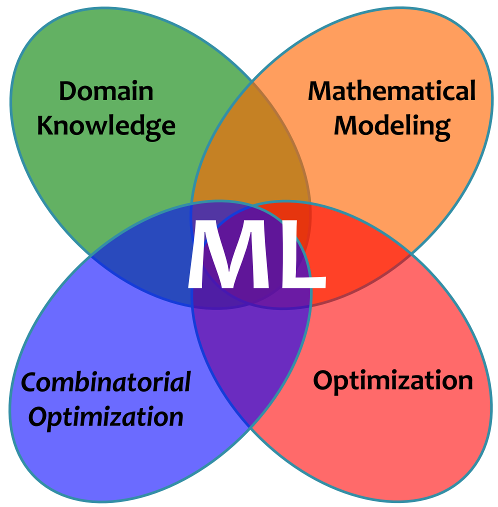
and $|\mathcal{Y}|$ is very large

Machine Learning

The **data** inspires the structures we want to predict

Our **model** defines a score for each structure

It also tells us what to optimize



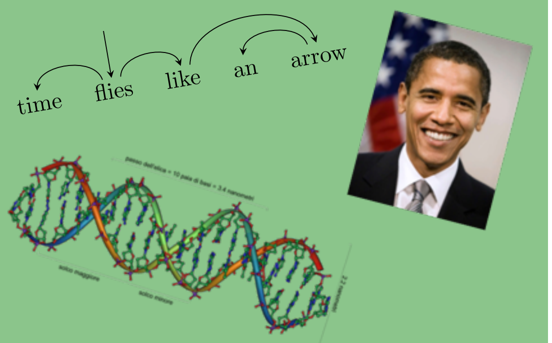
Inference finds {best structure, marginals, partition function} for a new observation

Learning tunes the parameters of the model

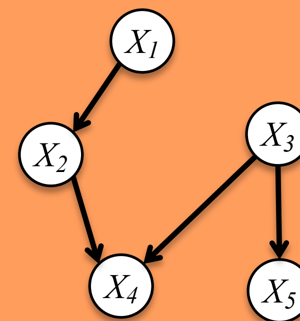
(**Inference** is usually called as a subroutine in learning)

Machine Learning

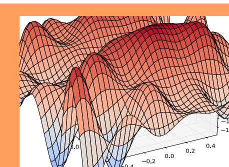
Data



Model



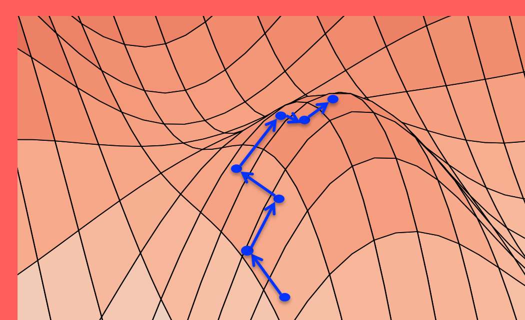
Objective



Inference

(**Inference** is usually called as a subroutine in learning)

Learning



BACKGROUND

Background: Chain Rule of Probability

For random variables A and B:

$$P(A, B) = P(A|B)P(B)$$

For random variables X_1, X_2, X_3, X_4:

$$\begin{aligned} P(X_1, X_2, X_3, X_4) = & P(X_1|X_2, X_3, X_4) \\ & P(X_2|X_3, X_4) \\ & P(X_3|X_4) \\ & P(X_4) \end{aligned}$$

Background: Conditional Independence

Random variables A and B are conditionally independent given C if:

$$P(A, B|C) = P(A|C)P(B|C) \quad (1)$$

or equivalently:

$$P(A|B, C) = P(A|C) \quad (2)$$

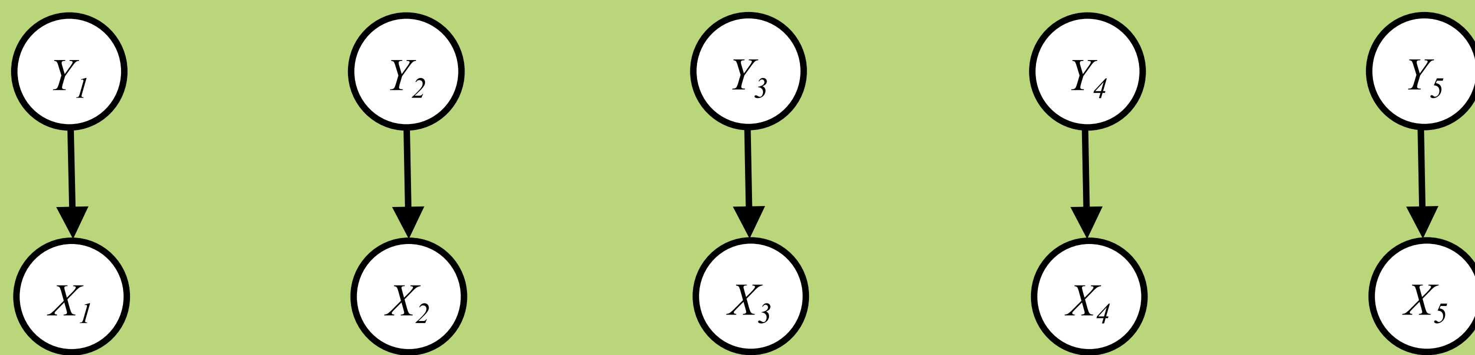
We write this as:

$$A \perp\!\!\!\perp B | C$$

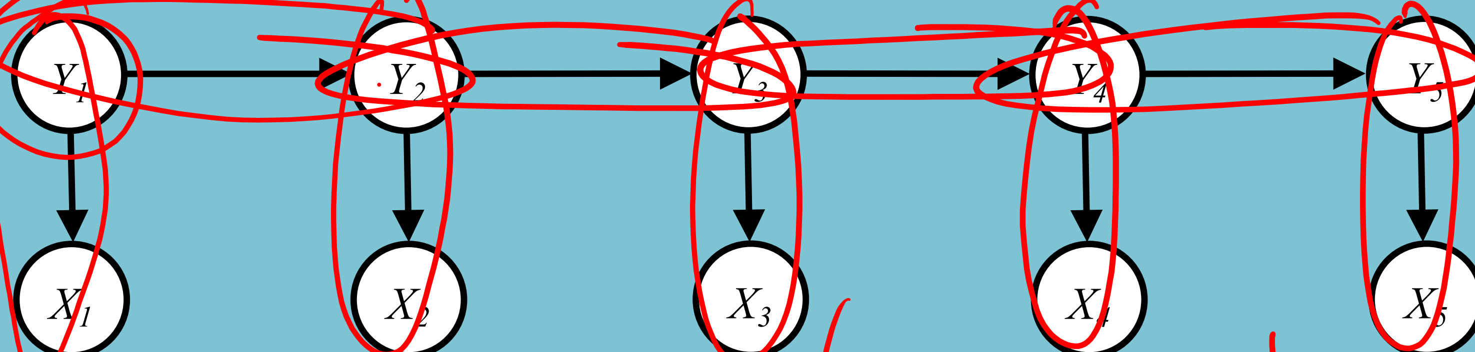
Later we will also write: $I<A, \{C\}, B>$

HIDDEN MARKOV MODEL (HMM)

From Mixture Model to HMM



"Naïve Bayes": $$P(\mathbf{X}, \mathbf{Y}) = \prod_{t=1}^{T} P(X_t|Y_t)p(Y_t)$$



HMM: $$P(\mathbf{X}, \mathbf{Y}) = P(Y_1)\left(\prod_{t=1}^{T} P(X_t|Y_t)\right)\left(\prod_{t=2}^{T} p(Y_t|Y_{t-1})\right)$$

HIDDEN MARKOV MODEL (HMM)

HMM Outline

- **Motivation**
  - Time Series Data
- **Hidden Markov Model (HMM)**
  - Example: Squirrel Hill Tunnel Closures [courtesy of Roni Rosenfeld]
  - Background: Markov Models
  - From Mixture Model to HMM
  - History of HMMs
  - Higher-order HMMs
- **Training HMMs**
  - (Supervised) Likelihood for HMM
  - Maximum Likelihood Estimation (MLE) for HMM
  - EM for HMM (aka. Baum-Welch algorithm)
- **Forward-Backward Algorithm**
  - Three Inference Problems for HMM
  - Great Ideas in ML: Message Passing
  - Example: Forward-Backward on 3-word Sentence
  - Derivation of Forward Algorithm
  - Forward-Backward Algorithm
  - Viterbi algorithm

Markov Models

Whiteboard

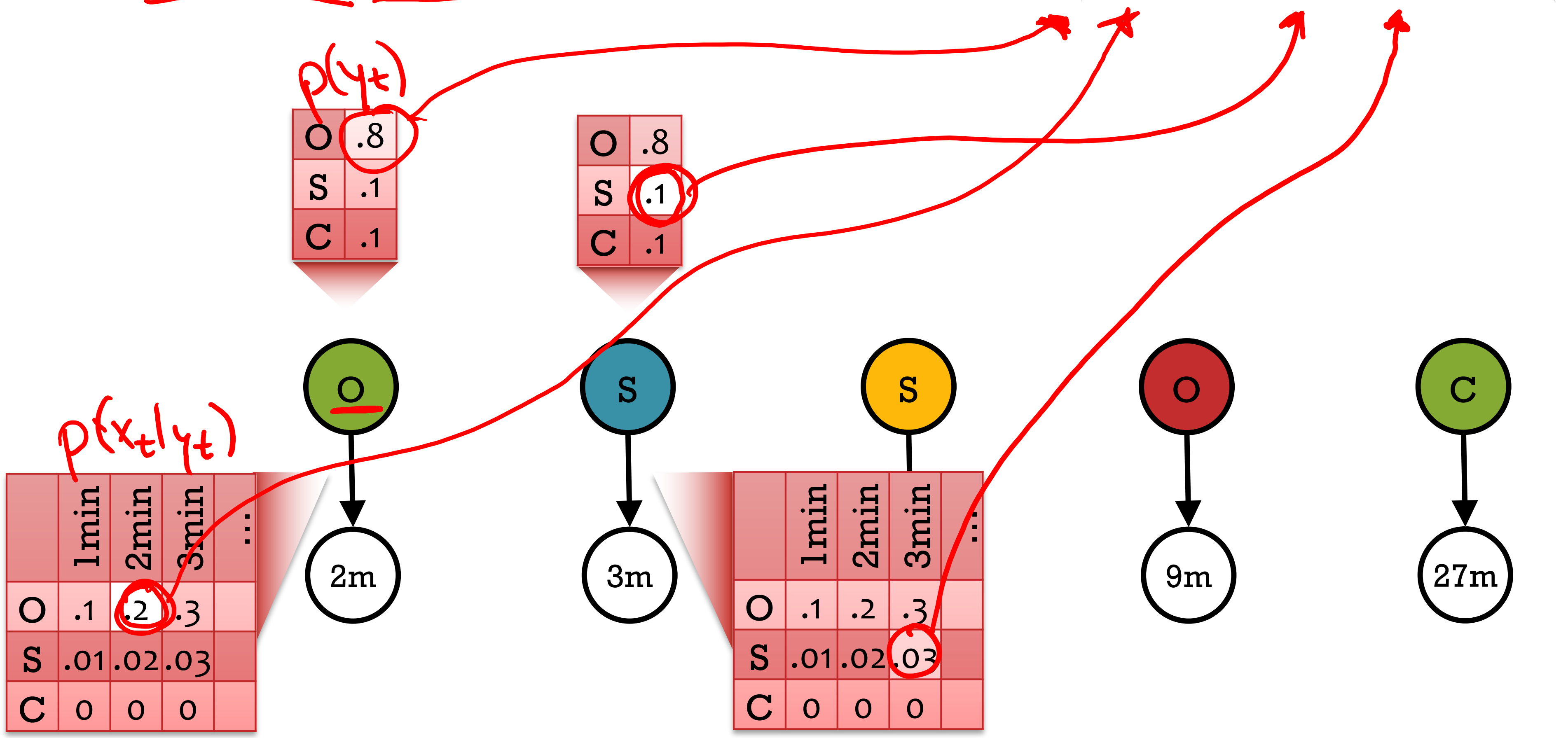
- Example: Tunnel Closures [courtesy of Roni Rosenfeld]
- First-order Markov assumption
- Conditional independence assumptions


SQUIRREL HILL SOUTH
Covode St
Nicholson St
Phillips Ave
Melvin St
Taylor Allderdice High School
Crombie St
Tilbury Ave
Beechwood Blvd
English Ln
Mt Royal Rd
Forward Ave
Alderson St
376
Beechwood Blvd
Flemington St
Mirror St
Inez Way
Burchfield Ave
Ira Way
Anita Ave
Monitor St
Kennebec St
Windsor St
Murray Ave
Saline St
Shady Ave
Ebdy St
Caton St
Lilac St
Fernwald Rd
Summerset Dr
Nine Mile Run
Rosemoor St
Landview Rd
Rosemont Ln
Melbourne St
Map data ©2018 Google
US Foods

Mixture Model for Time Series Data

We could treat each (tunnel state, travel time) pair as independent. This corresponds to a Naïve Bayes model with a single feature (travel time).

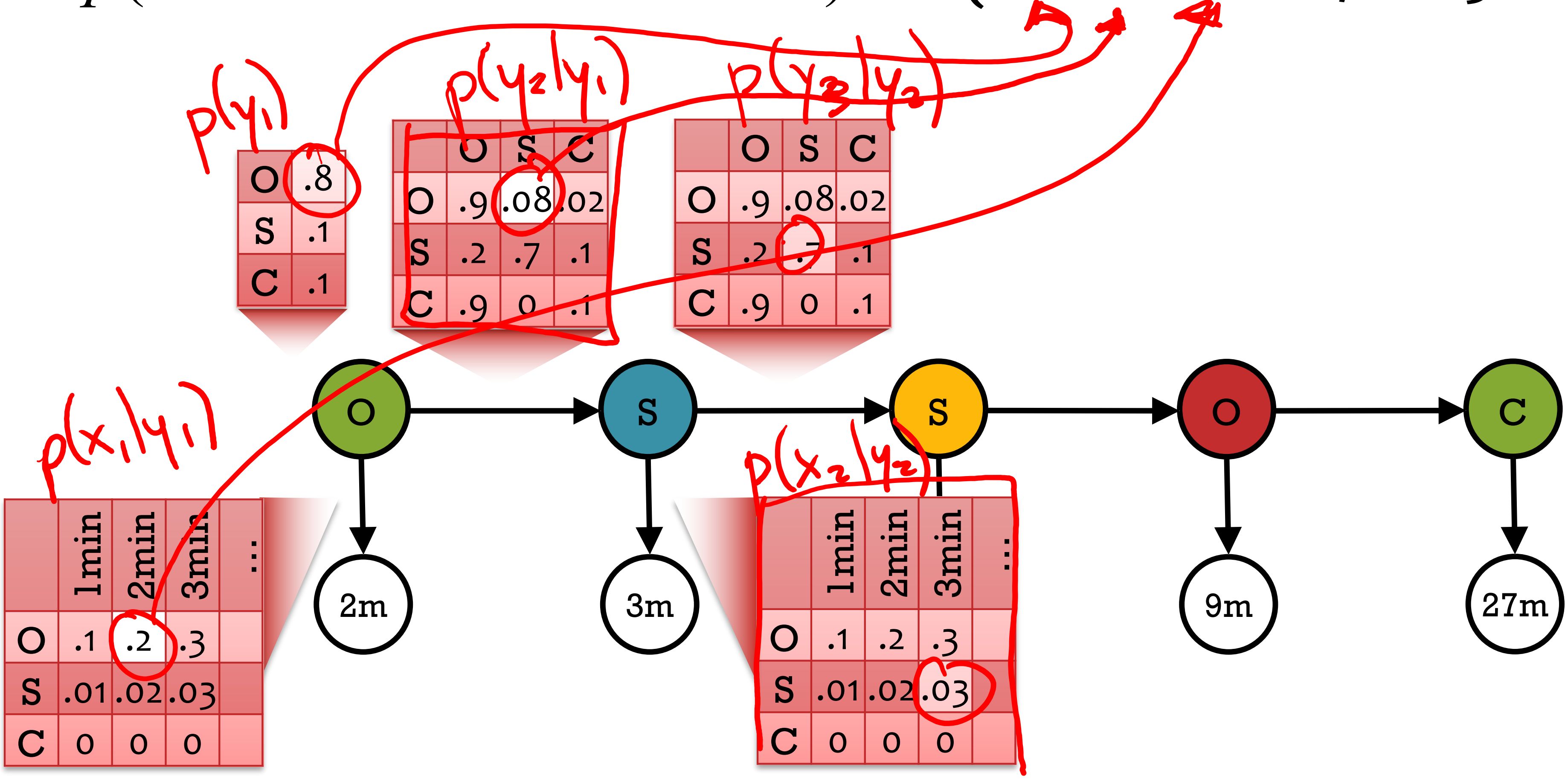
$$p(\text{O, S, S, O, C, 2m, 3m, 18m, 9m, 27m}) = (.8 * .2 * .1 * .03 * \ldots)$$



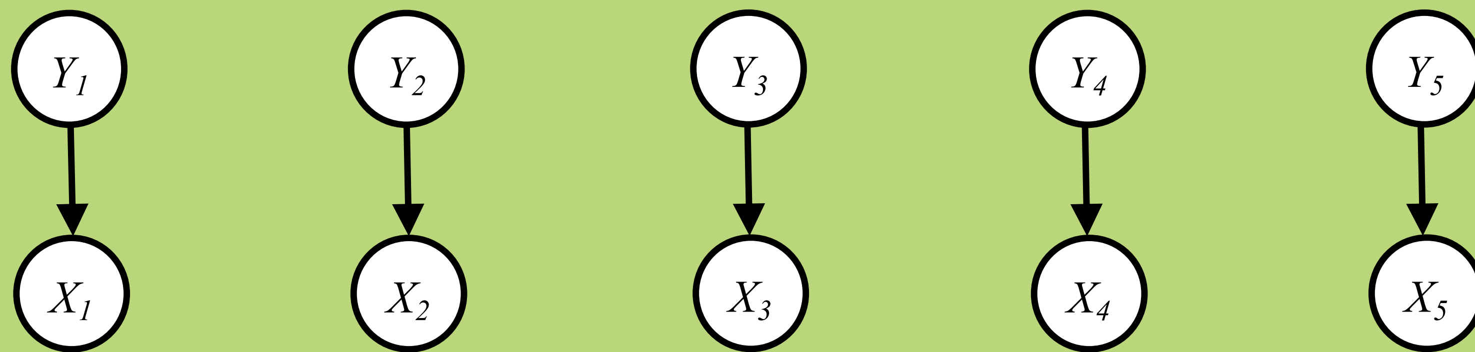
Hidden Markov Model

A Hidden Markov Model (HMM) provides a joint distribution over the the tunnel states / travel times with an assumption of dependence between adjacent tunnel states.

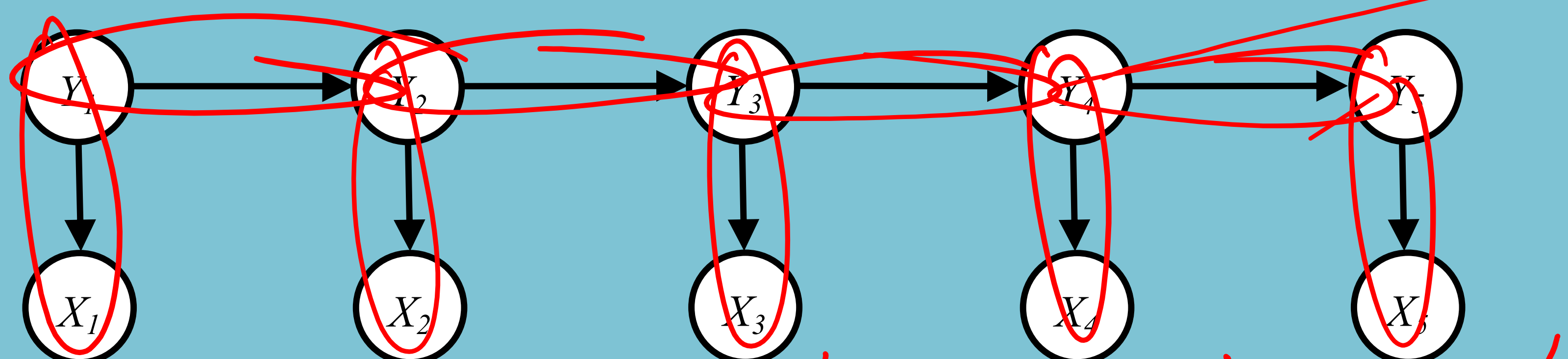
$$p(\text{O, S, S, O, C, 2m, 3m, 18m, 9m, 27m}) = (.8 * .08 * .2 * .7 * .03 * \ldots)$$



From Mixture Model to HMM

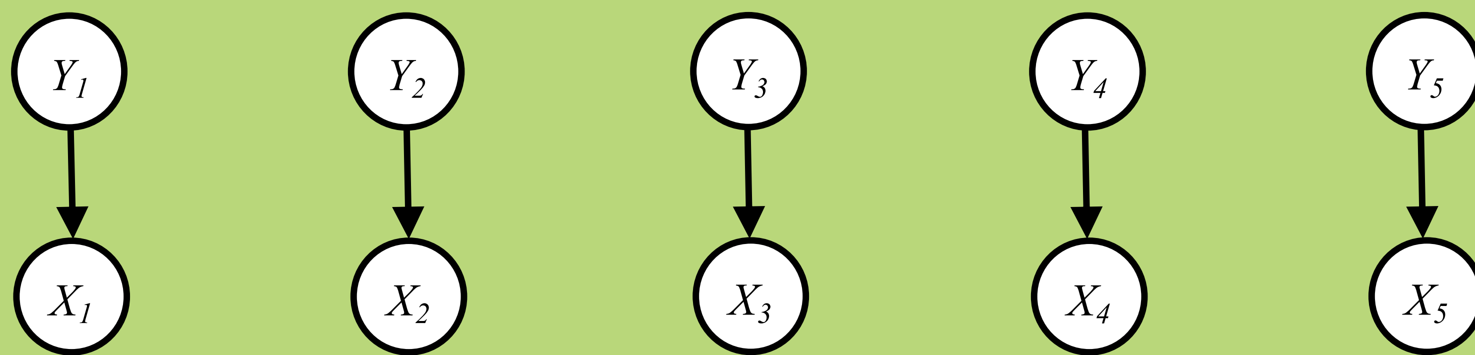


"Naïve Bayes": $$P(\mathbf{X}, \mathbf{Y}) = \prod_{t=1}^{T} P(X_t|Y_t)p(Y_t)$$

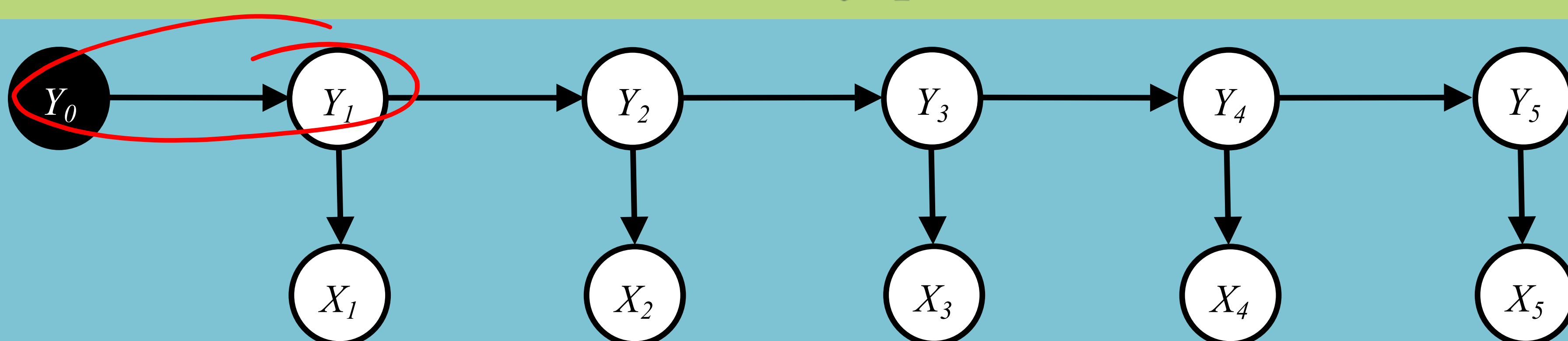


HMM: $$P(\mathbf{X}, \mathbf{Y}) = P(Y_1) \left(\prod_{t=1}^{T} P(X_t|Y_t) \right) \left(\prod_{t=2}^{T} p(Y_t|Y_{t-1}) \right)$$

From Mixture Model to HMM



"Naïve Bayes": $$P(\mathbf{X}, \mathbf{Y}) = \prod_{t=1}^{T} P(X_t | Y_t) p(Y_t)$$



HMM: $$P(\mathbf{X}, \mathbf{Y} | Y_0) = \prod_{t=1}^{T} P(X_t | Y_t) p(Y_t | Y_{t-1})$$

SUPERVISED LEARNING FOR HMMS

Recipe for Closed-form MLE

1. Assume data was generated i.i.d. from some model (i.e. write the generative story) HMM

   $x^{(i)} \sim p(x|\boldsymbol{\theta})$

2. Write log-likelihood

   $\ell(\boldsymbol{\theta}) = \log p(x^{(1)}|\boldsymbol{\theta}) + \ldots + \log p(x^{(N)}|\boldsymbol{\theta})$

3. Compute partial derivatives (i.e. gradient)

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_1 = \ldots$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_2 = \ldots$

   $\ldots$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_M = \ldots$

4. Set derivatives to zero and solve for $\boldsymbol{\theta}$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_m = 0$ for all $m \in \{1, \ldots, M\}$

   $\boldsymbol{\theta}^{MLE}$ = solution to system of M equations and M variables

5. Compute the second derivative and check that $\ell(\boldsymbol{\theta})$ is concave down at $\boldsymbol{\theta}^{MLE}$

MLE of Categorical Distribution

1. Suppose we have a **dataset** obtained by repeatedly rolling a M-sided (weighted) die N times. That is, we have data

$$\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$$

vector

where $x^{(i)} \in \{1, \ldots, M\}$ and $x^{(i)} \sim \text{Categorical}(\phi)$.

2. A random variable is **Categorical** written $X \sim \text{Categorical}(\phi)$ iff

$$P(X = x) = p(x; \phi) = \phi_x$$

where $x \in \{1, \ldots, M\}$ and $\sum_{m=1}^{M} \phi_m = 1$. The **log-likelihood** of the data becomes:

$\phi_m \geq 0$

$$\ell(\phi) = \sum_{i=1}^{N} \log \phi_{x^{(i)}} \text{ s.t. } \sum_{m=1}^{M} \phi_m = 1$$

3. Solving this *constrained* optimization problem yields the **maximum likelihood estimator** (MLE):

$$\phi_m^{MLE} = \frac{N_{x=m}}{N} = \frac{\sum_{i=1}^{N} \mathbb{I}(x^{(i)} = m)}{N}$$

Hidden Markov Model

HMM Parameters:

Emission matrix, $\mathbf{A}$, where $P(X_t = k | Y_t = j) = A_{j,k}, \forall t, k$

Transition matrix, $\mathbf{B}$, where $P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k$

Initial probs, $\mathbf{C}$, where $P(Y_1 = k) = C_k, \forall k$

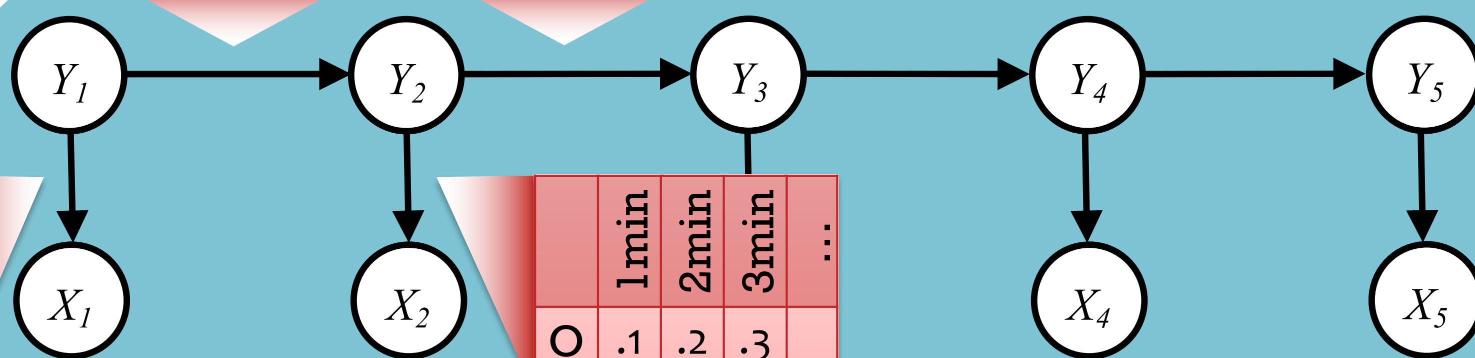
C

O	.8
S	.1
C	.1

B

	O	S	C
O	.9	.08	.02
S	.2	.7	.1
C	.9	0	.1

	O	S	C
O	.9	.08	.02
S	.2	.7	.1
C	.9	0	.1



A

	1min	2min	3min	…
O	.1	.2	.3	
S	.01	.02	.03	
C	0	0	0	

	1min	2min	3min	…
O	.1	.2	.3	
S	.01	.02	.03	
C	0	0	0	

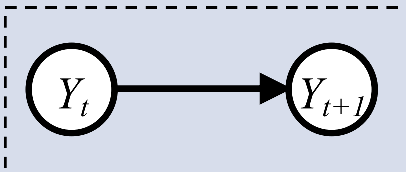
Training HMMs

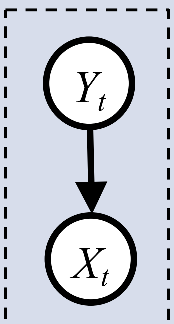
Whiteboard

- (Supervised) Likelihood for an HMM
- Maximum Likelihood Estimation (MLE) for HMM

Supervised Learning for HMMs

Learning an HMM decomposes into solving two (independent) Mixture Models





Data: $\mathcal{D} = \{(\vec{x}^{(i)}, \vec{y}^{(i)})\}_{i=1}^{N}$ $\quad \vec{x} = [x_1, \ldots, x_T]^T$, $\vec{y} = [y_1, \ldots, y_T]^T$

Likelihood:

$$\ell(A,B,C) = \sum_{i=1}^{N} \log p(\vec{x}^{(i)}, y^{(i)} \mid A,B,C)$$

$$= \sum_{i=1}^{N} \Big[\underbrace{\log p(y_1^{(i)} \mid C)}_{\text{initial}} + \underbrace{\Big(\sum_{t=2}^{T} \log p(y_t^{(i)} \mid y_{t-1}^{(i)}, B)\Big)}_{\text{transition}} + \underbrace{\Big(\sum_{t=1}^{T} \log p(x_t^{(i)} \mid y_t^{(i)}, A)\Big)}_{\text{emission}} \Big]$$

MLE:

$$\hat{A}, \hat{B}, \hat{C} = \underset{A,B,C}{\operatorname{argmax}}\; \ell(A,B,C)$$

$$\Rightarrow \hat{C} = \underset{C}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(y_1^{(i)} \mid C)$$

$$\hat{B} = \underset{B}{\operatorname{argmax}} \sum_{i=1}^{N} \sum_{t=2}^{T} \log p(y_t^{(i)} \mid y_{t-1}^{(i)}, B)$$

$$\hat{A} = \underset{A}{\operatorname{argmax}} \sum_{i=1}^{N} \sum_{t=1}^{T} \log p(x_t^{(i)} \mid y_t^{(i)}, A)$$

Can solve in closed form, which yields...

$$\hat{C}_k = \frac{\#(y_1^{(i)} = k)}{N} \quad \forall i,k$$

$$\hat{B}_{jk} = \frac{\#(y_t^{(i)} = k \text{ and } y_{t-1}^{(i)} = j)}{\#(y_{t-1}^{(i)} = j)} \quad \forall i, t>1, j, k$$

$$\hat{A}_{jk} = \frac{\#(x_t^{(i)} = k \text{ and } y_t^{(i)} = j)}{\#(y_t^{(i)} = j)} \quad \forall i, t, j, k$$

Hidden Markov Model

HMM Parameters:

Emission matrix, $\mathbf{A}$, where $P(X_t = k | Y_t = j) = A_{j,k}, \forall t, k$

Transition matrix, $\mathbf{B}$, where $P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k$

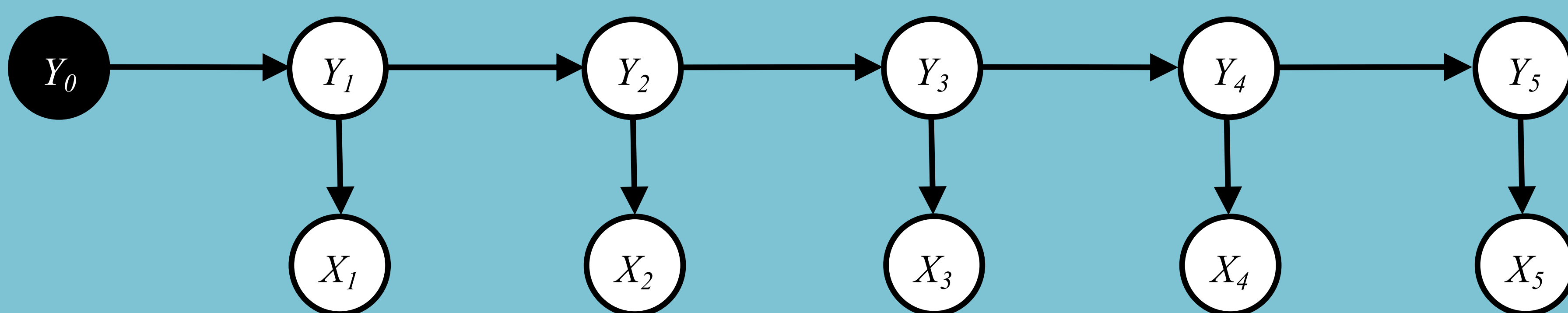
Assumption: $y_0 = \text{START}$

Generative Story:

$$Y_t \sim \text{Multinomial}(\mathbf{B}_{Y_{t-1}}) \ \forall t$$

$$X_t \sim \text{Multinomial}(\mathbf{A}_{Y_t}) \ \forall t$$

For notational convenience, we fold the *initial probabilities* **C** into the *transition matrix* **B** by our assumption.



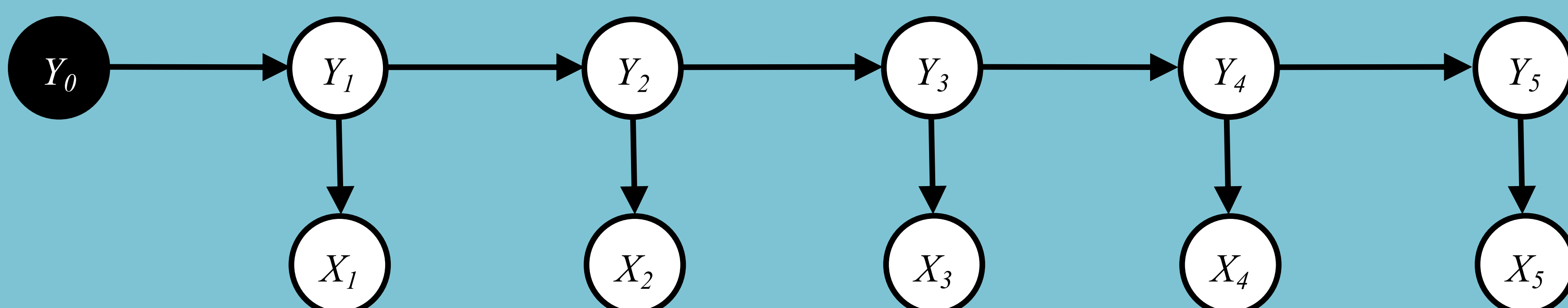
Hidden Markov Model

Joint Distribution:

$$y_0 = \text{START}$$

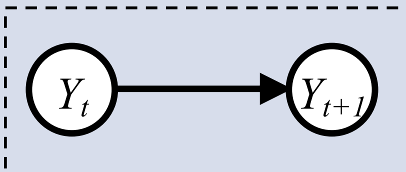
$$p(\mathbf{x}, \mathbf{y} | y_0) = \prod_{t=1}^{T} p(x_t | y_t) p(y_t | y_{t-1})$$

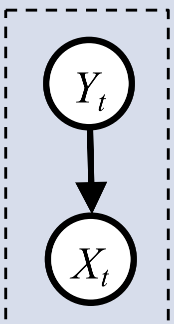
$$= \prod_{t=1}^{T} A_{y_t, x_t} B_{y_{t-1}, y_t}$$



Supervised Learning for HMMs

Learning an HMM decomposes into solving two (independent) Mixture Models





$$\mathcal{D} = \{(\vec{x}^{(i)}, \vec{y}^{(i)})\}_{i=1}^{N}$$

Likelihood:

$$\ell(A,B) = \sum_{i=1}^{N} \log p(\vec{x}^{(i)}, \vec{y}^{(i)})$$

$$= \sum_{i=1}^{N} \left[\sum_{t=1}^{T} \log p(y_t^{(i)} | y_{t-1}^{(i)}, B) + \log p(x_t^{(i)} | y_t^{(i)}, A) \right]$$

MLE:

$$\hat{A}, \hat{B} = \operatorname{argmax} \ \ell(A,B)$$

$$\hat{A} = \operatorname{argmax} \sum_{i=1}^{N} \left[\sum_{t=1}^{T} \log p(x_t^{(i)} | y_t^{(i)}, A) \right]$$

$$\hat{B} = \operatorname{argmax} \sum_{i=1}^{N} \left[\sum_{t=1}^{T} \log p(y_t^{(i)} | y_{t-1}^{(i)}, B) \right]$$

↑ can solve in closed form to get...

$$\hat{B}_{jk} = \frac{\#(y_t^{(i)} = k \text{ and } y_{t-1}^{(i)} = j)}{\#(y_{t-1}^{(i)} = j)}$$

$$\hat{A}_{jk} = \frac{\#(x_t^{(i)} = k \text{ and } y_t^{(i)} = j)}{\#(y_t^{(i)} = j)}$$