10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Hidden Markov Models

Matt Gormley
Lecture 22
April 2, 2018

Reminders

- **Homework 6: PAC Learning / Generative Models**
  - Out: Wed, Mar 28
  - Due: Wed, Apr 04 at 11:59pm
- **Homework 7: HMMs**
  - Out: Wed, Apr 04
  - Due: Mon, Apr 16 at 11:59pm

DISCRIMINATIVE AND GENERATIVE CLASSIFIERS

Generative vs. Discriminative

- **Generative Classifiers:**
  - Example: Naïve Bayes
  - Define a joint model of the observations **x** and the labels y: $p(\boldsymbol{x}, y)$
  - Learning maximizes (joint) likelihood
  - Use Bayes' Rule to classify based on the posterior:

$$p(y|\mathbf{x}) = p(\mathbf{x}|y)p(y)/p(\mathbf{x})$$

- **Discriminative Classifiers:**
  - Example: Logistic Regression
  - Directly model the conditional: $p(y|\mathbf{x})$
  - Learning maximizes conditional likelihood

Generative vs. Discriminative

Whiteboard

– Contrast: To model p(x) or not to model p(x)?

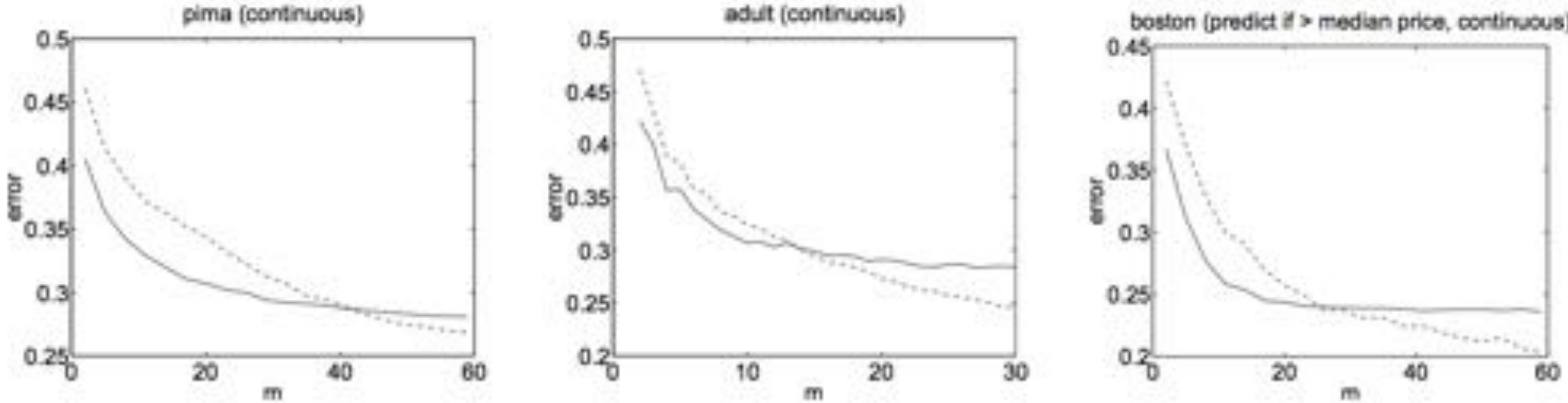
Generative vs. Discriminative

Finite Sample Analysis (Ng & Jordan, 2002)

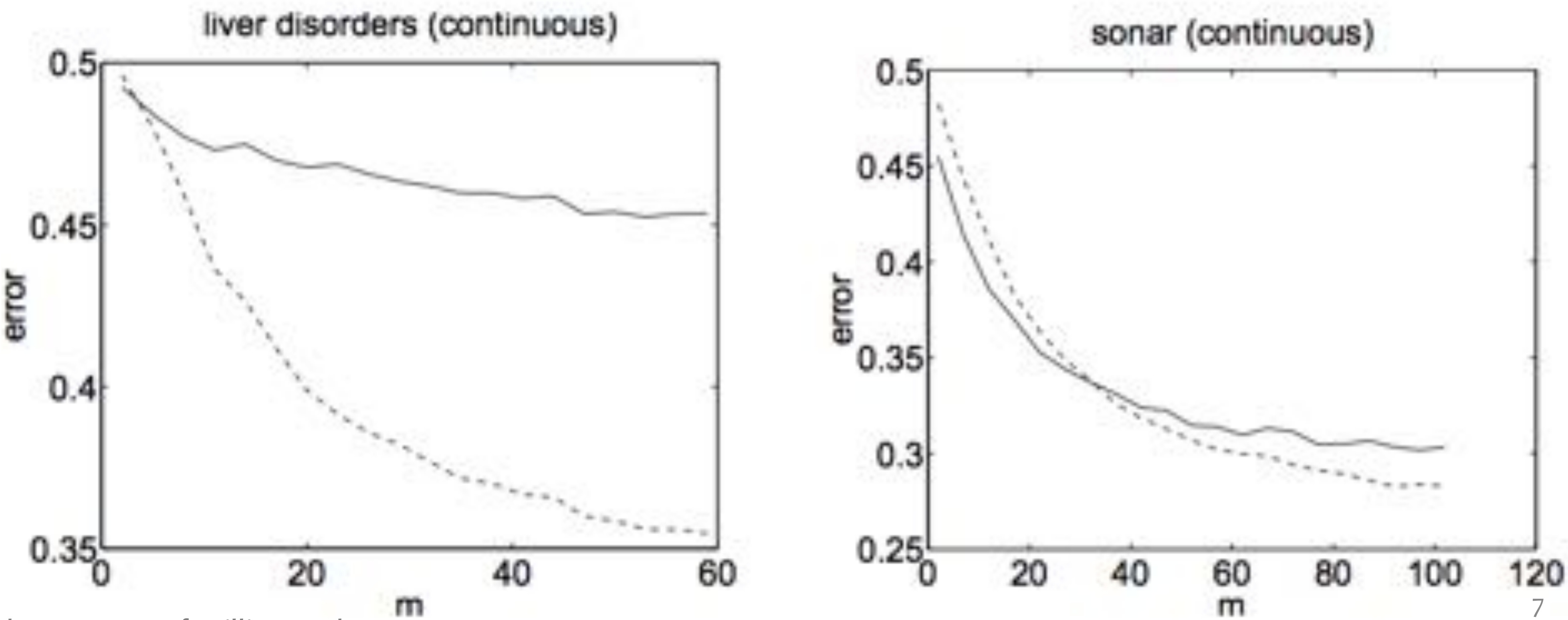
[Assume that we are learning from a finite training dataset]

If model assumptions are correct: Naive Bayes is a more efficient learner (requires fewer samples) than Logistic Regression

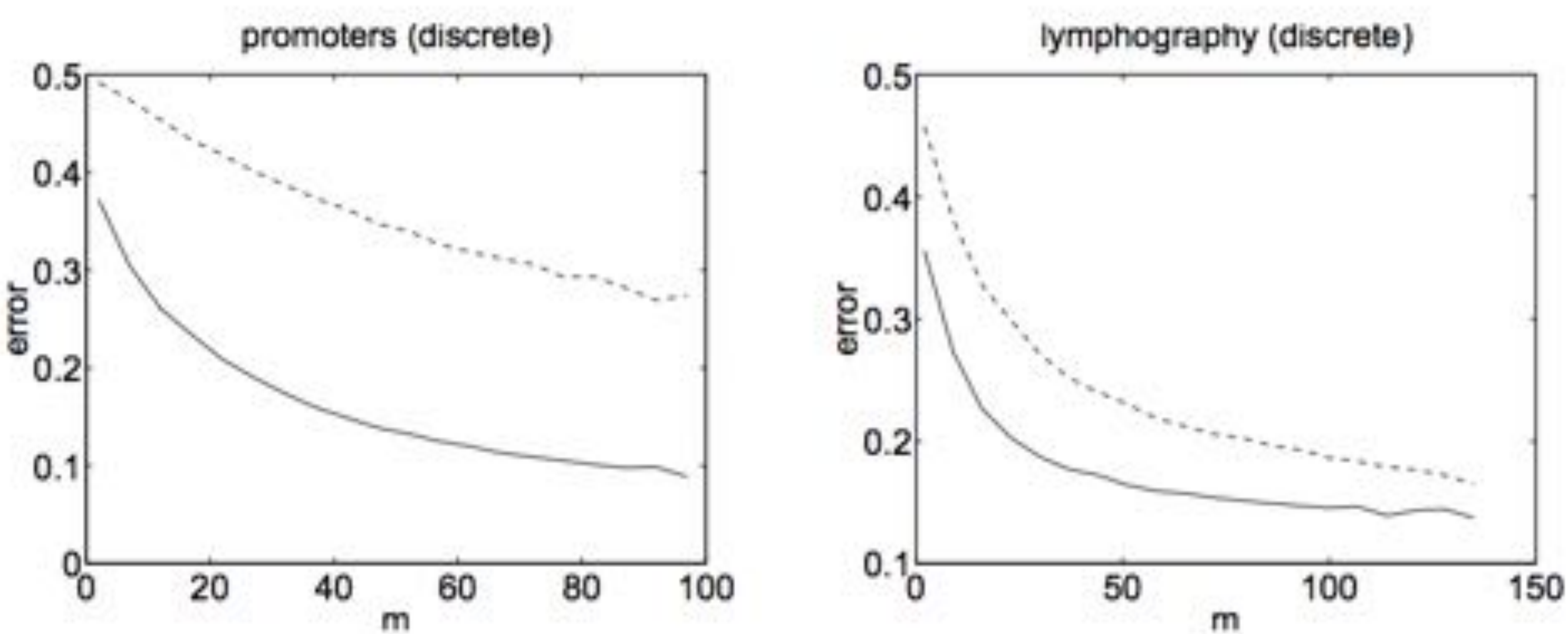
If model assumptions are incorrect: Logistic Regression has lower asymtotic error, and does better than Naïve Bayes



solid: NB dashed: LR



solid: NB dashed: LR



Naïve Bayes makes stronger assumptions about the data but needs fewer examples to estimate the parameters

"On Discriminative vs Generative Classifiers: …." Andrew Ng and Michael Jordan, NIPS 2001.

Slide courtesy of William Cohen

Generative vs. Discriminative

Learning (Parameter Estimation)

Naïve Bayes:
Parameters are decoupled → Closed form solution for MLE

Logistic Regression:
Parameters are coupled → No closed form solution – must use iterative optimization techniques instead

Naïve Bayes vs. Logistic Reg.

Learning (MAP Estimation of Parameters)

Bernoulli Naïve Bayes:
Parameters are probabilities → Beta prior (usually) pushes probabilities away from zero / one extremes

Logistic Regression:
Parameters are not probabilities → Gaussian prior encourages parameters to be close to zero

(effectively pushes the probabilities away from zero / one extremes)

Naïve Bayes vs. Logistic Reg.

Features

Naïve Bayes:
Features $\boldsymbol{x}$ are assumed to be conditionally independent given y. (i.e. Naïve Bayes Assumption)

Logistic Regression:
No assumptions are made about the form of the features $\boldsymbol{x}$. They can be dependent and correlated in any fashion.

MOTIVATION: STRUCTURED PREDICTION

Structured Prediction

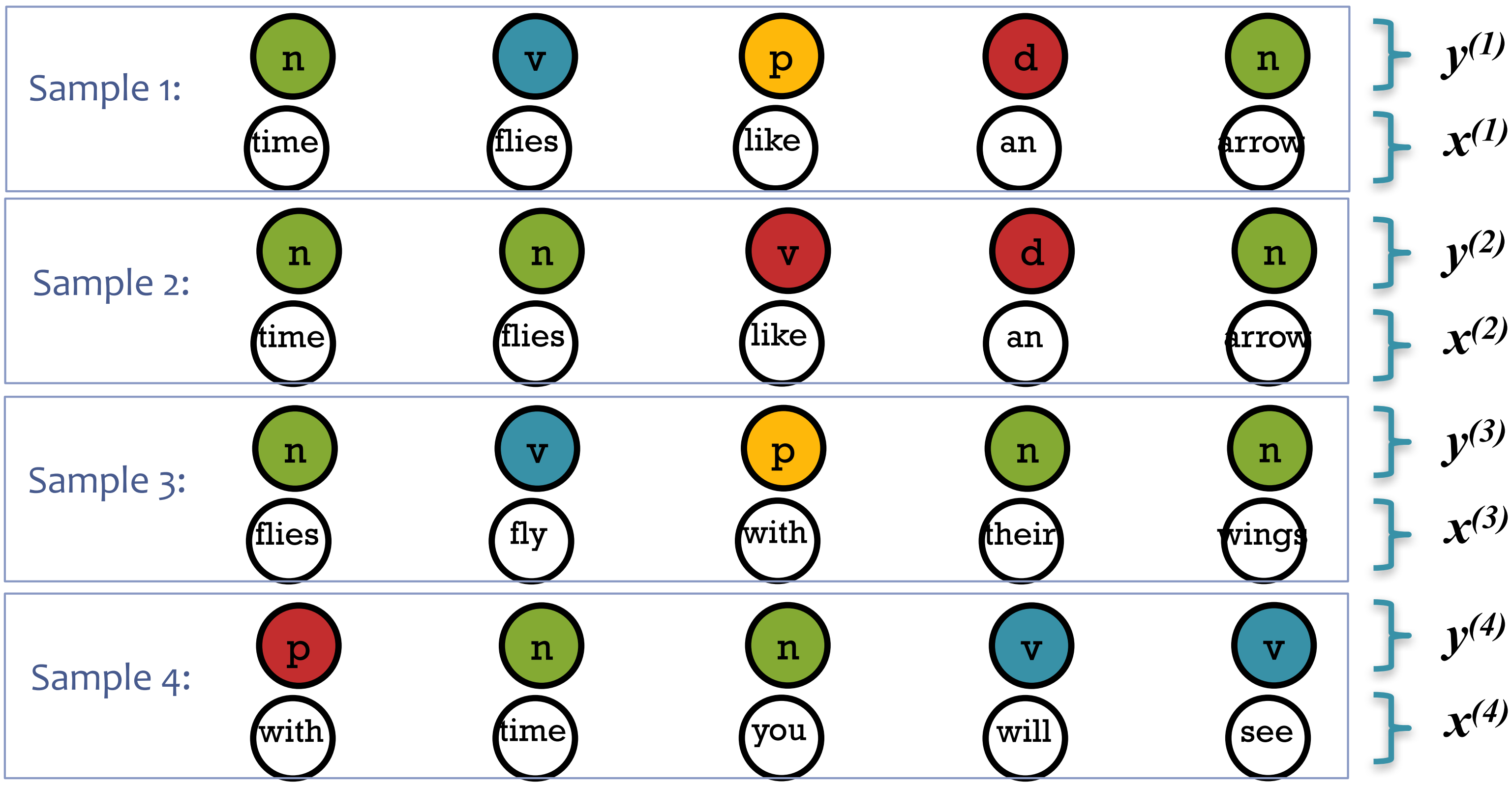
- Most of the models we've seen so far were for **classification**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  - Predict a (binary) **label:** y
- Many real-world problems require **structured prediction**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  -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

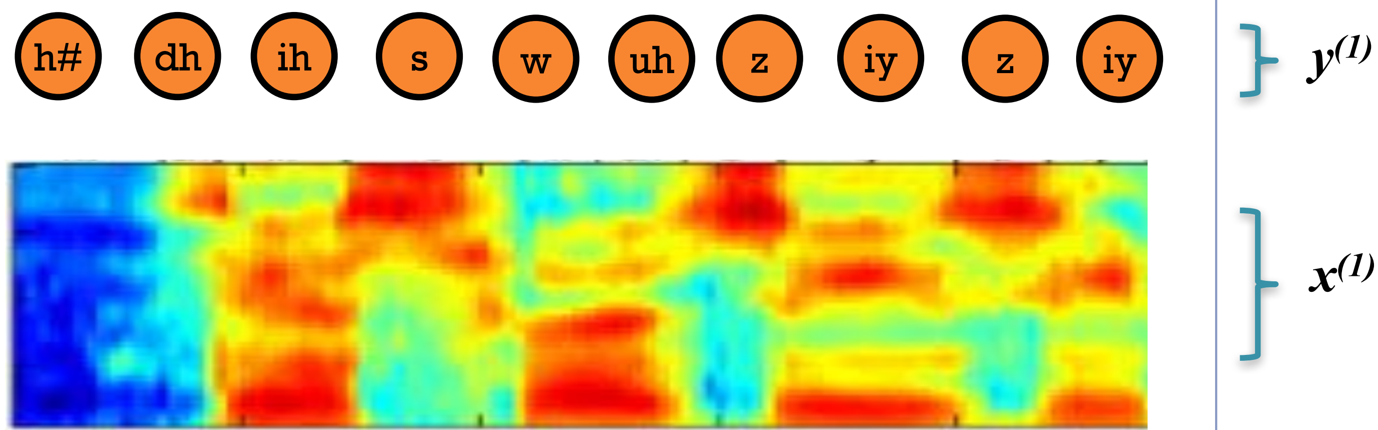
Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Figures from (Chatzis & Demiris, 2013)

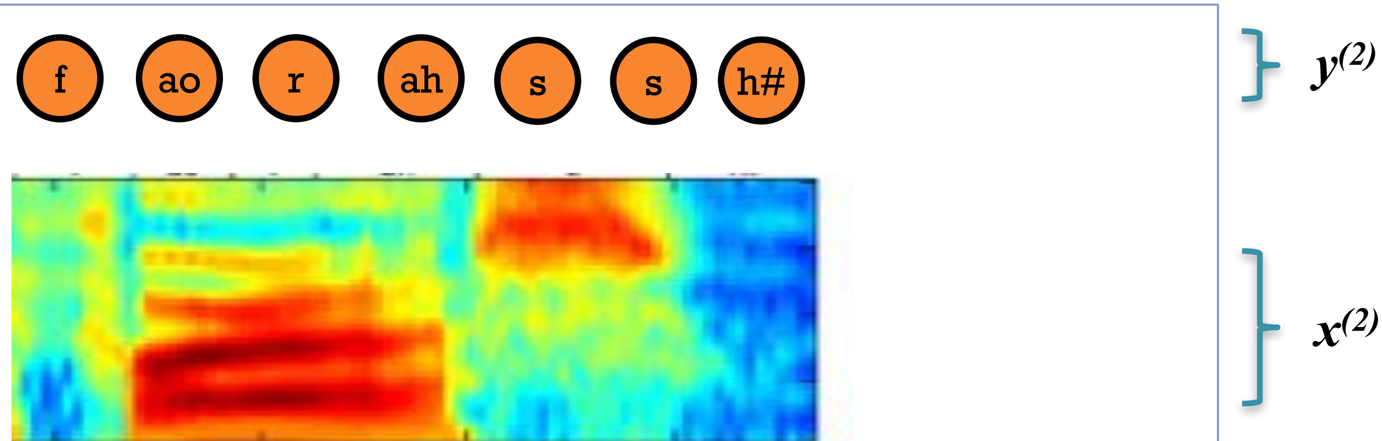
Dataset for Supervised Phoneme (Speech) Recognition

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

Sample 1:



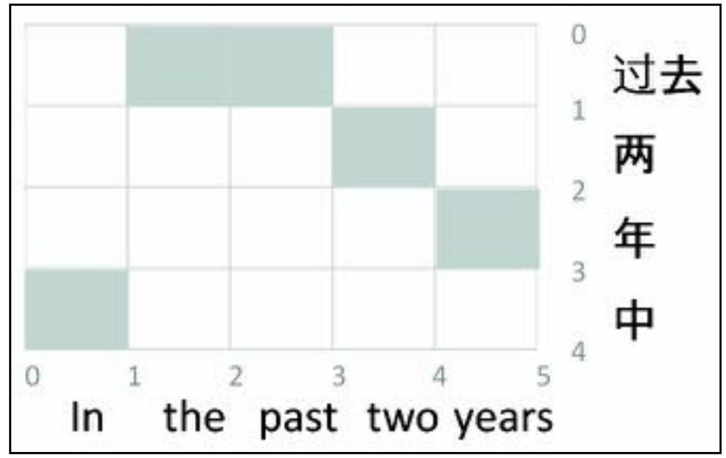
Sample 2:

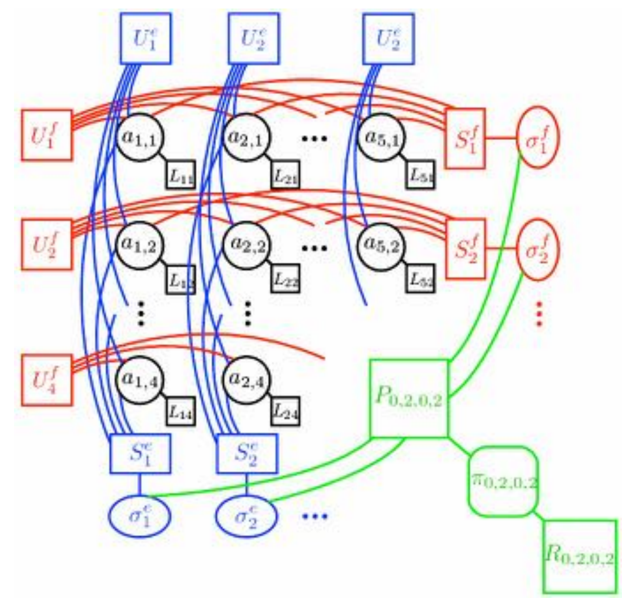


Figures from (Jansen & Niyogi, 2013)

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 (?)

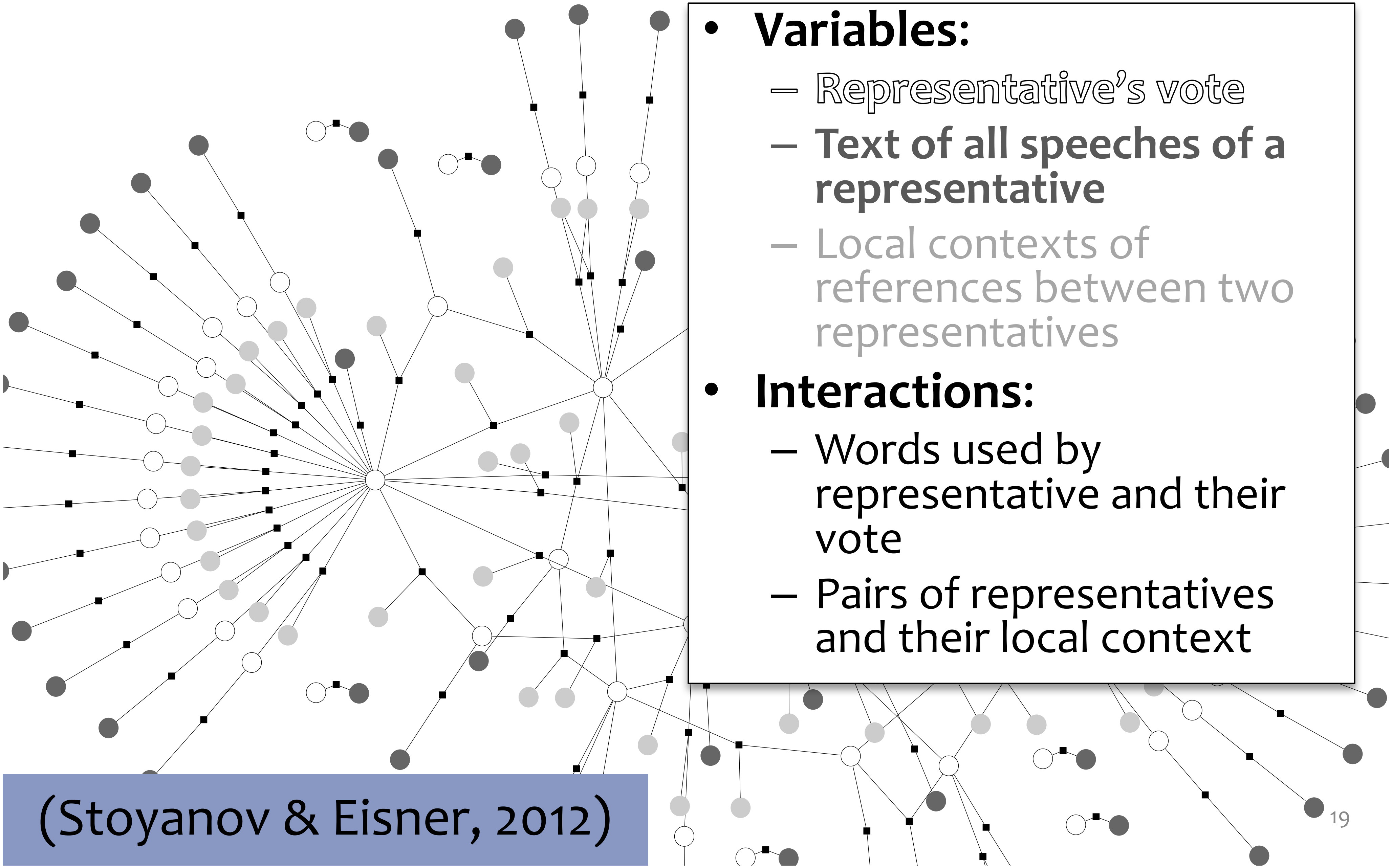




(Burkett & Klein, 2012)

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.

$\boldsymbol{x}^{(1)}$

pigeon $y^{(1)}$

$\boldsymbol{x}^{(2)}$

rhinoceros $y^{(2)}$

$\boldsymbol{x}^{(3)}$

leopard $y^{(3)}$

$\boldsymbol{x}^{(4)}$

llama $y^{(4)}$

Case Study: Object Recognition

Data consists of images x and labels y.

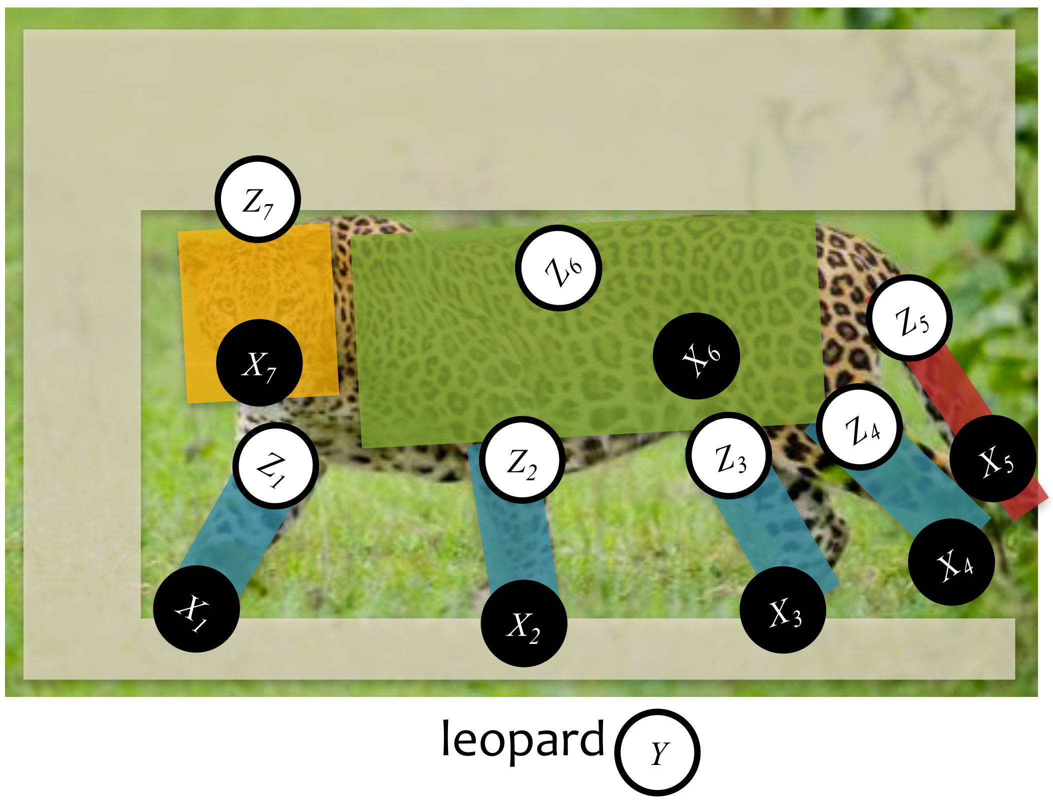
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

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

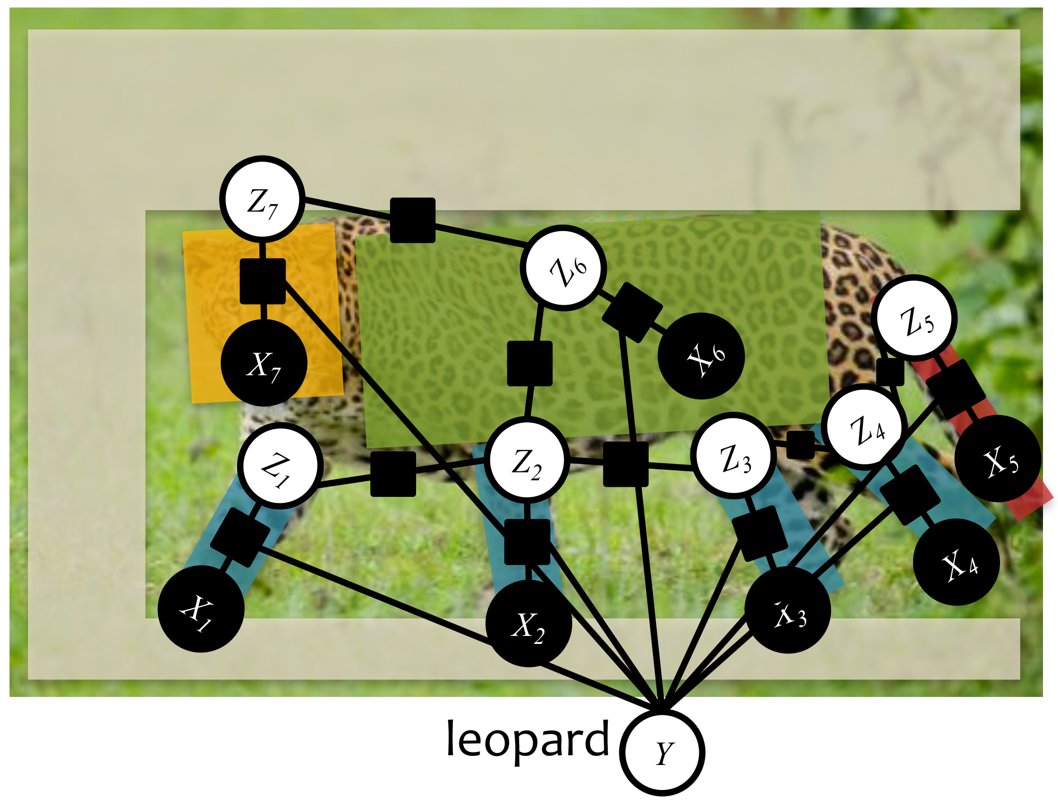


Case Study: Object Recognition

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

Classification

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

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

Structured Prediction

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

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

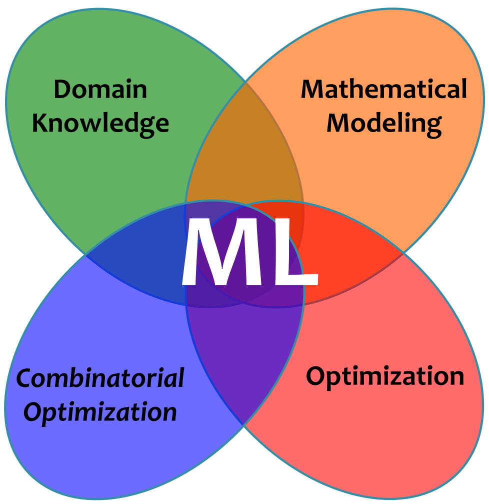
and $|\mathcal{Y}|$ is very large

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



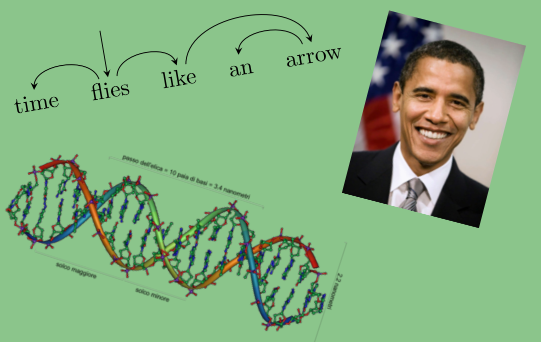
Inference finds {best structure, marginals, partition function} for a new observation

Learning tunes the parameters of the model

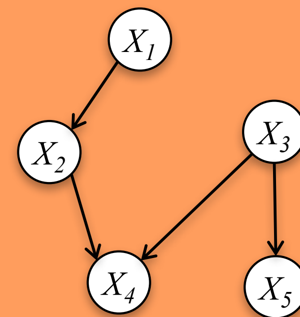
(**Inference** is usually called as a subroutine in learning)

Machine Learning

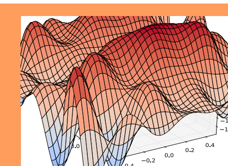
Data



Model



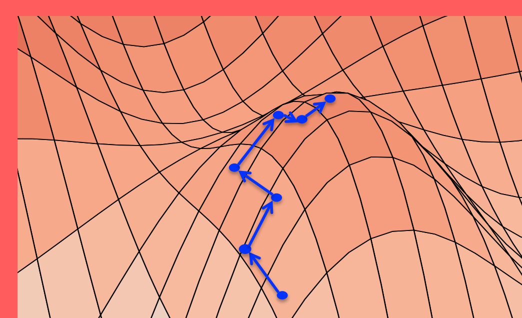
Objective



Inference

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

Learning



BACKGROUND

Background

Whiteboard

- Chain Rule of Probability
- Conditional Independence

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:

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

or equivalently:

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

We write this as:

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

Later we will also write: $I\langle A, \{C\}, B\rangle$

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

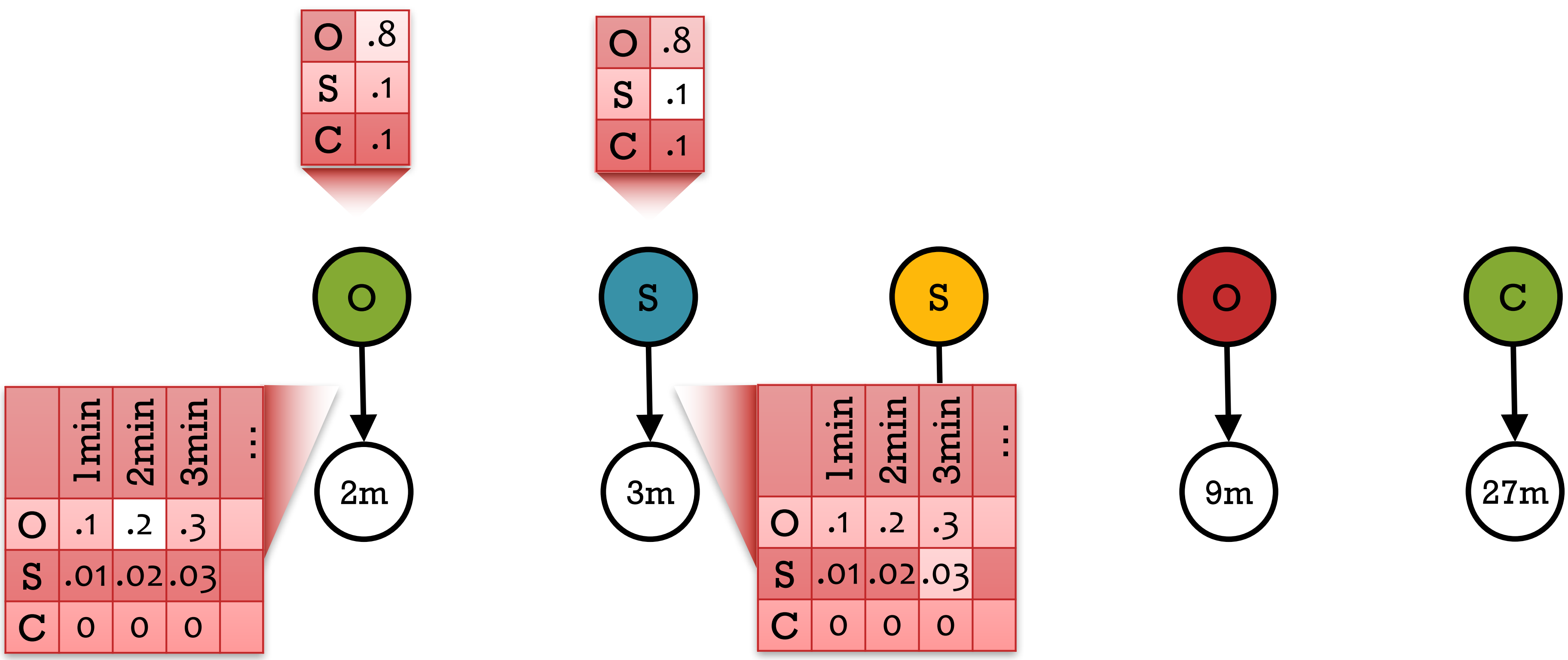
Whiteboard

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

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).

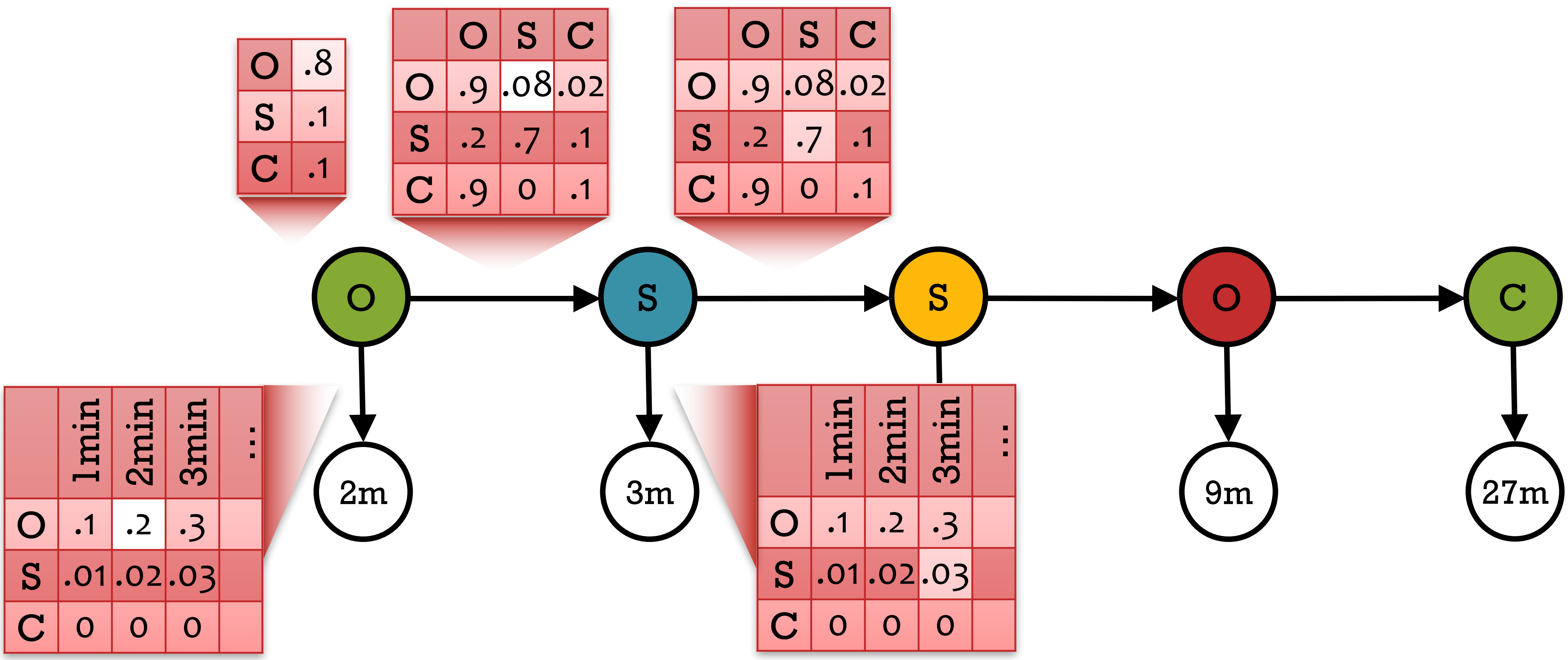
$$p(\mathrm{O, S, S, O, C, 2m, 3m, 18m, 9m, 27m}) = (.8 * .2 * .1 * .03 * \ldots)$$



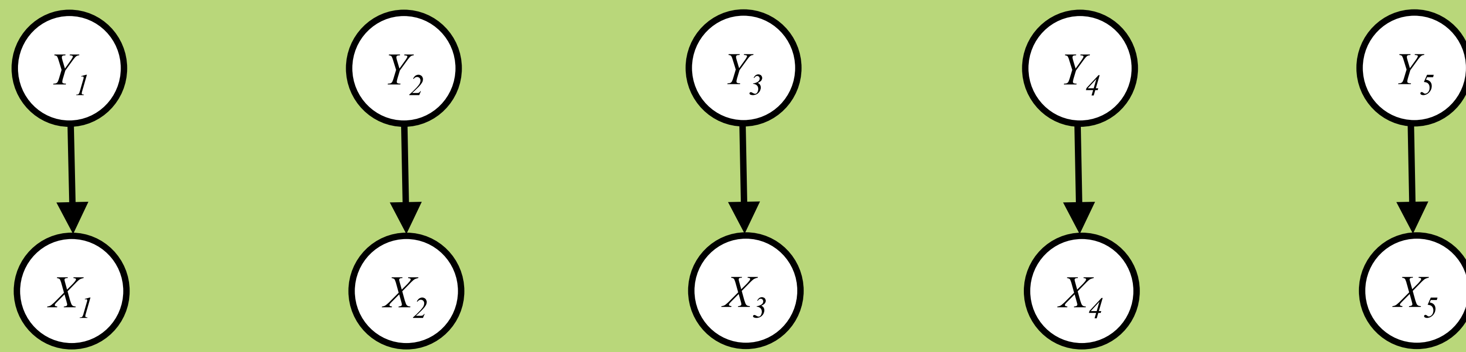
Hidden Markov Model

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.

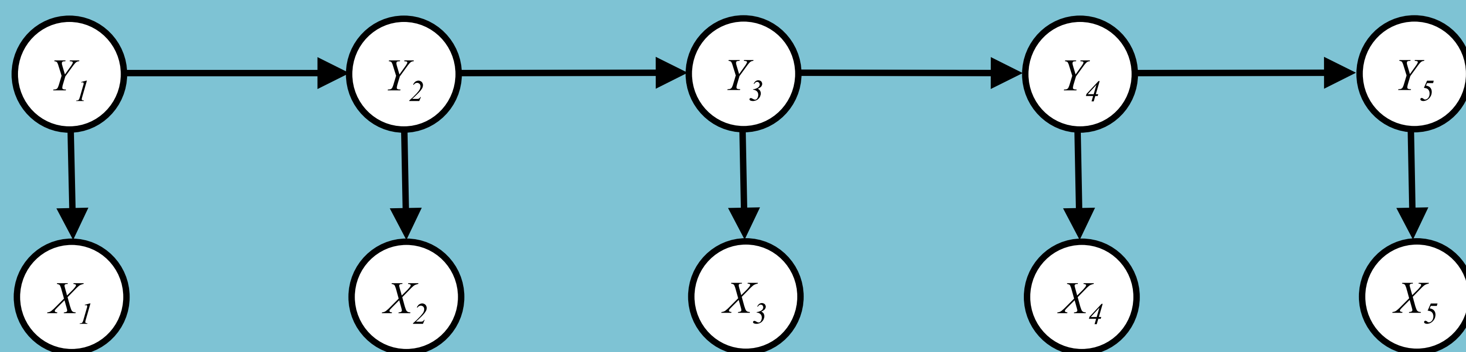
$$p(\text{O}, \text{S}, \text{S}, \text{O}, \text{C}, 2\text{m}, 3\text{m}, 18\text{m}, 9\text{m}, 27\text{m}) \quad = (.8 * .08 * .2 * .7 * .03 * \ldots)$$



From Mixture Model to HMM

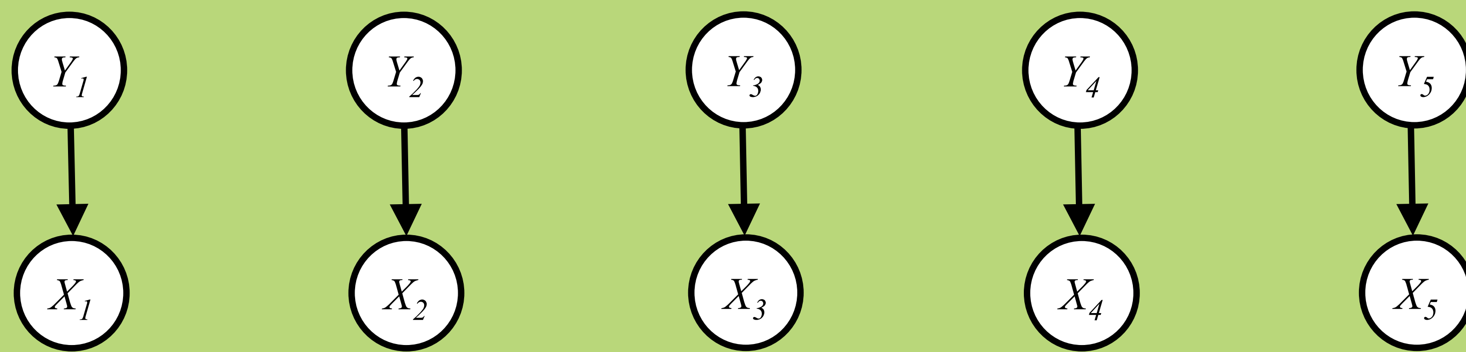


"Naïve Bayes": $$P(\mathbf{X}, \mathbf{Y}) = \prod_{t=1}^{T} P(X_t|Y_t)p(Y_t)$$

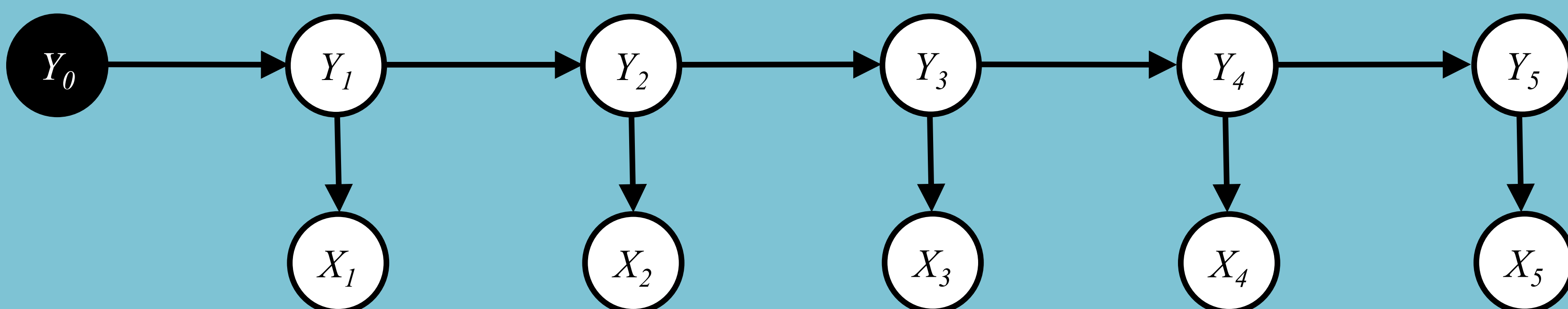


HMM: $$P(\mathbf{X}, \mathbf{Y}) = P(Y_1) \left(\prod_{t=1}^{T} P(X_t|Y_t) \right) \left(\prod_{t=2}^{T} p(Y_t|Y_{t-1}) \right)$$

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

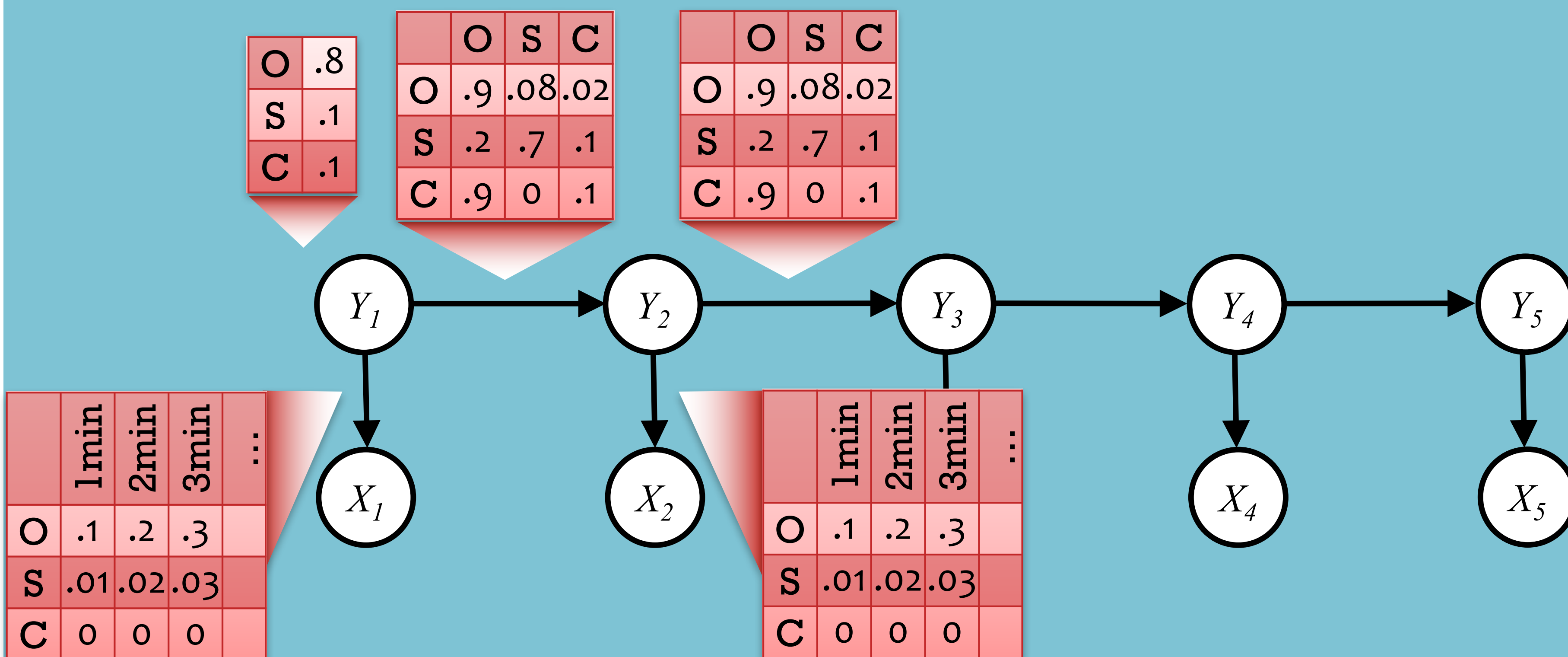
Hidden Markov Model

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 $P(Y_1 = k) = C_k, \forall k$



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 $P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k$

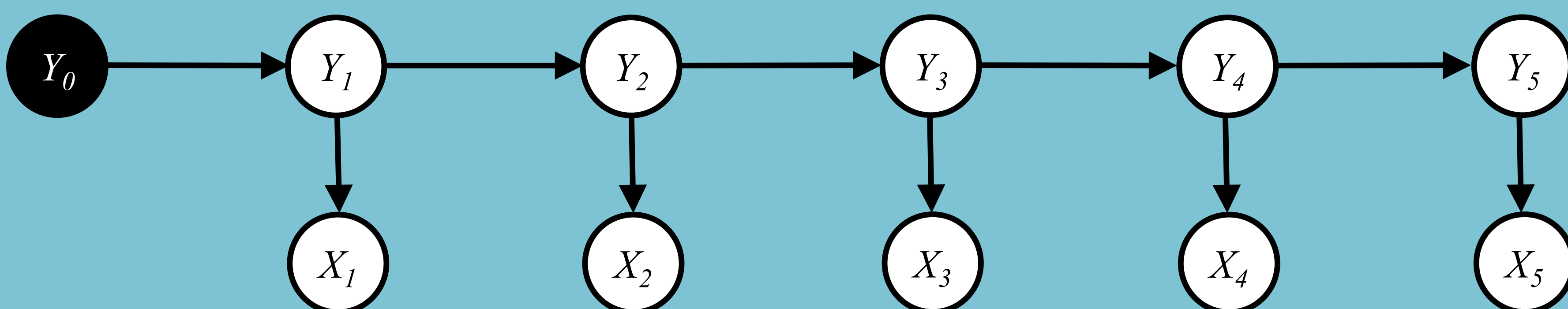
Assumption: $y_0 = \text{START}$

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.



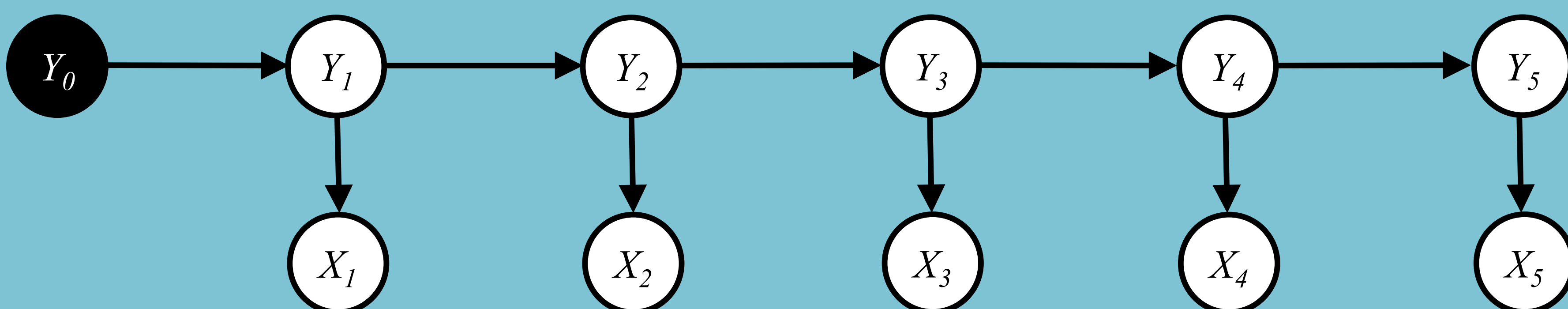
Hidden Markov Model

Joint Distribution:

$$y_0 = \text{START}$$

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

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



Training HMMs

Whiteboard

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