Machine Learning Department
School of Computer Science
Carnegie Mellon University

Exam 2 Review

+

Hidden Markov Models

Matt Gormley
Lecture 18
Mar. 24, 2023

Reminders

- Homework 6: Learning Theory / Generative Models
  - Out: Fri, Mar. 17
  - Due: Fri, Mar. 24 at 11:59pm
  - IMPORTANT: only 2 grace/late days permitted
- Practice Problems: Exam 2
  - Out: Fri, Mar. 24
- Exam 2
  - Thu, Mar. 30, 6:30pm – 8:30pm

EXAM 2 LOGISTICS

Exam 2

- **Time / Location**
  - **Time:** ~~**Thu, Nov 20**~~**, 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 8 – 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, ~~handwritten with pen/pencil or tablet~~)

Topics for Exam 1

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

Topics for Exam 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
  - MLE / MAP
  - Naïve Bayes
  - ~~Generative vs. Discriminative~~
- Regression
  - Linear Regression

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

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

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?

linear separators

$$h(x) = \begin{cases} 1 & \text{if } \sigma(\theta^T x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

(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

2.1 Train and test errors

Q1

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. 75%

   (b) Decrease the training data size. 13%

   (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). 6%

   (d) Decrease model complexity. 85%

   (e) Train on a combination of $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ and test on $\mathcal{D}^{\text{test}}$ 4%

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

   toxic

Samples Questions

2.1 Train and test errors

Q2

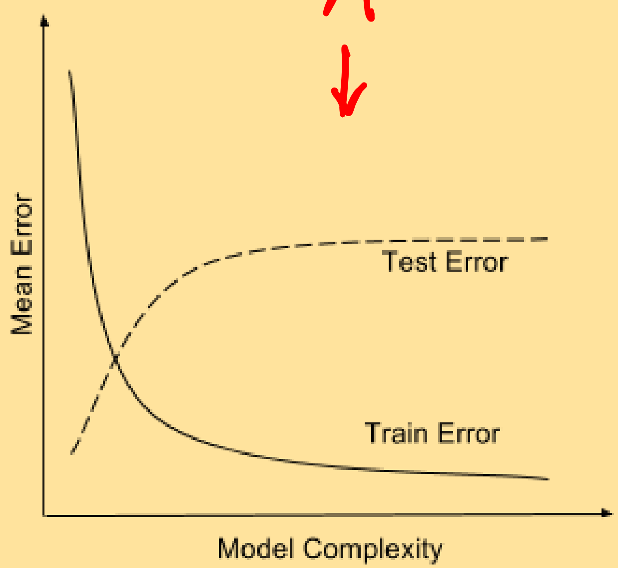
In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $\mathcal{D}^{\text{train}}$, and tested on a separate test set $\mathcal{D}^{\text{test}}$. You look at the test error, and find that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

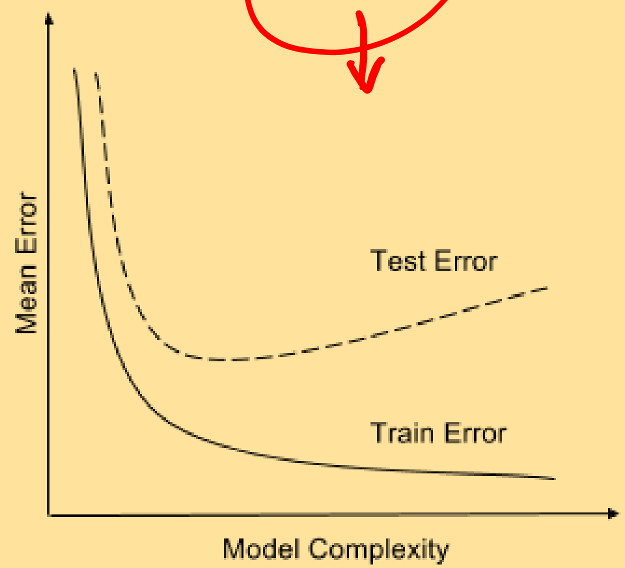
A

B

C=toxic



(a)

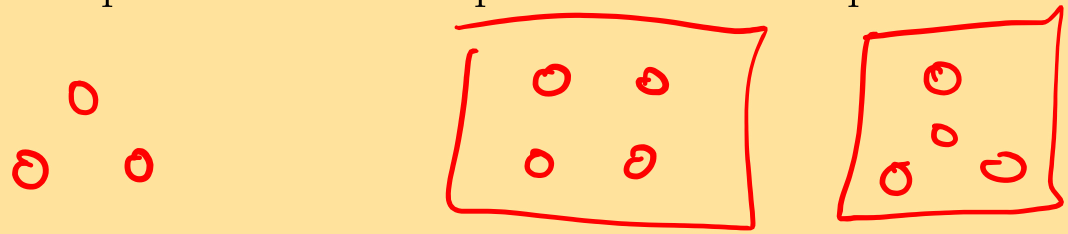


(b)

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 hypothesis space with infinite size is also infinite.

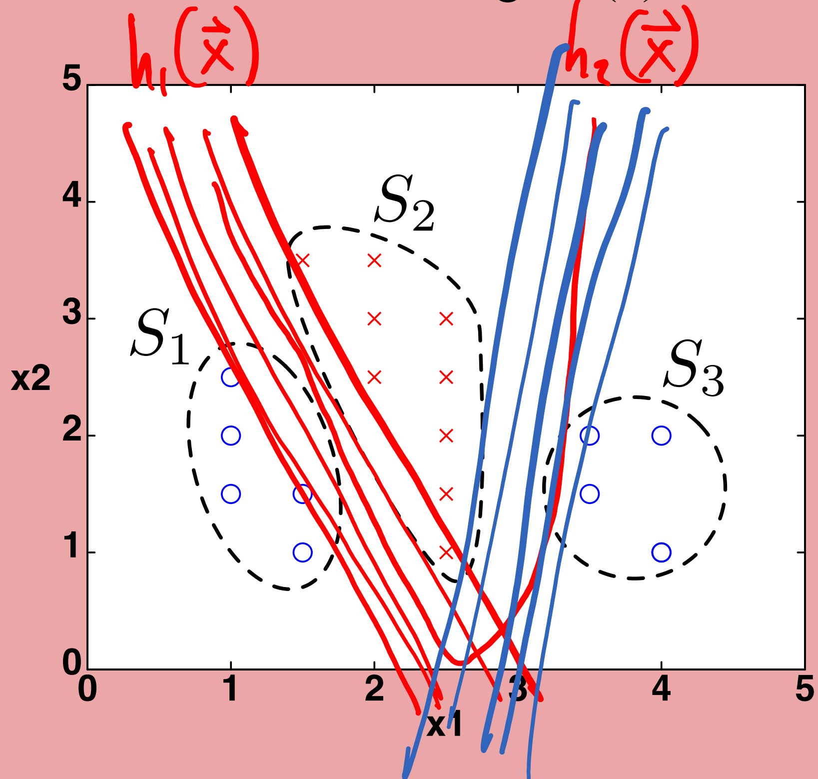
$|\mathcal{H}| = +\infty$

$VC(\mathcal{H}) = ?$

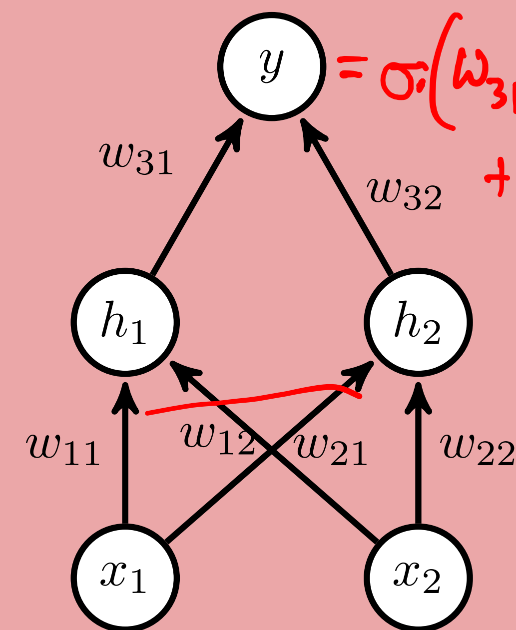
Sample Questions

Neural Networks Q3 $\boxed{A = \text{Yes}}$ B = No C = toxic

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.



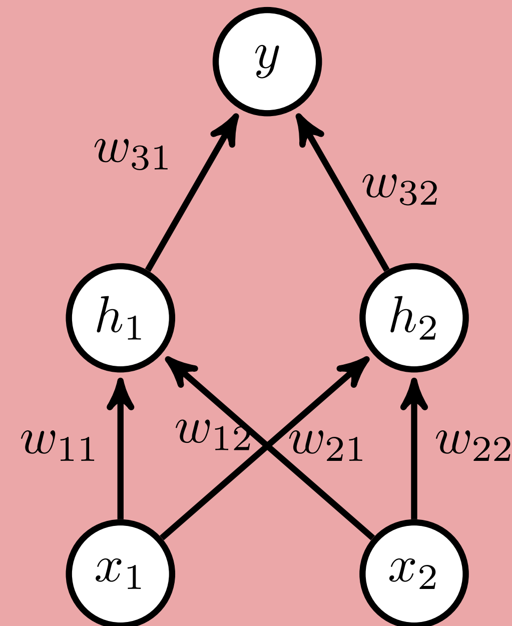
$= \sigma(w_{31} h_1 + w_{32} h_2 + w_{30})$

(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.

Gaussian N.B.

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

NB Assumptions

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

	Gen.	Disc.
MLE	$\prod_i p(\mathbf{x}^{(i)}, y^{(i)} \mid \boldsymbol{\theta})$	$\prod_i p(y^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta})$
MAP	$p(\boldsymbol{\theta}) \prod_i p(\mathbf{x}^{(i)}, y^{(i)} \mid \boldsymbol{\theta})$	$p(\boldsymbol{\theta}) \prod_i p(y^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta})$

MAP Estimation and Regularization

MAP

$$\begin{aligned} \boldsymbol{\theta}_{\text{MAP}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \log p(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ -\log p(\mathcal{D} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \end{aligned}$$

regularization

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ J_{\mathcal{D}}(\boldsymbol{\theta}) + r(\boldsymbol{\theta})$$

fit the data well

keep the model simple

Example: L2 regularization is equivalent to a Gaussian prior

Generative vs. Discriminative

Finite Sample Analysis (Ng & Jordan, 2001)

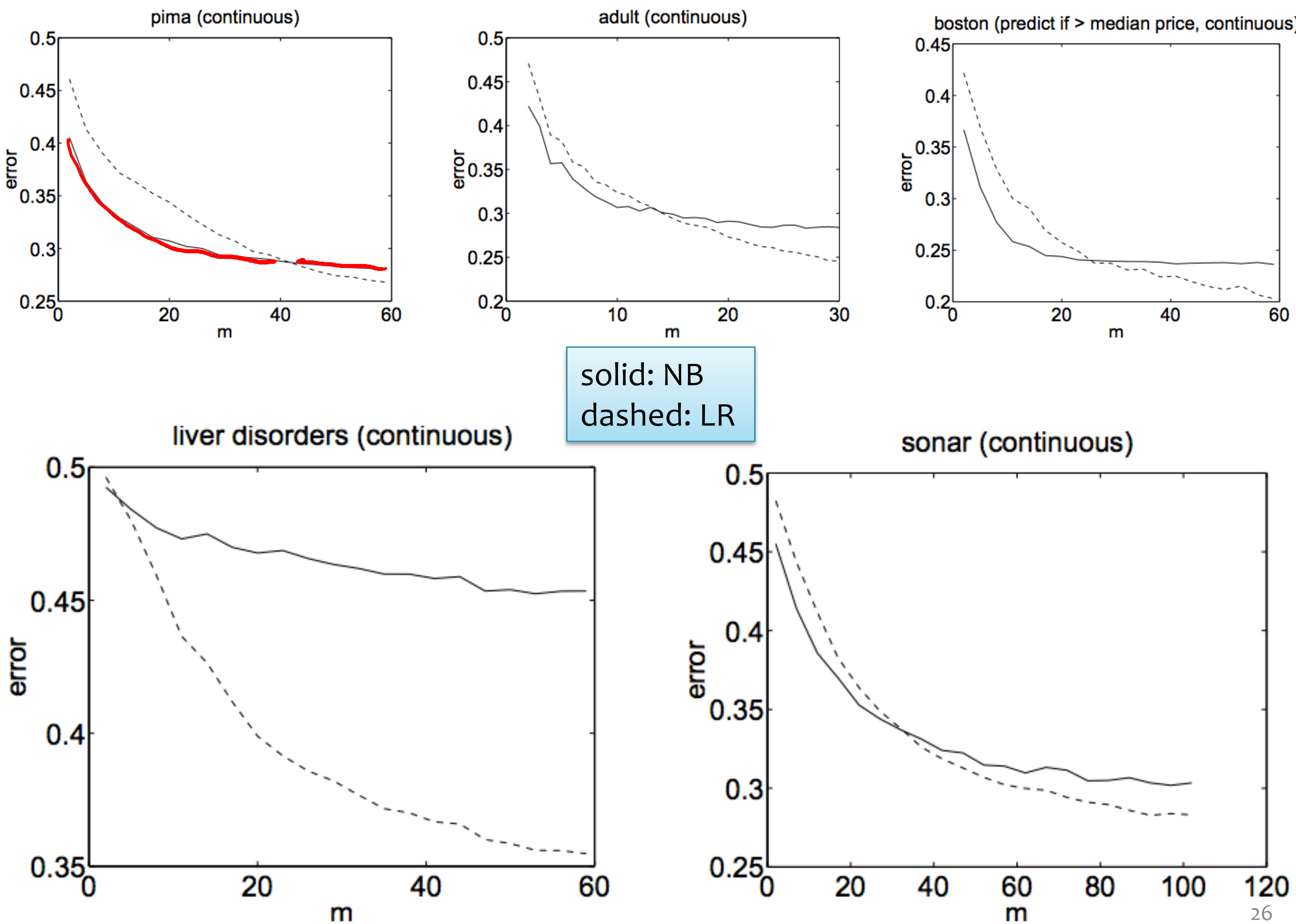
[Assume that we are learning from a finite training dataset]

Naïve Bayes and logistic regression form a *generative-discriminative* model pair:

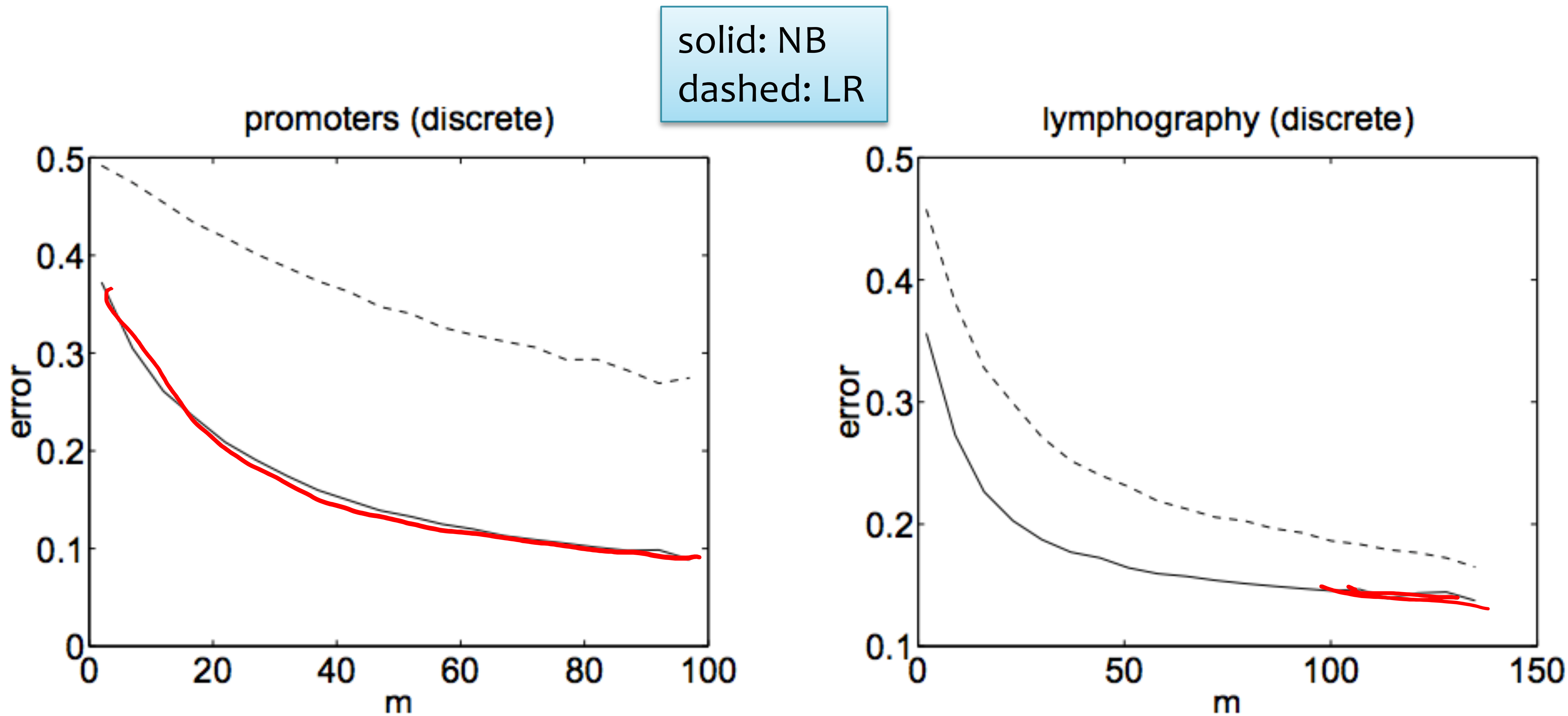
If model assumptions are correct: as the amount of training data increases, Gaussian Naïve Bayes and logistic regression approach the same (linear) decision boundary!

Furthermore, Gaussian Naïve Bayes is a more efficient learner (requires fewer samples) than Logistic Regression

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



Slide courtesy of William Cohen



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

Naïve Bayes vs. Logistic Reg.

Features

$p(x|y)\,p(y)$

Naïve Bayes:
Features 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 x. They can be dependent and correlated in any fashion.

$p(y|x)$

Naïve Bayes vs. Logistic Reg.

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

$p(y=1 \mid \vec{x}) = \sigma(\theta^T x)$

$p(y=0 \mid \vec{x}) = 1 - \sigma(\theta^T x)$

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 Regression

Question: Q4

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:

Summary

1. Naïve Bayes provides a framework for **generative modeling**
2. Choose the feature distributions p$(x_m|y)$ based on the data (e.g., Bernoulli for binary features, Gaussian for continuous features)
3. Train using **MLE** or **MAP** estimation
4. Make predictions by maximizing the posterior $p(y|\boldsymbol{x}')$

Learning Objectives

Naïve Bayes

You should be able to…

1. Write the generative story for Naive Bayes
2. Create a new Naive Bayes classifier using your favorite probability distribution as the event model
3. Apply the principle of maximum likelihood estimation (MLE) to learn the parameters of Bernoulli Naive Bayes
4. Motivate the need for MAP estimation through the deficiencies of MLE
5. Apply the principle of maximum a posteriori (MAP) estimation to learn the parameters of Bernoulli Naive Bayes
6. Select a suitable prior for a model parameter
7. Describe the tradeoffs of generative vs. discriminative models
8. Implement Bernoulli Naives Bayes
9. Describe how the variance affects whether a Gaussian Naive Bayes model will have a linear or nonlinear decision boundary

THE BIG PICTURE

ML Big Picture

Learning Paradigms:
What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation:
What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Application Areas
Key challenges?
NLP, Speech, Computer Vision, Robotics, Medicine, Search

Theoretical Foundations:
What principles guide learning?

- ❑ probabilistic
- ❑ information theoretic
- ❑ evolutionary search
- ❑ ML as optimization

Facets of Building ML Systems:
How to build systems that are robust, efficient, adaptive, effective?

1. *Data prep*
2. *Model selection*
3. *Training (optimization / search)*
4. *Hyperparameter tuning on validation data*
5. *(Blind) Assessment on test data*

Big Ideas in ML:
Which are the ideas driving development of the field?

- *inductive bias*
- *generalization / overfitting*
- *bias-variance decomposition*
- *generative vs. discriminative*
- *deep nets, graphical models*
- *PAC learning*
- *distant rewards*

Classification and Regression: The Big Picture

Recipe for Machine Learning

1. Given data $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
2. (a) Choose a decision function $h_{\boldsymbol{\theta}}(\mathbf{x}) = \cdots$ (parameterized by $\boldsymbol{\theta}$)
   (b) Choose an objective function $J_{\mathcal{D}}(\boldsymbol{\theta}) = \cdots$ (relies on data)
3. Learn by choosing parameters that optimize the objective $J_{\mathcal{D}}(\boldsymbol{\theta})$

$$\hat{\boldsymbol{\theta}} \approx \operatorname*{argmin}_{\boldsymbol{\theta}} J_{\mathcal{D}}(\boldsymbol{\theta})$$

4. Predict on new test example $\mathbf{x}_{\text{new}}$ using $h_{\boldsymbol{\theta}}(\cdot)$

$$\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x}_{\text{new}})$$

Decision Functions

- Perceptron: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T\mathbf{x})$
- Linear Regression: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T\mathbf{x}$
- Discriminative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname*{argmax}_{y} p_{\boldsymbol{\theta}}(y \mid \mathbf{x})$
  - Logistic Regression: $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\theta}^T\mathbf{x})$
  - Neural Net (classification): $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma((\mathbf{W}^{(2)})^T\sigma((\mathbf{W}^{(1)})^T\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})$
- Generative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname*{argmax}_{y} p_{\boldsymbol{\theta}}(\mathbf{x}, y)$
  - Naive Bayes: $p_{\boldsymbol{\theta}}(\mathbf{x}, y) = p_{\boldsymbol{\theta}}(y) \prod_{m=1}^{M} p_{\boldsymbol{\theta}}(x_m \mid y)$

Optimization Method

- Gradient Descent: $\boldsymbol{\theta} \to \boldsymbol{\theta} - \gamma\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta})$
- SGD: $\boldsymbol{\theta} \to \boldsymbol{\theta} - \gamma\nabla_{\boldsymbol{\theta}}J^{(i)}(\boldsymbol{\theta})$
  for $i \sim$ Uniform$(1, \ldots, N)$
  where $J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} J^{(i)}(\boldsymbol{\theta})$
- mini-batch SGD
- closed form
  1. compute partial derivatives
  2. set equal to zero and solve

Objective Function

- MLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$
- MCLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$
- L2 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_2^2$
  (same as Gaussian prior $p(\boldsymbol{\theta})$ over parameters)
- L1 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_1$
  (same as Laplace prior $p(\boldsymbol{\theta})$ over parameters)

MOTIVATION: STRUCTURED PREDICTION

Structured Prediction

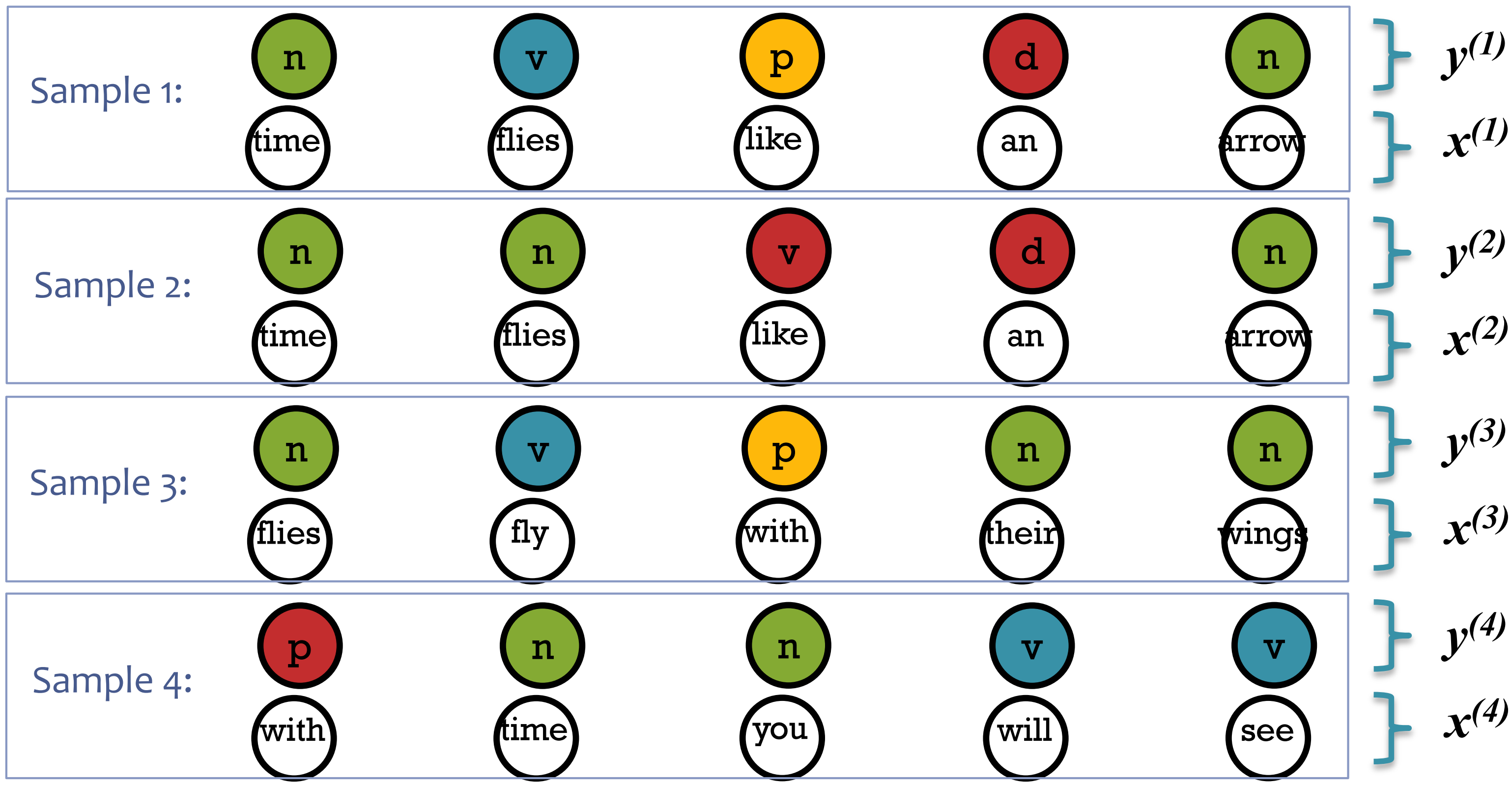
- Most of the models we've seen so far were for **classification**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  - Predict a (binary) **label:** $\boxed{y}$
- Many real-world problems require **structured prediction**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  - Predict a **structure:** $\boxed{\boldsymbol{y} = (y_1, y_2, \ldots, y_J)}$
- 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: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$



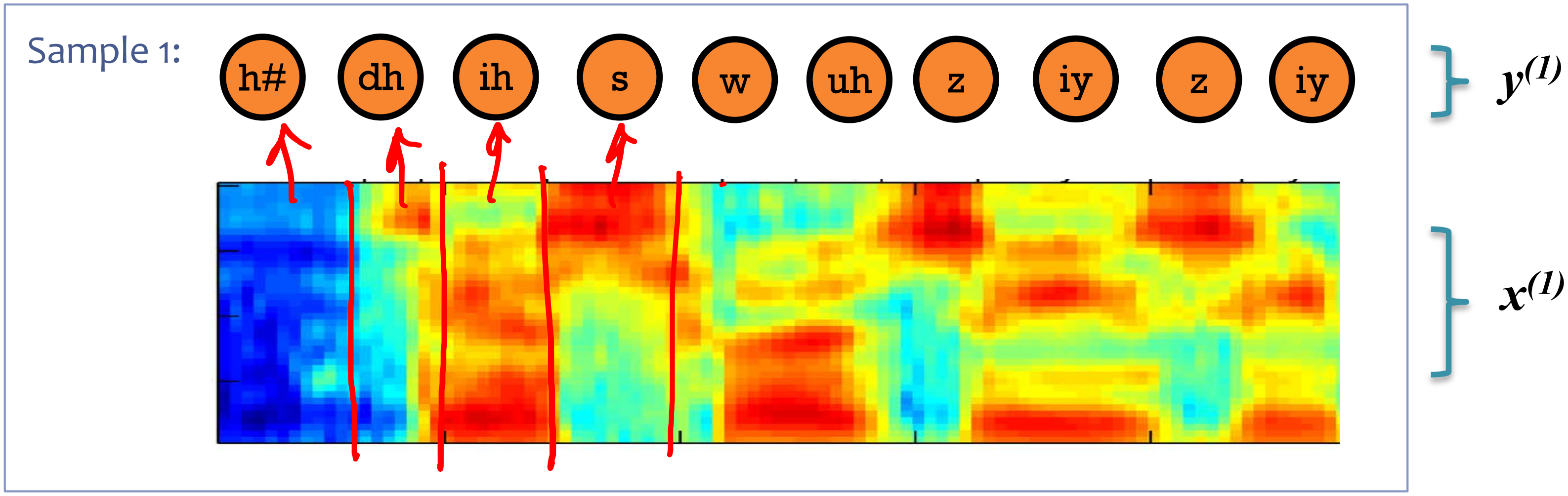
Dataset for Supervised Handwriting Recognition

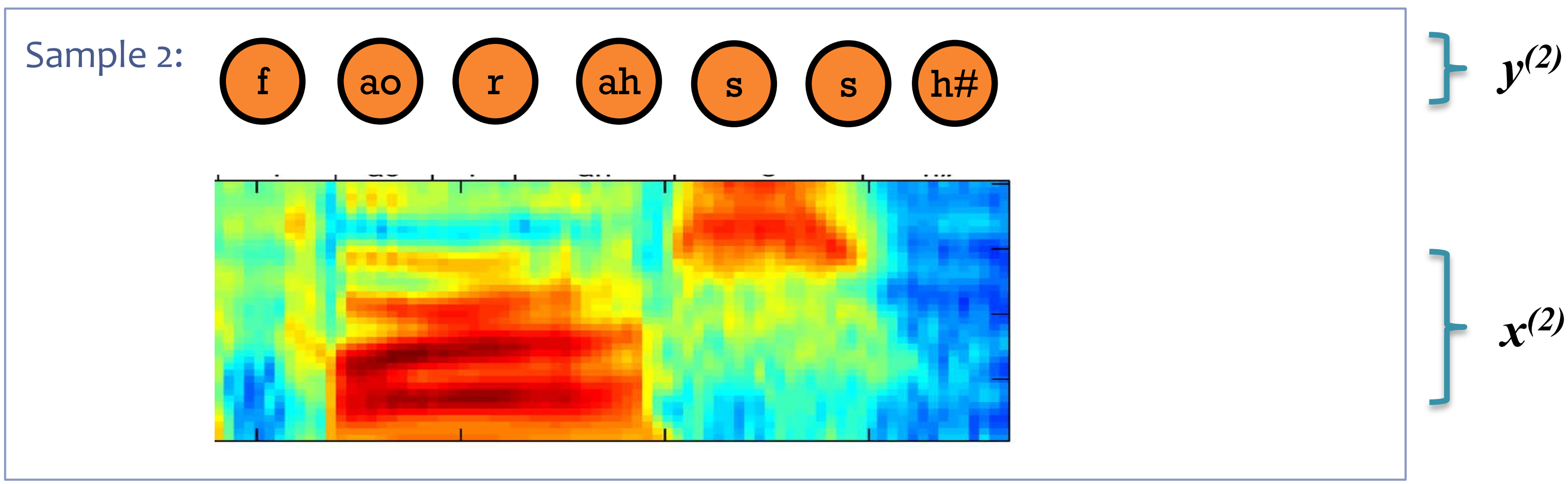
Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Figures from (Chatzis & Demiris, 2013)

Dataset for Supervised Phoneme (Speech) Recognition

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

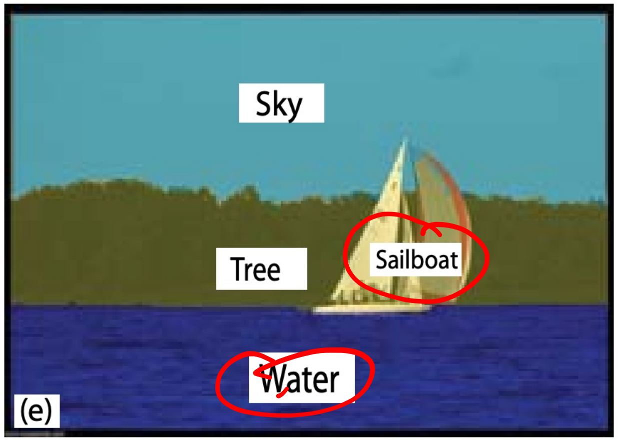




Figures from (Jansen & Niyogi, 2013)

(very small) Dataset for Scene Understanding

$x^{(1)}$

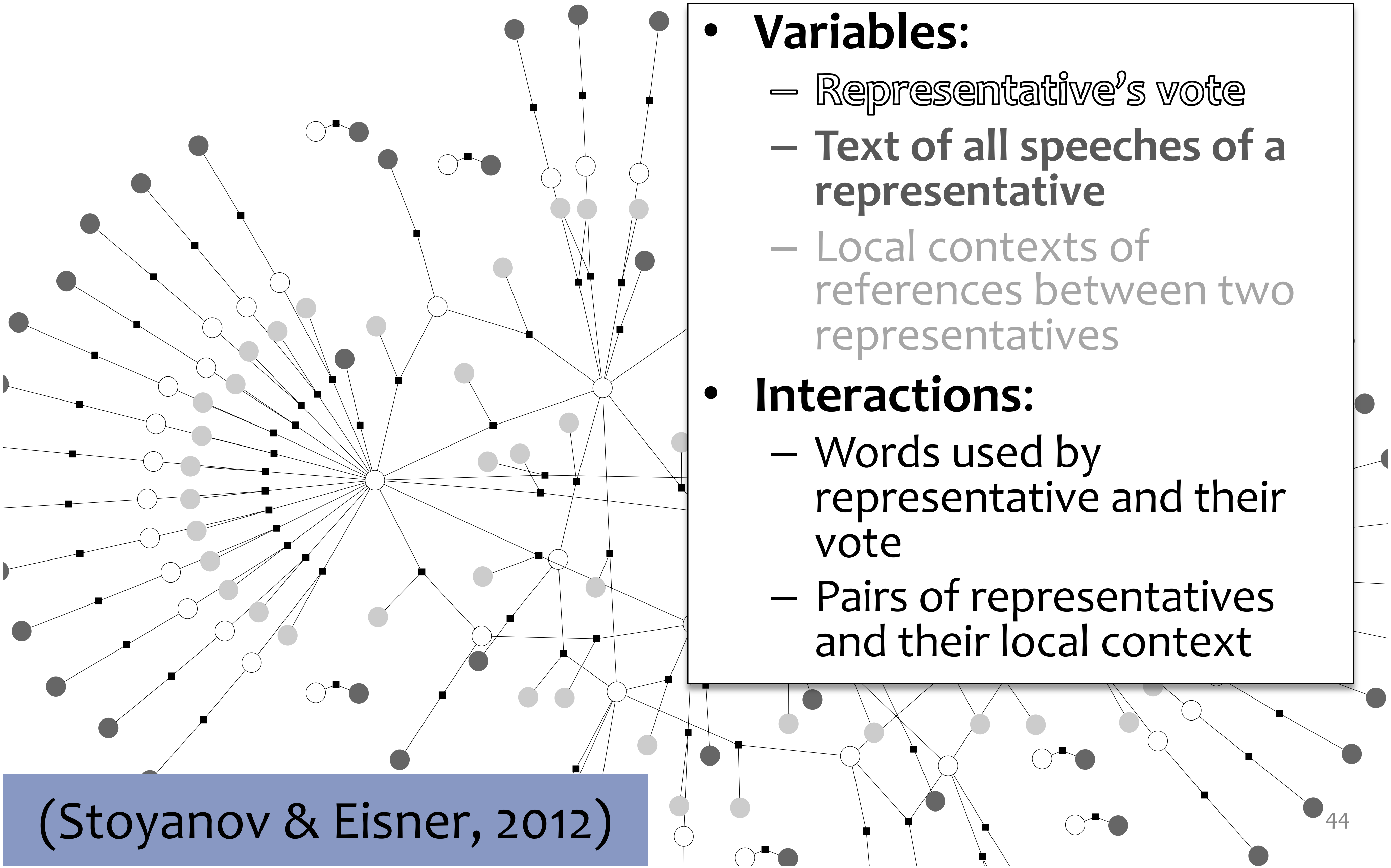


$y^{(1)}$

(Li et al., 2009)

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.

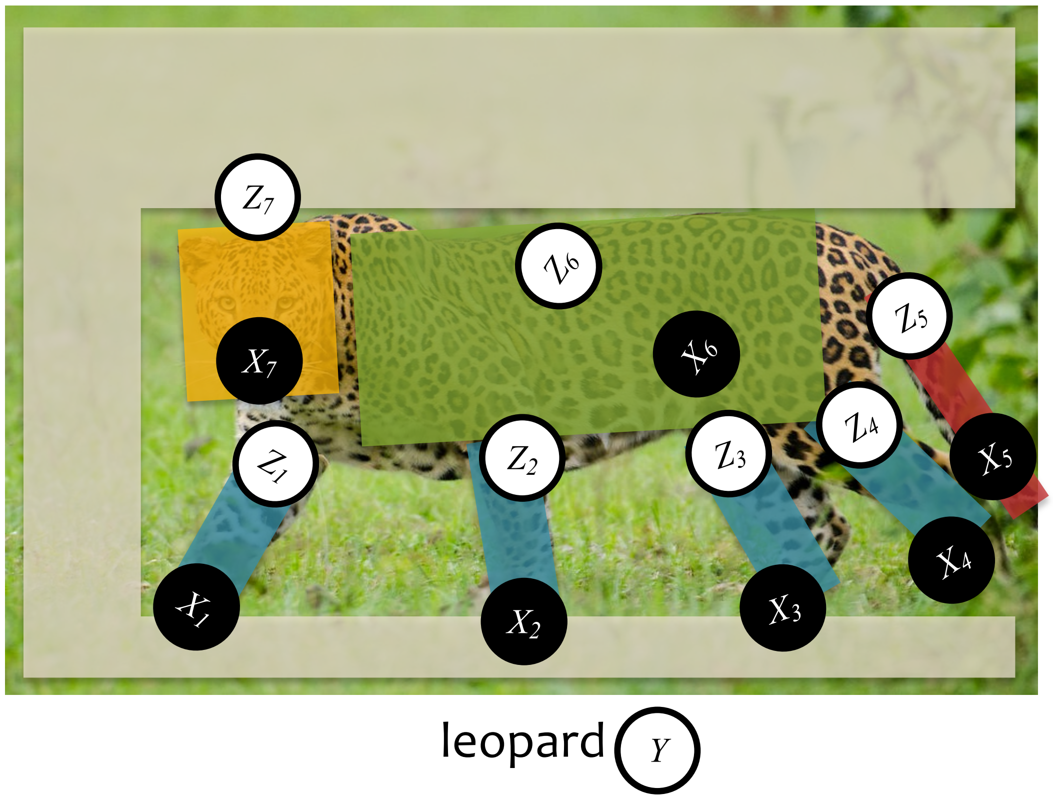
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

leopard

Case Study: Object Recognition

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

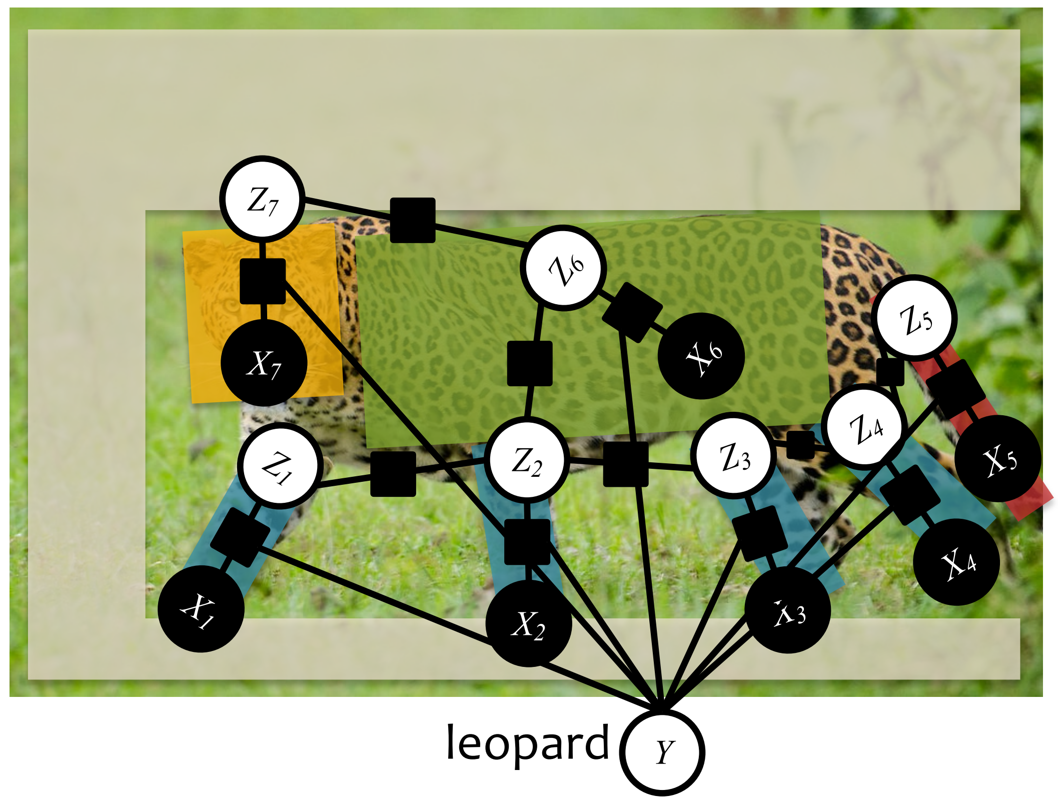
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ 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

$|\mathcal{Y}| = 45^{23}$

Classification

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|\mathbf{x})$$

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

Structured Prediction

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

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

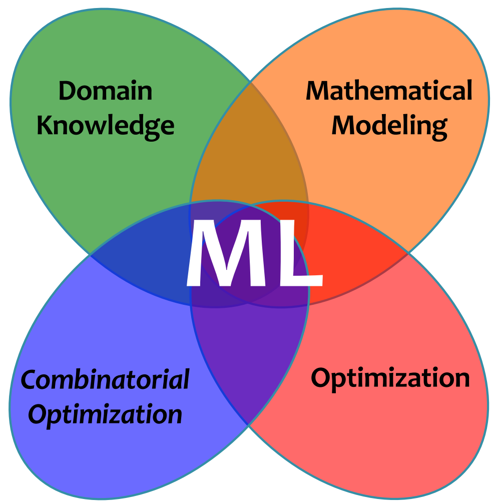
and $|\mathcal{Y}|$ is very large

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



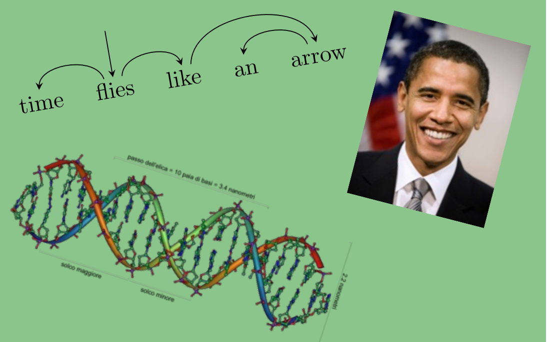
Inference finds {best structure, marginals, partition function} for a new observation

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

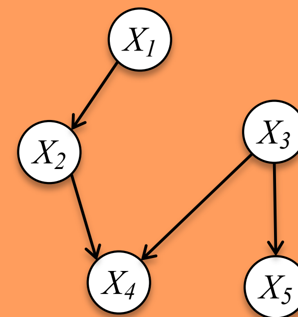
Learning tunes the parameters of the model

Machine Learning

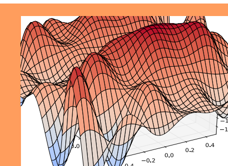
Data



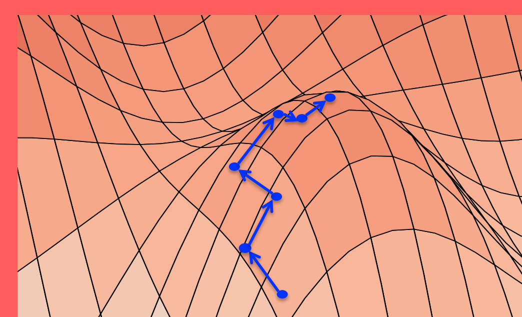
Model



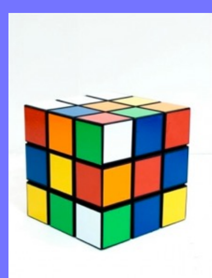
Objective



Learning



Inference



(**Inference** is usually called as a subroutine in 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:

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

or equivalently:

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

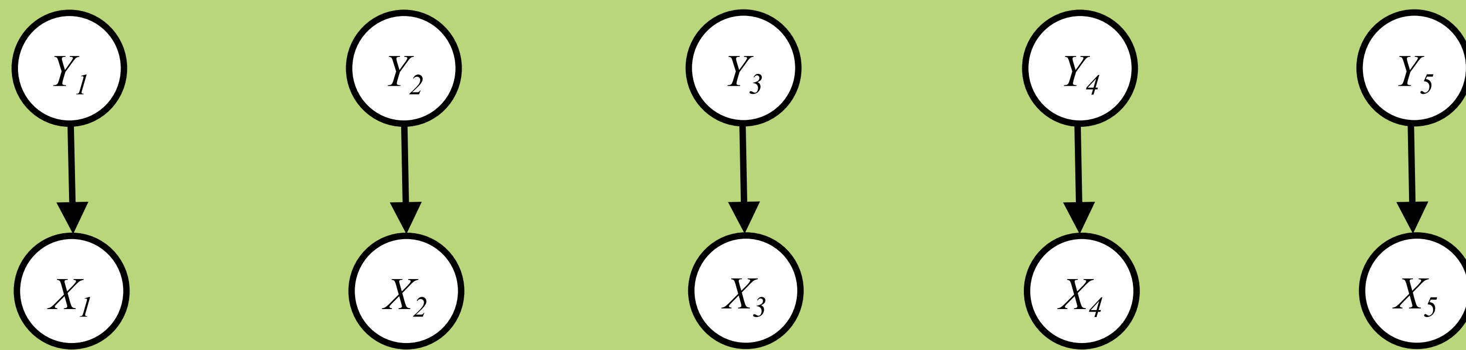
We write this as:

$$\boxed{A \perp\!\!\!\perp B|C}$$

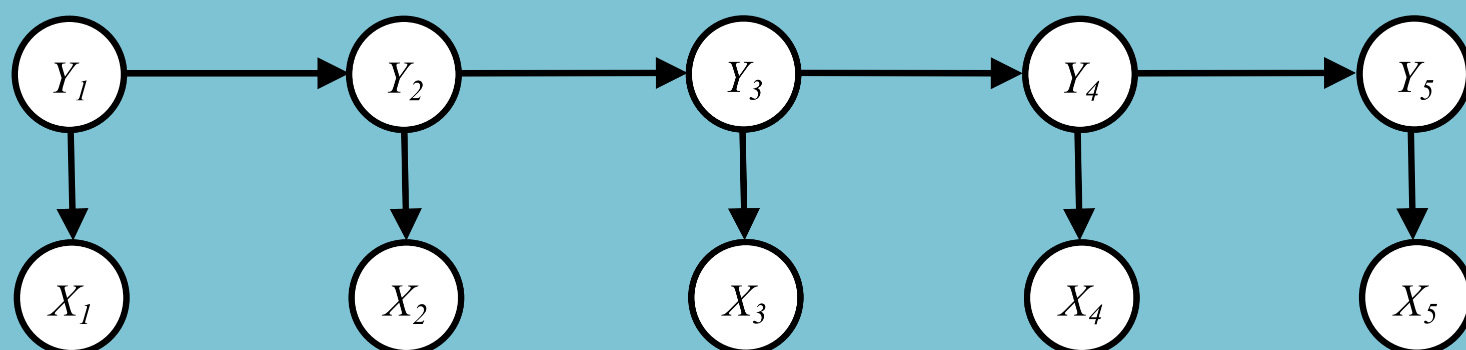
~~Later we will also write: $I<A, \{C\}, B>$~~

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

Markov Models

Whiteboard

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


callumbal

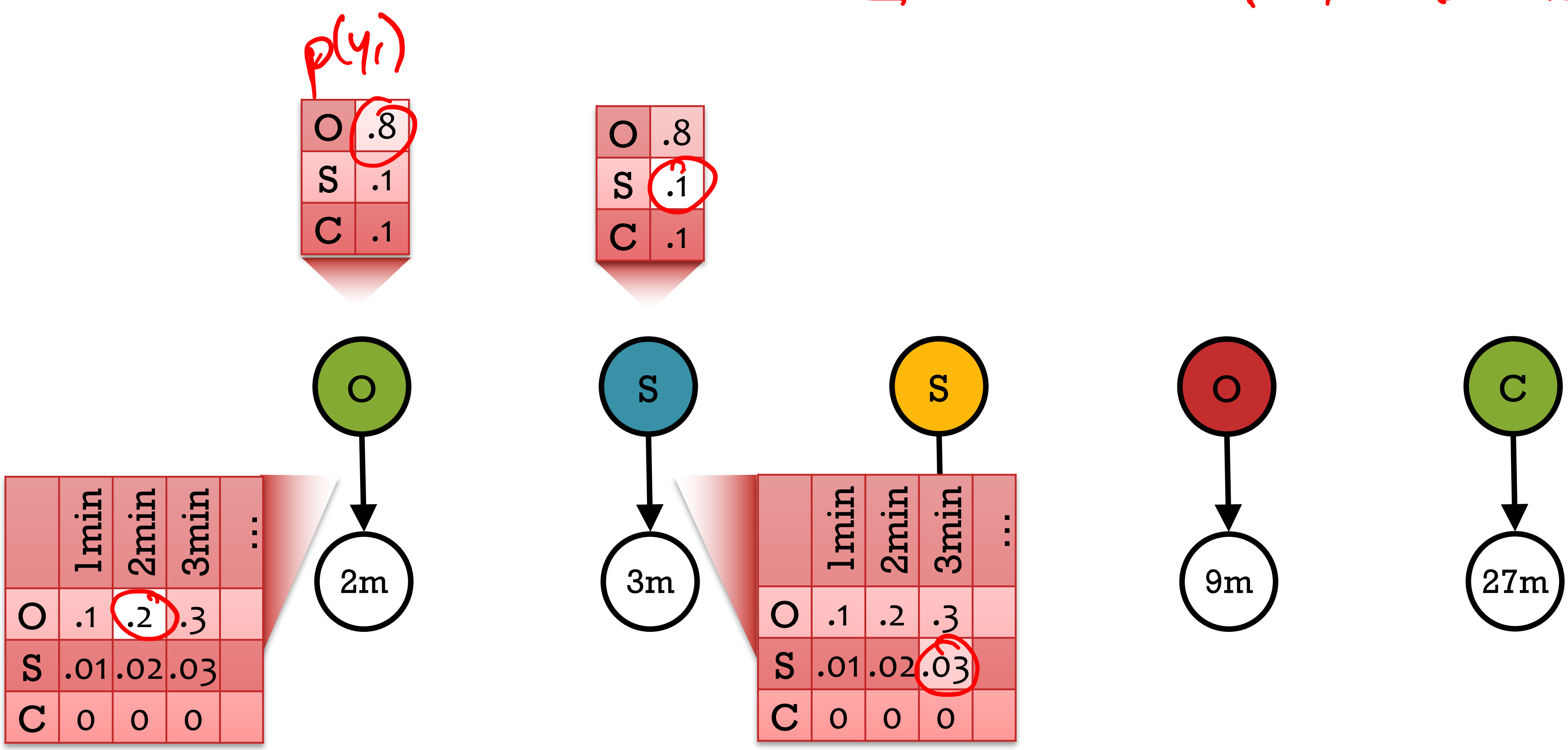
Totoro’s Tunnel


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

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).

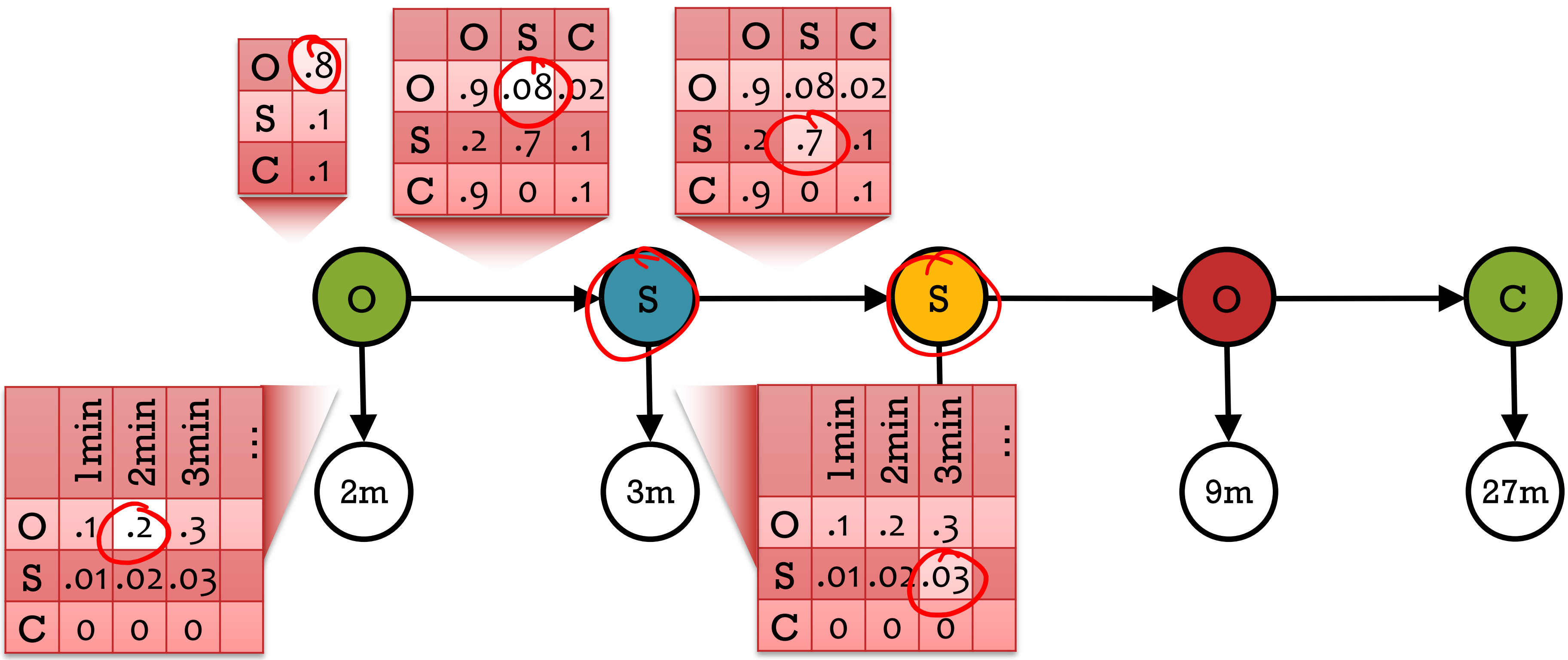
$$\underline{p(\mathrm{O}, \mathrm{S}, \mathrm{S}, \mathrm{O}, \mathrm{C},} \underline{2\mathrm{m}, 3\mathrm{m}, 18\mathrm{m}, 9\mathrm{m}, 27\mathrm{m})} = (.8 * \underline{.2} * \underline{.1} * \underline{.03} * \underline{\ldots})$$



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.

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



O	.8
S	.1
C	.1

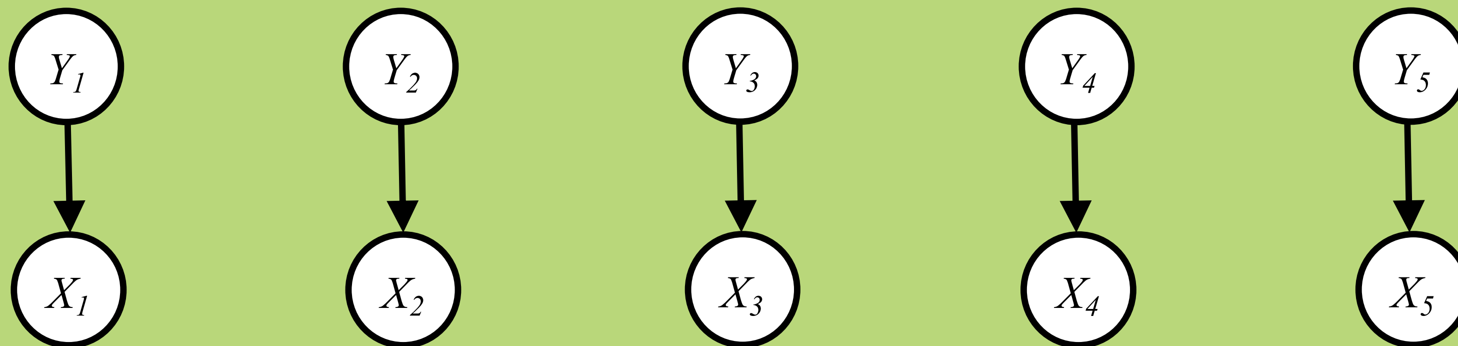
	O	S	C
O	.9	.08	.02
S	.2	.7	.1
C	.9	0	.1

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

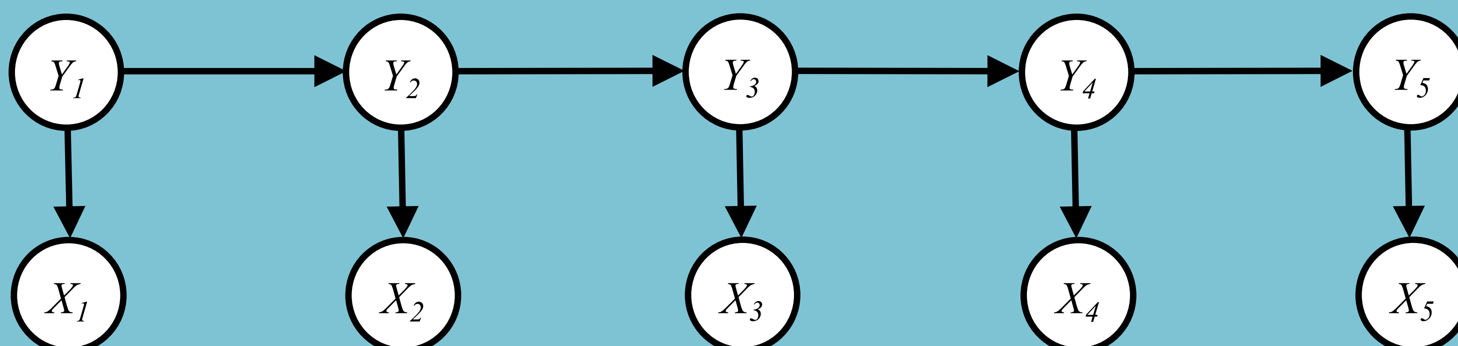
	1min	2min	3min	…
O	.1	.2	.3	
S	.01	.02	.03	
C	0	0	0	

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

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

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)
   $x^{(i)} \sim p(x|\boldsymbol{\theta})$
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)
   $\partial\ell(\boldsymbol{\theta})/\partial\theta_1 = \ldots$
   $\partial\ell(\boldsymbol{\theta})/\partial\theta_2 = \ldots$
   …
   $\partial\ell(\boldsymbol{\theta})/\partial\theta_M = \ldots$
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

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

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

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

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

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

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

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 (v1)

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

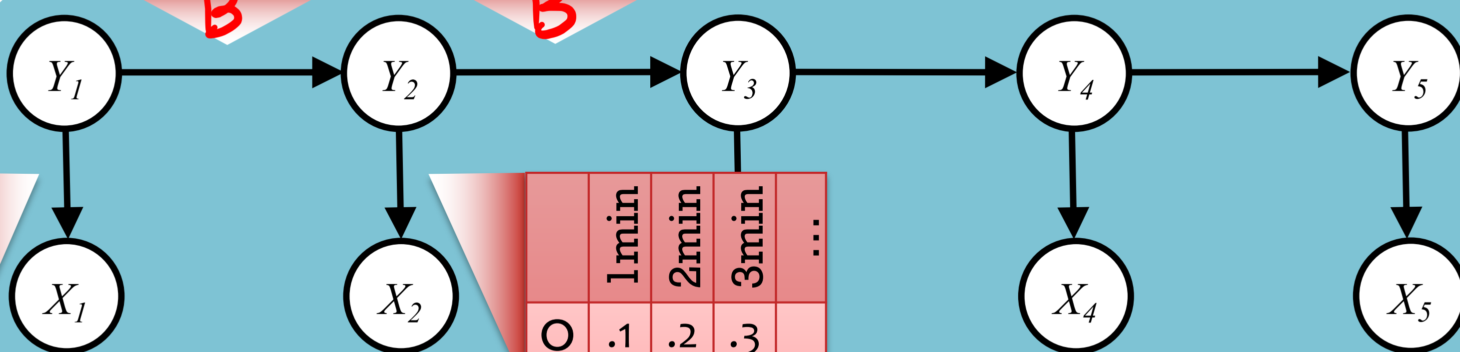
O	.8
S	.1
C	.1

B

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

B

	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	

A

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

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

Hidden Markov Model (v1)

HMM Parameters:

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

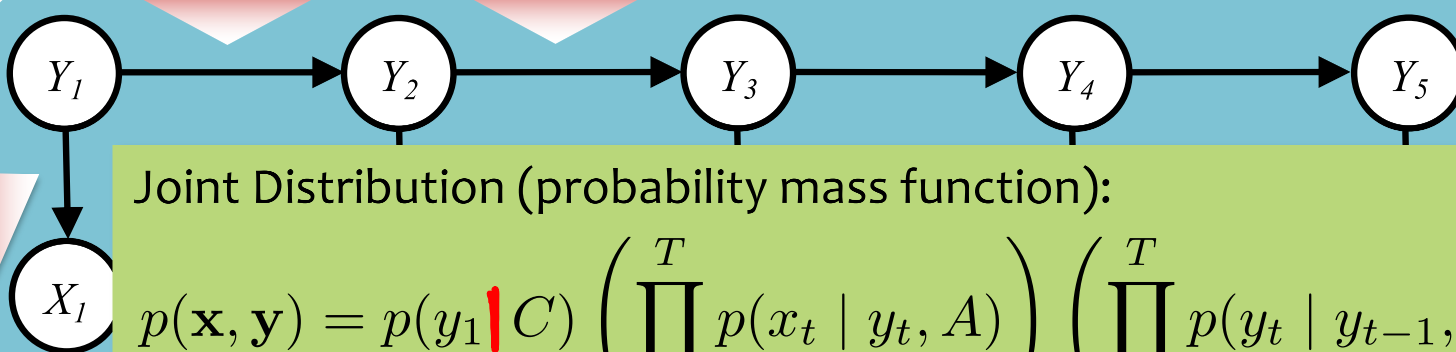
Transition matrix, $\mathbf{B}$, where $P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k$

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

O	.8
S	.1
C	.1

	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



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

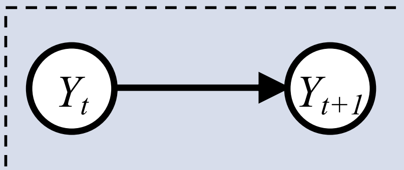
Joint Distribution (probability mass function):

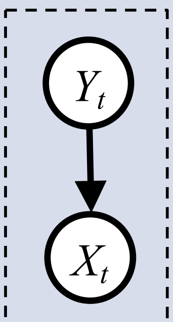
$$p(\mathbf{x}, \mathbf{y}) = p(y_1 | C) \left(\prod_{t=1}^{T} p(x_t \mid y_t, A) \right) \left(\prod_{t=2}^{T} p(y_t \mid y_{t-1}, B) \right)$$

$$= C_{y_1} \left(\prod_{t=1}^{T} A_{y_t, x_t} \right) \left(\prod_{t=2}^{T} B_{y_{t-1}, y_t} \right)$$

Supervised Learning for HMM (v1)

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





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

Likelihood:

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

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

MLE:

$$\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}} = \operatorname*{argmax}_{\mathbf{A},\mathbf{B},\mathbf{C}} \ell(\mathbf{A}, \mathbf{B}, \mathbf{C})$$

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

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

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

$N = 52$ (weeks)
$T = 5$ (M-F)

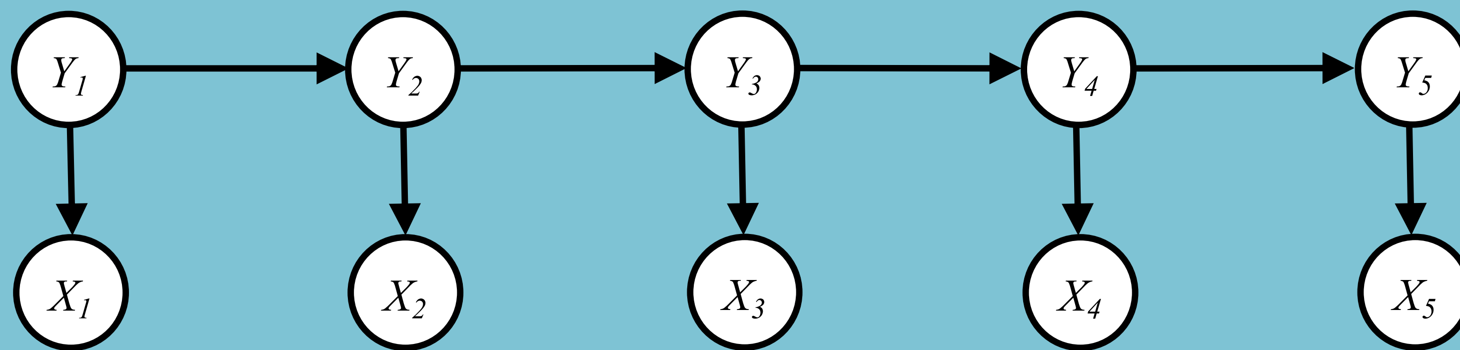
We can solve the above in closed form, which yields...

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

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

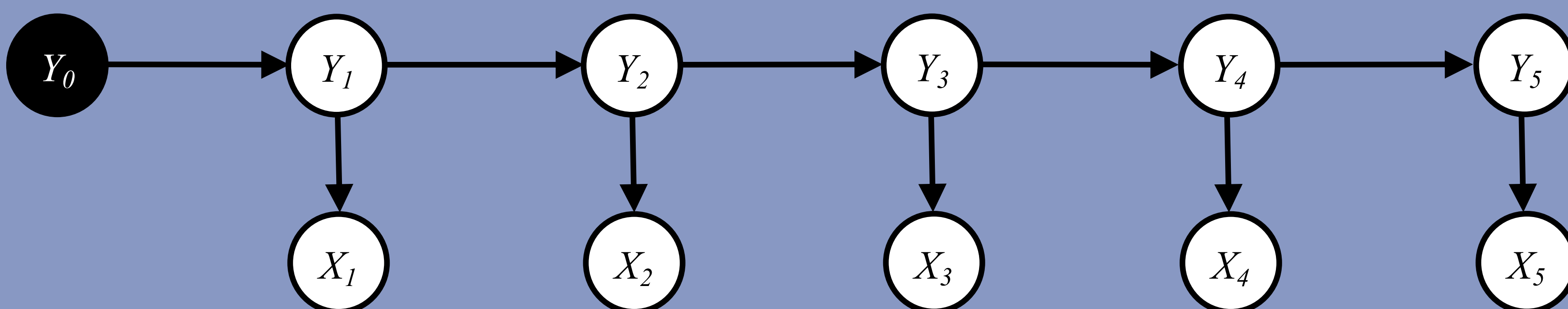
$$\hat{A}_{j,k} = \frac{\#(x_t^{(i)} = k \text{ and } y_t^{(i)} = j)}{\#(y_t^{(i)} = j)}, \forall j, k$$

HMM (two ways)



HMM (v1):

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



HMM (v2):

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

Hidden Markov Model (v2)

HMM Parameters:

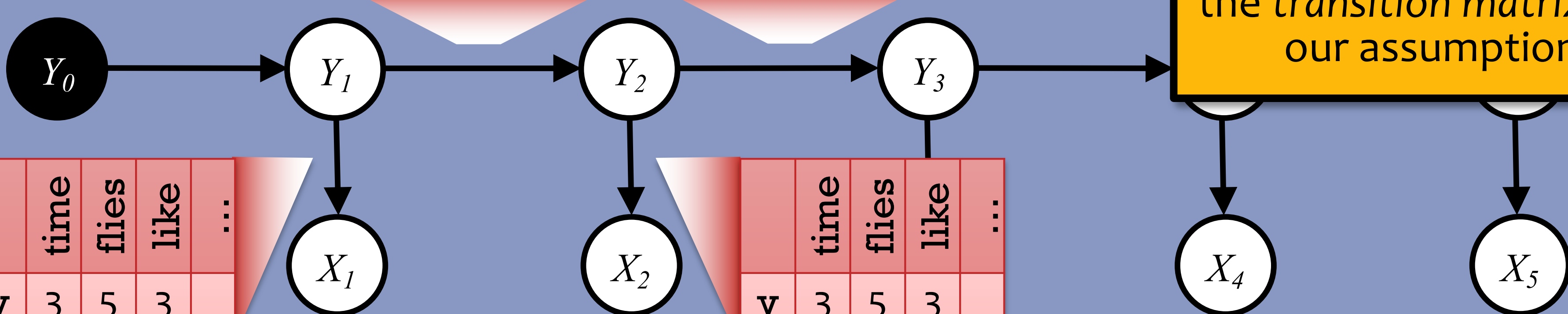
Emission matrix, $\mathbf{A}$, where $P(X_k = w | Y_k = t) = A_{t,w}, \forall k$

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

	O	S	C	Start
O	.9	.08	.02	0
S	.2	.7	.1	0
C	.9	0	.1	0
Start	0.8	0.1	0.1	0

	O	S	C	Start
O	.9	.08	.02	0
S	.2	.7	.1	0
C	.9	0	.1	0
Start	0.8	0.1	0.1	0

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



	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

Hidden Markov Model (v2)

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

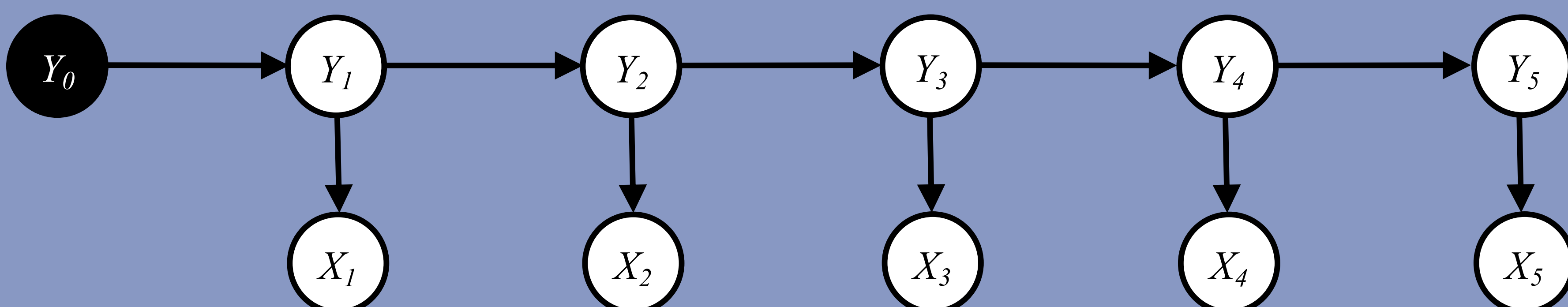
Assumption: $y_0 =$ START

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.



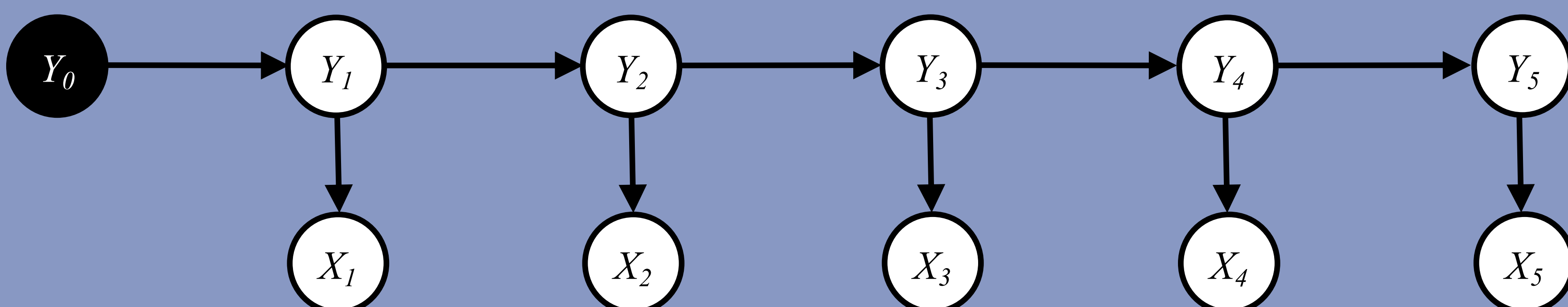
Hidden Markov Model (v2)

Joint Distribution (probability mass function):

$$y_0 = \text{START}$$

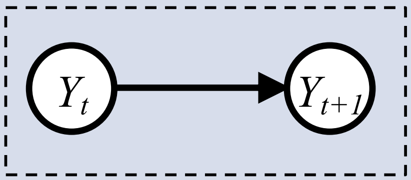
$$p(\mathbf{x}, \mathbf{y} | y_0) = \prod_{t=1}^{T} p(x_t | y_t) p(y_t | y_{t-1})$$

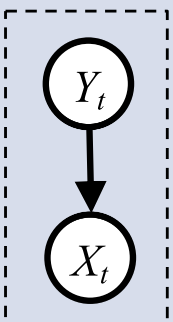
$$= \prod_{t=1}^{T} A_{y_t, x_t} B_{y_{t-1}, y_t}$$



Supervised Learning for HMM (v2)

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





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

We assume $y_0^{(i)} = $ START for all i

Likelihood:

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

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

MLE:

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

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

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

We can solve the above in closed form, which yields...

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

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