Naïve Bayes

Generative vs. Discriminative
Reminders

• Homework 6: Learning Theory / Generative Models
  – Out: Fri, Mar 20
  – Due: Fri, Mar 27 at 11:59pm

• Midterm Exam 2
  – Thu, Apr 2 – evening exam, details announced on Piazza

• Today’s In-Class Poll
  – http://poll.mlcourse.org
Q: Why would we use Naïve Bayes? Isn’t it too Naïve?

A: Naïve Bayes has one **key advantage** over methods like Perceptron, Logistic Regression, Neural Nets:

   **Training is lightning fast!**

   While other methods require slow iterative training procedures that might require hundreds of epochs, Naïve Bayes computes its parameters in closed form by counting.
NAÏVE BAYES
Model 1: Bernoulli Naïve Bayes

If HEADS, flip each red coin

If TAILS, flip each blue coin

We can generate data in this fashion. Though in practice we never would since our data is given.

Instead, this provides an explanation of how the data was generated (albeit a terrible one).
What’s wrong with the Naïve Bayes Assumption?

The features might not be independent!!

• Example 1:
  – If a document contains the word “Donald”, it’s extremely likely to contain the word “Trump”
  – These are not independent!

• Example 2:
  – If the petal width is very high, the petal length is also likely to be very high
Recipe for Closed-form MLE

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)
   \[ x^{(i)} \sim p(x|\theta) \]

2. Write log-likelihood
   \[ \ell(\theta) = \log p(x^{(1)}|\theta) + \ldots + \log p(x^{(N)}|\theta) \]

3. Compute partial derivatives (i.e. gradient)
   \[ \frac{\partial \ell(\theta)}{\partial \theta_1} = \ldots \]
   \[ \frac{\partial \ell(\theta)}{\partial \theta_2} = \ldots \]
   \[ \ldots \]
   \[ \frac{\partial \ell(\theta)}{\partial \theta_M} = \ldots \]

4. Set derivatives to zero and solve for \( \theta \)
   \[ \frac{\partial \ell(\theta)}{\partial \theta_m} = 0 \text{ for all } m \in \{1, \ldots, M\} \]
   \( \theta_{\text{MLE}} \) = solution to system of \( M \) equations and \( M \) variables

5. Compute the second derivative and check that \( \ell(\theta) \) is concave down at \( \theta_{\text{MLE}} \)
Naïve Bayes: Learning from Data

Whiteboard

– Data likelihood
– MLE for Naive Bayes
– Example: MLE for Naïve Bayes with Two Features
– MAP for Naive Bayes
NAÏVE BAYES: MODEL DETAILS
Model 1: Bernoulli Naïve Bayes

Data: Binary feature vectors, Binary labels
\[ x \in \{0, 1\}^M \]
\[ y \in \{0, 1\} \]

Generative Story:
\[ y \sim \text{Bernoulli}(\phi) \]
\[ x_1 \sim \text{Bernoulli}(\theta_{y,1}) \]
\[ x_2 \sim \text{Bernoulli}(\theta_{y,2}) \]
\[ \vdots \]
\[ x_M \sim \text{Bernoulli}(\theta_{y,M}) \]

Model:
\[ p_{\phi,\theta}(x, y) = p_{\phi,\theta}(x_1, \ldots, x_M, y) \]
\[ = p_{\phi}(y) \prod_{m=1}^{M} p_{\theta}(x_m|y) \]
\[ = \left[ (\phi)^y (1 - \phi)^{(1-y)} \right] \left[ \prod_{m=1}^{M} (\theta_{y,m})^{x_m} (1 - \theta_{y,m})^{(1-x_m)} \right] \]
Model 1: Bernoulli Naïve Bayes

Maximum Likelihood Estimation

**Training:** Find the class-conditional MLE parameters

**Count Variables:**

\[
N_{y=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)
\]

\[
N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)
\]

\[
N_{y=0, x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x_m^{(i)} = 1)
\]

\[
\ldots
\]

**Maximum Likelihood Estimators:**

\[
\phi = \frac{N_{y=1}}{N}
\]

\[
\theta_{0,m} = \frac{N_{y=0, x_m=1}}{N_{y=0}}
\]

\[
\theta_{1,m} = \frac{N_{y=1, x_m=1}}{N_{y=1}}
\]

\[\forall m \in \{1, \ldots, M\}\]
**Model 1: Bernoulli Naïve Bayes**

**Maximum Likelihood Estimation**

**Training:** Find the class-conditional MLE parameters

**Count Variables:**

\[ N_{y=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1) \]

\[ N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0) \]

\[ N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x_m^{(i)} = 1) \]

... 

\[ \phi = \frac{N_{y=1}}{N} \]

\[ \theta_{0,m} = \frac{N_{y=0,x_m=1}}{N_{y=0}} \]

\[ \theta_{1,m} = \frac{N_{y=1,x_m=1}}{N_{y=1}} \]

\[ \forall m \in \{1, \ldots, M\} \]

**Question 1:**

What is the MLE of \( \phi \)?

(A) 0/6 (B) 1/6 (C) 2/6 (D) 3/6 (E) 4/6 (F) 5/6 (G) 6/6 (H) None of the above
Model 1: Bernoulli Naïve Bayes

Maximum Likelihood Estimation

**Training:** Find the class-conditional MLE parameters

**Count Variables:**

\[ N_{y=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1) \]

\[ N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0) \]

\[ N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x_m^{(i)} = 1) \]

\[ \phi = \frac{N_{y=1}}{N} \]

**Maximum Likelihood Estimators:**

\[ \theta_{0,m} = \frac{N_{y=0,x_m=1}}{N_{y=0}} \]

\[ \theta_{1,m} = \frac{N_{y=1,x_m=1}}{N_{y=1}} \]

\[ \forall m \in \{1, \ldots, M\} \]

**Data:**

<table>
<thead>
<tr>
<th>y</th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>\ldots</th>
<th>x_M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
</tr>
</tbody>
</table>

**Question 2:**

What is the MLE of \( \theta_{0,1} \)?

(A) 0/6 (B) 1/6 (C) 2/6 (D) 3/6 (E) 4/6 (F) 5/6 (G) 6/6 (H) None of the above

Maximum Likelihood Estimation

Training: Find the class-conditional MLE parameters

Count Variables:

\[ N_{y=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1) \]

\[ N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0) \]

\[ N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x_m^{(i)} = 1) \]

\[ \phi = \frac{N_{y=1}}{N} \]

Maximum Likelihood Estimators:

\[ \theta_{0,m} = \frac{N_{y=0,x_m=1}}{N_{y=0}} \]

\[ \theta_{1,m} = \frac{N_{y=1,x_m=1}}{N_{y=1}} \]

\[ \forall m \in \{1, \ldots, M\} \]
Model 1: Bernoulli Naïve Bayes

Maximum Likelihood Estimation

Training: Find the class-conditional MLE parameters

Count Variables:

\[ N_{y=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1) \]
\[ N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0) \]
\[ N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x_m^{(i)} = 1) \]
\[ \vdots \]

Maximum Likelihood Estimators:

\[ \phi = \frac{N_{y=1}}{N} \]
\[ \theta_{0,m} = \frac{N_{y=0,x_m=1}}{N_{y=0}} \]
\[ \theta_{1,m} = \frac{N_{y=1,x_m=1}}{N_{y=1}} \]
\[ \forall m \in \{1, \ldots, M\} \]

MLE for Naïve Bayes is a splendid learning algorithm for when you have say billions of training examples and hundreds of millions of features!

You only need one pass through the data to perform some counting.
What does maximizing likelihood accomplish?

- There is only a finite amount of probability mass (i.e. sum-to-one constraint)
- MLE tries to allocate as much probability mass as possible to the things we have observed...

...at the expense of the things we have not observed
A Shortcoming of MLE

For Naïve Bayes, suppose we never observe the word “unicorn” in a real news article.

In this case, what is the MLE of the following quantity?

\[
p(x_{\text{unicorn}} = 1 \mid y=\text{real}) = \bigcirc
\]

Recall:

\[
\theta_{k,0} = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \land x_{k}^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)}
\]

Now suppose we observe the word “unicorn” at test time. What is the posterior probability that the article was a real article?

\[
p(y = \text{real} \mid x) = \frac{\prod_{m=1}^{M} p(x_{m} \mid y=\text{real})}{p(x)} \cdot p(y = \text{real})
\]
Recipe for Closed-form MAP Estimation

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)
   \[ \theta \sim p(\theta) \text{ and then for all } i: x^{(i)} \sim p(x|\theta) \]

2. Write log-likelihood
   \[ \ell_{\text{MAP}}(\theta) = \log p(\theta) + \log p(x^{(1)}|\theta) + \ldots + \log p(x^{(N)}|\theta) \]

3. Compute partial derivatives (i.e. gradient)
   \[ \frac{\partial \ell_{\text{MAP}}(\theta)}{\partial \theta_1} = \ldots \]
   \[ \frac{\partial \ell_{\text{MAP}}(\theta)}{\partial \theta_2} = \ldots \]
   \[ \ldots \]
   \[ \frac{\partial \ell_{\text{MAP}}(\theta)}{\partial \theta_M} = \ldots \]

4. Set derivatives to zero and solve for \( \theta \)
   \[ \frac{\partial \ell_{\text{MAP}}(\theta)}{\partial \theta_m} = 0 \text{ for all } m \in \{1, \ldots, M\} \]
   \[ \theta_{\text{MAP}} = \text{solution to system of } M \text{ equations and } M \text{ variables} \]

5. Compute the second derivative and check that \( \ell(\theta) \) is concave down at \( \theta_{\text{MAP}} \)
Model 1: Bernoulli Naïve Bayes

MAP Estimation (Beta Prior)

1. Generative Story: The parameters are drawn once for the entire dataset.
   \[
   \theta_{y^{(i)}, m} \sim \text{Beta}(\alpha, \beta)
   \]
   for \( i \in \{1, \ldots, N\} \):
   \( y^{(i)} \sim \text{Bernoulli}(\phi) \)
   for \( m \in \{1, \ldots, M\} \):
   \( x_{m}^{(i)} \sim \text{Bernoulli}(\theta_{y^{(i)}, m}) \)

2. Likelihood:
   \[
   \ell_{MAP}(\phi, \theta) = \log \left[ \frac{p(\phi, \theta | \alpha, \beta) p(D | \phi, \theta)}{\prod_{m=1}^{M} p(\theta_{0, m} | \alpha, \beta)} \left( \prod_{i=1}^{N} p(x^{(i)}, y^{(i)} | \phi, \theta) \right) \right]
   \]

3. MAP Estimators:
   \[
   (\phi^{MAP}, \theta^{MAP}) = \arg \max_{\phi, \theta} \ell_{MAP}(\phi, \theta)
   \]
   Take derivatives, set to zero and solve...
   \[
   \phi = \frac{N_{y=1}}{N}
   \]
   \[
   \theta_{0, m} = \frac{(\alpha - 1) + N_{y=0, x_{m}=1}}{(\alpha - 1) + (\beta - 1) + N_{y=0}}
   \]
   \[
   \theta_{1, m} = \frac{(\alpha - 1) + N_{y=1, x_{m}=1}}{(\alpha - 1) + (\beta - 1) + N_{y=1}}
   \]
   \( \forall m \in \{1, \ldots, M\} \)
Other NB Models

1. **Bernoulli** Naïve Bayes:
   - for **binary features**

2. **Multinomial** Naïve Bayes:
   - for **integer features**

3. **Gaussian** Naïve Bayes:
   - for **continuous features**

4. **Multi-class** Naïve Bayes:
   - for classification problems with > 2 classes
   - **event model** could be any of Bernoulli, Gaussian, Multinomial, depending on features
Model 2: Multinomial Naïve Bayes

Support:
Option 1: Integer vector (word IDs)
\[
x = [x_1, x_2, \ldots, x_M] \quad \text{where} \quad x_m \in \{1, \ldots, K\} \quad \text{a word id.}
\]

Generative Story:
\[
\text{for } i \in \{1, \ldots, N\}: \quad \text{···}
\]
\[
y^{(i)} \sim \text{Bernoulli}(\phi)
\]
\[
\text{for } j \in \{1, \ldots, M_i\}:
\]
\[
x_{(i)j}^{(i)} \sim \text{Multinomial}(\theta_{y^{(i)}}, 1)
\]

Model:
\[
p_{\phi, \theta}(x, y) = p_{\phi}(y) \prod_{k=1}^{K} p_{\theta_k}(x_k | y)
\]
\[
= (\phi)^y (1 - \phi)^{(1-y)} \prod_{j=1}^{M_i} \theta_{y, x_j}
\]
Model 3: Gaussian Naïve Bayes

Support: \( \mathbf{x} \in \mathbb{R}^M \)

Model: Product of prior and the event model

\[
p(x, y) = p(x_1, \ldots, x_M, y) \\
= p(y) \prod_{k=1}^{M} p(x_k | y)
\]

Gaussian Naïve Bayes assumes that \( p(x_k | y) \) is given by a Normal distribution.
Model 4: Multiclass Naïve Bayes

Model:
The only change is that we permit $y$ to range over $C$ classes.

\[
p(x, y) = p(x_1, \ldots, x_K, y)
= p(y) \prod_{k=1}^{K} p(x_k | y)
\]

Now, $y \sim \text{Multinomial}(\phi, 1)$ and we have a separate conditional distribution $p(x_k | y)$ for each of the $C$ classes.
**Generic Naïve Bayes Model**

**Model**: Product of **prior** and the event model

\[
P(X, Y) = P(Y) \prod_{k=1}^{K} P(X_k|Y)
\]

**Support**: Depends on the choice of **event model** \(P(X_k|Y)\)

**Training**: Find the **class-conditional** MLE parameters

For \(P(Y)\), we find the MLE using all the data. For each \(P(X_k|Y)\) we condition on the data with the corresponding

**Classification**: Find the class that maximizes the posterior

\[
\hat{y} = \arg\max_y p(y|x)
\]
Naïve Bayes Model

**Classification:**

\[
\hat{y} = \arg\max_y p(y|x) \quad \text{(posterior)}
\]

\[
= \arg\max_y \frac{p(x|y)p(y)}{p(x)} \quad \text{(by Bayes’ rule)}
\]

\[
= \arg\max_y p(x|y)p(y)
\]
VISUALIZING GAUSSIAN NAÏVE BAYES
Fisher Iris Dataset

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

<table>
<thead>
<tr>
<th>Species</th>
<th>Sepal Length</th>
<th>Sepal Width</th>
<th>Petal Length</th>
<th>Petal Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.3</td>
<td>3.0</td>
<td>1.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>4.9</td>
<td>3.6</td>
<td>1.4</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>5.3</td>
<td>3.7</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>4.9</td>
<td>2.4</td>
<td>3.3</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>5.7</td>
<td>2.8</td>
<td>4.1</td>
<td>1.3</td>
</tr>
<tr>
<td>1</td>
<td>6.3</td>
<td>3.3</td>
<td>4.7</td>
<td>1.6</td>
</tr>
<tr>
<td>1</td>
<td>6.7</td>
<td>3.0</td>
<td>5.0</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set
Slide from William Cohen
$y = \hat{W}x$

$p(x_i | y = \text{blue})$

$p(x_i | y = \text{red})$

$y = \text{red}$

$y = \text{blue}$
Naïve Bayes has a **linear** decision boundary if variance (sigma) is constant across classes.
Iris Data (2 classes)
Iris Data (2 classes)

Classification with Naive Bayes

$\sigma^2_{red and blue} = 1$

variance = 1
Iris Data (2 classes)

Classification with Naive Bayes

variance learned for each class
Iris Data (3 classes)
Iris Data (3 classes)

Classification with Naive Bayes

variance = 1
Iris Data (3 classes)

Classification with Naive Bayes

variance learned for each class
One Pocket
One Pocket

Classification with Naive Bayes

variance learned for each class
One Pocket

Naive Bayes Distribution

\[ P(y = \text{red}|x) = 0.5 \]

variance learned for each class
Summary

1. Naïve Bayes provides a framework for generative modeling
2. Choose $p(x_m | y)$ appropriate to the data (e.g. Bernoulli for binary features, Gaussian for continuous features)
3. Train by MLE or MAP
4. Classify by maximizing the posterior
Learning Objectives

Naïve Bayes

You should be able to...

1. Write the generative story for Naive Bayes
2. Create a new Naïve 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 Naïves Bayes
9. Employ the method of Lagrange multipliers to find the MLE parameters of Multinomial Naïve Bayes
10. Describe how the variance affects whether a Gaussian Naive Bayes model will have a linear or nonlinear decision boundary
DISCRIMINATIVE AND GENERATIVE CLASSIFIERS
Generative vs. Discriminative

• **Generative Classifiers:**
  – Example: Naïve Bayes
  – Define a joint model of the observations $x$ and the labels $y$: $p(x, y)$
  – Learning maximizes (joint) likelihood
  – Use Bayes’ Rule to classify based on the posterior:
    \[
    p(y|x) = \frac{p(x|y)p(y)}{p(x)}
    \]

• **Discriminative Classifiers:**
  – Example: Logistic Regression
  – Directly model the conditional $p(y|x)$
  – Learning maximizes conditional likelihood
Generative vs. Discriminative

\[
\begin{align*}
\text{MLE} &: \prod_i p(x^{(i)}, y^{(i)} | \theta) \\
\text{MAP} &: p(\theta) \prod_i p(x^{(i)}, y^{(i)} | \theta)
\end{align*}
\]

\[
\begin{align*}
\text{Disc.} &: \prod_i p(y^{(i)} | x^{(i)}, \theta) \\
&: p(\theta) \prod_i p(y^{(i)} | x^{(i)}, \theta)
\end{align*}
\]
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 asymptotic error, and does better than Naïve Bayes
Slide courtesy of William Cohen
Naïve Bayes makes stronger assumptions about the data but needs fewer examples to estimate the parameters

Generative vs. Discriminative Learning (Parameter Estimation)

**Naïve Bayes:**
Parameters are decoupled $\Rightarrow$ Closed form solution for MLE

**Logistic Regression:**
Parameters are coupled $\Rightarrow$ 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 \(\rightarrow\) Beta prior (usually) pushes probabilities away from zero / one extremes

**Logistic Regression:**
Parameters are not probabilities \(\rightarrow\) 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 $x$ 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.
MOTIVATION: STRUCTURED PREDICTION
Structured Prediction

• Most of the models we’ve seen so far were for classification
  – Given observations: \( x = (x_1, x_2, ..., x_K) \)
  – Predict a (binary) label: \( y \)

• Many real-world problems require structured prediction
  – Given observations: \( x = (x_1, x_2, ..., x_K) \)
  – Predict a structure: \( y = (y_1, y_2, ..., y_J) \)

• 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: \[ \mathcal{D} = \{ \mathbf{x}^{(n)}, \mathbf{y}^{(n)} \}_{n=1}^{N} \]

<table>
<thead>
<tr>
<th>Sample 1:</th>
<th>Sample 2:</th>
<th>Sample 3:</th>
<th>Sample 4:</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>n</td>
<td>n</td>
<td>p</td>
</tr>
<tr>
<td>time</td>
<td>flies</td>
<td>flies</td>
<td>with</td>
</tr>
<tr>
<td>v</td>
<td>like</td>
<td>with</td>
<td>time</td>
</tr>
<tr>
<td>p</td>
<td>an</td>
<td>their</td>
<td>you</td>
</tr>
<tr>
<td>d</td>
<td>arrow</td>
<td>wings</td>
<td>will</td>
</tr>
<tr>
<td>n</td>
<td>arrow</td>
<td>v</td>
<td>see</td>
</tr>
</tbody>
</table>

Data:
- Sample 1: \( y^{(1)} = \{ \mathbf{y}^{(1)}, \mathbf{x}^{(1)} \} \)
- Sample 2: \( y^{(2)} = \{ \mathbf{y}^{(2)}, \mathbf{x}^{(2)} \} \)
- Sample 3: \( y^{(3)} = \{ \mathbf{y}^{(3)}, \mathbf{x}^{(3)} \} \)
- Sample 4: \( y^{(4)} = \{ \mathbf{y}^{(4)}, \mathbf{x}^{(4)} \} \)
Dataset for Supervised Handwriting Recognition

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

Sample 1:
- \( y^{(1)} = \{e, x, p, e, c, t, e, d\} \)
- \( x^{(1)} = \{U, n, e, x, p, e, t, c, e, d\} \)

Sample 2:
- \( y^{(2)} = \{v, o, l, i, c, a, n, i, c\} \)
- \( x^{(2)} = \{V, D, I, C, A, N, I, C\} \)

Sample 3:
- \( y^{(3)} = \{e, m, b, r, a, c, e, s\} \)
- \( x^{(3)} = \{E, m, b, r, a, c, e, s\} \)

Figures from (Chatzis & Demiris, 2013)
Dataset for Supervised Phoneme (Speech) Recognition

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

Sample 1:
- h#
- dh
- ih
- s
- w
- uh
- z
- iy
- z
- iy

Sample 2:
- f
- ao
- r
- ah
- s
- s
- s
- h#

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 constraint” on alignment
  - Phrases are disjoint (?)

(Burkett & Klein, 2012)
Congressional Voting

Application:

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

(Shtoyanov & 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 $x$ and labels $y$.
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
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

\[ \hat{y} = \arg\max_y p(y|x) \]
where \( y \in \{+1, -1\} \)

\[ p(y=+1|x) \]
\[ p(y=-1|x) \]

Classification

\[ |y| = 23 \]
\[ |\mathcal{Y}| = 40 \]
\[ |\hat{y}| = 40^{23} \]

Structured Prediction

\[ \hat{y} = \arg\max_{y \in \mathcal{Y}} p(y|x) \]
where \( y \in \mathcal{Y} \)
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.

(\textit{Inference} is usually called as a subroutine in learning.)

**Learning** tunes the parameters of the model.
Machine Learning

Data

Inference

(Inference is usually called as a subroutine in learning)

Model

Objective

Learning

(time flies like an arrow)

3 Alice saw Bob on a hill with a telescope
BACKGROUND
Background: Chain Rule of Probability

For random variables $A$ and $B$:

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

For random variables $X_1, X_2, X_3, X_4$:

$$P(X_1, X_2, X_3, X_4) = P(X_1|X_2, X_3, X_4) \cdot \frac{P(X_2|X_3, X_4)}{P(X_2)} \cdot \frac{P(X_3|X_4)}{P(X_3)} \cdot P(X_4)$$
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:

$$A \perp B|C$$

Later we will also write: $I<A, \{C\}, B>$
HIDDEN MARKOV MODEL (HMM)
From Mixture Model to HMM

“Naïve Bayes”:

\[ P(X, Y) = \prod_{t=1}^{T} P(X_t|Y_t)p(Y_t) \]

HMM:

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