Bayesian learning

• Bayesian learning uses probability to model data and quantify uncertainty of predictions
  – Facilitates incorporation of prior knowledge
  – Gives optimal predictions
    • Allows for decision-theoretic reasoning
Your first consulting job

• A billionaire from the suburbs of Manhattan asks you a question:
  – He says: I have thumbtack, if I flip it, what’s the probability it will fall with the nail up?
  – You say: Please flip it a few times:

  – You say: The probability is:
    • P(heads) = 3/5

  – He says: Why???
  – You say: Because...
Outline of lecture

• Review of probability
• Maximum likelihood estimation

2 examples of Bayesian classifiers:
• Naïve Bayes
• Logistic regression
Random Variables

• A random variable is some aspect of the world about which we (may) have uncertainty
  – R = Is it raining?
  – D = How long will it take to drive to work?
  – L = Where am I?

• We denote random variables with capital letters

• Random variables have domains
  – R in \{true, false\} (sometimes write as \{+r, ¬r\})
  – D in \[0, \infty\)
  – L in possible locations, maybe \{(0,0), (0,1), …\}
Probability Distributions

- Discrete random variables have distributions

\[
\begin{array}{c|c}
T & P \\
\hline
\text{warm} & 0.5 \\
\text{cold} & 0.5 \\
\end{array}
\quad
\begin{array}{c|c}
W & P \\
\hline
\text{sun} & 0.6 \\
\text{rain} & 0.1 \\
\text{fog} & 0.3 \\
\text{meteor} & 0.0 \\
\end{array}
\]

- A discrete distribution is a TABLE of probabilities of values
- The probability of a state (lower case) is a single number

\[
P(W = \text{rain}) = 0.1 \quad P(\text{rain}) = 0.1
\]

- Must have:

\[
\forall x \ P(x) \geq 0 \quad \sum_x P(x) = 1
\]
Joint Distributions

• A joint distribution over a set of random variables: $X_1, X_2, \ldots X_n$
specifies a real number for each assignment:

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$P(x_1, x_2, \ldots x_n)$$

– How many assignments if $n$ variables with domain sizes $d$?

– Must obey:

$$P(x_1, x_2, \ldots x_n) \geq 0$$

$$\sum_{(x_1, x_2, \ldots x_n)} P(x_1, x_2, \ldots x_n) = 1$$

• For all but the smallest distributions, impractical to write out or estimate
  – Instead, we make additional assumptions about the distribution

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>1</td>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Marginal Distributions

- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding

\[ P(T, W) \]

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

\[ P(T) \]

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>0.5</td>
</tr>
<tr>
<td>cold</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\[ P(W) \]

<table>
<thead>
<tr>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.6</td>
</tr>
<tr>
<td>rain</td>
<td>0.4</td>
</tr>
</tbody>
</table>

\[ P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2) \]
Conditional Probabilities

- A simple relation between joint and conditional probabilities
  - In fact, this is taken as the definition of a conditional probability

\[ P(a|b) = \frac{P(a, b)}{P(b)} \]

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

[Diagram of conditional probability with overlapping circles indicating \( P(a), P(b), P(a, b) \)]

\[ P(W = r|T = c) = ??? \]
Conditional Distributions

- Conditional distributions are probability distributions over some variables given fixed values of others.

**Conditional Distributions**

\[
P(W|T = hot) \quad \quad P(W|T = cold)
\]

<table>
<thead>
<tr>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.8</td>
</tr>
<tr>
<td>rain</td>
<td>0.2</td>
</tr>
</tbody>
</table>

\[
P(W|T = hot) \quad \quad P(W|T = cold)
\]

<table>
<thead>
<tr>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>rain</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**Joint Distribution**

\[
P(T, W)
\]

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>
The Product Rule

• Sometimes have conditional distributions but want the joint

\[ P(x|y) = \frac{P(x, y)}{P(y)} \quad \leftrightarrow \quad P(x, y) = P(x|y)P(y) \]

• Example:

| \( P(W) \) | \( P(D|W) \) | \( P(D, W) \) |
|---|---|---|
| W | P | D | W | P | D | W | P |
| sun | 0.8 | wet | sun | 0.1 | wet | sun | 0.08 |
| | | dry | sun | 0.9 | dry | sun | 0.72 |
| | | wet | rain | 0.7 | wet | rain | 0.14 |
| | | dry | rain | 0.3 | dry | rain | 0.06 |
Bayes’ Rule

- Two ways to factor a joint distribution over two variables:

\[ P(x, y) = P(x|y)P(y) = P(y|x)P(x) \]

- Dividing, we get:

\[ P(x|y) = \frac{P(y|x)}{P(y)} P(x) \]

- Why is this at all helpful?
  - Let’s us build one conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many practical systems (e.g. ASR, MT)

- In the running for most important ML equation!
Returning to thumbtack example...

- \( P(\text{Heads}) = \theta, \ P(\text{Tails}) = 1 - \theta \)

- Flips are \emph{i.i.d.}: \( D = \{x_i | i = 1 \ldots n\} \), \( P(D | \theta) = \prod_i P(x_i | \theta) \)
  - Independent events
  - Identically distributed according to Bernoulli distribution

- Sequence \( D \) of \( \alpha_H \) Heads and \( \alpha_T \) Tails

\[
P(D | \theta) = \theta^{\alpha_H} (1 - \theta)^{\alpha_T}
\]

Called the “likelihood” of the data under the model
Maximum Likelihood Estimation

• **Data:** Observed set $D$ of $\alpha_H$ Heads and $\alpha_T$ Tails

• **Hypothesis:** Bernoulli distribution

• **Learning:** finding $\theta$ is an optimization problem
  – What’s the objective function?
    $$P(D \mid \theta) = \theta^{\alpha_H} (1 - \theta)^{\alpha_T}$$

• **MLE:** Choose $\theta$ to maximize probability of $D$
  $$\hat{\theta} = \arg \max_\theta P(D \mid \theta) = \arg \max_\theta \ln P(D \mid \theta)$$
Your first parameter learning algorithm

\[ \hat{\theta} = \arg \max_\theta \ln P(\mathcal{D} \mid \theta) \]

\[ = \arg \max_\theta \ln \theta^{\alpha_H} (1 - \theta)^{\alpha_T} \]

- Set derivative to zero, and solve!

\[ \frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = \frac{d}{d\theta} \left[ \ln \theta^{\alpha_H} (1 - \theta)^{\alpha_T} \right] \]

\[ = \frac{d}{d\theta} \left[ \alpha_H \ln \theta + \alpha_T \ln(1 - \theta) \right] \]

\[ = \alpha_H \frac{d}{d\theta} \ln \theta + \alpha_T \frac{d}{d\theta} \ln(1 - \theta) \]

\[ = \frac{\alpha_H}{\theta} - \frac{\alpha_T}{1 - \theta} = 0 \]

\[ \hat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T} \]
Data

\[ L(\theta; \mathcal{D}) = \ln P(\mathcal{D}|\theta) \]

\[ \hat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T} \]
What if I have prior beliefs?

- Billionaire says: Wait, I know that the thumbtack is “close” to 50-50. What can you do for me now?
- You say: I can learn it the Bayesian way...
- Rather than estimating a single $\theta$, we obtain a distribution over possible values of $\theta$

In the beginning

\[
\text{Pr}(\theta) \quad \text{Beta}(2,2)
\]

Observe flips e.g.: {tails, tails}

After observations

\[
\text{Pr}(\theta \mid D) \quad \text{Beta}(3,2)
\]
Bayesian Learning

• Use Bayes’ rule!

\[ P(\theta \mid D) = \frac{P(D \mid \theta) P(\theta)}{P(D)} \]

Data Likelihood

Normalization

Posterior

Prior

• Or equivalently: \( P(\theta \mid D) \propto P(D \mid \theta) P(\theta) \)

• For uniform priors, this reduces to maximum likelihood estimation!

\[ P(\theta) \propto 1 \quad P(\theta \mid D) \propto P(D \mid \theta) \]
Bayesian Learning for Thumbtacks

\[ P(\theta | D) \propto P(D | \theta) P(\theta) \]

Likelihood:

\[ P(D | \theta) = \theta^{\alpha_H} (1 - \theta)^{\alpha_T} \]

• What should the prior be?
  – Represent expert knowledge
  – Simple posterior form

• For binary variables, commonly used prior is the Beta distribution:

\[ P(\theta) = \frac{\theta^{\beta_H-1} (1 - \theta)^{\beta_T-1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T) \]
Beta prior distribution – $P(\theta)$

$$P(\theta) = \frac{\theta^{\beta_H-1}(1 - \theta)^{\beta_T-1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$

- Since the Beta distribution is conjugate to the Bernoulli distribution, the posterior distribution has a particularly simple form:

$$P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$$

$$\propto \theta^{\alpha_H} (1 - \theta)^{\alpha_T} \theta^{\beta_H-1}(1 - \theta)^{\beta_T-1}$$

$$= \theta^{\alpha_H + \beta_H - 1} (1 - \theta)^{\alpha_T + \beta_t + 1}$$

$$= Beta(\alpha_H + \beta_H, \alpha_T + \beta_T)$$
Using Bayesian inference for prediction

- We now have a **distribution** over parameters
- For any specific \( f \), a function of interest, compute the expected value of \( f \):

\[
E[f(\theta)] = \int_0^1 f(\theta) P(\theta | D) \, d\theta
\]

- Integral is often hard to compute
- *As more data is observed, prior is more concentrated*
- **MAP (Maximum a posteriori approximation)**: use most likely parameter to approximate the expectation

\[
\hat{\theta} = \arg \max_\theta P(\theta | D)
\]

\[
E[f(\theta)] \approx f(\hat{\theta})
\]
What about continuous variables?

- Billionaire says: If I am measuring a continuous variable, what can you do for me?
- You say: Let me tell you about Gaussians...

\[ P(x \mid \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]
Some properties of Gaussians

• Affine transformation (multiplying by scalar and adding a constant) are Gaussian
  \[ X \sim N(\mu, \sigma^2) \]
  \[ Y = aX + b \rightarrow Y \sim N(a\mu + b, a^2\sigma^2) \]

• Sum of Gaussians is Gaussian
  \[ X \sim N(\mu_X, \sigma^2_X) \]
  \[ Y \sim N(\mu_Y, \sigma^2_Y) \]
  \[ Z = X + Y \rightarrow Z \sim N(\mu_X + \mu_Y, \sigma^2_X + \sigma^2_Y) \]

• Easy to differentiate, as we will see soon!
Learning a Gaussian

• Collect a bunch of data
  – Hopefully, i.i.d. samples
  – e.g., exam scores

• Learn parameters
  – $\mu$ (“mean”)
  – $\sigma$ (“variance”)

\[
P(x \mid \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]
**MLE for Gaussian:**

\[ P(x \mid \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]

- Prob. of i.i.d. samples \( D = \{x_1, \ldots, x_N\} \):
  \[
  \mu_{MLE}, \sigma_{MLE} = \arg \max_{\mu, \sigma} P(D \mid \mu, \sigma)
  \]
  \[
  P(D \mid \mu, \sigma) = \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^{N} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}
  \]

- Log-likelihood of data:
  \[
  \ln P(D \mid \mu, \sigma) = \ln \left[ \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^{N} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \right]
  \]
  \[
  = -N \ln \sigma \sqrt{2\pi} - \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2\sigma^2}
  \]
Your second learning algorithm: MLE for mean of a Gaussian

• What’s MLE for mean?

\[
\frac{d}{d\mu} \ln P(D \mid \mu, \sigma) = \frac{d}{d\mu} \left[ -N \ln \sigma \sqrt{2\pi} - \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2\sigma^2} \right]
\]

\[
= \frac{d}{d\mu} \left[ -N \ln \sigma \sqrt{2\pi} \right] - \sum_{i=1}^{N} \frac{d}{d\mu} \left[ \frac{(x_i - \mu)^2}{2\sigma^2} \right]
\]

\[
= \sum_{i=1}^{N} \frac{(x_i - \mu)}{\sigma^2} = 0
\]

\[
= \sum_{i=1}^{N} x_i - N\mu = 0
\]

\[
\hat{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]
MLE for variance

- Again, set derivative to zero:

\[
\frac{d}{d\sigma} \ln P(D | \mu, \sigma) = \frac{d}{d\sigma} \left[ -N \ln \sigma \sqrt{2\pi} - \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2\sigma^2} \right] \\
= \frac{d}{d\sigma} \left[ -N \ln \sigma \sqrt{2\pi} \right] - \sum_{i=1}^{N} \frac{d}{d\sigma} \left[ \frac{(x_i - \mu)^2}{2\sigma^2} \right] \\
= -\frac{N}{\sigma} + \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{\sigma^3} = 0
\]

\[
\hat{\sigma}^2_{MLE} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu})^2
\]
Learning Gaussian parameters

- **MLE:**
  \[
  \hat{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^{N} x_i
  \]
  \[
  \hat{\sigma}^2_{MLE} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu})^2
  \]

- MLE for the variance of a Gaussian is **biased**
  - Expected result of estimation is **not** true parameter!
  - Unbiased variance estimator:
  \[
  \hat{\sigma}^2_{unbiased} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \hat{\mu})^2
  \]
Bayesian learning of Gaussian parameters

• Conjugate priors
  – Mean: Gaussian prior
  – Variance: Wishart Distribution

• Prior for mean:

\[
P(\mu \mid \eta, \lambda) = \frac{1}{\lambda \sqrt{2\pi}} e^{-\frac{(\mu - \eta)^2}{2\lambda^2}}
\]
Outline of lecture

- Review of probability
- Maximum likelihood estimation

2 examples of Bayesian classifiers:
- Naïve Bayes
- Logistic regression
Bayesian Classification

• Problem statement:
  – Given features $X_1, X_2, ..., X_n$
  – Predict a label $Y$

[Next several slides adapted from: Vibhav Gogate, Jonathan Huang, Luke Zettlemoyer, Carlos Guestrin, and Dan Weld]
Example Application

- **Digit Recognition**

  - $X_1, \ldots, X_n \in \{0, 1\}$ (Black vs. White pixels)
  - $Y \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
The Bayes Classifier

• If we had the joint distribution on $X_1,...,X_n$ and $Y$, could predict using:

$$\arg \max_Y P(Y|X_1,\ldots,X_n)$$

  – (for example: what is the probability that the image represents a 5 given its pixels?)

• So ... How do we compute that?
The Bayes Classifier

• Use Bayes Rule!

\[ P(Y|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y)P(Y)}{P(X_1, \ldots, X_n)} \]

Likelihood  Prior

Normalization Constant

• Why did this help? Well, we think that we might be able to specify how features are “generated” by the class label
The Bayes Classifier

• Let’s expand this for our digit recognition task:

\[
P(Y = 5|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y = 5)P(Y = 5)}{P(X_1, \ldots, X_n|Y = 5)P(Y = 5) + P(X_1, \ldots, X_n|Y = 6)P(Y = 6)}
\]

\[
P(Y = 6|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y = 6)P(Y = 6)}{P(X_1, \ldots, X_n|Y = 5)P(Y = 5) + P(X_1, \ldots, X_n|Y = 6)P(Y = 6)}
\]

• To classify, we’ll simply compute these probabilities, one per class, and predict based on which one is largest
Model Parameters

• How many parameters are required to specify the likelihood, $P(X_1,\ldots,X_n|Y)$?
  – (Supposing that each image is 30x30 pixels)

• The problem with explicitly modeling $P(X_1,\ldots,X_n|Y)$ is that there are usually way too many parameters:
  – We’ll run out of space
  – We’ll run out of time
  – And we’ll need tons of training data (which is usually not available)
Naïve Bayes

• **Naïve Bayes assumption:**
  
  – Features are independent given class:

  \[ P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y) \]
  
  \[ = P(X_1|Y)P(X_2|Y) \]

  – More generally:

  \[ P(X_1\ldots X_n|Y) = \prod_{i} P(X_i|Y) \]

• **How many parameters now?**
  
  • Suppose \( X \) is composed of \( n \) binary features
The Naïve Bayes Classifier

• Given:
  – Prior $P(Y)$
  – $n$ conditionally independent features $X$ given the class $Y$
  – For each $X_i$, we have likelihood $P(X_i | Y)$

• Decision rule:

$$y^* = h_{NB}(x) = \arg \max_y P(y)P(x_1, \ldots, x_n | y)$$

$$= \arg \max_y P(y) \prod_i P(x_i | y)$$

If certain assumption holds, NB is optimal classifier!
(they typically don’t)
A Digit Recognizer

- **Input:** pixel grids

- **Output:** a digit 0-9

Are the naïve Bayes assumptions realistic here?
What has to be learned?

\[ P(Y) \]

\begin{array}{|c|c|}
\hline
1 & 0.1 \\
2 & 0.1 \\
3 & 0.1 \\
4 & 0.1 \\
5 & 0.1 \\
6 & 0.1 \\
7 & 0.1 \\
8 & 0.1 \\
9 & 0.1 \\
0 & 0.1 \\
\hline
\end{array}

\[ P(F_{3,1} = \text{on}|Y) \quad P(F_{5,5} = \text{on}|Y) \]

\begin{array}{|c|c|}
\hline
1 & 0.01 \\
2 & 0.05 \\
3 & 0.05 \\
4 & 0.30 \\
5 & 0.80 \\
6 & 0.90 \\
7 & 0.05 \\
8 & 0.60 \\
9 & 0.50 \\
0 & 0.80 \\
\hline
\end{array}

\begin{array}{|c|c|}
\hline
1 & 0.05 \\
2 & 0.01 \\
3 & 0.90 \\
4 & 0.80 \\
5 & 0.90 \\
6 & 0.90 \\
7 & 0.25 \\
8 & 0.85 \\
9 & 0.60 \\
0 & 0.80 \\
\hline
\end{array}
MLE for the parameters of NB

• Given dataset
  – Count(A=a,B=b) ← number of examples where A=a and B=b

• MLE for discrete NB, simply:
  – Prior:
    \[ P(Y = y) = \frac{\text{Count}(Y = y)}{\sum_{y'} \text{Count}(Y = y')} \]
  – Observation distribution:
    \[ P(X_i = x | Y = y) = \frac{\text{Count}(X_i = x, Y = y)}{\sum_{x'} \text{Count}(X_i = x', Y = y)} \]
MLE for the parameters of NB

- Training amounts to, for each of the classes, averaging all of the examples together:
MAP estimation for NB

• Given dataset
  – Count(a, b) ← number of examples where A=a and B=b

• MAP estimation for discrete NB, simply:
  – Prior:
    \[ P(Y = y) = \frac{\text{Count}(Y = y)}{\sum_{y'} \text{Count}(Y = y')} \]
  – Observation distribution:
    \[ P(X_i = x | Y = y) = \frac{\text{Count}(X_i = x, Y = y) + a}{\sum_{x'} \text{Count}(X_i = x', Y = y) + |X_i| \cdot a} \]

• Called “smoothing”. Corresponds to Dirichlet prior!
What about if there is missing data?

- One of the key strengths of Bayesian approaches is that they can naturally handle missing data
  - Suppose don’t have value for some attribute $X_i$
    - applicant’s credit history unknown
    - some medical test not performed on patient
    - how to compute $P(X_1=x_1 \ldots X_j=? \ldots X_d=x_d \mid y)$
  - Easy with Naïve Bayes
    - ignore attribute in instance where its value is missing
    - compute likelihood based on observed attributes
    - no need to “fill in” or explicitly model missing values
    - based on conditional independence between attributes

$P(x_1 \ldots X_j \ldots x_d \mid y) = \prod_{i \neq j} P(x_i \mid y)$

[Slide from Victor Lavrenko and Nigel Goddard]
Naive Bayes = Linear Classifier

Theorem: assume that $x_i \in \{0, 1\}$ for all $i \in [1, N]$. Then, the Naive Bayes classifier is defined by

$$x \mapsto \text{sgn}(w \cdot x + b),$$
Outline of lecture

- Review of probability
- Maximum likelihood estimation

2 examples of Bayesian classifiers:
- Naïve Bayes
- Logistic regression

[Next several slides adapted from: Vibhav Gogate, Luke Zettlemoyer, Carlos Guestrin, and Dan Weld]
Logistic Regression

- Learn $P(Y|X)$ directly!
  - Assume a particular functional form
  - Linear classifier? On one side we say $P(Y=1|X)=1$, and on the other $P(Y=1|X)=0$
  - But, this is not differentiable (hard to learn)… doesn’t allow for label noise…
Logistic Regression

- Learn $P(Y|X)$ directly!
  - Assume a particular functional form
  - Sigmoid applied to a linear function of the data:

\[
P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}
\]
\[
P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}
\]

Logistic function (Sigmoid):

Features can be discrete or continuous!
Logistic Function in n Dimensions

\[ P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)} \]

Sigmoid applied to a linear function of the data:

Features can be discrete or continuous!
Logistic Regression: decision boundary

\[ P(Y = 1 | X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_iX_i)} \]

\[ P(Y = 0 | X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_iX_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_iX_i)} \]

- **Prediction**: Output the Y with highest \( P(Y|X) \)
  - For binary Y, output Y=0 if
    \[ 1 < \frac{P(Y = 0 | X)}{P(Y = 1 | X)} \]
    \[ 1 < \exp(w_0 + \sum_{i=1}^{n} w_iX_i) \]
    \[ 0 < w_0 + \sum_{i=1}^{n} w_iX_i \]

A Linear Classifier!
Generative (Naïve Bayes) maximizes **Data likelihood**

\[ \ln P(D | w) = \sum_{j=1}^{N} \ln P(x^j, y^j | w) \]

\[ = \sum_{j=1}^{N} \ln P(y^j | x^j, w) + \sum_{j=1}^{N} \ln P(x^j | w) \]

**Discriminative (Logistic Regr.)** maximizes **Conditional Data Likelihood**

\[ \ln P(D_Y | D_X, w) = \sum_{j=1}^{N} \ln P(y^j | x^j, w) \]

Focuses only on learning \( P(Y|X) \) - all that matters for classification
Maximizing Conditional Log Likelihood

\[ l(w) \equiv \ln \prod_j P(y_j^j|x_j^j, w) \]

\[ = \sum_j y_j^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i^n w_i x_i^j)) \]

Bad news: no closed-form solution to maximize \( l(w) \)

Good news: \( l(w) \) is concave function of \( w \)

No local maxima

Concave functions easy to optimize
Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave →

\[
\nabla_w l(w) = \left[ \frac{\partial l(w)}{\partial w_0}, \ldots, \frac{\partial l(w)}{\partial w_n} \right]'
\]

Learning rate, \( \eta > 0 \)

Update rule:

\[
\Delta w = \eta \nabla_w l(w)
\]

\[
w_i(t+1) \leftarrow w_i(t) + \eta \frac{\partial l(w)}{\partial w_i}
\]
Maximize Conditional Log Likelihood: Gradient ascent

\[ P(Y = 1|X, W) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)} \]

\[ l(w) = \sum_j y^j (w_0 + \sum_i^n w_i x^j_i) - \ln(1 + \exp(w_0 + \sum_i^n w_i x^j_i)) \]

\[ \frac{\partial l(w)}{\partial w_i} = \sum_j \left[ \frac{\partial}{\partial w_i} y^j (w_0 + \sum_i^n w_i x^j_i) - \frac{\partial}{\partial w_i} \ln \left(1 + \exp(w_0 + \sum_i^n w_i x^j_i)\right) \right] \]

\[ = \sum_j \left[ y^j x^j_i - \frac{x^j_i \exp(w_0 + \sum_i^n w_i x^j_i)}{1 + \exp(w_0 + \sum_i^n w_i x^j_i)} \right] \]

\[ = \sum_j x^j_i \left[ y^j - \frac{\exp(w_0 + \sum_i^n w_i x^j_i)}{1 + \exp(w_0 + \sum_i^n w_i x^j_i)} \right] \]

\[ \frac{\partial l(w)}{\partial w_i} = \sum_j x^j_i \left( y^j - P(Y^j = 1|x^j, w) \right) \]
Gradient Ascent for LR

Gradient ascent algorithm: (learning rate $\eta > 0$)

do:

$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y_j - \hat{P}(Y_j = 1 | x^j, w)]$

For $i=1$ to $n$: (iterate over features)

$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y_j - \hat{P}(Y_j = 1 | x^j, w)]$

until "change" < $\varepsilon$

Loop over training examples!
Naïve Bayes vs. Logistic Regression

**Learning:** $h: X \mapsto Y$

- $X$ – features
- $Y$ – target classes

**Generative**
- Assume functional form for
  - $P(X|Y)$ assume cond indep
  - $P(Y)$
  - Est. params from train data
- Gaussian NB for cont. features
- Bayes rule to calc. $P(Y|X=x)$:
  - $P(Y | X) \propto P(X | Y) P(Y)$
- **Indirect** computation
  - Can generate a sample of the data
  - Can easily handle missing data

**Discriminative**
- Assume functional form for
  - $P(Y|X)$ no assumptions
  - Est params from training data
- Handles discrete & cont features
  - Directly calculate $P(Y|X=x)$
  - Can’t generate data sample
Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

• Generative vs. Discriminative classifiers
• Asymptotic comparison
  (# training examples $\rightarrow$ infinity)
  – when model correct
    • NB, Linear Discriminant Analysis (with class independent variances), and Logistic Regression produce identical classifiers

  – when model incorrect
    • LR is less biased – does not assume conditional independence
      – therefore LR expected to outperform NB
Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

• Generative vs. Discriminative classifiers
• Non-asymptotic analysis
  – convergence rate of parameter estimates,
    \( (n = \# \text{ of attributes in } X) \)
    • Size of training data to get close to infinite data solution
    • Naïve Bayes needs \( O(\log n) \) samples
    • Logistic Regression needs \( O(n) \) samples

  – Naïve Bayes converges more quickly to its (perhaps less helpful) asymptotic estimates
Some experiments from UCI data sets

Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. $m$ (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naïve Bayes.