Naïve Bayes (Continued)
Naïve Bayes with Continuous (variables)
Logistic Regression

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

January 30th, 2006

©2006 Carlos Guestrin
Announcements

- Recitations stay on Thursdays
  - 5-6:30pm in Wean 5409
  - This week: Naïve Bayes & Logistic Regression

- **Extension** for the first homework:
  - Due Wed. Feb 8\textsuperscript{th} beginning of class
  - Mitchell’s chapter is most useful reading

- Go to the AI seminar:
  - Tuesdays 3:30pm, Wean 5409
  - [http://www.cs.cmu.edu/~aiseminar/](http://www.cs.cmu.edu/~aiseminar/)
  - This week’s seminar very relevant to what we are covering in class
Classification

- **Learn**: $h: X \mapsto Y$
  - $X$ – features
  - $Y$ – target classes

- Suppose you know $P(Y|X)$ exactly, how should you classify?
  - Bayes classifier:
    $$y^* = h_{Bayes}(x) = \arg\max_y P(Y=y | X=x)$$

- Why?
Optimal classification

- **Theorem:** Bayes classifier $h_{\text{Bayes}}$ is optimal!

  \[
  \text{if you know } p(x|y) \quad y^* = h_{\text{Bayes}}(x) = \operatorname*{arg\ max}_y P(y = y | x = x)
  \]

  **That is** $\text{error}_{\text{true}}(h_{\text{Bayes}}) \leq \text{error}_{\text{true}}(h), \forall h(x)$

- **Proof:**

  \[P(\text{error}) = \int_X p(\text{error} | x) \, dx = \int_X p(\text{error} | x) \cdot p(x) \, dx\]

  \[P(\text{error} | x) = \begin{cases} p(y = t | x) ; h(x) = f & 0.2 \\ p(y = f | x) ; h(x) = t & 0.8 \end{cases}\]

  Solve classification by learning $p(y | x)$
How hard is it to learn the optimal classifier?

Data =

- How do we represent these? How many parameters?
  - Prior, P(Y):
    - Suppose Y is composed of \( k \) classes
      \[ p(Y) \]
      \[ \begin{array}{cccc}
      A & B & C & D \\
      0.4 & 0.4 & 0.15 & 0.5 \\
      \end{array} \]
      \( K-1 \) parameters
  - Likelihood, P(X|Y):
    - Suppose X is composed of \( n \) binary features
      \[ p(X|Y) \]
      \[ \begin{array}{cc}
      y & f \\
      0.8 & 0.1 \\
      f & 0.1 & 0 \\
      \end{array} \]
      \[ 2^n - 1 \] parameters
      \[ p(X|Y) \leq K(2^n-1) \] "

- Complex model \( \rightarrow \) High variance with limited data!!!
Conditional Independence

- X is **conditionally independent** of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

\[ (\forall i, j, k) P(X = i | Y = j, Z = k) = P(X = i | Z = k) \]

- e.g., \( P(\text{Thunder} | \text{Rain, Lightning}) = P(\text{Thunder} | \text{Lightning}) \)

    **Conditioned on L, T or R are indep.**

- Equivalent to:

\[
P(X, Y | Z) = P(X | Z)P(Y | Z)
\]
The Naïve Bayes assumption

- 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
    \[
    p(X_i|Y) = k \left(2^{-1}\right) \quad ; \quad p(X|Y) = nk
    \]
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:

\[ y^* = h_{NB}(x) = \arg \max_y P(y) P(x_1, \ldots, x_n \mid y) \]

\[ = \arg \max_y P(y) \prod_i P(x_i \mid y) \]

If assumption holds, NB is optimal classifier!
MLE for the parameters of NB

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

MLE for NB, simply:
- Prior: \( P(Y=y) = \frac{\text{Count}(Y=y)}{N} \)

- Likelihood: \( P(X_i=x_i | Y_i=y_i) = \frac{\text{Count}(X_i=x_i, Y_i=y_i)}{\text{Count}(Y_i=y_i)} \)
Subtleties of NB classifier 1 – Violating the NB assumption

- Usually, features are not conditionally independent:

\[ P(X_1 \ldots X_n|Y) \neq \prod P(X_i|Y) \]

- Thus, in NB, actual probabilities \( P(Y|X) \) often biased towards 0 or 1 (see homework 1)

- Nonetheless, NB is the single most used classifier out there
  - NB often performs well, even when assumption is violated
  - [Domingos & Pazzani ’96] discuss some conditions for good performance
Subtleties of NB classifier 2 – Insufficient training data

- What if you never see a training instance where $X_1=a$ when $Y=b$?
  - e.g., $Y=${SpamEmail}, $X_1=${‘Enlargement’}
  - $P(X_1=a \mid Y=b) = 0 = \frac{\text{Count}(X_1=a, Y=b)}{\text{Count}(Y=b)} = 0$

- Thus, no matter what the values $X_2, \ldots, X_n$ take:
  - $P(Y=b \mid X_1=a, X_2, \ldots, X_n) = 0$
    - set it to $\epsilon$ (hack!)

- What now???
**MAP for Beta distribution**

\[
P(\theta \mid D) = \frac{\theta^{\beta_H + \alpha_H - 1} (1 - \theta)^{\beta_T + \alpha_T - 1}}{B(\beta_H + \alpha_H, \beta_T + \alpha_T)} \sim \text{Beta}(\beta_H + \alpha_H, \beta_T + \alpha_T)
\]

- **MAP**: use most likely parameter:

\[
\hat{\theta} = \arg \max_{\theta} P(\theta \mid D) = \frac{\beta_H + \alpha_H - 1}{\beta_H + \alpha_H + \beta_T + \alpha_T - 2}
\]

- Beta prior equivalent to extra thumbtack flips
- As \( N \to \infty \), prior is “forgotten”
- **But, for small sample size, prior is important!**
Bayesian learning for NB parameters – a.k.a. smoothing

- Dataset of \( N \) examples
- Prior
  - “distribution” \( Q(X_i, Y) \), \( Q(Y) \)
  - \( m \) “virtual” examples

- MAP estimate
  \[
  P(X_i | Y) \triangleq \frac{\text{count} \left( X_i = a_i, Y = b \right) + m \cdot Q(Y = b, X_i = a)}{\text{count} \left( Y = b \right) + m \cdot Q(Y = b)}
  \]

- Now, even if you never observe a feature/class, posterior probability never zero

\[\text{e.g., uniform} \Rightarrow Q(Y) = \begin{cases} 0.5 \\ 0.5 \end{cases} \]
Text classification

- Classify e-mails
  - $Y = \{\text{Spam}, \text{NotSpam}\}$

- Classify news articles
  - $Y = \{\text{what is the topic of the article?}\}$

- Classify webpages
  - $Y = \{\text{Student, professor, project, …}\}$

- What about the features $X$?
  - The text!

\[ P(Y|X) \]

\[ \text{What's } X? \]
Features $\mathbf{X}$ are entire document – $X_i$ for $i$th word in article

Article from rec.sport.hockey

Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year’s biggest and worst (opinion)
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most obvious candidate for pleasant surprise is Alex Zhitnik. He came highly touted as a defensive defenseman, but he’s clearly much more than that. Great skater and hard shot (though wish he were more accurate). In fact, he pretty much allowed the Kings to trade away that huge defensive liability Paul Coffey. Kelly Hrudey is only the biggest disappointment if you thought he was any good to begin with. But, at best, he’s only a mediocre goaltender. A better choice would be Tomas Sandstrom, though not through any fault of his own, but because some thugs in Toronto decided
NB for Text classification

- P(X|Y) is huge!!!
  - Article at least 1000 words, X={X₁,…,X₁₀₀₀}
  - Xᵢ represents iᵗʰ word in document, i.e., the domain of Xᵢ is entire vocabulary, e.g., Webster Dictionary (or more), 10,000 words, etc.

- NB assumption helps a lot!!!
  - P(Xᵢ=xᵢ|Y=y) is just the probability of observing word xᵢ in a document on topic y
  \[ P(Xᵢ=xᵢ|Y=y) \rightarrow 10^{9000-1} \text{ parameters} \]
  \[ P(Xᵢ=xᵢ|Y=y) \leq P(Xᵢ+1=xᵢ+1|Y=y) \]
  \[ h_{NB}(x) = \arg \max_y P(y) \prod_{i=1}^{\text{LengthDoc}} P(xᵢ|y) \]
Bag of words model

- Typical additional assumption – **Position in document doesn’t matter**: $P(X_i=x_i|Y=y) = P(X_k=x_i|Y=y)$
  - “Bag of words” model – order of words on the page ignored
  - Sounds really silly, but often works very well!

\[
P(y) \prod_{i=1}^{\text{LengthDoc}} P(x_i|y)
\]

*When the lecture is over, remember to wake up the person sitting next to you in the lecture room.*
Bag of words model

- Typical additional assumption – **Position in document doesn’t matter**: \( P(X_i=x_i|Y=y) = P(X_k=x_i|Y=y) \)
  - “Bag of words” model – order of words on the page ignored
  - Sounds really silly, but often works very well!

\[
P(y) \prod_{i=1}^{\text{Length Doc}} P(x_i|y)
\]

Grammar often improves results...
Bag of Words Approach

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.
NB with Bag of Words for text classification

- Learning phase:
  - Prior \( P(Y) = \frac{\text{count}(Y=y)}{N} \)
    - Count how many documents you have from each topic (+ prior)
  - \( P(X_i|Y) \)
    - For each topic, count how many times you saw word in documents of this topic (+ prior)

- Test phase:
  - For each document
    - Use naïve Bayes decision rule

\[
h_{NB}(x) = \arg \max_y P(y) \prod_{i=1}^{\text{LengthDoc}} P(x_i|y)
\]
Twenty News Groups results

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

<table>
<thead>
<tr>
<th>comp.graphics</th>
<th>misc.forsale</th>
</tr>
</thead>
<tbody>
<tr>
<td>comp.os.ms-windows.missc</td>
<td>rec.autos</td>
</tr>
<tr>
<td>comp.sys.ibm.pc.hardware</td>
<td>rec.motorcycles</td>
</tr>
<tr>
<td>comp.sys.mac.hardware</td>
<td>rec.sport.baseball</td>
</tr>
<tr>
<td>comp.windows.x</td>
<td>rec.sport.hockey</td>
</tr>
<tr>
<td>alt.atheism</td>
<td>sci.space</td>
</tr>
<tr>
<td>soc.religion.christian</td>
<td>sci.crypt</td>
</tr>
<tr>
<td>talk.religion.misc</td>
<td>sci.electronics</td>
</tr>
<tr>
<td>talk.politics.mideast</td>
<td>sci.med</td>
</tr>
<tr>
<td>talk.politics.misc</td>
<td></td>
</tr>
<tr>
<td>talk.politics.guns</td>
<td></td>
</tr>
</tbody>
</table>

Naive Bayes: 89% classification accuracy
Learning curve for Twenty News Groups

Accuracy vs. Training set size (1/3 withheld for test)
What if we have continuous $X_i$?

Eg., character recognition: $X_i$ is $i^{th}$ pixel

Gaussian Na"ive Bayes (GNB):

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik} \sqrt{2\pi}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

Sometimes assume variance
- is independent of $Y$ (i.e., $\sigma_i$),
- or independent of $X_i$ (i.e., $\sigma_k$)
- or both (i.e., $\sigma$)
Estimating Parameters:

\( Y \) discrete, \( X_i \) continuous

Maximum likelihood estimates:

\[
\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)
\]

\[
\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k) - 1} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)
\]

\( \delta(x) = 1 \text{ if } x \text{ true, else } 0 \)
Example: GNB for classifying mental states

~1 mm resolution
~2 images per sec.
15,000 voxels/image
non-invasive, safe

measures Blood Oxygen Level Dependent (BOLD) response

Typical impulse response
Brain scans can track activation with precision and sensitivity

[Mitchell et al.]
Gaussian Naïve Bayes: Learned $\mu_{\text{voxel,word}}$

$P(\text{BrainActivity} \mid \text{WordCategory} = \{\text{People, Animal}\})$

[Mitchell et al.]
Learned Bayes Models – Means for $P(\text{BrainActivity} \mid \text{WordCategory})$

Pairwise classification accuracy: 85%

People words

Animal words

[Mitchell et al.]
What you need to know about Naïve Bayes

- Types of learning problems
  - Learning is (just) function approximation!
- Optimal decision using Bayes Classifier
- Naïve Bayes classifier
  - What’s the assumption
  - Why we use it
  - How do we learn it
  - Why is Bayesian estimation important
- Text classification
  - Bag of words model
- Gaussian NB
  - Features are still conditionally independent
  - Each feature has a Gaussian distribution given class

©2006 Carlos Guestrin
Generative v. Discriminative classifiers – Intuition

- **Want to Learn:** $h: X \mapsto Y$
  - $X$ – features
  - $Y$ – target classes

- **Bayes optimal classifier** – $P(Y|X)$

- **Generative classifier**, e.g., Naïve Bayes:
  - Assume some **functional form for** $P(X|Y)$, $P(Y)$
  - Estimate parameters of $P(X|Y)$, $P(Y)$ directly from training data
  - Use Bayes rule to calculate $P(Y|X= x)$
  - This is a ‘**generative**’ model
    - **Indirect** computation of $P(Y|X)$ through Bayes rule
    - But, **can generate a sample of the data**, $P(X) = \sum_y P(y) P(X|y)$

- **Discriminative classifiers**, e.g., Logistic Regression:
  - Assume some **functional form for** $P(Y|X)$
  - Estimate parameters of $P(Y|X)$ directly from training data
  - This is the ‘**discriminative**’ model
    - **Directly learn** $P(Y|X)$
    - But **cannot obtain a sample of the data**, because $P(X)$ is not available
Logistic Regression

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

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

$$P(Y = 0|X) = 1 - P(Y = 1|X)$$

Logistic function (or Sigmoid):

$$g(z) = \frac{1}{1 + \exp(-z)}$$
Understanding the sigmoid

\[ g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{-(w_0 + \sum_i w_i x_i)}} \]

\[ w_0=2, w_1=1 \]
\[ w_0=0, w_1=1 \]
\[ w_0=0, w_1=0.5 \]
Logistic Regression – a Linear classifier

\[ g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{-(w_0 + \sum_i w_i x_i)}} \]
Very convenient!

\[ P(Y = 1|X = \langle X_1, \ldots, X_n \rangle) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)} \]

implies

\[ P(Y = 0|X = \langle X_1, \ldots, X_n \rangle) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)} \]

implies

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

implies

\[ \ln \frac{P(Y = 0|X)}{P(Y = 1|X)} = w_0 + \sum_i w_i X_i \]

linear classification rule!
Logistic regression more generally

- Logistic regression in more general case, where $Y \in \{Y_1 \ldots Y_R\}$: learn $R-1$ sets of weights

For $k<R$

$$P(Y = y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^{n} w_{ki}X_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}$$

For $k=R$ (normalization, so no weights for this class)

$$P(Y = y_R|X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}$$

Features can be discrete or continuous!
Logistic regression v. Naïve Bayes

Consider learning \( f : X \rightarrow Y \), where
- \( X \) is a vector of real-valued features, \( < X_1 \ldots X_n > \)
- \( Y \) is boolean

Could use a Gaussian Naïve Bayes classifier
- assume all \( X_i \) are conditionally independent given \( Y \)
- model \( P(X_i \mid Y = y_k) \) as Gaussian \( N(\mu_{ik}, \sigma_i) \)
- model \( P(Y) \) as Bernoulli(\( \theta, 1-\theta \))

What does that imply about the form of \( P(Y\mid X) \)?
Logistic regression v. Naïve Bayes

- Consider learning $f: X \rightarrow Y$, where
  - $X$ is a vector of real-valued features, $< X_1 \ldots X_n >$
  - $Y$ is boolean

- Could use a Gaussian Naïve Bayes classifier
  - assume all $X_i$ are conditionally independent given $Y$
  - model $P(X_i | Y = y_k)$ as Gaussian $N(\mu_{ik}, \sigma_i)$
  - model $P(Y)$ as Bernoulli($\theta, 1-\theta$)

- What does that imply about the form of $P(Y|X)$?

$$P(Y = 1 | X =< X_1, \ldots X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

Cool!!!!!
Derive form for $P(Y|X)$ for continuous $X_i$

$$P(Y = 1|X) = \frac{P(Y = 1)P(X|Y = 1)}{P(Y = 1)P(X|Y = 1) + P(Y = 0)P(X|Y = 0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp\left(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}\right)}$$

$$= \frac{1}{1 + \exp\left(\ln \frac{1}{\theta} \right) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}\text{ constant w.r.t. } X}$$
Ratio of class-conditional probabilities

\[ \ln \frac{P(X_i | Y = 0)}{P(X_i | Y = 1)} \rightarrow P(X_i = x | Y = y_k) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x - \mu_{ik})^2}{2\sigma_i^2}} \]

\[ = \ln \frac{1}{\sqrt{2\pi} \sigma_i} e^{-\frac{(x - \mu_i)^2}{2\sigma_i^2}} - \frac{(x - \mu_i)^2}{2\sigma_i^2} + \frac{(x - \mu_{il})^2}{2\sigma_i^2} \]

\[ = \ln \frac{1}{\sqrt{2\pi} \sigma_i} e^{-\frac{(x - \mu_i)^2}{2\sigma_i^2}} + \frac{\mu_i x}{\sigma_i^2} + \frac{\mu_{il} x}{2\sigma_i^2} - \frac{\mu_i^2}{2\sigma_i^2} \]

\[ = \frac{\mu_{il} - \mu_i}{\sigma_i^2} x + \frac{\mu_{il}^2 - \mu_i^2}{2\sigma_i^2} \]
Derive form for \( P(Y|X) \) for continuous \( X_i \)

\[
P(Y = 1|X) = \frac{P(Y = 1)P(X|Y = 1)}{P(Y = 1)P(X|Y = 1) + P(Y = 0)P(X|Y = 0)}
\]

\[
= \frac{1}{1 + \exp(\ln \frac{1-\theta}{\theta} + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}
\]

\[
= \frac{1}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)}
\]

\[
\omega_i = \frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2}
\]

\[
w_0 = \ln \frac{1-\theta}{\theta} + \sum \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}
\]
Gaussian Naïve Bayes v. Logistic Regression

- Representation equivalence
  - But only in a special case!!! (GNB with class-independent variances)
- But what’s the difference???
- LR makes no assumptions about $P(X|Y)$ in learning!!!
- Loss function!!!
  - Optimize different functions $\rightarrow$ Obtain different solutions
Loss functions: Likelihood v. Conditional Likelihood

- **Generative (Naïve Bayes) Loss function:**
  
  **Data likelihood**
  \[
  \ln P(\mathcal{D} | w) = \sum_{j=1}^{N} \ln P(x^j, y^j | w)
  \]
  
  \[
  = \sum_{j=1}^{N} \ln P(y^j | x^j, w) + \sum_{j=1}^{N} \ln P(x^j | w)
  \]

- Discriminative models cannot compute \(P(x^j | w)\)!

- But, discriminative (logistic regression) loss function:

  **Conditional Data Likelihood**
  \[
  \ln P(\mathcal{D}_Y | \mathcal{D}_X, w) = \sum_{j=1}^{N} \ln P(y^j | x^j, w)
  \]

  - Doesn’t waste effort learning \(P(X)\) – focuses on \(P(Y|X)\) all that matters for classification
Expressing Conditional Log Likelihood

\[ l(w) \equiv \sum_j \ln P(y^j|x^j, w) \]

\[ P(Y = 0|X, w) = \frac{1}{1 + \exp(w_0 + \sum_i w_i x_i)} \]

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

\[ l(w) = \sum_j y^j \ln P(y^j = 1|x^j, w) + (1 - y^j) \ln P(y^j = 0|x^j, w) \]
Maximizing Conditional Log Likelihood

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

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

Good news: \( l(w) \) is concave function of \( w \) \( \rightarrow \) no locally optimal solutions

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

Good news: concave functions easy to optimize
Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave
  \[ \rightarrow \text{Find optimum with gradient ascent} \]

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

Update rule: \[ \Delta w = \eta \nabla_w l(w) \]

\[ w_i \leftarrow w_i + \eta \frac{\partial l(w)}{\partial w_i} \]

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent much better (see reading)
Maximize Conditional Log Likelihood: Gradient ascent

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

Gradient ascent algorithm: iterate until change < \( \varepsilon \)

For all \( i \), \( w_i \leftarrow w_i + \eta \sum_j x_i^j [y_j^j - \hat{P}(Y^j = 1 \mid x_i^j, w)] \)

repeat
That’s all M(C)LE. How about MAP?

\[ p(w \mid Y, X) \propto P(Y \mid X, w)p(w) \]

- One common approach is to define priors on \( w \)
  - Normal distribution, zero mean, identity covariance
  - “Pushes” parameters towards zero
- Corresponds to **Regularization**
  - Helps avoid very large weights and overfitting
  - Explore this in your homework
  - More on this later in the semester

- MAP estimate

\[
w^* = \arg\max_w \ln \left[ p(w) \prod_{j=1}^N P(y^j \mid x^j, w) \right]
\]
Gradient of M(C)AP

\[
\frac{\partial}{\partial w_i} \ln \left[ p(w) \prod_{j=1}^{N} P(y^j \mid x^j, w) \right]
\]

\[
p(w) = \prod_i \frac{1}{\kappa \sqrt{2\pi}} \frac{-w_i^2}{e^{2\kappa^2}}
\]
MLE vs MAP

- Maximum conditional likelihood estimate

\[ w^* = \arg \max_w \ln \left( \prod_{j=1}^{N} P(y^j \mid x^j, w) \right) \]

\[ w_i \leftarrow w_i + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid x^j, w)] \]

- Maximum conditional a posteriori estimate

\[ w^* = \arg \max_w \ln \left( p(w) \prod_{j=1}^{N} P(y^j \mid x^j, w) \right) \]

\[ w_i \leftarrow w_i + \eta \left\{ -\lambda w_i + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid x^j, w)] \right\} \]
What you should know about Logistic Regression (LR)

- Gaussian Naïve Bayes with class-independent variances representationally equivalent to LR
  - Solution differs because of objective (loss) function
- In general, NB and LR make different assumptions
  - NB: Features independent given class $\rightarrow$ assumption on $P(X|Y)$
  - LR: Functional form of $P(Y|X)$, no assumption on $P(X|Y)$
- LR is a linear classifier
  - decision rule is a hyperplane
- LR optimized by conditional likelihood
  - no closed-form solution
  - concave $\rightarrow$ global optimum with gradient ascent
  - Maximum conditional a posteriori corresponds to regularization
Acknowledgements

- Some of the material is the presentation is courtesy of Tom Mitchell