Today:
• Naïve Bayes – Big Picture
• Logistic regression
• Gradient ascent
• Generative – discriminative classifiers

Readings:
Required:
• Mitchell: “Naïve Bayes and Logistic Regression” (see class website)

Optional
• Ng and Jordan paper (class website)

Gaussian Naïve Bayes – Big Picture
Logistic Regression

Idea:
• Naïve Bayes allows computing P(Y|X) by learning P(Y) and P(X|Y)
• Why not learn P(Y|X) directly?

Consider learning f: X → Y, where
• X is a vector of real-valued features, < X₁ ... Xₙ >
• Y is boolean
• assume all Xᵢ are conditionally independent given Y
• model P(Xᵢ | Y = yₖ) as Gaussian N(μᵢₖ, σᵢₖ)
• model P(Y) as Bernoulli (π)

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

\[ P(Y = 1 | X = < X₁, ..., Xₙ >) = \frac{1}{1 + exp(w₀ + \sumᵢ wᵢXᵢ)} \]
Derive form for $P(Y|X)$ for continuous $X_i$

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

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

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

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

$$P(x_i|y_i) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}}$$

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

Very convenient!

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

implies

$$P(Y = 0|X = < X_1, \ldots, X_n >) =$$

implies

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

implies

$$\ln \frac{P(Y = 0|X)}{P(Y = 1|X)} =$$
Very convenient!

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

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

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

Logistic regression more generally

- Logistic regression when Y not boolean (but still discrete-valued).
- Now \( y \in \{y_1, \ldots, y_R\} \) : 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 \( k = R \):

\[ P(Y = y_R|X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji} X_i)} \]
Training Logistic Regression: MCLE

- we have $L$ training examples: $\{(X^1, Y^1), \ldots, (X^L, Y^L)\}$

- maximum likelihood estimate for parameters $W$
  
  $W_{\text{MLE}} = \arg \max_W P(\langle X^1, Y^1 \rangle \ldots \langle X^L, Y^L \rangle | W) = \arg \max_W \prod_i P(\langle X^i, Y^i \rangle | W)$

- maximum conditional likelihood estimate

Training Logistic Regression: MCLE

- Choose parameters $W = \langle w_0, \ldots, w_n \rangle$ to maximize conditional likelihood of training data

  where

  $P(Y = 0 | X, W) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$

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

- Training data $D = \{(X^1, Y^1), \ldots, (X^L, Y^L)\}$

- Data likelihood $= \prod_i P(X^i, Y^i | W)$

- Data conditional likelihood $= \prod_i P(Y^i | X^i, W)$

  $W_{\text{MCLE}} = \arg \max_W \prod_i P(Y^i | W, X^i)$
Expressing Conditional Log Likelihood

\[ l(W) = \ln \prod_l P(Y^l | X^l, W) = \sum_l \ln P(Y^l | X^l, W) \]

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

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

\[ l(W) = \sum_l Y^l \ln P(Y^l = 1|X^l, W) + (1 - Y^l) \ln P(Y^l = 0|X^l, W) \]

\[ = \sum_l Y^l \ln \frac{P(Y^l = 1|X^l, W)}{P(Y^l = 0|X^l, W)} + \ln P(Y^l = 0|X^l, W) \]

\[ = \sum_l Y^l (w_0 + \sum_i w_i X^l_i) - \ln(1 + \exp(w_0 + \sum_i w_i X^l_i)) \]

Maximizing Conditional Log Likelihood

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

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

\[ l(W) = \ln \prod_l P(Y^l | X^l, W) \]

\[ = \sum_l Y^l (w_0 + \sum_i w_i X^l_i) - \ln(1 + \exp(w_0 + \sum_i w_i X^l_i)) \]

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

Bad news: no closed-form solution to maximize \( l(W) \)
Maximize Conditional Log Likelihood: Gradient Ascent

\[ l(W) \equiv \ln \prod_l P(Y^l | X^l, W) \]

\[ = \sum_l y^l(w_0 + \sum_i w_i x_i^l) - \ln(1 + \exp(w_0 + \sum_i w_i x_i^l)) \]

\[ \frac{\partial l(W)}{\partial w_i} = \sum_l x_i^l (y^l - \hat{P}(Y^l = 1 | X^l, W)) \]
Maximize Conditional Log Likelihood: Gradient Ascent

\[ l(W) \equiv \ln \prod_l P(Y^l|X^l, W) \]
\[ = \sum_l Y^l(w_0 + w_i X_i^l) - \ln(1 + \exp(w_0 + \sum_i w_i X_i^l)) \]

\[ \frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W)) \]

Gradient ascent algorithm: iterate until change < \( \varepsilon \)
For all \( i \), repeat
\[ w_i \leftarrow w_i + \eta \sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W)) \]

That’s all for M(C)LE. How about MAP?

- One common approach is to define priors on \( W \)
  - Normal distribution, zero mean, identity covariance
- Helps avoid very large weights and overfitting
- MAP estimate

\[ W \leftarrow \arg \max_W \ln P(W) \prod_l P(Y^l|X^l, W) \]

- Let’s assume Gaussian prior: \( W \sim \mathcal{N}(0, \sigma) \)
MLE vs MAP

- Maximum conditional likelihood estimate

\[ W \leftarrow \arg \max _{W} \ln \prod _{l} P(Y^l|X^l, W) \]

\[ w_i \leftarrow w_i + \eta \sum X_i (Y^l - \hat{P}(Y^l = 1|X^l, W)) \]

- Maximum a posteriori estimate with prior \( W \sim \mathcal{N}(0, \sigma I) \)

\[ W \leftarrow \arg \max _{W} \ln [P(W) \prod _{l} P(Y^l|X^l, W)] \]

\[ w_i \leftarrow w_i - \eta \lambda w_i + \eta \sum X_i (Y^l - \hat{P}(Y^l = 1|X^l, W)) \]

MAP estimates and Regularization

- Maximum a posteriori estimate with prior \( W \sim \mathcal{N}(0, \sigma I) \)

\[ W \leftarrow \arg \max _{W} \ln [P(W) \prod _{l} P(Y^l|X^l, W)] \]

\[ w_i \leftarrow w_i - \eta \lambda w_i + \eta \sum X_i (Y^l - \hat{P}(Y^l = 1|X^l, W)) \]

called a “regularization” term

- helps reduce overfitting, especially when training data is sparse
- keep weights nearer to zero (if \( P(W) \) is zero mean Gaussian prior), or whatever the prior suggests
- used very frequently in Logistic Regression
The Bottom Line

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

• Then \( P(Y|X) \) is of this form, and we can directly estimate \( W \)

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

• Furthermore, same holds if the \( X_i \) are boolean
  • trying proving that to yourself

Generative vs. Discriminative Classifiers

Training classifiers involves estimating \( f: X \rightarrow Y, \text{ or } P(Y|X) \)

Generative classifiers (e.g., Naïve Bayes)
  • Assume some functional form for \( P(X|Y), P(X) \)
  • Estimate parameters of \( P(X|Y), P(X) \) directly from training data
  • Use Bayes rule to calculate \( P(Y|X= x_i) \)

Discriminative classifiers (e.g., Logistic regression)
  • Assume some functional form for \( P(Y|X) \)
  • Estimate parameters of \( P(Y|X) \) directly from training data
Use Naïve Bayes or Logistic Regression?

Consider

• Restrictiveness of modeling assumptions

• Rate of convergence (in amount of training data) toward asymptotic hypothesis
  – i.e., the learning curve

Naïve Bayes vs Logistic Regression

Consider $Y$ boolean, $X_i$ continuous, $X=\langle X_1 \ldots X_n \rangle$

Number of parameters to estimate:

• NB:

• LR:

$$P(Y=0|X,W) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$P(Y=1|X,W) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}$$
Naïve Bayes vs Logistic Regression

Consider Y boolean, $X_i$ continuous, $X=<X_1 \ldots X_n>$

Number of parameters:
- NB: $4n +1$
- LR: $n+1$

Estimation method:
- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

G. Naïve Bayes vs. Logistic Regression

- Generative and Discriminative classifiers

- Asymptotic comparison (# training examples $\rightarrow$ infinity)
  - when conditional independence assumptions correct
    - GNB, LR produce identical classifiers
  - when conditional independence assumptions incorrect
    - LR is less biased – does not assume cond indep.
    - therefore expected to outperform GNB when both given infinite training data

[Ng & Jordan, 2002]
Naïve Bayes vs. Logistic Regression

- Generative and Discriminative classifiers

- *Non-asymptotic analysis* (see [Ng & Jordan, 2002])
  - convergence rate of parameter estimates – how many training examples needed to assure good estimates?
    - GNB order $\log n$ (where $n =$ # of attributes in $X$)
    - LR order $n$

GNB converges more quickly to its (perhaps less accurate) asymptotic estimates

Informally: because LR’s parameter estimates are coupled, but GNB’s are not

Some experiments from UCI data sets

[Ng & Jordan, 2002]
Summary: Naïve Bayes and Logistic Regression

• Modeling assumptions
  – Naïve Bayes more biased (cond. indep)
  – Both learn linear decision surfaces
• Convergence rate \((n=\text{number training examples})\)
  – Naïve Bayes \(\sim O(\log n)\)
  – Logistic regression \(\sim O(n)\)
• Bottom line
  – Naïve Bayes converges faster to its (potentially too restricted) final hypothesis

What you should know:

• Logistic regression
  – Functional form follows from Naïve Bayes assumptions
    • For Gaussian Naïve Bayes assuming variance \(\sigma_{ik} = \sigma_i\)
    • For discrete-valued Naïve Bayes too
  – But training procedure picks parameters without the conditional independence assumption
  – MLE training: pick \(W\) to maximize \(P(Y | X, W)\)
  – MAP training: pick \(W\) to maximize \(P(W | X,Y)\)
    • regularization: e.g., \(P(W) \sim N(0,\sigma)\)
    • helps reduce overfitting
• Gradient ascent/descent
  – General approach when closed-form solutions for MLE, MAP are unavailable
• Generative vs. Discriminative classifiers
  – Bias vs. variance tradeoff