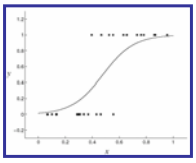
Machine Learning

10-701/15-781, Spring 2008

Logistic Regression

-- generative verses discriminative classifier



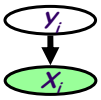
Eric Xing

Lecture 5, January 30, 2006

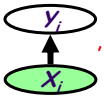
Reading: Chap. 3.1.3-4 CB

Generative vs. Discriminative Classifiers

- Goal: Wish to learn f: X $\rightarrow$ Y, e.g., P(Y|X)
- Generative classifiers (e.g., Naïve Bayes):
  - Assume some functional form for P(X|Y), P(Y)
    This is a '***generative***' model of the data!
  - Estimate parameters of P(X|Y), P(Y) directly from training data
  - Use Bayes rule to calculate P(Y|X= xi)

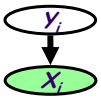


- Discriminative classifiers:
  - Directly assume some functional form for P(Y|X)
    This is a '***discriminative***' model of the data!
  - Estimate parameters of P(Y|X) directly from training data



Consider the a Gaussian Generative Classifier

- learning f: X → Y, where
  - X is a vector of real-valued features, < $X^1 \ldots X^m$ >
  - Y is boolean
- What does that imply about the form of P(Y|X)?



  - The joint probability of a datum and its label is:

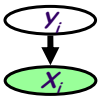
$$p(x_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times p(x_n \mid y_n^k = 1, \mu, \sigma)$$
$$= \pi_k \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ \frac{1}{2\sigma^2} (x_n - \mu_k)^2 \right\}$$

  - Given a datum x_n, we predict its label using the conditional probability of the label given the datum:

$$p(y_n^k = 1 \mid x_n, \mu, \sigma) = \frac{\pi_k \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ \frac{1}{2\sigma^2} (x_n - \mu_k)^2 \right\}}{\sum_{k'} \pi_{k'} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ \frac{1}{2\sigma^2} (x_n - \mu_{k'})^2 \right\}}$$

Naïve Bayes Classifier

- When X is multivariate-Gaussian vector:

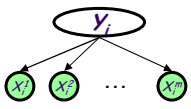


  - The joint probability of a datum and it label is:

$$p(\vec{x}_n, y_n^k = 1 \mid \vec{\mu}, \Sigma) = p(y_n^k = 1) \times p(\vec{x}_n \mid y_n^k = 1, \vec{\mu}, \Sigma)$$
$$= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{ -\frac{1}{2} (\vec{x}_n - \vec{\mu}_k)^T \Sigma^{-1} (\vec{x}_n - \vec{\mu}_k) \right\}$$

  - The naïve Bayes simplification

$$p(x_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times \prod_j p(x_n^j \mid y_n^k = 1, \mu_{k,j}, \sigma_{k,j})$$
$$= \pi_k \prod_j \frac{1}{(2\pi\sigma_{k,j}^2)^{1/2}} \exp\left\{ \frac{1}{2\sigma_{k,j}^2} (x_n^j - \mu_{k,j})^2 \right\}$$



  - More generally: $p(x_n, y_n \mid \eta, \pi) = p(y_n \mid \pi) \times \prod_{j=1}^{m} p(x_n^j \mid y_n, \eta)$
    - Where $p(. \mid .)$ is an arbitrary conditional (discrete or continuous) 1-D density

The predictive distribution

- Understanding the predictive distribution

$$p(y_n^k = 1 \mid x_n, \bar{\mu}, \Sigma, \pi) = \frac{p(y_n^k = 1, x_n \mid \bar{\mu}, \Sigma, \pi)}{p(x_n \mid \bar{\mu}, \Sigma)} = \frac{\pi_k N(x_n, \mid \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} N(x_n, \mid \mu_{k'}, \Sigma_{k'})} \quad *$$

- Under naïve Bayes assumption:

$$p(y_n^k = 1 \mid x_n, \bar{\mu}, \Sigma, \pi) = \frac{\pi_k \exp\left\{-\sum_j \left(\frac{1}{2\sigma_{k,j}^2}(x_n^j - \mu_k^j)^2 - \log \sigma_{k,j} - C\right)\right\}}{\sum_{k'} \pi_{k'} \exp\left\{-\sum_j \left(\frac{1}{2\sigma_{k',j}^2}(x_n^j - \mu_{k'}^j)^2 - \log \sigma_{k',j} - C\right)\right\}} \quad **$$

- For two class (i.e., *K*=2), and when the two classes haves the same variance, ** turns out to be the logistic function

$$p(y_n^1 = 1 \mid x_n) = \frac{1}{1 + \frac{\pi_2 \exp\left\{-\sum_j \left(\frac{1}{2\sigma_j^2}(x_n^j - \mu_2^j)^2 - \log \sigma_j - C\right)\right\}}{\pi_1 \exp\left\{-\sum_j \left(\frac{1}{2\sigma_j^2}(x_n^j - \mu_1^j)^2 - \log \sigma_j - C\right)\right\}}} = \frac{1}{1 + \exp\left\{-\sum_j \left(x_n^j \frac{1}{\sigma_j^2}(\mu_1^j - \mu_2^j) + \frac{1}{\sigma_j^2}([\mu_1^j]^2 - [\mu_2^j]^2)\right) + \log \frac{(1-\pi_1)}{\pi_1}\right\}}$$

$$= \frac{1}{1 + e^{-\theta^T x_n}}$$

$\Sigma x_n^j \cdot \theta_j + \theta_0$

$\rightarrow \theta^T x_n$

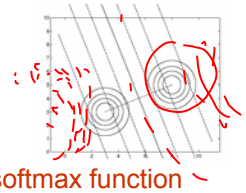
The decision boundary

- The predictive distribution

$$p(y_n^1 = 1 \mid x_n) = \frac{1}{1 + \exp\left\{-\sum_{j=1}^{M} \theta_j x_n^j - \theta_0\right\}} = \frac{1}{1 + e^{-\theta^T x_n}}$$

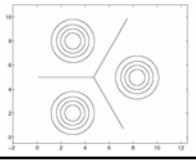
- The Bayes decision rule:

$$\ln \frac{p(y_n^1 = 1 \mid x_n)}{p(y_n^2 = 1 \mid x_n)} = \ln \left(\frac{\frac{1}{1 + e^{-\theta^T x_n}}}{\frac{e^{-\theta^T x_n}}{1 + e^{-\theta^T x_n}}} \right) = \theta^T x_n$$



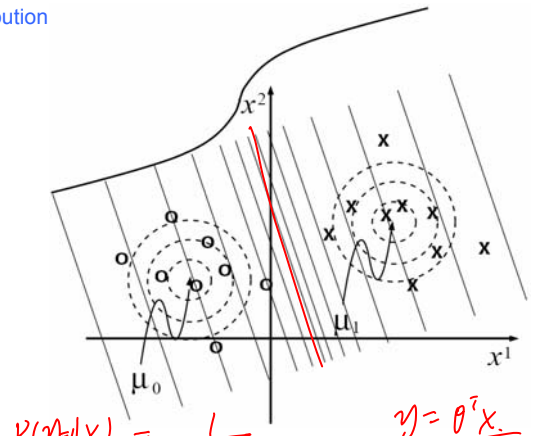
- For multiple class (i.e., *K*>2), * correspond to a softmax function

$$p(y_n^k = 1 \mid x_n) = \frac{e^{-\theta_k^T x_n}}{\sum_j e^{-\theta_j^T x_n}}$$



Discussion: Generative and discriminative classifiers

- Generative:
  - Modeling the joint distribution of all data
- Discriminative:
  - Modeling only points at the boundary
  - How? Regression!



$P(y=1|X) = \frac{1}{1+e^{-\theta^T x}}$

$y = \theta^T X$

$\theta^* = \arg\min J(\theta)$

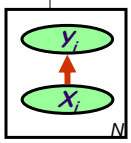
Linear regression

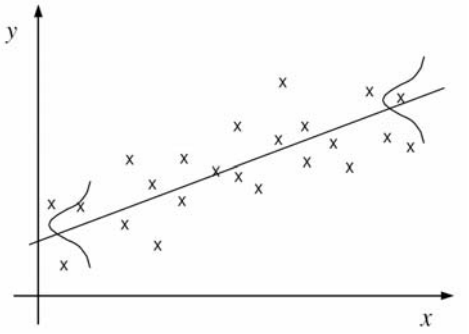
- The data:

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_N, y_N)\}$$

- Both nodes are observed:
  - X is an input vector
  - Y is a response vector

  (we first consider y as a generic continuous response vector, then we consider the special case of classification where y is a discrete indicator)
- A regression scheme can be used to model $p(y|x)$ directly, rather than $p(x,y)$





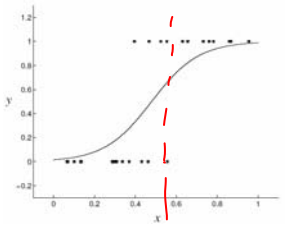
Logistic regression (sigmoid classifier)

- The condition distribution: a Bernoulli

$$p(y \mid x) = \mu(x)^{y}(1-\mu(x))^{1-y}$$

where μ is a logistic function

$$\mu(x) = \frac{1}{1+e^{-\theta^{T}x}}$$



- We can used the brute-force gradient method as in LR

- But we can also apply generic laws by observing the $p(y|x)$ is an exponential family function, more specifically, a generalized linear model (see future lectures …)

Training Logistic Regression: MCLE

- Estimate parameters θ=<θ_0, θ_1, ... θ_m> to maximize the **conditional likelihood** of training data

- Training data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$

- Data likelihood = $\prod_{i=1}^{N} P(x_i, y_i; \theta)$

- Data conditional likelihood = $\prod_{i=1}^{N} P(x_i|y_i; \theta)$

$$\theta = \arg\max_{\theta} \ln \prod_{i} P(y_i|x_i; \theta)$$

Expressing Conditional Log Likelihood

$$l(\theta) \equiv \ln \prod_i P(y_i|x_i;\theta) = \sum_i \ln P(y_i|x_i;\theta)$$

- Recall the logistic function: $\mu = \frac{1}{1-e^{-\theta^t x}}$

  and conditional likelihood: $P(y|x) = \mu(x)^y(1-\mu(x))^{1-y}$

$$\begin{aligned} l(\theta) = \sum_i \ln P(y_i|x_i;\theta) &= \sum_i y_i \ln u(x_i) + (1-y_i)\ln(1-\mu(x_i)) \\ &= \sum_i y_i \ln \frac{u(x_i)}{1-\mu(x_i)} + \ln(1-\mu(x_i)) \\ &= \sum_i y_i\theta^t x_i - \theta^t x_i + \ln(1-e^{-\theta^t x_i}) \\ &= \sum_i (y_i-1)\theta^t x_i + \ln(1-e^{-\theta^t x_i}) \end{aligned}$$

Maximizing Conditional Log Likelihood

- The objective:

$$\begin{aligned} l(\theta) &= \ln \prod_i P(y_i|x_i;\theta) \\ &= \sum_i (y_i-1)\theta^t x_i + \ln(1-e^{-\theta^t x_i}) \end{aligned}$$

- Good news: *l(θ)* is concave function of *θ*

- Bad news: no closed-form solution to maximize *l(θ)*

Gradient Ascent

$$l(\theta) = \ln \prod_i P(y_i|x_i;\theta)$$
$$= \sum_i (y_i - 1)\theta^t x_i + \ln(1 - e^{-\theta^t x_i}) = \sum_i (y_i - 1)\theta^t x_i - \ln \mu(\theta^t x_i)$$

- Property of sigmoid function:

$$\mu = \frac{1}{1 + e^{-t}} \qquad \frac{d\mu}{dt} = \mu(1 - \mu)$$

- The gradient:

$$\frac{\partial l(\theta)}{\partial \theta_j} =$$

The gradient ascent algorithm iterate until change < ε

For all *i*, $\theta_j \leftarrow \theta_j + \eta \sum_i (y_i - P(y_i = 0|x_i;\theta))x_i^j$

repeat

How about MAP?

- It is very common to use *regularized* maximum likelihood.
- One common approach is to define priors on θ
  - – Normal distribution, zero mean, identity covariance
  - Helps avoid very large weights and overfitting

$$P(y|x;\theta) = \mu(\theta^t x)^y (1 - \mu(\theta^t x))^{1-y}$$

$$P(\theta) = \text{Normal}(0, \lambda^{-1} I)$$

$$l'(\theta) = \sum_i (y_i - 1)\theta^t x_i - \ln \mu(\theta^t x_i) - \frac{\lambda}{2}\theta^t \theta$$

- MAP estimate

MLE vs MAP

- Maximum conditional likelihood estimate

$$\theta = \arg\max_{\theta} \ln \prod_i P(y_i|x_i;\theta)$$

$$\theta_j \leftarrow \theta_j + \eta \sum_i (y_i - P(y_i = 0|x_i;\theta))x_i^j$$

- Maximum a posteriori estimate

$$\theta = \arg\max_{\theta} \ln p(\theta) \prod_i P(y_i|x_i;\theta)$$

$$\theta_j \leftarrow \theta_j + \eta\Big(\sum_i (y_i - P(y_i = 0|x_i;\theta))x_i^j - \lambda\theta_j\Big)$$

Logistic regression: practical issues

- IRLS takes $O(Nd^3)$ per iteration, where N = number of training cases and d = dimension of input x.
- Quasi-Newton methods, that approximate the Hessian, work faster.
- Conjugate gradient takes $O(Nd)$ per iteration, and usually works best in practice.
- Stochastic gradient descent can also be used if N is large c.f. perceptron rule:

Naïve Bayes vs Logistic Regression

- Consider Y boolean, X continuous, X=<X^1 ... X^n>
- Number of parameters to estimate:

  NB:

  LR:

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

Naïve Bayes vs Logistic Regression

- Asymptotic comparison (# training examples → infinity)
- when model assumptions correct
  - NB, LR produce identical classifiers
- when model assumptions incorrect
  - LR is less biased – does not assume conditional independence
  - therefore expected to outperform NB

Naïve Bayes vs Logistic Regression

- Non-asymptotic analysis (see [Ng & Jordan, 2002])
- convergence rate of parameter estimates – how many training examples needed to assure good estimates?

  NB order log n (where n = # of attributes in X)

  LR order n
- NB converges more quickly to its (perhaps less helpful) asymptotic estimates

Rate of convergence: logistic regression

- Let $h_{Dis,m}$ be logistic regression trained on m examples in n dimensions. Then with high probability:

$$\epsilon(h_{Dis,m}) \leq \epsilon(h_{Dis,\infty}) + O\left(\sqrt{\frac{n}{m}\log nm}\right)$$

- Implication: if we want $\epsilon(h_{Dis,m}) \leq \epsilon(h_{Dis,\infty}) + \epsilon_0$

  for some small constant ε_0, it suffices to pick order n examples

  → Convergences to its asymptotic classifier, in order n examples

  - result follows from Vapnik's structural risk bound, plus fact that the "VC Dimension" of an n-dimensional linear separators is n

Rate of covergence: naïve Bayes parameters

- Let any ε_1, δ>0, and any $n \geq 0$ be fixed.
  Assume that for some fixed $\rho > 0$,
  we have that $\rho_0 \leq p(y = T) \leq 1 - \rho$

- Let $m = O((1/\epsilon_1^2)\log(n/\delta))$

- Then we probability at least 1-δ, after m examples:

1. For discrete input, $|\hat{p}(x_i|y=b) - p(x_i|y=b)| \leq \epsilon_1$ $|\hat{p}(y=b) - p(y=b)| \leq \epsilon_1$ for all i and b
2. For continuous inputs, $|\hat{\mu}_{i|y=b} - \mu_{i|y=b}| \leq \epsilon_1$ $|\hat{\sigma}^2_{i|y=b} - \sigma^2_{i|y=b}| \leq \epsilon_1$ for all i and b

Some experiments from UCI data sets

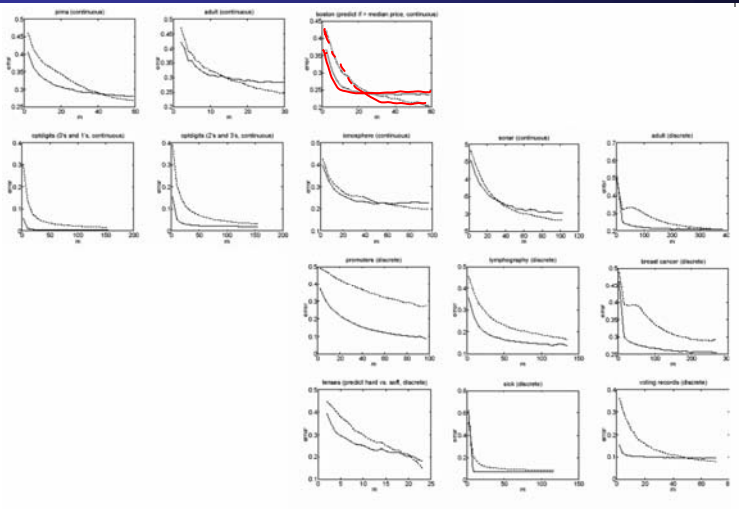


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. m (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

Summary

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