

Logistic Regression

Aarti Singh

Co-instructor: Barnabas Poczos

Machine Learning 10-401

Feb 2, 2016



MACHINE LEARNING DEPARTMENT



Logistic Regression

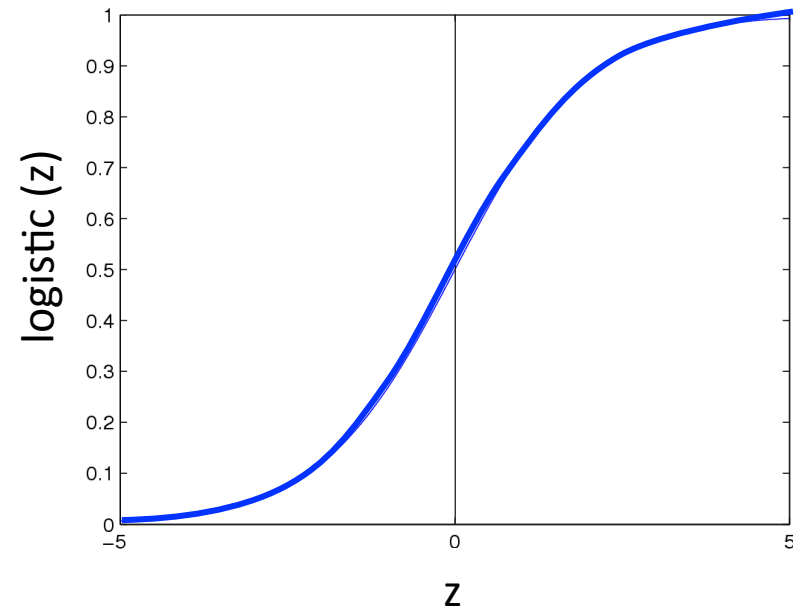
Not really regression

Assumes the following functional form for $P(Y|X)$:

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

Logistic function applied to a linear function of the data

Logistic
function
(or Sigmoid): $\frac{1}{1 + \exp(-z)}$



Features can be discrete or continuous!

Logistic Regression is a Linear Classifier!

Assumes the following functional form for $P(Y|X)$:

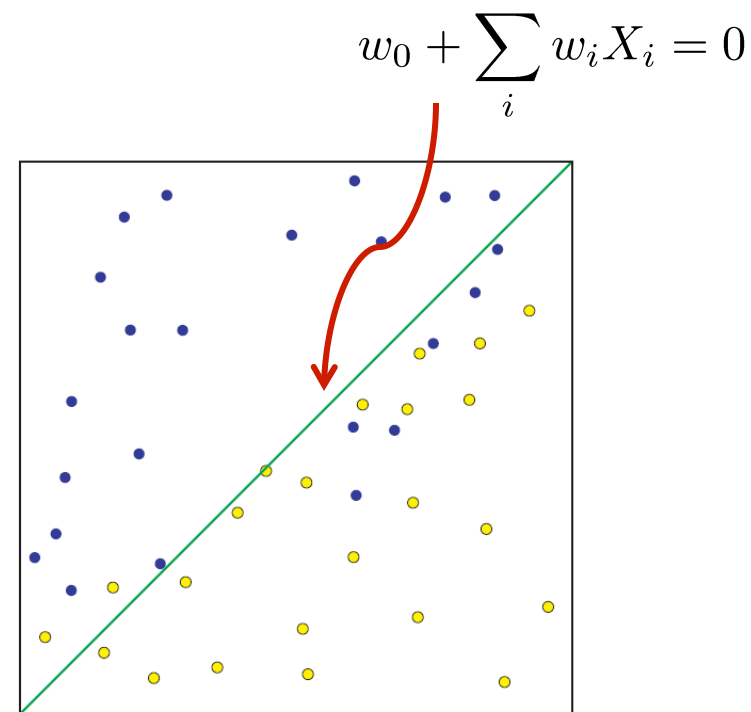
$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

Decision boundary: Note - Labels are 0,1

$$P(Y = 0|X) \geq P(Y = 1|X)$$

$$w_0 + \sum_i w_i X_i \geq 0$$

(Linear Decision Boundary)



Training Logistic Regression

How to learn the parameters w_0, w_1, \dots, w_d ? (d features)

Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^n$ $X^{(j)} = (X_1^{(j)}, \dots, X_d^{(j)})$

Maximum (Conditional) Likelihood Estimates

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \ln \left[\prod_{j=1}^n P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

Maximum (Conditional) A Posteriori Estimates

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

Discriminative philosophy – Don't waste effort learning $P(X)$, focus on $P(Y|X)$ – that's all that matters for classification!

Optimizing concave function

- M(C)LE and M(C)AP for Logistic Regression are concave objectives
- Maximum of a concave function can be reached by

Gradient Ascent Algorithm

Initialize: Pick \mathbf{w} at random

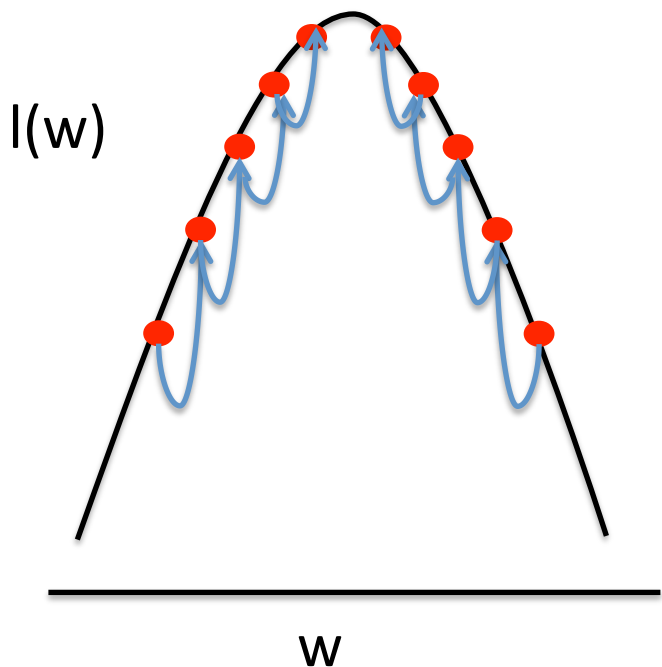
Gradient:

$$\nabla_{\mathbf{w}} l(\mathbf{w}) = \left[\frac{\partial l(\mathbf{w})}{\partial w_0}, \dots, \frac{\partial l(\mathbf{w})}{\partial w_d} \right]'$$

Update rule: ↖ Learning rate, $\eta > 0$

$$\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left. \frac{\partial l(\mathbf{w})}{\partial w_i} \right|_t$$



M(C)LE vs. M(C)AP

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \ln \left[\prod_{j=1}^n P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

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

- Maximum conditional a posteriori estimate

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

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\frac{1}{\kappa^2} w_i^{(t)} + \sum_j x_i^j [y^j - P(Y = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})] \right\}$$

Logistic Regression for more than 2 classes

- Logistic regression in more general case, where $Y \in \{y_1, \dots, y_K\}$

for $k < K$

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

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

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

Predict $f^*(x) = \arg \max_{Y=y} P(Y = y | X = x)$

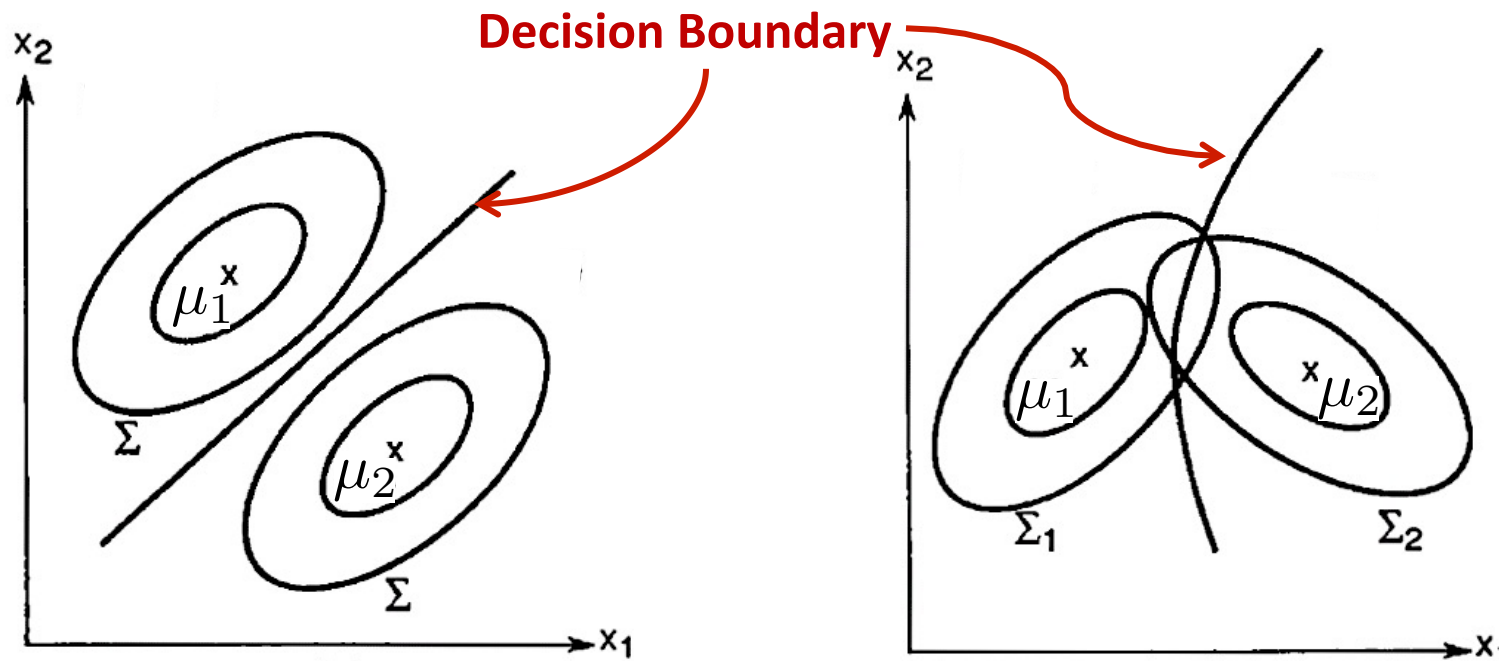
Is the decision boundary still linear?

Comparison with Gaussian Naïve Bayes

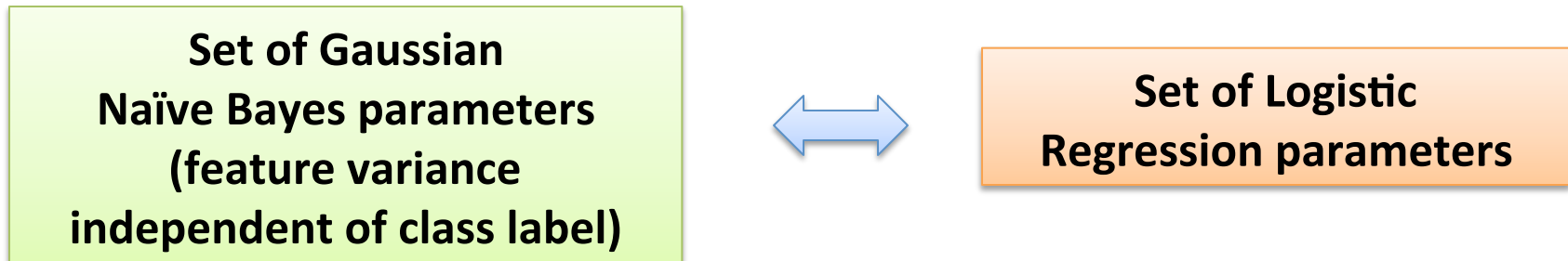
Decision Boundary of Bayes & Naïve Bayes Classifiers

- Binary classification with continuous features
decision boundary is set of points x : $P(Y=1|X=x) = P(Y=0|X=x)$

If class conditional feature distribution $P(X=x|Y=y)$ is 2-dim Gaussian $N(\mu_y, \Sigma_y)$



Gaussian Naïve Bayes vs. Logistic Regression



- Representation equivalence (both yield linear decision boundaries)
 - **But only in a special case!!!** (GNB with class-independent variances)
 - **LR makes no assumptions about $P(X|Y)$ in learning!!!**
 - **Optimize different functions (MLE/MCLE) or (MAP/MCAP)! Obtain different solutions**

Gaussian Naïve Bayes vs. Logistic Regression

[Ng & Jordan, NIPS 2001]

Given **infinite data** (asymptotically),

If conditional independence assumption holds,
Discriminative LR and generative NB perform similar.

$$\epsilon_{\text{Dis},\infty} \sim \epsilon_{\text{Gen},\infty}$$

If conditional independence assumption does NOT hold,
Discriminative LR outperforms generative NB.

$$\epsilon_{\text{Dis},\infty} < \epsilon_{\text{Gen},\infty}$$

Gaussian Naïve Bayes vs. Logistic Regression

Consider Y boolean, X_i continuous, $X = \langle X_1 \dots X_d \rangle$

Number of parameters:

- NB: $4d + 1$ $\theta, (\mu_{1,y}, \mu_{2,y}, \dots, \mu_{d,y}), (\sigma^2_{1,y}, \sigma^2_{2,y}, \dots, \sigma^2_{d,y})$ $y = 0, 1$
 $3d + 1$ if class independent variances
- LR: $d + 1$ w_0, w_1, \dots, w_d

Estimation method:

- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

Gaussian Naïve Bayes vs. Logistic Regression

Given **finite data** (n data points, d features),

[Ng & Jordan, NIPS 2001]

$$\epsilon_{\text{Dis},n} \leq \epsilon_{\text{Dis},\infty} + O\left(\sqrt{\frac{d}{n}}\right)$$

$$\epsilon_{\text{Gen},n} \leq \epsilon_{\text{Gen},\infty} + O\left(\sqrt{\frac{\log d}{n}}\right)$$

Naïve Bayes (generative) requires $n \sim \log d$ to converge to its asymptotic error, whereas Logistic regression (discriminative) requires $n \sim d$.

Why? “Independent class conditional densities”
* parameter estimates not coupled – each parameter is learnt independently, not jointly, from training data.

Naïve Bayes vs Logistic Regression

Verdict

Both learn a linear boundary (assuming class-ind
feature variance)

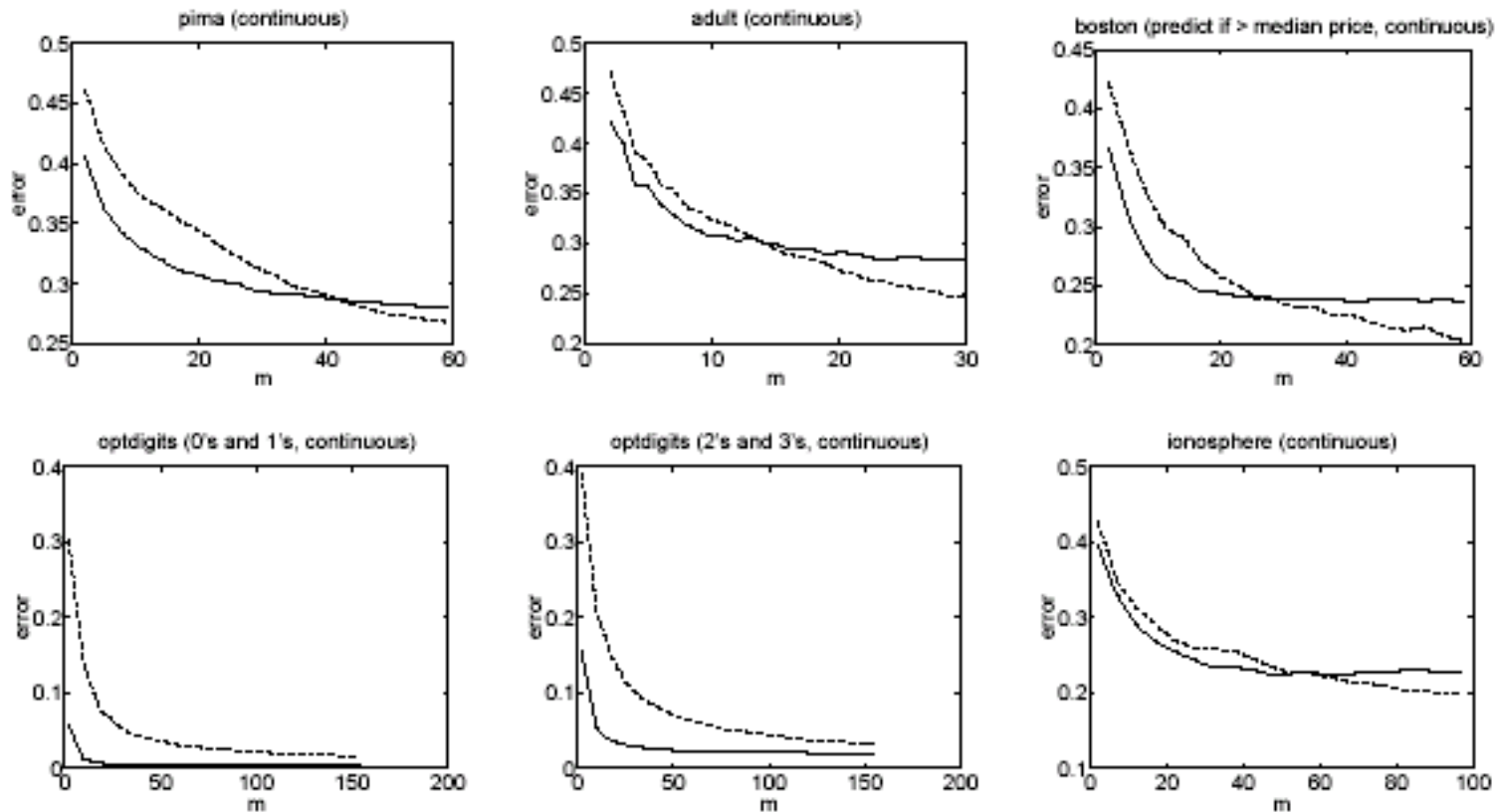
Naïve Bayes makes more restrictive assumptions
and has higher asymptotic error,

BUT

converges faster to its less accurate asymptotic
error.

Experimental Comparison (Ng-Jordan'01)

UCI Machine Learning Repository 15 datasets, 8 continuous features, 7 discrete features



More in
Paper...

— Naïve Bayes

----- Logistic Regression

What you should know

- LR is a linear classifier
- LR optimized by maximizing conditional likelihood or conditional posterior
 - no closed-form solution
 - concave ! global optimum with gradient ascent
- 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 ! assumption on $P(\mathbf{X}|Y)$
 - LR: Functional form of $P(Y|\mathbf{X})$, no assumption on $P(\mathbf{X}|Y)$
- Convergence rates
 - GNB (usually) needs less data
 - LR (usually) gets to better solutions in the limit