10-701 Introduction to Machine Learning (PhD)
Lecture 6: Logistic Regression

Leila Wehbe
Carnegie Mellon University
Machine Learning Department

Slides based on Tom Mitchell's
10-701 Spring 2016 material

Last time: Naïve Bayes in a Nutshell

Bayes rule:

$$P(Y = y_k | X_1 \ldots X_n) = \frac{P(Y = y_k) P(X_1 \ldots X_n | Y = y_k)}{\sum_j P(Y = y_j) P(X_1 \ldots X_n | Y = y_j)}$$

Assuming conditional independence among X_i's:

$$P(Y = y_k | X_1 \ldots X_n) = \frac{P(Y = y_k) \prod_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i | Y = y_j)}$$

(estimate in training)

So, to pick most probable Y for $X^{new} = (X_1, \ldots, X_n)$

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

(testing)

Naïve Bayes Algorithm – discrete X_i

- Train Naïve Bayes (examples)

  for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \; \pi_k \prod_i \theta_{ijk}$$

* probabilities must sum to 1, so need estimate only v-1 of these, where v is the number of values, which is 2 in the binary case

MAP estimates for bag of words

Map estimate for multinomial

\# seen "aardvark"
\# hallucinated "aardvark"

$$\theta_{jk} = \frac{\boxed{\alpha_{jk}} + \boxed{\beta_{jk}} - 1}{\boxed{\sum_m \alpha_{mk}} + \boxed{\sum_m \beta_{mk}} - 1}$$

\# seen words
\# hallucinated words

What β's should we choose?

Missing data?

What if we have continuous X_i ?

Gaussian Naïve Bayes (GNB): assume

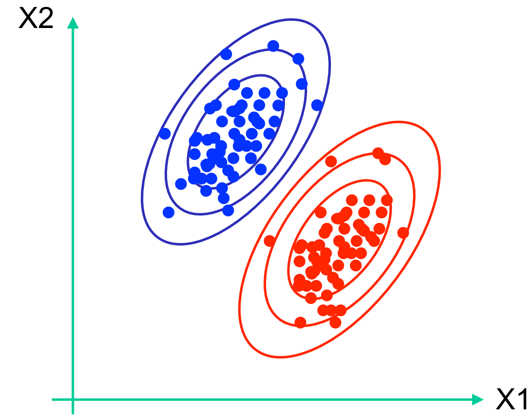
$$p(X_i = x | Y = y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} \; e^{-\frac{1}{2}\left(\frac{x-\mu_{ik}}{\sigma_{ik}}\right)^2}$$

Sometimes assume variance

- is independent of Y (i.e., σ_i),
- or independent of X_i (i.e., σ_k)
- or both (i.e., σ)

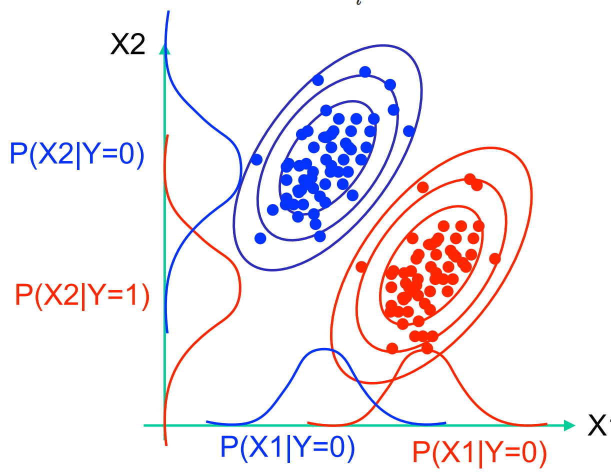
Gaussian Naïve Bayes – Big Picture

$$Y^{new} \leftarrow \arg\max_{y \in \{0,1\}} P(Y=y) \prod_i P(X_i^{new} | Y=y)$$ assume P(Y=1) = 0.5



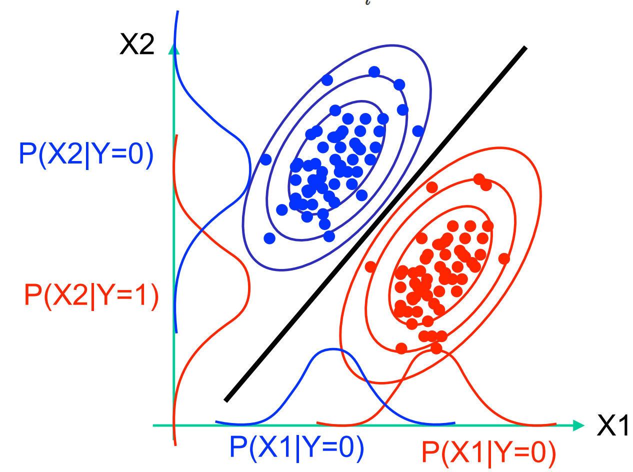
Gaussian Naïve Bayes – Big Picture

$$Y^{new} \leftarrow \arg\max_{y \in \{0,1\}} P(Y=y) \prod_i P(X_i^{new} | Y=y)$$ assume P(Y=1) = 0.5



Gaussian Naïve Bayes – Big Picture

$$Y^{new} \leftarrow \arg\max_{y \in \{0,1\}} P(Y=y) \prod_i P(X_i^{new} | Y=y)$$ assume P(Y=1) = 0.5



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, ($X_1 \ldots X_n$)
  - 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 (π)

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

$$P\big(Y=1|X=(X_1,...,X_n)\big) = \frac{1}{1+exp(w_0+\sum_i w_i X_w)}$$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

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

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

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

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

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

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

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

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

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

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

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

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

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \boxed{\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}}$$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

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

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

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

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \boxed{\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}}$$

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

$$\ln P(X_i = x_i|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} + \frac{-(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}$$

$$\ln \frac{P(X_i = x_i|Y = 0)}{P(X_i = x_i|Y = 1)} = \frac{1}{\sigma_{i0}\sqrt{2\pi}} + \frac{-(x_i - \mu_{i0})^2}{2\sigma_{i0}^2} - \frac{1}{\sigma_{i1}\sqrt{2\pi}} + \frac{-(x_i - \mu_{i1})^2}{2\sigma_{i1}^2}$$

$$= \frac{1}{\sigma_{i0}\sqrt{2\pi}} - \frac{1}{\sigma_{i1}\sqrt{2\pi}} - \frac{x_i^2 - 2x_i\mu_{i0} + \mu_{i0}^2}{2\sigma_{i0}^2} + \frac{x_i^2 - 2x_i\mu_{i1} + \mu_{i1}^2}{2\sigma_{i1}^2}$$

Now assume $\sigma_{ik} = \sigma_i$

Derive form for P(Y|X) for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

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

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

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

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \boxed{\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}}$$

$$\boxed{\sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2)}{2\sigma_i^2} \right)}$$

Linear function!

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

Very convenient!

$$P(Y=1|X=(X_1,...,X_n)) = \frac{1}{1+exp(w_0+\sum_i w_i X_w)}$$

$$P(Y=0|X=(X_1,...,X_n)) =$$

implies

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

implies

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

Very convenient!

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

$$P(Y=0|X=(X_1,...,X_n)) = \frac{exp(w_0+\sum_i w_i X_w)}{1+exp(w_0+\sum_i w_i X_w)}$$

implies

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

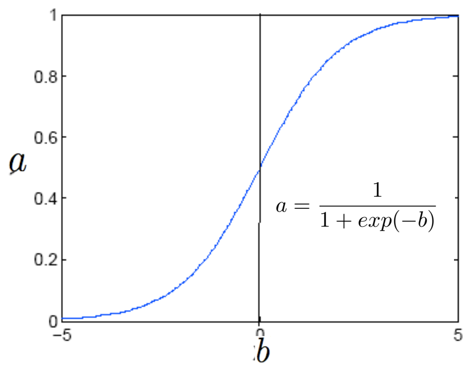
linear classification rule!

implies

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

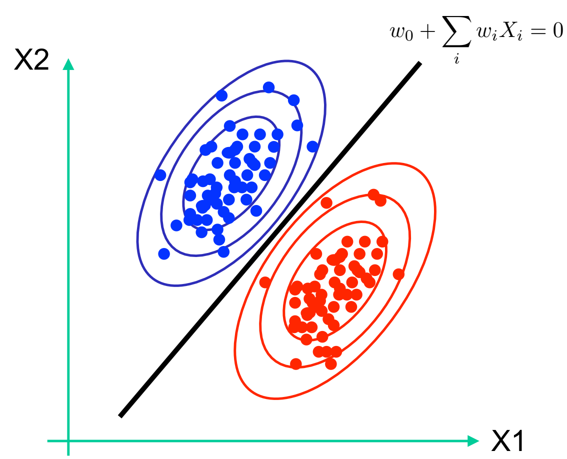
$<$ or $>$ 0

Logistic function



$$a = \frac{exp(b)}{1+exp(b)}$$

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



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: MLE, MCLE

- we have L training examples: $\{(X^1,Y^1),...,(X^L,Y^L)\}$
- maximum likelihood estimate for parameters W

$$W_{MLE}=\arg\max_W P((X^1,Y^1),...,(X^L,Y^L)|W)$$
$$=\arg\max_W \prod_l P(X^l,Y^l|W)$$

- maximum conditional likelihood estimate

$$W_{M(C)LE}=\arg\max_W \prod_l P(Y^l|X^l,W)$$

Training Logistic Regression: MCLE

- Choose parameters W=<w_0, ... w_n> to maximize conditional likelihood of training data

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

- Training data D = $\{(X^1,Y^1),...,(X^L,Y^L)\}$
- Data likelihood = $\prod_l P(X^l,Y^l|W)$
- Data conditional likelihood = $\prod_l P(Y^l|X^l,W)$

$$W_{MCLE}=\arg\max_W \prod_l P(Y^l|W,X^l)$$

Expressing Conditional Log Likelihood

$$l(W)\equiv \ln\prod_l P(Y^l|X^l,W)=\sum_l \ln P(Y^l|X^l,W)$$

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

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

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

Expressing Conditional Log Likelihood

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

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

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

$$\begin{aligned} l(W) &= \sum_l Y^l \ln P(Y^l = 1|X^l, W) + (1 - Y^l) \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l \ln \frac{P(Y^l = 1|X^l, W)}{P(Y^l = 0|X^l, W)} + \ln P(Y^l = 0|X^l, W) \end{aligned}$$

Expressing Conditional Log Likelihood

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

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

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

$$\begin{aligned} l(W) &= \sum_l Y^l \ln P(Y^l = 1|X^l, W) + (1 - Y^l) \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l \ln \frac{P(Y^l = 1|X^l, W)}{P(Y^l = 0|X^l, W)} + \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) \end{aligned}$$

Maximizing Conditional Log Likelihood

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

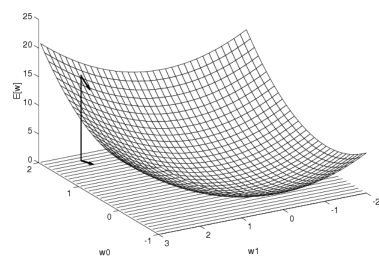
$$P(Y=1|X,W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$\begin{aligned} l(W) &\equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W) \\ &= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) \end{aligned}$$

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

Bad news: no closed-form solution to maximize $l(W)$

Gradient Descent



Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

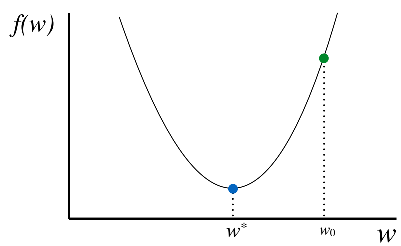
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Update the vector of parameters

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E_d(\mathbf{w})$$

Gradient Descent

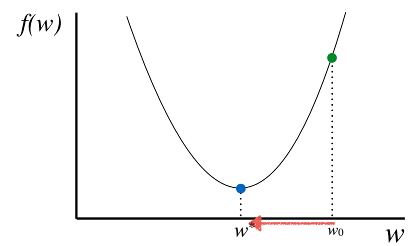
Start at a random point



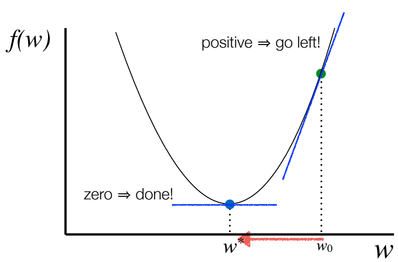
Gradient Descent

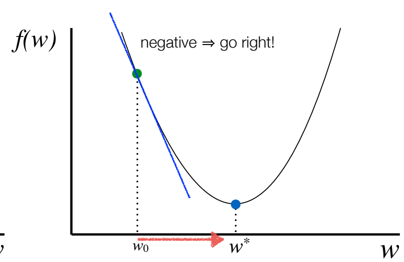
Start at a random point

Determine a descent direction



Choosing Descent Direction (1D)





We can only move in two directions

Negative slope is direction of descent!

Step Size

$$w_{t+1} \leftarrow w_t - \eta \nabla E_d(w_t)$$

Update rule

Negative Slope

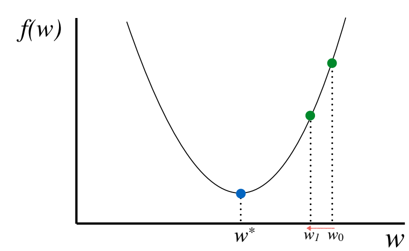
Gradient Descent

Start at a random point

Determine a descent direction

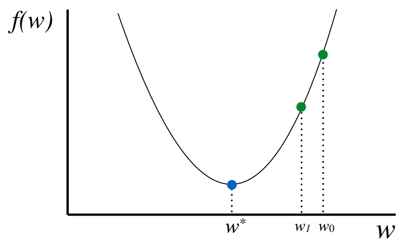
Choose a step size

Update



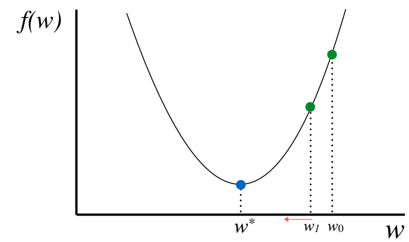
Gradient Descent

Start at a random point
Repeat
    Determine a descent direction
    Choose a step size
    Update
Until stopping criterion is satisfied



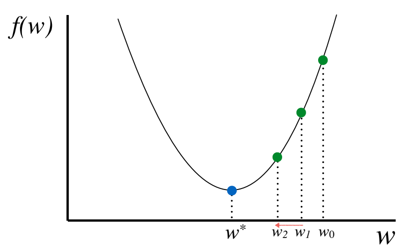
Gradient Descent

Start at a random point
Repeat
    Determine a descent direction
    Choose a step size
    Update
Until stopping criterion is satisfied



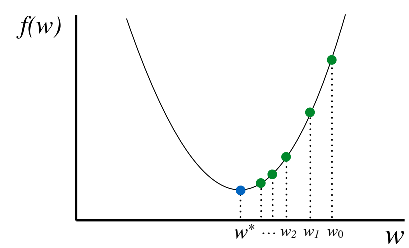
Gradient Descent

Start at a random point
Repeat
    Determine a descent direction
    Choose a step size
    Update
Until stopping criterion is satisfied

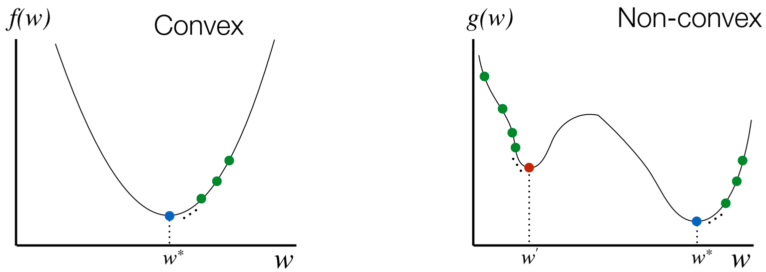


Gradient Descent

Start at a random point
Repeat
    Determine a descent direction
    Choose a step size
    Update
Until stopping criterion is satisfied



Where Will We Converge?



Any local minimum is a global minimum

Multiple local minima may exist

Gradient Descent:

Batch gradient: use error $E_D(\mathbf{w})$ over entire training set D

Do until satisfied:

1. Compute the gradient $\nabla E_D(\mathbf{w}) = \left[\frac{\partial E_D(\mathbf{w})}{\partial w_0} \dots \frac{\partial E_D(\mathbf{w})}{\partial w_n}\right]$
2. Update the vector of parameters: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E_D(\mathbf{w})$

Stochastic gradient: use error $E_d(\mathbf{w})$ over single examples $d \in D$

Do until satisfied:

1. Choose (with replacement) a random training example $d \in D$
2. Compute the gradient just for d: $\nabla E_d(\mathbf{w}) = \left[\frac{\partial E_d(\mathbf{w})}{\partial w_0} \dots \frac{\partial E_d(\mathbf{w})}{\partial w_n}\right]$
3. Update the vector of parameters: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E_d(\mathbf{w})$

Stochastic approximates Batch arbitrarily closely as $\eta \to 0$

Stochastic can be much faster when D is very large

Intermediate approach: use error over subsets of D

Maximize Conditional Log Likelihood: Gradient Ascent

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

$$= \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l))$$

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

Intuitive notion

Maximize Conditional Log Likelihood: Gradient Ascent

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

$$= \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l))$$

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

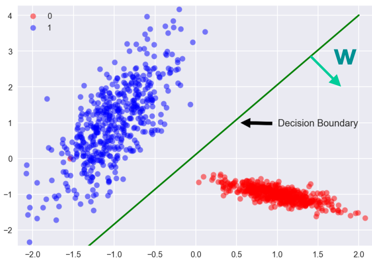
Gradient ascent algorithm: iterate until change < ε

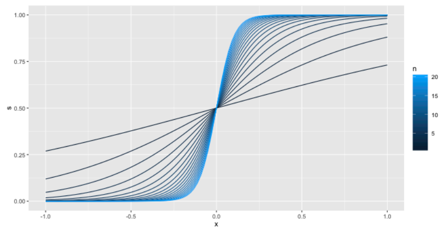
For all i, repeat

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

Need to regularize the weights

- $w \to \infty$ to maximize the probability of the data, if data linearly separable





That's all for M(C)LE. How about MAP?

- For MAP, need to define prior on W
  - given W = (w_1, … w_n)
    let's assume prior P(w_i) = N(0, σ)
  - i.e., assume zero mean, Gaussian prior for each w_i
- A kind of Occam's razor (simplest is best) prior
- Helps avoid very large weights and overfitting

MAP Estimates for Logistic Regression

$$W^{MAP} = \arg\max_W P(W|Y,X) = \frac{P(Y|W,X)P(W,X)}{P(Y,X)}$$

MAP Estimates for Logistic Regression

$$W^{MAP} = \arg\max_W P(W|Y,X) = \frac{P(Y|W,X)P(W,X)}{P(Y,X)}$$

$$= \arg\max_W P(Y|W,X)P(W,X)$$

$$= \arg\max_W P(Y|W,X)P(W)P(X)$$

let's assume P(W,X)=P(W) P(X)

$$= \arg\max_W P(Y|W,X)P(W)$$

$$W^{MAP} = \arg\max_W [\ln P(Y|W,X) + \ln P(W)]$$

zero mean Gaussian P(W)

$$P(W) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\sum_i w_i^2\right)$$

$$W^{MAP} = \arg\max_W [\ln P(Y|W,X) - \left(\frac{1}{2\sigma^2}\sum_i w_i^2\right)]$$

MLE vs MAP

- Maximum conditional likelihood estimate

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

$$\boxed{w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

- Maximum a posteriori estimate with prior

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

$$\boxed{w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

MAP estimates and Regularization

- Maximum a posteriori estimate with prior

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

$$\boxed{w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

called a "regularization" term
- helps reduce overfitting
- if P(W) is Gaussian, then encourages W to be near the mean of P(W) : zero here, but can easily use any mean
- used very frequently in Logistic Regression

The Bottom Line

- Consider learning f: X → 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 (π)
- Then P(Y|X) is of this form, and we can directly estimate W

$$P(Y = 1|X = (X_1, ..., X_n)) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_w)}$$

- Furthermore, same holds if the X_i are boolean
  - trying proving that to yourself

Generative vs. Discriminative Classifiers

Training classifiers involves estimating f: X → Y, 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

Naïve Bayes vs Logistic Regression

Consider Y boolean, X_i continuous, $X=(X_1 \ldots X_n)$

Number of parameters to estimate:

- NB:

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

- 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

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, $\leftarrow$ not $N(\mu_{ik},\sigma_{ik})$

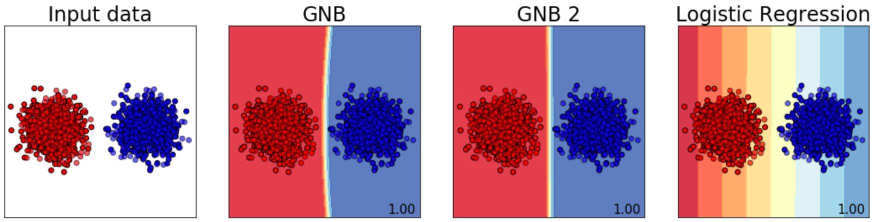
[Ng & Jordan, 2002]

Consider three learning methods:
•GNB (assumption 1 only) -- decision surface can be non-linear
•GNB2 (assumption 1 and 2) – decision surface linear
•LR -- decision surface linear, trained without assumption 1.

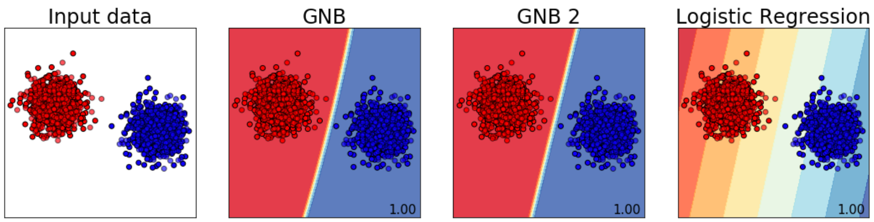
How do these methods perform if we have plenty of data and:

•Both (1) and (2) are satisfied:

Assumptions 1 and 2 are satisfied



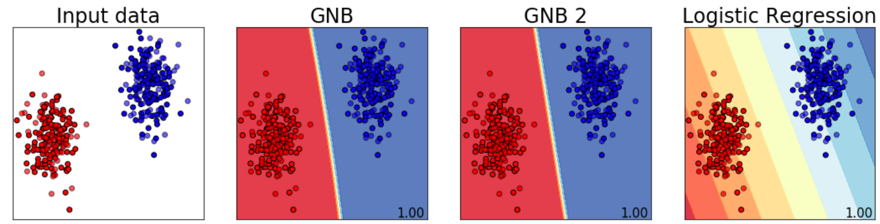
X_i's conditionally independent and variance is shared



In these cases, LR, GNB2 and GNB perform similarly

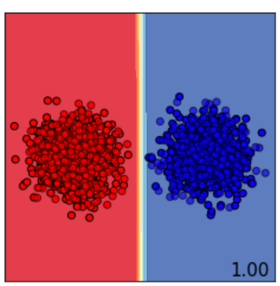
Assumptions 1 and 2 satisfied

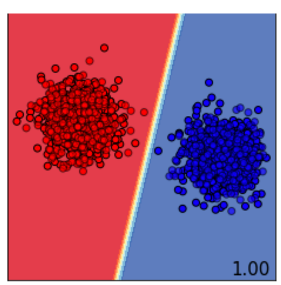
The decision boundary of GNB and GNB2 is sensitive to the locations of the means (since the variances are the same)



Assumptions 1 and 2 satisfied

If the variances of the Xi are the same (across classes and across i), the decision boundary of GNB2 and GNB is determined by the distance to the mean (perpendicular bisector)
If one of the coordinates of the two means are the same, then the decision boundary becomes perpendicular to that axis





G.Naïve Bayes vs. Logistic Regression

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, $\leftarrow$ not $N(\mu_{ik},\sigma_{ik})$

[Ng & Jordan, 2002]

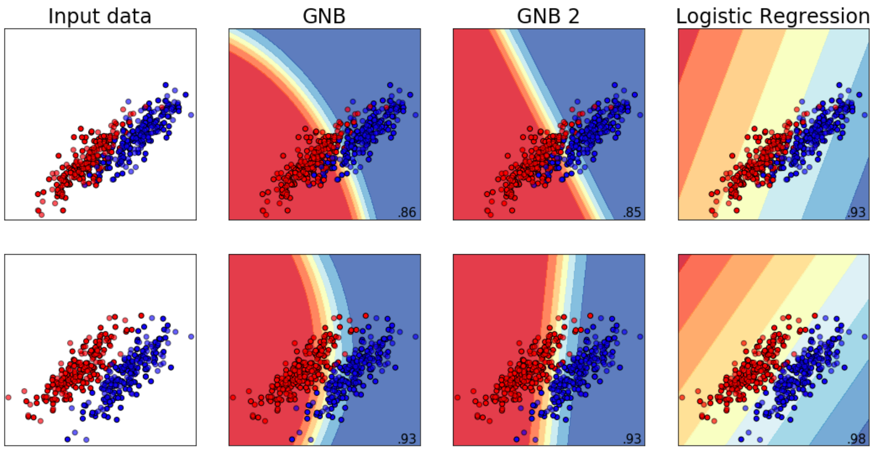
Consider three learning methods:
•GNB (assumption 1 only) -- decision surface can be non-linear
•GNB2 (assumption 1 and 2) – decision surface linear
•LR -- decision surface linear, trained without assumption 1.

How do these methods perform if we have plenty of data and:

•Both (1) and (2) are satisfied

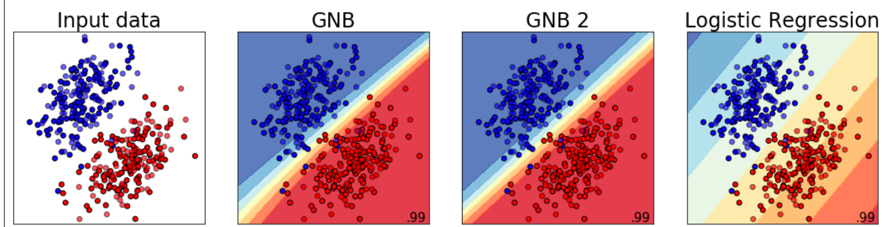
•(2) is satisfied, but not (1)

Assumption 2 satisfied and not 1



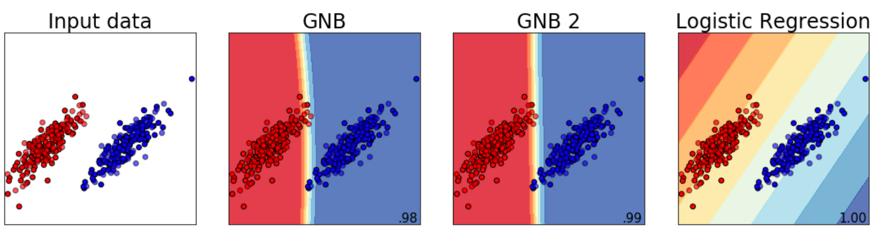
GNB2 can break (also GNB)

Assumption 2 satisfied and not 1



GNB2 and GNB can also work well

Assumption 2 satisfied and not 1



The decision boundary of GNB2 and GNB is also dependent on the means of the two classes. If one of the coordinates of the two means is the same, again, we have a decision boundary parallel to that axis

G.Naïve Bayes vs. Logistic Regression

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, ← not $N(\mu_{ik},\sigma_{ik})$

[Ng & Jordan, 2002]

Consider three learning methods:
- •GNB (assumption 1 only) -- decision surface can be non-linear
- •GNB2 (assumption 1 and 2) – decision surface linear
- •LR -- decision surface linear, trained without assumption 1.

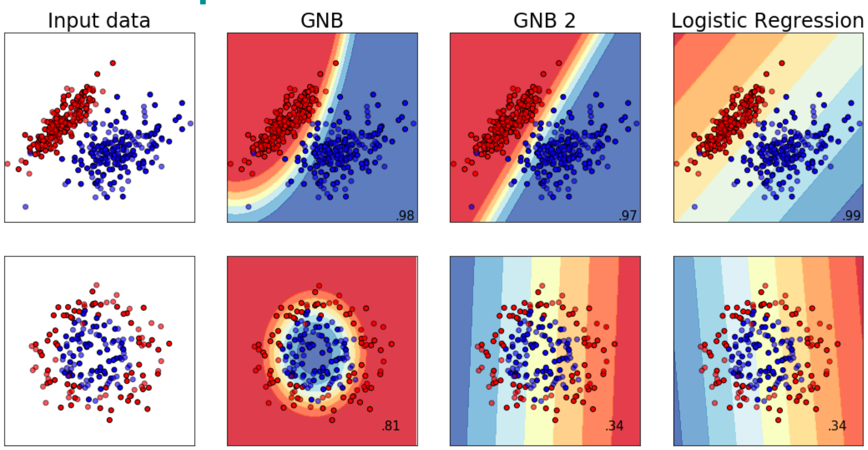
How do these methods perform if we have plenty of data and:

•Both (1) and (2) are satisfied

•(2) is satisfied, but not (1)

•Neither (1) nor (2) is satisfied

Assumptions 1 and 2 are not satisfied



Depending on the dataset, GNB and LR have different performances.
Even though LR and GNB2 can be expressed in the same way, LR has more flexibly to learn parameters that fit the data, and they are don't have to be tied to the marginal means and variance

G.Naïve Bayes vs. Logistic Regression

What if we have only finite training data?

They converge at different rates to their asymptotic (∞ data) error

Let $\epsilon_{A,m}$ refer to expected error of learning algorithm A after *m* training examples

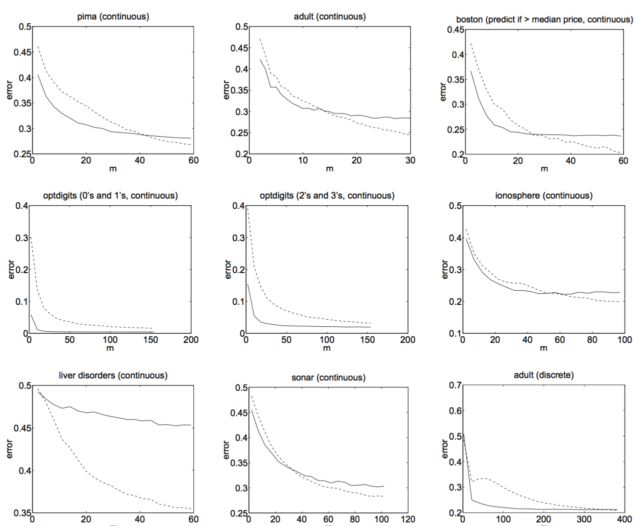
Let *n* be the number of features: $(X_1 \ldots X_n)$ [Ng & Jordan, 2002]

$$\epsilon_{LR,m} \leq \epsilon_{LR,\infty} + O\left(\sqrt{\frac{n}{m}}\right)$$

$$\epsilon_{GNB,m} \leq \epsilon_{GNB,\infty} + O\left(\sqrt{\frac{log(n)}{m}}\right)$$

So, GNB requires *m = O(log n)* to converge, but LR requires *m = O(n)*

Some experiments from UCI data sets



[Ng & Jordan, 2002]

Some experiments from UCI data sets

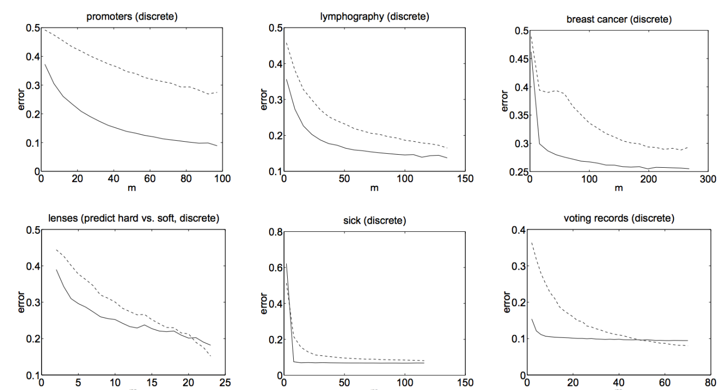


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. m (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

[Ng & Jordan, 2002]

Naïve Bayes vs. Logistic Regression

The bottom line:

GNB2 and LR both use linear decision surfaces, GNB need not

Given infinite data, LR is better or equal to GNB2 because *training procedure* does not make assumptions 1 or 2 (though our derivation of the form of P(Y|X) did).

But GNB2 converges more quickly to its perhaps-less-accurate asymptotic error. (more bias than LR)

And GNB is both more biased (assumption1) and less (no assumption 2) than LR, so either might outperform the other.

What you should know:

- Logistic regression
  - Functional form follows from Naïve Bayes assumptions
    - For Gaussian Naïve Bayes assuming variance $\sigma_{i,k} = \sigma_i$
    - For discrete-valued Naïve Bayes too
  - But training procedure picks parameters without making conditional independence assumption
  - MLE training: pick W to maximize P(Y | X, W)
  - MAP training: pick W to maximize P(W | X,Y)
    - 'regularization'
    - helps reduce overfitting

- Gradient ascent/descent
  - General approach when closed-form solutions unavailable

- Generative vs. Discriminative classifiers
  - Bias vs. variance tradeoff

Questions to think about:

- Can you use Naïve Bayes for a combination of discrete and real-valued X_i?

- How can we easily model the assumption that just 2 of the n attributes as dependent?

- What does the decision surface of a Naïve Bayes classifier look like?

- How would you select a subset of X_i's?