MLE vs. MAP

$D = \{x_1, \ldots, x_n\}$

- Maximum Likelihood estimation (MLE)

  Choose value that maximizes the probability of observed data

  $p(x) \sim Ber(\theta)$
  $p(x) \sim N(\mu, \Sigma) \equiv \theta$

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D|\theta)$$

likelihood

- Maximum *a posteriori* (MAP) estimation

  Choose value that is most probable given observed data and prior belief

$p(\theta)$ - prior
$\downarrow$
$p(\theta|D)$ - posterior

$$\hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta|D)$$
$$= \arg\max_{\theta} P(D|\theta)P(\theta)$$

likelihood prior

When is MAP same as MLE?

MLE vs. MAP for Bernoulli r.v.

- Maximum Likelihood estimation (MLE)

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D|\theta)$$

$$\hat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

$X \sim Ber(\theta)$ 0 or 1 (H or T)

$D = \{X_1 \ldots X_n\}$

α_H → no. of heads

$\alpha_H + \alpha_T$ → total no. of tosses

- Maximum *a posteriori* (MAP) estimation

$$\hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta|D)$$

$$= \arg\max_{\theta} P(D|\theta)P(\theta)$$

$$\hat{\theta}_{MAP} = \frac{\alpha_H + \beta_H - 1}{\alpha_H + \beta_H + \alpha_T + \beta_T - 2}$$

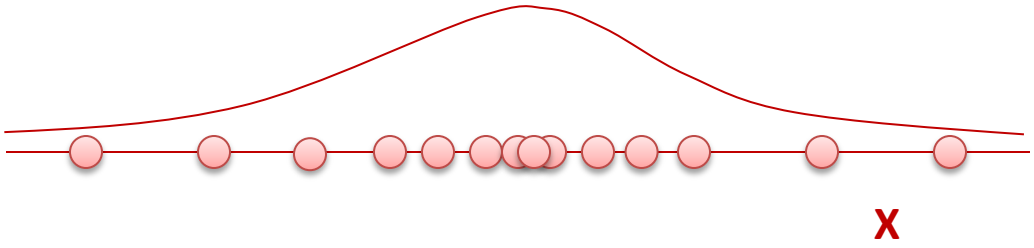
Conjugate prior

$P(\theta) \sim Beta(\beta_H, \beta_T)$

$P(\theta|D) \sim Beta(\beta_H + \alpha_H, \beta_T + \alpha_T)$

Gaussian distribution

Data, D =



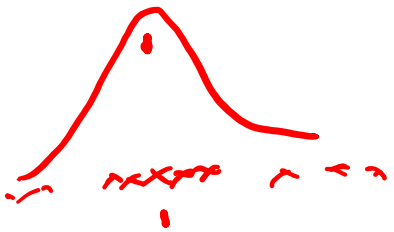
- Parameters: μ – mean, σ^2 - variance

- Data are **i.i.d.**:
  - **Independent** events
  - **Identically distributed** according to Gaussian distribution

MLE for Gaussian mean and variance

$\sigma^2 = E[(X-\mu)^2]$

Parameters $\theta = (\mu, \sigma^2)$



$$\hat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu}_{MLE})^2$$

MAP estimation for Gaussian r.v.

Parameters $\theta = (\mu, \sigma^2)$

$x \sim N(\mu, \sigma^2)$

- Mean μ (known σ^2): Gaussian prior $P(\mu) = N(\eta, \lambda^2)$

$$P(\mu \mid \eta, \lambda) = \frac{1}{\lambda\sqrt{2\pi}} e^{\frac{-(\mu-\eta)^2}{2\lambda^2}}$$

$$\hat{\mu}_{MAP} = \frac{\frac{1}{\sigma^2}\sum_{i=1}^{n} x_i + \frac{\eta}{\lambda^2}}{\frac{n}{\sigma^2} + \frac{1}{\lambda^2}} \qquad \hat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

As we get more data, effect of prior is "washed out"

$p(\theta|D) \propto p(D|\theta)\, p(\theta)$

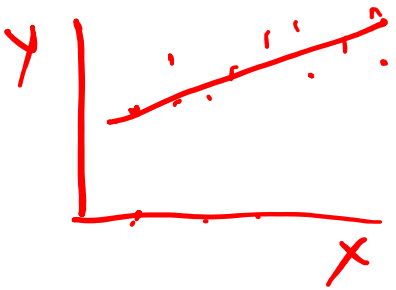
- Variance σ^2 (known μ): inv-Wishart Distribution

Logistic Regression

Aarti Singh

Machine Learning 10-701
Jan 30, 2023

Discriminative Classifiers



Bayes Classifier:

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

$\equiv P(X)$ $P(X|Y=y)$

Why not learn P(Y|X) directly? Or better yet, why not learn the decision boundary directly?

- Assume some functional form for P(Y|X) or for the decision boundary
- Estimate parameters of functional form directly from training data

Today we will see one such classifier – **Logistic Regression**

Logistic Regression

Not really regression

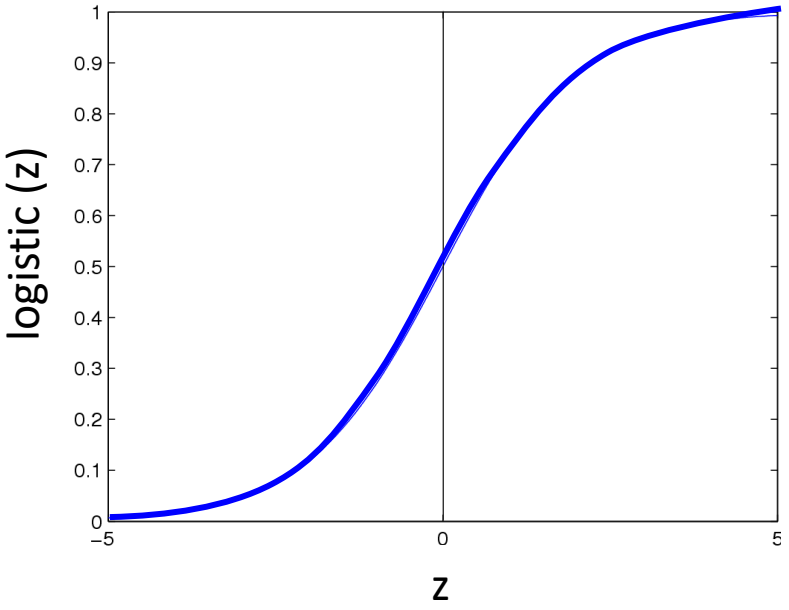
Assumes the following functional form for P(Y|X):

$z = \sum w_i X_i + w_0$

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

Logistic function applied to a linear function of the data

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



Features can be discrete or continuous!

Logistic Regression is a Linear Classifier!

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

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

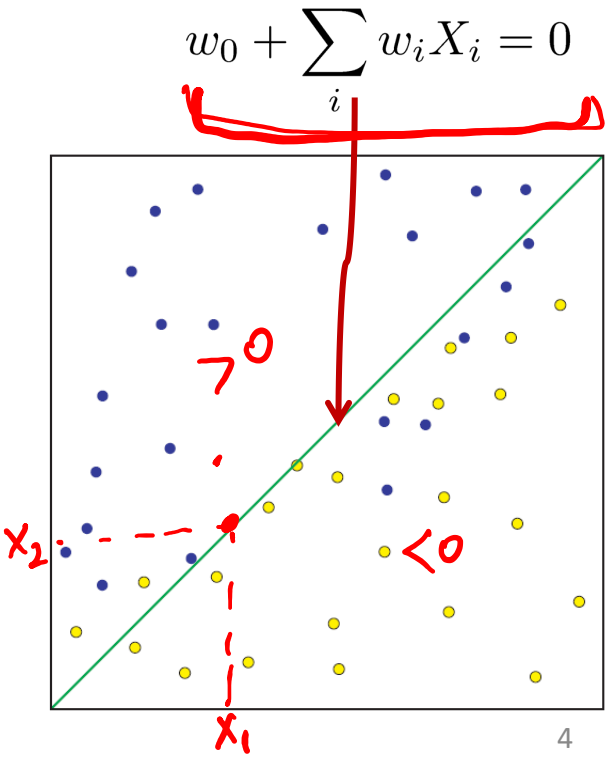
$\theta = \{w_0, w_1, \ldots w_d\}$
d - features

Decision boundary:

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

$$\Rightarrow w_0 + \sum_i w_i X_i \underset{0}{\overset{1}{\gtrless}} 0$$

(Linear Decision Boundary)



Training Logistic Regression

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

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

Maximum Likelihood Estimates

$$\widehat{\mathbf{w}}_{MLE} = \arg\max_{\mathbf{w}} \prod_{j=1}^{n} P(X^{(j)}, Y^{(j)} \mid \mathbf{w})$$

$P(D|\theta)$

$P(Y|X, w)\, P(X, w)$ ✗

But there is a problem …

Don't have a model for P(X) or P(X|Y) – only for P(Y|X)

Training Logistic Regression

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

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

Maximum (Conditional) Likelihood Estimates

$$\hat{\mathbf{w}}_{MCLE} = \arg\max_{\mathbf{w}} \prod_{j=1}^{n} P(Y^{(j)} \mid X^{(j)}, \mathbf{w})$$

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

Expressing Conditional log Likelihood

$P(Y=y|X) = \frac{e^{yz}}{1+e^{z}}$

$$P(Y=1|X,w) = \frac{1}{1+\exp(-w_0 - \sum_i w_i X_i)} = \frac{1}{1+e^{-z}} = \frac{e^{z}}{1+e^{z}}$$

$$P(Y=0|X,w) = \frac{1}{1+\exp(w_0 + \sum_i w_i X_i)} = 1 - \frac{1}{1+e^{-z}} = \frac{e^{-z}}{1+e^{-z}} = \frac{1}{1+e^{z}}$$

log likelihood

$$l(\mathbf{w}) \equiv \ln \prod_j P(y^j|\mathbf{x}^j, \mathbf{w})$$

$$= \log \prod_{j=1}^{n} \frac{e^{y^j z^j}}{1+e^{z^j}} = \sum_{i=1}^{n} \log \frac{e^{y^j z^j}}{1+e^{z^j}}$$

$$= \sum_{i=1}^{n} y^j z^j - \log(1+\exp(z^j))$$

$z^j = \sum_i w_i x_i^j$

Expressing Conditional log Likelihood

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

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

$$l(\mathbf{w}) \equiv \ln \prod_j P(y^j|\mathbf{x}^j,\mathbf{w})$$

$$= \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

$\frac{\partial}{\partial w_0} = \sum_j y^j - \frac{(\cdots)}{1+\exp(\cdots)}\Big|_{w_{0\,MLE}} = 0$ MCLE

Good news: $l(\mathbf{w})$ is concave function of $\mathbf{w}$!

Bad news: no closed-form solution to maximize $l(\mathbf{w})$ HW2!

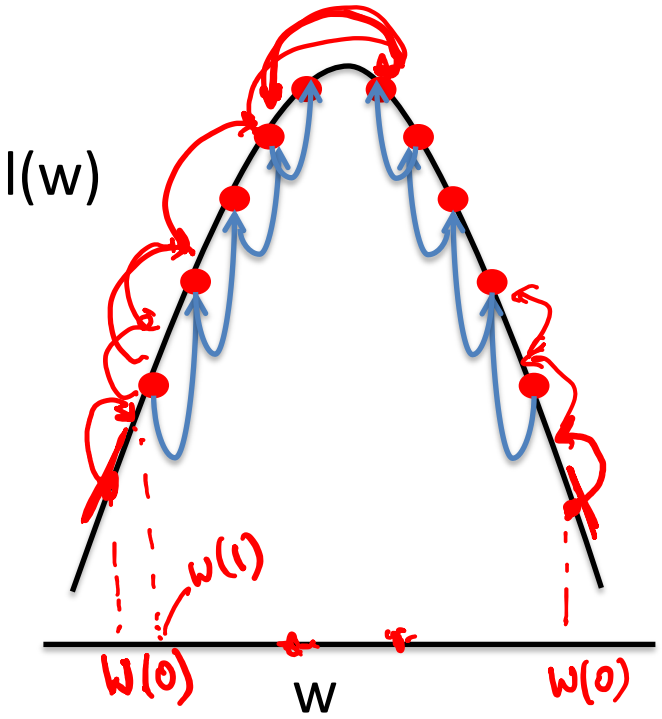
Good news: can use iterative optimization methods (gradient ascent)

Iteratively optimizing concave function

$\mathbf{w} = (w_0, w_1, \ldots w_d)$

- Conditional likelihood for Logistic Regression is concave
- Maximum of a concave function can be reached by

Gradient Ascent Algorithm



Initialize: Pick **w** at random

Gradient:

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

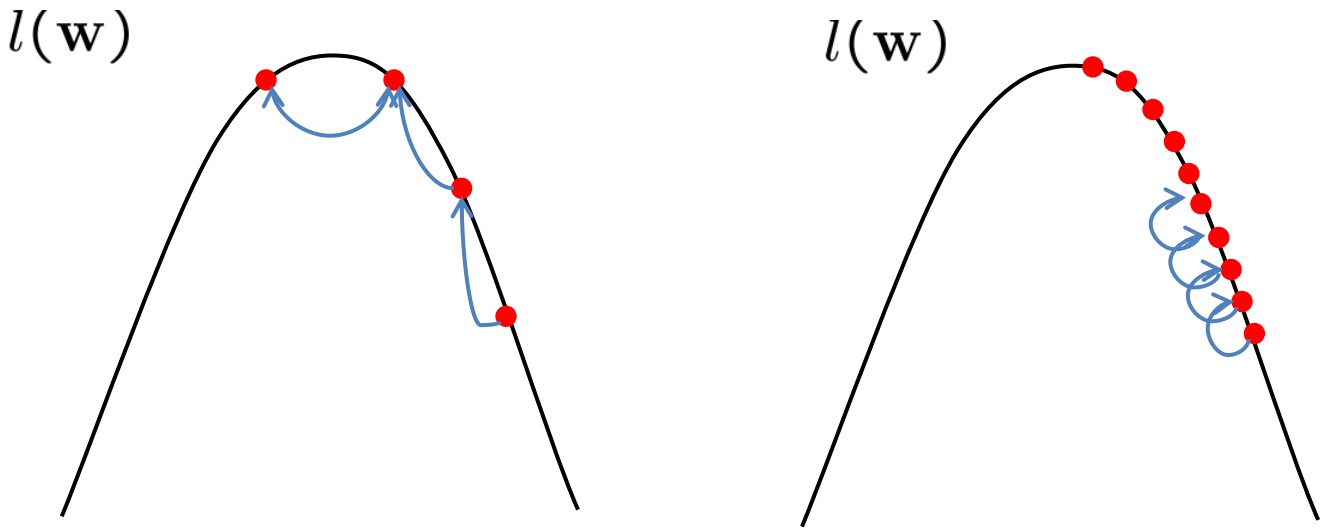
Update rule: Learning rate, $\eta > 0$

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

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

➢ Poll: Effect of step-size η?

Effect of step-size η



Large η => Fast convergence but larger residual error
Also possible oscillations

Small η => Slow convergence but small residual error

That's M(C)LE. How about M(C)AP?

$$p(\mathbf{w} \mid Y, \mathbf{X}) \propto P(Y \mid \mathbf{X}, \mathbf{w}) p(\mathbf{w})$$

conditional likelihood prior

- Define priors on **w**
  - Common assumption: Normal distribution, zero mean, identity covariance
  - "Pushes" parameters towards zero

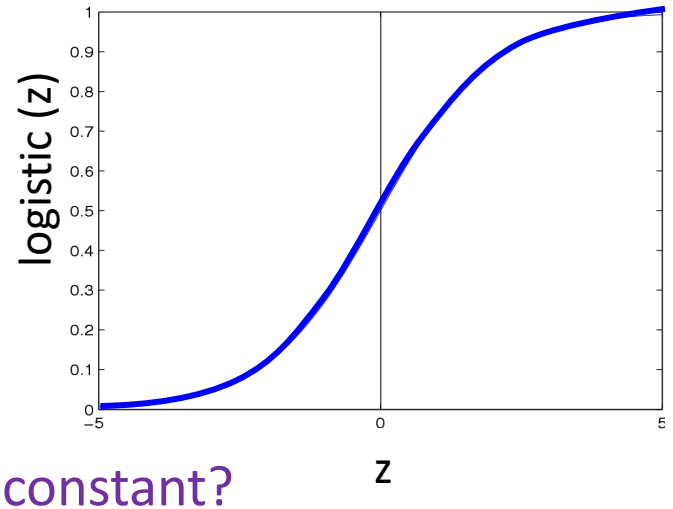
$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} e^{\frac{-w_i^2}{2\kappa^2}}$$

Zero-mean Gaussian prior

Logistic function (or Sigmoid):

$$\frac{1}{1 + exp(-z)}$$

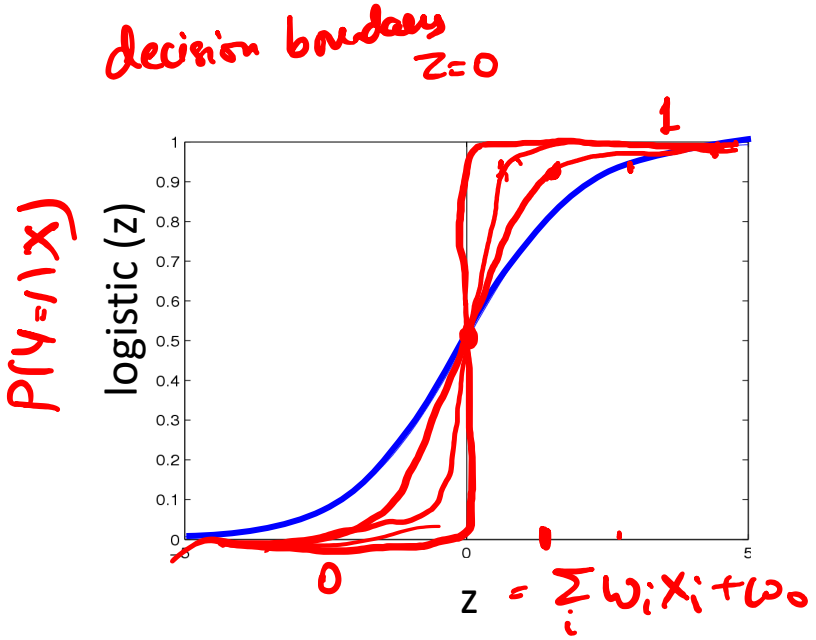
$z = \sum_i w_i x_i + w_0$



➢ What happens if we scale z by a large constant?

Logistic function (or Sigmoid):

$$\frac{1}{1+exp(-z)}$$



- Poll: What happens if we scale z (equivalently weight w) by a large constant?

A) The logistic classifier decision boundary shifts towards class 1
B) The logistic classifier decision boundary remains same ✓
C) The logistic classifier tries to separate the data perfectly ✓
D) The logistic classifier allows more mixing of labels on each side of decision boundary ✗

That's M(C)LE. How about M(C)AP?

$$p(\mathbf{w} \mid Y, \mathbf{X}) \propto P(Y \mid \mathbf{X}, \mathbf{w}) p(\mathbf{w})$$

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} e^{\frac{-w_i^2}{2\kappa^2}}$$

Zero-mean Gaussian prior

$N(0, \kappa \mathbf{I})$

$\ln p(\mathbf{w}) \propto -\sum_i \frac{w_i^2}{2\kappa^2}$

- M(C)AP estimate

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

$$= \arg\max_{\mathbf{w}} \ln \prod_{j=1}^{n} P(y^j \mid \mathbf{x}^j, \mathbf{w}) + \ln p(\mathbf{w})$$

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{j=1}^{n} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) - \sum_{i=1}^{d} \frac{w_i^2}{2\kappa^2}$$

Still concave objective!

Penalizes large weights

Gradient Ascent for M(C)LE

Gradient ascent rule for w_0:

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \left. \frac{\partial l(\mathbf{w})}{\partial w_0} \right|_t$$

$$l(\mathbf{w}) = \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

Gradient Ascent for M(C)LE

Gradient ascent rule for w_0:

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \left. \frac{\partial l(\mathbf{w})}{\partial w_0} \right|_t$$

$$l(\mathbf{w}) = \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

$$\frac{\partial l(\mathbf{w})}{\partial w_0} = \sum_j \left[y^j - \frac{1}{1 + exp(w_0 + \sum_i^d w_i x_i^j)} \cdot exp(w_0 + \sum_i^d w_i x_i^j) \right]$$

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

M(C)AP – Gradient

- Gradient

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} \; e^{\frac{-w_i^2}{2\kappa^2}}$$

Zero-mean Gaussian prior

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

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

Same as before

$$\propto \frac{-w_i}{\kappa^2}$$

$\ln p(\mathbf{w}) \propto -\frac{\sum_i w_i^2}{2\kappa^2}$

$\frac{\partial \ln p(\mathbf{w})}{\partial w_i} \propto -\frac{w_i}{\kappa^2}$

Extra term Penalizes large weights

Penalization = Regularization

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 $Y \in \{y_1, \ldots, y_K\}$

for $k<K$

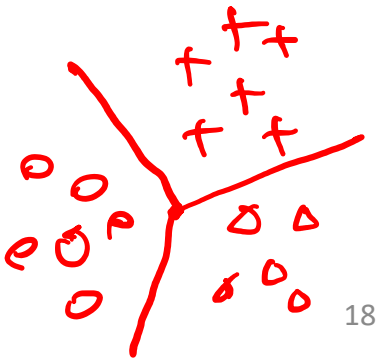
$$P(Y = y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^{d} w_{ki} X_i)}{1 + \sum_{j=1}^{K-1} \exp(w_{j0} + \sum_{i=1}^{d} w_{ji} X_i)}$$

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?



Gaussian Naïve Bayes vs. Logistic Regression

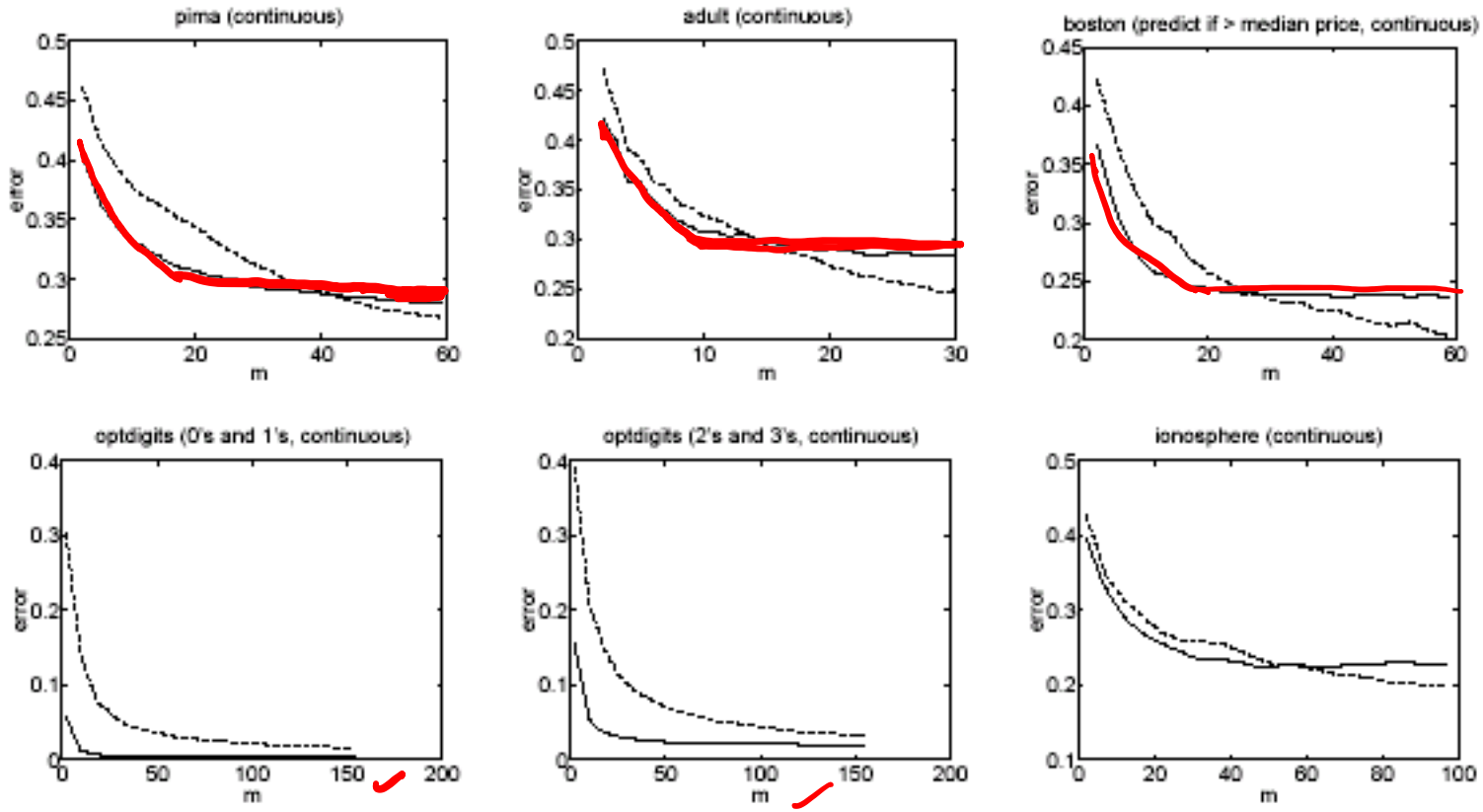
Set of Gaussian Naïve Bayes parameters (feature variance independent of class label)

Set of Logistic Regression parameters

- Representation equivalence (both can 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**

Experimental Comparison (Ng-Jordan'01)

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



More in Paper…

—— Naïve Bayes ----- Logistic Regression

Gaussian Naïve Bayes vs. Logistic Regression

Both GNB and LR have similar number $O(d)$ of parameters.

- GNB error converges faster with increasing number of samples as its parameter estimates are not coupled,

however,

- GNB has higher large sample error if conditional independence assumption DOES NOT hold.

GNB outperforms LR if conditional independence assumption holds.

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