VIRGINIA TECH

Classification: Logistic Regression

Lifu Huang

Computer Science, Virginia Tech

Feb. 9, 2021

Slides adapted from Luke Zettlemoyer, David Sontag, Bert Huang

Recap - Linear Regression

- Instances: $< X_j, t_j >$
- Learn: Mapping from X to $t(X)$
- Hypothesis space:
  - Given basic functions $\{h_1, h_2, \ldots, h_k\}$
  - $h_i(X) \in R$
  - define coefficients $W = \{w_1, w_2, \ldots, w_k\}$
  - linear regression: model is linear in the parameters
- Minimize the residual squared error:

$$\underbrace{t(\mathbf{x})}_{\text{data}} \approx \widehat{f}(\mathbf{x}) = \sum_i w_i h_i(\mathbf{x})$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

Recap: closed form solution

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} (\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})$$

$$\mathbf{F}(\mathbf{w}) = (\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})$$

$$\nabla_{\mathbf{w}} \mathbf{F}(\mathbf{w}) = \mathbf{0}$$

$$2\mathbf{H}^T(\mathbf{Hw} - \mathbf{t}) = \mathbf{0}$$

$$\mathbf{H}^T\mathbf{Hw} - \mathbf{H}^T\mathbf{t} = \mathbf{0}$$

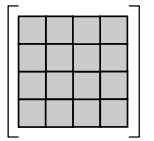
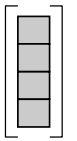
$$\mathbf{w}^* = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{t}$$

$$\frac{\partial X^T X}{\partial X} = 2X, \qquad \frac{\partial \beta^T X}{\partial X} = \beta$$

Normal Equations for the least squares problem

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \underbrace{(\mathbf{Hw} - \mathbf{t})^T (\mathbf{Hw} - \mathbf{t})}_{\text{residual error}}$$

$$\text{solution: } \mathbf{w}^* = \underbrace{(\mathbf{H}^T\mathbf{H})^{-1}}_{\mathbf{A}^{-1}} \underbrace{\mathbf{H}^T\mathbf{t}}_{\mathbf{b}} = \mathbf{A}^{-1}\mathbf{b}$$

where $\mathbf{A} = \mathbf{H}^T\mathbf{H} =$  $\mathbf{b} = \mathbf{H}^T\mathbf{t} =$ 

k×k matrix
for k basis functions

k×1 vector

Recap - Regularization

- Regularization: adding constraint on the coefficients *W*
- LASSO (Least Absolute Shrinkage and Selection Operator)
  - L1 norm: add the sum of magnitudes of the coefficients

$$\hat{w}_{LASSO} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

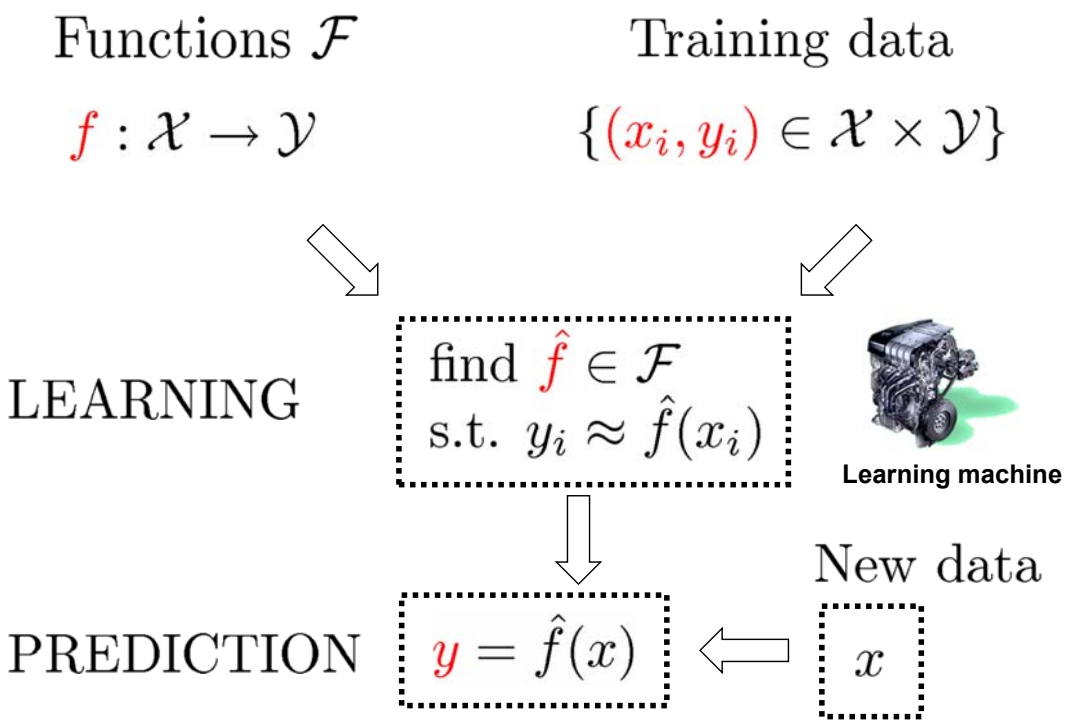
- Ridge Regression
  - L2 norm: add a squared penalize for large weights
  - Explicitly writing out bias feature ($h_0 = 1$), which is not penalized

$$\hat{w}_{ridge} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

λ is hyperparameter that balances tradeoffs

Supervised Learning: find f

- Given: Training set $\{(x_i, y_i) | i = 1 \dots n\}$
- Find: A good approximation to $f: X \rightarrow Y$
- Examples: what are X and Y?
  - Spam Detection: Map email to {Spam, Ham}
  - Digit Recognition: Map pixels to {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}



Lets take another probabilistic approach - discriminative

- Previously: directly estimate the data distribution $P(X, Y)$!
  - challenging due the size of distribution
  - make Naïve Bayes assumption: only need $P(X_i, Y)$
- But wait, we classify according to
  - $\max_{Y_i} P(Y_i|X)$
- Why not learn $P(Y|X)$ directly?

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

Logistic Regression

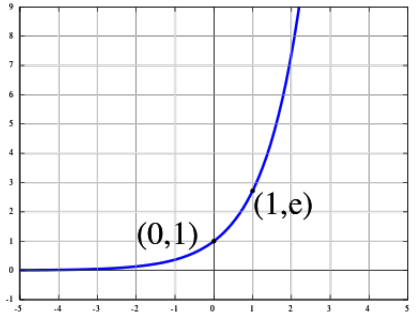
- Learn $P(Y|\boldsymbol{X})$ directly
  - Reuse ideas from regression, but let y-interpret define the probability

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

  - with normalization constants

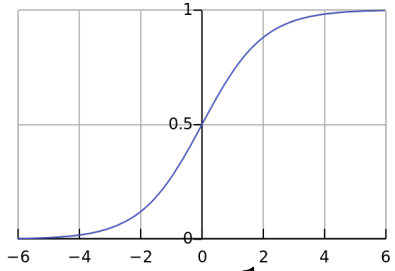
$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$
$$P(Y=1|\mathbf{X},\mathbf{w}) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

Exponential:



$$y = e^x = exp(x)$$

Logistic function:



$$y = \frac{1}{1+exp(-x)}$$

Logistic Regression: decision boundary

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

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

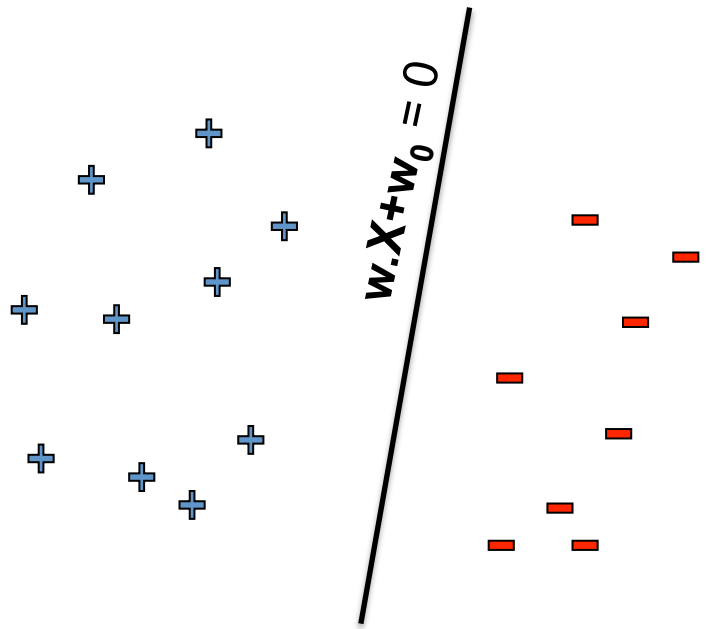
- Prediction: Output the Y with highest $P(Y|X)$
  - For binary Y, output $Y=1$ if

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

$$1 < \exp(w_0 + \sum_{i=1}^{n} w_i X_i)$$

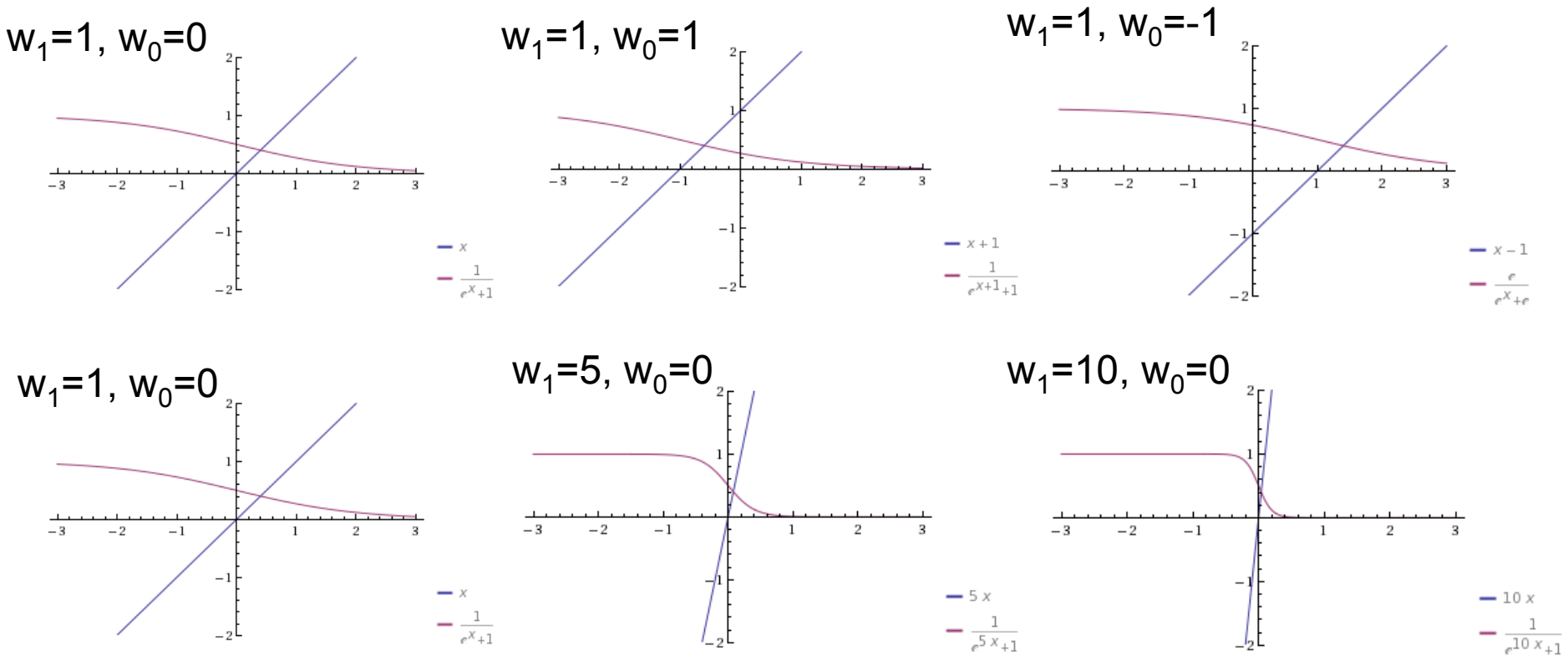
$$0 < w_0 + \sum_{i=1}^{n} w_i X_i$$

A Linear Classifier



Visualizing 1D Inputs

$$P(Y = 0|\mathbf{X}, \mathbf{w}) = \frac{1}{1 + exp(w_0 + w_1 x_1)}$$

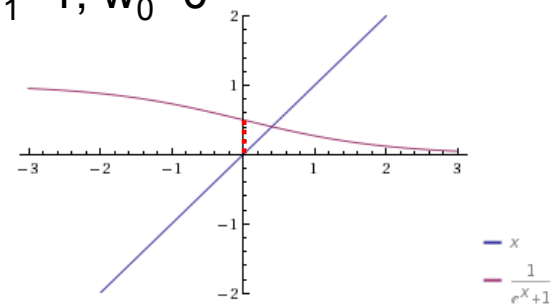


- Defines a probability distribution over Y in {0, 1} for every possible input X
- Decision boundary: P(Y=0|X,W) = 0.5 when at the y=0 point on the line
- Slot of the line defines how quickly probabilities go to 0 or 1 around decision boundary

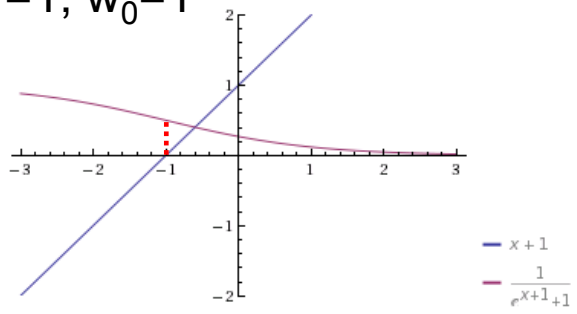
Visualizing 1D Inputs

$$P(Y = 0|\mathbf{X}, \mathbf{w}) = \frac{1}{1 + exp(w_0 + w_1 x_1)}$$

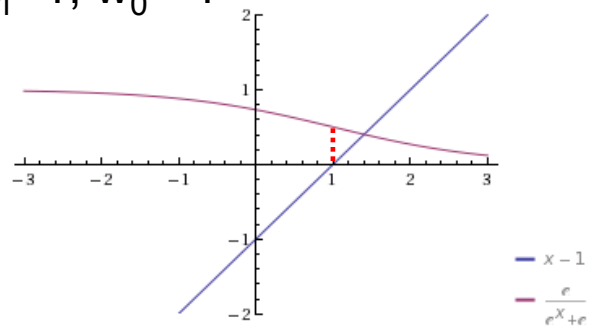
w_1=1, w_0=0



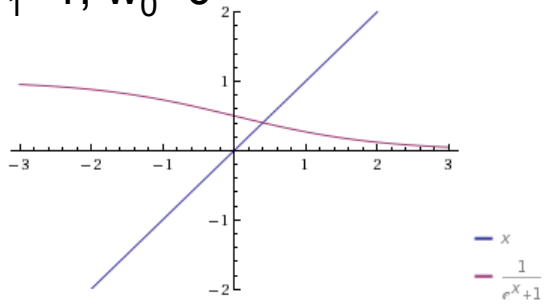
w_1=1, w_0=1



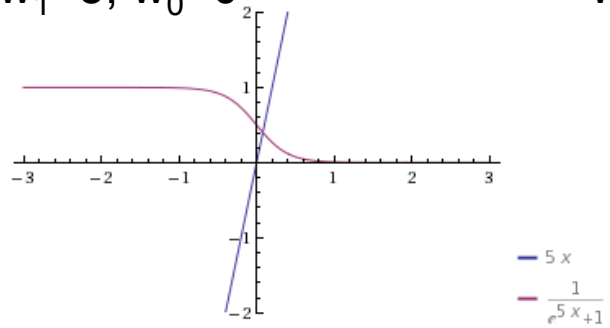
w_1=1, w_0=-1



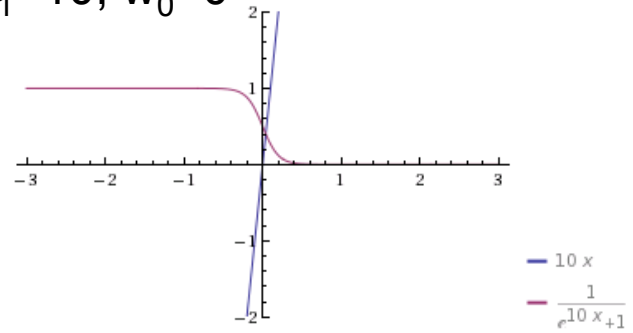
w_1=1, w_0=0



w_1=5, w_0=0



w_1=10, w_0=0



- Defines a probability distribution over Y in {0, 1} for every possible input X
- Decision boundary: P(Y=0|X,W) = 0.5 when at the y=0 point on the line
- Slot of the line defines how quickly probabilities go to 0 or 1 around decision boundary

Loss Functions / Learning Objectives

- Generative (e.g., Naïve Bayes) Loss Function:
  - Maximize Data Likelihood

$$\begin{aligned} \ln P(\mathcal{D} \mid \mathbf{w}) &= \sum_{j=1}^{N} \ln P(\mathbf{x}^j, y^j \mid \mathbf{w}) \\ &= \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) + \sum_{j=1}^{N} \ln P(\mathbf{x}^j \mid \mathbf{w}) \end{aligned}$$

- Discriminative (e.g., Logistic Regression) Loss Function:
  - Maximize Conditional Data Likelihood

$$\ln P(\mathcal{D}_Y \mid \mathcal{D}_\mathbf{X}, \mathbf{w}) = \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w})$$

  - Doesn't waste effort learning $P(X)$ – focuses on $P(Y|X)$ all that matters for classification

Conditional Log Likelihood (the binary case only)

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

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

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

equal because y^j is in {0,1}

$$l(\mathbf{w}) = \sum_j y^j \ln P(y^j=1|\mathbf{x}^j,\mathbf{w}) + (1-y^j)\ln P(y^j=0|\mathbf{x}^j,\mathbf{w})$$

remaining steps: substitute definitions, expand logs, and simplify

$$= \sum_j y^j \ln \frac{e^{w_0+\sum_i w_i X_i}}{1+e^{w_0+\sum_i w_i X_i}} + (1-y^j)\ln \frac{1}{1+e^{w_0+\sum_i w_i X_i}}$$

$$\cdots$$

$$= \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1+exp(w_0+\sum_i^n w_i x_i^j))$$

Logistic Regression Parameter Estimation: Maximize Conditional Log Likelihood

$$
\begin{aligned}
l(\mathbf{w}) &\equiv \ln \prod_j P(y^j|\mathbf{x}^j, \mathbf{w}) \\
&= \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j)) \\
&= \sum_{\mathrm{j}} y^j x^j w - In(1 + \exp(x^j w))
\end{aligned}
$$

Maximize Conditional Log Likelihood: Derivative

$$l(\mathbf{w}) = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j))$$

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

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

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

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

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j \left(y^j - P(Y^j = 1|x^j, w) \right)$$

Maximize Conditional Log Likelihood

$$l(\mathbf{w}) = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j))$$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j \left(y^j - P(Y^j = 1 | x^j, w)\right)$$

- No closed form solution to maximize $l(\boldsymbol{w})$
- Gradient Descent and Gradient Ascent !!
  - Gradient Descent: find a local minimum for a differentiable function

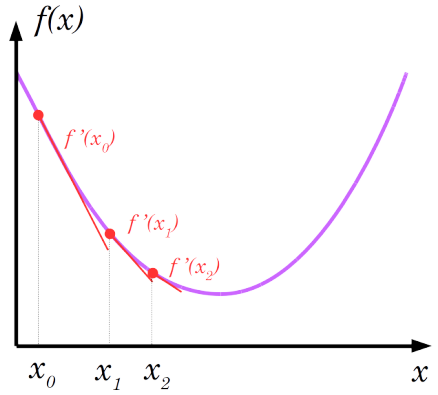
$$w_i^{t+1} \leftarrow w_i^t - \eta \frac{\partial l(w)}{\partial w_i}$$

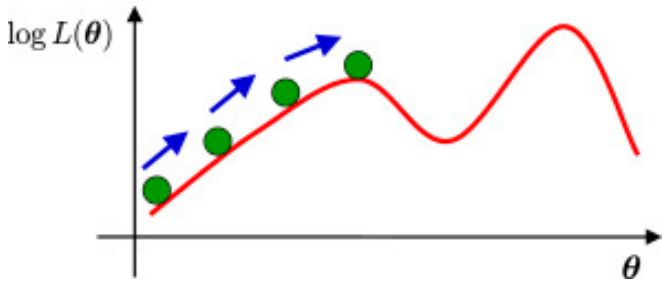
  - Gradient Ascent: find a local maximum for a differentiable function

$$w_i^{t+1} \leftarrow w_i^t + \eta \frac{\partial l(w)}{\partial w_i}$$

Gradient: direction and rate of fastest increase of the function

The **direction** of the **gradient** is the **direction** in which the function increases most quickly



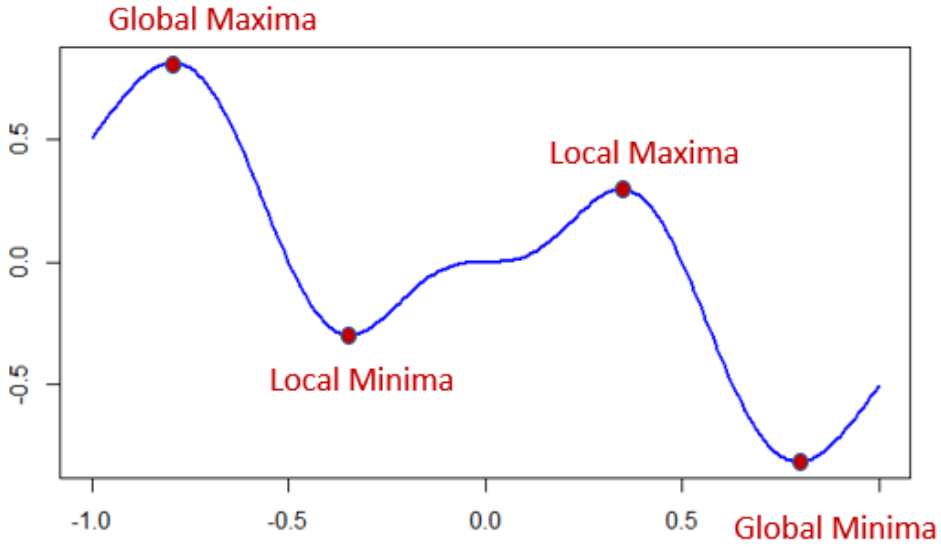


Gradient Ascent

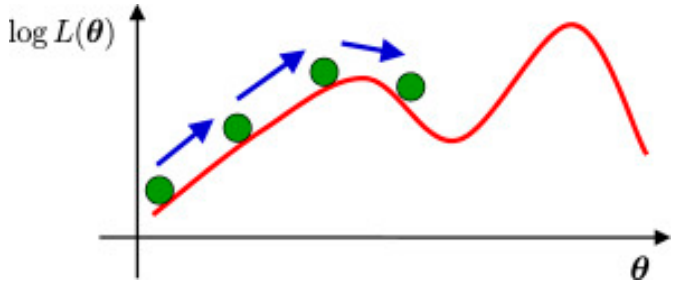
- Not guaranteed to find the global maxima

$$w_i^{t+1} \leftarrow w_i^t + \eta \frac{\partial l(w)}{\partial w_i}$$

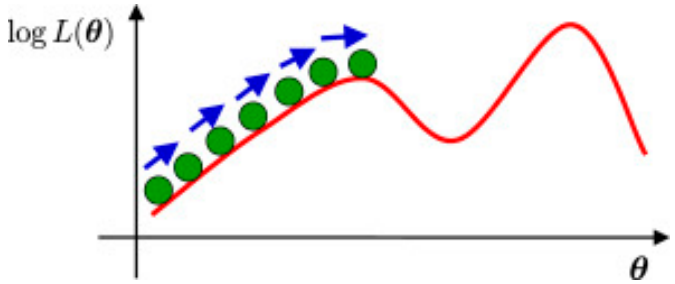
Learning rate



- Learning Rate: determine the step size at each iteration while moving toward the minima or maxima



Learning rate too large



Learning rate too small

Maximize Conditional Log Likelihood: Gradient Ascent

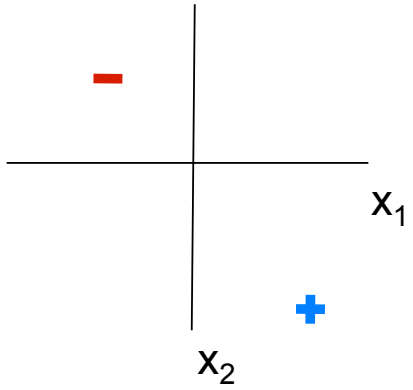
$$l(\mathbf{w}) = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j))$$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j \left(y^j - P(Y^j = 1 | x^j, w)\right)$$

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

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

x_1	x_2	y
3	-3	1
-2	2	0



t=0: w = $[w_0,w_1,w_2]$ = [0,0,0]

$P(Y^0=1|x^0,w)$ α exp(0+0*3+0*-3) = 0.5

$P(Y^1=1|x^1,w)$ α exp(0+0*-2+0*2) = 0.5

i=0, j=0: $x_0^0(y^0-P(Y^0=1|x^0,w))$ = 1(1-0.5) = 0.5

i=0, j=1: $x_0^1(y^1-P(Y^1=1|x^1,w))$ = 1(0-0.5) = -0.5

i=1, j=0: $x_1^0(y^0-P(Y^0=1|x^0,w))$ = 3(1-0.5) = 1.5

i=1, j=1: $x_1^1(y^1-P(Y^1=1|x^1,w))$ = -2(0-0.5) = 1.0

i=2, j=0: $x_2^0(y^0-P(Y^0=1|x^0,w))$ = -3(1-0.5) = -1.5

i=2, j=1: $x_2^1(y^1-P(Y^1=1|x^1,w))$ = 2(0-0.5) = -1.0

grad = [0.5-0.5, 1.5+1.0, -1.5-1] = [0,2.5,-2.5]

t=1: η=0.1 → w = [0,0,0] + 0.1 * [0,2.5,-2.5] = [0,0.25,-0.25]

$P(Y^0=1|x^0,w)$ α exp(0+0.25*3-0.25*-3) = 0.82

$P(Y^1=1|x^1,w)$ α exp(0+0.25*-2-0.25*2) = 0.27

i=0, j=0: $x_0^0(y^0-P(Y^0=1|x^0,w))$ = 1(1-0.82) = 0.18

i=0, j=1: $x_0^1(y^1-P(Y^1=1|x^1,w))$ = 1(0-0.27) = -0.27

i=1, j=0: $x_1^0(y^0-P(Y^0=1|x^0,w))$ = 3(1-0.82) = 0.54

i=1, j=1: $x_1^1(y^1-P(Y^1=1|x^1,w))$ = -2(0-0.27) = 0.54

i=2, j=0: $x_2^0(y^0-P(Y^0=1|x^0,w))$ = -3(1-0.82) = -0.54

i=2, j=1: $x_2^1(y^1-P(Y^1=1|x^1,w))$ = 2(0-0.27) = -0.54

grad = [0.13-0.27, 0.54+0.54, -0.54-0.54] = [-0.14,1.04,-1.04]

Gradient Ascent for Logistic Regression

Gradient ascent algorithm: (learning rate $\eta > 0$)

`do:`

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

`For i=1…n: (iterate over weights)`

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

`until “change” < ε`

Loop over training examples!

Logistic regression for multi-class classification

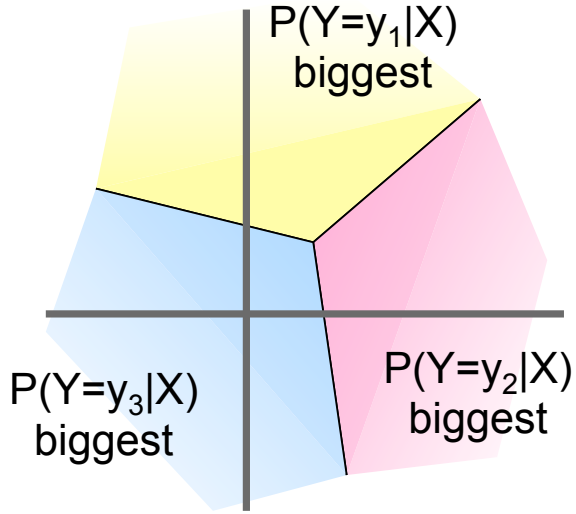
- Logistic regression in more general case, where set of possible Y is $\{y_1, y_2, \ldots, y_R\}$
  - Define a weight vector w_i for each $y_i, i = 1, \ldots, R-1$

$$P(Y = 1|X) \propto \exp(w_{10} + \sum_i w_{1i} X_i)$$

$$P(Y = 2|X) \propto \exp(w_{20} + \sum_i w_{2i} X_i)$$

$$\ldots$$

$$P(Y = r|X) = 1 - \sum_{j=1}^{r-1} P(Y = j|X)$$



  - For $k < R$

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

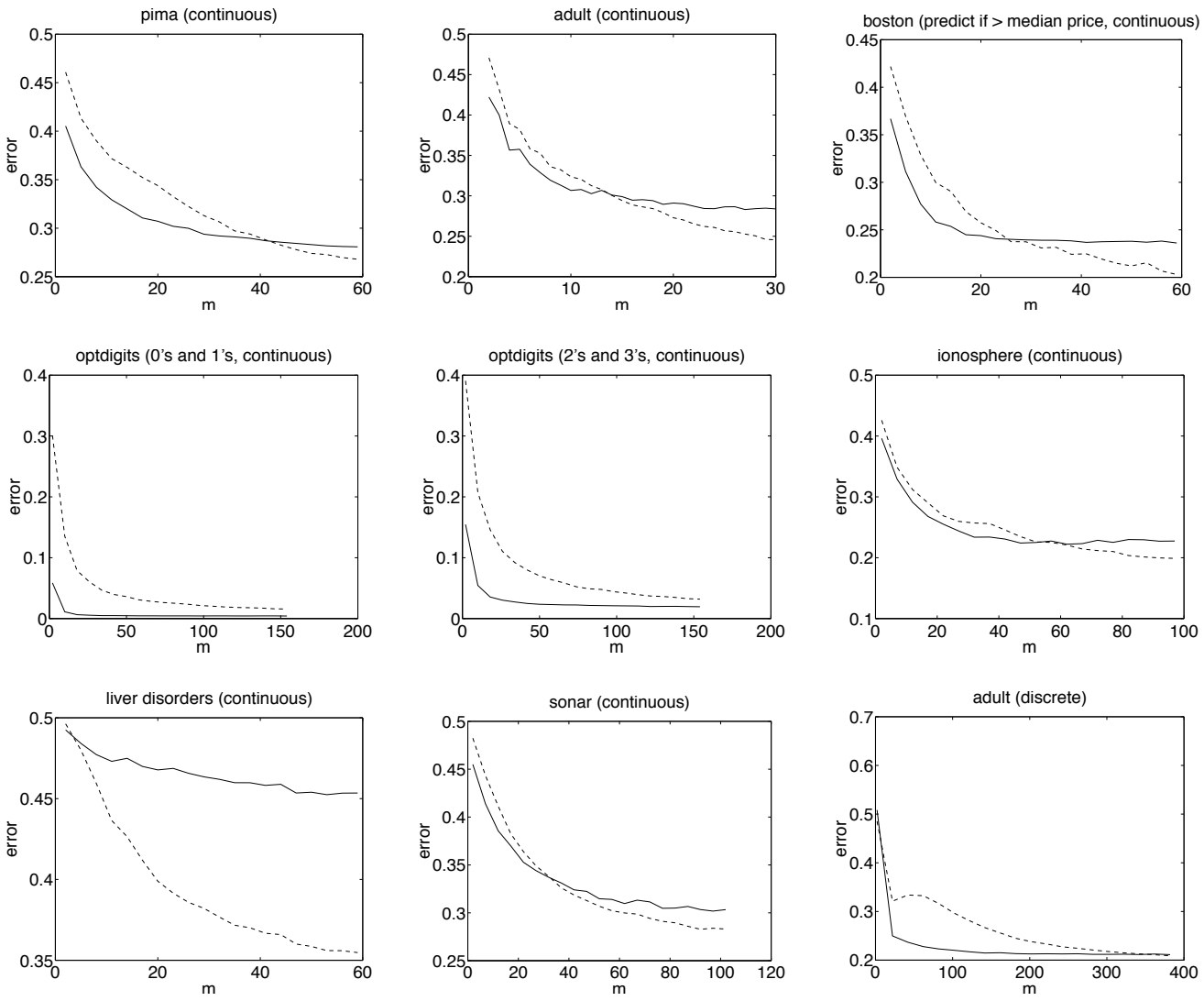
  - For $k = R$ (normalization, so no weights for this class)

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

Features can be discrete or continuous!

Logistic Regression v.s. (Gaussian) Naïve Bayes

- Both for classification, but …
  - Logistic Regression: directly estimate the likelihood of $P(Y|X)$ ← Discriminative Classifier
  - Naïve Bayes: estimate the data distribution $P(X,Y)$ ← Generative Classifier
- Assumptions
  - Naïve Bayes: assume all X_i are conditionally independent given Y
  - Logistic Regression: makes no assumptions
  - When model correct:
    - GNB and LR produce identical classifiers
  - When model incorrect:
    - LR is less biased – does not assume conditional independence
    - therefore, LR expected to outperform GNB
- Loss Function
  - Optimize different functions! Obtain different solutions



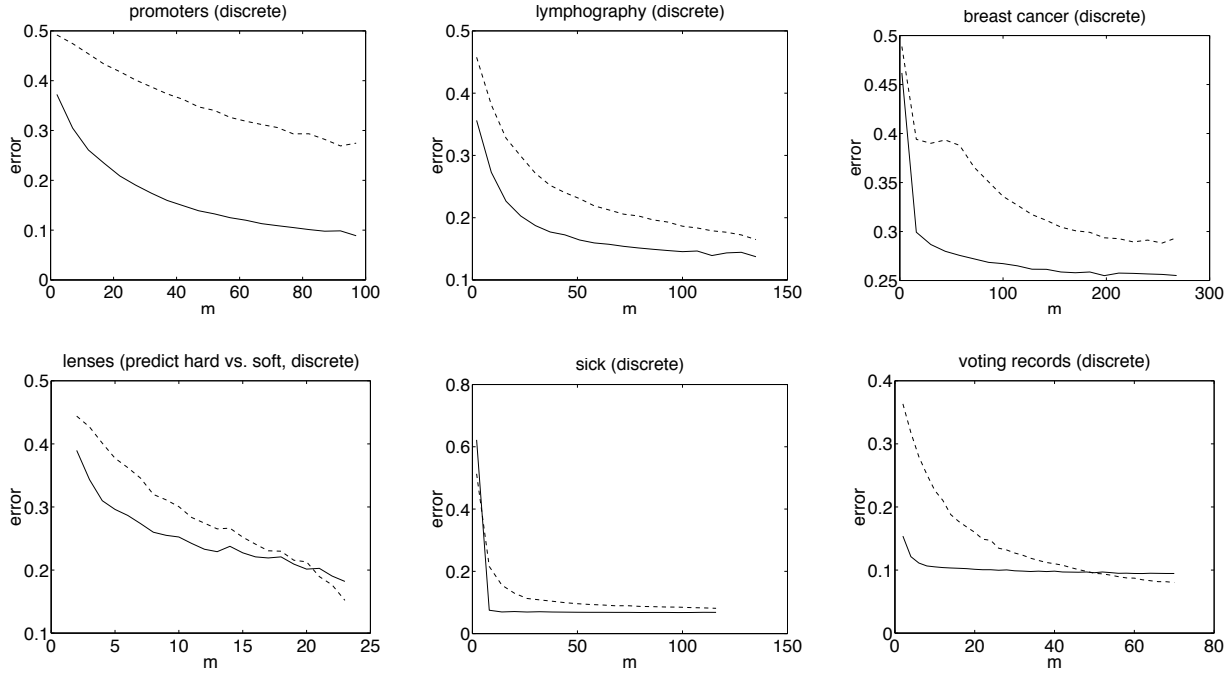


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. m (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

Some experiments from UCI data sets

from *On Discriminative v.s. Generative classifiers: A comparison of logistic regression and naïve bayes*.
Andrew Y. Ng, Michael I. Jordan. NIPS'2001

What you need to know about Logistic Regression

- LR is a linear classifier
- LR optimized by conditional likelihood
  - no closed form solution
  - gradient ascent v.s. gradient descent
  - how to optimize LR with gradient ascent
- Gaussian Naïve Bayes v.s. Logistic Regression
  - discriminative / generative
  - assumptions
  - optimization