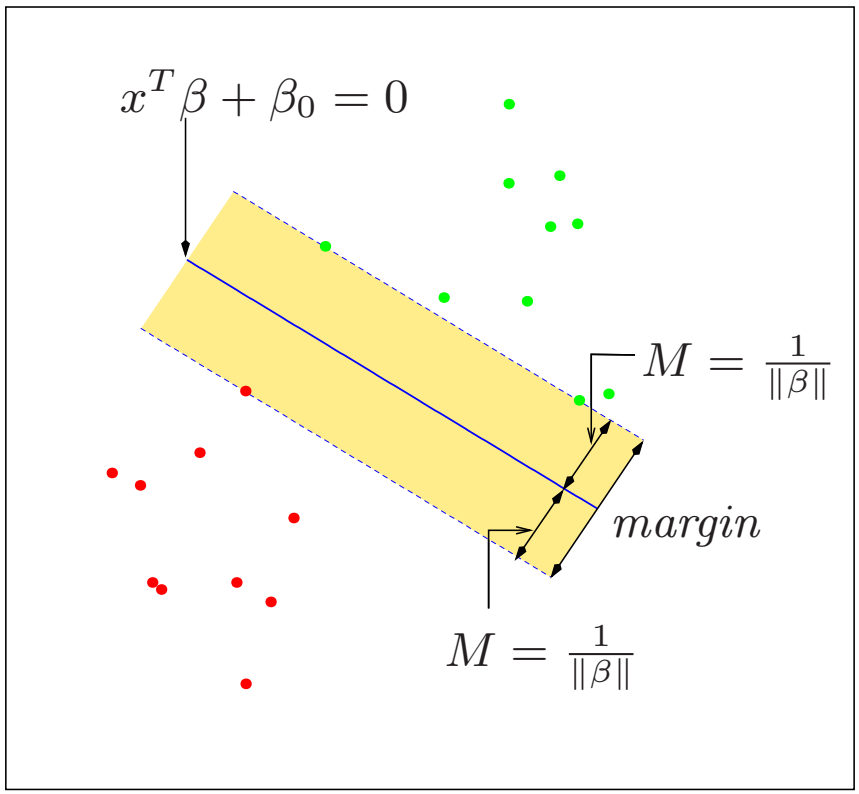
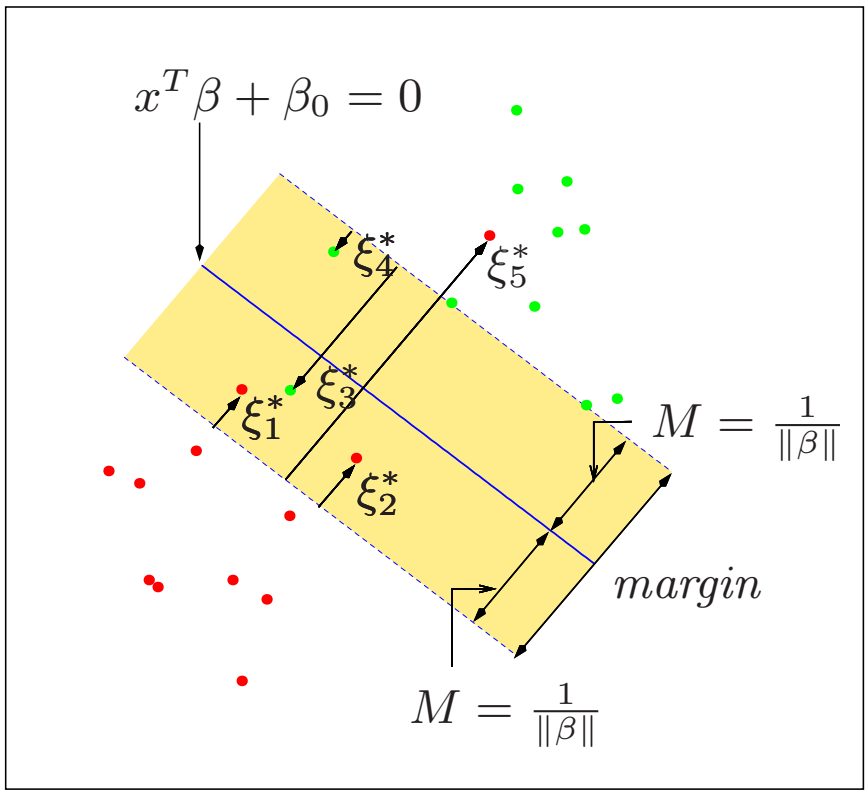
CPSC 8420 Advanced Machine Learning
Week 12: Logistic Regression

Dr. Kai Liu

November 13, 2020

Support Vector Machine



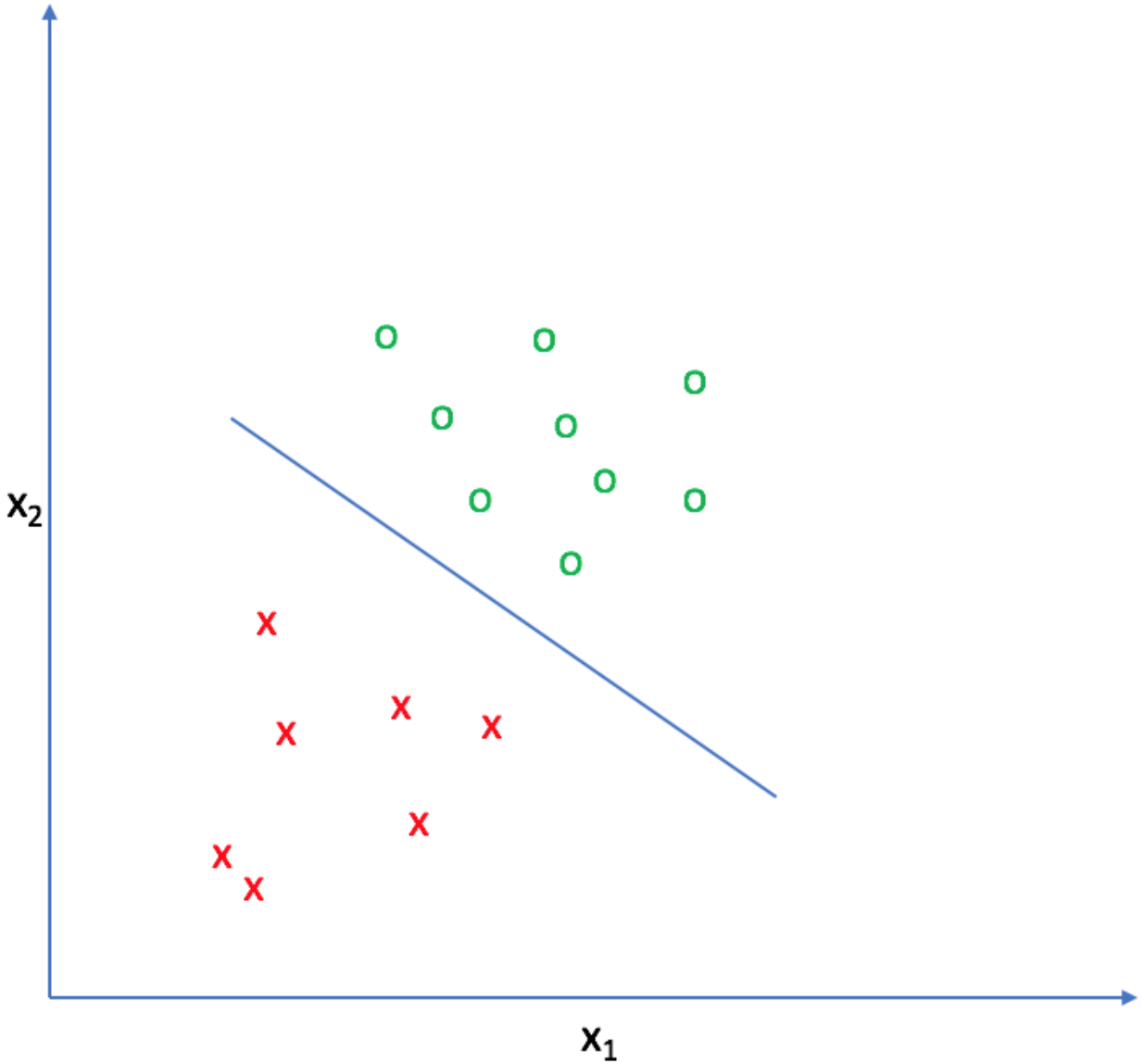


K-means v.s. NMF

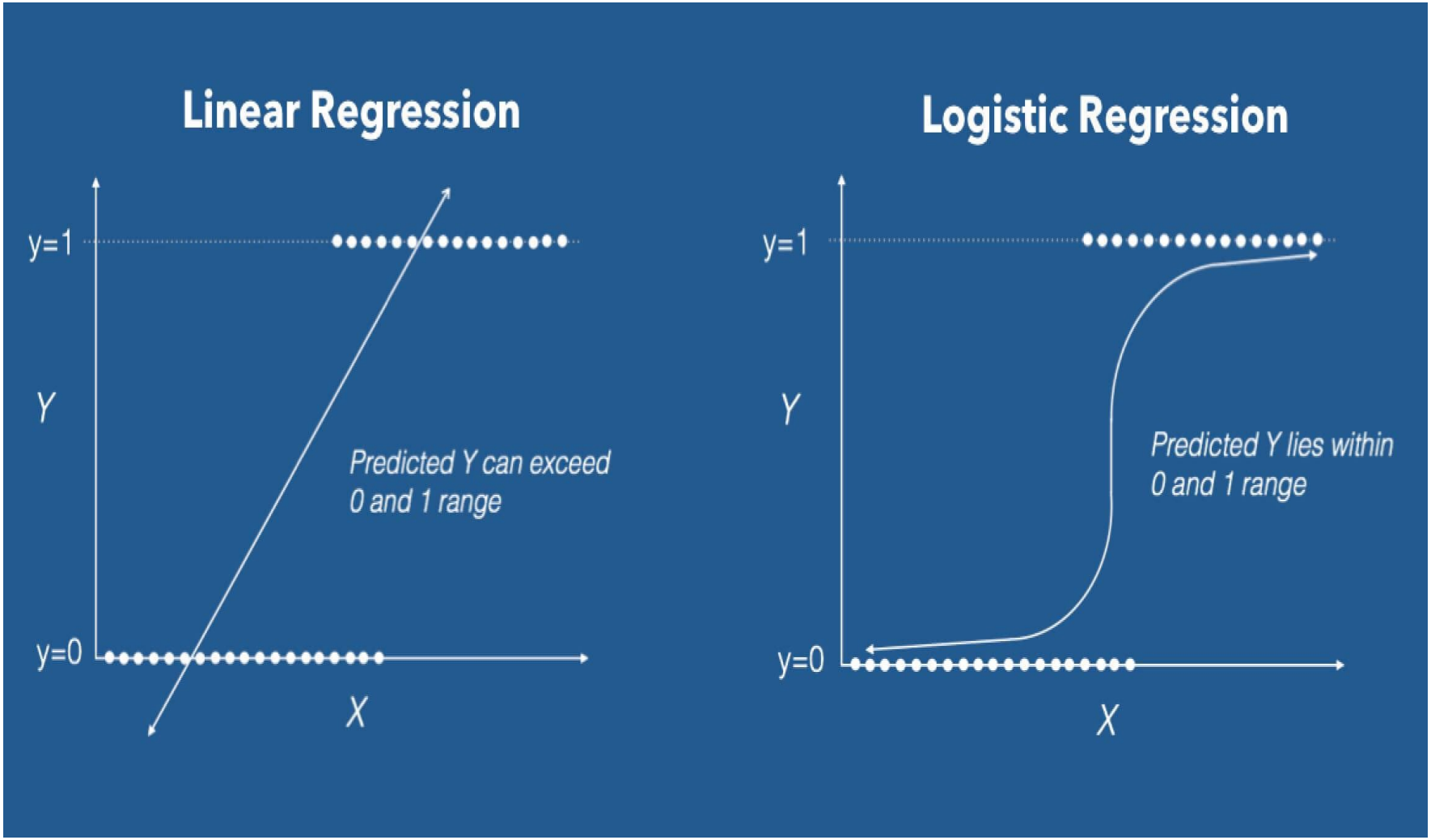
NMF	a	b	c	d
C1	0.9	0.15	0.8	0.25
C2	0.2	0.8	0.1	0.8

K-means	a	b	c	d
C1	1	0	1	0
C2	0	1	0	1

Decision Boundary (Perceptron)



Discrete Response



Surrogate Relaxation

$$p(y=1|x) = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}, \quad z = w^T x + b \tag{1}$$

Considering the objective is not continuous, thus we use a surrogate function:

$$y = \frac{1}{1 + e^{-(w^T x + b)}}, \tag{2}$$

which is even differentiable.

Surrogate Relaxation in RatioCut

If we introduce indicator vector: $h_j \in \{h_1, h_2, ..h_k\}, j \in [1, k]$, for any vector $h_j \in R^n$, we define: $h_{ij} = \begin{cases} 0 & v_i \notin A_j \\ \frac{1}{\sqrt{|A_j|}} & v_i \in A_j \end{cases}$, then:

$$
\begin{aligned}
h_i^T L h_i &= \frac{1}{2} \sum_{m=1} \sum_{n=1} w_{mn} (h_{im} - h_{in})^2 \\
&= \frac{1}{2} (\sum_{m \in A_i, n \notin A_i} w_{mn} (\frac{1}{\sqrt{|A_i|}} - 0)^2 + \sum_{m \notin A_i, n \in A_i} w_{mn} (0 - \frac{1}{\sqrt{|A_i|}})^2 \\
&= \frac{1}{2} (\sum_{m \in A_i, n \notin A_i} w_{mn} \frac{1}{|A_i|} + \sum_{m \notin A_i, n \in A_i} w_{mn} \frac{1}{|A_i|} \\
&= \frac{1}{2} (cut(A_i, \overline{A}_i) \frac{1}{|A_i|} + cut(\overline{A}_i, A_i) \frac{1}{|A_i|}) \\
&= \frac{cut(A_i, \overline{A}_i)}{|A_i|}
\end{aligned}
$$

Surrogate Relaxation in RatioCut

For a subset, its RatioCut is $h_i^T L h_i$, then for k subsets we have:

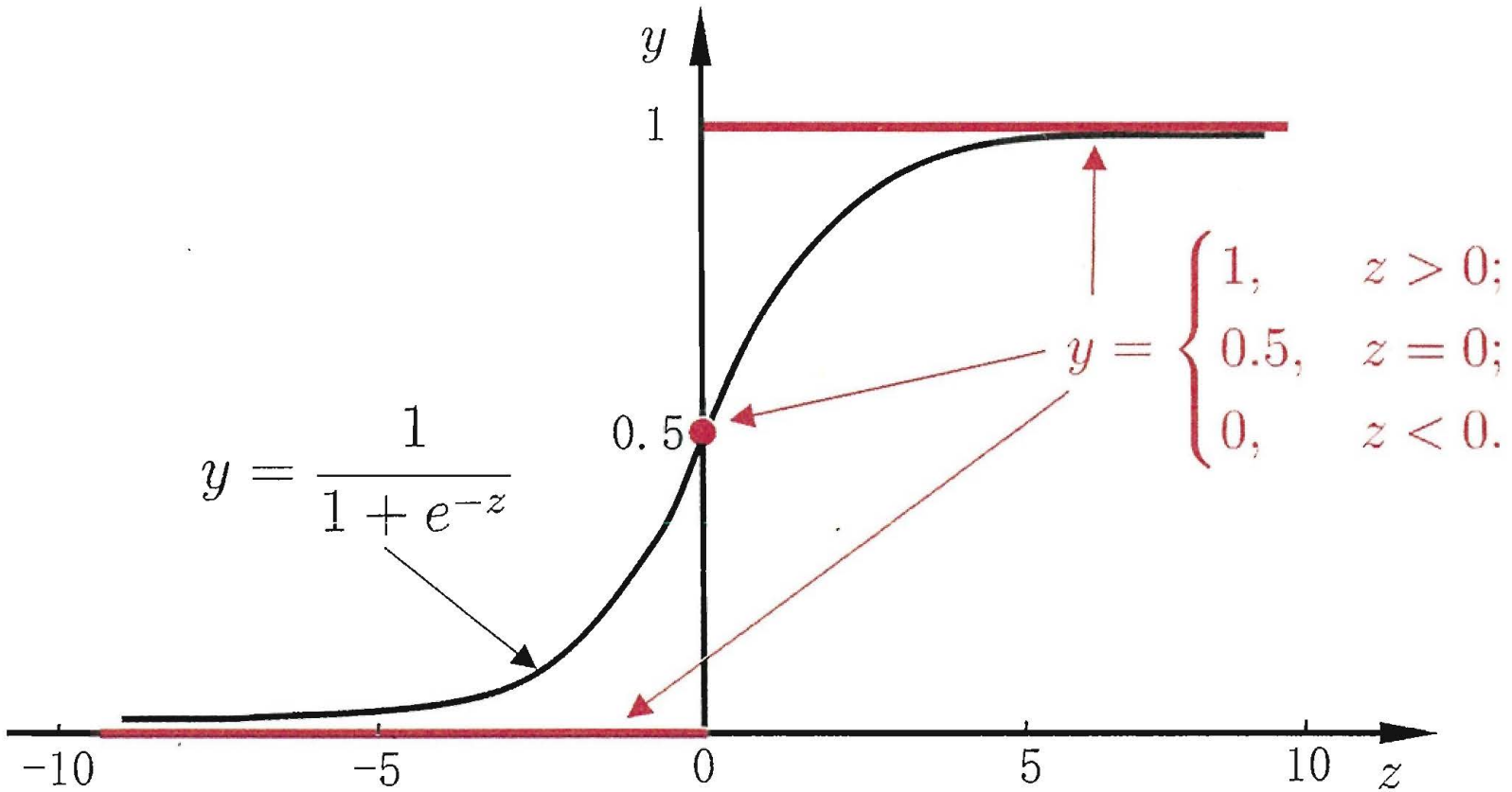
$$RatioCut(A_1, A_2, ...A_k) = \sum_{i=1}^{k} h_i^T L h_i = \sum_{i=1}^{k} (H^T LH)_{ii} = tr(H^T LH) \tag{3}$$

Unfortunately, this is an integer programming problem which we cannot solve efficiently. Instead, we relax the latter requirement and simply search an orthonormal matrix $H \in \mathbb{R}^{n \times k}$. By observing $H^T H = I$, we have the objective as:

$$\underbrace{arg\ min}_{H}\ tr(H^T LH)\ \ s.t.\ H^T H = I \tag{4}$$

Sigmoid Function

If we denote $z = w^T x + b$, then Sigmoid Function is 'S' shape:



Properties of Sigmoid Function

1. Bounded within $(0, 1)$
2. Monotonically increasing w.r.t. z
3. $ln\frac{y}{1-y} = w^T x + b$, which is called 'logit' or log odds
4. $y' = y(1 - y)$

Logistic Regression is not a Regression model, but a classification one. It will first fit $z = w^T x + b$, and then determine the probability to each class.

Likelihood Function

Recall that:

$$\begin{aligned} P(Y = 1|x) &= p(x) \\ P(Y = 0|x) &= 1 - p(x) \end{aligned} \tag{5}$$

where we can combine the two cases as one:

$$P(y_i|x_i, \theta) = [p(x_i)]^{y_i}[1 - p(x_i)]^{1-y_i},$$

then the likelihood function is:

$$L(w) = \prod_{i=1}^{m} [p(x_i)]^{y_i}[1 - p(x_i)]^{1-y_i}$$

Reformulation

Usually instead of

$$L(w) = \prod_{i=1}^{m} [p(x_i)]^{y_i} [1 - p(x_i)]^{1-y_i},$$

we take the log as:

$$\begin{aligned} L(w) &= \sum [y_i ln p(x_i) + (1 - y_i) ln(1 - p(x_i))] \\ &= \sum [y_i ln \frac{p(x_i)}{1 - p(x_i)} + ln(1 - p(x_i))] \\ &= \sum [y_i [\langle w, x_i \rangle + b] - ln(1 + e^{\langle w, x_i \rangle + b}) \end{aligned} \tag{6}$$

now if we denote $\beta = (w; b), \hat{x} = (x; 1)$, we formulate the objective as:

$$\min_{\beta} \sum_{i=1}^{m} ln(1 + e^{\langle \beta, \hat{x_i} \rangle}) - y_i \langle \beta, \hat{x_i} \rangle \tag{7}$$

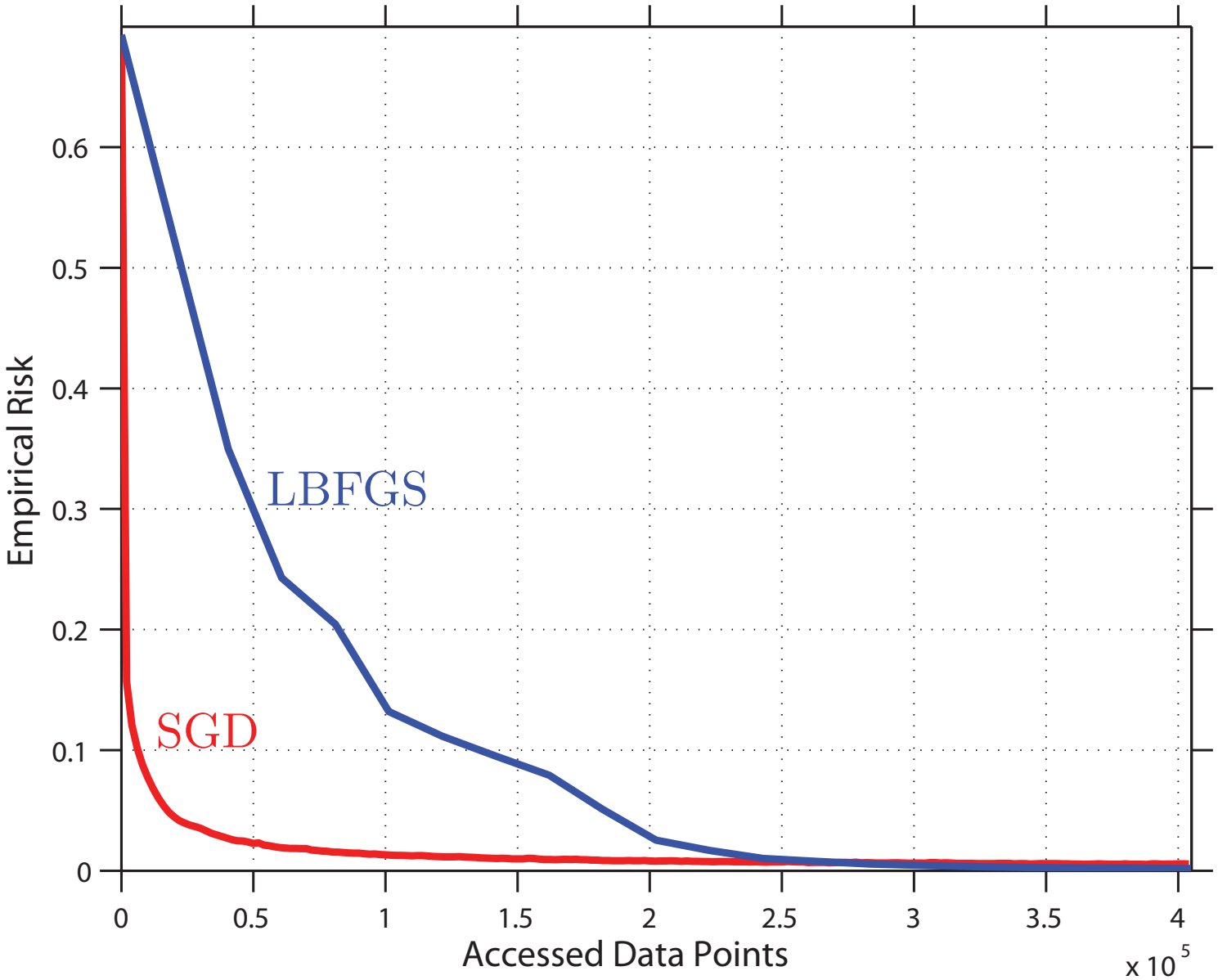
Optimization with Stochastic Gradient Descent (SGD)

$$\min_{\beta} \sum_{i=1}^{m} ln(1 + e^{\langle \beta, \hat{x_i} \rangle}) - y_i \langle \beta, \hat{x_i} \rangle \tag{8}$$

Objective function is convex, we can utilize gradient descent-based method to optimize the solution. Different with vanilla Gradient Desent, SGD only need compute the gradient at a single point with each update

$$\begin{aligned} g &= \frac{\partial J(w)}{\partial w} = (p(x_i) - y_i)x_i \\ w^{k+1} &= w^k - \alpha g \end{aligned} \tag{9}$$

SGD IS Efficient



Newton's Method

By Talor's expansion,

$$\varphi(w) = J(w^k) + \langle \nabla, w - w^k \rangle + \frac{1}{2}\langle H(w - w^k), w - w^k \rangle, \tag{10}$$

where w is close to w^k. Now set $\varphi^{'}(w) = 0$, we have:

$$w^{k+1} = w^k - H_k^{-1} \cdot \nabla \tag{11}$$

Let's reconsider Least Squares:

$$\min_{\beta} \|y - A\beta\|^2 \tag{12}$$

By leveraging Newton's method we have $H = A'A, \nabla = A'A\beta - A'y$, then $\beta^+ = \beta - H^{-1}\nabla = (A'A)^{-1}A'y$, that is it only needs one iteration to the optimal solution.

Logistic Regression by Newton's Method

Recall the objective:

$$\min_{\beta} \sum_{i=1}^{m} ln(1 + e^{\langle \beta, \hat{x}_i \rangle}) - y_i \langle \beta, \hat{x}_i \rangle \tag{13}$$

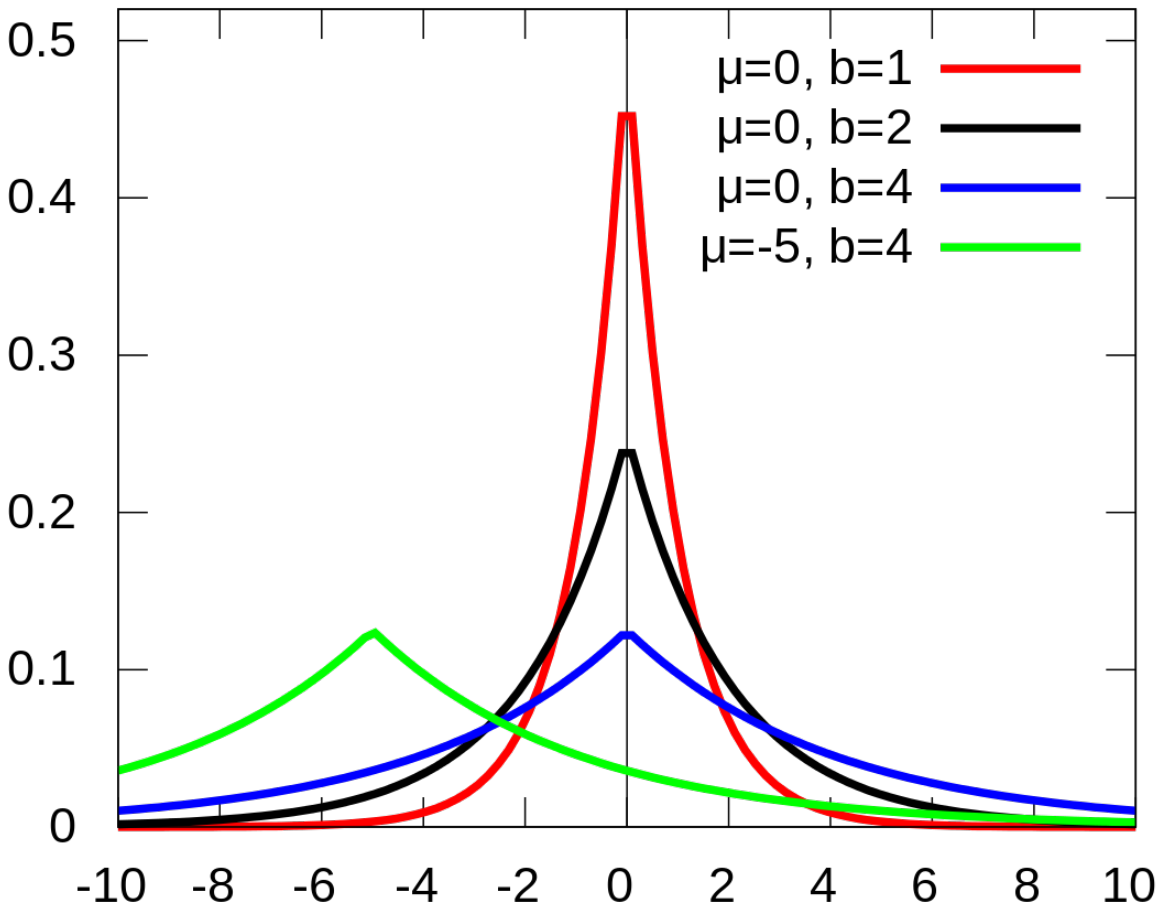
we have the first order and second order derivative as:

$$\begin{aligned} \nabla &= \sum_{i=1}^{m} [p(x_i) - y_i] x_i = X(p - y) \\ H &= \sum_{i=1}^{m} \hat{x}_i \hat{x}_i^T p(x_i)[1 - p(x_i)] = XWX^T, \end{aligned} \tag{14}$$

where W is an $m \times m$ diagonal matrix of weights with i-th diagonal element $p(x_i)[1 - p(x_i)]$.

Laplace Distribution

$$f(w|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|w-\mu|}{b}\right) \tag{15}$$



Likelihood Function

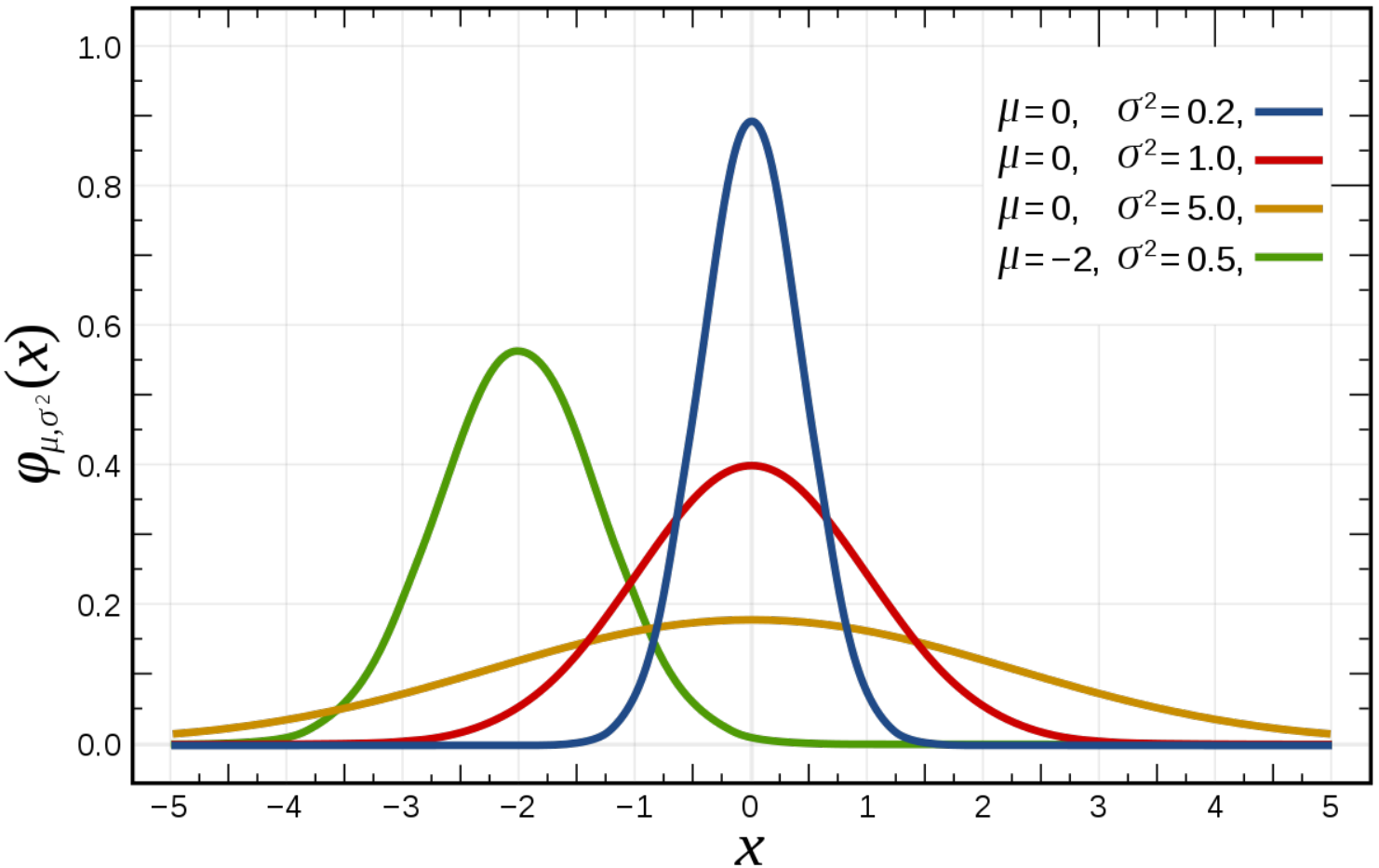
$$
\begin{aligned}
L(w) &= P(y|w,x)P(w) \\
&= \prod_{i=1}^{m} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} \prod_{j=1}^{d} \frac{1}{2b} \exp\left(-\frac{|w_j|}{b}\right)
\end{aligned} \tag{16}
$$

Note that $\max L(w) = -\min log[L(w)]$, then it is equivalent to optimize:

$$
\begin{aligned}
&-\sum_i [y_i \ln p(x_i) + (1-y_i) ln(1-p(x_i))] + \frac{1}{b}\sum_j |w_j| \\
&-\sum_i [y_i \ln p(x_i) + (1-y_i) ln(1-p(x_i))] + \frac{1}{b}\|w\|_1
\end{aligned} \tag{17}
$$

Gaussian Distribution

$$f(w|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(w-\mu)^2}{2\sigma^2}\right) \tag{18}$$



Likelihood Function

$$
\begin{aligned}
L(w) &= P(y|w,x)P(w) \\
&= \prod_{i=1}^{N} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{w_j^2}{2\sigma^2}\right)
\end{aligned} \tag{19}
$$

Note that $\max L(w) = -\min log[L(w)]$, then it is equivalent to optimize:

$$
\begin{aligned}
&-\sum_i [y_i \ln p(x_i) + (1-y_i) ln(1-p(x_i))] + \frac{1}{2\sigma^2}\sum_j w_j^2 \\
&-\sum_i [y_i \ln p(x_i) + (1-y_i) ln(1-p(x_i))] + \frac{1}{2\sigma^2}\|w\|^2
\end{aligned} \tag{20}
$$

Sparsity

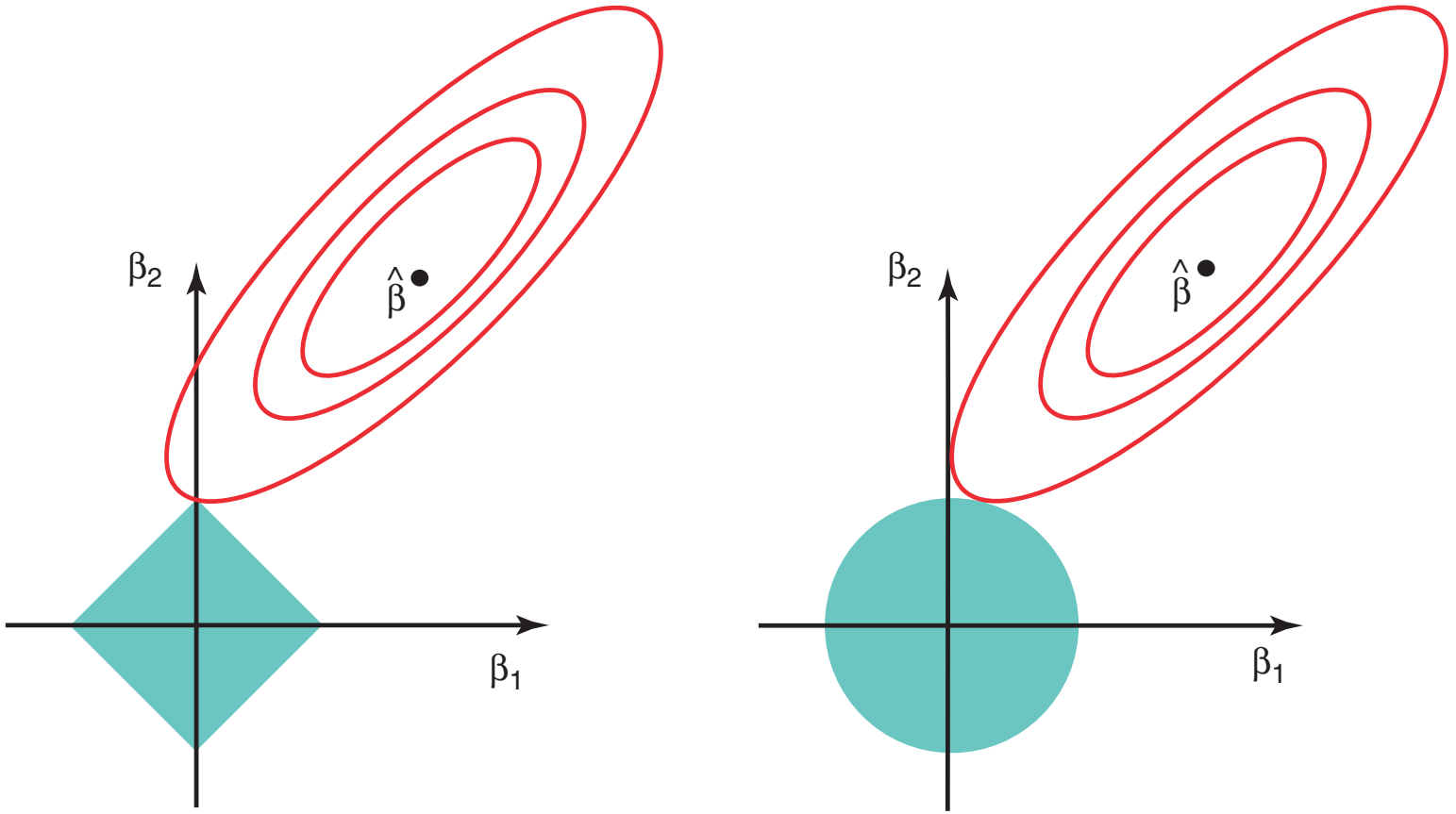
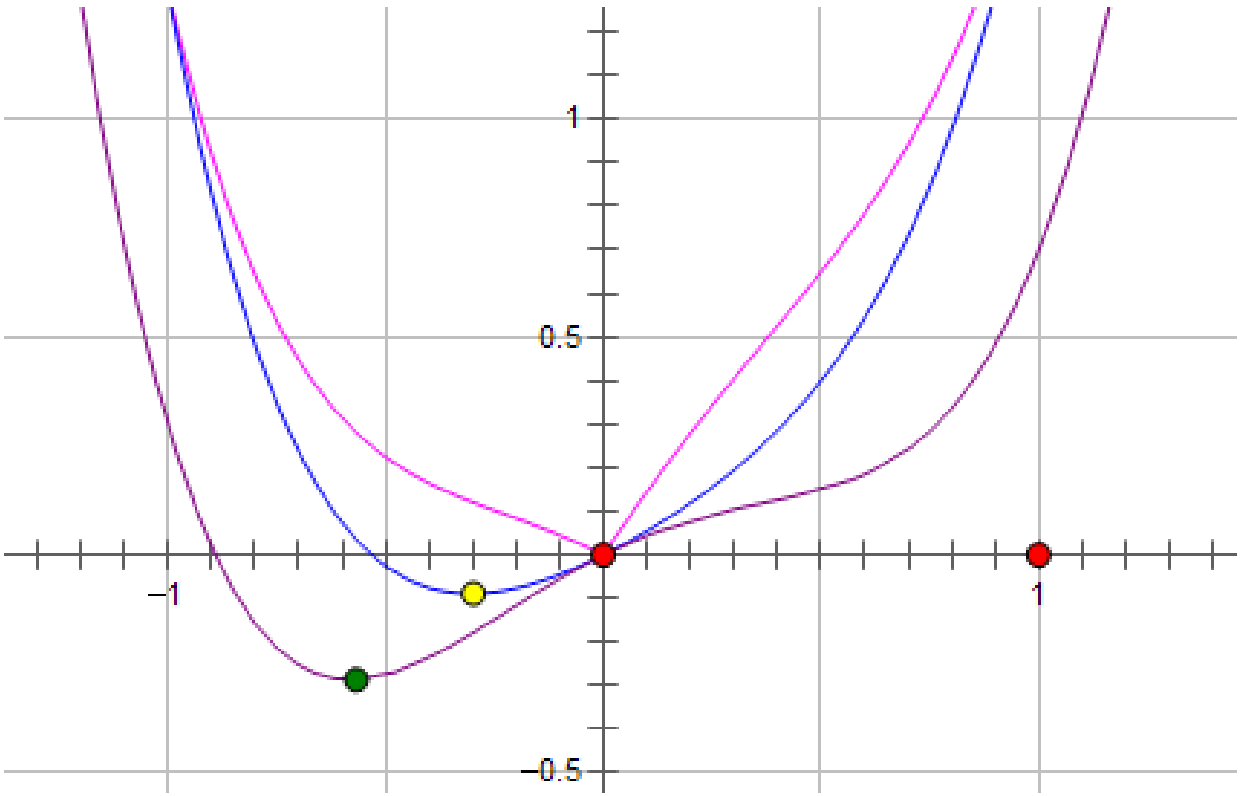


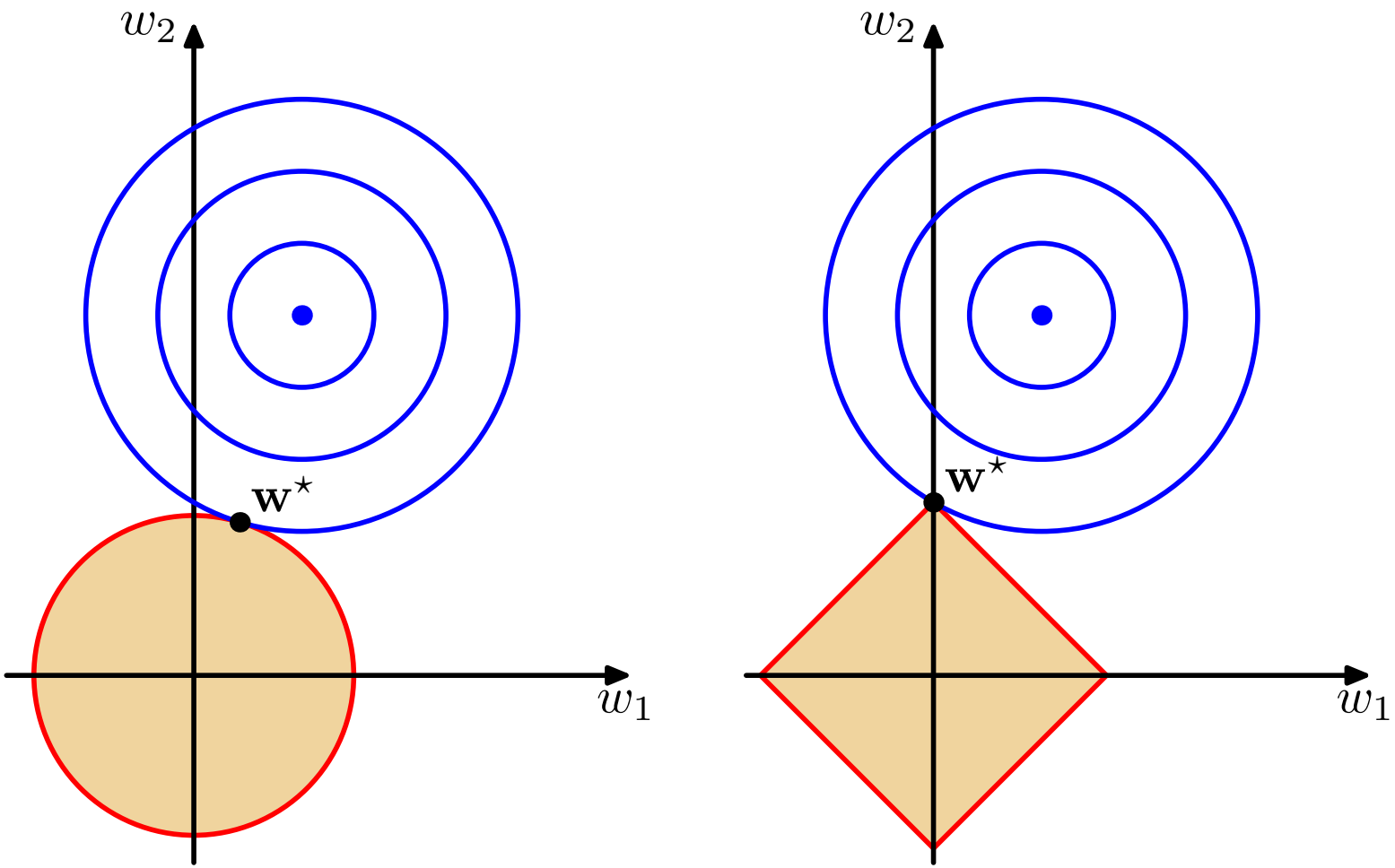
FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

Why Lasso Lead Sparsity?

Let's consider one dimensional case: $h(w) = f(w) + C|w|$, if $C > |f^{'}(0)|$ holds, then $h_l^{'}(0)h_r^{'}(0) = (f^{'}(0) + C)(f^{'}(0) - C) < 0$, and thereofore 0 is the optimal point.
$h(w) = f(w) + \frac{C}{2}w^2$, $h'(0) = f'(0) + Cw = f'(0)$, and thereofore 0 is the optimal point iff its gradient is 0.



Sparsity



Why Use Cross Entropy Instead of MSE

If we choose MSE as the objective, then: $loss = \sum_{i=1}^{m}(y_i - \sigma(w^T x_i))^2$, where $\sigma(w^T x_i) = \frac{1}{1+exp(-w^T x_i)}$, the gradient will be: $\frac{\partial loss}{\partial w} = \sum_{i=1}^{m}(-2(y_i - \sigma(w^T x_i))\sigma(w^T x_i)(1 - \sigma(w^T x_i))x_i)$, which is so-called vanishing gradient problem.

If we use the Cross Entropy as the cost: $\sum_{i=1}^{m} ln(1 + e^{\langle w, x_i \rangle}) - y_i \langle w, x_i \rangle$, the gradient will be: $\sum_{i=1}^{m}(\sigma(w^T x_i) - y_i)x_i$, it is free from gradient vanishing.

When to Use LR or SVM

Denote the dimension of feature is n, sample size is m:

- if $n > m$, then LR, since there are sufficient features.
- if $n < m$ and m is not that large, then SVM as features are in shortage.
- if $n < m$ and m is very large, then LR or Linear Kernel SVM, since sophisticated kernel (say *RBF*, etc) will be very computationally demanding.

Another Formulation of Logistic Regression

Recall the objective:

$$L(w) = \prod_{i=1}^{m} [p(x_i)]^{y_i} [1 - p(x_i)]^{1-y_i},$$

which is based on labels between 0 and 1. However, some instead use 1 and -1, where $P(+1 \mid x_i) = \sigma(w^\intercal x_i + b)$ and $P(-1 \mid x_i) = 1 - \sigma(w^\intercal x_i + b) = \sigma(-(w^\intercal x_i + b))$. In all, we have:

$$P(y_i \mid x_i) = \sigma(y_i(w^\intercal x_i + b))$$

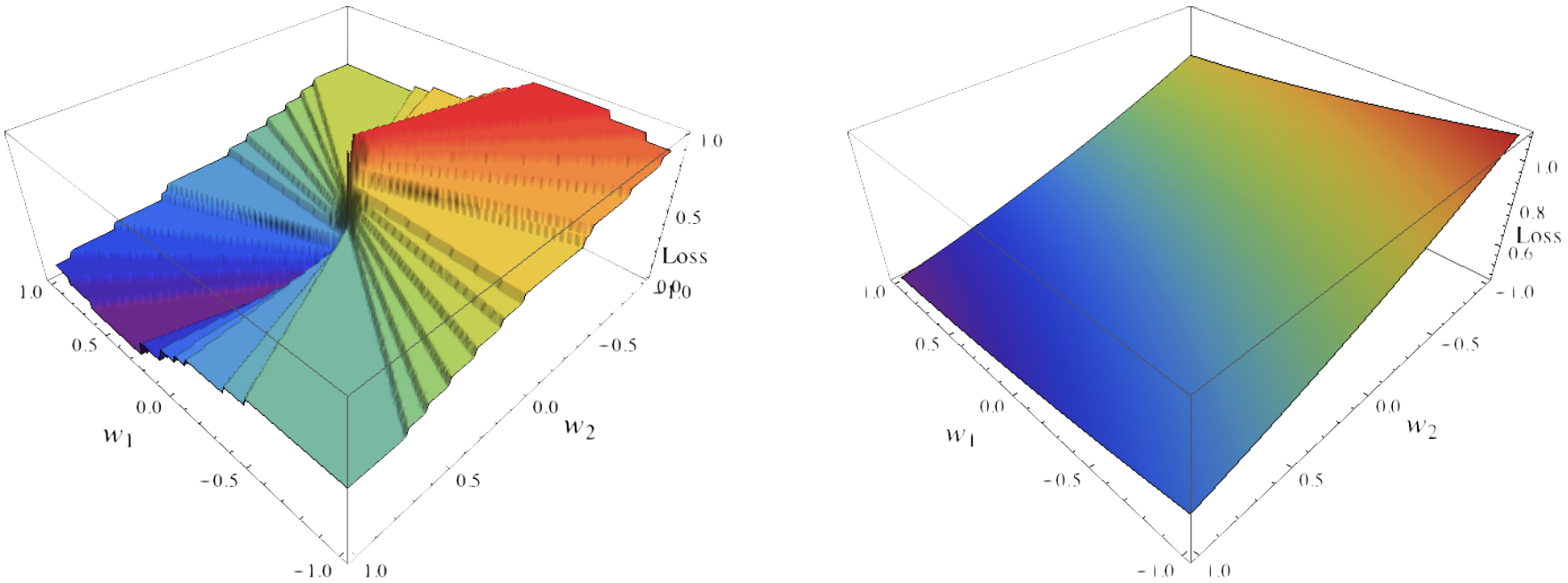
where $y_i \in +1, -1, \sigma(x) = \frac{1}{1+e^{-x}}$. Therefore, the objective is

$$-\sum_{i}^{m} \ln(\sigma(y_i(w^\intercal x_i + b))) = \sum_{i}^{m} \ln(1 + exp(-y_i(w^\intercal x_i + b))$$

and we can verify that it is convex.

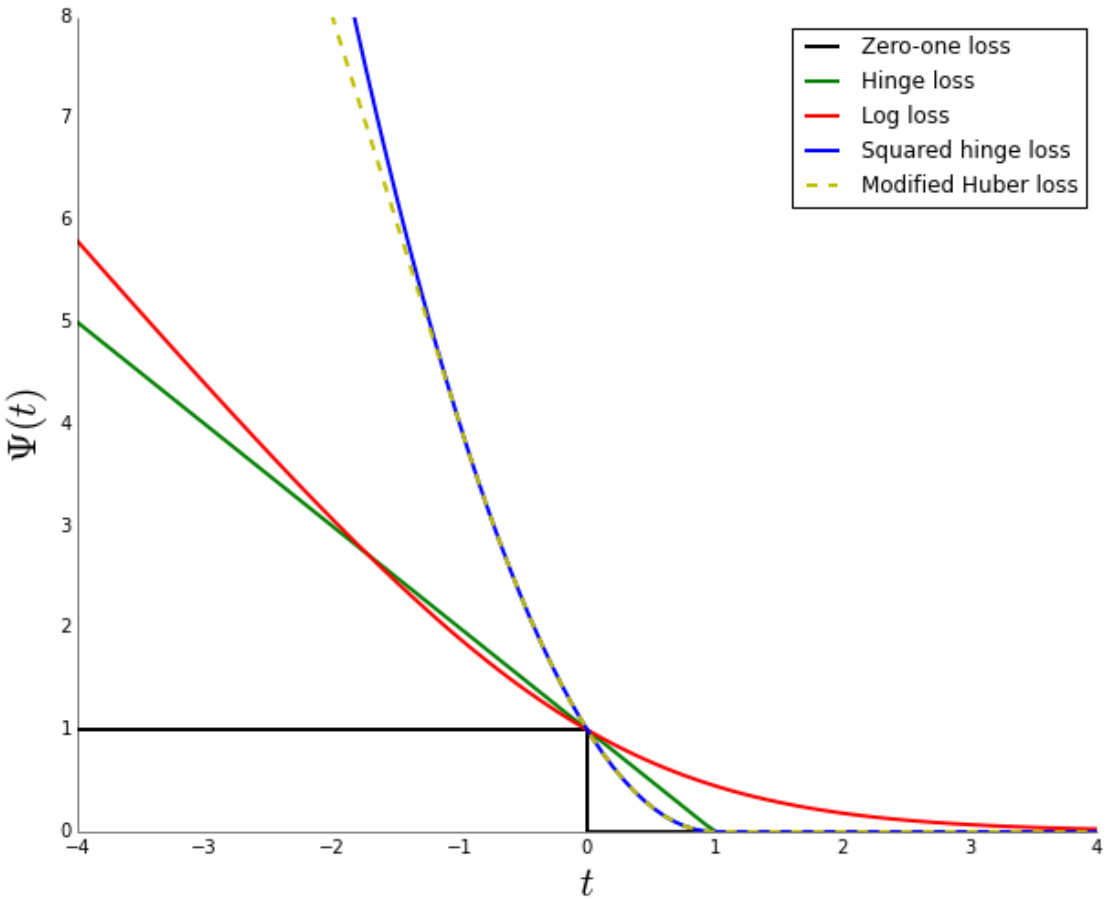
Surrogate Function for LR

The left figure is the original function, not easy to optimize, thus we use a surrogate funtion (right part), which is convex and continous.



Consistency/Calibration of Surrogate Function

The property of whether minimizing the surrogate function leads to simultanously minimizing the original function is often referred to as consistency or calibration. If it is convex then it is consistent iff its derivative at 0 is negative.



LR VS. SVM

$$\begin{aligned}
\text{SVM}: & \frac{1}{n}\sum_{i=1}^{n}(1-y_i[w_0+\mathrm{x}_i^T\mathrm{w}_1])^+ + \lambda\|\mathrm{w}\|^2 \\
\text{Logistic}: & \frac{1}{n}\sum_{i=1}^{n}\overbrace{-\ln\sigma(y_i[w_0+\mathrm{x}_i^T\mathrm{w}_1])}^{-\ln P(y_i|\mathrm{x},\mathrm{W})} + \lambda\|\mathrm{w}\|^2
\end{aligned} \tag{21}$$

which can be unified as:

$$\text{Both}: \frac{1}{n}\sum_{i=1}^{n} Loss(\overbrace{y_i[w_0+\mathrm{x}_i^T\mathrm{w}_1]}^{z}) + \lambda\|\mathrm{w}\|^2$$

SVM: $Loss(z) = (1-z)^+$, LR: $Loss(z) = \ln(1+\exp(-z))$.

LR VS. SVM

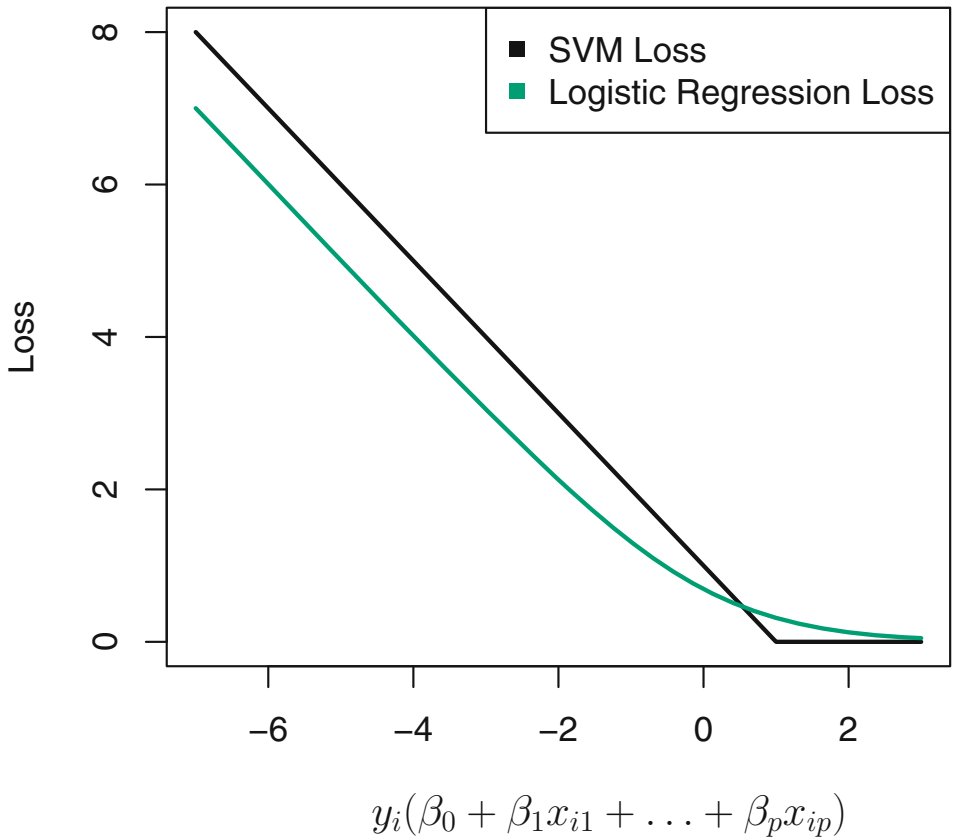


FIGURE 9.12. *The SVM and logistic regression loss functions are compared, as a function of* $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$. *When* $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$ *is greater than 1, then the SVM loss is zero, since this corresponds to an observation that is on the correct side of the margin. Overall, the two loss functions have quite similar behavior.*