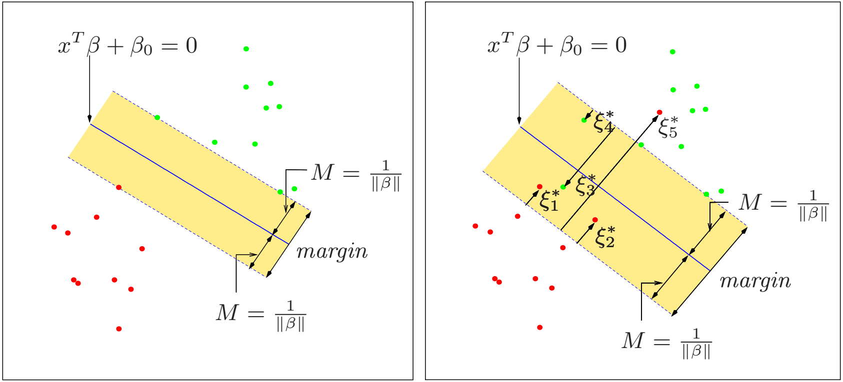
CPSC 8420 Advanced Machine Learning
Week 12: Logistic Regression

Dr. Kai Liu

November 10, 2020

Support Vector Machine

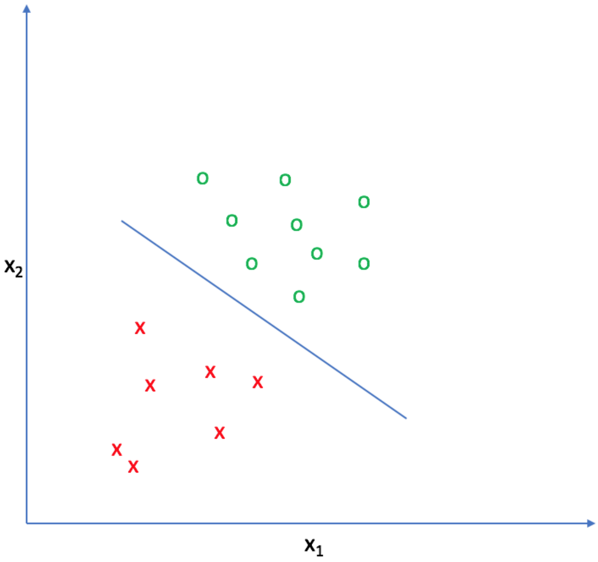


K-means v.s. NMF

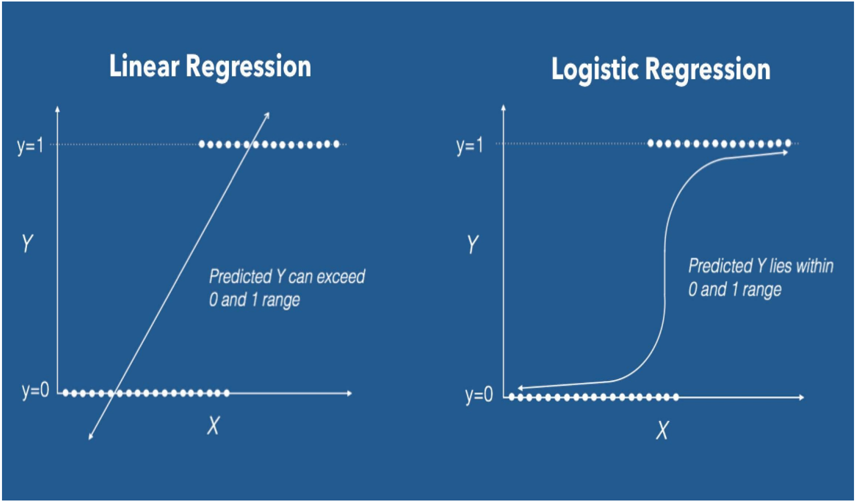
NMF	a	b	c	d
C1	0.9	0.15	0.8	0.25
C2	0.2	0.8	0.1	0.8

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

Decision Boundary (Perceptron)



Discrete Response



Surrogate Relaxation

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

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

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

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:

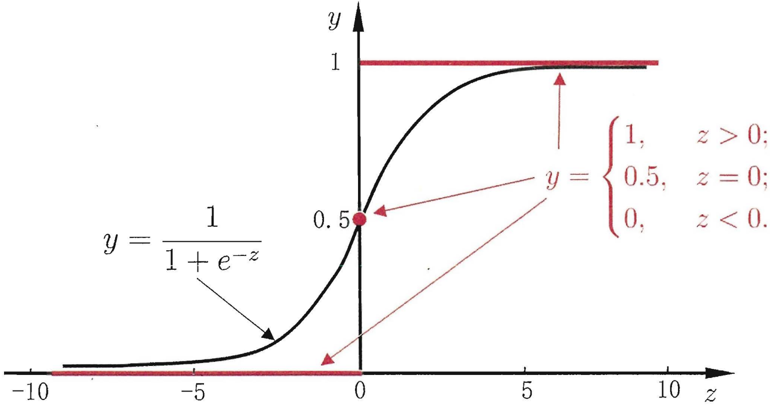
$$RatioCut(A_1, A_2, ...A_k) = \sum_{i=1}^{k} h_i^T L h_i = \sum_{i=1}^{k} (H^T L H)_{ii} = tr(H^T L H) \tag{3}$$

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:

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

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-odd
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:

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

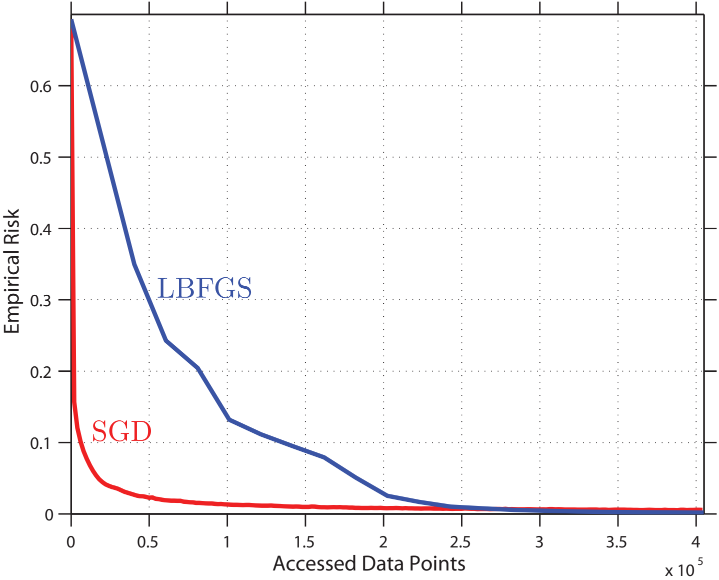
Optimization with Stochastic Gradient Descent (SGD)

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

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,

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

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

$$w^{k+1} = w^k - H_k^{-1} \cdot \nabla J \quad (11)$$

Let's reconsider Least Squares:

$$\min_{\beta} \|y - A\beta\|^2 \quad (12)$$

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 $p(x_i)[1 - p(x_i)]$.