$f(w) = \frac{1}{n}\sum_{i=1}^{n}\{-y_i w^T x_i + \log(1+\exp(w^T x_i))\} + \frac{\lambda}{2}\|w\|^2$, assume $\|X^i\| \le R$

a. $\nabla_w f(w) = \frac{1}{n}\sum_{i=1}^{n}(-y_i x_i + \frac{x_i \exp(w^T x_i)}{1+\exp(w^T x_i)}) + \lambda\|w\|$

$= \frac{1}{n}\sum_{i=1}^{n}(\sigma(w^T x_i) - y_i)x_i + \lambda\|w\|$, where $\sigma(w^T x_i) = \frac{\exp(w^T x_i)}{1+\exp(w^T x_i)}$

Steps: i initialize w_j values as random values ($j = 0, 1, \ldots, d$) where d is the number of features, $w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \end{bmatrix}$

ii compute $\nabla_w f(w)$ as above

iii compute $y_{k+1} = w_k - \eta \nabla_w f(w)$, $\eta > 0$ until it converges

iv if $\|y_{k+1}\| \le R$, $w_{k+1} = y_{k+1}$, if $\|y_{k+1}\| > R$, $w_{k+1} = \frac{R}{\|y_{k+1}\|} y_{k+1}$

b) Yes. for every $w_1 > w_2$, to prove $f(w_1) \ge f(w_2) + (w_1 - w_2)^T \nabla f(w_2) + \frac{\lambda}{2}\|w_1 - w_2\|^2$

we prove $\frac{1}{n}\sum_{i=1}^{n}\{-y_i w_1^T x_i + \log(1+\exp(w_1^T x_i))\} + \frac{\lambda}{2}\|w_1\|^2 \ge \frac{1}{n}\sum_{i=1}^{n}\{-y_i w_2^T x_i + \log(1+\exp(w_2^T x_i))\} + \frac{\lambda}{2}\|w_2\|^2$

$+ (w_1 - w_2)^T \cdot \{\frac{1}{n}\sum_{i=1}^{n}(\sigma(w_2^T x_i) - y_i)x_i + \lambda\|w_2\|\} + \frac{\lambda}{2}\|w_1 - w_2\|^2$

To prove $f(w)$ is strongly convex, we can prove $f(w) - \frac{\lambda}{2}\|w\|^2$ is convex

$\nabla^2(f(w) - \frac{\lambda}{2}\|w\|^2) = \frac{1}{n}\sum_{i=1}^{n}\{\frac{\exp(w^T x_i)(1+\exp(w^T x_i)) + \exp(w^T x_i)}{(1+\exp(w^T x_i))^2}\}x_i \ge 0$

since $\exp(w^T x_i) > 0$ and $x_i \ge 0$ $(i = 0, 1, \ldots, n)$

So the objective function $f(w)$ is strongly convex

We prove each part of objective function is β-smooth

(c) Yes. There exists $w_1 > w_2$ where $f(w_1) \le f(w_2) + (w_1 - w_2)^T \nabla f(w_2) + \frac{\beta}{2}\|w_1 - w_2\|^2$

if $\beta > \lambda$ then it is true since $f(x) = \log(1+\exp(x))$ is β-smooth since

$\log(\frac{1+\exp(x_2)}{1+\exp(x_1)}) + (x_1 - x_2)\frac{\exp(x_2)}{1+\exp(x_2)} + \frac{\beta}{2}\|x_1 - x_2\|^2 \ge 0$ if $\beta = 2(x_1 - x_2)^T$ because $\frac{\exp(x_2)}{1+\exp(x_2)} > 0$ and

$\log(\frac{1+\exp(x_2)}{1+\exp(x_1)})$ is growing less than quadratic function, and $-y^T w^T x$ is β_2-smooth since

$-y(w_1^T - w_2^T)x + (w_1 - w_2)^T(-yx) + \frac{\beta_2}{2}\|w_1 - w_2\|^2 \ge 0$ if $\beta_2 = 2yx$ and obviously $\frac{\lambda}{2}(\|w_2\|^2 - \|w_1\|^2) + (w_1 - w_2)^T \cdot \lambda\|w_2\| +$

$\frac{\beta}{2}\|w_1 - w_2\| \ge 0$ if $\beta > \lambda$ since $\|w_1 - w_2\|^2 \ge \|w_2\|^2 - \|w_1\|^2$ based on triangle inequality.

d) for β-smooth and α-strongly convex function.

strong-convexity: $f(w_T) - f(w^*) \le (w_T - w^*)^T \nabla f(w^*) + \frac{\beta}{2}\|w_T - w^*\|^2 = \frac{\beta}{2}\|w_T - w^*\|^2$ since $\nabla f(w^*) = 0$

$w_{T+1} = w_T - \eta_T \nabla f(w_T)$: $\|w_{T+1} - w^*\|^2 = \|w_T - w^*\|^2 + \eta_T^2\|\nabla f(w_T)\|^2 - 2\eta_T \nabla f(w_T)^T(w_T - w^*)$

$\le \|w_T - w^*\|^2 + \eta_T^2\|\nabla f(w_T)\|^2 - 2\eta_T(\frac{\alpha\beta}{\alpha+\beta}\|w_T - w^*\|^2 + \frac{1}{\alpha+\beta}\|\nabla f(x_T)\|^2)$

$= (1 - 2\eta_T\frac{\alpha\beta}{\alpha+\beta})\|w_T - w^*\|^2 + \eta_T(\eta_T - \frac{2}{\alpha+\beta})\|\nabla f(x_T)\|^2$

suppose $\eta_T \le \frac{2}{\alpha+\beta}$, $\|w_{T+1} - w^*\|^2 \le (1 - 2\eta_T\frac{\alpha\beta}{\alpha+\beta})\|w_T - w^*\|^2$

$\le \|w_0 - w^*\|^2 \prod_{n=1}^{k}(1 - \frac{2\eta_n\alpha\beta}{\alpha+\beta})$ and $1 - x \le e^{-x}$

so, $f(w_T) - f(w^*) \le \frac{\beta}{2}\|w_T - w^*\|^2 \le \frac{\beta}{2}\prod_{n=1}^{k}(1 - \frac{2\eta_n\alpha\beta}{\alpha+\beta})\|w_0 - w^*\|^2 \le \frac{\beta}{2}\|w_0 - w^*\|^2 e^{-\frac{2\alpha\beta}{\alpha+\beta}\sum_{n=1}^{k}\eta_n}$

2. initialize Φ^0 with mean vector of zero vector and covariance matrix of I

(a) E-step: assign class labels for every data point by using Bayes' rule to compute the likelihood that it belongs to certain class G_i, given the data point and the old set of parameters Φ^L and set the label G_i with the largest likelihood $P(G_i|X^0, \Phi^L) = h_i^0$ to that data point.

M-step: recompute the prior probability $P(G_i)$, mean vector and the covariance matrix with the posterior h_i^0, the data point X^0 and the Total number of class N, update the new set of parameters to be Φ^{L+1}, maximum likelihood is used to get new parameters

keep doing E-step & M-step until h_i^0 converges

(b) $\lambda_i^{L+1} = \frac{\sum_{0=1}^{N} h_i^0}{N}$, $\mu_i^{L+1} = \frac{\sum_{0=1}^{N} h_i^0 X^0}{\sum_{0=1}^{N} h_i^0}$

$\Sigma_i^{L+1} = \frac{\sum_{0=1}^{N} h_i^0 (X^0 - \mu_i^{L+1})(X^0 - \mu_i^{L+1})^T}{\sum_{0=1}^{N} h_i^0}$ where L is the step number

(c) $h_i^0 \equiv P(\lambda_i | X^0, \mu_i, \Sigma_i) = \frac{P(X^0 | \pi_i, \mu_i^L, \Sigma_i^L) P(\lambda_i)}{\sum_{j=1}^{K} P(X^0 | \pi_j, \mu_j^L, \Sigma_j^L) P(\lambda_j)}$, where L is step

and $P(X^0 | \lambda_i, \mu_i^L, \Sigma_i^L) = \lambda_i \cdot \frac{1}{(2\pi)^{n/2} |\Sigma_i^L|^{1/2}} \exp\left(-\frac{1}{2}(X^0 - \mu_i^L)^T \Sigma^{-1} (X^0 - \mu_i^L)\right)$

Q3

Summary

I use the gradient descent method with iteration number of 850 and step size of 0.00001 based on my experiment and some suggestions on the textbook.

MyLogisticReg2 with Boston50

K=0	K=1	K=2	K=3	K=4	Mean	Std
0.207920792079	0.188118811881	0.128712871287	0.247524752475	0.147058823529	0.18386721025	0.0425342027945

MyLogisticReg2 with Boston75

K=0	K=1	K=2	K=3	K=4	Mean	Std
0.207920792079	0.178217821782	0.227722772277	0.138613861386	0.107843137255	0.172063676956	0.0439652339392

LogisticRegression with Boston50

K=0	K=1	K=2	K=3	K=4	Mean	Std
0.128712871287	0.108910891089	0.0891089108911	0.277227722772	0.117647058824	0.144321490973	0.0677075481368

LogisticRegression with Boston75

K=0	K=1	K=2	K=3	K=4	Mean	Std
0.0891089108911	0.128712871287	0.138613861386	0.108910891089	0.0490196078431	0.102873228499	0.0318471419286

Q4

Summary

The feature matrix gets preprocessed by a standardized method in a way that for each feature column vector, every element gets subtracted by the mean of the column and divided by the standard deviation of the column so that every feature vector has mean of zero and standard deviation of one.

The gradient descent method with iteration number of 300 and step size of 0.00001 based on my experiment and some suggestions on the textbook. And I consider the lambda value of one to penalize big weight in each step.

MyLogisticRegGen with Digits

K=0	K=1	K=2	K=3	K=4	Mean	Std
0.075208913649	0.122562674095	0.0611111111111	0.050139275766	0.0388888888889	0.0695821727019	0.0290876487138

LogisticRegression with Digits

K=0	K=1	K=2	K=3	K=4	Mean	Std
0.0640668523677	0.116991643454	0.0611111111111	0.0389972144847	0.0166666666667	0.0595666976168	0.0334178913221