CSCI 5521: Fall'19
Introduction To Machine Learning
Homework 3

ARVIND RENGANATHAN renga016@umn.edu

3.

Error rates for MyLogisticReg2 with Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	S.D
0.16	0.26	0.23	0.23	0.22	0.21	0.03

Error rates for MyLogisticReg2 with Boston75						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	S.D
0.29	0.19	0.23	0.20	0.22	0.22	0.03

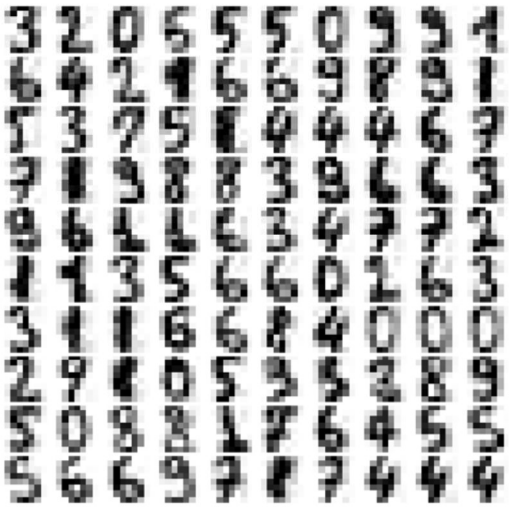
Error rates for Logistic regression with Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	S.D
0.14	0.07	0.12	0.06	0.10	0.10	0.03

Error rates for Logistic regression with Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	S.D
0.07	0.10	0.10	0.08	0.08	0.09	0.01

4.

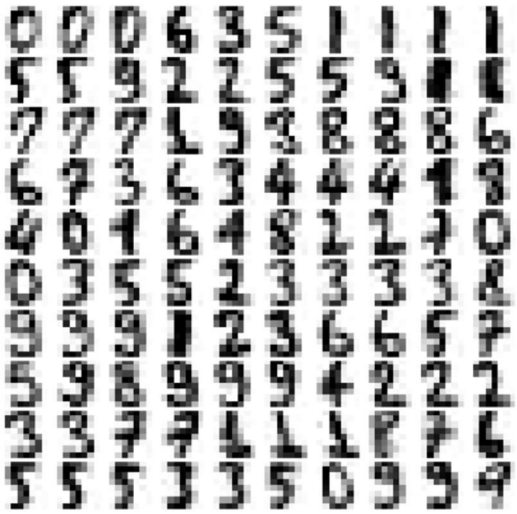
We found that for α = 90, we found d=21 to be sufficient and for α = 99 we found 41 to be sufficient which was expected. For α = 90

```
In [11]: digits_new90 = np.dot(data_new90, proj_matrix90.T)
         plot_digits(digits_new90)
```



For α = 99

```
In [16]: digits_new99 = np.dot(data_new99, proj_matrix99.T)
         plot_digits(digits_new99)
```



2. (i) Let EM Algorithm for a Gaussian mixture model with Σ_K being diagonal, with diagonal elements of σ_{Kj}^2

The Algorithm Alternates between 'E' & 'M' steps

E step: Using current values of the parameters, compute the responsibilities of the components for data items, by applying Bayes' Rule

$r_{iK} = P(\text{data item } i \text{ came from component } K | x_i)$

$$= \frac{\pi_K N(x_i | \mu_K, \Sigma_K)}{\sum_{K'} \pi_{K'} N(x_i | \mu_K', \Sigma_{K'})}$$

M Step: Using the current responsibilities, re-estimate the parameters, using weighted averages with weights given by the responsibilities

$$\pi_K = \frac{1}{N} \sum_i r_{ik}, \quad \mu_K = \sum r_{ik} x_i \Big/ \sum_i r_{ik} \quad \sigma_K^2 = \frac{\sum r_{ik} (x_i - \mu_K)^2}{\sum_i r_{ik}}$$

We start with initial guess at parameter values (or random), or perhaps with some initial guess at responsibilities (in which case we start with M). We alternate until little change.

b) M step.

$$\pi_k = \frac{1}{N} \sum_i p(G_k | x_i) \frac{N_k}{N}$$

$$\mu_k = \frac{1}{N_k} \sum_i p(G_k | x_i)$$

$$\pi_k = \frac{1}{N} \sum_i p(G_k | x_i)$$

$$\mu_k = \sum_i p(G_k | x_i) \cdot x_i \Big/ \sum_i p(G_k | x_i)$$

$$\Sigma_k = \sum_i p(G_k | x_i) \cdot (x_i - \mu_k)^2 \Big/ \sum_i p(G_k | x_i)$$

c) $$p(G_k | x_i) = \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} N(x_i | \mu_{k'}, \Sigma_{k'})}$$

1. $f(w) = \frac{1}{n}\sum_{i=1}^{n}\left\{-y_i w^T x_i + \log\left(1+\exp(w^T x_i)\right)\right\} + \frac{\lambda}{2}\|w\|^2$

$$f' = \frac{\partial f}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}\left(-y_i x_i + \frac{e^{w^T x_i}}{1+e^{w^T x_i}} x_i\right) + \lambda w$$

$$f'' = \frac{\partial^2 f}{\partial w^2} = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{(1+e^{w^T x_i})(e^{w^T x_i})x_i - e^{w^T x_i} e^{w^T x_i})x_i}{(1+e^{w^T x_i})^2}\right\} + \lambda$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{e^{w^T x_i}}{(1+e^{w^T x_i})^2} x_i^2\right) + \lambda$$

(i) Projected gradient descent →

$$w_{t+1} = w_t - \eta \frac{\partial f(w_t)}{\partial w_t}$$

$t \to t^{th}$ epoch

$t+1 \to (t+1)^{th}$ epoch

$$\therefore w_{t+1} = \begin{cases} w_{t+1} & \text{if } \|w_{t+1}\|_2 \le c \\ \frac{c}{\|w_t\|_2} w_{t+1} & \text{if } \|w_{t+1}\| > c \end{cases}$$

the above two conditions are for keeping $\|w\| \le c$ in each epoc.

Gradient descent: iterate until change $< \varepsilon$

b) To show that LR objective is convex, we consider the partial derivatives.

We have to minimize $-\log P(\vec{y}|X,\vec{w}) = \sum \log(1+\exp(-y_i \vec{w}^T x_i))$

Define $g(z) = \frac{1}{1+e^{-z}}$ $\quad 1-g(z) = \frac{e^{-z}}{1+e^{-z}}$

$$\frac{\partial g(z)}{\partial z} = -g(z)(1-g(z))$$

$$\frac{\partial \log P(\vec{y}|X,\vec{w})}{\partial w_j} = -\sum_{i=1}^{n} y_i x_{ij} (1-g(y_i \vec{w}^T \vec{x_i}))$$

$$\frac{\partial^2 \log P(\vec{y}|X,\vec{w})}{\partial w_j \partial w_k} = \sum_{i=1}^{n} y_i^2 x_{ij} x_{ik} g(y_i \vec{w}^T \vec{x_i})(1-g(y_i \vec{w}^T \vec{x_i}))$$

To show objective function is convex, we first show hessian is positive semi definite.

A Matrix M is PSD iff $\vec{a}^T M \vec{a} \geq 0$ for all vectors $\vec{a}$. Let ∇^2 be hessian for our objective.

Defin $P_i = g(y_i \vec{w}^T \vec{x_i})(1-g(y_i \vec{w}^T \vec{x_i}))$ & $P_{ij} = x_{ij}\sqrt{P_i}$

Then

$$\vec{a}^T \nabla^2 \vec{a} = \sum_{i=1}^{n} \sum_{j=1}^{d} \sum_{k=1}^{d} a_j a_k x_{ij} x_{ik} p_i,$$

$$= \sum_{i=1}^{n} \vec{a}^T \vec{p_i} \vec{p_i}^T \vec{a} \geq 0$$

as $\vec{a}^T \vec{p_i} \vec{p_i}^T \vec{a} = (\vec{a}^T \vec{p_i})^2 \geq 0$

hence, the hessian is PSD. so objective function is convex.

On adding L2 regularizer $\frac{\lambda}{2}||w||^2$ or $\frac{\lambda}{2} \vec{w}^T w$ to the objective, the hessian is positive definite & hence objective function is strictly convex. $\lambda > 0$. It is strictly convex & differentiable

d) The bound is defined as

$$f(w_T) - f(w^*) \leq \frac{\beta}{2} \exp\left(\frac{-\lambda T}{\frac{\beta}{\alpha} + 1}\right) ||w_1 - w^*||^2$$

Here β as is in 1. c) it is β smooth upper bound

α ~~is as in 1 b) it is step~~ is α strongly convex lower bound.

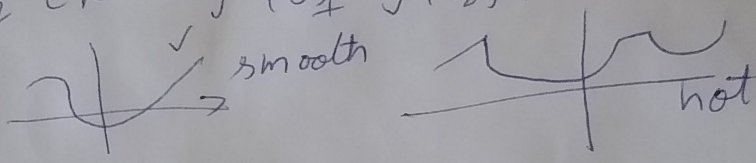
T is step number T of the projected gradient descent algorithm

w_1 is the w initialized at the start of projected gradient descent algorithm

w^* is the optimum value for w

c) A function $g: R^p \to R$ is L smooth if and only if it is differentiable & is L-Lipschitz continuous

$\forall \theta_1, \theta_2 \in R^p \quad \|g'(\theta_1) - g'(\theta_2)\| \leq L\|\theta_1 - \theta_2\|$



$$f(w) \leq f(w') + \langle \nabla f(w'), w - w' \rangle + \frac{\beta}{2}\|w - w'\|^2,$$

$\forall w, w' \in W$

All convex quadratic & logistic is smooth.

for smoothness $y_i = 0$

it is enough to check that

$g: R \to R \quad g(t) = \log(1 + \exp(t))$

has lipschitz gradient & it does

because its second derivative is bounded

$\nabla f(\beta) = -g^+(h(\beta)) X_i^T, \quad h(\beta) = X_i \cdot \beta$

& it is twice differentiable

$$\Rightarrow 0 \leq f''(\theta) \leq \beta I,$$