CSCI 5521 Denis Rybkin
Section 2 rybki001@umn.edu

Homework 3
Due Friday, November 20

Problem 1. Let $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ be a given training set where $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{0, 1\}$. We consider the following regularized logistic regression objective function:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\{-y_i\mathbf{w}^T\mathbf{x}_i + \log(1 + \exp(\mathbf{w}^T\mathbf{x}_i))\} + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

where $\lambda > 0$ is a constant. Let $\mathbf{w}^*$ be the global minimized of the objective, and let $||\mathbf{w}^*||_2 \leq c$, for some known constant $c > 0$.

(a) (10 points) Clearly show and explain the steps of the projected gradient descent algorithm for optimizing the regularized logistic regression objective function. The steps should include an exact expression for the gradient.

We have the regularized logistic regression objective function:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\{-y_i\mathbf{w}^T\mathbf{x}_i + \log(1 + \exp(\mathbf{w}^T\mathbf{x}_i))\} + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

which has the hypothesis function:

$$h(\mathbf{w}^T x_i) = \frac{\exp(\mathbf{w}^T x_i)}{1 + \exp(\mathbf{w}^T x_i)}$$

In order to derive the gradient of $f(\mathbf{w})$, let's first find the derivative:

$$\frac{\partial}{\partial \mathbf{w}} \log(1 + \exp(\mathbf{w}^T x_i)$$

We need to use the Chain Rule to solve it. Let's show each step:

$$\frac{\partial}{\partial \mathbf{w}} \log(1 + \exp(\mathbf{w}^T x_i)) = \frac{1}{1 + \exp(\mathbf{w}^T x_i)} \ \frac{\partial}{\partial \mathbf{w}}(1 + \exp(\mathbf{w}^T x_i)),$$

$$\frac{\partial}{\partial \mathbf{w}}(\exp(1 + \mathbf{w}^T x_i)) = \frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T x_i) \ \exp(\mathbf{w}^T x_i) = x_i \exp(\mathbf{w}^T x_i)$$

Then, using the results, we have:

$$\frac{\partial}{\partial \mathbf{w}} \log(1 + \exp(\mathbf{w}^T x_i)) = \frac{x_i \exp(\mathbf{w}^T x_i)}{1 + \exp(\mathbf{w}^T x_i)}.$$

Now, we can derive the gradient of $f(\mathbf{w})$ using the results from above:

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

It can be further simplified into the following form:

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

Using the hypothesis function, we can also rewrite it as:

$$\nabla_f = \sum_{i=1}^{n} \{ x_i (h(\mathbf{w}^T x_i) - y_i) \} + \lambda \mathbf{w}$$

Then the update rule for projected gradient descent is

$$\begin{aligned} \mathbf{w}_t &= 0 \\ &\dots \text{ (repeat)} \\ \mathbf{w}'_{t+1} &= \mathbf{w}_t - \eta [x_i (h(\mathbf{w}^T x_i) - y_i) + \lambda \mathbf{w}] \\ \mathbf{w}_{t+1} &= \Pi_X (\mathbf{w}'_{t+1}) \end{aligned}$$

(Iterate Until Convergence)

where $\Pi_X(\mathbf{x}) = \underset{y \in X}{\operatorname{argmin}} ||x - y||$

$\Pi_X(\mathbf{w}'_{t+1})$ projects the update $\mathbf{w}'_{t+1}$ back to the point in the constrained region that is nearest to it in case if it ends up outside the constrained region.

Alternatively, we can express it as:

$$\mathbf{w}_{t+1} = \begin{cases} \mathbf{w}'_{t+1}, & \text{if } ||\mathbf{w}'_{t+1}|| \leq R \\ \frac{R}{||\mathbf{w}'_{t+1}||} \mathbf{w}'_{t+1}, & \text{if } ||\mathbf{w}'_{t+1}|| > R \end{cases}$$

where R is a radius for a ball that constrains updates to the constrained region (that form is also given in the lecture slides).

(b) (5 points) Is the objective function strongly convex? Clearly explain your answer by stating and using the definition of strong convexity.

The objective function is strongly convex.

We can prove it by splitting the function into multiple parts and showing that each of them is either convex or strongly convex. After that, we will use the properties of convexity (and additive properties for convex functions) to show that our function is strongly convex. The proof relies on the assumption that properties of convexity are additive (which is indeed the case).

We have:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\{-y_i\mathbf{w}^T\mathbf{x}_i + \log(1+\exp(\mathbf{w}^T\mathbf{x}_i))\} + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

We can conveniently split it into 3 parts:

$$\begin{aligned} g_1(\mathbf{w}) &= -y_i\mathbf{w}^T x_i \\ g_2(\mathbf{w}) &= \log(1+\exp(\mathbf{w}^T x_i) \\ g_3(\mathbf{w}) &= \frac{\lambda}{2}||\mathbf{w}||_2^2 \end{aligned}$$

Then, we will have

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} g_1(\mathbf{w}) + g_2(\mathbf{w}) + g_3(\mathbf{w})$$

Let's prove that g_1 is convex:

$$\begin{aligned} -y_i\mathbf{w}_1^T x_i &\geq -y_i\mathbf{w}_2^T x_i + (\mathbf{w}_1 - \mathbf{w}_2)(-y_i x_i) \\ (\frac{1}{y_i x_i})(-y_i\mathbf{w}_1^T x_i) &\geq (\frac{1}{y_i x_i})(-y_i\mathbf{w}_2^T x_i + (\mathbf{w}_1 - \mathbf{w}_2)(-y_i x_i)) \\ -\mathbf{w}_1 &\geq -\mathbf{w}_2 - \mathbf{w}_1 + \mathbf{w}_2 \\ -\mathbf{w}_1 &\geq -\mathbf{w}_1 \end{aligned}$$

To show that g_2 is convex, we need to make some simplifications. Assume that $\exp(\mathbf{w}^T x_i)$ dominates in the log, and rewrite g_2 as:

$$g_2(x) = \log(\exp(\mathbf{w}^T x_i)) = \mathbf{w}^T x_i$$

Then we can easily prove convexity:

$$\begin{aligned}
\mathbf{w}_1^T x_i &\geq \mathbf{w}_2^T x_i + (\mathbf{w}_1 - \mathbf{w}_2)x_i \\
(\frac{1}{x_i})(\mathbf{w}_1^T x_i) &\geq (\frac{1}{x_i})(\mathbf{w}_2^T x_i + (\mathbf{w}_1 - \mathbf{w}_2)x_i) \\
\mathbf{w}_1 &\geq \mathbf{w}_2 + \mathbf{w}_1 - \mathbf{w}_2 \\
\mathbf{w}_1 &\geq \mathbf{w}_1
\end{aligned}$$

Let's prove that g_3 is strongly convex:

$$\begin{aligned}
\mathbf{w}_1^2 &\geq \mathbf{w}_2^2 + (\mathbf{w}_1 - \mathbf{w}_2)2\mathbf{w}_2 + \frac{\alpha}{2}(\mathbf{w}_1 - \mathbf{w}_2)^2 \\
\mathbf{w}_1^2 &\geq \mathbf{w}_2^2 + (\mathbf{w}_1 - \mathbf{w}_2)2\mathbf{w}_2 + \frac{\alpha}{2}(\mathbf{w}_1 - \mathbf{w}_2)^2
\end{aligned}$$

Let's assume that $\alpha = 2$:

$$\begin{aligned}
&\mathbf{w}_1^2 \geq \mathbf{w}_2^2 + (\mathbf{w}_1 - \mathbf{w}_2)2\mathbf{w}_2 + (\mathbf{w}_1 - \mathbf{w}_2)^2 \\
&\mathbf{w}_1^2 \geq \mathbf{w}_2^2 + 2\mathbf{w}_1\mathbf{w}_2 - 2\mathbf{w}_2^2 + \mathbf{w}_1^2 - 2\mathbf{w}_1\mathbf{w}_2 + \mathbf{w}_2^2 \\
&\mathbf{w}_1^2 \geq \mathbf{w}_1^2, \text{ if } \alpha \leq 2
\end{aligned}$$

Hence, if $\alpha \leq 2$ then g_3 is strongly convex.

From before, we know that

$$\begin{aligned}
g_1(x) &\geq -y_i\mathbf{w}^T x_i + (x - y)(-y_i x_i) \\
g_2(x) &\geq \log(1 + \exp(\mathbf{w}^T x_i) + (x - y)\frac{x_i \exp(\mathbf{w}^T x)}{1 + \exp(\mathbf{w}^T x_i)}
\end{aligned}$$

Then, we have:

$$g_1(x) + g_2(x) \geq -y_i\mathbf{w}^T x_i + \log(1 + \exp(\mathbf{w}^T x_i) + \left((x - y)(-y_i x_i) + (x - y)\frac{x_i \exp(\mathbf{w}^T x)}{1 + \exp(\mathbf{w}^T x_i)}\right)$$

$$g_1(x) + g_2(x) \geq -y_i\mathbf{w}^T x_i + \log(1 + \exp(\mathbf{w}^T x_i) + (x - y)\left((-y_i x_i) + \frac{x_i \exp(\mathbf{w}^T x)}{1 + \exp(\mathbf{w}^T x_i)}\right)$$

And hence

$$\begin{aligned}
f(y) &= -y_i\mathbf{w}^T x_i + \log(1 + \exp(\mathbf{w}^T x_i) \\
\nabla f(y) &= (-y_i x_i) + \frac{x_i \exp(\mathbf{w}^T x)}{1 + \exp(\mathbf{w}^T x_i)}
\end{aligned}$$

Therefore, we have:

$$f(x) \geq f(y) + (x - y)\nabla f(y) + \frac{\lambda}{2}||\mathbf{w}||^2$$

if $\lambda = \alpha \leq 2$

Since our function can be written in the form for a strongly convex function and we proved convexity and strong-convexity of its components, we can conclude that the objective function is strongly convex.

(c) (5 points) Is the objective function smooth? Clearly explain your answer by stating and using the definition of smoothness.

The objective function is smooth.

Similarly to part (b), we will split the function into three parts and examine each of them, but this time, all components must be smooth. Then we will use additive properties to conclude that the objective function is smooth as well.

We have:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\{-y_i\mathbf{w}^T\mathbf{x}_i + \log(1 + \exp(\mathbf{w}^T\mathbf{x}_i))\} + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

We can conveniently split it into 3 parts:

$$\begin{aligned} g_1(\mathbf{w}) &= -y_i\mathbf{w}^T x_i \\ g_2(\mathbf{w}) &= \log(1 + \exp(\mathbf{w}^T x_i) \\ g_3(\mathbf{w}) &= \frac{\lambda}{2}||\mathbf{w}||_2^2 \end{aligned}$$

Then, we will have

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} g_1(\mathbf{w}) + g_2(\mathbf{w}) + g_3(\mathbf{w})$$

We will be using the fact that a function is smooth if its derivative is continuous for our domain.

First, let's do it for g_1:

$$\nabla g_1 = -y_i x_i$$

Notice that ∇g is a constant function and therefore it is continuous for all $\mathbf{w}$.

Hence, g_1 is smooth.

For g_2, we have:

$$\nabla g_2 = \frac{\exp(\mathbf{w}^T x_i) x_i}{1 + \exp(\mathbf{w}^T x_i)}$$

It is clear that for all $\mathbf{w}^T x$

$$\lim_{\mathbf{w}^T x \to -\infty} \nabla g_2 = 0 \quad \text{and} \quad \lim_{\mathbf{w}^T x \to \infty} \nabla g_2 = 1$$

Hence g_2 is smooth.

Now, for g_3 we have:

$$\nabla g_3 = \lambda ||\mathbf{w}||$$

It's clear that ∇g_3 is defined for all $\mathbf{w}$. Therefore, g_3 is smooth.

Since g_1, g_2, and g_3 are all smooth, we can conclude that the objective function is also smooth.

(d) (5 points) Let $\mathbf{w}_T$ be the iterate after T steps of the projected gradient descent algorithm with a suitably chosen step size depending on the properties of the function. What is a bound on the difference $f(\mathbf{w}_T) - f(\mathbf{w}^*)$? Clearly explain all quantities in the bound.

Since our function is strongly convex and smooth (we showed it in parts (b) and (c)), we have the rate of convergence given by the following equation:

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{\beta}{2} \exp(-\frac{4T}{\frac{\beta}{\alpha} + 1}) ||\mathbf{x}_0 - \mathbf{x}^*||^2$$

The rate at which our estimate after T iterations, $\mathbf{x}_T$, approaches our optimum, $\mathbf{x}^*$ with a fixed step size, is exponential. It is given roughly by $\exp(-CT)$.

Problem 2. Let $\mathcal{X} = \{x_1, ..., x_n\}, x_i \in \mathbb{R}^d$ be a set of n samples drawn i.i.d from a mixture of k multivariate Gaussian distributions in $\mathbb{R}^d$. For component $G_h, h = 1, ..., k$ let π_h, μ_h, Σ_h respectively denote the prior probability, mean, and covariance matrix of G_h. We will focus on the expectation maximization (EM) algorithm for learning the mixture model, in particular for estimating the parameters $\{(\pi_h, \mu_h, \Sigma_h), h = 1, ..., k\}$ as well as the posterior probabilities $p(G_h|\mathbf{x}_i)$.

(a) (10 points) In your own words, describe the EM algorithm for mixture of Gaussians, highlighting the two key steps (E- and M-), illustrating the methods used in the steps on a high level, and what information they need.

Expectation-Maximization (EM) algorithm for mixture of Gaussians is an unsupervised algorithm where we have the training data given as $X = \{\mathbf{x}^t\}_t$. However, we don't have the labels $\mathbf{r}^t$. The goal of EM is to estimate the labels representing the components (which is similar to classes in the supervised case).

Let's describe the E-step and the M-step of the EM algorithm for the multivariate Gaussian mixture model.

During the E-step, our goal is to estimate the latent labels $\mathbf{z}_h^t$ given current estimates of the component prior, mean, and covariance.

During the M-step we update the component prior, mean, and covariance given the labels estimated in the E-step. We define $\mathbf{z}^t$ to be a vector of indicator variables where we have $z_h^t = 1$ if $\mathbf{x}^t$ belongs to the cluster G_h and 0 otherwise.

Before we can begin EM, we would need the initial estimates for the component parameters. We can get them by running the k-means clustering algorithm in order to give us initial estimates of the component mean, covariance, and the prior. Once we have our initial component parameters and

our labels, we can do the E-step.

In the E-step, the need to estimate the labels from the estimated component parameters Φ.

Let's define:

$$\mathcal{Q}(\Phi|\Phi^l) = \sum_t \sum_h E(z_h^t|X, \Phi^l)(\log \pi_h + \log p_h(\mathbf{x}^t|\Phi^l))$$

where:

$$E(z_h^t|X, \Phi^l) = P(G_h|\mathbf{x}^t, \Phi^l)$$

This means that the expected value of z_h^t is the posterior probability $P(G_h|\mathbf{x}^t, \Phi^l)$ where $\mathbf{x}^t$ is generated by component G_h.

Since $P(G_h|\mathbf{x}^t, \Phi^l)$ is a probability, its value is between 0 and 1 and hence, it is a soft label. In the previous equations, Φ represents the estimated component prior, mean, and covariance (because we are describing EM for Gaussian Mixtures).

In the M-step, we maximize $\mathcal{Q}$ to get the next set of parameters Φ^l, the component prior, mean, and covariance:

$$\Phi^l = arg \max_{\Phi} \mathcal{Q}(\Phi|\Phi^l)$$

By doing so, we estimate the component prior:

$$\pi_h = \frac{\sum_t P(G_h|\mathbf{x}^t, \Phi^l)}{N}$$

We also describe the Gaussian component parameters estimated by the M-step above. Once we estimate the new parameters Φ^l, we begin again with the E-step.

Note that EM takes an infinite number of iterations to converge, so we stop when the parameters don't change much or when we've reached some maximum number of iterations. Also, if we knew where component $\mathbf{x}_t$ came from, we wouldn't need the E-step and we would only run M-step to get our

classification.

(b) Assuming the posterior probabilities $p(G_h|x_i)$ are known, show the estimates of the component prior, mean, and covariance $\pi_h, \mu_h, \Sigma_h, h = 1, \ldots, k$ given by the M-step (you do not need to show the derivation).

The component prior is dependent on the posterior probability $P(G_h|\mathbf{x}^t, \Phi^l)$ which is estimated from the E-step:

$$\pi_h = \frac{\sum_t P(G_h|\mathbf{x}^t, \Phi^l)}{N}$$

We know that $P(G_h|\mathbf{x}^t, \Phi^l)$ is a probability between 0 and 1, hence π_i is estimated by the proportion of data points for the component G_h.

The component mean is given by:

$$\mu_h^{l+1} = \frac{\sum_t \mathbf{x}^t P(G_h|\mathbf{x}^t, \Phi^l)}{\sum_t P(G_h|\mathbf{x}^t, \Phi^l)}$$

Observe that we use the posterior probability $P(G_h|\mathbf{x}^t, \Phi^l)$ to estimate the component mean used in the next iteration of the E-step.

The component covariance is given by:

$$\Sigma_h^{l+1} = \frac{\sum_t P(G_h|\mathbf{x}^t, \Phi^l)(\mathbf{x}^t - \mu_h^{l+1})(\mathbf{x}^t - \mu_h^{l+1})^T}{\sum_t P(G_h|\mathbf{x}^t, \Phi^l)}$$

Similarly, we use $P(G_h|\mathbf{x}^t, \Phi^l)$ in the estimation of Σ_h^{l+1}.

Also, observe similarities of estimations above to the ones in the multivariate Gaussian distribution:

$$\pi_h = \frac{\sum_t r_h^t}{N} \qquad \mu_h = \frac{\sum_t r_h^t \mathbf{x}^t}{\sum_t r_h^t} \qquad \Sigma_h = \frac{\sum_t r_h^t (\mathbf{x}^t - \mu_h)(\mathbf{x}^t - \mu_h)^T}{\sum_t r_h^t}$$

where we have the labels r_h^t instead of the $P(G_h|\mathbf{x}^t, \Phi^l)$.

(c) Assuming the component prior, mean, and covariance $\pi_h, \mu_h, \Sigma_h, h = 1, \ldots k$, are known, show how the posterior probabilities, $p(G_h|x_i)$ are computed in the E-step.

The component posterior probabilities $P(G_h|\mathbf{x}^t, \Phi^l)$ are computed by the following:

$$P(G_h|\mathbf{x}^t, \Phi^l) = \frac{\pi_h|\Sigma_h|^{-1/2}\exp[-(1/2)(\mathbf{x}^t - \mu_h^t)^T\Sigma_h^{-1}(\mathbf{x}^t - \mu_h^t)]}{\sum_j \pi_j|\Sigma_j|^{-1/2}\exp[-(1/2)(\mathbf{x}^t - \mu_j)^T\Sigma_j^{-1}(\mathbf{x}^t - \mu_j)]}$$

Observe that the numerator consists of the multivairite Gaussian density equation for the component G_h, which is calculated using the estimates π_h, μ_h, and Σ_h. The denominator is the total density estimated by the Gaussian density for all components G_j and it acts as a normalizing term (makes $P(G_h|\mathbf{x}^t, \Phi^l)$ a proper probability).

Problem 3. Please find my implementation in the Code_HW3 folder of the homework. I implemented a 2-class Logistic Regression based on the provided description alongside with the materials from the lectures and the textbook. I used batch gradient descent with and a constant step size $\eta = 0.01$, 500 iterations, and a regularization parameter $\lambda = 1.0$ in my implementation (found them to be quite efficient in producing proper output). The results seems to be similar to the built-in function from scikit-learn.

Error rates for MyLogisticReg2 with Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.0882	0.1584	0.1584	0.1089	0.2079	0.1444	0.0421

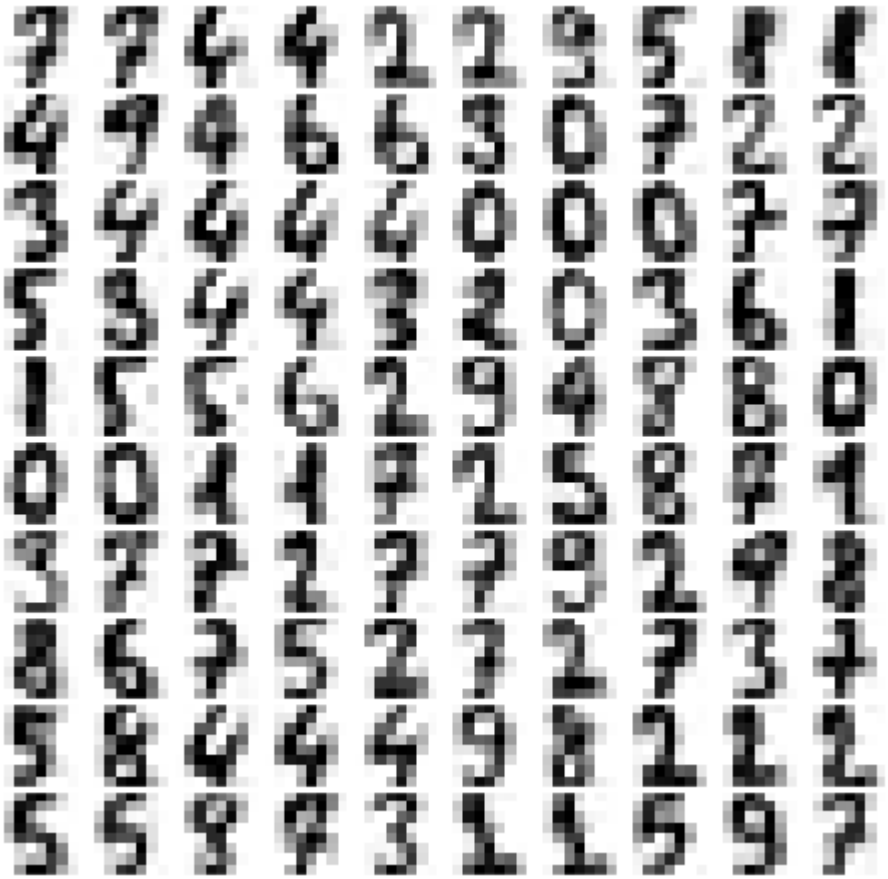
Error rates for MyLogisticReg2 with Boston25						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.0980	0.1089	0.0990	0.1584	0.0891	0.1107	0.0247

Error rates for LogisticRegression with Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.1373	0.1584	0.1287	0.1782	0.1485	0.1502	0.0172

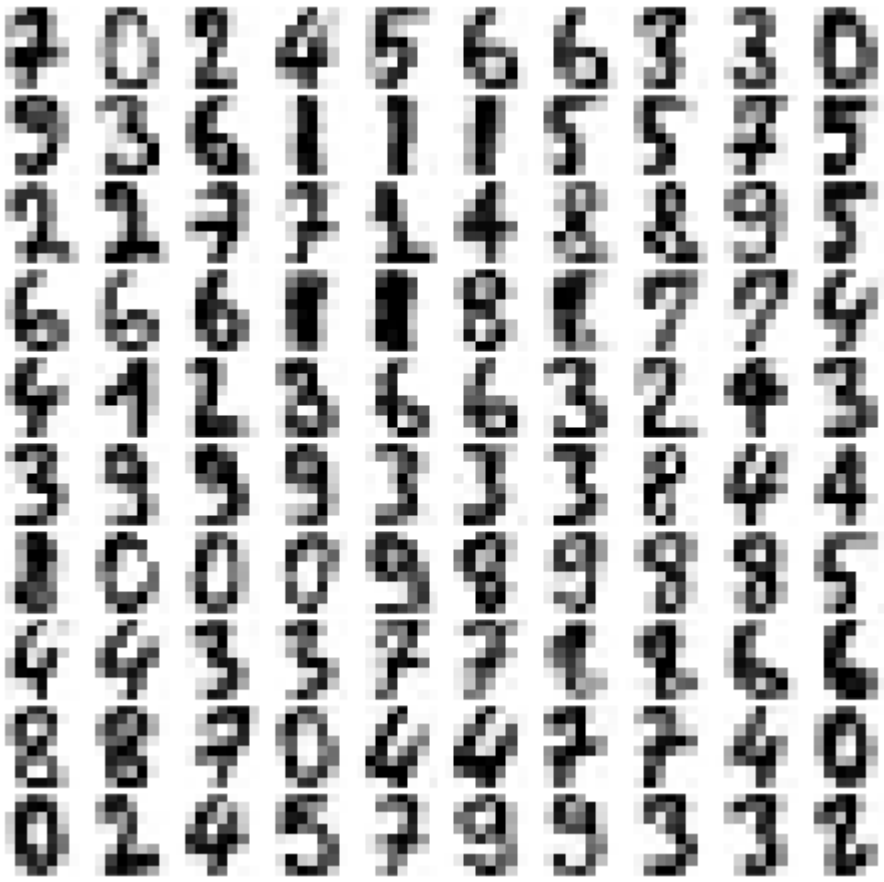
Error rates for LogisticRegression with Boston25						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.0980	0.1584	0.0891	0.0891	0.0990	0.1067	0.0262

Problem 4. I implemented myPCA in Jupyter NoteBook called *my_generate_digits.ipynb*, which can be found in the code folder for current homework. Please take a look at the NoteBook and the comments I left while implementing the function and for more details.

After using myPCA to preserve 90% of the original variance in the projected space, I got the following result:



After using myPCA to preserve 99% of the original variance in the projected space, I got the following result:



For the given data, myPCA seems to produce the results that are similar to the original function.