Stats 315A – HW2 Solutions

8th February, 2012

If there are any questions regarding the solutions or the grades of HW 2, please contact Austen (ahead@stanford.edu) with 'Stats315A-hw2-grading' in the subject line.

Grade Distribution: Total 100 Points

Problem 1: 14
Problem 2: 15
Problem 3: 20
Problem 4: 10
Problem 5: 17
Problem 6: 24

Problem 1 (ESL Exercise 3.12 & 3.30)

ESL 3.12

We augment the original (centered) data matrix $\mathbf{X}$ with p additional rows $\sqrt{\lambda}\mathbf{I}$, and augment the original response $\mathbf{y}$ with p zeros. So, we have

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}, \quad \tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}. \tag{1}$$

Now note

$$\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\beta\|_2^2 = \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix} \beta \right\|_2^2 = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_2^2. \tag{2}$$

So the ordinary least squares objective on the augmented data set is equivalent to the ridge regression objective on the original data set.

ESL 3.30

The solution is similar to 3.12. We augment the original (centered) data matrix $\mathbf{X}$ with p additional rows $\sqrt{\lambda\alpha}\mathbf{I}$, and augment the original response $\mathbf{y}$ with p zeros. So, we have

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda\alpha}\mathbf{I} \end{bmatrix}, \quad \tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}. \tag{3}$$

Also set $\gamma = \lambda(1-\alpha)$. Now we have

$$\begin{aligned} \|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\beta\|_2^2 + \gamma\|\beta\|_1 &= \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\alpha\|\beta\|_2^2 + \gamma\|\beta\|_1 \\ &= \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\left[\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1\right]. \end{aligned} \tag{4}$$

Thus minimizing the RHS over β is equivalent to minimizing the LHS over β, which is a lasso problem.

Problem 2 (ESL Exercise 3.23)

a)

$$\begin{aligned} |\langle \mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle| &= |\langle \mathbf{x}_j, \alpha(\mathbf{y} - x\hat{\beta}) + (1-\alpha)\mathbf{y}\rangle| && (5) \\ &= |\alpha\langle \mathbf{x}_j, \mathbf{y} - x\hat{\beta}\rangle + (1-\alpha)\langle \mathbf{x}_j, \mathbf{y}\rangle| && (6) \end{aligned}$$

Since the residual $\mathbf{y} - \mathbf{X}\hat{\beta}$ is orthogonal to each column of $\mathbf{X}$, $\langle \mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle = 0$ for any j. Hence

$$\frac{1}{N}|\langle \mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle| = \frac{1}{N}|(1-\alpha)\langle \mathbf{x}_j, \mathbf{y}\rangle| = (1-\alpha)\lambda. \tag{7}$$

for any $j = 1, \cdots, p$. Hence, the correlations of each $\mathbf{x}_j$ and the residuals remain equal in magnitude as we progress towards $\mathbf{u}$ (this is the numerator of that correlation).

b) Use the definition of Residual Sum of Squares:

$$RSS = \left\|\mathbf{y} - \mathbf{X}\hat{\beta}\right\|^2 = \langle \mathbf{y} - \mathbf{X}\hat{\beta}, \mathbf{y} - \mathbf{X}\hat{\beta}\rangle \tag{8}$$

And recall that $\langle \mathbf{y} - \mathbf{X}\hat{\beta}, \mathbf{X}\hat{\beta}\rangle = 0$ (according to the similar reason in part (a)). Then

$$\begin{aligned} \frac{1}{N}\|\mathbf{y} - \mathbf{u}(\alpha)\|^2 &= \frac{1}{N}\left\|\alpha(\mathbf{y} - \mathbf{X}\hat{\beta}) + (1-\alpha)\mathbf{y}\right\|^2 \\ &= \frac{1}{N}(1-\alpha)^2\|\mathbf{y}\|^2 + \frac{1}{N}\alpha^2\left\|\mathbf{y} - \mathbf{X}\hat{\beta}\right\|^2 + \frac{2}{N}\alpha(1-\alpha)\langle \mathbf{y} - \mathbf{X}\hat{\beta}, \mathbf{y}\rangle \\ &= (1-\alpha)^2 + \frac{1}{N}\alpha^2 RSS + \frac{2}{N}\alpha(1-\alpha)(\langle \mathbf{y} - \mathbf{X}\hat{\beta}, \mathbf{y} - \mathbf{X}\hat{\beta}\rangle + \langle \mathbf{y} - \mathbf{X}\hat{\beta}, \mathbf{X}\hat{\beta}\rangle) \\ &= (1-\alpha)^2 + \frac{\alpha(2-\alpha)}{N}RSS \end{aligned} \tag{9}$$

Thus

$$\lambda(\alpha) = \frac{\frac{1}{N}|\langle \mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle|}{\sqrt{\frac{\|\mathbf{x}_j\|^2}{N}}\sqrt{\frac{\|\mathbf{y}-\mathbf{u}(\alpha)\|^2}{N}}} = \frac{\lambda(1-\alpha)}{(1-\alpha)^2 + \frac{\alpha(2-\alpha)}{N}RSS} \tag{10}$$

We can further simplify this to $\lambda(\alpha) = \frac{\lambda}{1+\frac{\alpha(2-\alpha)}{(1-\alpha)^2}\frac{RSS}{N}}$, which is a decreasing function as of α from 0 to 1 (since $\frac{\alpha(2-\alpha)}{(1-\alpha)^2} = -1 + \frac{1}{(1-\alpha)^2}$, which is increasing in α).

c) Note that by construction, the LARS algorithm takes the current residual ($\mathbf{y}$ in part (a) and (b)) and the set of variables currently active ($\mathbf{X}$ in part (a) and (b)) and moves the current selection toward the OLS solution of the residual on this active set. Hence, from part (a), the correlations between the residual and the active set of variables remain tied.
Also, from part (b), the magnitude of this correlation decreases monotonically until some inactive variable has the same correlation.

Problem 3 (ESL Exercise 4.2)

a)

As in class, the predicted class $G(x)$ is $\text{argmax}_k \delta_k(x)$ where

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log \pi_k$$

where π_k, μ_k and Σ must be estimated from the data. Letting $\widehat{\pi}_k = N_k/N$, we see that the LDA rule classifies to class 2 if

$$x^T\widehat{\Sigma}^{-1}\widehat{\mu}_2 - \frac{1}{2}\widehat{\mu}_2^T\widehat{\Sigma}^{-1}\widehat{\mu}_2 + \log \widehat{\pi}_2 > x^T\widehat{\Sigma}^{-1}\widehat{\mu}_1 - \frac{1}{2}\widehat{\mu}_1^T\widehat{\Sigma}^{-1}\widehat{\mu}_1 + \log \widehat{\pi}_1$$

which is equivalent to the condition

$$x^T\widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1) > \frac{1}{2}\widehat{\mu}_2^T\widehat{\Sigma}^{-1}\widehat{\mu}_2 - \frac{1}{2}\widehat{\mu}_1^T\widehat{\Sigma}^{-1}\widehat{\mu}_1 + \log(N_1/N) - \log(N_2/N) \tag{11}$$

as desired.

b)

The solution to the least squares problem

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta^T x_i\right)^2$$

has $\widehat{\beta}_0 = \bar{y} - \widehat{\beta}^T\bar{x}$ (follow from the first normal equation). So, without loss of generality we can center both y and x and work with the following equivalent set up

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta' x_i\right)^2 \equiv \min_{\beta_1} \sum_{i=1}^{n} \left((y_i - \bar{y}) - \beta'(x_i - \bar{x})\right)^2.$$

Note that $\bar{y}$ is scalar and $\bar{x}$ is a p-dimensional vector. By the problem we already have $\bar{y} = 0$ (already centered). We represent the centered X by X_c and will rewrite y (when needed) as

$$y = -\frac{N}{N_1}\, y^1 + \frac{N}{N_2}\, y^2$$

where y^i is a vector of 1's and 0's representing the i^{th} class where $i = 1, 2$. And so,

$$X^T y^i = N_i \, \widehat{\mu}_i \text{ for } i = 1, 2.$$

For the centered model the Normal Equations are:

$$X_c^T X_c \, \beta = X_c^T y. \tag{12}$$

It is easy to see that

$$X_c^T y = X^T y - N\bar{y} \cdot \bar{x} = X^T y = N\bigg(N_2^{-1} X^T y^2 - N_1^{-1} X^T y^1 \bigg) = N(\widehat{\mu}_2 - \widehat{\mu}_1)$$

which is the R.H.S of the desired equation.

The L.H.S follows as the total covariance matrix ($\widehat{\Sigma}_{\text{T}}$) is the sum of between-class ($\widehat{\Sigma}_{\text{B}}$) and within-class covariance ($\widehat{\Sigma}_{\text{W}}$) matrices. Note that, this formula assumes all variances are computed with N^{-1} standardization rather than using the corresponding degrees of freedom.
So, we have $N\widehat{\Sigma}_{\text{W}} = (N-2)\widehat{\Sigma}$ and the between-class covariance is given by

$$\widehat{\Sigma}_{\text{B}} = \frac{N_1}{N}\bigg(\widehat{\mu}_1 - \bar{x}\bigg)\bigg(\widehat{\mu}_1 - \bar{x}\bigg)^T + \frac{N_2}{N}\bigg(\widehat{\mu}_2 - \bar{x}\bigg)\bigg(\widehat{\mu}_2 - \bar{x}\bigg)^T \text{ where } \bar{x} = \frac{N_1\widehat{\mu}_1 + N_2\widehat{\mu}_2}{N}$$

which simplifies to

$$\widehat{\Sigma}_{\text{B}} = \frac{N_1 N_2}{N^2}\bigg(\widehat{\mu}_1 - \widehat{\mu}_2\bigg)\bigg(\widehat{\mu}_1 - \widehat{\mu}_2\bigg)^T.$$

Thus we have,

$$X_c^T X_c = N\widehat{\Sigma}_{\text{T}} = N\widehat{\Sigma}_{\text{W}} + N\widehat{\Sigma}_{\text{B}} = (N-2)\widehat{\Sigma} + N\widehat{\Sigma}_{\text{B}}.$$

c)

Let $\alpha = (\widehat{\mu}_2 - \widehat{\mu}_1)^T\beta \in \mathbb{R}$ then $\widehat{\Sigma}_B\beta = (\widehat{\mu}_2 - \widehat{\mu}_1)(\widehat{\mu}_2 - \widehat{\mu}_1)^T\beta = (\widehat{\mu}_2 - \widehat{\mu}_1)\alpha$ is in the direction of $(\widehat{\mu}_2 - \widehat{\mu}_1)$. Using Equation (4.56) we have,

$$(N-2)\widehat{\Sigma}\beta = N(\widehat{\mu}_2 - \widehat{\mu}_1) - \frac{N_1 N_2}{N}\widehat{\Sigma}_B\beta = (\widehat{\mu}_2 - \widehat{\mu}_1)\left[N - \frac{N_1 N_2}{N}\alpha\right]$$

which implies $\widehat{\beta} \propto \widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1)$.

d)

Any other coding $y^\star$ of the classes can be written as $y^\star = c_1 y + c_0\mathbf{1}$ where c_1 and c_0 are constants. Now the RHS of Normal Equations (12) is,

$$X_c^T y^\star = c_1 X_c^T y + c_0 X_c^T \mathbf{1} = c_1 X_c^T y$$

as $X_c^T\mathbf{1} = 0$. The LHS of Equation (12) is independent of y so the desired equation (4.56) still holds (up to a scalar) and the results are still true.

e)

By part (c), we have $\widehat{\beta} = \gamma\widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1)$ where γ is the proportionality constant. Also, from (b) we know $\widehat{\beta}_0 = -\bar{x}^T\widehat{\beta}$. So,

$$\widehat{f}(x) = \widehat{\beta}_0 + \widehat{\beta}^T x = -\frac{1}{N}(N_1\widehat{\mu}_1 + N_2\widehat{\mu}_2)^T\gamma\widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1) + \gamma(\widehat{\mu}_2 - \widehat{\mu}_1)^T\widehat{\Sigma}^{-1}x$$

and it classifies to class 2 if $\widehat{f}(x) > 0$ i.e. iff

$$\begin{aligned}\gamma(\widehat{\mu}_2 - \widehat{\mu}_1)^T\widehat{\Sigma}^{-1}x >& \frac{1}{N}(N_1\widehat{\mu}_1 + N_2\widehat{\mu}_2)^T\gamma\widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1) \\ =& \frac{\gamma}{N}\left[N_1\widehat{\mu}_1^T\widehat{\Sigma}^{-1}\widehat{\mu}_2 + N_2\widehat{\mu}_2^T\widehat{\Sigma}^{-1}\widehat{\mu}_2 - N_1\widehat{\mu}_1^T\widehat{\Sigma}^{-1}\widehat{\mu}_1 - N_2\widehat{\mu}_2^T\widehat{\Sigma}^{-1}\widehat{\mu}_1\right].\end{aligned}$$

This is same as the LDA rule (given by Equation (11)) only when $N_1 = N_2$ and $\gamma > 0$ (which as proved later always holds) in which case the rule becomes

$$(\widehat{\mu}_2 - \widehat{\mu}_1)^T\widehat{\Sigma}^{-1}x > \frac{N_1}{N}\left[\widehat{\mu}_2^T\widehat{\Sigma}^{-1}\widehat{\mu}_2 - \widehat{\mu}_1^T\widehat{\Sigma}^{-1}\widehat{\mu}_1\right].$$

Now we show $\gamma > 0$ always. By plugging in $\widehat{\beta} = \gamma\widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1)$ in Equation (4.56) we have

$$\gamma \times \left[(N-2)||\widehat{\mu}_2 - \widehat{\mu}_1||^2 + \frac{N_1N_2}{N}(\widehat{\mu}_2 - \widehat{\mu}_1)^T\widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1)\right] = N||\widehat{\mu}_2 - \widehat{\mu}_1||^2$$

$$\text{which implies } \gamma = \frac{N||\widehat{\mu}_2 - \widehat{\mu}_1||^2}{(N-2)||\widehat{\mu}_2 - \widehat{\mu}_1||^2 + \frac{N_1N_2}{N}(\widehat{\mu}_2 - \widehat{\mu}_1)^T\widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1)} > 0.$$

Problem 4 (discriminating variables are uncorrelated)

Since the covariance remains constant for changes in the mean, assume that the variables are mean-centered. The covariance between the top two disciminant variables is

$$\text{Cov}(Z_1, Z_2) = \text{Cov}(a_1'X, a_2'X) = a_1'\text{Cov}(X)a_2 \tag{13}$$

We know that $\text{Cov}(X) = \mathbf{B} + \mathbf{W}$, where $\mathbf{B}$ denotes the between class covariance and $\mathbf{W}$ is the within class covariance (see ESL 2nd ed page 114).

We need to show that:

$$\text{Cov}(Z_1, Z_2) = a_1'(\mathbf{B} + \mathbf{W})a_2 = 0 \tag{14}$$

For this observe that $a_1'\mathbf{W}a_2 = 0$, by definition, since a_2 is orthogonal in $\mathbf{W}$ to a_1 (see ESL page 116). We will show that $a_1'\mathbf{B}a_2 = 0$, to complete the proof.

Now a_1 is by definition an eigenvector of $\mathbf{W}^{-1}\mathbf{B}$, and so

$$a_1'(\mathbf{B})a_2 = (\mathbf{B}a_1)'a_2 \propto a_1'\mathbf{W}a_2 = 0 \tag{15}$$

which completes the proof.

For the sample-version, the above arguments carry through, with $\widehat{\mathbf{B}}, \widehat{\mathbf{W}}$ and $\widehat{\text{Cov}}(X)$, denoting the sample estimates and the discriminant directions a_1, a_2 being defined with respect to the sample versions.

Problem 5 (ridge regression)

a)

$$\hat{\beta}_\lambda = (X^T X + \lambda I_p)^{-1} X^T y, \tag{16}$$

where $X^T X + \lambda I_p$ is the $10,000 \times 10,000$ matrix. The computations is of the order of $p^3 = 10^{12}$ operations.

b) The derivative condition is

$$(X^T X + \lambda I_p)\hat{\beta}_\lambda = X^T y. \tag{17}$$

which means

$$\lambda \hat{\beta}_\lambda = X^T (y - X\hat{\beta}_\lambda) \tag{18}$$

So $\alpha = \frac{1}{\lambda}\left(y - X\hat{\beta}_\lambda\right)$.

c) By part (b),

$$\hat{\beta}_\lambda = X^T \alpha = V D U^T \alpha \tag{19}$$

So let $\theta = DU^T\alpha$, which gives $\hat{\beta}_\lambda = V\theta$.

Observe $\beta^T\beta = \theta^T\theta$ and $X\beta = UD\theta$, if we let $X^* = UD$, then we can get $\hat{\theta}_\lambda$ by solving the following ridge regression problem

$$\hat{\theta}_\lambda = \arg\min_\theta |y - X^*\theta|^2 + \lambda\theta^T\theta \tag{20}$$

then $\hat{\beta}_\lambda = V\hat{\theta}_\lambda$. Because X^* is a $n \times n$ matrix, the computation for $\hat{\theta}_\lambda$ is at least dramatically reduced to order $n^3 = 50^3$.

Additionally. If writing down the expression for $\hat{\theta}_\lambda$, we see

$$\hat{\theta}_\lambda = (X^{*T} X^* + \lambda I_n)^{-1} X^* y = (D^2 + \lambda I_n)^{-1} D U^T y. \tag{21}$$

Since D is a diagonal matrix, the inversion of $D^2 + \lambda I_n$ is trivial for all λ. Hence computing $\hat{\theta}_\lambda$ almost costs nothing once the SVD of X is done; and we can use it for all λ without any extra cost.

d) With a new observation x_0 we predict $x_0\hat{\beta}_\lambda$. We can optimize the regression fit by selecting a good λ using cross-validation.

Problem 6 (zipcode classification)

Classification Method	Training Error	Test Error
(a) LDA on original 256 dimensional data	0.0159	0.0874
(b) LDA on the leading 49 principal components	0.0438	0.0853
(c) LDA on filtered data	0.0336	0.0752
(d) Multiple linear logistic regression on filtered data	0.0028	0.0915

Conclusion & Comments:

I. Among the 4 classifiers, method (c) (LDA on the Filtered data) has the best performance on the test set.

II. Compared to method (a), (b) increases both training error and test error. This suggests using principal components for feature selection in this context can be the wrong thing to do since the direction in the feature space that best separates the classes may have low variance (in which case we would be throwing out the very information we need).

III. The training error is way higher than the test error in (a). This suggests that lda in the original space (part a) is overfitting to the training set. By filtering, we reduce the model complexity (as shown at the end of this problem) and thereby decrease overfitting.

IV. The multiple linear logistic regression on the filtered data yielded worse prediction performance than the ordinary LDA on the filtered data which suggests that sigmoid-curved general linear models may not be proper in this case.

Filtering & Constrained GLMs

We compare fitting a constrained linear logistic model with the full set of pixels to fitting an unconstrained linear logistic model with the filtered features of procedure (c).

We adopt a different indexing convention of the feature vector $x \in \Re^{256}$. Let $x_{i,j}^{(k)}$ (for $k = 1, \ldots, 4$ and $i, j = 1, \ldots, 8$) denote the kth pixel value of the 2×2 block at position (i, j), and let $\beta_{i,j}^{(k)}$ be the corresponding coefficient. Consider the logistic linear model

$$f(x) = \frac{1}{1 + \exp(-x^T \beta\,)} = \left\{ 1 + \exp\left(\sum_{i=1}^{8} \sum_{j=1}^{8} \sum_{k=1}^{4} \beta_{i,j}^{(k)} x_{i,j}^{(k)} . \right) \right\}^{-1}$$

The 2×2 block constraints are of the form $\beta_{i,j}^{(1)} = \beta_{i,j}^{(2)} = \beta_{i,j}^{(3)} = \beta_{i,j}^{(4)}$ for all i and j. Thus,

$$\begin{aligned} x^T \beta &= \sum_{i=1}^{8} \sum_{j=1}^{8} \sum_{k=1}^{4} \beta_{i,j}^{(1)} x_{i,j}^{(k)} \\ &= \sum_{i=1}^{8} \sum_{j=1}^{8} \beta_{i,j}^{(1)} \sum_{k=1}^{4} x_{i,j}^{(k)} \\ &= \sum_{i=1}^{8} \sum_{j=1}^{8} 4 \times \beta_{i,j}^{(1)} z_{i,j} \\ &= z^T \tilde{\beta}, \end{aligned}$$

where $z \in \Re^{64}$ is the filtered feature vector from part (c). Thus, optimizing over $\beta \in \Re^{256}$ subject to the constraints is equivalent to optimizing (with no constraints) over $\tilde{\beta} \in \Re^{64}$ with the filtered data.

Test Error vs Deviance explained on training data

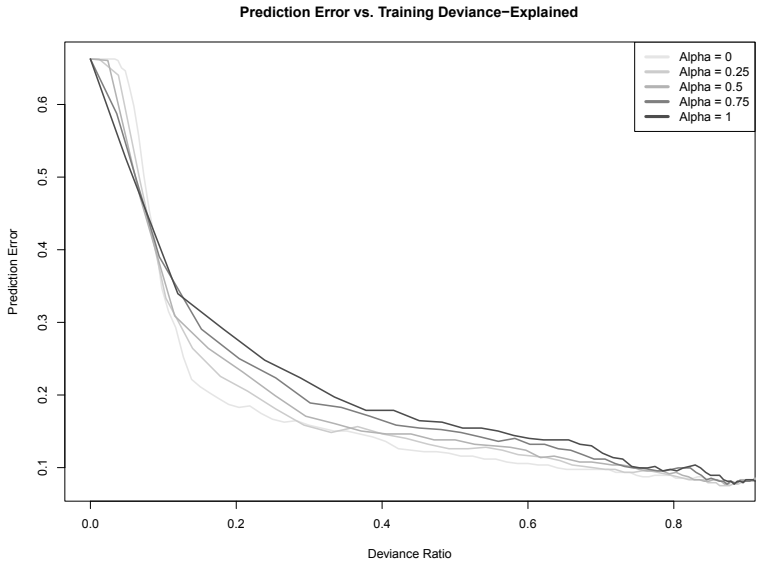

Figure 1: Test error vs % Deviance explained on the training data

The figure shows the plot of test error against the % deviance explained on the training data for $\alpha = 0,0.25,0.5,0.75$ and 1 respectively. Test-error is generally decreasing in % deviance explained. The rate of decrease slackens down around 40 % but isn't drastic enough to stop training at that level. Also, as the value of α decreases we have lower test error curves. This implies that the filtered data has a dense representation w.r.t to the Harr basis.

R-code:

```
library(MASS);
# load data if necessary:
if(sum(ls()=="train.2") + sum(ls()=="train.3") < 2){
path = "http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/zip.digits/"
train.3 = read.csv(paste(path,"train.3",sep=""),header=F)
train.5 = read.csv(paste(path,"train.5",sep=""),header=F)
train.8 = read.csv(paste(path,"train.8",sep=""),header=F)
}
xtrain = rbind(as.matrix(train.3),as.matrix(train.5),as.matrix(train.8))
ytrain = rep(c(3,5,8),c(nrow(train.3),nrow(train.5),nrow(train.8)))
test = as.matrix(read.table("zip.test"))
```

```
ytest = test[,1]
xtest = test[ytest==3 | ytest==5 | ytest==8,-1]
ytest = ytest[ytest==3 | ytest==5 | ytest==8]
# view digit as an image
showDigit <- function(x,...){
p=length(x)
d = sqrt(p)
if(d != round(d)) stop("Must be perfect square")
x=matrix(as.numeric(x),d,d)
image(x[,d:1],axes=F,...)
}
#part a ------------------------------------------------------------------------
l=lda(xtrain,ytrain)
sum(predict(l)$class!=ytrain)/length(ytrain)
sum(predict(l,xtest)$class!=ytest)/length(ytest)
#part b ------------------------------------------------------------------------
#centering based on the training data
xtrain.center<-apply(xtrain,2,mean);
xxtrain = scale(xtrain,center=xtrain.center,scale=F)
xxtest = scale(xtest,center=xtrain.center,scale=F)
V = svd(xxtrain)$v[,1:49]
pcstrain = xxtrain%*% V
pcstest = xxtest%*% V
l=lda(pcstrain,ytrain)
sum(predict(l)$class!=ytrain)/length(ytrain)
sum(predict(l,pcstest)$class!=ytest)/length(ytest)
#part c ------------------------------------------------------------------------
filterdigit <- function(x){
# averages each non-overlapping 2x2 block
x = matrix(x,16,16)
evens = 2 * (1:8)
x = x[evens,] + x[-evens,]
x = x[,evens] + x[,-evens]
as.vector(x)/4
}
xtrain = t(apply(xtrain,1,filterdigit))
xtest = t(apply(xtest,1,filterdigit))
l = lda(xtrain,ytrain)
sum(predict(l)$class!=ytrain)/length(ytrain)
sum(predict(l,xtest)$class!=ytest)/length(ytest)
#part d ------------------------------------------------------------------------
library(glmnet)
l = glmnet(xtrain,factor(ytrain),family="multinomial")
sum(as.numeric(predict(l,xtrain,s=0,type="class"))!=ytrain)/length(ytrain)
sum(as.numeric(predict(l,xtest,s=0,type="class"))!=ytest)/length(ytest)
#plot of deviance explained vs test err
alpha.values= c(0,.25,.5,.75,1);
pred.err = matrix(0,length(alpha.values),100)
dev.explained = matrix(0,length(alpha.values),100)
for (i in 1:length(alpha.values)){
l= glmnet(xtrain,factor(ytrain),family="multinomial",alpha=alpha.values[i]);
dev.explained[i,] = l$dev.ratio;
pred.err[i,] = apply(matrix(as.numeric(predict(l,xtest,type="class")!=ytest),ncol
    =100),2,mean);
}
my.colors = c("gray90","gray80","gray70","gray50","gray30","black")
```

```
plot(dev.explained[1,],pred.err[1,],type="l",col=my.colors[1],lwd=2,
     xlab="Deviance Ratio",ylab="Prediction Error");
title("Prediction Error vs. Training Deviance-Explained")
for(i in 2:length(alpha.values)){
lines(dev.explained[i,],pred.err[i,],col=my.colors[i],lwd=2)
}
alpha.legend = character(length(alpha.values))
for(i in 1:length(alpha.values)){alpha.legend[i] = paste("Alpha =",alpha.values[i])}
legend(x="topright",legend = alpha.legend,lwd=4,col=my.colors)
```

'R-code'