Statistical Foundations of Data Science

Jianqing Fan

Princeton University

https://fan.princeton.edu

ZOOM ID Lectures: **970 4936 8998**   Office Hours: **996 4030 7631**

Annotated Lecture Notes: web view

3. Generalized Linear Models and Penalized Likelihood

3.1 Generalized linear models

Read materials and R-implementations here

http://orfe.princeton.edu/%7Ejqfan/fan/classes/245/chap12.pdf

Binary Response

Dichotomized response: Very frequently

Example: (Gene expression and autism) Over 60K gene expression profiles (Next Generation Sequence) are measured among 104 samples: 47 autisms and 57 healthy controls, along with gender, brain region, age, and sites. Of interest is to find the genes that are associated with autism. We select top 5 differently expressed (**feature screening**) by using two-sample t-test and would like to examine their effect on the response along with other variables.

Response: $Y = 1$ and 0, indicating autism or not.

Question: How to model $p(\mathbf{x}) = P(Y = 1 | \mathbf{X} = \mathbf{x})$?

Modeling Binary Data

If a latent response (e.g. severity) follows

$$Z = \alpha + \beta^T \mathbf{X} - \varepsilon, \qquad \textbf{linear model},$$

but we only get $Y = I(Z > c)$ for an unknown c.

Conditional probability: if $\varepsilon \sim F$, we have

$$p(\mathbf{x}) = P(Y = 1 | X = x) = P(\alpha + \mathbf{x}^T\beta - \varepsilon > c | \mathbf{x}) = F(\beta_0 + \mathbf{x}^T\beta)$$

where $\beta_0 = \alpha - c$.

Link function = $F^{-1}(\cdot)$.

★ **probit link**: $F(x) = \Phi(x)$, normal cdf. ⟶ $p(\mathbf{x}) = \Phi(\beta_0 + \mathbf{x}^T\beta)$

★ **logit link**: $F(x) = \frac{\exp(x)}{1+\exp(x)}$, ⟶ $p(\mathbf{x}) = \frac{\exp(\beta_0 + \mathbf{x}^T\beta)}{1+\exp(\beta_0+\mathbf{x}^T\beta)}$ *(softmax)*

Dynamic pricing – another application

Price: $v(\mathbf{x}) = \mathbf{x}^T\theta - \varepsilon$, $\quad \mathbf{x} =$ attributes (e.g. Airbnb), $\quad \varepsilon \sim F$.

Observe: $Y = 1$ if $v(x) > p$, $\quad p$ asked price.

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = F(\mathbf{x}^T\theta - p)$$

Optimal price: $p^*(\mathbf{x}) = \text{argmax}_p \ pF(\mathbf{x}^T\theta - p)$

expected rev.

Goal: learn θ and F from data $\{(\mathbf{x}_t, p_t, y_t)\}$ dynamically with min regret.

★GLIM with unknown link.

Binomial distribution: a member of exponential family

Suppose that $(Y|\mathbf{X}=\mathbf{x}) \sim \text{Binomial}(m, p(\mathbf{x}))$. Then

$$\begin{aligned}
& P(Y=y|\mathbf{X}=\mathbf{x}) \\
= \; & \binom{m}{y} p(\mathbf{x})^y (1-p(\mathbf{x}))^{m-y} \\
= \; & \exp\Big\{ y \underbrace{\log\frac{p}{1-p}}_{\theta=\text{canonical parameter}} + \underbrace{m\log(1-p)}_{-b(\theta)} + \underbrace{\log\binom{m}{y}}_{c(y)} \Big\}.
\end{aligned}$$

■ Binary response: $m=1$.

Normal distribution

If $(Y|\mathbf{X}=\mathbf{x}) \sim N(\mu(\mathbf{x}), \sigma^2)$, then

$$\begin{aligned} f(y;\mu,\sigma) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\mu(\mathbf{x}))^2}{2\sigma^2}\right) \\ &= \exp\left(\frac{y\mu-\mu^2/2}{\sigma^2}-\frac{y^2}{2\sigma^2}-\log\sqrt{2\pi}\sigma\right). \end{aligned}$$

Here $\theta=\mu, \phi=\sigma^2$, $b(\theta)=\theta^2/2$ and $c(y,\phi)=-\frac{y^2}{2\sigma^2}-\log(\sqrt{2\pi}\sigma)$.

■The canonical link function is the identity link $g(t)=t$.

Generalized linear models

Purpose: To accommodate various types of responses (binary, categorical, counts, continuous)

GLIM: $f(y|\mathbf{X}=\mathbf{x};\theta) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right\}$ with $g(\mu(\mathbf{x})) = \mathbf{x}^T\beta$

can. param. disp. para

Regression: $\mu(\mathbf{x}) \equiv E(Y|\mathbf{x}) = b'(\theta(\mathbf{x}))$ (fact)

General link: $g(\mu(\mathbf{x})) = \mathbf{x}^T\beta \quad \Longleftrightarrow \quad \theta(\mathbf{x}) = (b')^{-1}(g^{-1}(\mathbf{x}^T\beta))$.

canonial link: take $\quad g(\mu) = b'^{-1}(\mu) = \theta = \mathbf{x}^T\beta.$

★normal: $g(\mu) = \mu \qquad$ Bernoulli: $g(p) = \log\frac{p}{1-p}$ = logit link

Poisson Distribution

Assume that $(Y|\mathbf{X}=\mathbf{x}) \sim \text{Poisson}(\lambda(\mathbf{x}))$. Then

$$
\begin{aligned}
P(Y=y|\mathbf{X}=\mathbf{x}) &= \frac{\lambda(\mathbf{x})^y \exp(-\lambda(\mathbf{x}))}{y!} \\
&= \exp(y \underbrace{\log\lambda(\mathbf{x})}_{\theta(\mathbf{x})} - \underbrace{\lambda(\mathbf{x})}_{b(\theta(\mathbf{x}))} \underbrace{-\log y!}_{c(y,\phi)}).
\end{aligned}
$$

$b(\theta) = \lambda = \exp(\theta), \qquad c(y,\phi) = -\log y!, \qquad \text{with } \phi = 1.$

- Useful for situations in which mean and variance approx. the same.

Statistical inferences

Likelihood: $\ell_n(\beta) = \sum_{i=1}^n \log f(y_i|\mathbf{x}_i) \propto \sum_{i=1}^n [y_i\theta_i - b(\theta_i)]$, $\theta_i = \mathbf{x}_i^T\beta$.

Estimated Variance: $\widehat{\text{var}}(\widehat{\beta}) = -[\ell_n''(\widehat{\beta})]^{-1} = \phi[\sum_{i=1}^n b''(\theta_i)\mathbf{x}\mathbf{x}_i^T]^{-1}$

Deviance: Let $\tilde{\theta}_i = (b')^{-1}(y_i)$ be unrestricted MLE. ⟵ ext. of RSS

$$
\begin{aligned}
D(\mathbf{y};\widehat{\mu}) &= 2\{\max_{\theta \text{ free}} \ell_n(\theta) - \max_{\theta \in model} \ell_n(\theta)\} \\
&= \sum_{i=1}^n 2\{y_i(\tilde{\theta}_i - \widehat{\theta}_i) - b(\tilde{\theta}_i) + b(\widehat{\theta}_i)\} \equiv \sum_{i=1}^n d_i^2.
\end{aligned}
$$

Deviance residuals: $r_{D,i} = d_i \operatorname{sgn}(y_i - \widehat{\mu}_i)$.

$$
\begin{aligned}
&\text{Deviance(smaller model)} - \text{Deviance(larger model)} \\
= \quad & 2\{\max_{\theta \in \Theta_1} \ell_n(\theta) - \max_{\theta \in \Theta_0} \ell_n(\theta)\} \to \chi^2_{\dim(\Theta_1)-\dim(\Theta_0)}.
\end{aligned}
$$

Example: (**Gene expression and autism**) Over 60K gene expression profiles (Next Generation Sequence) are measured among 104 samples: 47 autisms and 57 healthy controls, along with gender, brain region, age, and sites. Of interest is to find the genes that are associated with autism. We select top 5 differently expressed by using two-sample t-test and fit logistic regression along with other variables.

Data: autism.csv

```
> autism  = read.csv("autism.csv")    #reading the data
> aut.glm = glm(Autism ~ . , family=binomial, data=autism)
>    #fitting the model
> summary(aut.glm)   #summarize the fit

Call:
glm(formula = Autism ~ ., family = binomial, data = autism)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4105  -0.5834  -0.1647   0.4863   2.5613

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.33425    2.56463  -0.520 0.602889
GenderM      0.14585    0.73279   0.199 0.842233
Age         -0.05945    0.02871  -2.071 0.038365 *
SiteM       -3.43602    0.95416  -3.601 0.000317 ***
Reg          1.17445    0.57933   2.027 0.042636 *
Gene1       -0.10237    0.14148  -0.724 0.469332
```

```
Gene2         0.43250    0.32752   1.321 0.186658
Gene3         0.78675    0.26275   2.994 0.002751 **
Gene5        -0.66137    0.30426  -2.174 0.029729 *
NA.           0.08676    0.26373   0.329 0.742165
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 143.212  on 103  degrees of freedom
Residual deviance:  74.617  on  94  degrees of freedom
AIC: 94.617
```

We now select model by using stepwise procedure `step(aut.glm)`. It selects the model:

```
> aut.glm1 = glm(Autism ~ Age + Site + Reg + Gene3 + Gene5,
             family=binomial, data=autism)
> summary(aut.glm1)     #summarize the fit

             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.01125    2.12388   0.005 0.995773
Age         -0.06377    0.02804  -2.275 0.022928 *
SiteM       -3.31923    0.85777  -3.870 0.000109 ***
Reg          1.05110    0.52212   2.013 0.044099 *
Gene3        0.89643    0.22623   3.962 7.42e-05 ***
Gene5       -0.51391    0.18172  -2.828 0.004684 **
```

We now predict (in-sample) and compute the misclassification rate. For each given $\mathbf{x}$, we compute $p(\mathbf{x}) = \frac{\exp(\widehat{\beta}^T \mathbf{x})}{1+\exp(\widehat{\beta}^T \mathbf{x})}$, which is the estimated probability $P(Y|\mathbf{X} = x)$. Classify it as 1 if $p(\mathbf{x}) > 0.5$. The in-sample misclassification rate is 13.46%

```
> logit = predict(aut.glm1)            #fitted log(odd-ratios)
> prob = exp(logit)/(1+exp(logit))    #fitted probability
> classification = (prob > 0.5)       #classification
     ### equivalent to directly using   (logit > 0)
> mean(autism[,1] != classification)   #compute misclassification rate
[1] 0.1346154
```

3.2 Penalized Quasi-likelihood

Penalized Quasi-likelihood *(Sec 5.4 & 5.5)*

Objective: Find **sparse** β to minimize $Q(\beta) = \sum_{i=1}^n L(Y_i, \mathbf{x}_i^T\beta)$.

- **GLIM**: $L(Y_i, \mathbf{x}_i^T\beta) = b(\mathbf{x}_i^T\beta) - Y_i\mathbf{x}_i^T\beta$. ⟵ neg. log-likelihood
- **Classification**: $Y = \pm 1$.

  ★SVM $L(Y_i, \mathbf{x}_i^T\beta) = (1 - Y_i\mathbf{x}_i^T\beta)_+$.

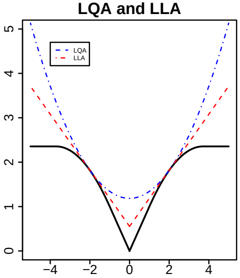
  ★AdaBoost $L(Y_i, \mathbf{x}_i^T\beta) = \exp(-Y_i\mathbf{x}_i^T\beta)$.
- **Robustness**: $L(Y_i, \mathbf{x}_i^T\beta) = |Y_i - \mathbf{x}_i^T\beta|$.
- **Quantile regression**; $L(y, x) = \alpha x_+ + (1-\alpha)x_-$.

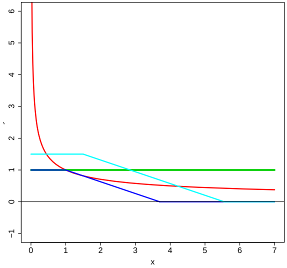
Solution: minimize $Q(\beta) = \sum_{i=1}^n L(Y_i, \mathbf{x}_i^T\beta) + \sum_{j=1}^p p_\lambda(|\beta_j|)$.

Iterated reweighted Convex Optimization

$$Q(\beta) = \sum_{i=1}^{n} L(\mathbf{X}_i^T\beta, Y_i) + \sum_{j=1}^{p} \mathbf{p}_\lambda(|\beta_\mathbf{j}|).$$

$$p_\lambda(|\beta_j^{(k)}|) + \mathbf{p}'_\lambda(|\beta_\mathbf{j}^{(\mathbf{k})}|)(|\beta_\mathbf{j}| - |\beta_j^{(k)}|)$$





- $\beta^{(0)} = 0 \Longrightarrow w_j^{(0)} = p'_\lambda(0+) \Longrightarrow$ LASSO.
- Iteration reduces the bias: $w_j^{(k)} = p'_\lambda(|\beta_j^{(k)}|)$
- Zero is a non-absorbing state (comparing adaptive-Lasso $w_j = 1/|\beta_j^{(k)}|^\gamma$).

Oracle estimator and oracle properties

Active set: $S = \{j, \beta_{j,0} \neq 0\}$ (non-sparse set).

$s = |S|$ —**intrinsic dim** $\ll n$.

Oracle estimator: $\widehat{\beta}^o_{S^c} = 0, \quad \widehat{\beta}^o_S = \mathrm{argmin}\{\sum_{i=1}^n L(Y_i, \mathbf{x}_{i,S}^T \beta_S)\}$.

Oracle property: Behave similarly to the oracle estimator:

$$P\{\widehat{\beta}_{S^c} = 0\} \to 1, \qquad \mathbf{a}^T \widehat{\beta}_S \overset{d}{\approx} \mathbf{a}^T \widehat{\beta}^o_S.$$

or more strongly $P\{\widehat{\beta} = \widehat{\beta}^o\} \to 1$.

3.3 Properties of Penalized Likelihood

★Classical low-dimensional results *(Sec 5.8.2)*

★Folded concave PMLE has an oracle property;

★Lasso can not have;

★PMLE has L_2 rate $O_p(\sqrt{s}n^{-1/2})$ and oracle property.

Consistency: Finite p

Let β_0 the true value of β. Denote

$$a_n = \max\{p'_\lambda(|\beta_{j0}|) : \beta_{j0} \neq 0\},$$
$$b_n = \max\{|p''_\lambda(|\beta_{j0}|)| : \beta_{j0} \neq 0\}$$

Theorem 3.1 (finite p). If $b_n \to 0$, exists a local maximizer $\widehat{\beta}$ such that

$$\|\widehat{\beta} - \beta_0\| = O_P(\mathbf{n^{-1/2} + a_n}).$$

- By choosing a proper λ_n, **root-n consistency**
- If $\lambda_n \to 0$, **root-n consistency** for Hard and SCAD (*Bias* $=$ *0*).

Oracle Property

$\beta_0 = (\beta_{10}^T, \beta_{20}^T)^T$. WLOG, assume that $\beta_{20} = \mathbf{0}$.

Theorem 3.2 *(Fan & Li, 01)* If $\lambda_n \to 0$ and $\sqrt{n}\lambda_n \to \infty$, and

$$\liminf_{n\to\infty} \liminf_{\theta\to 0^+} \lambda_n^{-1} p'_{\lambda_n}(\theta) > 0,$$

then root-n local max $\widehat{\beta} = (\widehat{\beta}_1^T, \widehat{\beta}_2^T)^T$ in Thm 3.2 satisfies

1. (**Sparsity**) $\widehat{\beta}_2 = \mathbf{0}$;
2. (**Asymptotic Normality**) For Hard, SCAD, MCP,

$$\sqrt{n}(\widehat{\beta}_1 - \beta_{10}) \to N\{\mathbf{0}, I_1^{-1}(\beta_{10})\}, \qquad \widehat{\beta}_2 = \mathbf{0},$$

where $I_1(\beta_{10})$ = Fisher information knowing $\beta_2 = \mathbf{0}$ **(Oracle property)**.

Comments

★ For L_1 penalty, $a_n = \lambda_n$.

- Root-n consistency requires that $\lambda_n = O_P(n^{-1/2})$ (**bias**).
- Oracle property requires that $\sqrt{n}\lambda_n \to \infty$ (**Sparsistency**).
- They can not be satisfied simultaneously.

★ No oracle property for LASSO *(Fan and Li, 01; Zou, 06)*

★ Extend results to $d_n = O(n^{1/5})$ for general model *(Fan and Peng, 04)*

★ SCAD is an oracle estimator *(Kim, et al., 08)*

Strong oracle property under ultrahigh dimensions

Conditions for GLIM (*Fan and Lv, 2011*): SCAD-like penalty

■ min signal: $d_n = \min\{|\beta_{0,j}| : \beta_{0,j} \neq 0\} \gg \lambda_{\mathbf{n}}$.

■ Design matrix $\mathbf{X}$ satisfies

$$\left\| \mathbf{X}_2^T b''(\theta_0) \mathbf{X}_1 \left[\mathbf{X}_1^T b''(\theta_0) \mathbf{X}_1 \right]^{-1} \right\|_\infty = O(n^{\alpha_1}). \qquad \theta_0 = \mathbf{X}\beta_0$$

♣ For LS, it reduces to **irrepresentable condition** on $\|\mathbf{X}_2^T \mathbf{X}_1 [\mathbf{X}_1^T \mathbf{X}_1]^{-1}\|_\infty$, **much weaker**

■ **Choice of λ**: $\lambda_n \gg n^{-(0.5-\alpha_1)}(\log n)^2, \quad \alpha_1 < 1/2$.

Strong oracle property

Capacity: $s = o(n)$, $\log p = O(n^{2\alpha_1})$.

Theorem 3.3: There is a local maximizer such that

$\widehat{\beta}_2 = \mathbf{0}$ and $\|\widehat{\beta} - \beta_0\|_2 = O_P(\sqrt{s}n^{-1/2})$ and

$$\sqrt{n}\left(\widehat{\beta}_1 - \beta_1\right) \xrightarrow{D} N(\mathbf{0}, \phi\left[n^{-1}\mathbf{X}_1^T b''(\theta_0)\mathbf{X}_1\right]^{-1}).$$

Fisher Information

Good News: All local minimizers lie within statist. precision (*Loh and Wainwright, 14, AOS)*

Summary of Theoretical Studies

1. Lasso and SCAD have good MSE property and predictive power.
2. Lasso has model selection consistency, but requires **restricted** conditions, depending on size of the true model and correlations of predictors. This leads to **false negatives and many false positives**.
3. SCAD has **better** model selection consistency, possess **oracle** properties, about the same computation as Lasso.

3.4 Numerical Properties

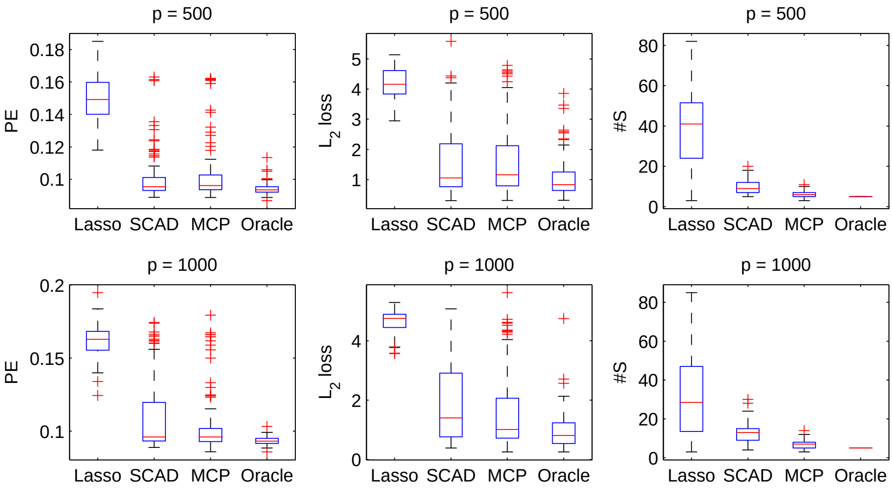
Logistic regression — small p

- Covariate $\mathbf{x} \sim N(0, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = (0.5^{|i-j|})$.
- $\beta_1 = (2.5, -1.9, 2.8, -2.2, 3)^T$, $n = 200$, $p = 25$.

Measures	Lasso	SCAD	MCP	Oracle
PE	**0.11**(0.01)	**0.10**(0.01)	0.10(0.01)	0.09(0.00)
L_2 loss	**3.06**(0.66)	**0.94**(0.55)	0.94(0.55)	0.88(0.34)
L_1 loss	**7.25**(1.10)	**1.87**(1.46)	1.87(1.46)	1.73(0.77)
Deviance	**129.4**(19.2)	**111.8**(15.8)	111.82(15.80)	113.12(16.05)
#S	**9(**2.97)	**5**(0.74)	5(0.74)	5(0)
FN	0(0)	0(0)	0(0)	0(0)

★Lasso has false negatives, creating many false positives in high-d.

Logistic regression — large p



★Lasso has false negatives, creating many false positives in high-d.

Neuroblastoma Data (MAQC-II)

1. 251 patients of the German Neuroblastoma Trials NB90-NB2004, diagnosed between 1989 and 2004, aged from 0 to 296 months (median 15 months).
2. 251 customized oligonucleotide microarray with $p = 10,707$.
3. focus on "3-year Event Free Survival", ($n = 239$ w/ 49 "$+$" and 190 "$-$").
4. **Aims**: To study which genes are responsible for neuroblastoma and their risk association.

Results

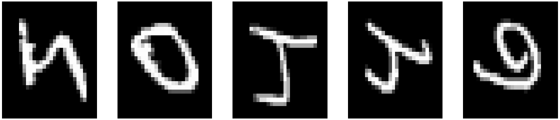
Training set and endpoints:

1. **"3-y EFS"**: Random 25 "$+$" and 100 "$-$".
2. **"Gender"**: Random 120 males and 50 females. Total: 246.

Table: Classification errors in the neuroblastoma data set

	3-year EFS		Gender	
Method	# of genes	Test error	# of genes	Test error
Lasso	**56**	**23/114**	**4**	**5/126**
SCAD	**10**	**18/114**	**2**	**4/126**
MCP	7	23/114	1	12/126
SIS	5	19/114	6	4/126

Example: The Mixed National Institute of Standards and Technology (MNIST for short) data consists of 70000 handwritten digits (28$\times$ 28 grey images, the images are rotated in the same way): 60K for training and 10K for testing. It has been popularly used as a benchmark data set for machine learning algorithms. It is included in the Keras package. See https://tensorflow.rstudio.com/guide/keras/



```
                                     #install R then Rstudio
install.packages("keras")            #install the package, use only Rstudio
library(keras)                       #use the package
install_keras()                      #needed only for the first time

   ########## extracting data  ###########
library(keras)
mnist <- dataset_mnist()
x_train <- mnist$train$x
y_train <- mnist$train$y
x_test <- mnist$test$x
y_test <- mnist$test$y
dim(x_train)
[1] 60000    28    28
```

```
y_train[1:15]
 [1] 5 0 4 1 9 2 1 3 1 4 3 5 3 6 1

    #let us take a look of the data
  par(mfrow=c(1,5), mar=c(5,1,1,1)+0.1)  #set graph margin c(5,5,3,1)+.1
  image(x_train[1,,], axes = FALSE, col = grey(seq(0, 1, length = 256)))
  image(x_train[2,,], axes = FALSE, col = grey(seq(0, 1, length = 256)))
  image(x_train[3,,], axes = FALSE, col = grey(seq(0, 1, length = 256)))
  image(x_test[7,,], axes = FALSE, col = grey(seq(0, 1, length = 256)))
  image(x_test[12,,], axes = FALSE, col = grey(seq(0, 1, length = 256)))
c(y_train[1:3],y_test[7], y_test[12])
[1] 5 0 4 4 6


############################################################################
############## Building  Logistic and Penalized Logistic Regression #########
############################################################################
  #reshape into a matrix and rescale them; create binary variable for digit 4

xtrain <- array_reshape(x_train, c(nrow(x_train), 784))
xtest <- array_reshape(x_test, c(nrow(x_test), 784))
xtrain <- xtrain / 255
xtest <- xtest / 255
ytrain = rep(0,60000); ytrain[y_train==4] = 1;          #classify digit 4
ytrain[1:20]; sum(ytrain)                    #show 20 incidence and total cases
[1] 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
[1] 5842
```

```
ytest = rep(0,10000); ytest[y_test ==4] = 1;
  ##### delete variables has small variances, 67 have exact 0 variance
ind = (1:784)[apply(xtrain, 2, var)> 0.1]   ### 269 variables remaining
xtrain1 = xtrain[,ind]
xtest1  = xtest[, ind]
dim(xtrain1); dim(xtest1)
[1] 60000   269
[1] 10000   269


   ##################################################
   ##### logistic regression fit and prediction  #####
   ##################################################
data_train = data.frame(Y=ytrain, xtrain1)
fit.glim = glm(Y~., data=data_train, family="binomial")  #fitting the model
sum(abs(fit.glim$coef) > 0.01)                           #eff no. of para= 268
logit = predict(fit.glim, newdata=data.frame(xtest1))     ##prediction logit
classification = (logit > 0)                             ##classification
mean(ytest != classification)                            #compute misclassification rat
[1] 0.0214

  ####################################
  #########  Lasso fitting ###########
  ####################################
library(glmnet)
set.seed(1000)                         #fixed random seed
```

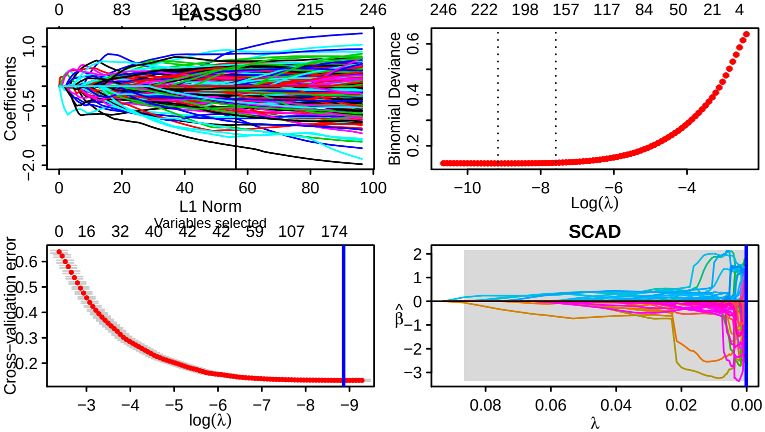
```
fit.lasso <- cv.glmnet(xtrain1, ytrain, family="binomial", nfolds=5, alpha=1)
  ##fit.cvglm1$lambda.min                 # the selected lambda
lambda = fit.lasso$lambda.1se           # lambda at 1 se
beta.lasso <- coef(fit.lasso, s=lambda) ###coef at 1se
sum(abs(beta.lasso) > 0.01)             # Number of variables selected.
[1] 168
logit2 = predict(fit.lasso, newx=xtest1, s=lambda)        ##predict
classification = (logit2  > 0)                           ##classification
mean(ytest != classification)                            #misclassification rate
[1] 0.0214

pdf("MNIST.pdf", width=4.6, height=2.6, pointsize=8)
par(mfrow = c(2,2), mar=c(5,5,3,1)+0.1, mex=0.5)
plot(fit.lasso$glmnet.fit); title('LASSO')               #Lasso solution path
abline(v=sum(abs(beta.lasso[-1])))                       #place where solution is selec
plot(fit.lasso)                      #Estimated MSE


  #####################################
  ##########  SCAD fitting ###########
  #####################################
library('ncvreg')                        #loading the library for use
fit.SCAD <- cv.ncvreg(xtrain1, ytrain, family="binomial", nfolds=5, penalty="SCAD")   #
beta.SCAD <- coef(fit.SCAD)                #fitted coefficients
sum(abs(beta.SCAD) > 0.01)               # Number of variables selected=173
logit3 = predict(fit.SCAD, X=xtest1)                #prediction at new data
classification = (logit3  > 0)                      #classification
```

```
mean(ytest != classification)                      #misclassification rate
[1] 0.0218

plot(fit.SCAD)
abline(v=log(fit.SCAD$lambda.min),lwd=2,col=4)
fit.SCADpath <-  ncvreg(xtrain1, ytrain, family="binomial", nfolds=5, penalty="SCAD")
plot(fit.SCADpath, main="SCAD")                ### solution path
abline(v=fit.SCAD$lambda.min,lwd=2,col=4)
dev.off()                                      ##close the current device

fit.SCAD2 = ncvreg(xtrain1, ytrain, family="binomial", nfolds=5,
 penalty="SCAD", lambda = 1.5*fit.SCAD$lambda.min)  #1.5*optimal choice
beta.SCAD2 <- coef(fit.SCAD2)                #fitted coefficients
sum(abs(beta.SCAD2) > 0.01)                  # Number of variables selected=186
logit3 = predict(fit.SCAD2, X=xtest1)                 #prediction at new data
classification = (logit3  > 0)                     #classification
mean(ytest != classification)                      #misclassification rate
```

Summary: After choosing variables with variance > 0.1, we end up with 269 variables. The logistic regression gives a misclassification rates of 0.0214 and effective uses 268 variables.

Lasso gives a misclassification rate of 0.0214 and uses effectively 168 variables.

SCAD gives a misclassification rate of 0.0210 and uses effectively 210 variables. If we choose 1.5 times of the optimal lambda, SCAD gives a misclassification rate of 0.0211 and uses effectively 186 variables.



★top panel: Lasso fit ★bottom pane: SCAD fit

3.5 One-Step Estimator

Fan, Xue and Zou (2014). Strong oracle optimality of folded concave penalized estimation. (§5.9.2)

Description of main results *(sec 5.9.2)*

- **LLA**: Compute $\widehat{\beta}^{(m)} = \operatorname{argmin}_\beta \; \ell_n(\beta) + \sum_j \widehat{w}_j^{(m-1)} \cdot |\beta_j|$.
  Update $\widehat{w}_j^{(m)} = P'_\lambda(|\widehat{\beta}_j^{(m)}|)$.

★ 1. the problem is **localizable**

2. the oracle estimator is **well behaved**,

the one-step LLA $(m = 1)$ gives the oracle estimator.

★ Once the oracle estimator is obtained, the LLA algorithm **converges**:

next iteration produces the same estimator.

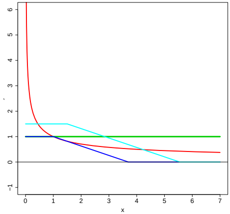
Folded concave penalty

Folded concave penalty: $P_\lambda(|t|)$ on $t \in \mathbb{R}$ satisfying

(i) increasing, differentiable and concave in $t \in [0, \infty)$

(ii) $P'_\lambda(t) \geq a_1 \lambda$ for $t \in (0, a_2\lambda]$

(iii) $P'_\lambda(t) = 0$ for $t \in [a\lambda, \infty)$ for a constant $a > a_2$



Remark: $a_1 = a_2 = 1$ for SCAD and $a_1 = 1 - a^{-1}$, $a_2 = 1$ for MCP

One-step LLA estimator

Theorem 3.4: Suppose that $\|\beta^\star_{\mathcal{A}}\|_{\min} > (a+1)\lambda$. Under event

$$\mathcal{E}_1 = \underbrace{\left\{\|\widehat{\beta}^{initial} - \beta^\star\|_{\max} \le a_2\lambda\right\}}_{\textbf{localizable}} \cap \underbrace{\left\{\|\nabla_{\mathcal{A}^c}\ell_n(\widehat{\beta}^{oracle})\|_{\max} < a_1\lambda\right\}}_{\textbf{oracle regularity}},$$

LLA **finds** the oracle estimator $\widehat{\beta}^{oracle}$ in **one iteration**:

$$L_n(\beta) = \ell_n(\beta) + \sum_j w_j|\beta_j|, \qquad w_j = P'_\lambda(|\widehat{\beta}_j^{initial}|)$$

★$\lambda = \sqrt{(\log p)/n}$.

★$E\nabla_{\mathcal{A}^c}\ell(\beta^\star) = 0$.

■OLS: $\nabla_{\mathcal{A}^c}\ell_n(\widehat{\beta}^{oracle}) = \mathbf{X}_{\mathcal{A}^c}(\mathbf{I}_n - \mathbf{P}_{\mathcal{A}})\varepsilon$.

Insights of LLA

1. Localizable & signal strength $\Longrightarrow |\widehat{\beta}_{\mathcal{A}}|_{\min} > a\lambda$.

$$\text{Folded concavity} \Longrightarrow \mathbf{w_j = 0, j \in \mathcal{A}}, \quad \mathbf{w_j > a_1\lambda, j \notin \mathcal{A}}.$$

2. One-step estimator: $\widehat{\beta}^{(1)} = \arg\min_\beta L_n(\beta)$, where $L_n(\beta) = \ell_n(\beta) + \sum_{j\in\mathcal{A}^c} w_j|\beta_j|$.

3. Convexity and score equation of oracle entails

$$\ell_n(\beta) \quad \geq \quad \underbrace{\ell_n(\widehat{\beta}^{oracle})}_{=\mathbf{L_n}(\widehat{\beta}^{\mathbf{oracle}})} + \sum_{\mathbf{j}\in\mathcal{A}^{\mathbf{c}}} \nabla_j \ell_n(\widehat{\beta}^{oracle})(\beta_j - \underbrace{\widehat{\beta}_j^{oracle}}_{=\mathbf{0}})$$

4. $L_n(\beta) - L_n(\widehat{\beta}^{oracle}) \geq \sum_{j\in\mathcal{A}^c}\{a_1\lambda - |\nabla_j \ell_n(\widehat{\beta}^{oracle})|\}\,|\beta_j| \geq \mathbf{0}$

Two-step LLA estimator

Theorem 3.5: Under the event

$$\mathcal{E}_2 = \underbrace{\left\{ \|\nabla_{\mathcal{A}^c} \ell_n(\widehat{\beta}^{oracle})\|_{\max} < a_1\lambda \right\} \cap \left\{ \|\widehat{\beta}^{oracle}_{\mathcal{A}}\|_{\min} > a\lambda \right\}}_{\text{oracle regularity}},$$

when the LLA algorithm finds $\widehat{\beta}^{oracle}$, the next step is still $\widehat{\beta}^{oracle}$.

- ★ Related to **uniform** convergence of the oracle estimator.
- ★ Oracle regularities have been verified for linear model, logistic regression, Gaussian covariance model *(Fan, Xue, Zou, 14)*.
- ★ LASSO or Danzig with a smaller penalty can be used as initial estimators.

3.5 Risk Properties

Analysis of Decomposable Regularization

Negahban, et al. (2012, stat. sci. 538-557), §5.9

Loh and Wainwright (2015, JMLR, 559-616) deals with folded concave penalties. §6.6

Preliminaries *(Sec 5.10)*

Problem: $\widehat{\theta} = \mathrm{argmin}\{L_n(\theta) + \lambda_n R(\theta)\}$

Restricted Strong Convexity: For all $\Delta \in \mathcal{C}$,

$$L_n(\theta^* + \boldsymbol{\Delta}) - L_n(\theta^*) - \langle \nabla L_n(\theta^*), \boldsymbol{\Delta} \rangle \quad \geq \quad \kappa_L \|\Delta\|^2 - \tau_L,$$

for some $\kappa_L > 0$ and $\tau_L > 0$.

Decomposability: For a given pair $\mathcal{M} \subset \overline{\mathcal{M}}$, we have

$R(\theta + \gamma) = R(\theta) + R(\gamma)$ for all $\theta \in \mathcal{M}$ and $\gamma \in \overline{\mathcal{M}}^{\perp}$.

Example: L_1-norm, $\overline{\mathcal{M}} = \mathcal{M} = \{\theta_j = 0, \forall j \notin S\}$.

Norms

Dual norm: $R^*(\mathbf{v}) = \sup_{\mathbf{u}\neq 0}\langle \mathbf{u}, \mathbf{v}\rangle / R(\mathbf{u})$.

Example: Dual of L_1-norm is L_∞.

Subspace compatibility constant: $\Psi(\mathcal{M}) = \sup_{u\in\mathcal{M}/\{0\}} R(\mathbf{u})/\|\mathbf{u}\|$

For L_1-norm, $\Psi(\mathcal{M}) = \sqrt{|\mathcal{M}|}$

Theorem 3.6. If $\lambda_n \geq 2R^*(\nabla L_n(\theta^*))$, then

★ $\|\widehat{\theta}_{\lambda_n} - \theta^*\|^2 \leq e_{err} + e_{app} + 2\lambda_n \tau_L^2/\kappa_L$

■ $e_{err} = 9\lambda_n^2 \Psi^2(\overline{\mathcal{M}})/\kappa_L^2$ and $e_{app} = 4\lambda_n R(\theta^*_{\mathcal{M}^\perp})/\kappa_L$.

★ $R(\widehat{\theta}_{\lambda_n} - \theta^*) \leq 4\Psi(\overline{\mathcal{M}})\|\widehat{\theta}_{\lambda_n} - \theta^*\| + 4R(\theta^*_{\mathcal{M}^\perp})$

Remarks

- Deterministic and nonasymptotic result
- When $\tau_L = 0$ and $\theta^*_{\mathcal{M}^\perp} = 0$, $\|\widehat{\theta}_{\lambda_n} - \theta^*\|^2 \le 9\lambda_n^2 \Psi^2(\overline{\mathcal{M}})/\kappa_L^2$.
- For L_1 penalty, we need $\lambda_n \ge 2\|\nabla L_n(\theta^*)\|_\infty$. **Best result**:

$$\|\widehat{\theta}_{\lambda_n} - \theta^*\|^2 \asymp s\|\nabla L_n(\theta^*)\|_\infty^2, \qquad s = |\mathcal{M}|$$

- Lasso requires $\lambda_n \ge 2\|n^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta^*)\|_\infty$. Thus,

$$\|\widehat{\theta}_{\lambda_n} - \beta^*\|^2 \asymp s\|n^{-1}\mathbf{X}^T\varepsilon\|_\infty^2 = O_p\left(\frac{s\log p}{n}\right)$$

- Second result for L_1 loss: $\|\widehat{\theta}_{\lambda_n} - \theta^*\|_1 \le 4\sqrt{s}\|\widehat{\theta}_{\lambda_n} - \theta^*\|$

Idea of Proofs

Lemma 1: Let $F(\mathbf{x})$ be convex w/ $F(\mathbf{0}) = 0$ and set $\mathcal{C}$ is a cone with vertex 0, i.e. if $\mathbf{x} \in \mathcal{C}$, then $a\mathbf{x} \in \mathcal{C}$ for any $a \geq 0$. If $F(\mathbf{x}) > 0$, $\forall \mathbf{x} \in \mathcal{C} \bigcap \{\|\mathbf{x}\| = \delta\}$, then $\widehat{\mathbf{x}} = \mathrm{argmin}_{x \in \mathcal{C}} F(x)$ must have $\|\widehat{\mathbf{x}}\| < \delta$.

Lemma 2: Let $\widehat{\boldsymbol{\Delta}} = \widehat{\theta} - \theta^*$. For convex $L(\beta)$, if $R^*(\nabla L(\theta^*)) \leq \frac{1}{2}\lambda_n$, then

$$R(\widehat{\boldsymbol{\Delta}}_{\bar{\mathcal{M}}^\perp}) \leq 3R(\widehat{\boldsymbol{\Delta}}_{\bar{\mathcal{M}}}) + 4R(\widehat{\theta}^*_{\mathcal{M}^\perp}).$$

■Let $F(\boldsymbol{\Delta}) = L_n(\theta^* + \boldsymbol{\Delta}) - L_n(\theta^*) + \lambda_n\{R(\theta^* + \boldsymbol{\Delta}) - R(\theta^*)\}$. Bound $F(\boldsymbol{\Delta})$ by a **quadratic** so that we can use Lemma 1 to bound $\|\widehat{\boldsymbol{\Delta}}\|$.

★Read proofs in Section 5.8.