P8160 Group Project Presentation

Optimization and Bootstrap

Xinlei Chen, Guojing Wu, Yujing Yao

Department of Biostatistics, Columbia University

April 19, 2019

1. Introduction
2. Project: Optimization
3. Project: Bootstrap

Introduction

Introduction of today's presentation

- **Group project 2:** Optimization algorithms on a breast cancer diagnosis dataset
    - Build a predictive model based on logistic regression to faciliate cancer diagnosis
    - Compare methods including Newton Raphson, Gradient Decent with general logistic regression and Pathwise Coordinate Descent with regularized logistic regression
- **Group project 3:** Bootstrapping on developing classification model
    - Build a predictive model based on regularized logistic regression to faciliate down syndrome diagnosis
    - Compare methods including Pathwise Coordinate Descent and smoothed bootstrap estimation

Project: Optimization

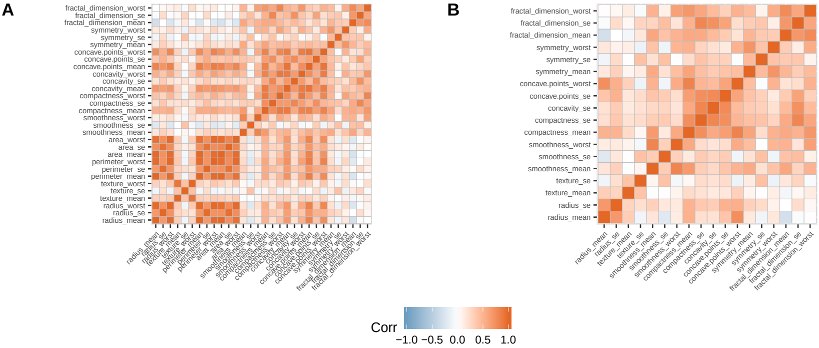
Background

Breast Cancer Data

- The data breast-cancer.csv 33 columns.
  - Covariate “ID” lables individual breast tissue images
  - Covariate “Diagnonsis” indentifies if the image is coming from cancer tissue or benign cases.
  - Mean, standard deviation and the largest values of the distributions of 10 features are computed for the cellnuclei for each case.
- Have 569 row
  - There are 357 benign and 212 malignant cases.

Multicollinearity of the Dataset

- **Variable Selection:** Reduce multicollinearity based on:
  - correlation coefficient ≤ 0.7
  - eigenvalue of correlation matrix ≥ 0.01



Method

Logistic Regression Model

- **Notations**
  - y: the vector of n response random variable
  - X: the $n \times (p+1)$ design matrix (X_i denote the ith row)
  - β: the $(p+1) \times 1$ coefficient vector
- **Objective function**: maximize log-likelihood function

$$\max_{\beta \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \{y_i(X_i\beta) - \log(1 + \exp(X_i\beta))\}$$

- Gradient: $\nabla l(\beta) = X^T(y - p)$, Hessian: $\nabla^2 l(\beta) = -X^T W X$ where $p = \frac{\exp(X\beta)}{1+\exp(X\beta)}$, $W = diag(p_i(1-p_i)), i = 1, \cdots, n$. The Hessian is negative definite.

Newton Raphson Algorithm

- *Newton Raphson with step-halving*

$$\beta_{i+1}(\gamma) = \beta_i - \gamma[\nabla^2 l(\beta_i)]^{-1}\nabla l(\beta_i)$$

- Algorithm
    - initilize the estimates denoted as β_0
    - use the principle of newton raphson to update the estimate, the algorithm of optimizing the step size is
        - Set $\gamma = 1$
        - If $f(\theta_{i+1}(1)) \geq f(\theta_i)$, then set $\theta_{i+1} = \theta_{i+1}(1)$
        - If $f(\theta_{i+1}(1)) \leq f(\theta_i)$, search for a value $\gamma \in (0,1)$ for which $f(\theta_{i+1}(\gamma)) \geq f(\theta_i)$, set $\theta_{i+1} = \theta_{i+1}(\gamma)$
    - stop searching until the convergences of the estimates
- *Gradient Descent:*

$$\beta_{i+1} = \beta_i + H_i \nabla f(\beta_i)$$

LASSO with Pathwise Coordinate Descent

- Object function: maximize the penalized log likelihood with some $\lambda \geq 0$:

$$\min_{\beta \in \mathbb{R}^{p+1}} \{ \frac{1}{2n} \sum_{i=1}^{n} (z_i - \sum_{j=0}^{p} x_{i,j}\beta_j)^2 + \lambda P(\beta)) \}$$

- Coordinate-wise descent with weighted update:

$$\tilde{\beta}_j^{lasso}(\lambda) \leftarrow \frac{S(\sum_{i=1}^{n} \omega_i x_{i,j}(y_i - \tilde{y}_i^{(-j)}), \lambda)}{\sum_{i=1}^{n} \omega_i x_{i,j}^2}$$

where $\tilde{y}_i^{(-j)} = \sum_{k \neq j} x_{i,k}\tilde{\beta}_k$ and $S(\hat{\beta}, \lambda) = sign(\hat{\beta})(|\hat{\beta}| - \lambda)_+$

Logistic-LASSO Model

- Object function:

$$\max_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \{y_i(X_i\beta) - \log(1 + \exp(X_i\beta))\} - \lambda \sum_{j=0}^{p} |\beta_j|$$

- Quardratic approximation to the negative log likelihood by taylor expansion

$$f(\beta) = -\frac{1}{2n} \sum_{i=1}^{n} w_i(z_i - \sum_{j=0}^{p} x_{i,j}\beta_j)^2 + C(\tilde{\beta})$$

where

$$z_i = \tilde{\beta}_0 + x_i^T\tilde{\beta} + \frac{y_i - \tilde{p}(x_i)}{\tilde{p}(x_i)(1 - \tilde{p}(x_i))}, \text{working response}$$

$$w_i = \tilde{p}(x_i)(1 - \tilde{p}(x_i)), \text{working weights}$$

Logistic-LASSO Model with Pathwise Coordinate Descent Algorithm

- Algorithm
  - outer loop: start with λ that all the coefficients are forced to be zerp, then decrement λ;
  - middle loop: update the quardratic $f(\beta)$ using the current estimates of parameters;
  - inner loop: run the coordinate descent algorithm on the penalized weighted least square problem.

Result

Estimation Path and Cross Validation for LASSO

Figure 2: A path of solutions with a sequence of descendi

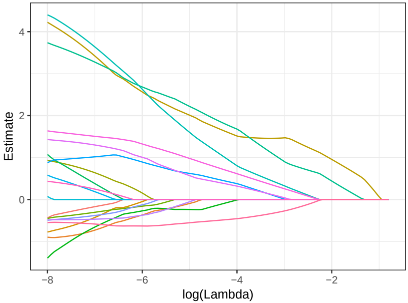
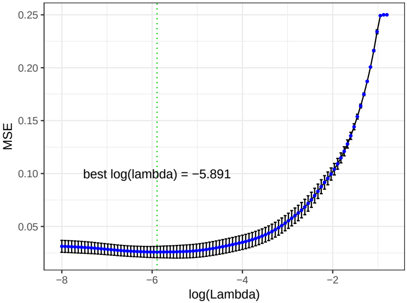


Figure 3: Lasso regression by 5 fold cross validation



Model Comparison: Prediction Performance

Table 1: The comparison of performance for estimation algorithms and models

	GLM package	Newton Raphson	Gradient Decent	Logistic Lasso	Lasso package
iteration times	NA	12	1001	100	NA
MSE	0.02	0.02	0.02	0.02	0.02

[a] Dataset: Breast Cancer Diagnosis

Model Comparison: estimation

	GLM package	Newton Raphson	Gradient Decent	Logistic Lasso	Lasso package
radius_mean	4.43	4.43	3.18	2.63	2.71
texture_mean	1.89	1.89	1.34	1.29	1.37
smoothness_mean	0.78	0.78	0.47	0.00	0.00
compactness_mean	-1.14	-1.14	-0.59	0.00	0.00
symmetry_mean	-0.63	-0.63	-0.44	-0.10	-0.14
fractal_dimension_mean	-0.66	-0.66	-0.72	-0.14	-0.21
radius_se	5.13	5.13	3.28	2.50	2.58
texture_se	0.59	0.59	0.46	0.00	0.00
smoothness_se	1.10	1.10	0.77	0.00	0.00
compactness_se	-0.80	-0.80	-0.68	-0.33	-0.38
concavity_se	1.24	1.24	0.88	0.08	0.19
concave.points_se	-1.11	-1.11	-0.80	0.00	0.00
symmetry_se	-0.53	-0.53	-0.39	-0.36	-0.42
fractal_dimension_se	-2.73	-2.73	-1.55	-0.25	-0.31
smoothness_worst	0.31	0.31	0.31	0.86	0.92
concave.points_worst	5.13	5.13	3.65	2.48	2.62
symmetry_worst	1.60	1.60	1.28	0.97	1.06
fractal_dimension_worst	2.19	2.19	1.41	0.00	0.00
intercept	-0.62	-0.62	-0.71	-0.63	-0.77

[a] Dataset: Breast Cancer Diagnosis

Conclusion

Conclusion and Discussion

- The results of our methods are consistent to the estimation from R's built-in packages
    - Newton-Raphson has the convincing estimation and it converged quickly
    - Gradient decent method showed similar estimation as Newton-Raphson method but it was less efficient
    - For Pathwise Coordinate descent with LASSO logistic, according to the result of 5 fold cross validation and estimation result, the λ with the lowest MSE and it shrunk six parameters to zero, which is comparable to the result by R's built-in packages.
- Prediction capability of logistic regression and penalized logistic regression are similar

Project: Bootstrap

Background

Down Syndrome Data

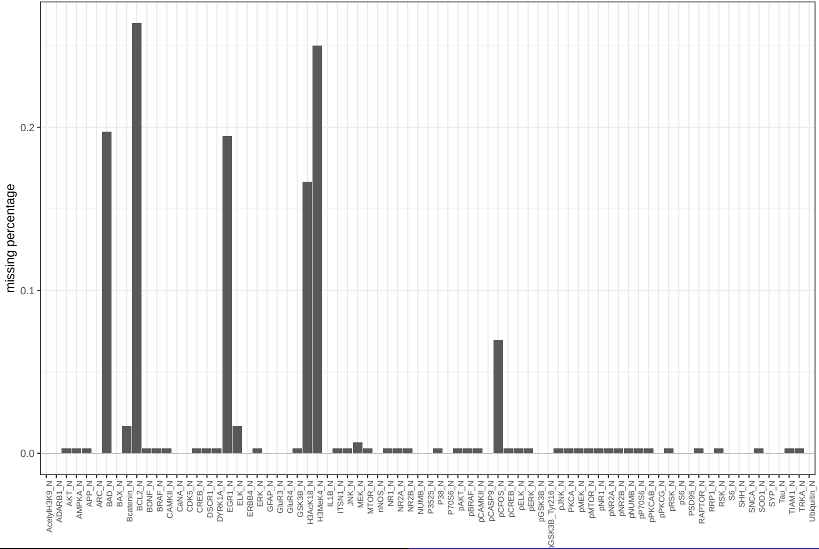
The data Down.csv consists of the expression levels of 77 proteins/protein modifications that produced detectable signals in the nuclear fraction of cortex. It has 1080 rows and 79 columns. The first column MouseID identifies individual mice; The column 2-78 are values of expression levels of 77 proteins. Column 79 indicates whether the mouse is a control or has Down syndrome. The goal is to develop classification model based on the proteins expression levels.

Missingness

- **Variable Selection:** Delete variables with high missing rate ($\geq 15\%$)

Also due to the intrinsic correlation between individual proteins, it's impossible to apply normal regression methods to this dataset because of sigularity propblem. Instead, we choose regularized methods, LASSO, to be more specific.

Missingness



Methods

Pathwise Coordinate Descent with Regularized Logistic Regression

- Object: maximize the penalized log likelihood:

$$\max_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \{y_i(X_i\beta) - \log(1 + \exp(X_i\beta))\} - \lambda \sum_{j=0}^{p} |\beta_j|$$

for some $\lambda \geq 0$

- Coordinate-wise descent with weighted update:

$$\tilde{\beta}_j^{lasso}(\lambda) \leftarrow \frac{S(\sum_{i=1}^{n} \omega_i x_{i,j}(y_i - \tilde{y}_i^{(-j)}), \lambda)}{\sum_{i=1}^{n} \omega_i x_{i,j}^2}$$

where $\tilde{y}_i^{(-j)} = \sum_{k \neq j} x_{i,k} \tilde{\beta}_k$ and $S(\hat{\beta}, \lambda) = sign(\hat{\beta})(|\hat{\beta}| - \lambda)_+$

Smoothed Bootstrap Estimation and Inference

- First we need to prepare a couple of candidate models
- for each bootstrap in bootstrap with B times, select the best model and get estimates for the coefficient denoted as $t(y^*)$
- smooth $\hat{\mu} = t(y)$ by averaging over the bootstrap replications, defining

$$\tilde{\mu} = s(y) = \frac{1}{B}\sum_{i=1}^{B} t(y^*)$$

Smoothed Bootstrap Estimation and Inference

And in addition to the percentile confidence interval, the nonparametric delta-method estimate of standard deviation for $s(y)$ in the nonideal case is:

$$\tilde{sd}_B = [\sum_{i=1}^{n} \hat{cov}_j^2]^{1/2}$$

where

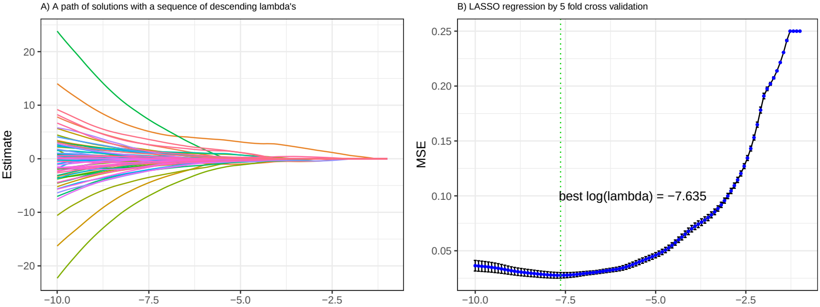
$$\hat{cov}_j = \sum_{i=1}^{B} (Y^*_{ij} - Y^*_{.j})(t^*_i - t^*_{.})/B$$

with $Y^*_{.j} = \sum_{i=1}^{B} Y^*_{ij}/B$ and $t^*_{.} = \sum_{i=1}^{B} t^*_i/B = s(y)$.

Result

Pathwise Coordinate Descent Logistic-LASSO

The left-hand side plot shows us that as the λ increases, all the variable estimates of parameters shrink accordingly since we penalize all the parameters. When $\lambda = 0$, the result is the same as least square method and when λ is too large, all the estimates of parameters shrink to 0. The right-hand side plot shows us the cross validation result for choosing the best λ.



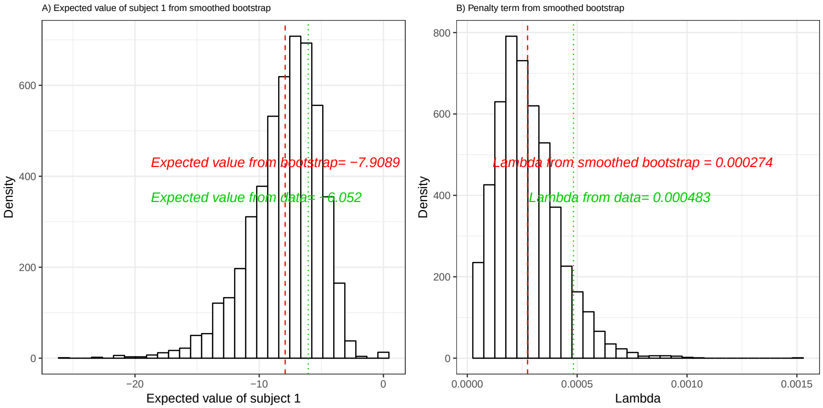
Model Selection Based on Smooth Bootstrap Estimation for Logistic-LASSO

algorithm:

- bootstrap data from the original dataset
- do cross validation and select the best λ_i^* for each repetition
- calculate average $\lambda^* = \frac{1}{B}\sum_{i=1}^{B}\lambda_i^*$

We can see the discrepancy between results of PCD-LASSO and smooth bootstrap estimation for Logistic LASSO both in prediction and finding the best λ, the results of PCD-LASSO is deviated from the center of empirical distribution.

Lambda Selection Based on Smooth Bootstrap Estimation for Logistic LASSO



Cross Validation for Model Prediction Comparison

We used 10 fold cross-validation to compare two different models, one is with λ selected from data, the other is selected from the SBE. Table 1 shows us that while the Cross Validation MSE are similar between these two methods, smooth bootstrap estimation provides a more accurate classification result.

Table 2: The comparison of performance for two models

	Misclassification rate	Mean squred error
Penalty chosen by data	0.0353	0.0229
Penalty selected from smoothed bootstrap	0.0335	0.0216

[a] Dataset: Proteins expression levels of Down syndrome

Significant Random Variable Selection from Smooth Bootstrap Estimation

Table 2 & 3 provide the full results of smooth bootstrap estimation for logistic LASSO. Our identification criterions here are:

1. the chosen probability greater than 96%
2. smooth bootstrap estimation confidence interval excludes zero.

Based on that, we got 27 proteins that meets these two criterions (Table 4).

Significant Proteins with Bootstap time=5000

	origin	prob	coef	sd	lower	upper	lower.new	upper.new
ITSN1_N	7.9040	1.00	9.6725	2.1862	4.4201	16.9149	5.3875	13.9575
pELK_N	-2.1788	0.99	-2.7419	1.1203	-6.1524	-0.4194	-4.9377	-0.5461
pNR1_N	-2.4536	0.96	-2.6193	1.2315	-5.9502	0.0000	-5.0330	-0.2056
pRSK_N	-2.4102	1.00	-2.8974	0.7656	-5.2917	-1.1120	-4.3980	-1.3968
AKT_N	2.7674	1.00	3.3722	0.8799	1.2074	6.0927	1.6476	5.0968
BRAF_N	-4.8560	1.00	-5.6899	1.7241	-10.9610	-1.8681	-9.0691	-2.3107
CAMKII_N	-1.5901	0.99	-2.3111	0.8889	-4.9441	-0.3901	-4.0533	-0.5689
CREB_N	-1.2469	0.98	-1.3510	0.5539	-2.9474	-0.0163	-2.4366	-0.2654
ELK_N	-3.6872	1.00	-4.6751	1.0545	-8.0746	-2.1847	-6.7419	-2.6083
ERK_N	-7.4243	1.00	-8.7471	1.6856	-14.5871	-4.7828	-12.0509	-5.4433
MEK_N	1.3308	0.98	1.6328	0.7152	0.0194	3.6599	0.2310	3.0346
TRKA_N	3.7756	1.00	5.5845	2.1337	2.0454	11.9326	1.4024	9.7666
APP_N	5.3514	1.00	7.8719	1.4402	5.0187	13.0828	5.0491	10.6947
MTOR_N	-2.3190	0.99	-2.8751	0.9748	-5.7528	-0.7263	-4.7857	-0.9645
DSCR1_N	1.2781	0.98	1.5514	0.6235	0.0022	3.3412	0.3293	2.7735
RAPTOR_N	-1.7634	0.96	-2.1489	1.0112	-4.9061	0.0000	-4.1309	-0.1669
TIAM1_N	2.7743	1.00	3.4095	1.0557	1.1476	6.4593	1.3403	5.4787
NUMB_N	1.4306	0.98	1.8409	0.8104	0.0385	4.1741	0.2525	3.4293
ERBB4_N	1.4902	1.00	2.0181	0.5463	0.7995	3.6211	0.9474	3.0888
Tau_N	1.5522	1.00	2.2831	0.5607	1.0326	4.3017	1.1841	3.3821
GluR3_N	-1.3348	1.00	-1.7384	0.4746	-3.2161	-0.7186	-2.6686	-0.8082
IL1B_N	-1.4177	0.99	-1.9549	0.6874	-4.1387	-0.4795	-3.3022	-0.6076
P3525_N	1.0465	0.97	1.2122	0.5624	0.0000	2.7394	0.1099	2.3145
Ubiquitin_N	0.9464	0.97	1.3435	0.6419	0.0000	3.1608	0.0854	2.6016
SHH_N	-1.5405	1.00	-1.9631	0.5122	-3.6381	-0.8420	-2.9670	-0.9592
SYP_N	-0.9364	0.99	-1.2874	0.4838	-2.6677	-0.1815	-2.2356	-0.3392
CaNA_N	1.7695	0.99	2.3003	0.7909	0.4918	4.8228	0.7501	3.8505

[a] origin: estimation from PCD-LASSO

[b] prob: chosen probability from bootstrap, coef: estimation from SBE

[c] sd: nonparamatric delta-method estimate of standard deviation

[d] lower, upper: quantile CI; lower.new, upper.new: CI from nonparamatric delta-method estimate

Conclusion