Generalized Linear Models and the Lasso

GLM

$$Y_1, \ldots, Y_n \text{ independent},$$

$$\underbrace{g}_{\text{link fct.}} \; \mathbb{E}[Y_i | X_i = x] = \mu + \sum_{j=1}^{p} \beta_j x^{(j)}$$

example: binary classification

$$Y_i \sim \text{ Bernoulli}(\pi(X_i))(\in \{0,1\}),$$

$$\log(\frac{\pi(x)}{1-\pi(x)}) = \mu + \sum_{j=1}^{p} \beta_j x^{(j)} \;\; \text{(logit link)}$$

(note that $\pi(x) = \mathbb{E}[Y | X = x]$)

conditional probability (density) of $Y|X = x$ is of the form

$$p(y|x) = p_{f(x)}(y) = p_{\mu,\beta}(y|x)$$

$\leadsto$ negative log-likelihood

$$-\sum_{i=1}^{n} \log(p_{\mu,\beta}(Y_i|X_i)) = n^{-1} \sum_{i=1}^{n} \underbrace{\rho_{\mu,\beta}}_{\text{loss fct.}} (X_i, Y_i)$$

Lasso estimator:

$$\hat{\mu}, \hat{\beta} = \operatorname{argmin}_{\mu,\beta}(n^{-1} \sum_{i=1}^{n} \rho_{\mu,\beta}(X_i, Y_i) + \lambda\|\beta\|_1)$$

Note: no penalty for intercept term

many standard models yield a loss function ($= -\log(p(y|x))$) which is convex in μ, β
$\rightsquigarrow$ Lasso can be computed efficiently

example: binary classification
$Y_i \sim$ Bernoulli($\pi(x)$) with

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \mu + \sum_{j=1}^{p} \beta_j x^{(j)}.$$

negative log-likelihood equals

$$-\sum_{i=1}^{n} \log(p_{\mu,\beta}(Y_i|X_i)) = \sum_{i=1}^{n} \left(-Y_i f_{\mu,\beta}(X_i) + \log(1+\exp(f_{\mu,\beta}(X_i)))\right),$$

and the corresponding loss function is

$$\rho_{\mu,\beta}(x,y) = -y(\mu + \sum_{j=1}^{p} \beta_j x^{(j)}) + \log(1+\exp(\mu + \sum_{j=1}^{p} \beta_j x^{(j)})).$$

in terms of the linear predictor $f(x) = \mu + \sum_{j=1}^{p} \beta_j x^{(j)}$
$\leadsto$ loss function equals

$$\rho(x, y) = \rho(f(x), y) = -yf + \log(1 + \exp(f)),$$

where $f(x) = f$

this is a convex function in f since

- the first term is linear
- the second term has positive second derivative
- and the sum of convex functions is convex

furthermore: $f = f_{\mu,\beta}(x) = \mu + \sum_{j=1}^{p} \beta_j x^{(j)}$ is linear $\leadsto$

$$\rho_{\mu,\beta}(x, y) = h_y(f_{\mu,\beta}(x))$$

is convex in μ, β
as a composition of a convex function $h_y(\cdot)$ (convex for all y)
and a linear function

The Group Lasso (Yuan & Lin, 2006)

high-dimensional parameter vector is structured into q groups or partitions (known a-priori):

$$\mathcal{G}_1, \ldots, \mathcal{G}_q \subseteq \{1, \ldots, p\}, \text{ disjoint and } \cup_g \mathcal{G}_g = \{1, \ldots, p\}$$

corresponding coefficients: $\beta_{\mathcal{G}} = \{\beta_j;\ j \in \mathcal{G}\}$

Example: categorical covariates

$X^{(1)}, \ldots, X^{(p)}$ are factors (categorical variables)
each with 4 levels (e.g. “letters” from DNA)

for encoding a main effect: 3 parameters
for encoding a first-order interaction: 9 parameters
and so on ...

parameterization (e.g. sum contrasts) is structured as follows:

- intercept: no penalty
- main effect of $X^{(1)}$: group $\mathcal{G}_1$ with $df = 3$
- main effect of $X^{(2)}$: group $\mathcal{G}_2$ with $df = 3$
- ...
- first-order interaction of $X^{(1)}$ and $X^{(2)}$: $\mathcal{G}_{p+1}$ with $df = 9$
- ...

often, we want sparsity on the group-level
either all parameters of an effect are zero or not

often, we want sparsity on the group-level
either all parameters of an effect are zero or not

this can be achieved with the Group-Lasso penalty

$$\lambda \sum_{g=1}^{q} s(df_g) \underbrace{\|\beta_{\mathcal{G}_g}\|_2}_{\sqrt{\|\cdot\|_2^2}}$$

typically $s(df_{\mathcal{G}_g}) = \sqrt{df_{\mathcal{G}_g}}$ so that $s(df_{\mathcal{G}_g})\|\beta_{\mathcal{G}_g}\|_2 = O(df_g)$

properties of Group-Lasso penalty

- for group-sizes $|\mathcal{G}_g| \equiv 1 \leadsto$ standard Lasso-penalty
- convex penalty $\leadsto$ convex optimization for standard likelihoods (exponential family models)
- either $(\hat{\beta}_{\mathcal{G}}(\lambda))_j = 0$ or $\neq 0$ for all $j \in \mathcal{G}$
- penalty is invariant under orthonormal transformation
  e.g. invariant when requiring orthonormal parameterization for factors

Some aspects from theory

"again":

- optimal prediction and estimation (oracle inequality)
- group screening: $\hat{S} \supseteq \underbrace{S_0}_{\text{set of active groups}}$ with high prob.

Computation and KKT

criterion function

$$Q_\lambda(\beta) = n^{-1} \sum_{i=1}^{n} \underbrace{\rho_\beta(x_i, Y_i)}_{\text{loss fct.}} + \lambda \sum_{g=1}^{G} s(df_g) \|\beta_g\|_2,$$

loss function $\rho_\beta(.,.)$ convex in β

KKT conditions:

$$\nabla\rho(\hat{\beta})_g + \lambda s(df_g) \frac{\hat{\beta}_{\mathcal{G}_g}}{\|\hat{\beta}_{\mathcal{G}_g}\|_2} = 0 \text{ if } \hat{\beta}_{\mathcal{G}_g} \neq 0 \text{ (not the 0-vector)},$$

$$\|\nabla\rho(\hat{\beta})_g\|_2 \leq \lambda s(df_g) \text{ if } \hat{\beta}_{\mathcal{G}_g} \equiv 0.$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1, \beta_2 = \beta_2^{(0)}, \ldots, \beta_j = \beta_j^{(0)}, \ldots, \beta_p = \beta_p^{(0)})$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$) keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1 = \beta_1^{(1)}, \beta_2, \ldots, \beta_j = \beta_j^{(0)}, \ldots, \beta_p = \beta_p^{(0)})$$

$$\uparrow$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1 = \beta_1^{(1)}, \beta_2 = \beta_2^{(1)}, \ldots, \underset{\uparrow}{\beta_j}, \ldots, \beta_p = \beta_p^{(0)})$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1 = \beta_1^{(1)}, \beta_2 = \beta_2^{(1)}, \ldots, \beta_j = \beta_j^{(1)}, \ldots, \underset{\uparrow}{\beta_p})$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1, \beta_2 = \beta_2^{(1)}, \ldots, \beta_j = \beta_j^{(1)}, \ldots, \beta_p = \beta_p^{(1)})$$

$$\uparrow$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \dots, p, 1, 2, \dots$
  or $j = 1, \dots, q, 1, 2, \dots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1}, \beta_{\mathcal{G}_2} = \beta_{\mathcal{G}_2}^{(0)}, \dots, \beta_{\mathcal{G}_j} = \beta_{\mathcal{G}_j}^{(0)}, \dots, \beta_{\mathcal{G}_q} = \beta_{\mathcal{G}_q}^{(0)})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1} = \beta_{\mathcal{G}1}^{(1)}, \beta_{\mathcal{G}_2}, \ldots, \beta_{\mathcal{G}_j} = \beta_{\mathcal{G}_j}^{(0)}, \ldots, \beta_{\mathcal{G}_q} = \beta_{\mathcal{G}_q}^{(0)})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \dots, p, 1, 2, \dots$
  or $j = 1, \dots, q, 1, 2, \dots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1} = \beta_{\mathcal{G}1}^{(1)}, \beta_{\mathcal{G}_2} = \beta_{\mathcal{G}_2}^{(1)}, \dots, \beta_{\mathcal{G}_j}, \dots, \beta_{\mathcal{G}_q} = \beta_{\mathcal{G}_q}^{(0)})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1} = \beta^{(1)}_{\mathcal{G}1}, \beta_{\mathcal{G}_2} = \beta^{(1)}_{\mathcal{G}_2}, \ldots, \beta_{\mathcal{G}_j} = \beta^{(1)}_{\mathcal{G}_j}, \ldots, \beta_{\mathcal{G}_q})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1}, \beta_{\mathcal{G}_2} = \beta_{\mathcal{G}_2}^{(1)}, \ldots, \beta_{\mathcal{G}_j} = \beta_{\mathcal{G}_j}^{(1)}, \ldots, \beta_{\mathcal{G}_q} = \beta_{\mathcal{G}_q}^{(1)})$

$$\uparrow$$

for Gaussian log-likelihood (squared error loss):
blockwise up-dates are easy and closed-form solutions exist (use KKT)

for other loss functions (e.g. logistic loss):
blockwise up-dates: no closed-form solution

$\rightsquigarrow$

strategy which is fast: improve every coordinate/group numerically, but not until numerical convergence
(by using quadratic approximation of log-likelihood function for improving/optimization of a single block)

and further tricks... (still allowing provable numerical convergence)

How fast?

logistic case: $p = 10^6$, $n = 100$
group-size = 20, sparsity: 2 active groups = 40 parameters

for 10 different λ-values
CPU using `grplasso`: 203.16 seconds $\approx$ 3.5 minutes
(dual core processor with 2.6 GHz and 32 GB RAM)

we can easily deal today with predictors in the Mega's
i.e. $p \approx 10^6 - 10^7$

How fast?

logistic case: $p = 10^6$, $n = 100$
group-size = 20, sparsity: 2 active groups = 40 parameters

for 10 different λ-values
CPU using `grplasso`: 203.16 seconds $\approx$ 3.5 minutes
(dual core processor with 2.6 GHz and 32 GB RAM)

we can easily deal today with predictors in the Mega's
i.e. $p \approx 10^6 - 10^7$

DNA splice site detection: (mainly) prediction problem

DNA sequence

$$\ldots ACGGC \ldots \quad \underbrace{E\ E\ E \quad \underbrace{GC}_{\text{potential donor site}} \quad I\ I\ I\ I}_{\text{3 positions exon } GC \text{ 4 positions intron}} \quad \ldots AAC \ldots$$

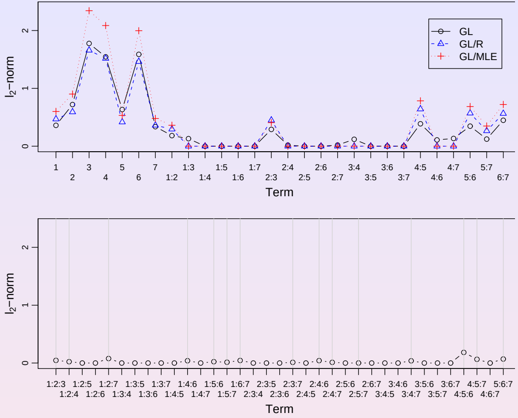
response $Y \in \{0, 1\}$: splice or non-splice site
predictor variables: 7 factors each having 4 levels
(full dimension: $4^7 = 16'384$)
data:

training:	$5'610$ true splice sites
	$5'610$ non-splice sites
	plus an unbalanced validation set
test data:	$4'208$ true splice sites
	$89'717$ non-splice sites

logistic regression:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \text{ main effects} + \text{first order interactions} + \ldots$$

use the Group-Lasso which selects whole terms



- mainly neighboring DNA positions show interactions (has been “known” and “debated”)
- no interaction among exons and introns (with Group Lasso method)
- no second-order interactions (with Group Lasso method)

predictive power:
competitive with “state to the art” maximum entropy modeling from Yeo and Burge (2004)

correlation between true and predicted class

Logistic Group Lasso	0.6593
max. entropy (Yeo and Burge)	0.6589

our model (not necessarily the method/algorithm) is simple and has clear interpretation a slight generalization: generalized Group Lasso penalty

$$\lambda \sum_{j=1}^{q} s(df_j) \sqrt{\beta_{\mathcal{G}_j}^T A_j \beta_{\mathcal{G}_j}},$$

where A_j are positive definite $df_j \times df_j$ matrices
A_j positive definite $\leadsto$ can re-parameterize:

$$\tilde{\beta}_{\mathcal{G}_j} = A_j^{1/2} \beta_{\mathcal{G}_j},$$

and hence

$$\lambda \sum_{j=1}^{q} s(df_j) \|\tilde{\beta}_{\mathcal{G}_j}\|_2$$

matrix $A_j^{1/2}$: use e.g. the Choleski decomposition
of course, we also need to re-parameterize the (generalized) linear model part