An algorithm for high-dimensional generalized linear mixed models using $\ell_1$-penalization

Jürg Schelldorfer

joint work with Peter Bühlmann

encouraged by Stephan Dlugosz

FOR916 Workshop on "Mathematical Statistics meets Econometrics"

June 9, 2011
data set of Stephan Dlugosz:

administrative data set about employment:

Binary response variable $Y \in \{\text{employed}, \text{unemployed}\}$

covariates $X$: income, sex, age group, employment duration,....

quarterly results of $(Y, X)$ of many workers over several years
response variable from the exponential family
continuous covariates
grouped observations (think of longitudinal data, repeated measures data)

**Goal:**
Performing variable selection in the setup where AIC, BIC, cAIC, mAIC, ... are computationally infeasible (i.e. \( n \approx p, n \ll p \))
## Overview

<table>
<thead>
<tr>
<th>Models</th>
<th>$n &gt; p$</th>
<th>$n \ll p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generalized Linear Models (GLMs)</td>
<td>MLE (IRLS)</td>
<td>Lasso (R: glmnet)</td>
</tr>
<tr>
<td>Generalized Linear Mixed Models (GLMMs)</td>
<td>MLE (R: glmer)</td>
<td>?</td>
</tr>
</tbody>
</table>

$n$: number of observations  
$p$: number of variables
Generalized Linear Model (GLM)

For $n$ observations $(y_i, x_i^T)$

- $(y_i, x_i^T)$ are independent for $i = 1, \ldots, n$
- $y_i$ has a density of the form

$$
\exp \left\{ \phi^{-1} \left( y_i \xi_i - b(\xi_i) \right) + c(y_i, \phi) \right\} \text{ with } \mu_i = \mathbb{E}[y_i]
$$

- $g(\mu) = \eta$ with $\eta = X\beta$

Then estimate $\beta$ by

$$
\hat{\beta}_{MLE} = \arg\min_\beta - \ell(\beta)
$$
For $n \ll p$ we should not use the MLE. Use the Lasso (Tibshirani, 1996)

$$\hat{\beta}(\lambda) = \arg\min_{\beta} -\ell(\beta) + \lambda \|\beta\|_1, \quad \lambda > 0$$

with the following properties:

- The Lasso does variable selection (i.e. some coefficients are set exactly to zero)
- Convex optimization problem, which can be solved efficiently
Generalized Linear Mixed Model (GLMM)

\[ g = 1, \ldots, N \] independent groups/clusters/subjects
\[ j = 1, \ldots, n_g \] observations for group/cluster/subject \( g \)
\[ n = \sum_{g=1}^{N} n_g \] total number of observations

\( y \): \( n \)-dim response variable
\( b \): \( q \)-dim (correlated) random effects

\( \beta \in \mathbb{R}^p \) fixed-effects parameters
\( \theta \in \mathbb{R}^L \) covariance parameters
\( \phi \) dispersion parameter

\( X : n \times p \) model matrix for \( \beta \)
\( Z : n \times q \) model matrix for \( b \)
\( \Sigma_\theta : q \times q \) covariance matrix, determined by \( \theta \)
Model Assumptions:
- $y_i | b$ are independent for $i = 1, \ldots, n$
- $y_i | b$ has a density of the form
  \[
  \exp \left\{ \phi^{-1} \left( y_i \xi_i - b(\xi_i) \right) + c(y_i, \phi) \right\} \text{ with } \mu_i = \mathbb{E}[y_i | b]
  \]
- $g(\mu) = \eta$ with $\eta = X\beta + Zb$
- $b \sim \mathcal{N}_q(0, \Sigma_\theta)$ with $\Sigma_\theta \geq 0$ for $\theta \in \mathbb{R}^L$

\[
(\hat{\beta}, \hat{\theta}, \hat{\phi})_{\text{MLE}} = \arg\min_{\beta, \theta, \phi} - \log L(\beta, \theta, \phi)
\]
### Recap

<table>
<thead>
<tr>
<th></th>
<th>$n &gt; p$</th>
<th>$n \ll p$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generalized Linear</strong></td>
<td>MLE [IRLS] ✓</td>
<td>Lasso [R: glmnet] ✓</td>
</tr>
<tr>
<td><strong>Models (GLMs)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>!</td>
</tr>
<tr>
<td><strong>Mixed Models (GLMMs)</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Additionally to a GLMM, assume

1. $n = \sum_{i=1}^{N} n_g \ll p$
2. the true $\beta_0$ is sparse
3. $L$ small

**Aim:** Estimate $\beta, \theta, \phi$ and predict $b$
The GLMMLasso estimator

**Key Idea 1: Lasso-type penalty**

Estimate the parameters \((\beta, \theta, \phi)\) by minimizing

\[
Q_\lambda(\beta, \theta, \phi) := -2 \log L(\beta, \theta, \phi) + \lambda \| \beta \|_1,
\]

\[
(\hat{\beta}, \hat{\theta}, \hat{\phi}) := \text{argmin}_{\beta, \theta, \phi} Q_\lambda(\beta, \theta, \phi).
\]

Remark: In general, \(L(\beta, \theta, \phi)\) cannot be computed explicitly.
**Key Idea 2: Laplace approximation** to approximate the integrand of $L(\beta, \theta, \phi)$ by a quadratic function.

\[
I = \int_{\mathbb{R}^q} e^{-S(b)} \, db \approx (2\pi)^{q/2} |S''(\tilde{b})|^{-1/2} e^{-S(\tilde{b})}
\]

where $\tilde{b} = \text{argmin}_b S(b)$ is the mode of $-S(b)$.

Hence

\[
Q_\lambda(\beta, \theta, \phi) \sim \tilde{Q}^{LA}_\lambda(\beta, \theta, \phi)
\]
The GLMMLasso estimator is defined by

\[(\hat{\beta}, \hat{\theta}, \hat{\phi}) := \arg\min_{\beta, \theta, \phi} \tilde{Q}_\lambda^{LA}(\beta, \theta, \phi)\]

Remark: It is a non-convex optimization problem!
The GLMMLasso algorithm I

How to calculate

\[(\hat{\beta}, \hat{\theta}, \hat{\phi}) := \text{argmin}_{\beta, \theta, \phi} \tilde{Q}_{\lambda}^{LA}(\beta, \theta, \phi)\]?

**Key Idea 3:** coordinate-wise optimization with inexact line search, i.e. optimize \(\tilde{Q}_{\lambda}^{LA}\) w.r.t. one coordinate keeping all other coordinates fixed (Tseng and Yun, 2009):

- **Quadratic approximation** of the objective function
- calculate the **gradient**
- **Inexact line search** using the Armijo rule
The GLMMLasso algorithm II

GLMMLasso algorithm

(0) Choose a starting value \((\beta^{(0)}, \theta^{(0)}, \phi^{(0)})\).

Repeat for \(s = 1, 2, \ldots\)

(1) (Fixed-effects parameter optimization) For \(k = 1, \ldots, p\)
   a) (Laplace approximation)
      Calculate the Laplace approximation \(\tilde{Q}^{LA}_{\lambda}(., ., .)\)
   b) (Quadratic approximation and inexact line search)
      i) Approximate the second derivative by \(h_k^{(s)} > 0\).
      ii) Calculate the descent direction \(d_k^{(s)} \in \mathbb{R}\)
      iii) Choose a step size \(\alpha_k^{(s)} > 0\) such that there is a decrease in the objective function.

(2) (Covariance parameter optimization) For \(l = 1, \ldots, L\)

\[
\theta_l^{(s)} = \arg\min_{\theta_l} \tilde{Q}^{LA}_{\lambda}(., ., .)
\]

(3) (Dispersion parameter optimization)

\[
\phi^{(s)} = \arg\min_{\phi} \tilde{Q}^{LA}_{\lambda}(., ., .)
\]

until convergence.
Tools to speed up

Two ingredients which speed up the algorithm remarkably:

- **KeyIdea 4a**: regard $\hat{b}$ as fixed for the quadratic approximation w.r.t. $\beta_k$
- **KeyIdea 4b**: active-set algorithm cycle through the non-zero coefficients $\beta_k$, and only through all $p$ coefficients every $D$th iteration

This two ingredients make it feasible to calculate large data sets (i.e. $n = 400$ and $p = 4000$)!
The price to pay

Small additional bias in the parameter estimates, and similar variable selection properties.
This is ongoing work with Stephan Dlugosz on administrative data.
### Take-home message

<table>
<thead>
<tr>
<th></th>
<th>$n &gt; p$</th>
<th>$n \ll p$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generalized Linear Models (GLMs)</strong></td>
<td>MLE [IRLS]</td>
<td>Lasso [R: glmnet]</td>
</tr>
<tr>
<td><strong>Generalized Linear Mixed Models (GLMMs)</strong></td>
<td>MLE [R: glmer]</td>
<td>GLMMLasso KeyIdea 1-4</td>
</tr>
</tbody>
</table>
Thank you!

Questions?
P. Tseng and S. Yun; A Coordinate Gradient Descent Method for Nonsmooth Separable Minimization; Mathematical Programming (2009)


D. Bates; lme4: Mixed-effects modeling with R (to appear)