Procedia Computer Science

Volume 29, 2014, Pages 821–830

ICCS 2014. 14th International Conference on Computational Science

COFADMM: A Computational features selection with Alternating Direction Method of Multipliers

Mohammed El Anbari[1], Sidra Alam[2], and Halima Bensmail[1*]

[1] Qatar Computing Research Center
melanbari@qf.org.qa, hbensmail@qf.org.qa
[2] Carnegie Mellon University @ Qatar
sidra.m.alam@gmail.com

Abstract

Due to the explosion in size and complexity of Big Data, it is increasingly important to be able to solve problems with very large number of features. Classical feature selection procedures involves combinatorial optimization, with computational time increasing exponentially with the number of features. During the last decade, penalized regression has emerged as an attractive alternative for regularization and high dimensional feature selection problems. Alternating Direction Method of Multipliers (ADMM) optimization is suited for distributed convex optimization and distributed computing for big data. The purpose of this paper is to propose a broader algorithm COFADMM which combines the strength of convex penalized techniques in feature selection for big data and the power of the ADMM for optimization. We show that combining the ADMM algorithm with COFADMM can provide a path of solutions efficiently and quickly. COFADMM is easy to use, is available in C, Matlab upon request from the corresponding author.

Keywords: features selection, lasso, least angle regression algorithm, coordinate descent algorithm, ADMM Algorithm

1 Introduction

During Big data era: wiki, WSJ, white house, McKinsey report to name a few, many challenges are raised related to the methodology when dealing with the big number of variables, in the efficiency when we face data with large sample size and/or large number of variables, and in the memory when sample size is large and needs a distributed computing to solve it via MapReduce or Hadoop. In fact, ADMM has been proposed for a variety of machine learning problems of recent interest, including the Lasso, sparse logistic regression, basic pursuit, covariance matrix estimation, support vector machines among many others as an easy and elegant tool of optimization. For regression problems of type $\mathbf{y} = \mathbf{X}\beta + \varepsilon$, where $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_p)$ is an $n \times p$ matrix

*corresponding author.

of p feature vectors of dimension n, ε is a random error vector with $\mathbf{E}(\varepsilon) = 0$ and β is a vector of unknown parameters to be estimated, a great deal of attention has been given to the estimation and feature selection in high-dimensional linear regression models. Several important methods among others have been proposed. Most of these methods are based on penalized least squares, which perform estimation and feature selection in a continuous fashion by shrinking some of the regression coefficients towards zero and setting some of them to exactly zero. The most popular regularization approach is the least absolute shrinkage and selection operator called Lasso [10], which is based on the penalized least squares by the ℓ_1 penalty on the vector parameter. [5] developed an efficient algorithm called Lars (Least Angle Regression) to find the entire solution path for the Lasso. Despite its good properties, the Lasso has serious limitations namely (a) it selects at most n features if $p > n$; (b) produces significant bias towards 0 for large regression coefficients; (c)fails to do grouped selection and tends to select one feature from a group and ignore the others in the presence of highly correlated features; (d)ignores a structured data as in the case of highly ordered features.

Many methods have been proposed to deal with these limitations. The most popular among these methods is the elastic net [12], which uses both the Ridge and Lasso constraints. In the same spirit, [1] proposed a method called Oscar, which is based on the penalized least squares with a penalty function combining the ℓ_1 and the pairwise ℓ_∞ norms. Oscar forces some of the grouped coefficients to be identically equal, encouraging correlated features that have similar effect on the response to form clusters represented by the same coefficients. However, the computation of the Oscar estimates is based on a sequential quadratic programming algorithm which is slow for large p. In the same setting, [7] considered the Smooth-Lasso procedure (SLasso), a modification of the Fused-Lasso procedure [9], in which a second ℓ_1 Fused penalty is replaced by the smooth ℓ_2 norm penalty. From an algorithmic point of view, to find the solutions in [12], [6], [3] and [7], each of the corresponding optimization problem can be seen as a Lasso problem by introducing new observations, and then use Least Angle Regression algorithm (LARS) or coordinate-descent (Gauss-Seidel) algorithm. It is interesting to note that (i) for $p >> n$ the augmented data set has $p+n$ observations and p variables, which can slow the computation considerably; (ii) if the original design matrix is normalized, there is no guarantees the augmented design matrix will behave similarly, which can cause a loss of a part of the interpretation of the big data; and (iii) the coordinate-descent algorithm proceeds by "one at a time" philosophy, e.g. it minimizes the loss function of β_j while maintaining components $\{\beta_k, k \neq j\}$ fixed at their actual values, in this case we cannot develop Gauss-Seidel for a grouped variable selection problem. To overcome these limitations, we derive a unified alternating direction method of multipliers based algorithm (COFADMM) to handle Big Data features selection with lasso-type estimator. We propose a doubly regularized model with a general penalty term of the form:

$$(\mu/2)\beta^t \mathbf{Q}\beta + \lambda \sum_{j=1}^{p} \hat{\omega}_j |\beta_j|, \tag{1}$$

where $\lambda, \mu \geq 0$ are two tuning parameters, $\hat{\omega} = (\hat{\omega}_1, ..., \hat{\omega}_p)^t$ and $\mathbf{Q} = (q_{ij})_{1 \leq i,j \leq p}$ are weights associated with the ℓ_1 and ℓ_2 norms respectively, which are fixed in advance.

The advantage of our algorithm are: (1)Provide a general frame to deal with the limitations of un-weighed versions of lasso-type estimates. A weighted version possesses the oracle properties of selecting the subset of interesting variables with a proper choice of the weights and increasing the number of hits and decreasing the number of false positives. (2) Combine the strengths of Lasso and a quadratic penalty designed to capture additional structure on the features in high dimensional setting. (3) Develop an easy and fast algorithm using the "Alternating Direction Method of Multipliers" approach to find optimal estimator without augmenting or normalizing

data.
In the following, we emphasize on the advantage of using ADMM on a general lasso-type model with a general penalty term and we will show later that this approach is powerful as it provides fast and optimal solution (closed form).We will discuss briefly some results on its convergence and stoping criterion in Section 2. The problem formulation and its corresponding ADMM algorithm are considered in Section 3. Section 4 is devoted to numerical experimentations on an artificial and a real data with a large number of features. We end the paper with a brief discussion in Section 5.

2 Alternating Direction Method of Multipliers

Recently, the alternating direction method of multipliers (ADMM) has been revisited and successfully applied to solving large scale problems arising from different applications. In this section we give an overview of ADMM. Consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & f(\beta) + g(\xi) \\ \text{subject to} \quad & \beta - \xi = 0, \end{aligned} \tag{2}$$

where f and g are two convex functions and $\beta, \xi \in R^p$. In this optimization problem, we have two sets of variables, with separable objective. The augmented Lagrangian for this problem is:

$$L_\tau(\beta, \xi, \delta) = f(\beta) + g(\xi) + \delta^t(\beta - \xi) + (\tau/2)\|\beta - \xi\|_2^2,$$

where δ is the dual variable for the constraint $\beta - \xi = 0$ and $\tau > 0$ is a penalty parameter.
The augmented Lagrangian methods were developed in part to bring robustness to the *dual ascent method*, and in particular, to yield convergence without strong assumptions like strict convexity or finiteness of f and g.
At iteration k, the ADMM algorithm consists of the three steps:

$$\beta^{k+1} := \arg\min_\beta L_\tau(\beta, \xi^k, \delta^k), \quad //\beta\text{-minimization} \tag{3}$$

$$\xi^{k+1} := \arg\min_\xi L_\tau(\beta^{k+1}, \xi, \delta^k), \quad //\xi\text{-minimization} \tag{4}$$

$$\delta^{k+1} := \delta^k + \tau(\beta^{k+1} - \xi^{k+1}). \quad //\text{dual-update} \tag{5}$$

- In the first step of the ADMM algorithm, we fix ξ and δ and minimize the augmented Lagrangian over β.
- In the second step, we fix β and δ and minimize the augmented Lagrangian over ξ.
- Finally, we update the dual variable δ.

If we consider the *scaled dual variable* $\eta = (1/\tau)\delta$ and the residual $r = \eta - \xi$, the ADMM algorithm can be expressed on its scaled dual form as (we will use the scaled form in the paper):

$$\beta^{k+1} := \arg\min_\beta \left\{ f(\beta) + (\tau/2)\|\beta - \xi^k + \eta^k\|_2^2 \right\}; \tag{6}$$

$$\xi^{k+1} := \arg\min_\xi \left\{ g(\xi) + (\tau/2)\|\beta^{k+1} - \xi + \eta^k\|_2^2 \right\}; \tag{7}$$

$$\eta^{k+1} := \eta^k + \beta^{k+1} - \xi^{k+1}. \tag{8}$$

Stopping criteria The primal and dual residuals at iteration k have the forms:

$$e^k_{pri} = (\beta^k - \xi^k), \quad e^k_{dual} = -\tau(\eta^k - \eta^{k-1}).$$

The ADMM algorithm terminates when the primal and dual residuals satisfy stopping criterion. A typical stopping criterion is given in [2] where the authors propose to terminate when $\|e^k_{pri}\| \leq \epsilon^{pri}$, $\|e^k_{dual}\| \leq \epsilon^{dual}$. The tolerances $\epsilon^{pri} > 0$ and $\epsilon^{dual} > 0$ can be chosen using an absolute and relative criterion, such as $\epsilon^{pri} = \sqrt{p}\epsilon^{abs} + \epsilon^{rel} \max\{\|\beta^k\|_2, \|\eta^k\|_2\}$; and $\epsilon^{dual} = \sqrt{p}\epsilon^{abs} + \epsilon^{rel}\tau\|\eta^k\|_2$, where $\epsilon^{abs} > 0$ and $\epsilon^{rel} > 0$ are absolute and relative tolerances. A reasonable value for the relative stopping criterion is $\epsilon^{rel} = 10^{-3}$ or 10^{-4}, while ϵ^{abs} depends on the scale of the typical variable (see [2] for details).

3 Problem formulation and method

In this section we derive an efficient Alternating Direction Method of Multipliers algorithm for a class of Lasso-type estimators with a general penalty term of the form (1). To check if a variable is important or not, we estimate its coefficient $\hat{\beta}$ that minimizes (1) and is a solution of the generic problem:

$$\hat{\beta}_{\text{COFADMM}}(\lambda, \mu) = \arg\min_{\beta} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \frac{\mu}{2}\beta^t\mathbf{Q}\beta + \lambda\sum_{j=1}^{p}\hat{\omega}_j|\beta_j|, \tag{9}$$

where λ, μ are two non negative tuning parameters, $\mathbf{Q}$ is a positive semi-definite matrix. Equation (9) combines the strengths of regularized techniques of type Lasso and a quadratic penalty designed to capture additional structure on the features. When $\hat{\omega}_j = 1$, it is straightforward to show that all type of lasso models (Lasso, Enet, Slasso, L1Cp and Wfusion) are particular case of (9) using an augmented data reparameterization of the form

$$\mathbf{X}^*_{(n+p)\times p} = \begin{pmatrix} \mathbf{X} \\ \sqrt{\mu}\mathbf{L}^t \end{pmatrix}; \mathbf{Q} = \mathbf{L}\mathbf{L}^t; \quad \mathbf{y}^*_{(n+p)} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix},$$

Therefore any efficient algorithm developed to find the whole solution path of the Lasso like least angle regression or coordinate descent algorithm can be applied. Unfortunately, the good properties of the two optimization techniques are overshadowed by the difficulties (i), (ii) and (iii). To deal with those problems, we propose to solve (9) using the ADMM algorithm. The idea is simple and straightforward. First, we propose to re-write (9) on the following ADMM form:

$$\begin{aligned} &\frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + (\mu/2)\beta^t\mathbf{Q}\beta + \lambda\sum_{j=1}^{p}\hat{\omega}_j|\xi_j| \\ &\qquad \text{subject to} \quad \beta - \xi = 0. \end{aligned} \tag{10}$$

If we write $f(\beta) = (1/2)\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + (\mu/2)\beta^t\mathbf{Q}\beta$, $g(\xi) = \lambda\sum_{j=1}^{p}\hat{\omega}_j|\xi_j|$ and $\hat{\omega}_j = (|\hat{\beta}_j| + 1/n)^{-1}$ then (9) becomes (2). Here we can see that f and g are two convex functions. Applying the ADMM algorithm to (10), we have to do the following three steps at each iteration:

The β-minimization step.

This step updates β^k by:

$$\begin{aligned}\beta^{k+1} &:= \arg\min_{\beta}\left\{f(\beta)+(\tau/2)\|\beta-\xi^k+\eta^k\|_2^2\right\}\\ &:= \arg\min_{\beta}\left\{(1/2)\|\mathbf{y}-\mathbf{X}\beta\|_2^2+(\mu/2)\beta^t\mathbf{Q}\beta+(\tau/2)\|\beta-\xi^k+\eta^k\|_2^2\right\}\\ &:= \left(\mathbf{X}^t\mathbf{X}+\mu\mathbf{Q}+\tau I_p\right)^{-1}\times\left(\mathbf{X}^t\mathbf{y}+\tau(\xi^k-\eta^k)\right) \qquad (11)\end{aligned}$$

The ξ-minimization step.

This step updates ξ^k by:

$$\begin{aligned}\xi^{k+1} &:= \arg\min_{\xi}\left\{g(\xi)+(\tau/2)\|\beta^{k+1}-\xi+\eta^k\|_2^2\right\}\\ &:= \arg\min_{\xi}\left\{\lambda\sum_{j=1}^{p}\hat{\omega}_j|\xi_j|+\frac{\tau}{2}\|\beta^{k+1}-\xi+\eta^k\|_2^2\right\}.\end{aligned}$$

We show in the appendix that the solution consists of updating each component ξ_j^k for $j=1,...,p$ by:

$$\begin{aligned}\xi_j^{k+1} &:= sign(\beta_j^{k+1}+\eta_j^k)\max\left(|\beta_j^{k+1}+\eta_j^k|-\frac{\lambda\hat{\omega}_j}{\tau},0\right)\\ &:= \mathrm{S}_{\frac{\lambda\hat{\omega}_j}{\tau}}\left(\beta_j^{k+1}+\eta_j^k\right), \qquad (12)\end{aligned}$$

where

$$\mathrm{S}_{\kappa}(a)=(1-\kappa/|a|)_+a=\begin{cases}a-\kappa & \text{if } a>\kappa\\ 0 & \text{if } |a|\leq\kappa\\ a+\kappa & \text{if } a<-\kappa\end{cases}$$

is the soft thresholding function introduced and analyzed by [4]. The dual-update step is straightforward and consists of updating η^k by $\eta^{k+1}:=\eta^k+\beta^{k+1}-\xi^{k+1}$.
It is worth to notice that since $\tau>0$, $\mu\geq 0$, $\mathbf{X}^t\mathbf{X}$ and $\mathbf{Q}$ are positive semi-definite matrices, $(\mathbf{X}^t\mathbf{X}+\mu\mathbf{Q}+\tau I_p)$ is always invertible. If $p>n$, let $M=\mu Q+\tau I_p$, to alleviate the cost of calculations, we can exploit the Woodbury formula for $(\mathbf{X}^t\mathbf{X}+M)^{-1}$. The following algorithm shows the complete details of COFADMM:

Tuning parameters selection In practice, it is important to select appropriate tuning parameters in order to obtain a good prediction precision and to control the amount of sparsity in the model. Choosing the tuning parameters can be done via minimizing an estimate of the out-of-sample prediction error. If a validation set is available, this can be estimated directly. Lacking a validation set one can use 10-fold cross validation. In our experimentations λ takes 100 logarithmically equally spaced values, $\mu\in\{0,0.1,1,10,100\}$ and $\gamma\in\{0.5,1,2.5,5,25\}$.
Algorithm 1 describes COFADMM with ADMM steps and Table 1 summarizes COFADMM as a general form of different lasso-type model.

Initialize the variables: $\beta^0 = \mathbf{0}, \xi^0 = \mathbf{0}, \eta^0 = \mathbf{0}$;
Select a scalar $\tau > 0$;
while $k = 0, 1, 2, ...$ *until convergence* **do**
  if $j = 0$ *to* p **then**
    $\beta^{k+1} := \left(\mathbf{X}^t\mathbf{X} + \mu\mathbf{Q} + \tau I_p\right)^{-1}\left(\mathbf{X}^t\mathbf{y} + \tau(\xi^k - \eta^k)\right)$;
    $\xi_j^{k+1} := sign(\beta_j^{k+1} + \eta_j^k)\max\left(|\beta_j^{k+1} + \eta_j^k| - \frac{\lambda\hat{\omega}_j}{\tau}, 0\right)$;
    $\eta^{k+1} := \eta^k + \beta^{k+1} - \xi^{k+1}$
  else
    $\mathbf{M} = \mu\mathbf{Q} + \tau I_p$;
    Use *Woodbury* for $(\mathbf{X}^t\mathbf{X} + M)^{-1}$
  end
end

Algorithm 1: Description of COFADMM with ADMM steps.

Method	**TP**	**Q**
M1	$\lambda \geq 0, \mu = 0$	-
M2	$\lambda \geq 0, \mu \geq 0$	Identity matrix of order p
M3	$\lambda \geq 0, \mu \geq 0$	$\mathbf{Q} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix}$
M4	$\lambda \geq 0, \mu \geq 0$	$q_{ij} = \begin{cases} 2\sum_{s \neq i} \frac{1}{1-\rho_{is}^2}, & i = j \\ -2\frac{\rho_{ij}}{1-\rho_{ij}^2}. & i \neq j \end{cases}$
M5	$\gamma > 0$	$\frac{1}{p} \times \begin{pmatrix} \sum w_{1k} & -s_{12}w_{12} & \cdots & -s_{1p}w_{1p} \\ -s_{12}w_{12} & \sum w_{2k} & \cdots & \vdots \\ \vdots & \vdots & \ddots & -s_{p-1,p}w_{p-1,p} \\ -s_{1p}w_{1p} & \cdots & -s_{p-1,p}w_{p-1,p} & \sum w_{pk} \end{pmatrix}$

Table 1: Summary of the five regularization methods as particular case of COFADMM. M1 states for COFADMM$_{Lasso}$, M2 for COFADMM$_{Enet}$, M3 COFADMM$_{Slasso}$, M4 for COFADMM$_{L1cp}$ and M5 for COFADMM$_{Wfusion}$. TP is used for tunning parameters.

4 Performance

All the experiments are performed on a PC machine with Intel Core i7 CPU and 8 GB RAM under Matlab R2009a. In this section, we present one artificial study and a Glioblastoma microarray data set analysis to illustrate the performance of the COFADMM algorithm under various conditions (time for big data and selection of variables with collinearity). In both numerical experimentations τ is fixed and $\mu \in \{0, 0.1, 1, 10, 100\}$, so we can catch few initial factorizations of $(\mathbf{X}^t\mathbf{X} + \mu\mathbf{Q} + \tau I_p)^{-1}$ to make subsequent iterations much cheaper.

4.1 Performance on artificial data.

This example is the same considered in [2]. The explanatory matrix has $p = 50000$ features and $n = 15000$ observations. The data is generated as follows. We first choose $\mathbf{X}_{ij} \sim N(0, 1)$ and then normalize the columns to have unit ℓ_2 norm. The regression coefficient $\beta \in \mathbf{R}^p$ is generated with 100 nonzero components, each sampled from a $N(0, 1)$. The noise vector is a $N(\mathbf{0}, 10^{-3}\mathbf{I}_n)$. In terms of timings, it is clear that COFADMM is the winner (see Table 2), followed by lasso-type models with Gauss-Seidel, while lasso-type models with LARS are very slow for this high dimensional example. The coordinate-descent slows down under high correlation. We also note that despite the computational advantage, coordinate-descent suffers from the limitations (ii) and (iii) cited in the introduction.

Method	Tuning parameters	*No.*	T	Method	T(GS)	T(LARS)
COFADMM-Lasso	$\lambda = \lambda_0 = \|\mathbf{X}^t\mathbf{y}\|_\infty$	15	**8.00**	Lasso	57.58	800.88
COFADMM-Enet	$\lambda = \lambda_0, \mu = \frac{1}{100}$	35	**5.12**	Enet	6.30	> 7602.35
COFADMM-L1cp	$\lambda = \lambda_0, \mu = \frac{1}{100}$	38	**6.13**	L1cp	67.53	$> 24h$
COFADMM-Wfusion	$\lambda = \lambda_0, \mu = \frac{1}{100}, \gamma = \frac{5}{2}$	37	6.73	Wfusion	**4.56**	$> 24h$
COFADMM-Slasso	$\lambda = \lambda_0, \mu = \frac{1}{100}$	34	5.44	Slasso	**3.79**	$> 24h$

Table 2: $No.$ is the number of iteration to convergence of COFADMM and $T(GS)$ and $T(LARS)$ are the time in seconds of lasso-type models using Gauss-Seidel and LARS.

4.2 Performance on Real Data.

American Association of Neurological Surgeons (AANS) define astrocytoma as the most invasive type of glial tumor, because of their rapid growth potential, and their spread to nearby brain tissue. the median survival rate is three months without therapeutic intervention, and by optimal therapy, resection, radiation and chemotherapy, the survival can be extended to fifteen months, and fewer than 25% of patients surviving up to two years [8].
Global gene expression data from a set of clinical tumor samples of $n = 111$ are obtained by high-density Affymetrix arrays. Expression values of 360 genes are available. We use the logarithm of time to death as the response variable. As mentioned before, due to the large number of the genes, traditional algorithms fail to analyze this data. Regularization methods of type-lasso selected no more than 40 genes and failed to include the groups of significant genes while COFADMM$_{Lasso}$, COFADMM$_{ENet}$, COFADMM$_{L1CP}$ and COFADMM$_{Wfusion}$ selected 45, 53, 58 and 50 respectively including significant genes. Due to the grouping effect, our method selected significant genes and revealed that some of the genes that are negatively associated with survival are expressed in neurons (VSNL1), in cytoplasm and/or nucleus (S100A4), and in plasma membrane (RGS3). In fact, (VSNL1) is a member of recovering family and neuronal calcium-sensor protein, (S100) proteins is a group of calcium-binding proteins which has a crucial role in motility features of tumor astrocytes by modifications in organization of actin cytoskeleton and the expression of its different regulators. Finally, (RGS) gene regulates the duration of cell signaling and is ubiquitous negative regulators of G signaling by stimulating the rate of GTP hydrolysis on G protein alpha subunits. Moreover, Bassoon (BSN) gene which was identified to be positively associated with the patients' survival by COFADMM has two double zinc finger domains located in the amino terminal part and three coiled-coil domains that may play a role in its interaction with other presynaptic proteins, and harbors a stretch of polyglutamine encoded by CAG repeats. The localization of this protein suggest its role

in the structural and functional organization of the synaptic vesicle cycle, and the release of neurotransmitter by calcium-triggered exocytosis [11].

5 Conclusion and Outloook

In this paper, we have adapted the *ADMM* algorithm for a class of Lasso-type estimators in the context of linear regression models and proposed a model with a general penalty term to guard against sparsity for high-dimensional features, a very challenging problem of big data. We have seen that the proposed algorithm can provide efficiently the estimators for a grid of values of the tuning parameters. It has been demonstrated through simulated and a Glioblastoma data set, that the proposed algorithm gives good performances in both prediction and feature selection viewpoints in a very competitive computational time. The algorithm is particularly useful for situations such that the number of regressors is much larger than sample size. The adaptation of this algorithm to other penalized techniques in the context of linear models; extensions to generalized linear models and support vector machine with more complex penalties is a work in progress.

6 Acknowledgments

We thank Zaki Mohammed from RPI for his critical comments which helped improve the quality of the manuscript.

References

[1] H.D. Bondell and B. J Reich. Simultaneous regression shrinkage, feature selection and clustering of predictors with oscar. *Biometrics.*, 64:115–123, 2007.

[2] E. Chu B. Peleato Boyd, S.N. Parikh and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[3] Z. J. Daye and X. J. Jeng. Shrinkage and model selection with correlated variables via weighted fusion. *J. Neurochem.*, 53:1284–1298, 2009.

[4] D. Donoho and I Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika.*, 81(3):425–455, 1994.

[5] T. Johnstone I. Efron, B. Hastie and R. Tibshirani. Least angle regression. *Annals of Statistics.*, 32:407–499, 2004.

[6] M. El Anbari and A. Mkhadri. Penalized regression with a combination of the l_1 norm and the correlation based penalty. *Sankyha B Journal.*, pages 1–21, 2013.

[7] M. Hebiri and S. van De Geer. The smooth-lasso and other $l_1 + l_2$ penalized methods. *Electronic Journal of Statistics.*, 5:1184–1226, 2011.

[8] Philipp-Niclas Pfenning. *RGS4, CD95L and B7H3: Targeting evasive resistance and the immune privilege of glioblastoma.* PhD thesis, Faculties for the Natural Sciences and for Mathematics of the Ruperto-Carola University of Heidelberg, Germany, 2011.

[9] M. Rosset S. Zhu J. Tibshirani, R. Sunders and K Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal statistical Society, B.*, 67:91–108, 2005.

[10] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal statistical Society, B.*, 58:267–288, 1996.

[11] Langnaese K. Richter K. Kindler S. Soyke A. Wex H. Smalla K.H. Kmpf U. Frnzer J.T. Stumm M. Garner C.C. Gundelfinger E.D. Tom Dieck S., Sanmarti-Vila L. Bassoon, a novel zinc-finger cag/glutamine-repeat protein selectively localized at the active zone of presynaptic nerve terminals. *The Journal of Cell Biology.*, 142(2):499–509, 1998.

[12] H. Zou and T. Hastie. Regularization and variable selection via the elastic-net. *Journal of the Royal statistical Society, B.*, 67:301–320, 2005.

Appendix

Proof of the β-minimization step

$$\begin{aligned}\beta^{k+1} &= \arg\min_{\beta}\left\{(1/2)\|\mathbf{y}-\mathbf{X}\beta\|_2^2+(\mu/2)\beta^t\mathbf{Q}\beta+(\tau/2)\|\mathbf{A}\beta+\mathbf{B}\xi^k-\mathbf{c}+\eta^k\|_2^2\right\}\\ &= \arg\min_{\beta}\left\{(1/2)\|\mathbf{y}-\mathbf{X}\beta\|_2^2+(\mu/2)\beta^t\mathbf{Q}\beta+(\tau/2)\|\beta+\eta^k-\xi^k\|_2^2\right\}\end{aligned}$$

Let $T_1(\beta)=\frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|_2^2+\frac{\mu}{2}\beta^tQ\beta$, $T_2(\beta)=\frac{\tau}{2}\|\beta+\eta^k-\xi^k\|_2^2$, and $\upsilon=\eta-\tau\xi$, then we have $T_1(\beta)=\frac{1}{2}\beta^t(\mathbf{X}^t\mathbf{X}+\mu\mathbf{Q})\beta-\mathbf{y}^t\mathbf{X}\beta+\frac{1}{2}\|\mathbf{y}\|_2^2$ and $T_2(\beta)=\frac{\tau}{2}\left\{\beta^t\beta+2(\eta^k-\xi^k)^t\beta+\|\eta^k-\xi^k\|_2^2\right\}$. So, the β-minimization step is equivalent to minimize

$$\begin{aligned}T(\beta) &= T_1(\beta)+T_2(\beta)\\ &= \frac{1}{2}\beta^t\left(\mathbf{X}^t\mathbf{X}+\mu\mathbf{Q}+\tau I_p\right)\beta+\left(\tau\upsilon^k-\mathbf{y}^t\mathbf{X}\right)\beta+\frac{1}{2}\left(\|\mathbf{y}\|_2^2+\tau\|\upsilon^k\|_2^2\right)\end{aligned}$$

The partial differential of T with respect to β is $\left(\mathbf{X}^t\mathbf{X}+\mu\mathbf{Q}+\tau I_p\right)\beta+\left(\tau\upsilon^k-\mathbf{y}^t\mathbf{X}\right)$. Since β^{k+1} is the minimizer of T, we must have $\left(\mathbf{X}^t\mathbf{X}+\mu\mathbf{Q}+\tau I_p\right)\beta^{k+1}+\left(\tau\upsilon^k-\mathbf{y}^t\mathbf{X}\right)=0$. Finally, $\beta^{k+1}=\left(\mathbf{X}^t\mathbf{X}+\mu\mathbf{Q}+\tau I_p\right)^{-1}\left(\mathbf{X}^t\mathbf{y}+\tau\upsilon^k\right)$

Proof of the ξ-minimization step For the ξ-minimization step, we invoke subdifferential calculus. Recall that it consists of updating ξ^k by:

$$\begin{aligned}\xi^{k+1} &= \arg\min_{\xi}\left\{g(\xi)+\frac{\tau}{2}\|\beta^{k+1}-\xi+\eta^k\|_2^2\right\}\\ &= \tau\arg\min_{\xi}\left\{\frac{\lambda}{\tau}\sum_{j=1}^{p}\hat{\omega}_j|\xi_j|+\frac{1}{2}\|\xi-(\beta^{k+1}+\eta^k)\|_2^2\right\},\end{aligned}$$

So the ξ-minimization step is equivalent to minimize the function $h(\xi)=(1/2)\|\xi-\mathbf{d}^k\|_2^2+\frac{\lambda}{\tau}\sum_{j=1}^{p}\hat{\omega}_j|\xi_j|$ over ξ where $\mathbf{d}^k=\beta^{k+1}+\eta^k$. The optimization problem above is convex. For a minimizer ξ^* of $h(.)$, it is necessary and sufficient that the subdifferential (denoted $\partial h(\xi^*)$) at ξ^* contains zero. Now, for each index j, either $\xi_j^*=0$ or $\xi_j^*\neq 0$. We begin with the case $\xi_j^*\neq 0$. This means that the ordinary first derivative at ξ^* has to be zero: $\xi_j^*-d_j^k+\frac{\lambda\hat{\omega}_j}{\tau}sign(\xi_j^*)=0$, then $\xi_j^*=d_j^k-\frac{\lambda\hat{\omega}_j}{\tau}sign(\xi_j^*)$.

Since we assume $\xi_j^*\neq 0$, this means that either $d_j^k>\frac{\lambda\hat{\omega}_j}{\tau}$ or $d_j^k<-\frac{\lambda\hat{\omega}_j}{\tau}$, and this means that $sign(\xi_j^*)=sign(d_j^k)$. In both cases, we have: $\xi_j^*=d_j^k-\frac{\lambda\hat{\omega}_j}{\tau}sign(\xi_j^*)=sign(d_j^k)\left(|d_j^k|-\frac{\lambda\hat{\omega}_j}{\tau}\right)$. So now we know, that for each index j, if $|d_j^k|$ is greater than $\frac{\lambda\hat{\omega}_j}{\tau}$, then ξ_j^* is simply shrinking d_j^k by $\frac{\lambda\hat{\omega}_j}{\tau}$ towards 0.

On the other hand, if $\xi_j^* = 0$, the subdifferential at ξ^* has to include the zero element. That is: if $\xi_j^* = 0 : \xi_j^* - d_j^k + \frac{\lambda\hat{\omega}_j}{\tau} e = 0$ for some $e \in [-1, 1]$. But this is equivalent to $|d_j^k| \leq \frac{\lambda\hat{\omega}_j}{\tau}$ if $\xi_j^* = 0$. Putting all the cases together, we get that: $\xi_j^* = sign(d_j^k) \max\left(|d_j^k| - \frac{\lambda\hat{\omega}_j}{\tau}, 0\right) = \mathrm{S}_{\frac{\lambda\hat{\omega}_j}{\tau}}(d_j^k)$, where $\mathrm{S}_\kappa(.)$ is the soft thresholding function introduced and analyzed by [4].