Appendix to "Regularized Estimation of Piecewise Constant Gaussian Graphical Models: The Group-Fused Graphical Lasso" published in the Journal of Computational and Graphical Statistics

Alexander J. Gibberd and James D. B. Nelson
Department of Statistical Science, University College London

February 9, 2017

Appendix

Eigen-decomposition for Likelihood

Proposition. *Given the symmetric matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{P\times P}$ obey $\boldsymbol{X}^{-1} - \gamma \boldsymbol{X} = \boldsymbol{Y}$ for some constant γ then $\boldsymbol{X}$ and $\boldsymbol{Y}$ share the same eigenvectors. Further to this, it is also the case that the ith eigenvalues of $\boldsymbol{X}$ and $\boldsymbol{Y}$ denoted λ_{X_i} and λ_{Y_i} will satisfy the quadratic equation $\lambda_{X_i}^{-1} - \gamma\lambda_{X_i} = \lambda_{Y_i}$.*

A matrix $\boldsymbol{A}$ is invertible iff all of its eigenvalues are non-zero, thus:

$$\boldsymbol{A}\boldsymbol{v} = \lambda_{A_i}\boldsymbol{v} \iff \boldsymbol{A}^{-1}\boldsymbol{v} = \frac{1}{\lambda_{A_i}}\boldsymbol{v}.$$

Letting $\boldsymbol{A}^{-1} = \boldsymbol{Y} + \gamma\boldsymbol{X}$, from the above we find $\boldsymbol{A}^{-1}\boldsymbol{v}_i = \boldsymbol{v}_i/\lambda_{A_i}$ and thus $\boldsymbol{Y}\boldsymbol{v}_i + \gamma\boldsymbol{X}\boldsymbol{v}_i = \boldsymbol{v}_i/\lambda_{X_i}$. We now have $\boldsymbol{Y}\boldsymbol{v}_i = \boldsymbol{v}_i/\lambda_{X_i} - \gamma\boldsymbol{X}\boldsymbol{v}_i$, thus $\boldsymbol{v}_i/\lambda_{X_i} - \gamma\lambda_{X_i}\boldsymbol{v}_i = \lambda_{Y_i}\boldsymbol{v}_i$. Dividing through by the common eigenvector we find the quadratic relation $\lambda_{X_i}^{-1} - \gamma\lambda_{X_i} = \lambda_{Y_i}$.

Solving the group lasso a note on GFLSeg

To solve the group lasso problem in the GFGL subroutine we use the *GFLseg* algorithm developed by Bleakley and Vert (2011). This algorithm utilizes a natural block structure in the group lasso problem (we formulate Eq. 14 in this form):

$$\hat{\boldsymbol{\Gamma}} := \underset{\boldsymbol{\Gamma}\in\mathbb{R}^{(T-1)\times P}}{\arg\min} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\Gamma}\|_2^2 + \lambda_2\|\boldsymbol{\Gamma}\|_{2,1} \ ,$$

where $\boldsymbol{Y}$ is a data or target matrix and $\boldsymbol{X}$ is referred to as the design matrix. We see that the group lasso problem as formulated above is linearly separable across the groups, given by rows in $\boldsymbol{\Gamma}$. We can write the regularizer as; $\|\boldsymbol{\Gamma}\|_{2,1} = \sum_{T-1} \|\Gamma_{t,\cdot}\|$ and note that the sum of squared term can also be decomposed across such groups (in our application the groups refer to time slices).

The update for block t can be found according to (Bleakley and Vert, 2011):

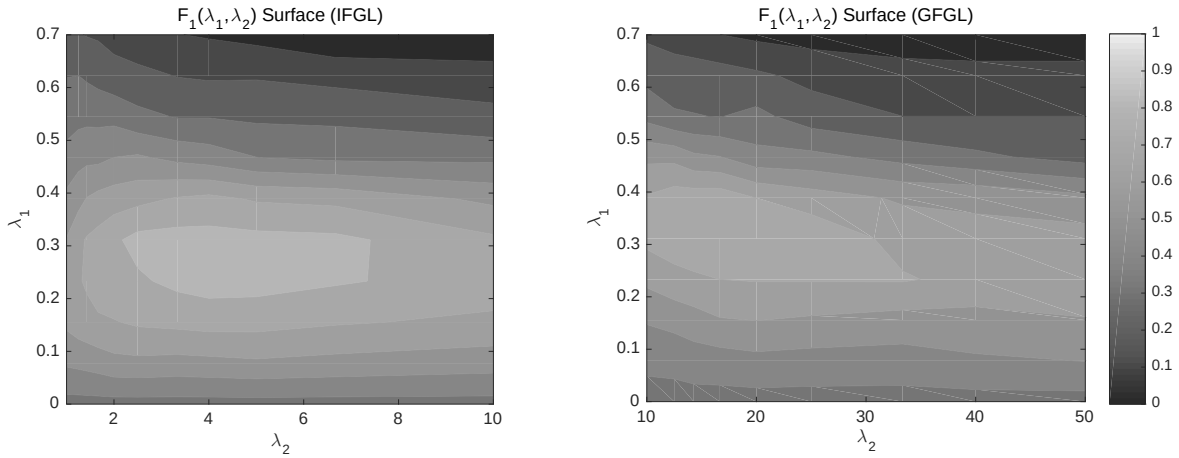
$$\Gamma_{t,\cdot} \leftarrow \frac{1}{\|X_{\cdot,t}\|^2}\left(1 - \frac{\lambda_2}{\|\boldsymbol{e}_t^{-t}\|}\right)_+ \boldsymbol{e}_t^{-t} \ ,$$

where $\boldsymbol{e}_t^{-t} = X_{\cdot,t}^\top, \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\Gamma}^{-t})$, and $\boldsymbol{\Gamma}^{-t}$ denotes the matrix $\boldsymbol{\Gamma}$ with the t-th row set to zero. If one applies the above update scheme then the estimates are guaranteed to converge (Yuan and Lin, 2006). To speed up the algorithm Bleakley et al. adopt an active set strategy. This takes advantage of the fact we expect only few active blocks (which would correspond to changepoints), one simply iterates between adding blocks to the active set $\mathcal{A}$ according to maximal violation of the KKT conditions and updating blocks in $\mathcal{A}$ according to the above. The KKT conditions for the group lasso are given as:

$$\begin{aligned} -\boldsymbol{e}_t + \frac{\lambda_2 \Gamma_{t,\cdot}}{\|\Gamma_{t,\cdot}\|} = 0 \quad \forall\, \Gamma_{t,\cdot} \neq 0 \ , \\ \|-\boldsymbol{e}_t\| \leq \lambda_2 \quad \forall\, \Gamma_{t,\cdot} = 0 \ , \end{aligned}$$

where $\boldsymbol{e}_t = X_{\cdot,t}^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\Gamma})$ is the residual projected along the t-th group.

Sensitivity to hyper-parameters (λ_1, λ_2)



(a) (b)

Figure 1: Example of averaged F_1-score surfaces for; a) IFGL, and b) GFGL, $P = 10, T = 50$. Color represents F_1-score averaged over each time-series for a particular (λ_1, λ_2) setting.

The surface formed by $F_1(\lambda_1, \lambda_2)$ as a function of hyper-parameters provides intuition as to how the sparsity and smoothing regularizers constrain the estimate. As the example in Figure1 demonstrates, recovery performance is coupled to both λ_1 and λ_2. In the IFGL example, it appears that the performance gradient with respect to smoothing (variation in λ_2) is fairly independent of λ_1. With GFGL the dependence between λ_1 and λ_2 may be greater due to the grouping effect of the smoothing regularizer. For example, in the GFGL case, it appears that to achieve a given level of performance increased sparsity regularization (increased λ_1) is required for a small λ_2. In the paper, we fix the sparsity level λ_1 and discuss what happens under a range of smoothing parameters λ_2 (see changepoint density plots, i.e. Fig. 2). In the gene-dependency application, one can quite clearly see the inter-dependency of the smoothing and sparsity parameters for GFGL, it is less pronounced for IFGL (see Fig. 5).

Parameter estimation via BIC

It is interesting to consider the application of in-sample estimation methods for tuning parameters. Whilst in traditional linear regression models, one adopts a degree of freedom based on the number of parameters free to vary, in our regularized estimators the effective *degrees of freedom* are much harder to estimate. In the canonical sparse-estimation model of the lasso (Tibshirani, 1996), it can be shown that at least in standard asymptotic settings the degrees of freedom are given simply by counting the number of non-zero parameters.

One previously suggested estimate of the degrees of freedom (Monti et al., 2014), was used in IFGL type models and considers counting the number of active edges at $t = 1$ and corresponding changes for $t = 1, \ldots, T$. More formally, we can define this as a part corresponding to the changes: $k_{\text{diff}} = |\{\mathbf{1}(\Theta_{i,j,t} \neq \Theta_{i,j,t-1}) \,|\forall i \neq j, t = 2, \ldots, T\}|$, and the initial edges, such that $k_{\text{total}} = k_{\text{diff}} + |\{\mathbf{1}(\Theta_{i,j,1} \neq 0) \mid \forall i \neq j\}|$. In Figure 2 below, we compare the BIC surfaces defined as:

$$\text{BIC}(\lambda_1, \lambda_2) \propto -2L(\{\hat{\boldsymbol{\Theta}}\}, \boldsymbol{Y}) + k_{\text{total}}(\log(T) - \log(2\pi)) \; .$$

As BIC is a form of in-sample estimation for the tuning parameters, Fig. 2 only presents analysis on a single synthetic data-set. This is in contrast to Fig. 1 where the surfaces are averaged across a set of N_{train} time-series. It is, however, clear from these examples that the BIC heuristic implemented does not appropriately select a set of parameters which will perform well in terms of selecting the correct model structure. We see quite clearly that the minima of the BIC surface does not correspond with good F_1-score results.

We hypothesize that this is due in part to the large bias imparted on the likelihood term by the shrinkage, and relatively strong priors we are using in this circumstance. One may attempt to correct for this by using GFGL/IFGL as a first stage screening step and then re-fitting a GGM based on the identified sparsity pattern. In the GFGL case, estimation of the degrees of freedom may be complicated by the presence of grouping effects. However, it is not obvious how to effectively estimate these, we therefore leave this as a potential topic for future research.

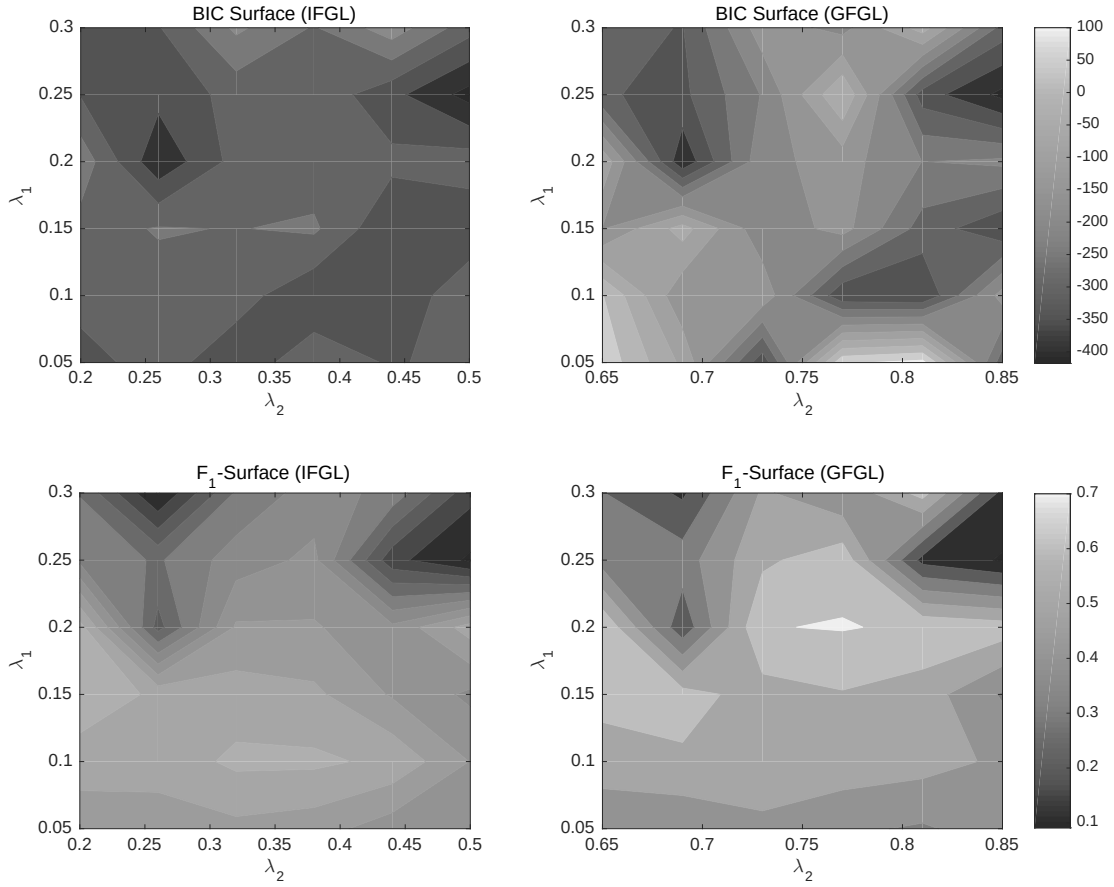


Figure 2: Examples of BIC vs F_1-score surfaces for IFGL (left) and GFGL (right), $P = 10, T = 50$.

Accompanying MATLAB code

Code associated with this paper is available online. This package contains the basic Alternating Directed Method of Multipliers algorithm, alongside files to simulate data, demonstrate, and visualise the recovery of graphical structures.

Requirements

This package depends on two libraries for efficiently computing projection operators:

1. Within GFGL, for solving the group lasso projection we utilise a block-coordinate descent routine developed by Bleakley and Vert (2011) in the paper *"The group fused Lasso for multiple change-point detection"*.

   `http://cbio.ensmp.fr/~jvert/svn/GFLseg/html/`

2. For the solving the independent Fused Graphical Lasso (IFGL) we utilise the efficient routines described in the paper *"An Efficient Algorithm for a Class of Fused Lasso Problems"* Liu et al. (2010). Such routines for sparse learning are conveniently packaged in the *Sparse Learning with Efficient Projections (SLEP)* package.

   `http://www.yelab.net/software/SLEP/`

*Files are downloaded in the absence of required libraries being specified on the MATLAB path

Installation

NOTE: The installation procedure has been tested in Mac OSX and Linux (Redhat) but relies on the use of wget, tar and unzip commands. This may not work automatically in Windows. The required packages (SLEP and GFLseg) may be required to be installed by hand and can be found at the addresses given in "Requirements".

1. Extract the contents of this folder into a directory included on the MATLAB path
2. Run *install.m* to attempt to automatically download dependencies

Examples

We give two examples of GFGL and IFGL applied to simulated data. The first "normalDemo()" demonstrates recovery of graphical structure in the standard T>P setting; the second "hdDemo()" looks at the high-dimensional case. Hyper-parameters can be specified through setting lambda1 (for sparsity), or lambda2 (for smoothness), note I/G refer to parameters for IFGL/GFGL respectively.

```
>>normalDemo(lambda1G,lambda2G,lambda1I,lambda2I)
```

Estimation of a graph with size $P = 5$, $T = 30$, with $n = 5$ true edges. Data is from zero-mean Gaussian with dynamic correlation structure specified by the graph and a single changepoint located at cp=T/2.

Plot 1 - presents the estimated graph within each segment alongside the recovered estimates for GFGL and IFGL.

Plot 2 - plots the estimates for the active ground-truth edges (highlighting changepoints) as a function of $t = 1, .., T$. The grouping property of GFGL in this setting.

High-dimensional setting

```
>>hdDemo(lambda1G,lambda2G,lambda1I,lambda2I)
```

Same as above but in *high dimensional* ($P > T$) setting, with $P = 20$,$T = 10$, and $n = 5$ true edges

References

Bleakley, K. and Vert, J. P. "The group fused Lasso for multiple change-point detection." *arXiv preprint arXiv:1106.4199*, 1–24 (2011).

Liu, J., Yuan, L., and Ye, J. "An efficient algorithm for a class of fused lasso problems." *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 323 (2010).

Monti, R. P., Hellyer, P., Sharp, D., Leech, R., Anagnostopoulos, C., and Montana, G. "Estimating time-varying brain connectivity networks from functional MRI time series." *NeuroImage* (2014).

Tibshirani, R. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (1996).

Yuan, M. and Lin, Y. "Model selection and estimation in regression with grouped variables." *Journal of the Royal Statistical Society* (2006).