A fast homotopy algorithm for a large class of weighted classification problems

Pierre Gutierrez[1,2], **Guillem Rigaill**[1], **Julien Chiquet**[2],
{pgutierrez,julien.chiquet}@genopole.cnrs.fr, rigaill@evry.inra.fr
[1]Unité de Recherche en Génomique Végétale INRA-CNRS-Universitée d'Évry, France
[2]Laboratoire Statistique et Génome UMR CNRS 8071-USC INRA-Universitée d'Évry, France

Abstract

We propose a penalized method to solve the one-way ANOVA problem by collapsing the coefficients of K conditions. We introduce a large class of weights for which our homotopy algorithm is in $\mathcal{O}(K \log(K))$. These weights induce a balanced tree structure and simplify the interpretation of the results. As an example we consider phenotypic data: given a trait, we reconstruct a balanced tree structure and assess its agreement with the known phylogeny. Our proposal is easily extended to more than one dimension for clustering problems.

1 Introduction

With the advent of new high-throughput technologies, it is possible to compare features across a very large number, K, of conditions. For one feature one typically apply one-way ANOVA to test for any significant difference between conditions. Large K leads to multiple-testing and algorithmic problems since the number of pairwise tests is in $\mathcal{O}(K^2)$. Furthermore, each test is performed independently and the resulting structure between the conditions is not necessarily simple and easily interpretable.

In this work, we propose a penalized version of the one-way ANOVA achieving these goals by constructing a hierarchical structure on the conditions at a low computational cost. To this purpose, we use a fusion penalty that collapses the coefficients within the conditions in the same manner as the fused-Lasso [Tibshirani et al., 2005]. We prove that for a large class of weights no split can occur along the path of solutions. These weights lead to a balanced tree structure.

An analogous strategy called "Cas-ANOVA" has been investigated in Bondell and Reich [2008] for multi-factor ANOVA. They propose some weights which, coupled with a two stage strategy like in the adaptive Lasso [Zou, 2006], enjoys asymptotic consistency for a fixed number of conditions. Still, it can be shown that these weights do not lead to a tree. Indeed, as soon as the number of individual by condition is unbalanced, splits can easily occur along the solution path. Moreover, the optimization procedure of Bondell and Reich is quadratic in K and only provides the solution for a given λ. We also experienced numerical instability using their weights.

Hocking et al. [2011] proposed a similar penalty in the context of clustering. When there is just one individual per condition and for fixed weights equal to one, they showed that no split can occur along the path of solutions and proposed an efficient algorithm: "ClusterPath". However these weights typically lead to unbalanced hierarchies. We extend their results to the case of several individuals per condition – or replicates – and to a larger class of weights that induces a balanced tree structure.

In this short note, we illustrate our weighted penalty in a one-dimensional setup. Since it boils down to the one-way ANOVA, we call our method "fused-ANOVA". A straightforward generalization of this work to a p-dimensional space is to consider successive calls of the one-dimensional fused-ANOVA for each dimension, as is done in "ClusterPath".

2 The fused-ANOVA Model

We first recall the classical one-way ANOVA setup. Let Y_{ik} be the intensity of a continuous random variable for sample i in condition k, which decomposes as

$$Y_{ik} = \beta_k + \varepsilon_{ik}, \quad \varepsilon_{ik} \sim \mathcal{N}(0, \sigma_{ik}^2), \tag{1}$$

where β_k is the mean parameter of condition k. We denote by K the total number of conditions, n_k the number of samples in condition k and $n = \sum_k n_k$ the total sample size. The estimators $\hat{\beta}_k$ are usually adjusted using ordinary least squares:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{k=1}^{K} \sum_{i=1}^{n_k} (Y_{ik} - \beta_k)^2 \right\}, \tag{2}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)$ is a K dimensional vector that contains the means of the K conditions. Since we consider the model without an intercept term, there is no identifiability issue in (1) and no additional constraint is needed to solve (2).

When the number of conditions K is large, it is natural to assume that the underlying number of different β_k is small. To encode this, we use a (generalized) fused-Lasso penalty:

$$\hat{\boldsymbol{\beta}}_\lambda = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{k=1}^{K} \sum_{i=1}^{n_k} (Y_{ik} - \beta_k)^2 + \lambda \sum_{k,\ell} w_{kl} \, |\beta_k - \beta_\ell| \right\}. \tag{3}$$

This encourages the absolute differences between β_k to be small: the larger the λ, the smaller the differences will be. The weights $w_{k\ell}$ may be interpreted as a prior on the differences between the means of two conditions. An appropriate choice of weights is discussed in the next paragraph.

3 Fast homotopy algorithm for distance decaying weights

The optimization problem (3) can be solved by the homotopy algorithm proposed in Hoefling [2010]. A schematic view of this algorithm is depicted below.

Algorithm 1: Schematic view of the homotopy algorithm for the generalized fused-Lasso

Input: data and weights (Y_{ik}, w_{kl})

Initialization for $\lambda = 0$
Initialize β_k parameters (equal to the empirical means)
Initialize the list of possible next events (only fusion at this stage)

while *all groups are not fused* **do**

> Find the next event (having the smallest λ), it can be a split or a fusion
> Update β_k parameters accordingly
> Update the list of possible next events (fusion and split)

end

Output: DAG of fusion and split events and associated values of the parameters

For unspecified weights, split events may occur in Algorithm 1. However, the absence of splits is highly desirable because if there is no split,

1. the order of the β_k always matches the order of the empirical mean of each condition;
2. the recovered structure is a tree which simplifies the interpretation;
3. the total number of iterations is guaranteed to be small and equal to K;
4. we avoid maximum flow problems whose resolution is computationally demanding.

In the following Theorem, we characterize a large class of weights for which we prove the absence of splits.

Theorem 1. *The path of solutions does not contain splits when weights are chosen such that*

$$w_{k\ell} = n_k n_\ell \, f(|\bar{Y}_k - \bar{Y}_\ell|),$$

where $f(\cdot)$ is a decreasing positive function.

Proof. Schematically, the proof relies on the following lemma, which is itself proven by induction.

Lemma 1. *Consider $\mathcal{A}$ and $\mathcal{B}$ two sets with r elements in $\mathbb{R}$ such that $\mathcal{A} = \{a_1 > \cdots > a_r\}$ is ordered. Also denote by $\Omega^{(r)}$ the set of all permutation of $\{1, \dots, r\}$ and $\hat{\omega}(\mathcal{B})$ the permutation of $\{1, \dots, r\}$ ordering the elements in the decreasing order. Then*

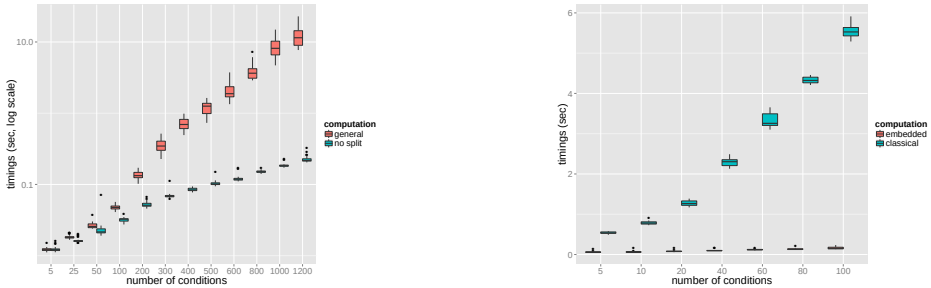
$$\hat{\omega}(\mathcal{B}) = \underset{\omega \in \Omega^{(r)}}{\arg\min} \left\{ \sum_{i=1}^{r} (a_i - b_{\omega(i)})^2 \right\}.$$

From this lemma and using weights as in Theorem 1, it can be shown that the fused-ANOVA loss (3) preserves the order, i.e. if condition k as a smaller empirical mean than condition ℓ then for all λ we have $\beta_k \leq \beta_\ell$. From this, we get that a split cannot occur in the solution path essentially because it would disrupt the order. □

Class of weights. The class of weights in Theorem 1 are natural in the sense that they linearly increase with the number of samples in each condition and they decrease with the mean distance between two conditions. Typical choice for f are the Gaussian kernel $\exp\{-\gamma x^2\}$ or the Laplace kernel $\exp\{-\gamma|x|\}$ where γ is a tuning parameter. Taking $f(x) = 1$ with all $n_k = 1$ leads to the weights proposed by Hocking et al.. Taking $f(x) = 1/x$ leads to an adaptive penalization as in Zou [2006]. However Cas-ANOVA weights $\sqrt{n_k + n_\ell}/(\bar{Y}_k - \bar{Y}_\ell)$ do not fall in this category.

Implementation of the algorithm. We implemented both the general and the without split version of algorithm 1 in C++. For the latter, the complexity of our implementation is $\mathcal{O}(K \log K)$. We also provide a fast cross validation (CV) procedure to select λ for both the general and the no split algorithms. The idea behind this procedure is to take advantage of the DAG structure of the path of solutions along λ. Rather than computing the CV error for each condition separately, we traverse each edge of the DAG once and only once and compute simultaneously the error of all conditions going through this edge. If we consider a perfectly balanced tree and a grid of P values of λ we achieve $\mathcal{O}(P \log(P))$ rather than a $\mathcal{O}(P^2)$ complexity.

Figure 1 illustrates the good performance of our implementation. The left panel shows the performance of our algorithm compared to the general algorithm 1 for weights matching the assumptions of Theorem 1. The right panel illustrates the performance of our embedded CV procedure compared to the naive implementation.

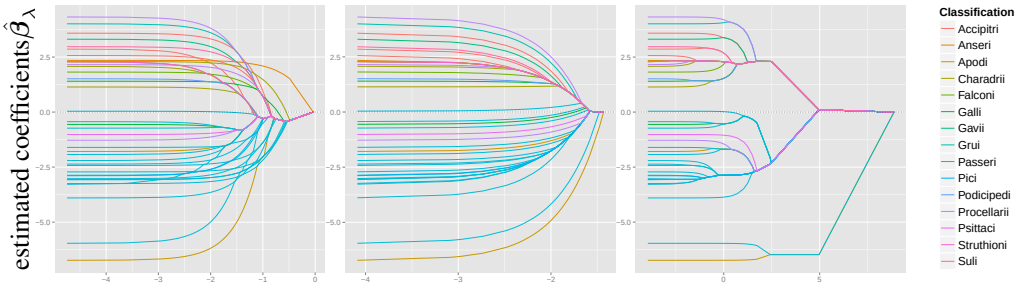


a) improvements induced by absence of splits b) improvements induced by embedded CV

Figure 1: timings comparison for a) general/without split algorithm and b) naive/embedded CV. We consider model (1) with a varying number of conditions K, $n_k = 20$, $\sigma_{ik}^2 = 1$ and β_k independently drawn from $\mathcal{U}([0, 20])$. Experiments are replicated 20 times for averaging.

4 Distance decaying weights simplifies the interpretation: application to phylogenetic data

We download the "Animal Ageing Longevity Database", publicly available at http://genomics.senescence.info/species/, which provides various features for many animal species. Here, we consider predicting the birth weight for 40 bird families classified in 15 orders and regrouping a total of $n = 184$ individuals. The number of birds per family is not constant. We then checked whether the recovered classification matches the orders of these families. Recovered solutions path are plotted in Figure 2 for a) the non-adaptive weighting scheme of Cas-ANOVA[1] ; b) the "default" weights $w_{k\ell} = n_k n_\ell$; and c) the Laplace weights $w_{k\ell} = n_k n_\ell \exp \gamma |\bar{Y}_k - \bar{Y}_\ell|$, where γ has been tuned in order to maximize the adjusted rand index with the phylogenetic classification. Only choices b) and c) fulfill the assumptions of Theorem 1. On the left panel, the Cas-ANOVA path includes many splits which make interpretation rather difficult. On the middle panel, default weights, as expected, provide a tree structure. Still, the structure of this tree is unbalanced and thus not fully satisfactory in the sense that small groups often fuse with very large ones. Finally, the right panel shows the tree reconstructed using the Laplace weights. Not only its structure is much more balanced, but it is also in better agreement with the known phylogenetic classification. The Laplace weights improved the rand index by 5% compared to ClusterPath.



a) Cas-ANOVA weights b) Default weights c) Laplace weights

Figure 2: Reconstructed phylogenetic trees for various weighting schemes. Families classified in the same order share the same color.

5 Conclusion

This work provides a fast algorithm to solve a weighted fused Lasso penalty devoted to ANOVA and clustering problems. For a large class of weights we achieve a $K \log(K)$ complexity where K is the number of conditions in ANOVA or the number of sample in clustering. These weights also lead to a balanced hierarchical structure on the conditions which is easily interpretable.

References

H.D. Bondell and B.J. Reich. Simultaneous factor selection and collapsing levels in anova. *Biometrics*, 65(1):169–177, 2008.

T. Hocking, F. Vert, J.-P.and Bach, and A. Joulin. Clusterpath: an algorithm for clustering using convex fusion penalties. In *Proceedings of the 28th ICML*, pages 745–752, 2011.

H. Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108, 2005.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[1]Because of unbalanced sample sizes between conditions the algorithm for the adaptive version of Cas-ANOVA did not converge.