L_1/L_p Regularization of Differences

Marcel van Gerven and Tom Heskes
Institute for Computing and Information Sciences
Radboud University Nijmegen
Toernooiveld 1, 6525 ED, Nijmegen, The Netherlands
{marcelge,tomh}@cs.ru.nl

Abstract

In this paper, we introduce L_1/L_p regularization of differences as a new regularization approach that can directly regularize models such as the naive Bayes classifier and (autoregressive) hidden Markov models. An algorithm is developed that selects values of the regularization parameter based on a derived stability condition. for the regularized naive Bayes classifier, we show that the method performs comparably to a filtering algorithm based on mutual information for eight datasets that have been selected from the UCI machine learning repository.

1 Introduction

L_1 regularization is a well-known approach to feature selection [1]. This is achieved by penalizing models that use a large number of features and can be interpreted as finding a MAP solution with a Laplace prior on individual features [2]. When we are dealing with models that are composed of factors then we would like feature selection to be defined on the factor level instead of on the level of factor components. In order to achieve this objective, recent research has focused on the notion of L_1/L_p regularization [3, 4, 5]. By placing an L_1 norm on the factor level and an L_p norm with $p > 1$ on the level of factor components one is able to construct models that consist of a small number of factors.

It is important to realize that L_1/L_p regularization assumes that a factor is selected if and only if any of its factor components is non-zero. However, this not the most natural interpretation of feature selection for a particular family of models. Consider a generative model where the class variable depends on a set of feature variables that can each be described by a set of class-conditional parameters. Interpreting these feature parameters as factors in our model, assuming that feature selection is equivalent to all parameters equaling zero makes no sense. Rather, for a feature to be removed from the model, we would like all class-conditional parameters to be identical, since in that case, the feature is not used to distinguish the classes.

In this paper, we develop an approach that allows for the regularization of these types of models. We call this approach L_1/L_p *regularization of differences*. Furthermore, we derive stability conditions that allow us to determine when (i.e., at what value of a regularization parameter) each feature is selected. We use these conditions to construct an efficient algorithm for feature selection in large models. The approach is demonstrated by defining the *regularized naive Bayes* classifier. Classification performance is validated using eight datasets that have been selected from the UCI machine learning repository.

2 L_1/L_p Regularization

Consider a set of parameters Θ that is partitioned into factors $i \in (1, \ldots, I)$, where factor components $\boldsymbol{\theta}_i \equiv (\boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iJ})$ belong together in some, as yet unspecified, way. Let the L_p norm be defined as $||\boldsymbol{x}||_p \equiv \sqrt[p]{\sum_{x \in \boldsymbol{x}} |x|^p}$. In L_1/L_p regularization, we consider objective functions of the form

$$E(\Theta) = \ell(\Theta) + \lambda \Delta_p(\Theta) \,,$$

with $\ell(\Theta)$ a loss term, such as the negative loglikelihood of data given a particular probability model parameterized by Θ, λ a regularization constant, and the regularization term

$$\Delta_p(\Theta) \equiv ||(||f(\boldsymbol{\theta}_1)||_p, \ldots, ||f(\boldsymbol{\theta}_I)||_p)||_1 \,.$$

The regularization term penalizes model complexity, where the L_1 norm on factors $\boldsymbol{\theta}_i$ leads to the selection of a small number of factors and the regularization of factor components is achieved by the L_p norm when $p > 1$. The function f is used to denote some arbitrary transformation of the factors. In this paper, we consider the case where f is used to center subfactors of $\boldsymbol{\theta}_i$ to their mean. We will refer to this approach as L_1/L_p *regularization of differences*. The regularization term is then written as:

$$\Delta_p(\Theta) \equiv \sum_{i=1}^{I} \sqrt[p]{\sum_{j=1}^{J} \sum_{k=1}^{K} \left|\theta_{ijk} - \bar{\boldsymbol{\theta}}_{ij}\right|^p} \,, \tag{1}$$

where factors are indexed by i, factor components by j, and subcomponents by k. The introduction of subcomponents is useful if a feature is represented by means of multiple (class-conditional) parameters.

A main point of interest is to determine when (i.e, for what values of the regularization parameter λ) a feature becomes part of the model. Consider fixing all vectors $\boldsymbol{\theta}_l$ for $l \neq i$ to some $\boldsymbol{\theta}_l^*$. We would like to investigate whether it makes sense to move $\boldsymbol{\theta}_i$ away from $\bar{\boldsymbol{\theta}}_i^*$, defined as

$$\bar{\boldsymbol{\theta}}_i^* = \underset{\boldsymbol{\theta_i} = [\boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iJ}]}{\operatorname{argmin}} \ell(\Theta_i^*(\boldsymbol{\theta}_i)) \,,$$

where the minimum is under the constraint that all components of $\boldsymbol{\theta}_{ij}$ should be the same, i.e., that the regularization term is equal to zero. Now it can be shown that the solution $\Theta_i^*(\bar{\boldsymbol{\theta}}_i^*)$ is stable if and only if

$$\lambda \geq \min_{\gamma_1,\ldots,\gamma_J} \left\| \left. \frac{\partial \ell(\Theta)}{\partial \boldsymbol{\theta}_i} \right|_{\Theta=\Theta_i^*(\bar{\boldsymbol{\theta}}_i^*)} - [\gamma_1 \mathbf{1}_K, \ldots, \gamma_J \mathbf{1}_K]^T \right\|_{p/(p-1)} , \tag{2}$$

the proof of which is deferred to the Appendix.

Note that we do not now in general which values of γ_j minimize the condition 2. However, there are a few special cases for which we do know exactly what the minimizers are. For $p = 2$, (2) simplifies to

$$\lambda \geq \left\| \left. \frac{\partial \ell(\Theta)}{\partial \boldsymbol{\theta}_i} \right|_{\Theta=\Theta_i^*(\bar{\boldsymbol{\theta}}_i^*)} \right\|_2 , \tag{3}$$

since the optimal γ for $p = 2$ is the average of the component-wise gradients, which vanishes at the optimum. For $p = \infty$, (2) simplifies to

$$\lambda \geq \sum_{j=1}^{J} \sum_{k=1}^{K} \left| \left. \frac{\partial \ell(\Theta)}{\partial \theta_{ijk}} \right|_{\Theta=\Theta_i^*(\bar{\boldsymbol{\theta}}_i^*)} - m_j \right| \tag{4}$$

since the median m_j of components $\frac{\partial \ell(\Theta)}{\partial \theta_{ijk}}$ with $1 \leq k \leq K$ evaluated at $\Theta = \Theta_i^*(\bar{\boldsymbol{\theta}}_i^*)$ is the minimizer of the absolute deviations.

It is important to realize that we only consider moving $\boldsymbol{\theta}_i$ away from the *best solution* $\bar{\boldsymbol{\theta}}_i^*$, i.e., the one minimizing the loss term and not just any solution with equal components. Note further that the same result applies when we define the differences with respect to, for example, the median instead of the mean: the important point is that the regularization term forces all components to be the same.

3 Traversing the regularization path

Given a value for the regularization parameter λ, a solution in terms of Θ is found by minimizing the objective function $E(\Theta)$. This minimization is achieved by updating each factor i in terms of its gradient.Since the regularization term is a convex function of Θ, and since the sum of two convex functions is also convex, we are guaranteed to find a unique optimal solution, as long as the loss term is also a convex function of Θ. The loss term and its partial derivatives depend on the chosen form, but the partial derivatives of the regularization term are given by

$$\frac{\partial \Delta_p(\Theta)}{\partial \theta_{ijk}} = \left(|\delta_{ijk}|^{p-1} \operatorname{sgn}(\delta_{ijk}) - \frac{1}{K} \sum_{l=1}^{K} |\delta_{ijl}|^{p-1} \operatorname{sgn}(\delta_{ijl}) \right) ||\boldsymbol{\delta}_i||_p^{1-p} , \tag{5}$$

where $\delta_{ijk} = \theta_{ijk} - \bar{\boldsymbol{\theta}}_{ij}$ and $\boldsymbol{\delta}_i = (\delta_{i11}, \ldots, \delta_{iJK})$.

Ideally, we wish to find an optimal solution associated with a particular value of λ. E.g., if we are dealing with a classification problem then we wish to find the regularized solution that maximizes some measure of the classification performance. Typically, λ is allowed to vary over a fixed interval and obtained solutions are tested on a separate validation set. However, this approach can be costly when the interval is sampled with high resolution. In this paper, we present an algorithm that restricts itself to those values of λ for which solutions become unstable. This is determined by condition (2). We use $S_i(\Theta)$ to refer to the stability condition for factor i. Algorithm 1 only evaluates solutions at those values of λ that correspond to the stability conditions.

Algorithm 1 Traversal of the regularization path.

input $\Theta = (\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_I^*)$, τ, $\lambda = \infty$
$A = \varnothing$
repeat
  repeat
    $\Theta' \leftarrow \Theta$
    for $i \in A$ **do**
      $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \epsilon \nabla_i E(\Theta)$
    end for
  until $\Theta' - \Theta < \tau$
  $\boldsymbol{\Theta} = \boldsymbol{\Theta} \cup \{\Theta\}$
  $a \leftarrow \operatorname{argmax}_{i \notin A} S_i(\Theta)$
  $\lambda \leftarrow S_a(\Theta)$
  $A \leftarrow A \cup \{a\}$
until $A = \{1, \ldots, I\}$
return $\boldsymbol{\Theta}$

The algorithm proceeds as follows. It starts with an infinite λ that is associated with the solution $\Theta = (\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_I^*)$, initialized at $\boldsymbol{\theta}_i^* = \bar{\boldsymbol{\theta}}_i^*$. Then, in each iteration it moves in the direction of the negative gradient of the objective function $E(\Theta)$. Since the gradients of the loss term and the regularization term are used, we need the partial derivatives of both. The gradient of the regularization term is undefined at $\boldsymbol{\theta}_i^*$, in which case the algorithm sets the gradient to zero by convention. The step size ϵ is determined using a one-dimensional line-search [6]. This procedure ensures that factors unequal to $\boldsymbol{\theta}_i^*$ are properly minimized by decreasing the objective function. For factors that are equal to $\boldsymbol{\theta}_i^*$, the mentioned convention induces a move in the direction of the negative gradient of the loss function, whereas the subsequent line-search may set this factor back to $\boldsymbol{\theta}_i^*$, as it evaluates the (regularized) objective function. The tolerance parameter τ is used to determine the precision of the minimization steps. The stability conditions are used to determine a new value of λ in the next iteration. The algorithm returns a model for all evaluated values of λ. Note that these conditions are reevaluated in each iteration and we need at most I such iterations.

Our regularization algorithm is similar to the Boosted Lasso (BLasso) algo-

rithm by Zhao and Yu [7], where the regularization path for L_1 regularization is produced by computing new values of λ during a forward step and the objective function is minimized during a backward step. The forward step in BLasso can be viewed as a numerical method to find the boundaries that are given by our stability conditions.[1] In [7] all features are always considered in the forward step, which has the effect that the regularization path is also approximated for values of λ that do not lead to changes in the selected features.[2] Our algorithm also bears some resemblance to the grafting procedure by Perkins et al. [8] in the sense that parameters are moved to the active set A one by one, while continuously minimizing the objective function. The grafting procedure however moves parameters to the active set based on the magnitude of the gradient of the objective function and only produces a result for a fixed λ such that traversing the regularization path requires an evaluation for a fixed grid of lambdas.

4 Application: regularized naive Bayes

In previous sections, we have derived the stability conditions for L_1/L_p regularization of differences and constructed an algorithm to traverse the regularization path. L_1/L_p regularization of differences is a general technique, but here we use it to develop the *regularized naive Bayes classifier*. This classifier has the advantage that stability conditions do not change during optimization, which simplifies computations.

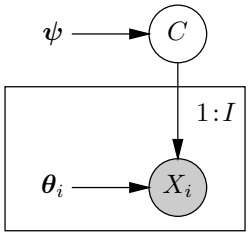

Figure 1: Plate model for the naive Bayes classifier depicting the independence of features $i \in \{1, \ldots, I\}$ given the class variable.

The naive Bayes classifier assumes that features $\mathbf{x} = (x_1, \ldots, x_I)$ are independent given the class value c (Fig. 1). Therefore, the joint probability distribution can be expressed as

$$p(c, \mathbf{x}) = p(c) \prod_{i=1}^{I} p(x_i \mid c) \ . \tag{6}$$

[1] We remark that the forward step of the BLasso algorithm is independent of the step size ϵ (for reasonably small ϵ), and needs to be computed only for those parameters that have not yet been added to the active set.

[2] If this behavior is desired then we recommend considering intermediate λ's that are equally spaced between the values of λ that are found by Algorithm 1.

For the class variable, we just choose the maximum likelihood solution

$$p(c \mid \boldsymbol{\psi}) = \frac{N_c}{N}$$

where $N_{\boldsymbol{z}}$ represents the number of occurences of $\boldsymbol{z}$ in the data $D = \{(c_n, \mathbf{x}_n)\}_{n=1}^N$. In contrast, feature parameters are assumed to be regularized using L_1/L_2 regularization of differences with factors $\boldsymbol{\theta}_i$. Hence, we wish to minimize the objective function

$$E(\Theta) = -\sum_{n=1}^{N}\sum_{i=1}^{I} \log p(x_{ni} \mid c_n, \boldsymbol{\theta}_i) + \lambda \Delta_p(\Theta) \tag{7}$$

where the first term is the negative loglikelihood of the parameters given the data D and the second term regularizes the parameters Θ. Note that we can interpret (7) to give the MAP solution if we use the Laplace prior $p(\Theta) = \frac{\lambda}{2}\exp\left(-\lambda\Delta_p(\Theta)\right)$.

The exact form of the objective function is determined by defining the conditional distributions for the features. In this paper, we assume that features are continous and normally distributed conditional on the class variable. Hence, we use conditional Gaussian densities:

$$p(x_i \mid c, \boldsymbol{\theta}_i) = \frac{1}{\sqrt{2\pi}\sigma_{ic}} \exp - \frac{(x_i - \mu_{ic})^2}{2\sigma_{ic}^2} \quad .$$

where we define $\boldsymbol{\theta}_i = [\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i]$ with $\boldsymbol{\mu}_i = [\mu_{i1}, \ldots, \mu_{iC}]$ and $\boldsymbol{\sigma}_i = [\sigma_{i1}, \ldots, \sigma_{iC}]$. The loss term then becomes

$$\ell(\Theta) \quad = \quad \sum_{n=1}^{N}\sum_{i=1}^{I}\left(\frac{(x_{ni} - \mu_{ic_n})^2}{2\sigma_{ic_n}^2} + \log \sigma_{ic_n}\right) + \text{constant} \tag{8}$$

with partial derivatives

$$\begin{array}{rcl} \frac{\partial \ell(\Theta)}{\partial \mu_{ic}} & = & \sum_{n;c_n=c} \frac{\mu_{ic} - x_{ni}}{\sigma_{ic}^2} \\ \frac{\partial \ell(\Theta)}{\partial \sigma_{ic}} & = & \frac{1}{\sigma_{ic}}\left(N_c - \frac{\sum_{n;c_n=c}(x_{ni} - \mu_{ic})^2}{\sigma_{ic}^2}\right) . \end{array}$$

We are now in the position to ask when the solution $\Theta_i^*(\bar{\boldsymbol{\theta}}_i^*)$ remains stable. First, note that the constrained solution $\bar{\boldsymbol{\theta}}_i^*$ starts at

$$\hat{\mu}_{ic} = \frac{1}{N}\sum_n x_{ni} \text{ and } \hat{\sigma}_{ic}^2 = \frac{1}{N}\sum_n (x_{ni} - \hat{\mu}_{ic_n})^2 \quad .$$

By plugging this in into (3) we obtain

$$\lambda \geq \|\boldsymbol{h}\|_{p/(p-1)} \tag{9}$$

with

$$\boldsymbol{h} = \left(\tfrac{N_1(\hat{\mu}_i - \hat{\mu}_{i1})}{\hat{\sigma}_i^2}, \ldots, \tfrac{N_C(\hat{\mu}_i - \hat{\mu}_{iC})}{\hat{\sigma}_i^2}, \tfrac{1}{\hat{\sigma}_i}\left(N_1 - \tfrac{\sum_{n;c_n=1}(x_{ni} - \hat{\mu}_i)^2}{\hat{\sigma}_i^2}\right), \ldots, \tfrac{1}{\hat{\sigma}_i}\left(N_C - \tfrac{\sum_{n;c_n=C}(x_{ni} - \hat{\mu}_i)^2}{\hat{\sigma}_i^2}\right)\right) \quad .$$

Since, for the naive Bayes classifier, the partial derivatives of the loss with respect to a feature i are independent of all other features $l \neq i$, we can precompute the λ's for which each feature becomes unstable in Algorithm 1 using condition (9).

5 Experiments

In order to examine the performance of Algorithm 1, we have tested the regularized naive Bayes classifier on eight high-dimensional datasets that have been obtained from the UCI machine learning repository. We trained and tested our models using ten-fold cross-validation. Regularized solutions were computed using Algorithm 1 and the optimal model was selected by testing on a separate validation set (10% of the training data).

For the naive Bayes model, the goal of L_1/L_p regularization of differences is to identify which features (i.e., channel-frequency combinations) are used to solve the classification problem. As such, the method can be seen as a feature selection method. Furthermore, since the stability conditions need only be computed once, we obtain an ordering of the features in terms of importance. Hence, it is similar to filtering approaches to feature selection.

Table 1: Classification accuracies for eight UCI datasets using filtering based on mutual information (MI), filtering based on the L_1/L_2 stability condition (i.e., the order in which the features would be selected), and L_1/L_2 regularization using Algorithm 1.

Dataset	**MI filter**	L_1/L_2 **filter**	L_1/L_2 **Regularization**
mfeat	0.78	0.79	0.79
wdbc	0.91	0.91	0.82
ionosphere	0.83	0.82	0.83
sonar	0.62	0.61	0.61
spambase	0.73	0.82	0.82
spectf	0.76	0.73	0.73
vehicle	0.43	0.48	0.48
waveform	0.80	0.80	0.80
average	0.73	0.75	0.74

Table 1 compares the classification accuracies for two filtering approaches and L_1/L_2 regularization. Although the L_1/L_2 filtering approach performs best, it is evident that there are only small differences between the approaches. For the spambase dataset we observe a larger difference between filtering based on mutual information (72%) and the other two approaches (82%).

Table 2 shows how many features are selected on average. Again, although the L_1/L_2 filtering approach selects the smallest number of features, there are no large differences. Note that filtering based on mutual information selects a large number of features for the spambase dataset, which may explain the bad results for this dataset.

Table 2: Number of selected features for eight UCI datasets using filtering based on mutual information (MI), filtering based on the L_1/L_2 stability condition (i.e., the order in which the features would be selected), and L_1/L_2 regularization using Algorithm 1.

Dataset	**MI filter**	L_1/L_2 **filter**	**Regularization**
mfeat	21	19	20
wdbc	14	15	15
ionosphere	17	14	21
sonar	15	16	17
spambase	49	19	20
spectf	2	2	3
vehicle	6	5	6
waveform	15	15	16
average	17	13	15

6 Conclusion

Regularization based on differences is a general technique that can be used in various settings. Here, we have demonstrated its use with respect to the naive Bayes classifier, but we have also employed the technique for regularizing autoregressive hidden Markov models. However, the results so far do not demonstrate a large advantage when applying the technique. In fact, for the regularized naive Bayes classifier, filtering based on the L_1/L_2 stability conditions seems to perform best. However, it may be too early to dismiss L_1/L_p regularization of differences based on the results that have been obtained so far.

Appendix

Proof of Equation (2). Since the vectors $\boldsymbol{\theta}_l$ are fixed for $l \neq i$, we can restrict ourselves to study the dependency of $E(\Theta)$ on $\boldsymbol{\theta}_i$:

$$E(\boldsymbol{\theta}_i) \equiv E(\Theta_i^*(\boldsymbol{\theta}_i)) = \ell(\Theta_i^*(\boldsymbol{\theta}_i)) + \lambda \Delta_p(\boldsymbol{\theta}_i) + \text{ constant } .$$

We now make a change of variables from $\boldsymbol{\theta}_i = [\boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iJ}]$ to their means $\boldsymbol{m} = [m_1, \ldots, m_J]$ and the vector $\boldsymbol{\delta} = [\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_J]$ of differences, with $m_j \equiv \frac{1}{K} \sum_{k=1}^{K} \theta_{ijk}$ and $\delta_{jk} \equiv \theta_{ijk} - m_j$. In terms of these new variables we have

$$E(\boldsymbol{m}, \boldsymbol{\delta}) \equiv \ell(\Theta_i^*(\boldsymbol{m}, \boldsymbol{\delta})) + \lambda \|\boldsymbol{\delta}\|_p \,,$$

with obvious redefinition of $\Theta_i^*(\boldsymbol{m}, \boldsymbol{\delta}) \equiv \Theta_i^*(\boldsymbol{\theta}_i(\boldsymbol{m}, \boldsymbol{\delta}))$. Since there is no regularization w.r.t. $\boldsymbol{m}$, we immediately find that for the solution $(\boldsymbol{m}, \mathbf{0})$ to be stable we need $\boldsymbol{m}$ to be a (local) minimum of $E(\boldsymbol{m}, \mathbf{0})$, which we therefore assume in the following. We then also omit dependencies on $\boldsymbol{m}$ in our notation. The solution $\boldsymbol{\delta} = \mathbf{0}$ is stable if it holds that

$$E(\boldsymbol{\delta}) \geq E(\mathbf{0}) \quad \text{for any choice of infinitesimal } \boldsymbol{\delta} \text{ satisfying the constraint } \textstyle\sum_j \delta_j = 0. \tag{10}$$

A first order Taylor expansion for $\boldsymbol{\delta}$ close to $\mathbf{0}$ yields:

$$E(\boldsymbol{\delta}) = E(\mathbf{0}) + \sum_{j=1}^{J}\sum_{k=1}^{K} \delta_{jk} \left.\frac{\partial \ell(\Theta)}{\partial \theta_{ijk}}\right|_{\Theta=\Theta_i^*(\mathbf{0})} + \lambda\|\boldsymbol{\delta}\|_p \equiv E(\mathbf{0}) + \boldsymbol{g}^T\boldsymbol{\delta} + \lambda\|\boldsymbol{\delta}\|_p \,,$$

where here and in the following we ignore higher order terms and we defined, for ease of notation, $\boldsymbol{g} \equiv \left.\frac{\partial \ell(\Theta)}{\partial \boldsymbol{\theta}_i}\right|_{\Theta=\Theta_i^*(\mathbf{0})}$. The condition (10) then boils down to

$$\lambda \geq -\frac{\boldsymbol{g}^T\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|_p} \quad \forall_{\boldsymbol{\delta};\sum_k \delta_{jk}=0 \quad \forall_j} \,,$$

and thus to

$$\lambda \geq \max_{\boldsymbol{\delta};\sum_k \delta_{jk}=0 \forall_j} \left[-\frac{\boldsymbol{g}^T\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|_p}\right] = \max_{\boldsymbol{\delta};\|\boldsymbol{\delta}\|_p=1;\sum_k \delta_{jk}=0 \forall_j} \boldsymbol{g}^T\boldsymbol{\delta} \,. \tag{11}$$

The last step follows from the observation that the function to be minimized is insensitive to scaling of $\boldsymbol{\delta}$ (as it should) and we can therefore constrain the norm of $\boldsymbol{\delta}$ to any arbitrary value (here chosen to be 1). The first follows from a symmetry argument. Introducing a Lagrange multiplier γ_j for each constraint $\sum_k \delta_{jk} = 0$, we note that this is equivalent to

$$\lambda \geq \min_{\gamma_1,\ldots,\gamma_J} \max_{\boldsymbol{\delta};\|\boldsymbol{\delta}\|_p=1} (\boldsymbol{g}-\boldsymbol{\gamma})^T\boldsymbol{\delta}$$

where we use $\boldsymbol{\gamma} = [\gamma_1\mathbf{1}_K, \ldots, \gamma_J\mathbf{1}_K]^T$. Temporarily disregarding the minimization and introducing an additional Lagrange multiplier ζ for the constraint $\|\boldsymbol{\delta}\|_p = 1$, we get the Lagrangian

$$\mathcal{L}(\boldsymbol{\delta},\zeta) = (\boldsymbol{g}-\boldsymbol{\gamma})^T\boldsymbol{\delta} + \zeta(\|\boldsymbol{\delta}\|_p - 1) \,,$$

with derivative

$$\frac{\partial \mathcal{L}(\boldsymbol{\delta},\zeta)}{\partial \delta_j} = (g_j - \delta_j) + \zeta\|\boldsymbol{\delta}\|_p^{1/p-1}|\delta_j|^{p-1}\operatorname{sgn}(\delta_j) \,.$$

Setting this derivative to zero, we see that the optimal solution $\boldsymbol{\delta}^*$ obeys (for $p > 1$)

$$\delta_j^* \propto |g_j - \gamma_j|^{1/(p-1)} \operatorname{sgn}(g_j - \gamma_j) \,.$$

Furthermore, we see that we obtain a maximum when the proportionality constant is positive, and a minimum when it is negative. Plugging this into (11) and reintroducing the minimization, we finally obtain, after some rewriting

$$\lambda \geq \min_{\gamma_1,\ldots,\gamma_J} \|\boldsymbol{g}-\boldsymbol{\gamma}\|_{p/(p-1)} \,.$$

□

References

[1] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[2] P. M. Williams. Bayesian regularisation and pruning using a Laplace prior. *Neural Computation*, 7(1):117–143, 1995.

[3] G. Obozinski, B. Taskar, and M. I. Jordan. Multi-task feature selection. Technical report, UC Berkeley, Berkeley, 2006.

[4] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J R Statist Soc B*, 68(1):49–67, 2006.

[5] L. Meier, S. van de Geer, and P. Bühlmann. The group Lasso for logistic regression. *Journal of the Royal Statistical Society: Series B*, 70(1):53–71, 2008.

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 3rd edition, 2007.

[7] P. Zhao and B. Yu. Stagewise Lasso. *Journal of Machine Learning Research*, 8:2701–2726, 2007.

[8] S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3:1333–1356, 2003.