Localized Lasso for High-Dimensional Regression

Makoto Yamada
Kyoto University, PRESTO JST
myamada@kuicr.kyoto-u.ac.jp

Koh Takeuchi, Tomoharu Iwata
NTT Communication Science Laboratories
{takeuchi.koh,iwata}@lab.ntt.co.jp

John Shawe-Taylor
University College London
j.shawe-taylor@ucl.ac.uk

Samuel Kaski
Aalto University
samuel.kaski@aalto.fi

1 Introduction

A common problem in molecular medicine, shared by many other fields, is to learn predictions from data consisting of a large number of features (e.g., genes) and a small number of samples (e.g., drugs or patients). A key challenge is to tailor or "personalize" the predictions for each data sample. essentially solving a multi-task learning problem [1, 2] where in each task $n = 1$. The features (genes) important for predictions can be different for different samples (patients or drugs), and reporting the important features is a key part of the data analysis, requiring models that are interpretable in addition to having high prediction accuracy. That is, the problem can be regarded as a *local* feature selection and prediction problem, which would be hard for existing multi-task learning approaches [1, 2].

Sparsity-based linear feature selection methods such as Lasso [3] are popular and useful for large p, small n problems. Standard feature selection methods select the same small set of features for all samples, which is too restrictive for the multi-task type of problems, where for instance effects of different drugs may be based on different features, and dimensionality needs to be minimized due to the very small sample size.

Recently, the network Lasso [4] method has been proposed for learning local functions $f(\boldsymbol{x}_i; \boldsymbol{w}_i), i = 1, \ldots, n$, by using network (graph) information between samples. In network Lasso, a group regularizer is introduced to the difference of the coefficient vectors between linked coefficients (i.e., $\boldsymbol{w}_i - \boldsymbol{w}_j$), making them similar. We can use this regularizer to make the local models borrow strength from linked models. In the network Lasso, sparsity has so far been used only for making the coefficient vectors similar instead of for feature selection, resulting in dense models.

We propose a sparse variant of the network Lasso, called the localized Lasso, which helps to choose interpretable features for each sample. More specifically, we propose to incorporate the sample-wise exclusive regularizer into the network Lasso framework. By imposing the network regularizer, we can borrow strength between samples neighboring in the graph, up to clustering or "stratifying" the samples according to how the predictions are made. Furthermore, by imposing a sample-wise exclusive group regularizer, each learned model is made sparse but the support remains non-empty, in contrast to what could happen with naive regularization. As a result, the sparsity pattern and the weights become similar for neighboring models. We propose an efficient iterative least squares algorithm and show that the algorithm will obtain a globally optimal solution. Through experiments on synthetic and real-world datasets, we show that the proposed localized Lasso outperforms state-of-the-art methods even with a smaller number of features.

2 Problem Formulation

Let us denote an input vector by $\boldsymbol{x} = [x^{(1)}, \ldots, x^{(d)}]^\top \in \mathbb{R}^d$ and the corresponding output value $y \in \mathbb{R}$. The set of samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ has been drawn i.i.d. from a joint probability density $p(\boldsymbol{x}, y)$.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Algorithm 1 Iterative Least-Squares Algorithm for solving Eq. (2)

Input: $\boldsymbol{Z} \in \mathbb{R}^{n \times (dn)}$, $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{R} \in \mathbb{R}^{n \times n}$, λ_1, and λ_2.
Output: $\boldsymbol{W} \in \mathbb{R}^{n \times d}$.
Set $t = 0$, Initialize $\boldsymbol{F}_g^{(0)}$, $\boldsymbol{F}_e^{(0)}$.
repeat
  Compute $\mathrm{vec}(\boldsymbol{W}^{(t+1)}) = (\lambda_1 \boldsymbol{F}_g^{(t)} + \lambda_2 \boldsymbol{F}_e^{(t)})^{-1} \boldsymbol{Z}^\top (\boldsymbol{I}_n + \boldsymbol{Z}(\lambda_1 \boldsymbol{F}_g^{(t)} + \lambda_2 \boldsymbol{F}_e^{(t)})^{-1} \boldsymbol{Z}^\top)^{-1} \boldsymbol{y}$,
  Update $\boldsymbol{F}_g^{(t+1)}$, where $\boldsymbol{F}_g^{(t+1)} = \boldsymbol{I}_d \otimes \boldsymbol{C}^{(t+1)}$.
  Update $\boldsymbol{F}_e^{(t+1)}$, where $[\boldsymbol{F}_e^{(t+1)}]_{\ell,\ell} = \sum_{k=1}^n \frac{I_{k,\ell} \|\boldsymbol{w}_k^{(t+1)}\|_1}{[\mathrm{vec}(|\boldsymbol{W}^{(t+1)}|)]_\ell}$.
  $t = t + 1$.
until Converges

We further assume a graph $\boldsymbol{R} \in \mathbb{R}^{n \times n}$, where $[\boldsymbol{R}]_{i,j} = r_{ij} \geq 0$ is the coefficient that represents the relatedness between the sample pair $(\boldsymbol{x}_i, y_i)$ and $(\boldsymbol{x}_j, y_j)$. In this paper, we assume that $\boldsymbol{R} = \boldsymbol{R}^\top$ and the diagonal elements of $\boldsymbol{R}$ are zero (i.e., $r_{11} = r_{22} = \ldots = r_{nn} = 0$).

The goal in this paper is to select multiple sets of features such that each set of features is locally associated with an individual data point or a cluster, from the training input-output samples and the graph information $\boldsymbol{R}$.

3 Proposed method

In particular, we aim to learn a model with an interpretable sparsity pattern in the features.

We employ the following linear model for each sample i:

$$y_i = \boldsymbol{w}_i^\top \boldsymbol{x}_i. \tag{1}$$

Here $\boldsymbol{w}_i \in \mathbb{R}^d$ contains the regression coefficients for sample $\boldsymbol{x}_i$ and $^\top$ denotes the transpose. Note that in regression problems the weight vectors are typically assumed to be equal, $\boldsymbol{w} = \boldsymbol{w}_1 = \ldots = \boldsymbol{w}_n$. Since we cannot assume the models to be based on the same features, and we want to interpret the support of the model for each sample, we use local models.

The optimization problem of the *localized lasso* can be written as

$$\min_{\boldsymbol{W}} \; J(\boldsymbol{W}) = \sum_{i=1}^n (y_i - \boldsymbol{w}_i^\top \boldsymbol{x}_i)^2 + \lambda_1 \sum_{i,j=1}^n r_{ij} \|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2 + \lambda_2 \sum_{i=1}^n \|\boldsymbol{w}_i\|_1^2, \tag{2}$$

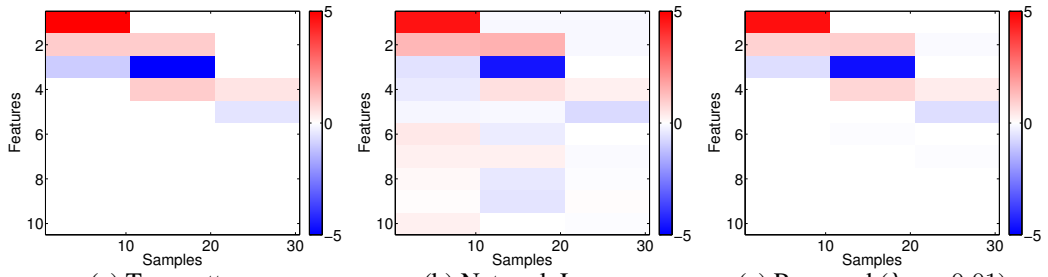
where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are the regularization parameters. By imposing the network regularization (second term) [4], we regularize the model parameters $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ to be similar if $r_{ij} > 0$. If λ_1 is large, we will effectively cluster the samples according to how similar the $\boldsymbol{w}_i$s are, that is, according to the prediction criteria in the local models. More specifically, when $\|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2$ is small (possibly zero), we can regard the i-th sample and j-th sample to belong to the same cluster.

The third term is the $\ell_{1,2}$ regularizer (a.k.a., exclusive regularizer) [5, 6, 7]. By imposing the $\ell_{1,2}$ regularizer, we can select a small number of elements within each $\boldsymbol{w}_i$.

Predicting for new test sample: For predicting on test sample $\boldsymbol{x}$, we use the estimated local models $\widehat{\boldsymbol{w}}_k$ which are linked to the input $\boldsymbol{x}$. More specifically, we solve the Weber problem [4]

$$\min_{\boldsymbol{w}} \quad \sum_{i=1}^n r_i' \|\boldsymbol{w} - \widehat{\boldsymbol{w}}_i\|_2, \tag{3}$$

where $r_i' \geq 0$ is the link information between the test sample and the training sample $\boldsymbol{x}_i$. Since this problem is convex, we can solve it efficiently by an iterative update formula. If there is no link information available, we simply average all $\widehat{\boldsymbol{w}}_i$s to estimate $\widehat{\boldsymbol{w}}$, and then predict as $\widehat{y} = \widehat{\boldsymbol{w}}^\top \boldsymbol{x}$.



(a) True pattern. (b) Network Lasso. (c) Proposed ($\lambda_2 = 0.01$).

(d) Proposed ($\lambda_2 = 0.05$). (e) Network Lasso + ℓ_1 ($\lambda_2 = 0.1$). (f) Network Lasso + ℓ_1 ($\lambda_2 = 0.3$).

Figure 1: The learned coefficient matrix for the synthetic data for the different methods. Proposed = Localized Lasso. For Network Lasso + ℓ_1, we use the ℓ_1 regularizer instead of $\ell_{1,2}$ and $\lambda_2 \geq 0$ is the regularization parameter for the ℓ_1 term.

4 Experiments

In this section, we first illustrate our proposed method on synthetic data and then compare it with existing methods using a real-world dataset.

We compared our proposed method with Lasso [3], Elastic Net [8], FORMULA [9], and Network Lasso [4, 10]. For Lasso, Elastic Net, and FORMULA, we used the publicly available packages. For the network Lasso implementation, we set the regularization parameter to $\lambda_2 = 0$ in the localized Lasso. For supervised regression problems, all tuning parameters are determined by 3-fold nested cross validation. The experiments were run on a 3GHz AMD Opteron Processor with 48GB of RAM.

4.1 Synthetic experiments (High-dimensional regression)

We illustrate the behavior of the proposed method using a synthetic high-dimensional dataset.

In this experiment, we first generated the input variables as $x_{k,i} \sim \text{Unif}(-1, 1), k = 1, \ldots, 10, i = 1, \ldots, 30$. Then, we generated the corresponding output as

$$y_i = \begin{cases} 5x_{1,i} + x_{2,i} - x_{3,i} + 0.1e_i & (i = 1, \ldots, 10) \\ x_{2,i} - 5x_{3,i} + x_{4,i} + 0.1e_i & (i = 11 \ldots, 20) \\ 0.5x_{4,i} - 0.5x_{5,i} + 0.1e_i & (i = 21 \ldots 30) \end{cases}, \tag{4}$$

where $x_{k,i}$ is the value of the k-th feature in the i-th sample and $e_i \sim N(0, 1)$. In addition to the input-output pairs, we also randomly generated the link information matrix $\boldsymbol{R} \in \{0, 1\}^{30 \times 30}$. In the link information matrix, only 40% of true links are observed.We experimentally set the regularization parameter for the proposed method to $\lambda_1 = 5$ and $\lambda_2 = \{0.01, 1, 10\}$. For the network Lasso, we used $\lambda_1 = 5$. Moreover, we compared the proposed method with the network Lasso + ℓ_1 regularizer, in which we used $\lambda_1 = 5$ and $\lambda_2 = \{0.05, 0.5\}$, where λ_2 is the regularization parameter for the ℓ_1 regularizer.

Figures 1(a)-(f) show the true coefficient pattern and the results of the learned coefficient matrices $\boldsymbol{W}$ by using the localized Lasso, the network Lasso, and[1] the network Lasso + ℓ_1. As can be seen, most of the unrelated coefficients of the proposed method are shrunk to zero. On the other hand, for the network Lasso, many unrelated coefficients take non-zero values. Thus, by incorporating the

[1]Note that the combination of the network lasso and ℓ_1 is also new.

Table 1: Test root MSE on toxicogenomics data. The best method under sigfinicance level 5% (Wilcoxon signed-rank test) is reported in bold.

	Blood			Breast			Prostate			Average
	GI50	TGI	LC50	GI50	TGI	LC50	GI50	TGI	LC50	
Localized Lasso	1.030	0.622	0.529	1.129	0.627	0.562	1.297	0.518	0.539	**0.760**
Network Lasso	1.096	0.918	0.921	1.368	0.821	1.065	1.475	0.711	0.690	1.007
FORMULA	1.503	1.179	1.253	1.367	1.109	1.197	1.376	1.121	1.129	1.248
Lasso	1.201	1.006	0.514	1.435	0.879	0.560	1.455	0.763	0.523	0.926
Elastic Net	1.129	0.875	0.514	1.164	0.800	0.560	1.130	0.633	0.505	0.812
Kernel Regression	1.070	0.808	0.623	1.165	0.677	0.688	1.466	0.551	0.509	0.839

Table 2: The number of selected features (genes) on toxicogenomics data. For Localized Lasso, Network Lasso, FORMULA, and Elastic Net, we select features by checking $\|\boldsymbol{W}_{\cdot,i}\|_2 > 10^{-5}$, where $\boldsymbol{W}_{\cdot,i} \in \mathbb{R}^n$ is the i-th column of $\boldsymbol{W}$.

	Blood			Breast			Prostate			Average
	GI50	TGI	LC50	GI50	TGI	LC50	GI50	TGI	LC50	
Localized Lasso	32.7	33.4	92.5	92.6	125.9	58.2	35.1	53.9	43.4	63.4
Network Lasso	1039.6	1047.3	1052.2	1054.6	1051.5	1053.3	1060.5	1052.9	1053.1	1061.0
FORMULA	576.6	445.6	550.5	914.3	936.7	776.3	942.0	712.2	633.6	720.8
Lasso	29.6	12.0	1.0	12.0	1.9	1.0	12.5	4.4	3.8	8.7
Elastic Net	310.8	91.4	39.2	124.9	77.2	1.0	116.6	87.9	98.9	105.3

exclusive regularization in addition to the network regularization, we can learn sparse patterns in high-dimensional regression problems. Moreover, by setting the $\ell_{1,2}$ regularizer term to be stronger, we can obtain a sparser pattern within the $\boldsymbol{w}_i$. In contrast, Network Lasso + ℓ_1, which produces a similar pattern when the regularization is weak (Fig. 1(e)), shrinks many local models to zero if the regularization parameter λ_2 is large (Fig. 1(f)). This shows that the network Lasso + ℓ_1 is sensitive to the setting of the regularization parameter. Moreover, since we want to interpret features for each sample (or model), the $\ell_{1,2}$ norm is more suited than ℓ_1 for our tasks.

4.2 Prediction in Toxicogenomics (High-dimensional regression)

We evaluate our proposed method on the task of predicting toxicity of drugs on three cancer cell lines, based on gene expression measurements. The *Gene Expression* data includes the differential expression of 1106 genes in three different cancer types, for a collection of 53 drugs (i.e., $\boldsymbol{X}_l = [\boldsymbol{x}_1^{(l)}, \ldots, \boldsymbol{x}_{53}^{(l)}] \in \mathbb{R}^{1106\times 53}, l = 1, 2, 3$). The learning data on *Toxicity* to be predicted contains three dose-dependent toxicity profiles of the corresponding 53 drugs over the three cancers (i.e., $\boldsymbol{Y}_l = [\boldsymbol{y}_1^{(l)}, \ldots, \boldsymbol{y}_{53}^{(l)}] \in \mathbb{R}^{3\times 53}, l = 1, 2, 3$). The gene expression data of the three cancers (Blood, Breast and Prostate) comes from the Connectivity Map [11] and was processed to obtain treatment vs. control differential expression. The toxicity screening data from the NCI-60 database [12], summarizes the toxicity of drug treatments in three variables, GI50, LC50 and TGI, representing the 50% growth inhibition, 50% lethal concentration, and total growth inhibition levels. The data were confirmed to represent dose-dependent toxicity profiles for the doses used in the corresponding gene expression dataset.

In this experiment, we randomly split the data into training and test sets. The training set consisted of 48 drugs and the test set of 5 drugs. Moreover, we introduced a bias term in the proposed method (i.e., $[\boldsymbol{x}^\top\ 1]^\top \in \mathbb{R}^{d+1}$), and regularized the entire $\boldsymbol{w}_i$s in the network regularization term, and only $\boldsymbol{v}_i \in \mathbb{R}^{d-1}$ in the $\ell_{1,2}$ regularization term; here $\boldsymbol{w}_i = [\boldsymbol{v}_i^\top\ 1]^\top$. We computed the graph information using the input $\boldsymbol{X}$ as

$$\boldsymbol{R} = \frac{\boldsymbol{S}^\top + \boldsymbol{S}}{2}, \quad [\boldsymbol{S}]_{ij} = \begin{cases} 1 & \boldsymbol{x}_j \text{ is a 5 nearest neighbor of } \boldsymbol{x}_i \\ 0 & \text{Otherwise} \end{cases}.$$

We repeated the experiments 20 times and report the average test RMSE scores in Table 1. We observed that the proposed localized Lasso outperforms state-of-the-art linear methods. Moreover, the proposed method also outperformed the *nonlinear* kernel regression method, which has high predictive power but cannot identify features.

In Table 2, we report the number of selected features in each method. It is clear that the number of selected features in the proposed method is much smaller than that of the network Lasso. In some cases Lasso and Elastic net selected only one feature. This means that the features were shrunken to zero and only bias term remained. In summary, the proposed method is suited for producing interpretable sparse models in addition to having high predictive power.

References

[1] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. *NIPS*, 2007.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

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

[4] David Hallac, Jure Leskovec, and Stephen Boyd. Network lasso: Clustering and optimization in large graphs. In *KDD*, 2015.

[5] Matthieu Kowalski. Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303–324, 2009.

[6] Yang Zhou, Rong Jin, and Steven CH Hoi. Exclusive lasso for multi-task feature selection. In *AISTATS*, 2010.

[7] Deguang Kong, Ryohei Fujimaki, Ji Liu, Feiping Nie, and Chris Ding. Exclusive feature learning on arbitrary structures via ℓ_{12}-norm. In *NIPS*, 2014.

[8] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[9] Jianpeng Xu, Jiayu Zhou, and Pang-Ning Tan. FORMULA: FactORized MUlti-task LeArning for task discovery in personalized medical models. In *SDM*, 2015.

[10] David Hallac, Christopher Wong, Steven Diamond, Rok Sosic, Stephen Boyd, and Jure Leskovec. Snapvx: A network-based convex optimization solver. *arXiv preprint arXiv:1509.06397*, 2015.

[11] Justin Lamb et al. The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313(5795):1929–1935, 2006.

[12] Robert H Shoemaker. The NCI60 human tumour cell line anticancer drug screen. *Nature Reviews Cancer*, 6(10):813–823, 2006.