Group Lasso with Spatial Correlation

Tian Cao*

Department of Computer Science
UNC-Chapel Hill

1 Introduction

In this report, I will focus on group lasso problem proposed in [4]. I will first introduce the standard group lasso then extend it with spatial correlation.

2 Group Lasso

In [4, 3], the proposed group lasso solves the convex optimization problem

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\|b - D\alpha\|_2^2 + \lambda \sum_{g=1}^{G} \|\alpha_g\|_2 \tag{1}$$

where $b \in \mathbf{R}^m$ is the data, $D \in \mathbf{R}^{m \times n}$ is the matrix of dictionary or features, and b and D are given, we want to estimate coefficient α, $\alpha \in \mathbf{R}^n$. Suppose the D are divided into G groups, columnwise such that, $D = (D_1, ..., D_G)$, with $D_g \in \mathbf{R}^{m \times n_g}$, if no overlapping between groups, $\sum_{g=1}^{G} n_g = n$. Thus we have the corresponding with group information $\alpha = [\alpha_1^\top, ..., \alpha_G^\top]^\top$, with $\alpha_g \in \mathbf{R}^{n_g}$. If $n_g = 1$, problem (1) reduces to the lasso problem (ℓ_1 regularized problem). λ is the parameter to control the sparsity at group level.

3 Group Lasso with Spatial Correlation (GLSC)

3.1 Formulation

Now consider we have a series of input data $b_i, i \in (1, 2, ...M)$, if b_i is independent, we can solve problem (1) for each b_i separately. However, when b_i is spatially correlated with each other (here spatial correlation means the input data shares some spatial properties), for example, a couple of image patches with overlapping, we cannot directly solve problem (1) for each b_i because b_i is not independent. The goal is to introduce group sparsity regularizer at the same time take the data correlation in to consideration, i.e., correlated

*tiancao@cs.unc.edu

data b_i, b_{i+1} should have similar sparsity pattern at group level. The group lasso with spatial correlation is formulated as follows,

$$\begin{aligned} \underset{\alpha_i, \tilde{\alpha}_i}{\text{minimize}} \quad & \sum_{i=1}^{M} (\frac{1}{2}\|b_i - D\alpha_i\|_2^2 + \lambda \sum_{g=1}^{G} \|(\tilde{\alpha}_i)_g\|_2) \\ \text{subject to} \quad & \alpha_i - L_i s = 0, \quad i = 1, \ldots, M, \\ & \tilde{\alpha}_i - \tilde{L}_i s = 0, \quad i = 1, \ldots, M. \end{aligned} \tag{2}$$

where α_i is a coefficient corresponding to the input data b_i, L_i is a matrix to select the ith local coefficient from s, $s \in \mathbf{R}^{Mn}$, $L_i \in \mathbf{R}^{n \times Mn}$. $\tilde{\alpha}_i$ is a coefficient corresponding to input data d_i and its spatial neighbors, similarly, $\tilde{L}_i$ is a matrix to select the ith local coefficient with spatial correlation from s. Here, $(\tilde{\alpha}_i)_g$ are the groups of the coefficient $\tilde{\alpha}_i$ based on the group information of D_g. Following [1], we call α_i, $\tilde{\alpha}_i$ local variables (or local copies of the global variable) and s global variable. Here, each local variable α_i is a subset of the global variable s. $\tilde{\alpha}_i$ is a local coefficient considering the spatial correlation, and the size of $\tilde{\alpha}_i$ is based on how we define the neighborhood of the ith input data b_i. For example, if we only consider one horizontal or vertical neighbor of b_i, then $\tilde{\alpha}_i \in \mathbf{R}^{2n}$, $(\tilde{\alpha}_i)_g \in \mathbf{R}^{2n_g}$, $\tilde{L}_i \in \mathbf{R}^{2n \times Mn}$, (see example in Fig. 1(b)), and if we consider one horizontal and one vertical neighbors of b_i, we have $\tilde{\alpha}_i \in \mathbf{R}^{3n}$, $(\tilde{\alpha}_i)_g \in \mathbf{R}^{3n_g}$,$\tilde{L}_i \in \mathbf{R}^{3n \times Mn}$(see example in Fig. 1(c)). λ is the parameter to control the sparsity at group level.

3.2 Example

Here we show an example to demonstrate how to choose the parameters in (2). Suppose we have an input image I, where the image size is 100×100, and the image patch size is 50×50, see example in Fig. 1(a). We extract image patches without overlap (for overlapping patches, the parameter setting is similar), and vectorize each image patch to b_i, $i = 1, 2, 3, 4$, $b_i \in \mathbf{R}^{2500}$. Suppose We have the Dictionary D with the number of dictionary elements 100, thus $D \in \mathbf{R}^{2500 \times 100}$. Thus the global coefficient $s \in \mathbf{R}^{400}$ and the local coefficient α_i, $i = 1, 2, 3, 4$, $\alpha_i \in \mathbf{R}^{100}$. Now the L_i, $i = 1, 2, 3, 4$, is the matrix to select the ith local coefficient from a, $L_i \in \mathbf{R}^{100 \times 400}$. For example, L_2 is defined as

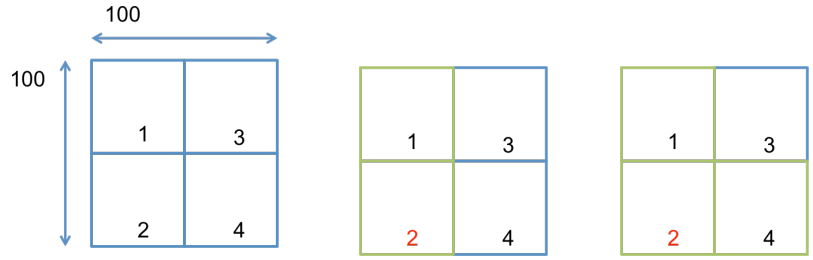
$$L_2 = \left.\begin{bmatrix} \overbrace{0}^{1-100} & \overbrace{1 \ldots 0}^{101-200} & \overbrace{0}^{201-300} & \overbrace{0}^{301-400} \\ \ddots & \ddots & \ddots & \ddots \\ 0 & 0 \ldots 1 & 0 & 0 \end{bmatrix}\right\} 100. \tag{3}$$

Similarly, we have $\tilde{L}_i$, $i = 1, 2, 3, 4$, is the matrix to select the ith local coefficient with spatial correlation from a. If we only consider one vertical neighbor of each image patch, $\tilde{L}_i \in \mathbf{R}^{200 \times 400}$ (we consider one vertical neighbor for the 2×2 patches example in Fig. 1(b). For images with more patches, we can consider the cases with more than one neighbors, thus $\tilde{L}_i \in \mathbf{R}^{(k+1)n \times Mn}$, where k is the number of neighbors, n is the number of dictionary

elements and M is the number of patches). For example, $\tilde{L}_2$ is defined as

$$\tilde{L}_2 = \begin{bmatrix} \overbrace{1 \dots 0}^{1-200} & \overbrace{0}^{201-300} & \overbrace{0}^{301-400} \\ \ddots & \ddots & \ddots \\ 0 \dots 1 & 0 & 0 \end{bmatrix} \Bigg\} 200. \tag{4}$$

Here $\tilde{L}_i$ only consider one vertical neighboring patch, however, we can design different $\tilde{L}_i$ based on how we define the neighborhood patches.



(a) Input image with 4 patches

(b) 1 vertical neighbor of patch 2

(c) 2 neighbors (one vertical and one horizontal) of patch 2

Figure 1: Examples of input image with its corresponding patches and the neighborhood for patch 2. The number in each image patch indicate the index of the patch. The neighboring patches(including patch 2) are highlighted in green color.

4 Numerical Solution

We can rewrite problem (2) as,

$$\begin{aligned} \underset{x_i, \tilde{x}_i}{\text{minimize}} \quad & \sum_{i=1}^{M} (f_i(x_i) + g_i(\tilde{x}_i)) \\ \text{subject to} \quad & x_i - L_i z = 0, \quad i = 1, \dots, M, \\ & \tilde{x}_i - \tilde{L}_i z = 0, \quad i = 1, \dots, M. \end{aligned} \tag{5}$$

where $f_i(x) = \frac{1}{2}\|b_i - Dx\|_2^2$, $g_i(x) = \lambda \sum_{g=1}^{G} \|x_g\|_2$, $x_i = \alpha_i$, $\tilde{x}_i = \tilde{\alpha}_i$ and $z = s$. Here x_i and $\tilde{x}_i$ are local variables and z is a global variable. Now problem (5) is a general form of consensus optimization problem which can be solved by ADMM [1, 2](see appendix B for the detail about the consensus optimization problem). Problem (5) is a general formulation thus x_i and $\tilde{x}_i$ can be any local components of the global variable z.

The augmented Lagrangian for (5) is

$$\begin{aligned} L_\rho(x_i, \tilde{x}_i, z, y_i, \tilde{y}_i) = \sum_{i=1}^{M} (f_i(x_i) + y_i^\top (x_i - L_i z) + (\rho_1/2)\|x_i - L_i z\|_2^2 + \\ g_i(\tilde{x}_i) + \tilde{y}_i^\top (\tilde{x}_i - \tilde{L}_i z) + (\rho_2/2)\|\tilde{x}_i - \tilde{L}_i z\|_2^2). \end{aligned}$$

with dual variables y_i and $\tilde{y}_i$. Then the ADMM algorithm is

$$
\begin{aligned}
x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}}(f_i(x_i) + (y_i^k)^\top x_i + (\rho_1/2)\|x_i - L_i z^k\|_2^2) = \mathbf{prox}_{f_i,\rho_1}(L_i z^k - (1/\rho_1)y_i^k) \\
\tilde{x}_i^{k+1} &= \underset{\tilde{x}_i}{\operatorname{argmin}}(g_i(\tilde{x}_i) + (\tilde{y}_i^k)^\top \tilde{x}_i + (\rho_2/2)\|\tilde{x}_i - \tilde{L}_i z^k\|_2^2) = \mathbf{prox}_{g_i,\rho_2}(\tilde{L}_i z^k - (1/\rho_2)\tilde{y}_i^k) \\
z^{k+1} &= \underset{z}{\operatorname{argmin}}(\sum_{i=1}^{M}(-y_i^\top L_i z + (\rho_1/2)\|x_i^{k+1} - L_i z\|_2^2 - \tilde{y}_i^\top \tilde{L}_i z + (\rho_2/2)\|\tilde{x}_i^{k+1} - \tilde{L}_i z\|_2^2) \\
y_i^{k+1} &= y_i^k + \rho_1(x_i^{k+1} - L_i z^{k+1}) \\
\tilde{y}_i^{k+1} &= \tilde{y}_i^k + \rho_2(\tilde{x}_i^{k+1} - \tilde{L}_i z^{k+1}).
\end{aligned}
$$

where the x_i, $\tilde{x}_i$, y_i and $\tilde{y}_i$ update can be implemented independently in parallel for each i, $\mathbf{prox}_{f,\rho}$ is the proximity operator of f with penalty ρ [2]. The proximity operator is defined as

$$\mathbf{prox}_{f,\rho}(v) = \underset{x}{\operatorname{argmin}}(f(x) + (\rho/2)\|x - v\|_2^2).$$

The proximity operator can be solved analytically if the function f is simple enough. The numerical derivations of the proximity operator for different functions f used in this report are show in Appendix A.1.

The z update step can be evaluated as,

$$z = Q^{-1}\sum_{i=1}^{M}(L_i^\top(x_i^{k+1} + (1/\rho_1)y_i^k) + \tilde{L}_i^\top(\tilde{x}_i^{k+1} + (1/\rho_2)\tilde{y}_i^k)),$$

where $Q = \sum_i^M(L_i^\top L_i + \tilde{L}_i^\top \tilde{L}_i)$ [2].

The pseudo code of ADMM algorithm to solve GLSC problem is illustrated as follows,

Input: $L_i, \tilde{L}_i, Q, y_i^0 = 0, \tilde{y}_i^0 = 0, z^0 = 0, i = 1, \ldots, M$
Output: z

for $k = 0, 1, \ldots$ **do**
  //$x_i, \tilde{x}_i$ update can be carried out in parallel.
  for $i = 1, \ldots, M$ **do**
    $x_i^{k+1} = \mathbf{prox}_{f_i,\rho_1}(L_i z^k - (1/\rho_1)y_i^k)$;
    $\tilde{x}_i^{k+1} = \mathbf{prox}_{g_i,\rho_2}(\tilde{L}_i z^k - (1/\rho_2)\tilde{y}_i^k)$;
  end
  $z^{k+1} = Q^{-1}\sum_{i=1}^{M}(L_i^\top(x_i^{k+1} + (1/\rho_1)y_i^k) + \tilde{L}_i^\top(\tilde{x}_i^{k+1} + (1/\rho_2)\tilde{y}_i^k))$;
  //$y_i, \tilde{y}_i$ update can be carried out in parallel.
  for $i = 1, \ldots, M$ **do**
    $y_i^{k+1} = y_i^k + \rho_1(x_i^{k+1} - L_i z^{k+1})$;
    $\tilde{y}_i^{k+1} = \tilde{y}_i^k + \rho_2(\tilde{x}_i^{k+1} - \tilde{L}_i z^{k+1})$;
  end
end
return z.

Algorithm 1: ADMM for GLSC problem

Let $u = (1/\rho)y$, we have the scaled form ADMM [1],

Input: $L_i, \tilde{L}_i, Q, y_i^0 = 0, \tilde{y}_i^0 = 0, z^0 = 0, i = 1, \ldots, M$
Output: z

for $k = 0, 1, \ldots$ **do**
  $//x_i, \tilde{x}_i$ *update can be carried out in parallel.*
  for $i = 1, \ldots, M$ **do**
    $x_i^{k+1} = \mathbf{prox}_{f_i,\rho_1}(L_i z^k - u_i^k)$;
    $\tilde{x}_i^{k+1} = \mathbf{prox}_{g_i,\rho_2}(\tilde{L}_i z^k - \tilde{u}_i^k)$;
  end
  $z^{k+1} = Q^{-1} \sum_{i=1}^{M} (L_i^\top (x_i^{k+1} + u_i^k) + \tilde{L}_i^\top (\tilde{x}_i^{k+1} + \tilde{u}_i^k))$;
  $//u_i, \tilde{u}_i$ *update can be carried out in parallel.*
  for $i = 1, \ldots, M$ **do**
    $u_i^{k+1} = u_i^k + x_i^{k+1} - L_i z^{k+1}$;
    $\tilde{u}_i^{k+1} = \tilde{u}_i^k + \tilde{x}_i^{k+1} - \tilde{L}_i z^{k+1}$;
  end
end
return z.

Algorithm 2: Scaled Form ADMM for GLSC problem

For a distributed implementation, it is often to group the local computation(i.e., the x-update and u-update), so we write ADMM as [1]

Input: $L_i, \tilde{L}_i, Q, y_i^0 = 0, \tilde{y}_i^0 = 0, z^0 = 0, i = 1, \ldots, M$
Output: z

for $k = 0, 1, \ldots$ **do**
  for $i = 1, \ldots, M$ **do**
    $//u_i, \tilde{u}_i$ *update can be carried out in parallel.*
    $u_i^{k+1} = u_i^k + x_i^{k+1} - L_i z^{k+1}$;
    $\tilde{u}_i^{k+1} = \tilde{u}_i^k + \tilde{x}_i^{k+1} - \tilde{L}_i z^{k+1}$;
    $//x_i, \tilde{x}_i$ *update can be carried out in parallel.*
    $x_i^{k+1} = \mathbf{prox}_{f_i,\rho_1}(L_i z^k - u_i^k)$;
    $\tilde{x}_i^{k+1} = \mathbf{prox}_{g_i,\rho_2}(\tilde{L}_i z^k - \tilde{u}_i^k)$;
  end
  $z^{k+1} = Q^{-1} \sum_{i=1}^{M} (L_i^\top (x_i^{k+1} + u_i^k) + \tilde{L}_i^\top (\tilde{x}_i^{k+1} + \tilde{u}_i^k))$;
end
return z.

Algorithm 3: Distributed Implementation of Scaled Form ADMM for GLSC problem

5 Group of Dictionary

Usually the group of D are given in group lasso problem. For the GLSC problem, we can learn the group information directly from the spatial information. Fig. 2 illustrate an example how to learning the group information of dictionary from the spatial information of the training data.

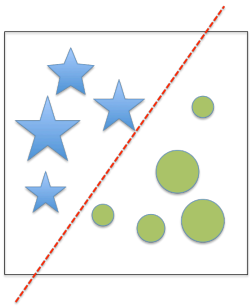


Figure 2: An example of spatial information used for dictionary learning. Here, we consider pentagons and circles are from different groups in the training image and the red dash line is a rough segmentation of the two groups. The dictionary D_1, D_2 can be learned from pentagons and circles respectively.

6 Discussion

A Numerical Derivations

A.1 Proximity operators.

The proximity operator is defined as

$$\mathbf{prox}_{f,\rho}(v) = \underset{x}{\operatorname{argmin}}(f(x) + (\rho/2)\|x - v\|_2^2).$$

A.1.1 $f(x) = \frac{1}{2}\|b - Dx\|_2^2$.

The proximity operator $\mathbf{prox}_{f,\rho}(v)$ can be solved by minimizing

$$E(x) = \frac{1}{2}\|b - Dx\|_2^2 + \frac{\rho}{2}\|x - v\|_2^2.$$

Differentiating the energy yields

$$\frac{\partial E}{\partial x} = -D^\top(b - Dx) + \rho(x - v) = 0.$$

Hence we obtain

$$x = (D^\top D + \rho I)^{-1}(D^\top b + \rho v).$$

A.1.2 $f(x) = \lambda \sum_{g=1}^{G} \|x_g\|_2,\ g = 1, \ldots, G$.

The proximity operator $\mathbf{prox}_{f,\rho}(v)$ can be solved by minimizing

$$E(x) = \lambda \sum_{g=1}^{G} \|x_g\|_2 + \frac{\rho}{2}\|x - v\|_2^2.$$

Here $E(x)$ is separable for groups $g = 1, \ldots, G$. Thus we can solve $E(x)$ for each group using block coordinate descent separately. For group l, we minimizing

$$E(x_l) = \lambda\|x_l\|_2 + \frac{\rho}{2}\|x_l - v_l\|_2^2,$$

which is equal to minimizing

$$E(x_l) = \lambda\|x_l\|_2 + \frac{\rho}{2}x_l^\top x_l - \rho v_l^\top x_l. \tag{6}$$

If $\|\rho v_l\| \leq \lambda$, then $-\rho v_l^\top x_l + \lambda\|x_l\|_2 \geq 0$ for all x_l. Thus the objective (6) is nonnegative in this case, and $x_l = 0$ solves (6).

If $\|\rho v_l\| > \lambda$, we have the subgradient equation which satisfy

$$\lambda\frac{x_l}{\|x_l\|_2} + \rho x_l - \rho v_l = 0.$$

After rearranging we have,

$$(1 + \frac{\lambda}{\rho\|x_l\|_2})x_l = v_l. \tag{7}$$

Taking the norm of both sides we have

$$\|x_l\|_2 = (\|v_l\|_2 - \frac{\lambda}{\rho})_+ \tag{8}$$

If we plug (8) into (7), we get

$$x_l = (1 - \frac{\lambda}{\rho\|v_l\|_2})_+ v_l = S_{\lambda/\rho}(v_l).$$

Thus we have

$$\mathbf{prox}_{f,\rho}(v_l) = \begin{cases} 0, & \text{if } \|\rho v_l\| \leq \lambda, \\ S_{\lambda/\rho}(v_l), & \text{otherwise.} \end{cases} \tag{9}$$

B General Form Consensus Problem

Consider the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{M} f_i(x_i) \\ \text{subject to} & x_i - \tilde{z}_i = 0, \quad i = 1, \ldots, M, \end{array} \tag{10}$$

where $x_i \in \mathbf{R}^{n_i}, i = 1, \ldots, M$, are local variables, f_i are the objective functions corresponding to x_i. Each of the local variables consists of a selection of the components of the global variable $z \in \mathbf{R}^n$. $\tilde{z}_i$ are local components of z. Intuitively, $\tilde{z}_i$ is the global variable's idea of what the local variable x_i should be.

The augmented Lagrangian for (10) is

$$L_\rho(x, z, y) = \sum_{i=1}^{M}(f_i(x_i) + y^\top(x_i - \tilde{z}_i) + (\rho/2)\|x_i - \tilde{z}_i\|_2^2),$$

with dual variable $y_i \in \mathbf{R}^{n_i}$. Then the ADMM consists of the iterations

$$
\begin{aligned}
x_i^{k+1} &= \operatorname*{argmin}_{x_i}(f_i(x_i) + (y_i^k)^\top + (\rho/2)\|x_i - \tilde{z}_i^k\|_2^2) \\
z^{k+1} &= \operatorname*{argmin}_{z}(\sum_{i=1}^{m}(-(y_i^k)^\top \tilde{z}_i + (\rho/2)\|x_i^{k+1} - \tilde{z}_i\|_2^2)) \\
y_i^{k+1} &= y_i^k + \rho(x_i^{k+1} - \tilde{z}_i^{k+1}).
\end{aligned}
$$

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Machine Learning*, 3(1):1–123, 2010.

[2] P.L. Combettes and J.C. Pesquet. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212, 2011.

[3] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[4] Ming Yuan, Ming Yuan, Yi Lin, and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.