A Brief Tutorial on Group ℓ_1-norm

Peng Xu

Pattern Recognition and Intelligent Systems Lab. (PRIS), Beijing University of Posts and Telecommunications (BUPT), Beijing, China.
peng.xu@bupt.edu.cn www.pengxu.net

Abstract. In this tutorial, firstly I introduce Group ℓ_1-norm briefly. I will also define a simple unconstrained "multi-view" category-level regression model using Group ℓ_1-norm, as application example. This regression model also can be regarded as a sparse-coding or subspace learning "framework". I will explain the "physical meaning" of the "view-selection" based on Group ℓ_1-norm. And an iterative algorithm will be illustrated clearly. Finally, the MATLAB code will be released in my Github repository.

1 Introduction

The Group ℓ_1-norm (G_1-norm) of the matrix $\mathbf{M}$ is defined as

$$\|\mathbf{M}\|_{G_1} = \sum_{i=1}^{n} \sum_{j=1}^{k} \|\mathbf{m}_i^j\|_2 \ , \tag{1}$$

where $\mathbf{m}_i^j$ the j-th segment vector in i-th column of $\mathbf{M}$.

Group ℓ_1-norm (G_1-norm) has the effect of structured sparsity and can be used to conduct the "view-selection" in multi-view learning problem [1, 2].

2 Application Example of Group ℓ_1-norm

In this section, I will define a simple multi-view subspace learning model as the application example of Group ℓ_1-norm, and give the solution algorithm.

2.1 Notations

Matrices and column vectors will be consistently denoted as bold uppercase letters and bold lowercase letters, respectively. Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, we will express its i-th row as $\mathbf{M}^i$ and j-th column as $\mathbf{M}_j$.

The Frobenius norm of the matrix $\mathbf{M}$ is defined as

$$\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^{m} \|\mathbf{M}^i\|_2^2} \ . \tag{2}$$

2.2 Problem formulation

Suppose there are n data samples, which are denoted as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$. $\mathbf{x}_i \in \mathbb{R}^d$ is formed by stacking features from k views, and the feature for each view j is a d_j dimensional vector, i.e. $d = \sum_{j=1}^{k} d_j$. And $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T \in \mathbb{R}^{n\times c}$ denotes the class label matrix and c is the amount of data categories.

Our model can be described as a minimization problem:

$$J = \min_{\mathbf{W}} \|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\|_F^2 + \lambda_1\|\mathbf{W}\|_{G_1}, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{d\times c}$ is the projection matrix for the original data domain $\mathbf{X}$. $\mathbf{W}$ contains the weights for the features from each individual view for c different categories. According to the structure of $\mathbf{X}^T$, the values of $\mathbf{W}$ can be grouped as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1^1 & \mathbf{W}_2^1 & \cdots & \mathbf{W}_c^1 \\ \mathbf{W}_1^2 & \mathbf{W}_2^2 & \cdots & \mathbf{W}_c^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_1^k & \mathbf{W}_2^k & \cdots & \mathbf{W}_c^k \end{bmatrix}, \tag{4}$$

where $\mathbf{W}_i^j \in \mathbb{R}^{d_j}$ is a weighting vector contains the weights for all features in the j-th view with respect to the i-th class.

2.3 View-Selection based on Group ℓ_1-norm

As defined in $\mathbf{W}$ and Group ℓ_1-norm of $\mathbf{W}$, all the weight vectors for all the views are organized under the ℓ_1-norm framework. Hence the interaction among all the views can be captured by the Group ℓ_1-norm regularizer, termed as "view-selection".

2.4 A mathematical solution

The designed objective function contains the non-smooth regularization terms of Group ℓ_1-norm, which is difficult to solve by general methods. Our objective function has no constraint conditions. We can use variable separation approach to derive an alternative iterative algorithm to solve it [1].

Take the derivative of the objective J with respect to $\mathbf{W}_i$ $(1 \le i \le c)$, we have [1]

$$\frac{\partial J}{\partial \mathbf{W}_i} = 2\mathbf{X}\mathbf{X}^T\mathbf{W}_i - 2\mathbf{X}\mathbf{Y}_i + \lambda_1\mathbf{D}^i\mathbf{W}_i \ , \tag{5}$$

[1] When $\|\mathbf{W}_i^j\|_2 = 0$, (3) is not differentiable. Following [3], a small perturbation can be introduced to smooth the j-th diagonal block of $\mathbf{D}^i$ as $\frac{1}{2\sqrt{\|\mathbf{W}_i^j\|_2^2+\zeta}}\mathbf{I}_j$. We set $\zeta = 1.0000e - 8$ in our following experiments.

where $\mathbf{D}^i$ is a block diagonal matrix with the j-th diagonal block as $\frac{1}{2\|\mathbf{W}_i^j\|_2}\mathbf{I}_j$, $\mathbf{I}_j$ is an identity matrix with the same size as d_j, $\mathbf{W}_i^j$ is the j-th segment of $\mathbf{W}_i$ and includes the wighting vector for the features in the j-th view.

Set $\frac{\partial J}{\partial \mathbf{W}_i} = 0$, we can get

$$\mathbf{W}_i = (2\mathbf{X}\mathbf{X}^T + \lambda_1 \mathbf{D}^i)^{-1} 2\mathbf{X}\mathbf{Y}_i \ . \tag{6}$$

We can optimize them alternatively and iteratively until convergence. During each optimization step of $\mathbf{W}$, it can be obtained column by column.

Empirically, $\mathbf{W}$ can be initialized randomly. In order to reach convergence more faster, we can initialize $\mathbf{W}$ by setting as

$$\frac{\partial \|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\|_F^2}{\partial \mathbf{W}} = 2\mathbf{X}\mathbf{X}^T\mathbf{W} - 2\mathbf{X}\mathbf{Y} = 0 \ . \tag{7}$$

Then the initial value of $\mathbf{W}$ can be set as [2]

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^T + 0.0000001I)^{-1}\mathbf{X}\mathbf{Y} \ . \tag{8}$$

The whole algorithm is described in Algorithm 1.

Algorithm 1 Iterative algorithm for this category-level regression model.

Input: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T \in \mathbb{R}^{n\times c}$.

1. Set $t = 0$. Initialize $\mathbf{W}_t$ by solving $\min_{\mathbf{W}} \|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\|_F^2$.

while not converge **do**

2. Calculate the block diagonal matrices $(D^i)_{t+1}$ $(1 \leqslant i \leqslant c)$.
where the j-th diagonal block of $(D^i)_{t+1}$ is $\frac{1}{2\|(\mathbf{W}_i^j)_t\|_2} I_j$
3. For each $\mathbf{W}_i$ $(1 \leqslant i \leqslant c)$,
$(\mathbf{W}_i)_{t+1} \leftarrow (\mathbf{X}\mathbf{X}^T + \lambda_1 (\mathbf{D}^i)_{t+1})^{-1} 2\mathbf{X}\mathbf{Y}_i$.
4. $t \leftarrow t + 1$.

end while

Output: $\mathbf{W} \in \mathbb{R}^{d\times c}$

This iterative algorithm also can be extended to solve the "view-selection" models of [1, 2].

2.5 Programming Implement

The MATLAB code with detailed annotation is available in my Github repository, https://github.com/PengBoXiangShang.

[2] A small perturbation can be introduced to ensure the matrix invertible. Here, $\mathbf{I} \in \mathbb{R}^{d\times d}$.

References

1. Wang, H., Nie, F., Huang, H.: Multi-view clustering and feature learning via structured sparsity. In: ICML. (2013)
2. Xu, P., Yin, Q., Qi, Y., Song, Y.Z., Ma, Z., Wang, L., Guo, J.: Instance-level coupled subspace learning for fine-grained sketch-based image retrieval. In: ECCV workshops. (2016)
3. Gorodnitsky, I.F., Rao, B.D.: Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. IEEE Transactions on Signal Processing **45**(3) (1997) 600–616