Tree Guided Learning for Structured Sparsity

Avinash Bukkittu, Varsha Maragi

Columbia University

ab4377@columbia.edu, vgm2115@columbia.edu

April 28, 2017

Overview

1. Motivation
2. Grouped Tree Structure Regularization
3. Experimental Setup
4. Results
5. Conclusion and Future Work

Motivation (1)

- In typical high dimensional data not all features are important. Thus, we strive for a sparse solution using sparsity inducing norms.
- L1-norm (i.e Lasso penalty) is commonly used as a sparsity inducing norm. However, Lasso fails to capture the inherent structure in the features.
- To overcome this, people have suggested ways to put features into discrete groups (Group Lasso)

Motivation(2)

Group Lasso objective function

$$\min_{\beta} \frac{1}{2}||X\beta - Y||_2^2 + \lambda \sum_{j=0}^{g} ||\beta_{G_j}||_2$$

A few points

- The inclusion/exclusion of a group in an all in/all out fashion prevents overlapping of features
- This association of a feature to only a single group limits us from imposing a structure over the features
- This motivated us to look for extensions to Group Lasso where the groups are overlapping and can be structured.
- We demonstrate a method to induce structured sparsity in high dimensional data using a tree-guided regularization.

Grouped Tree Structure Regularization

Tree guided regularization is defined as

$$\phi(\boldsymbol{\beta}) = \sum_{i=0}^{d} \sum_{j=1}^{n_i} w_j^i ||\boldsymbol{\beta}_{G_j^i}||_2 \text{ where } \boldsymbol{\beta} \in \mathbb{R}^p$$

- w_j^i are the pre-defined weights
- d is the depth of the tree
- n_i is the $\#$ of nodes at depth i

Grouped Tree Structure Regularization

The objective function with grouped tree structured regularization is

$$\min_{\beta} loss(\beta) + \lambda \sum_{i=0}^{d} \sum_{j=1}^{n_i} w_j^i ||\beta_{G_j^i}||_2$$

- *loss* can be any convex loss function
- The nodes of the tree represent the grouping structure
- The weights for each nodes should be pre-defined
- The features are at the leaf nodes
- Nodes at a particular level do not overlap.
- A parent node is the superset of its child nodes.

An example

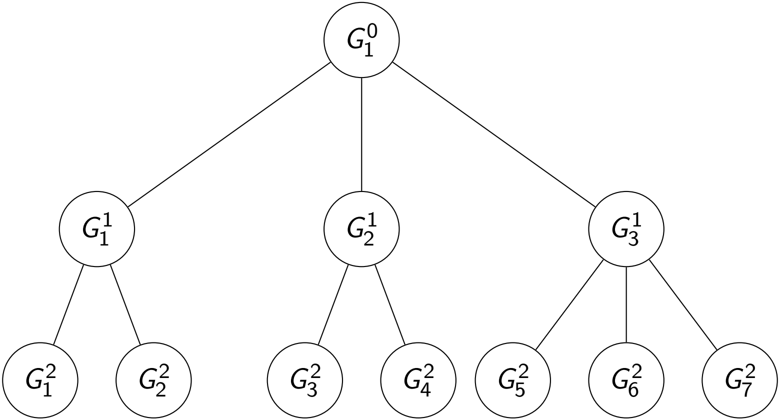

Figure: An example for tree-guided regularization. Here, $G_1^2 = \{\beta_1\}$, $G_2^2 = \{\beta_2\}$, $G_2^2 = \{\beta_3\}$,...,$G_7^2 = \{\beta_7\}$, $G_1^1 = \{\beta_1, \beta_2\}$ and so on.

Moreau-Yosida Regularization

The Moreau-Yosida regularization associated with the grouped tree structure is

$$\phi_\lambda(\boldsymbol{v}) = \min_{\beta} \frac{1}{2}||\boldsymbol{\beta} - \boldsymbol{v}||_2^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{n_i} w_j^i ||\boldsymbol{\beta}_{G_j^i}||$$

[Jun et. al.,2010] show how to solve the objective function using the solution to the above Moreau-Yosida regularization

Experimental Setup

- **JAFFE**[1] - A dataset of 213 images with 10 subjects each having six facial expressions
- Expressions include Happy, Sad, Surprise, Disgust, Anger & Fear
- Experts have classified each of the images into one of the six categories and have also rated each image on a scale of 5 for each expression

Our task - We applied tree-guided regularization on these images to see the improvements

[1] http://www.kasrl.org/jaffe.html

Why do we need a sparse solution?

- If we consider every pixel as a feature, then not all features are useful.
- This implies the weights associated with most of the pixels should be zero.
- This idea promotes sparsity in the solution.
- The question is - Can these features be structured? If yes, we would aim for a sparse & **structured** solution

Exploiting tree structure in the data

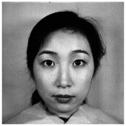

Figure: Original Image

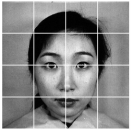

Figure: Divided into 4x4 blocks

Figure: Divided into 16x16 blocks

Result 1 - Regression(Square loss)

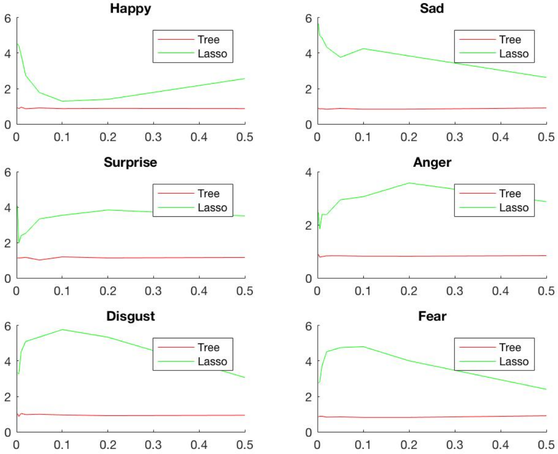

Figure: Comparison of tree guided sparsity with lasso. X-axis are the λ values, Y-axis are the RMSE values

Result 2 - Classification(Logistic loss)

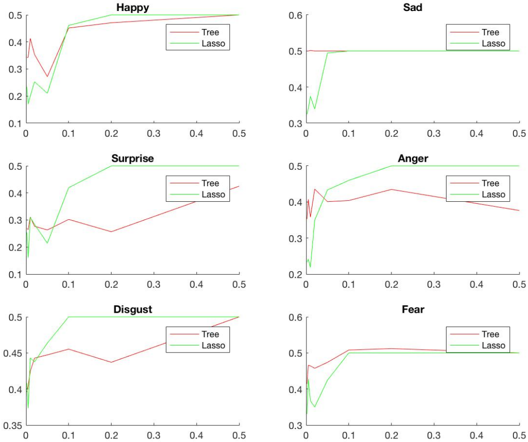

Figure: Comparison of tree guided sparsity with lasso. X-axis are the λ values, Y-axis are the Balanced Error Rates(BER)

Multi-task learning

- So far we imposed a tree structure on the features for a regression/classification task
- We can extend this idea to multi-task regression models where the k-regression tasks are related via a tree like structure.

Multi-task learning (2)

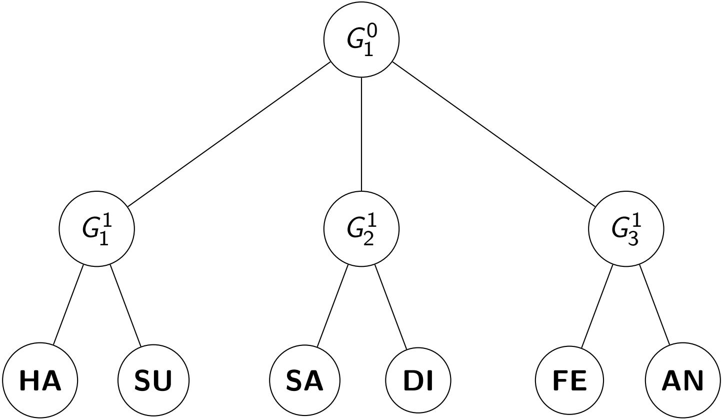

Figure: Multi-task learning applied to JAFFE dataset. Similar expressions are grouped together.

Result 3 - Multi-task learning

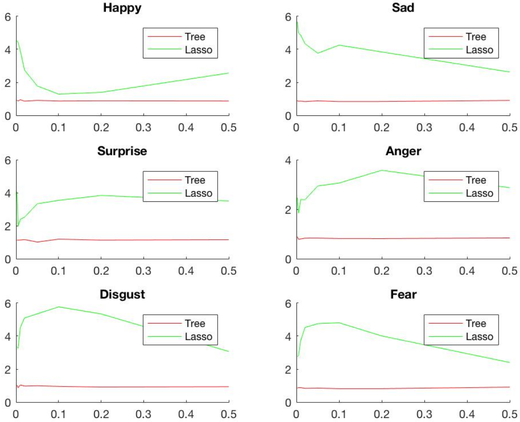

Figure: Comparison of tree structured multi-task learning vs lasso penalty (no structure assumption among the output variables)

Why should the method work?

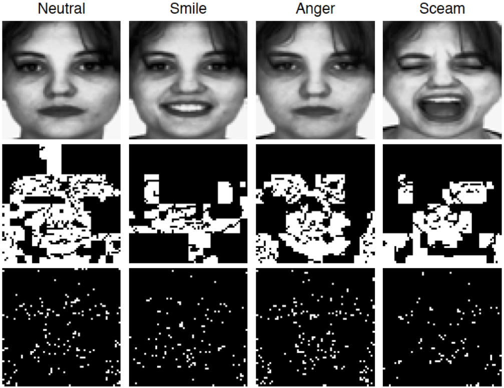

Figure: First row are the markers obtained for tree structured regularization. Second row corresponds to markers from lasso penalty.

Conclusion and Future Work

- Sparse solutions which exploit the hidden structure of the input variables is better than simply striving for a sparse solution
- Structure can be assumed over input features or over the output variables in case of multi-task regression
- An interesting area to look would be to incorporate both the structure of both input variables and output variables for predictions tasks.

References

Jun Liu & Jieping Ye (2010)
Moreau-Yosida Regularization for Grouped Tree Structure Learning
Advances in Neural Information Processing Systems 1459–1467

Seyoung Kim & Eric P. Xing
Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity

SLEP - Sparse Learning with Efficient Projections
http://yelab.net/software/SLEP/