

# Techniques for group-wise feature selection and estimation

Aditya Chindhade  
Carnegie Mellon University  
Master's Project Report  
2018

Advisor: Prof. Nikolaos V. Sahindis

# Contents

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Regularization and feature selection</b>	<b>3</b>
3.1	Overview . . . . .	3
3.2	The Ridge . . . . .	3
3.3	The Lasso . . . . .	4
3.4	The Elastic Net . . . . .	5
3.5	The Group-Lasso . . . . .	6
3.6	Best subset selection . . . . .	7
<b>4</b>	<b>The ALAMO approach</b>	<b>8</b>
4.1	Surrogate Model Generation . . . . .	8
4.2	Adaptive Sampling . . . . .	8
4.3	Constrained regression in ALAMO . . . . .	8
<b>5</b>	<b>Group-sparsity in neural networks</b>	<b>10</b>
5.1	Neural networks and deep learning . . . . .	10
5.2	Overfitting and regularization . . . . .	10
5.3	Group lasso penalty in a neural network . . . . .	10
<b>6</b>	<b>Experiments</b>	<b>11</b>
6.1	Dataset description . . . . .	11
6.2	Feature transformations . . . . .	11
6.3	Correlation plot . . . . .	12
6.4	Modeling . . . . .	12
6.4.1	Elastic Net . . . . .	12
6.4.2	Group Lasso . . . . .	12
6.4.3	ALAMO . . . . .	12
6.4.4	Group-sparse neural network . . . . .	12
<b>7</b>	<b>Results and Discussion</b>	<b>13</b>
7.1	Elastic Net . . . . .	13
7.2	Group Lasso . . . . .	14
7.3	ALAMO . . . . .	16
7.4	Group-lasso penalized neural network . . . . .	17
<b>8</b>	<b>Conclusion</b>	<b>18</b>

# 1 Abstract

This work presents a novel integer-programming based feature selection technique for constrained regression and comparative benchmarks with group-wise feature selection techniques. These techniques are particularly useful for incorporating prior knowledge about the structure of features in terms of hierarchy and group-wise dependence in the model building process. The uniqueness of the approach is the superior control in terms of specifying groups of features, thereby allowing the modeler to specify dependence between features and include those dependencies in the model building process. Apart from having a higher accuracy in comparison with the group-lasso, this novel approach also converges in fewer iterations. Utilizing an iterative adaptive sampling approach optimized by internal non-linear optimization solvers, this approach penalizes model complexity by using Akaike information criterion, thereby ensuring a simpler and more interpretable model while maintaining high accuracy. These capabilities have been incorporated in an integer-programming solver, ALAMO.

The paper also includes benchmarks with the group-lasso penalty on a neural network to obtain sparsity in terms of hidden units, thereby incorporating structured-sparsity based regularization in a neural network. This not only solves the problem of overfitting, but also makes the network sparser, thereby making predictions faster using the sparse network. This serves as a deterministic equivalent of the dropout, allowing control on setting individual weights within a hidden layer to zero.

All benchmarks are tested out on the low birth weight dataset, a popular dataset involving group-wise structure in the predictors responsible for affecting the weight of a new born child. This is primarily modeled as a regression problem.

## 2 Introduction

Some of the major challenges in machine learning approaches include overfitting, high dimensionality and lack of interpretability. The aim of regularization is to mainly to avoid overfitting and in some cases also to reduce dimensionality. Feature selection techniques aim to incorporate model interpretability by selecting predictors either based upon statistical significance or upon prior knowledge of the problem at hand. Sparsity-based regularization techniques refer to the approaches for setting the coefficients corresponding to certain predictors to exactly zero, thus providing faster computation, dimensionality reduction and improving model interpretability.

The first regularization technique dates back to 1970, when Hoerl et.al.[1] incorporated the  $\ell_2$  norm to shrink the coefficients of a regression solution. The Lasso[2] technique was invented 1996 and became one the highest cited papers in modern statistics and machine learning. Unlike the Ridge which aims at shrinking individual coefficients in the regression problem, the lasso acts as a variable selection tool by incorporating sparsity in the model.

The elastic net, formulated in 2005 [3] aims at achieving a balance between the ridge and the lasso, thus doing two jobs: variable selection and shrinkage. The elastic net has quickly grown to become a popular tool in modeling financial modeling due to its unique functionality [4].

Yuan and Lin [5] invented the group-lasso technique for incorporating prior knowledge about the group-wise structure of predictors. The model has occurred multiple times in highly cited articles [6, 7] and has secured a unique position in the statistical machine learning community, owing to its fundamental nature and utility. The following section describes these models in further detail.

### 3 Regularization and feature selection

#### 3.1 Overview

#### 3.2 The Ridge

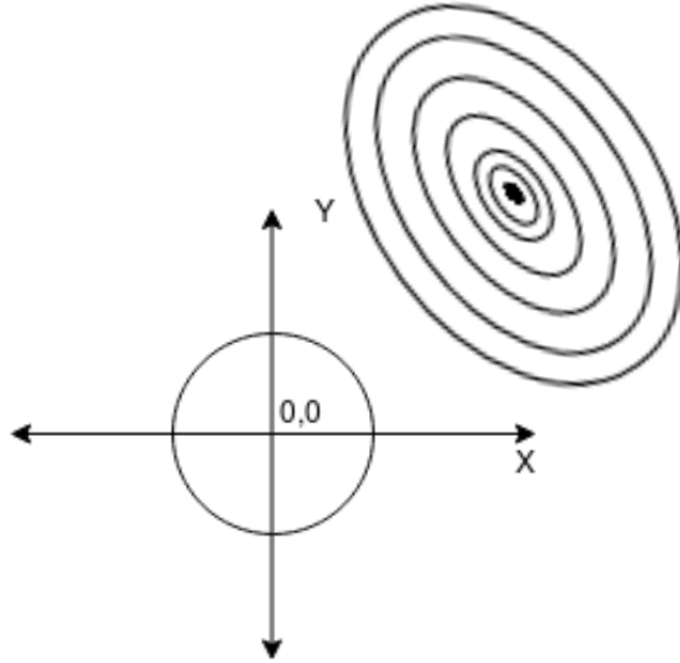


Figure 1: The ridge penalty as a constrained minimization problem

The ridge utilizes the  $L_2$  norm of the coefficient vector. It is responsible for shrinkage of coefficients. The degree of shrinkage is controlled by  $\lambda$ . It is a convex penalty and hence is tractable. The optimization problem for the ridge can be formally stated as:

$$\min_{\beta \in R^p} \left( \|y - \beta_0 \mathbf{1} - \sum_{j=1}^J \mathbf{X}_j \beta_j\|_2^2 + \lambda \|\beta\|_2^2 \right) \quad (1)$$

### 3.3 The Lasso

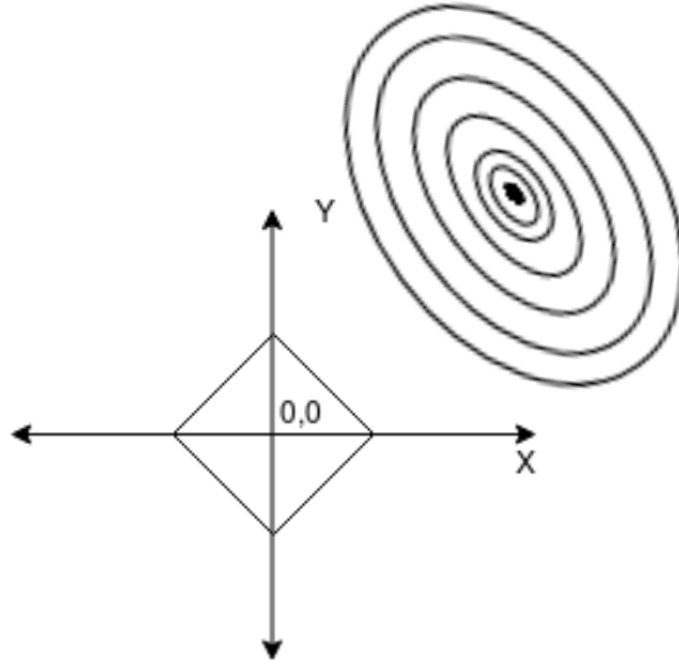


Figure 2: The lasso penalty as a constrained minimization problem

The ridge utilizes the  $L_2$  norm of the coefficient vector. It is responsible for sparsity in the coefficient vector by setting coefficients to 0. The degree of sparsity is controlled by  $\lambda$ . Similar to the ridge, the lasso is a convex penalty and hence is tractable. The optimization problem for the ridge can be formally stated as:

$$\min_{\beta \in R^p} \left( \|\mathbf{y} - \beta_0 \mathbf{1} - \sum_{j=1}^J \mathbf{X}_j \beta_j\|_2^2 + \lambda \|\beta\|_1 \right) \quad (2)$$

### 3.4 The Elastic Net

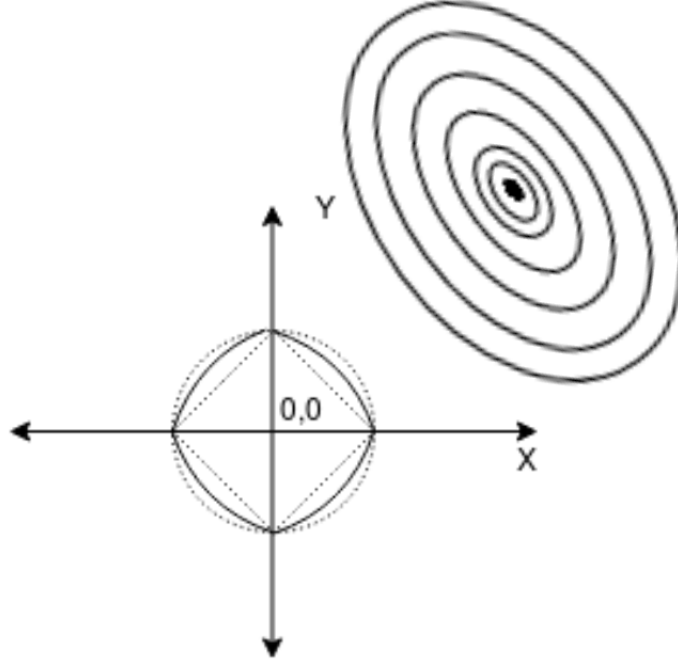


Figure 3: The elastic net penalty as a constrained minimization problem

The objective function for the elastic net penalized linear model is defined as:

$$\min_{\beta \in R^p} \left( \|\mathbf{y} - \beta_0 \mathbf{1} - \sum_{j=1}^J \mathbf{X}_j \beta_j\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 \right) \quad (3)$$

The elastic net [3] can be considered as an intermediate between the ridge and the lasso, addressing the shrinkage and variable selection problems simultaneously. In this representation,  $\mathbf{y}$  is the target variable,  $\mathbf{X}_\ell$  is the input matrix,  $\beta_0$  is the weight of the bias term and  $\beta_j$  is a vector corresponding to the weight of each predictor variable.  $\lambda_1$  is the penalty for the  $\ell_1$  norm and  $\lambda_2$  is the penalty for the  $\ell_2$  norm. The relative importance of the lasso (variable selection) and the ridge (weight shrinkage) is governed by the  $\lambda_1$  and  $\lambda_2$  values. The requirement of two hyper-parameters can be reduced to one using a relative weight  $\alpha$ , given by:

$$\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

The objective function with a single hyper-parameter then becomes:

$$\min_{\beta \in R^p} \left( \|\mathbf{y} - \beta_0 \mathbf{1} - \sum_{j=1}^J \mathbf{X}_j \beta_j\|_2^2 + (1 - \alpha) \|\beta_j\|_1 + \alpha \|\beta_j\|_2^2 \right) \quad (4)$$

### 3.5 The Group-Lasso

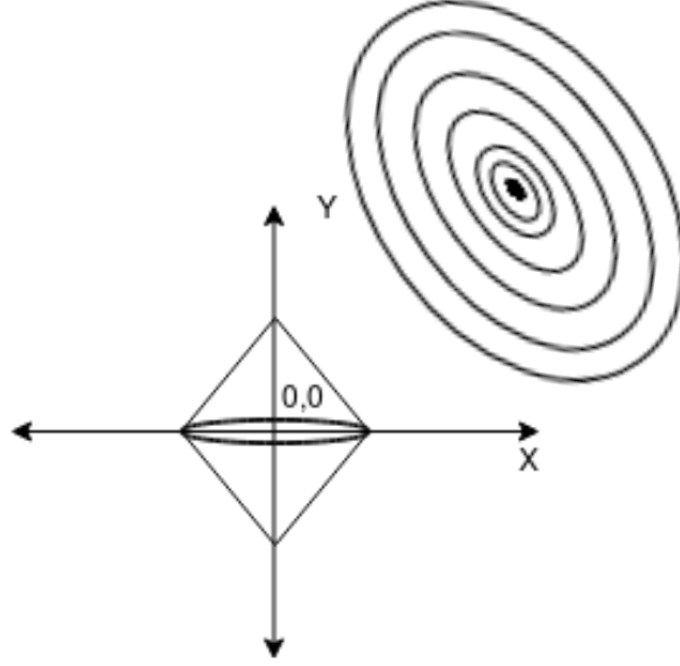


Figure 4: The group lasso penalty as a constrained minimization problem

Two-step penalty Outer penalty - L1 - selecting feature groups Inner penalty - L2 - shrinkage of coefficients within selected groups.  $L+1$  parameters: hyperparameters or inversely proportional to group size

The group lasso [5] penalized linear model has the following objective function:

$$\min_{\beta \in R^p} \left( \|y - \beta_0 \mathbf{1} - \sum_{\ell=1}^L \mathbf{X}_{\ell} \beta_{\ell}\|_2^2 + \lambda \sum_{\ell=1}^L \sqrt{p_{\ell}} \|\beta_{\ell}\|_2 \right) \quad (5)$$

In this representation,  $y$  is the target variable,  $\mathbf{X}_{\ell}$  is the input matrix,  $\beta_0$  is the weight of the bias term and  $\beta_{\ell}$  is a vector corresponding to the weight of each predictor variable across all the groups.  $\lambda$  is the penalty for the group lasso and  $\sqrt{p_{\ell}}$  is the penalty proportional to the size of each group  $\ell$ . This ensures normalizing the penalty according to the size of a group. The  $\ell_2$  penalty (also known as the ridge) is applied within a group to reduce the magnitude of each of the weights within a group. These norms, being positive are directly added up and effectively work as an  $\ell_1$  norm across the groups of weights. The  $\ell_1$  norm (also known as the lasso) is responsible for selecting groups of variables and can set an entire group-wise coefficient to zero, thereby inducing sparsity in the model.



### 3.6 Best subset selection

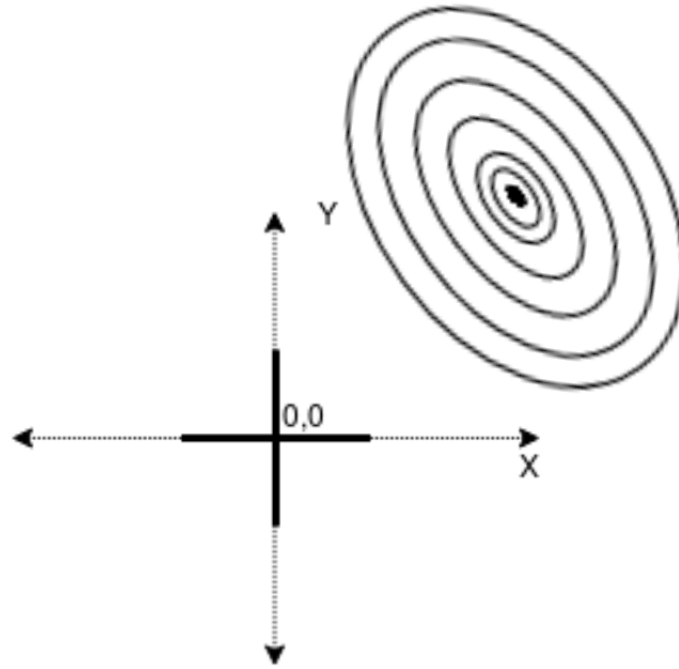


Figure 5: Best subset selection as a constrained minimization problem

The best subset technique for feature selection is essentially an exhaustive search across all feature subsets to find the feature subset that minimizes the loss function with respect to the features vector consisting of the feature subset selected. While this isn't exactly the same as calculating a vector norm, it is known as the  $L_0$  norm owing to the effective feature set obtained. Each feature can either be included in the model or excluded from it. This leads to the following relation:

$$\text{Number of subsets} = 2^{\text{number of features}} \quad (6)$$

Being exponential in the number of features, the best subset selection technique is often considered intractable. However, due to the exhaustive search across feature subsets to find the best subset, this technique is widely considered as the holy grail of statistics and has been of interest lately due to advances in computing and enhancements in memory.

## 4 The ALAMO approach

ALAMO [8, 9, 10] is a black-box modeling toolbox developed at the Sahinidis Optimization Group, Carnegie Mellon University. ALAMO implements an integer-programming based best-subset selection strategy for variable selection which utilizes the hallowed  $\ell_0$  norm for variable selection and solves non-convex the optimization problem using derivative-free optimization solvers. The ALAMO approach relies on two key steps:

### 4.1 Surrogate Model Generation

The solution approach begins by sampling an initial set of points from the input data set, which are then used for building an initial model with, starting with the lowest complexity. Various combinations of simple basis functions are considered for generating algebraic models, which are then solved using an optimization framework.

### 4.2 Adaptive Sampling

Adaptive sampling is an active learning [11, 12] approach which involves querying the data for selecting the next set of points such that maximum model accuracy can be obtained with minimum additional information. This is done by an error-maximization strategy, which means that the points which have the largest deviation from the model are included in the next sample for model building. The estimate can also be made robust (insensitive to outliers) by setting a threshold on the maximum deviation allowed.

### 4.3 Constrained regression in ALAMO

ALAMO allows manual specification of constraints on features and includes specification of feature groups. These feature groups could be defined by either polynomial transformations of the same feature forming a group or based on prior knowledge about heirarchical structure within the features.

- REQ: If group  $i$  is selected, group  $j$  is also selected.
- XCL: If group  $i$  is selected, group  $j$  is NOT selected and vice-versa.
- NMT: Not more than  $k$  variables within group  $i$  must be selected.
- ATL: At least  $k$  variables within group  $i$  must be selected.

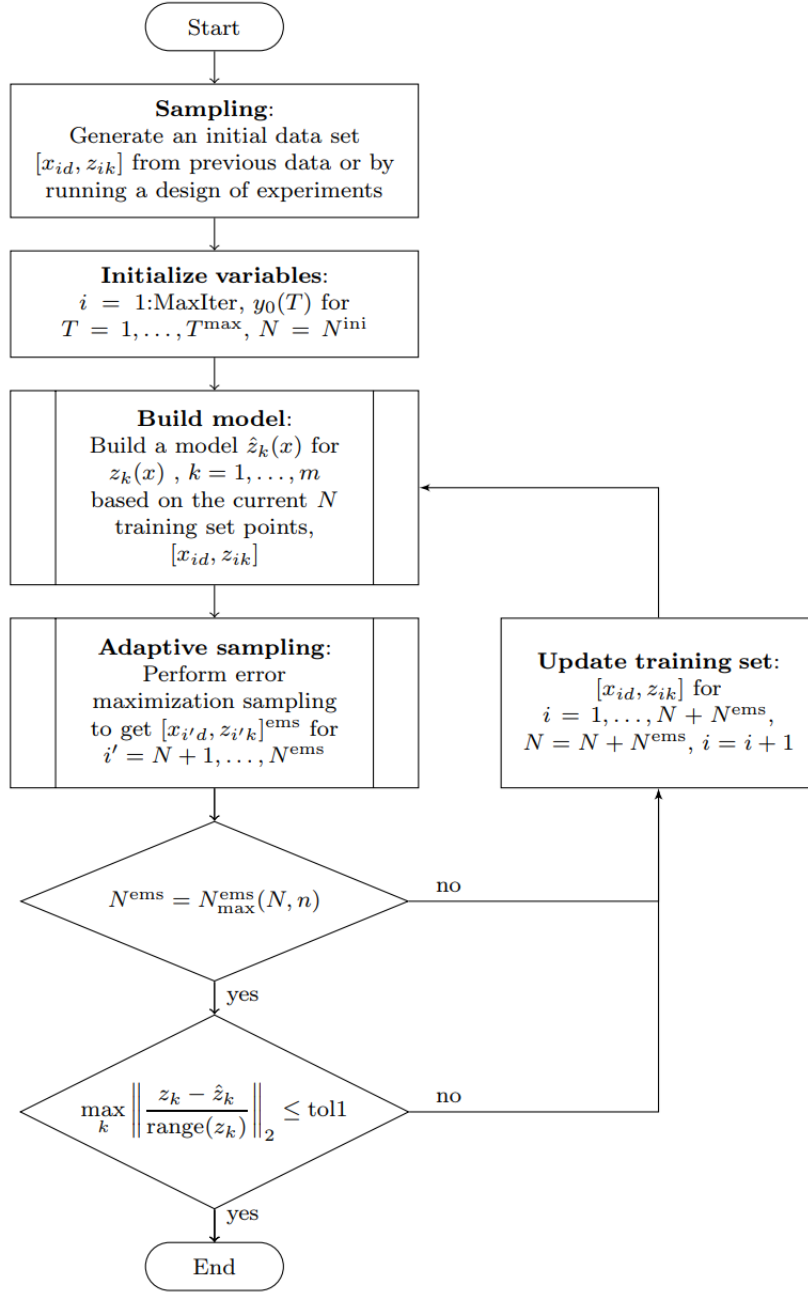


Figure 6: The ALAMO approach

## 5 Group-sparsity in neural networks

### 5.1 Neural networks and deep learning

Explain neural networks and their working in brief Although neural networks were conceptualized in the mid 20<sup>th</sup> Century, it wasn't until the recent advances in computing power that they came into vogue and remarkable accuracies[13]were achieved due to addition of more hidden units or stacking more layers. However, the presence of too many parameters or weights quickly led to the model overfitting to the training data. To solve this problem, dropout[14], a method which involves setting neurons inactive in a stochastic manner was adopted in popular platforms. However, the inherent stochastic nature of the dropout leads to a lack of reliability and interpretability.

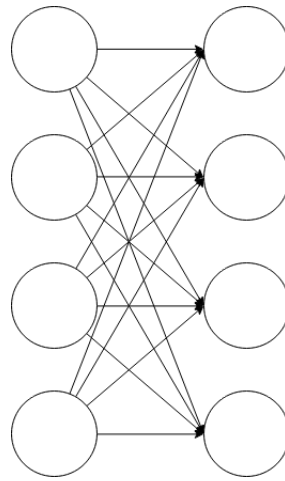


Figure 7: Neural network representation

### 5.2 Overfitting and regularization

Additionally, it was proven [15] that in most cases, only a small fraction of the learned weights contribute to the accuracy and a large number of weights can be set to zero, without significantly affecting the accuracy of the model.

### 5.3 Group lasso penalty in a neural network

More recently, group sparse regularization techniques have been applied to incorporate sparsity in the weights of a neural, thus incorporating structure in the sparsification and thereby increasing the interpretability of the model. This method serves as an additional proof to the proposition that most of the learned weights could be set to zero, by comparing accuracy with number of active neurons.

## 6 Experiments

### 6.1 Dataset description

The data-set under consideration was first published in a book - Applied Logistic Regression by Hosmer and Lemeshow [16] and consists of the data were collected at Baystate Medical Center, Springfield, Mass during 1986. The data in the raw format can be found in popular repositories like MLData (<http://www.mldata.org/repository/data/viewslug/uci-20070111-lowbwt/>). Basic transformations on the features lead to the following features:

age1,age2,age3:Orthogonal polynomials of first, second, and third degree representing mothers age in years (float)

lwt1,lwt2,lwt3:Orthogonal polynomials of first, second, and third degree representing mothers weight in pounds at last menstrual period (float)

white,black:Indicator functions for mothers race; "other" is reference group (boolean)

smoke:Smoking status during pregnancy (boolean)

ptl1,ptl2m:Indicator functions for one or for two or more previous premature labors, respectively. No previous premature labors is the reference category. (integer)

ht:History of hypertension (boolean)


ui:Presence of uterine irritability (boolean)

ftv1,ftv2,ftv3m:Indicator functions for one, for two, or for three or more physician visits during the first trimester, respectively. No visits is the reference category. (boolean)

### 6.2 Feature transformations

Additional features corresponding to polynomial transformations One-hot encoding for categorical features Mean-normalization for continuous

Feature groups are identified based on either polynomial transformations of the same primary variable or prior knowledge about association



	age1	age2	age3	lwt1	lwt2	lwt3	white	black	smoke	ptl1	ptl2m	ht	ui	ftv1	ftv2	ftv3m	bwt
0	-0.058334	0.011046	0.029562	0.124463	-0.021339	-0.130731	0	1	0	0	0	0	1	0	0	0	2.523
1	0.134366	0.055246	-0.096907	0.060067	-0.069228	-0.033348	0	0	0	0	0	0	0	0	0	1	2.551
2	-0.044570	-0.009415	0.045089	-0.059184	0.037463	0.004618	1	0	1	0	0	0	0	1	0	0	2.557
3	-0.030806	-0.026244	0.052490	-0.052029	0.023907	0.019035	1	0	1	0	0	0	1	0	1	0	2.594
4	-0.072099	0.035142	0.004822	-0.054414	0.028324	0.014572	1	0	1	0	0	0	1	0	0	0	2.600

Figure 8: Identification of feature groups

## 6.3 Correlation plot

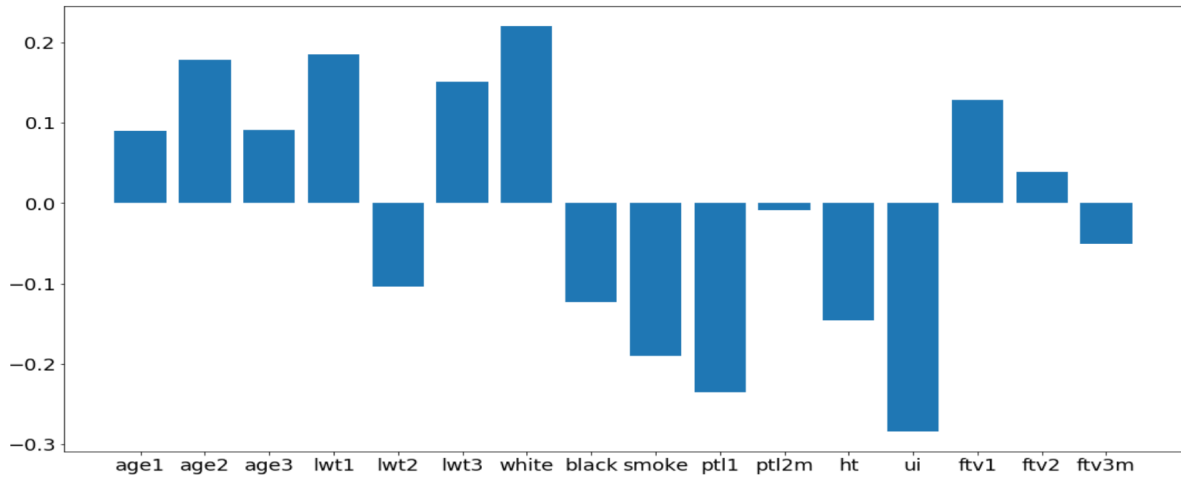


Figure 9: Correlation plot: Pearson correlation of each feature with target variable

## 6.4 Modeling

The following models were used for fitting the data and identifying which features were selected

### 6.4.1 Elastic Net

The Elastic Net was implemented using the glmnet framework in R.

### 6.4.2 Group Lasso

The Group Lasso was implemented using the grpreg framework in R.

### 6.4.3 ALAMO

ALAMO version 2018.6.20 was downloaded from The Optimization Firm website: [www.minlp.com](http://www.minlp.com)

### 6.4.4 Group-sparse neural network

The group lasso was implemented in a neural network using the source code by [17] and a 4 layer deep neural network was built using Keras with Tensorflow backend.

## 7 Results and Discussion

### 7.1 Elastic Net

Elastic net regularization is implemented using the 'glmnet' package in R. The elastic net penalty, by nature, does not account for group-wise variable selection. As a result, sparsity can be observed in the model, however, the sparsity is unstructured and does not take into account the group-wise structure of the predictors. The sparsity pattern can be observed as follows:

Penalty	0.01	0.05	0.1	0.2	0.5	1
(Intercept)	3.04399682125432	2.99862111924456	2.97451861554578	2.94729609773121	2.9445873015873	2.9445873015873
age1	0	0	0	0	0	0
age2	1.44051051842287	0.893538133354436	0.305271674803786	0	0	0
age3	0.78086581048347	0.277461839300162	0	0	0	0
lwt1	1.72064278608912	0.922138791531523	0.188841821495976	0	0	0
lwt2	0	0	0	0	0	0
lwt3	1.23031669045937	0.643316528641855	0.0127156281429045	0	0	0
white	0.283580137772934	0.240028719523993	0.1315299109623	0	0	0
black	-0.128034497587724	-0.0156065518331853	0	0	0	0
smoke	-0.264225200231336	-0.198028390816008	-0.0912396753272265	0	0	0
ptl1	-0.284813413430795	-0.225704188508818	-0.14210767582366	0	0	0
ptl2m	0.153956928628921	0	0	0	0	0
ht	-0.516322737244609	-0.306829255474887	-0.0545396147704061	0	0	0
ui	-0.452482179034149	-0.360364466248038	-0.266681936403788	-0.0182843739713636	0	0
ftv1	0.0696453392572871	0.0211442404548328	0	0	0	0
ftv2	0	0	0	0	0	0
ftv3m	-0.135558951958573	0	0	0	0	0

Table 1: Sparsity pattern for the elastic net

From the sparsity pattern of the elastic net penalized linear model and from the correlation plot in the previous section, it can be seen that the coefficients that are set to 0 are the ones which are least correlated with the target variable. At low values of  $\lambda$ , the least correlated features are set to 0. However, as the value of  $\lambda$  increases, more and more coefficients are set to 0 and the order in which this is done is governed by the correlation of the feature with the target variable.

## 7.2 Group Lasso

Upon running a sweep across different group-regularization penalty values from 0.01 to 0.19, it is found that as the penalty goes on increasing, the group-lasso excludes more and more groups from the model, thus inducing more sparsity in the model.

It was also observed that as more and more groups are excluded from the model, the MSE goes on increasing, due to an increase in the bias.

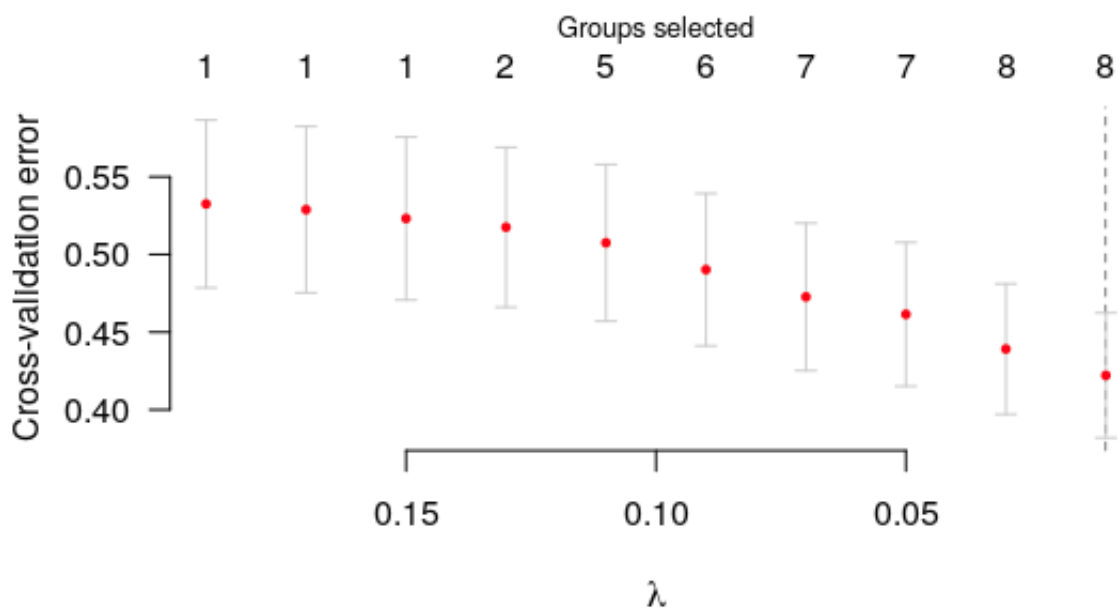


Figure 10: Cross-validation error vs penalty

A trade-off between MSE and model interpretability is important. For example, the lowest MSE occurs when all 8 groups are selected, which means that unwanted predictors such as number of physician visits influence the output weight of the child. Hence, we settle for 7 groups, although it gives a slightly higher MSE than that with 8 groups. This occurs at a penalty value of 0.04. With 7 groups, we get an MSE of 0.44, which is comparable to the one found in the original group lasso paper.



The following table demonstrates the group-wise sparsity induced in the model:

Penalty	0.01	0.03	0.04	0.05	0.07	0.09	0.11	0.13	0.15	0.17	0.19
(Intercept)	3.0434	3.0369	3.0349	3.0289	3.0176	3.0073	2.9964	2.9794	2.9681	2.9598	2.9515
age1	0.0055	0.1225	0.1450	0.1407	0.0824	0	0	0	0	0	0
age2	1.3883	0.9979	0.8113	0.6259	0.2669	0	0	0	0	0	0
age3	0.8102	0.6025	0.4936	0.3767	0.1568	0	0	0	0	0	0
lwt1	1.6682	1.1717	0.9478	0.7468	0.3743	0.0342	0	0	0	0	0
lwt2	-0.0180	-0.1336	-0.1576	-0.1584	-0.1137	-0.0134	0	0	0	0	0
lwt3	1.2186	0.8884	0.7292	0.5828	0.2969	0.0273	0	0	0	0	0
white	0.2732	0.2308	0.2093	0.1835	0.1328	0.0817	0.0217	0	0	0	0
black	-0.1337	-0.0916	-0.0746	-0.0610	-0.0382	-0.0207	-0.0064	0	0	0	0
smoke	-0.2634	-0.2277	-0.2101	-0.1878	-0.1453	-0.1040	-0.0531	-0.0077	0	0	0
ptl1	-0.2736	-0.2265	-0.1997	-0.1742	-0.1190	-0.0598	-0.0004	0	0	0	0
ptl2m	0.1840	0.1105	0.0817	0.0570	0.0224	0.0044	0.0000	0	0	0	0
ht	-0.5108	-0.4015	-0.3493	-0.2977	-0.1982	-0.1023	-0.0223	0	0	0	0
ui	-0.4586	-0.4174	-0.3988	-0.3805	-0.3466	-0.3153	-0.2678	-0.2147	-0.1590	-0.1027	-0.0464
ftv1	0.0688	0.0211	0	0	0	0	0	0	0	0	0
ftv2	0.0215	0.0081	0	0	0	0	0	0	0	0	0
ftv3m	-0.1159	-0.0268	0	0	0	0	0	0	0	0	0

Table 2: Sparsity pattern for the group lasso

The sparsity pattern upon mean-normalization of the data can be seen as follows:

Penalty	0.000000	0.001000	0.002000	0.003000	0.004000	0.005000	0.006000	0.007000	0.008000	0.009000	0.010000
(Intercept)	0.408308	0.407470	0.406824	0.408491	0.413713	0.419661	0.425577	0.431550	0.437533	0.442420	0.447314
V1	-0.050094	-0.045197	-0.040482	-0.035889	-0.032678	-0.029972	-0.027733	-0.025572	-0.023512	-0.022112	-0.020708
V2	-0.142799	-0.137403	-0.131842	-0.125891	-0.120009	-0.114354	-0.108833	-0.103255	-0.097624	-0.092039	-0.086413
V3	-0.162437	-0.154920	-0.147381	-0.140226	-0.132939	-0.125760	-0.118750	-0.111769	-0.104824	-0.098055	-0.091343
V4	0.016164	0.010402	0.004833	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V5	-0.003375	-0.002185	-0.001065	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V6	0.004636	0.003334	0.001751	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V7	0.012781	0.010961	0.009152	0.007400	0.005382	0.003669	0.002398	0.001159	0.000000	0.000000	0.000000
V8	-0.005466	-0.004499	-0.003591	-0.002819	-0.002523	-0.002011	-0.001383	-0.000703	0.000000	0.000000	0.000000
V9	-0.008701	-0.006843	-0.005014	-0.003258	-0.001283	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V10	0.003976	0.003415	0.002866	0.002194	0.001580	0.001123	0.000975	0.000843	0.000729	0.000727	0.000719
V11	0.095628	0.091209	0.086667	0.081750	0.077499	0.073443	0.069734	0.066031	0.062324	0.058339	0.054358
V12	0.003376	0.001643	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V13	0.003473	0.001552	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V14	0.003564	0.002546	0.001467	0.000347	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V15	-0.001652	-0.001033	-0.000506	-0.000090	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
V16	0.010427	0.007143	0.003934	0.000870	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
errors	0.036471	0.034605	0.033251	0.032387	0.031968	0.031590	0.031285	0.031017	0.030780	0.030574	0.030401

Table 3: Sparsity pattern of the elastic net upon mean-normalization

It can be observed that the penalties have reduced by an order of magnitude to achieve the same level of sparsity in the model. This is due to the reduction in magnitude of the elements of the decision matrix.

### 7.3 ALAMO

Upon running the same experiment with ALAMO, with the constraints that the groups exist as specified in the R program, with an ATLEAST requirement on all groups except the number of physician visits. ALAMO, too, excludes this group from the model when the ATL constraint is excludes for this particular group.

Additionally, it is observed that ALAMO gave a slightly better MSE of 0.402 as compared to 0.44 with 'grpreg' in R. Both are run on 4-fold cross-validations. The weights learned by ALAMO are also quite different from those learned by grpreg.

Model	MSE
(age1, age2, age3, lwt1, lwt2, lwt3, white)	0.559
(age1, age2, age3, lwt1, lwt2, lwt3, white,1)	0.0733
(age1, age2, age3, lwt1, lwt2, lwt3, white, ui, 1)	0.0684
(age1, age2, age3, lwt1, lwt2, lwt3, white, ht, ui, 1)	0.0646
(age1, age2, age3, lwt1, lwt2, lwt3, white, smoke, ht, ui, 1)	0.0614

Table 4: Solution paths of ALAMO

The linear model can be found to be as follows:

$$Y(\text{estimate}) = -0.30 * \text{age1} + 0.46 * \text{age2} + 0.35 * \text{age3} + 0.63 * \text{lwt1} + 0.16 * \text{lwt2} + 0.42 * \text{lwt3} + 0.15 * \text{white} - 0.13 * \text{smoke} - 0.26 * \text{ht} - 0.22 * \text{ui} + 1.1$$

An interesting observation is that in case of log-transformation, ALAMO selects the same model with or without specifying the group constraints. As is evident from both instances, both models leave out ftv from the regression, but ALAMO has a slightly better accuracy as compared to the group lasso.

## 7.4 Group-lasso penalized neural network

A feed-forward neural network with 4 layers was implemented using Keras with Tensor-Flow backend and the L21 (group lasso) penalty as specified in [17]. While the penalty doesn't allow for a manual specification of groups, it is interesting to see how the network gets sparser, while maintaining the accuracy. The neurons whose weights have been set to 0 by the group lasso penalty are defined as 'inactive' neurons. The following plot shows the active neurons vs training epochs.

A 4-fold cross validation resulted in a mean squared loss of 0.02541.

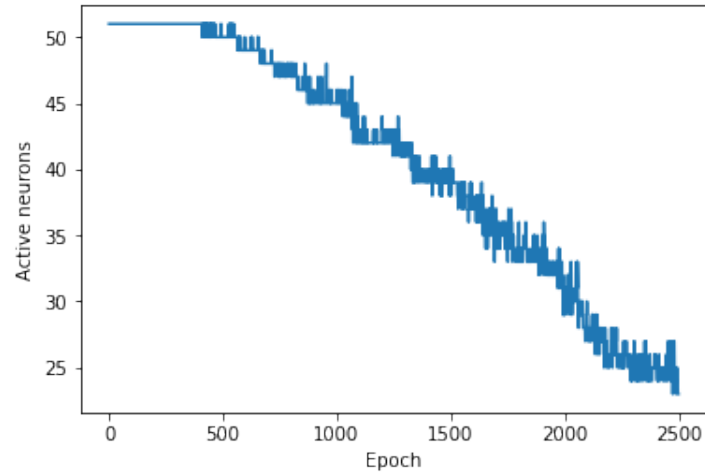


Figure 11: Number of active neurons vs epochs

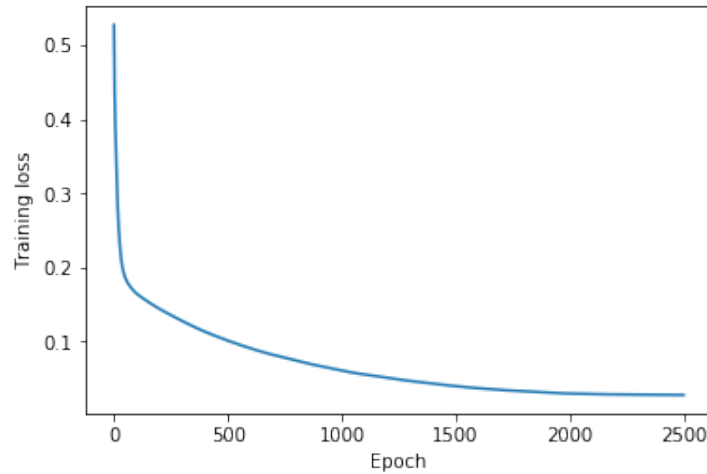


Figure 12: Mean-square error vs epochs

## 8 Conclusion

The following table summarizes the results from the 4 approaches on the raw data and on mean-normalized data:

	MSE on Raw data	MSE (normalized)	Runtime (s)	Model size
Group-lasso	0.441	0.032387	3.86	11
ALAMO	0.402	0.06359	11.3	11
Group-sparse Neural Network	0.386	0.02541	115.018	-

The runtimes reported above have been calculated on a dataset with 100,000 data points and 16 features. The number of epochs for the neural network were brought down to 10 for achieving comparable runtimes. It is worth noting that upon performing mean-normalization of the data, the mean-squared error goes down by an order of magnitude.

The elastic net serves as an intermediate penalty between the lasso and the ridge and accounts for variable shrinkage and selection. However, it does not allow for incorporation of prior knowledge about the group-wise nature of predictors.

The motivation for incorporating a group lasso penalty in neural networks is to reduce the dimensionality of the input space, thus reducing overall model size by setting to 0 the corresponding coefficients in the weight matrices of the consequent layers. This results in an overall sparser model, which has the advantages of fast calculations. The motivation is akin to that of the dropout [14], one of the most highly cited papers in deep learning. This has tremendous applications in cases where pre-trained networks are required for fast classification (e.g. self-driving cars).

All models discussed above have their unique characteristics. The group lasso incorporates structured sparsity and allows for variable and group specification. However, the variable selection is done entirely by the tool as it learns the weights from data. ALAMO, on the other hand, allows for manual specification on the group-wise structure of predictors and also accounts for constraints on the presence and absence of certain groups of variables in the model. It also allows for non-linear transformations of the predictors, allowing for learning more complex functions from data. ALAMO also has a better cross-validation accuracy as compared to the group lasso. However, model building in ALAMO is slightly slower than the group-lasso due to the non-convex nature of the  $L_0$  penalty that ALAMO tries to solve and the iterative nature of the branch and bound algorithm that it utilizes. In conclusion, ALAMO provides a higher control over the model and allows for building a simpler and in consequence, a more interpretable model within comparable runtimes.

## References

- [1] Arthur E Hoerl and Robert W Kennard. “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1 (1970), pp. 55–67.
- [2] Robert Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.
- [3] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320.
- [4] Hyun Hak Kim and Norman R Swanson. “Forecasting financial and macroeconomic variables using data reduction methods: New empirical evidence”. In: *Journal of Econometrics* 178 (2014), pp. 352–367.
- [5] Ming Yuan and Yi Lin. “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), pp. 49–67.
- [6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. “Unsupervised learning”. In: *The elements of statistical learning*. Springer, 2009, pp. 485–585.
- [7] Stephen Boyd et al. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.
- [8] Alison Cozad, Nikolaos V Sahinidis, and David C Miller. “Learning surrogate models for simulation-based optimization”. In: *AIChE Journal* 60.6 (2014), pp. 2211–2227.
- [9] Alison Cozad, Nikolaos V Sahinidis, and David C Miller. “A combined first-principles and data-driven approach to model building”. In: *Computers & Chemical Engineering* 73 (2015), pp. 116–127.
- [10] Zachary T Wilson and Nikolaos V Sahinidis. “The ALAMO approach to machine learning”. In: *Computers & Chemical Engineering* 106 (2017), pp. 785–795.
- [11] Dana Angluin. “Queries and concept learning”. In: *Machine learning* 2.4 (1988), pp. 319–342.
- [12] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009.
- [13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), p. 436.
- [14] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.
- [15] Misha Denil et al. “Predicting parameters in deep learning”. In: *Advances in neural information processing systems*. 2013, pp. 2148–2156.

- [16] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. Vol. 398. John Wiley & Sons, 2013.
- [17] Simone Scardapane et al. “Group sparse regularization for deep neural networks”. In: *Neurocomputing* 241 (2017), pp. 81–89.