High-Dim Project

name1
Columbia University
uni1@columbia.edu

name2
Columbia University
uni2@columbia.edu

name3
Columbia University
uni3@columbia.edu

1 Introduction

2 Background

2.1 Latent Group Lasso

2.2 Multiclass Classification with Group Lasso

3 Our Model

Given n training vectors $x_i \in R_d$ and their class labels $y_i \in \{1, ..., m\}$, our goal is to compute W such that it maximizes the accurarcy of our prediction and it is group-wise sparse.

In our model, we minimize the following objective function :

$$minimize_{W \in R^{dxm}} F(W) =$$
$$\frac{1}{n} \sum_{i=1}^{n} \sum_{r \neq y_i} max(1 - W_{:y_i} \cdot x_i + W_{:r} \cdot x_i, 0)^2$$
$$+\lambda \sum_{g=1}^{|G|} \sum_{m=1}^{d} \|W_{g,m}\|_2$$

The first term is the multiclass squared hinge loss function. We want the dot product of an instance and its feature vector to be as large as possible, and the dot product of this instance and the rest feature vectors to be as small as possible. And as long as their difference is greater then a margin (1 in this case), we won't penalize it. In the second term, $W_{g,m}$ means a block of weights in group g and class m. The L2 norm regulization is computed and sum up for each block. The $\lambda > 0$ is a parameter controls the trade-off between the hinge loss and the L2-norm regulization.

4 Data

4.1 Newsgroup Data

4.1.1 Group Identification

4.2 Artificial data

For the datasets described above, we can't tell with 100 percent confidence that the datasets follow the assumptions of the group structures for the features. And even if they are indeed structured that way, we maybe wrong with the method of coming up with the groups. These issues make it difficult to access our model.

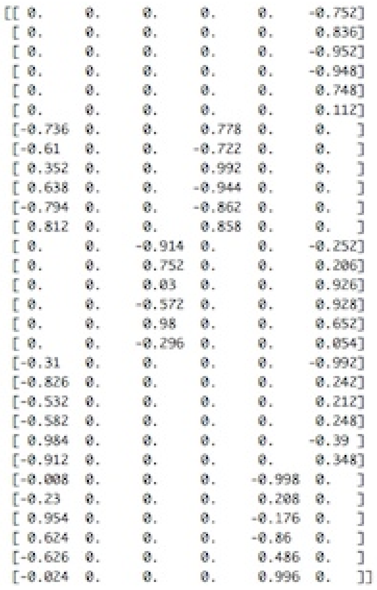
To get rid of all these problems and validate the effectiveness of our model, we created artificial data that followed the underlying assumptions of the model. First, we generate a sparse weight matrix W to represent the relationship between features and classes. The weight matrix W has an internal structure in which features are grouped together. And also, only a small number of groups have non-zero weights. This makes the matrix sparse.

Then we generate random vectors, each of which has the length of the number of all features, and calculate dot product with the weight matrix W to get the class assignments for these random vectors. The random vetors X and the class assignments Y make up the training data set.

Our goal is to infer this weight matrix W from X and Y using our model. By generating the data set using this method, we can test the effectiveness of our model on a noiseless dataset with right underlying assumptions.

5 Results

6 Conclusion



```
[[ 0.     0.     0.     0.     0.    -0.752]
 [ 0.     0.     0.     0.     0.     0.836]
 [ 0.     0.     0.     0.     0.    -0.952]
 [ 0.     0.     0.     0.     0.    -0.948]
 [ 0.     0.     0.     0.     0.     0.748]
 [ 0.     0.     0.     0.     0.     0.112]
 [-0.736  0.     0.     0.778  0.     0.   ]
 [-0.61   0.     0.    -0.722  0.     0.   ]
 [ 0.352  0.     0.     0.992  0.     0.   ]
 [ 0.638  0.     0.    -0.944  0.     0.   ]
 [-0.794  0.     0.    -0.862  0.     0.   ]
 [ 0.812  0.     0.     0.858  0.     0.   ]
 [ 0.     0.    -0.914  0.     0.    -0.252]
 [ 0.     0.     0.752  0.     0.     0.206]
 [ 0.     0.     0.03   0.     0.     0.926]
 [ 0.     0.    -0.572  0.     0.     0.928]
 [ 0.     0.     0.98   0.     0.     0.652]
 [ 0.     0.    -0.296  0.     0.     0.054]
 [-0.31   0.     0.     0.     0.    -0.992]
 [-0.826  0.     0.     0.     0.     0.242]
 [-0.532  0.     0.     0.     0.     0.212]
 [-0.582  0.     0.     0.     0.     0.248]
 [ 0.984  0.     0.     0.     0.    -0.39 ]
 [-0.912  0.     0.     0.     0.     0.348]
 [-0.008  0.     0.     0.    -0.998  0.   ]
 [-0.23   0.     0.     0.     0.208  0.   ]
 [ 0.954  0.     0.     0.    -0.176  0.   ]
 [ 0.624  0.     0.     0.    -0.86   0.   ]
 [-0.626  0.     0.     0.     0.486  0.   ]
 [-0.024  0.     0.     0.     0.996  0.   ]]
```

Figure 1: Group-wise sparse weight matrix generated: 6 classes, 30 features in 5 groups