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 Multiclass Classification with Group Lasso

The task of multiclass classification involves the prediction of a class label l where the number of possible labels is $k > 2$. More often than not, the original problem is transformed into k binary classification problems, i.e. 1–vs.–all classification and positive prediction with the highest confidence is selected as the label. This approach has the disadvantage of having to train k different models.

An alternative formulation, direct multiclass classification, tackles this problem directly by solving the following argmax problem:

$$y_i = \mathrm{argmax}_c \, W_{:c}^T x_i$$

where $W \in \mathcal{R}^{p \times k}$ is a weight matrix, with W_{ij} corresponding to the i-th feature of class j. In this paper, we refer to features as elements in the instance data x. A feature in x is associated with k weights in W, one for each class.

The decision function above suggests a max-margin style loss function. More specifically, we use the squared hinge loss:

$$l(W) = \sum_{i=1}^{n} \sum_{r \neq y_i}^{k} \max\left(1 - (W_{:y_i}^T x_i - W_{:r}^T x_i), 0\right)^2$$

The minimization of l directly will lead to a minimizer W^* that is dense. Sparse solutions are often explicitly sought, with model compactness leading to fast prediction at test time. In order to obtain a sparse W^*, a regularization term $r(W)$ is often applied, yielding the objective function:

$$\min_W l(W) + r(W).$$

Many choices are available for the regularizer r. In (ref ???), they use the group lasso, where each row in W is a group. The associated regularizer then is $r(W) = \lambda \sum_j^p \|W_{j:}\|_2$ where λ is a parameter that adjusts the strength of the regularization. This has the effect of producing a few rows of non-zero values in W; since each row corresponds to an individual feature, the optimal sparse W^* yields a fast-evaluating decision function, i.e. most features are ignored at test time.

To minimize this multiclass classification group lasso objective, ??? use coordiante descent, iteratively solving a sub-problem with respect to a single group. Figure ? shows a general outline of algorithm that involves computing the partial gradient with respect to the current group j, the prox operator of the L2 norm, and a final line search to identify an appropriate step size for the current update.

for $i \leftarrow 1, \ldots, max\ iters$ **do**
  for $j \leftarrow 1, \ldots, p$ **do**
    Compute gradient $l'(W)_{j:}$
    Choose $\mathcal{L}_j$
    Compute
      $V_j = W_{j:} - \frac{1}{\mathcal{L}_j} l'(W)_{j:}$
      $W_{j:}^* = \mathrm{Prox}_{\frac{\lambda}{\mathcal{L}_j}\|\cdot\|_j}(V_j)$
      $\delta = W_{j:}^* - W_{j:}$
    Choose α
    $W_{j:} \leftarrow W_{j:} + \alpha\delta$
  end
end

Efficient computation of this objective is possible by storing current loss for each data point. Let A be an $n \times k$ matrix where the i, r-th element corresponds to $(1 - (W_{:y_i}^T x_i - W_{:r} x_i)$. The gradient can then be calculated as $l'(W)_j = \frac{2}{n} \sum_{i=1}^{n} \sum_{r \neq y_i} \max(A_{ir}, 0)(x_{ij} e_{y_i} - x_{ij} e_r)$ where e_r is a k dimensional vector with

zeroes everywhere except for a 1 at the r-th position. We only have to examine elements in A for which the corresponding x_{ij} is non-zero. When x_i is sparse, more often than not x_{ij} is zero and can be ignored.

2.2 Latent Group Lasso

One limitation of group lasso is that it assumes that group assignments are non-overlapping. In some domains, this can be too restrictive an assumption. For example, in document classification, individual words are used as features. If we were to construct groupings of these features, we might run into a case where one word could reasonably be added to several groups. The overlapping or latent group lasso was introduced to handled such cases.

??? develop a theoretical justification for the latent group lasso, as well as its equivalence to a regular group lasso in a higher dimensional space. Let $\mathcal{G}$ be the set of (possibly overlapping) groups, where $g \in \mathcal{G}$ is a set of indices of covariates associated with that group. Let our data consist of vectors x_i in p dimensions, and let w be the corresponding weight vector in p dimensions that we would like to learn. Finally, define $\mathrm{supp}(v)$ to be the support of v, i.e. the indices of the non-zero elements in v.

For each group $g \in \mathcal{G}$ we associate a latent vector $v^g \in \mathcal{R}^p$ where $\mathrm{supp}(v^g) = g$, i.e. the nonzero elements in the v^g correspond to the indices in the group g. The original weight vector w can be interpreted as a sum of the latent vectors, or $w = \sum_{g\in\mathcal{G}} v^g$. ??? arrive at the following minimization problem

$$\min_{w,v^g} l(w) + \lambda \sum_{g\in\mathcal{G}} d_g \|v^g\|_2$$

$$\text{s.t.} \quad w = \sum_g v^g$$

??? show that when the original problem is regression, $w^T x = \left(\sum_g v^g\right)^T x = \hat{v}^T \hat{x}$ where $\hat{v} = (v^g)_{g\in\mathcal{G}}$ and $\hat{x} = \bigoplus_{g\in\mathcal{G}} (x_i)_{i\in g}$, i.e. $\hat{x}$ is the restrictions of each g stacked on top of each other. $\hat{x}, \hat{v}$ have dimension $\sum_{g\in\mathcal{G}} |g|$ In this formulation, the optimal $\hat{v}^*$ can be found using regular non-overlapping 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 :

$$\min_{W\in R^{dxm}} F(W) =$$

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{r\neq y_i} \max(1 - (W_{:y_i}^T \cdot x_i - W_{:r}^T \cdot x_i), 0)^2$$

$$+\lambda \sum_{g\in\mathcal{G}}\sum_{r=1}^{d} \|W_{g,r}\|_2$$

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, $W_{g,r}$ 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.

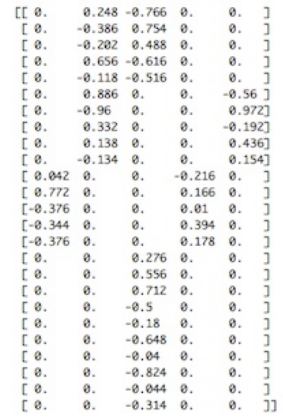

Figure 1: Group-wise sparse weight matrix generated: 5 classes, 25 features in 5 groups

Then we generate random vectors, each of which has a 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

5.1 Newsgroup Data

5.2 Artificial Data

Shape Recovery. One of the main indicator of the effectiveness of our model is to see whether the calculated weight matrix is sparse group-wise. Our experiments show so. The following figures shows in a typical trial, the generated target weight matrix and the recovered weight matrix by our model. By comparing them side by side, we can tell that their sparsity patterns are similar.

The different weights between the calculated matrix and the target matrix can be due to many factors. First, sample coverage is a major factor. In our simulation data, the sample size is small. Limited by the time complexity of the algorithm, it's difficult to complete the computation for a very large sample size in a reasonable time. Also,

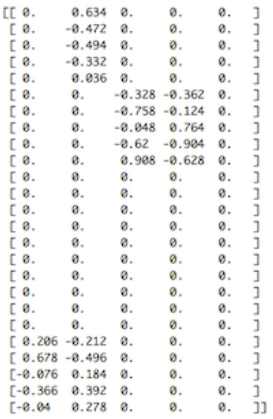

Figure 2: Target group-wise sparse weight matrix generated: 5 classes, 25 features in 5 groups

the sampling is random. There is no guaranteed that the inferred weights leads to the target weights.

Accuracy. In our experiments, the generator algorithm was configured to produce 150 random vectors from the underlying model where it consists of 5 classes and 25 features in 5 groups. The accuracy achieved was about 60% to 70%.

6 Conclusion

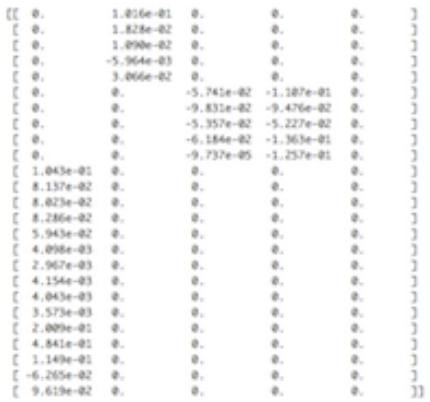

Figure 3: Weight matrix calculated from the training data. It's similar to the one generated.