High-Dim Project

Chris Kedzie
Columbia University
kedzie@cs.columbia.edu

Dahong Liu
Columbia University
dl2868@columbia.edu

Xiao Zhu
Columbia University
xz2362@columbia.edu

Yichi Zhang
Columbia University
yz2657@columbia.edu

1 Introduction

We motivate our project by considering the text classification problem: can we accurately classify text documents by topic. With the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data. Text categorization techniques are used not only to filter spam emails, as we are mostly familiar, but also to classify news stories, to find interesting information on the WWW, and to guide a user's search through hypertext. Since building text classifiers by hand is difficult and time-consuming, it is advantageous to learn classifiers from examples.

In this project we will use the "20 Newsgroups" dataset, popular in machine learning literature, to explore some variants of latent group lasso for text classification. This dataset contains about 20,000 documents split roughly evenly amongst the 20 predefined topics. Every training example comes in the form of a document-term matrix $\mathbf{M}$ that is $D \times V$ where $D = 11307$. The vocabulary size $V = 61188$ words.

In section 2, we will provide some background on latent group lasso and multiclass classification. Latent group Lasso is based on applying the usual group Lasso penalty on a set of latent variables when groups are overlapping. In section 3, we will present our model, which uses hinge loss for training classifiers. The following sections will include detailed explanations of our datasets and the results we gathered.

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:

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

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 ← 1, ..., max iters do
    for j ← 1, ..., p do
        Compute gradient l'(W)_{j:}
        Choose L_j
        Compute
            V_j = W_{j:} − (1/L_j) l'(W)_{j:}
            W*_{j:} = Prox_{(λ/L_j)||·||_j}(V_j)
            δ = W*_{j:} − W_{j:}
        Choose α
        W_{j:} ← W_{j:} + αδ
    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 :

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

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.

```
[[ 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

5 Results

6 Conclusion