Orthogonal Matching Pursuit for Text Classification

Konstantinos Skianis[1], Nikolaos Tziortziotis[1], Michalis Vazirgiannis[1,2]
[1]LIX, École Polytechnique, France
[2]Athens University of Economics and Business, Greece
{kskianis, mvazirg}@lix.polytechnique.fr, ntziorzi@gmail.com

Abstract

In text classification, the problem of overfitting arises due to the high dimensionality, making regularization essential. Although classic regularizers provide sparsity, they fail to return highly accurate models. On the contrary, state-of-the-art group-lasso regularizers provide better results at the expense of low sparsity. In this paper, we apply a greedy variable selection algorithm, called Orthogonal Matching Pursuit, for the text classification task. We also extend standard group OMP by introducing overlapping Group OMP to handle overlapping groups of features. Empirical analysis verifies that both OMP and overlapping GOMP constitute powerful regularizers, able to produce effective and very sparse models. Code and data are available online[1].

1 Introduction

The overall high dimensionality of textual data is of major importance in text classification (also known as text categorization), opinion mining, noisy text normalization and other NLP tasks. Since in most cases a high number of words occurs, one can easily fall in the case of overfitting. Regularization remains a key element for addressing overfitting in tasks like text classification, domain adaptation and neural machine translation (Chen and Rosenfeld, 2000; Lu et al., 2016; Barone et al., 2017). Along with better generalization capabilities, a proper scheme of regularization can also introduce sparsity. Recently, a number of text regularization techniques have been proposed in the context of deep learning (Qian et al., 2016; Ma et al., 2017; Zhang et al., 2017).

Apart from ℓ_1, ℓ_2 and elastic net, a very popular method for regularizing text classification is group lasso. Yogatama and Smith (2014b) introduced a group lasso variant to utilize groups of

[1]github.com/y3nk0/OMP-for-Text-Classification

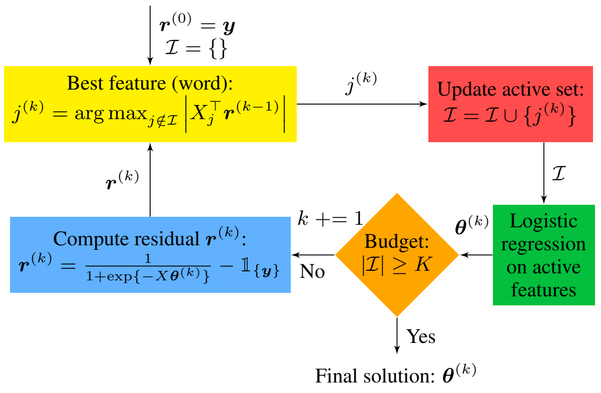
$r^{(0)} = y$
$\mathcal{I} = \{\}$
Best feature (word):
$j^{(k)} = \arg\max_{j \notin \mathcal{I}} \left| X_j^\top r^{(k-1)} \right|$
$j^{(k)}$
Update active set:
$\mathcal{I} = \mathcal{I} \cup \{j^{(k)}\}$
$\mathcal{I}$
Logistic regression on active features
$\theta^{(k)}$
Budget: $|\mathcal{I}| \geq K$
$k += 1$
No
Yes
Compute residual $r^{(k)}$:
$r^{(k)} = \frac{1}{1+\exp\{-X\theta^{(k)}\}} - \mathbb{1}_{\{y\}}$
$r^{(k)}$
Final solution: $\theta^{(k)}$

Figure 1: OMP pipeline where $X \in \mathbb{R}^{N \times d}$ is the design matrix, $y \in \mathbb{R}^N$ is the response vector, K is our budget and $\mathcal{I}$ the set of active features.

words for logistic models. Occasionally though, these groupings are either not available or hard to be extracted. Moreover, no ground truth groups of words exist to validate their quality. Furthermore, group lasso can also fail to create sparse models. Lastly, there has been little work in overlapping group regularization for text, since words can appear in different groups, following the intuition that they can share multiple contexts or topics.

In this work, we apply two greedy variable selection techniques to the text classification task, Orthogonal Matching Pursuit (OMP) and overlapping group Orthogonal Matching Pursuit (GOMP). In the case of GOMP, we build upon work of Lozano et al. (2011), where the authors propose the GOMP algorithm for Logistic Regression for selecting relevant groups of features. More specifically, standard GOMP is based on the assumption that a number of disjoint groups of features are available. Nevertheless, in most cases, these groups are not disjoint. To overcome this problem we extend GOMP to handle overlapping groups of features. We empirically show that both OMP and overlapping GOMP provide highly accurate models, while producing very sparse models compared to group lasso variants. Figure 1 illustrates schematically the pipeline of OMP.

Our contribution can be summarized in the following novel aspects: (1) apply OMP to text classification; (2) introduce overlapping GOMP, moving from disjoint to overlapping groups; (3) analyze their efficiency in accuracy and sparsity, compared to group lasso variants and state-of-the-art deep learning models.

The rest of the paper is organized as follows. Section 2 presents the background about the classification task, and Section 3 gives an overview of the related work. Section 4 formally introduces the proposed OMP and overlapping GOMP algorithms for the text classification problem. Experimental results are presented in Section 5. We conclude the paper in Section 6 by discussing possible future directions.

2 Background & Notation

In this section, we set the theoretical and practical background, needed to tackle text classification.

2.1 Loss minimization

In the binary classification problem, the objective is to assign an instance vector $\boldsymbol{x} \in \mathbb{R}^d$, which represents a document in our setting, to a binary response $y \in \{-1, 1\}$. In text classification, d represents the size of our dictionary concatenated with an additional bias term.

We begin by transforming the classification problem into a loss minization problem. For that purpose, a loss function should be defined that quantifies the loss between the prediction of a classifier and the true class label, y_i, associated with a specific document (instance), $\boldsymbol{x}_i$.

Logistic regression models the class conditional probability, as:

$$P(Y = y|\boldsymbol{x}) = \frac{1}{1 + \exp\{-y(\boldsymbol{\theta}^\top \boldsymbol{x})\}}, \quad (1)$$

where vector $\boldsymbol{\theta}$ contains the unknown model's parameters, and the hyperplane $\boldsymbol{\theta}^\top \boldsymbol{x} = 0$ is the decision boundary of the classifier that separates the two classes. Given a training set of i.i.d. data point $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, we find the optimal model's parameters, $\boldsymbol{\theta}^*$, by minimizing the negative log likelihood:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\theta}, y_i), \quad (2)$$

where $\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\theta}, y_i) = \log[1 + \exp\{-y_i(\boldsymbol{\theta}^\top \boldsymbol{x}_i)\}]$ is the loss function of our model. It should be also noticed that other loss functions can be used such as hinge loss, square loss, etc.. For linear classifiers such as Linear Least Square Fit, Logistic Regression and linear Support Vector Machines, in the case of binary predictions the $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, y)$ is respectively $[1 - y(\boldsymbol{\theta}^\top \boldsymbol{x})]^2$ (squared loss), $\log[1 + \exp\{-y(\boldsymbol{\theta}^\top \boldsymbol{x})\}$ (log loss) and $[1 - y(\boldsymbol{\theta}^\top \boldsymbol{x})]_+$ (hinge loss).

2.2 Regularization

By only minimizing the empirical risk, a model can be led to severe overfitting in the case where the number of the features (dictionary) is much higher than the number of the instances (documents) in training set. In practice, it yields models with poor generalization capabilities (i.e., lower performances on the test set) that fit the noise contained in the training dataset instead of learning the underlying pattern we are trying to capture. Additionally, if two hypothesis lead to similar low empirical risks, one should select the "simpler" model for better generalization capabilities.

Interpretation The concept of regularization encompasses all these ideas. It can be interpreted as taking into account the model complexity by discouraging feature weights from reaching large values; incorporating prior knowledge to help the learning by making prior assumptions on the feature weights and their distribution; and helping compensate ill-posed conditions.

Expected risk Regularization takes the form of additional constraints to the minimization problem, i. e., a budget on the feature weights, which are often relaxed into a penalty term $\Omega(\boldsymbol{\theta})$ controlled via a Lagrange multiplier λ (see Boyd and Vandenberghe (2004) for more details about the theory behind convex optimization). Therefore, the overall expected risk (Vapnik, 1991) can be expressed as the weighted sum of two components: the empirical risk and a penalty term, called "Loss+Penalty" (Hastie et al., 2009). In this way, the optimal set of feature weights $\boldsymbol{\theta}^*$ is found as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\theta}, y_i) + \lambda \Omega(\boldsymbol{\theta}) \quad (3)$$

where the free parameter $\lambda \geq 0$ governs the importance of the penalty term compared with the loss term.

3 Related Work

In this section, we review particularly relevant prior work on regularization for text classification and more specifically methods based on group lasso.

In many applications of statistics and machine learning, the number of exploratory variables may be very large, while only a small subset may truly be relevant in explaining the response to be modelled. In certain cases, the dimensionality of the predictor space may also exceed the number of examples. Then the only way to avoid overfitting is via some form of capacity control over the family of dependencies being explored. Estimation of sparse models that are supported on a small set of input variables is thus highly desirable, with the additional benefit of leading to parsimonious models, which can be used not only for predictive purposes but also to understand the effects (or lack thereof) of the candidate predictors on the response.

More specifically, regularization in text scenarios is essential as it can lead to removing unnecessary words along with their weights. For example, in text classification, we may only care for a small subset of the vocabulary that is important during the learning process, by penalizing independently or in grouped way noisy and irrelevant words.

With noiseness we refer to user-generated words that may increase the dimensionality and complexity of a problem, while having a clear decreasing effect in performance.

Another example task is text normalization, where we want to transform lexical variants of words to their canonical forms. Text normalization can be seen as a machine learning problem (Ikeda et al., 2016) and thus regularization techniques can be applied.

Next we present standard regularization methods, which prove to be effective for classification tasks. We also use them later as baselines for our experiments.

ℓ_1, ℓ_2 regularization Two of the most used regularization schemes are ℓ_1-regularization, called *Lasso* (Tibshirani, 1996) or *basis pursuit* in signal processing (Chen et al., 2001), and ℓ_2-regularization, called *ridge* (Hoerl and Kennard, 1970) or *Tikhonov* (Tikhonov and Arsenin, 1977), which involve adding a penalty term (ℓ_1 and ℓ_2 norms of the parameter vector, respectively) to the error function:

$$\Omega_{lasso}(\boldsymbol{\theta}) = \sum_{i=1}^{d} |\theta_i| = \|\boldsymbol{\theta}\|_1, \tag{4}$$

$$\Omega_{rigde}(\boldsymbol{\theta}) = \sum_{i=1}^{d} \theta_i^2 = \|\boldsymbol{\theta}\|_2^2. \tag{5}$$

Elastic net A linear combination of the ℓ_1 and ℓ_2 penalties has been also introduced by Zou and Hastie (2005), called *elastic net*. Although ℓ_1 and elastic net can be very effective in terms of sparsity, the accuracy achieved by these regularizers can be low. On the contrary, ℓ_2 can deliver sufficient accuracy at the cost of zero sparsity. The need for new methods that outperform the aforementioned approaches in both accuracy and sparsity is evident.

Group structured regularization In many problems a predefined grouping structure exists within the explanatory variables, and it is natural to incorporate the prior knowledge so that the support of the model should be a union over some subset of these variable groups. Group structured regularization has been proposed to address the problem of overfitting, given we are provided with groups of features. Group lasso is a special case of group regularization proposed by Yuan and Lin (2006), to avoid large ℓ_2 norms for groups of weights, given we are provided with groups of features. The main idea is to penalize together features that may share some properties.

Group structured regularization or variable group selection problem is a well-studied problem, based on minimizing a loss function penalized by a regularization term designed to encourage sparsity at the variable group level. Specifically, a number of variants of the ℓ_1-regularized lasso algorithm (Tibshirani, 1996) have been proposed for the variable group selection problem, and their properties have been extensively studied recently. First, for linear regression, Yuan and Lin (2006) proposed the group lasso algorithm as an extension of lasso, which minimizes the squared error penalized by the sum of ℓ_2-norms of the group variable coefficients across groups. Here the use of ℓ_2-norm within the groups and ℓ_1-norm across the groups encourages sparsity at the group level.

In addition, group lasso has been extended to logistic regression for binary classification, by replacing the squared error with the logistic error (Kim et al., 2006; Meier et al., 2008), and several

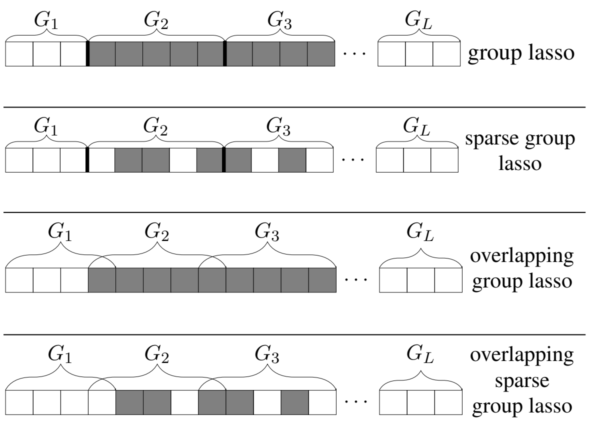

Figure 2: A graphical representation of different group lasso architectures. Grey boxes depict active features.

extensions thereof have been proposed (Roth and Fischer, 2008).

Later, sparse group lasso and overlapping group lasso were introduced (Obozinski et al., 2011) to additionally penalize features inside the groups, while the latter can be used when groups include features that can be shared between them.

In Figure 2, we illustrate the selection of features by the most used group lasso regularizers. In group lasso, a group of features is selected and all its features are used. Next, in the sparse group lasso case, groups of features are selected again but not all the features belonging to them are used. In the overlapping group lasso, groups can share features between them. Finally, we may have sparse group lasso with overlaps.

Linguistic structured regularizers As mentioned previously, words that appear together in the same context, share topics or even have a similar meaning, may form groups that capture semantic or syntactic prior information. Hence we can feed these groups to group lasso. Yogatama and Smith (2014a) used the Alternating Direction Method of Multipliers algorithm (ADMM) (Boyd et al., 2011) for group lasso, an algorithm that solves convex optimization problems by breaking them into smaller pieces. In this work, groups extracted by Latent Dirichlet Allocation (LDA) and sentences were used for structured regularization. Next, Skianis et al. (2016) extended their work by utilizing topics extracted by Latent Semantic Indexing (LSI) (Deerwester et al., 1990), communities in Graph-of-Word (GoW) (Rousseau and Vazirgiannis, 2013) and clusters in the word2vec (w2v) (Mikolov et al., 2013) space. They also performed a computational analysis in terms of the number and size of groups and how it can affect learning times.

While current state-of-the-art methods either focus on finding the most meaningful groups of features or how to further "optimize" the group lasso approach, the attempts carry as well the disadvantages of group lasso architectures. In some cases, we may not be able to extract "good" groups of words. As presented in the next section, we want to explore new ways of regularization on groups, diverging from group lasso, that can give high accuracy with high sparsity.

4 OMP for Text Classification

The vanilla Matching Pursuit (MP) algorithm (Mallat and Zhang, 1993) has its origin in signal processing where it is mainly used in the compressed sensing task. Actually, it approximates the original "signal" iteratively improving the current solution by minimizing the norm of the residual (approximation error). It can also be considered as a forward greedy algorithm for feature selection (dictionary learning problem), that at each iteration uses the correlation between the residual and the candidate features to (greedily) decide which feature to add next. The correlation between the residual and the candidate features is considered to be the length of the orthogonal projection. Then, it subtracts off the correlated part from the residual and performs the same procedure on the updated residual. The algorithm terminates when the residual is lower than a predefined threshold. The final solution is obtained by combining the selected features weighted by their respective correlation values, which are calculated at each iteration.

Orthogonal Matching Pursuit (Pati et al., 1993) is one of the most famous extensions of the matching pursuit algorithm. Similar to MP, OMP can be used for the dictionary learning task where it constitutes a competitive alternative to lasso algorithm. The way it differs from the standard MP is that at every step, all the coefficients extracted so far are updated, by computing the orthogonal projection of the data onto the set of features selected so far. In this way, the newly derived residual is orthogonal to not only the immediately selected feature at the current iteration, but also to all the features that have already been selected. Therefore, OMP never selects the same feature twice. Tropp (2004) provided a theoretical analysis of OMP,

Algorithm 1 Logistic-OMP

```
Input: $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times d}$, $\boldsymbol{y} \in \{-1, 1\}^N$, $K$ (budget), $\epsilon$ (precision), $\lambda$ (regularization factor).
Initialize: $\mathcal{I} = \emptyset$, $\boldsymbol{r}^{(0)} = \boldsymbol{y}$, $k = 1$
1: while $|\mathcal{I}| \leq K$ do
2:     $j^{(k)} = \arg\max_{j \notin \mathcal{I}} \left| X_j^\top \boldsymbol{r}^{(k-1)} \right|$
3:     if $|X_{j^{(k)}}^\top \boldsymbol{r}^{(k-1)}| \leq \epsilon$ then
4:         break
5:     end if
6:     $\mathcal{I} = \mathcal{I} \cup \{j^{(k)}\}$
7:     $\boldsymbol{\theta}^{(k)} = \text{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^N \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\theta}, y_i) + \lambda \|\boldsymbol{\theta}\|_2^2$ $s.t.\ \ supp(\boldsymbol{\theta}) \subseteq \mathcal{I}$
8:     $\boldsymbol{r}^{(k)} = \frac{1}{1+\exp\{-X\boldsymbol{\theta}^{(k)}\}} - \mathbb{1}_{\{\boldsymbol{y}\}}$
9:     $k \mathrel{+}= 1$
10: end while
11: return $\boldsymbol{\theta}^{(k)}, \mathcal{I}$
```

which has been generalized by Zhang (2009) on the stochastic noise case.

In the following part, we explain the main steps of the logistic OMP algorithm in detail. Given a training set, we define $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times d}$ to be the (dictionary) matrix of features (or variables) vectors, with each column X_j to represent a feature, $\boldsymbol{f}_j \in \mathbb{R}^N$. Let also $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$ denote the response vector. For any set of indices $\mathcal{I}$, let $X_\mathcal{I}$ denote a subset of features from X, such that feature $\boldsymbol{f}_j$ is included in $X_\mathcal{I}$ if $j \in \mathcal{I}$. Thus, $X_\mathcal{I} = \{\boldsymbol{f}_j, j \in \mathcal{I}\}$, with the columns $\boldsymbol{f}_j$ to be arranged in ascending order.

OMP starts by setting the residual equal to the response vector, $\boldsymbol{r}^{(0)} = \boldsymbol{y}$, assuming that the set of indices $\mathcal{I}$ (contains the indices of the active features) is initially empty. At each iteration k, OMP activates the feature that has the maximum correlation with the residual $\boldsymbol{r}^{(k-1)}$ (calculated in the previous step):

$$j^{(k)} = \arg\max_{j \notin \mathcal{I}} \left| X_j^\top \boldsymbol{r}^{(k-1)} \right|. \tag{6}$$

Then, we incorporate the index $j^{(k)}$ to the set $\mathcal{I}$, i. e., $\mathcal{I} = \mathcal{I} \cup \{j^{(k)}\}$. Afterwards, we apply the ordinary logistic regression by considering only the *active* features. More specifically, we get the optimal coefficients by minimizing the negative log likelihood along with an ℓ_2 penalty term:

$$\boldsymbol{\theta}^{(k)} = \underset{\boldsymbol{\theta}}{\text{argmin}} \sum_{i=1}^N \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\theta}, y_i) + \lambda \|\boldsymbol{\theta}\|_2^2, \quad s.t. \quad supp(\boldsymbol{\theta}) \subseteq \mathcal{I} \tag{7}$$

where $supp(\boldsymbol{\theta}) = \{j : \theta_j \neq 0\}$. Roughly speaking, the values of the coefficients correspond to inactive features (indices) forced to be equal to zero.

Algorithm 2 Logistic Overlapping GOMP

```
Input: $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times d}$, $\boldsymbol{y} \in \{-1, 1\}^N$, $\{G_1, \ldots, G_J\}$ (group structure), $K$ (budget), $\epsilon$ (precision), $\lambda$ (regularization factor).
Initialize: $\mathcal{I} = \emptyset$, $\boldsymbol{r}^{(0)} = \boldsymbol{y}$, $k = 1$
1: while $|\mathcal{I}| \leq K$ do
2:     $j^{(k)} = \arg\max_j \frac{1}{|G_j|} \left\| X_{G_j}^\top \boldsymbol{r}^{(k-1)} \right\|_2^2$
3:     if $\left\| X_{G_{j^{(k)}}}^\top \boldsymbol{r}^{(k-1)} \right\|_2^2 \leq \epsilon$ then
4:         break
5:     end if
6:     $\mathcal{I} = \mathcal{I} \cup \{G_{j^{(k)}}\}$
7:     for $i = 1$ to $J$ do
8:         $G_i = G_i \setminus G_{j^{(k)}}$
9:     end for
10:    $\boldsymbol{\theta}^{(k)} = \text{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^N \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\theta}, y_i) + \lambda \|\boldsymbol{\theta}\|_2^2$ $s.t.\ \ supp(\boldsymbol{\theta}) \subseteq \mathcal{I}$
11:    $\boldsymbol{r}^{(k)} = \frac{1}{1+\exp\{-X\boldsymbol{\theta}^{(k)}\}} - \mathbb{1}_{\{\boldsymbol{y}\}}$
12:    $k \mathrel{+}= 1$
13: end while
14: return $\boldsymbol{\theta}^{(k)}, \mathcal{I}$
```

Finally, we calculate the updated residual:

$$\boldsymbol{r}^{(k)} = \frac{1}{1 + \exp\{-X\boldsymbol{\theta}^{(k)}\}} - \mathbb{1}_{\{\boldsymbol{y}\}}, \tag{8}$$

where $\mathbb{1}_{\{\boldsymbol{y}\}} \triangleq \mathbb{1}\{y_i \in \{1\}, \forall i \in \{1, \ldots, n\}\}$ indicates if instance $\boldsymbol{x}_i$ belongs to class 1 or not. We repeat the process until the residual becomes smaller than a predefined threshold, $\epsilon \geq 0$, or a desired number of active features, K (budget), has been selected. Through our empirical analysis we set $\epsilon = 0$, examining only the number of active features. An overview of logistic-OMP is given in Alg. 1. A detailed analysis of the algorithm's complexity is provided by Tropp and Gilbert (2007).

4.1 Overlapping Group OMP

The Group OMP (GOMP) algorithm was originally introduced by Swirszcz et al. (2009) for linear regression models, and extended by Lozano et al. (2011) in order to select groups of variables in logistic regression models. Following the notion of group lasso, GOMP utilizes prior knowledge about groups of features in order to penalize large weights in a collective way. Given that we have words sharing some properties, we can leverage these grouping for regularization purposes.

Similar to Lozano et al. (2011), let us assume that a natural grouping structure exists within the variables consisting of J groups $X_{G_1}, \ldots, X_{G_J}$, where $G_i \subset \{1, \ldots, d\}$, and $X_{G_i} \in \mathbb{R}^{N \times |G_i|}$. The standard GOMP algorithm also assumes that the groups are disjoint, $G_i \cap G_j = \emptyset$ for $i \neq j$. We will remove this assumption later on, by proposing the overlapping GOMP algorithm that is able to handle overlapping groups of features. GOMP operates in the same way with OMP but instead of selecting a single feature, it selects a group of features with the maximum correlation between them and the residual:

$$j^{(k)} = \arg\max_j \left\| X_{G_j}^\top \boldsymbol{r}^{(k-1)} \right\|_2^2. \tag{9}$$

In the case where the groups are not orthonormalized (i. e., $X_{G_j}^\top X_{G_j} = I_{G_j}$, where I_{G_j} is the identity matrix of size $\mathbb{R}^{|G_j|\times|G_j|}$), we select the best group based on the next criterion:

$$j^{(k)} = \arg\max_j \left| \left(\boldsymbol{r}^{(k-1)}\right)^\top X_{G_j} (X_{G_j}^\top X_{G_j})^{-1} X_{G_j}^\top \boldsymbol{r}^{(k-1)} \right|. \tag{10}$$

During our empirical analysis, we have noticed that the aforementioned criteria benefit large groups. This becomes apparent especially in the case where the size of the groups is not balanced. In this way, groups with a large number of “irrelevant” features are highly probable to be added. For instance, it is more probable to add a group that consists of 2 good features and 100 bad features, instead of a group that contains only 2 good features. To deal with situations like this one, we consider the average correlation between the group’s features and the residual:

$$j^{(k)} = \arg\max_j \frac{1}{|G_j|} \left\| X_{G_j}^\top \boldsymbol{r}^{(k-1)} \right\|_2^2. \tag{11}$$

Overlapping GOMP extends the standard GOMP in the case where the groups of indices are overlapping, i. e., $G_i \cap G_j \neq \emptyset$ for $i \neq j$. The main difference with GOMP is that each time a group becomes active, we remove its indices from each inactive group: $G_i = G_i \setminus G_{j^{(k)}}, \quad \forall i \in \{1, \dots, J\}$. In this way, the theoretical properties of GOMP hold also in the case of the overlapping GOMP algorithm. A sketch of the overlapping GOMP is shown in Alg. 2.

5 Experiments

Next, we present the data, setup and results of our empirical analysis on the text classification task.

5.1 Datasets

Topic categorization. From the 20 Newsgroups[2] dataset, we examine four classification tasks. We end up with binary classification problems, where we classify a document according to two related categories: i) *comp.sys*: ibm.pc.hardware vs. mac.hardware; ii) *rec.sport*: baseball vs. hockey; iii) *sci*: med vs. space and iv) *religion*: alt.atheism vs. soc.religion.christian.

[2]qwone.com/~jason/20Newsgroups/

	Dataset	Train	Dev	Test	Voc
20NG	science	949	238	790	25787
	sports	957	240	796	21938
	religion	863	216	717	18822
	comp.	934	234	777	16282
Sentiment	vote	1175	257	860	19813
	movie	1600	200	200	43800
	books	1440	360	200	21545
	dvd	1440	360	200	21086
	electr.	1440	360	200	10961
	kitch.	1440	360	200	9248

Table 1: Descriptive statistics of the datasets.

Sentiment analysis. The sentiment analysis datasets we examined include movie reviews (Pang and Lee, 2004; Zaidan and Eisner, 2008)[3], floor speeches by U.S. Congressmen deciding “yea”/“nay” votes on the bill under discussion (Thomas et al., 2006)[3] and product reviews from Amazon (Blitzer et al., 2007)[4].

Table 1 summarizes statistics about the aforementioned datasets used in our experiments. We choose small datasets intentionally, like Yogatama and Smith (2014b), so that we can observe the regularization effect clearly.

5.2 Experimental setup

In our setup, as features we use unigram frequency concatenated with an additional bias term. We reproduce standard regularizers like lasso, ridge, elastic and state-of-the-art structured regularizers like sentence, LDA, GoW and w2v groups (Skianis et al., 2016) as baselines and compare them with the proposed OMP and GOMP. We used pre-trained Google vectors introduced by Mikolov et al. (2013) and apply k-means clustering (Lloyd, 1982) algorithm with maximum 2000 clusters. For each word belonging to a cluster, we also keep the top 5 nearest words so that we introduce overlapping groups.

For the learning part we used Matlab and specifically code provided by Schmidt et al. (2007). If no pre-defined split exists, we separate the training

[3]cs.cornell.edu/~ainur/data.html

[4]cs.jhu.edu/~mdredze/datasets/sentiment/

	Dataset	no reg	lasso	ridge	elastic	OMP	Group lasso regularizers					GOMP
							LDA	LSI	sen	GoW	w2v	
20NG	science	0.946	0.916	0.954	0.954	0.964*	**0.968**	**0.968***	0.942	0.967*	**0.968***	0.953*
	sports	0.908	0.907	0.925	0.920	0.949*	0.959	0.964*	**0.966**	0.959*	0.946*	0.951*
	religion	0.894	0.876	0.895	0.890	0.902*	0.918	0.907*	**0.934**	0.911*	0.916*	0.902*
	computer	0.846	0.843	0.869	0.856	0.876*	0.891	0.885*	0.904	0.885*	**0.911***	0.902*
Sentiment	vote	0.606	0.643	0.616	0.622	0.684*	0.658	0.653	0.656	0.640	0.651	**0.687***
	movie	0.865	0.860	0.870	0.875	0.860*	**0.900**	0.895	0.895	0.895	0.890	0.850
	books	0.750	0.770	0.760	0.780	0.800	0.790	0.795	0.785	0.790	0.800	**0.805***
	dvd	0.765	0.735	0.770	0.760	0.785	0.800	0.805*	0.785	0.795*	0.795*	**0.820***
	electr.	0.790	0.800	0.800	0.825	**0.830**	0.800	0.815	0.805	0.820	0.815	0.800
	kitch.	0.760	0.800	0.775	0.800	0.825	0.845	**0.860***	0.855	0.840	0.855*	0.830

Table 2: Accuracy on the test sets. Bold font marks the best performance for a dataset, while * indicates statistical significance at $p < 0.05$ using micro sign test against lasso. For GOMP, we use w2v clusters and add all unigram features as individual groups.

	Dataset	no reg	lasso	ridge	elastic	OMP	Group lasso regularizers					GOMP
							LDA	LSI	sen	GoW	w2v	
20NG	science	100	**1**	100	63	2.7	19	20	86	19	21	5.8
	sports	100	**1**	100	5	1.8	60	11	6.4	55	44	7.7
	religion	100	**1.1**	100	3	1.5	94	31	99	10	85	1.5
	computer	100	1.6	100	7	**0.6**	40	35	77	38	18	4.9
Sentiment	vote	100	**0.1**	100	8	5	15	16	13	97	13	1.5
	movie	100	1.3	100	59	**0.9**	72	81	55	90	62	2.3
	books	100	**3.3**	100	14	4.6	41	74	72	90	99	8.3
	dvd	100	**2**	100	28	2.8	64	8	8	58	64	9
	electr.	100	**4**	100	6	6.3	10	8	43	8	9	12
	kitch.	100	4.5	100	79	**4.3**	73	44	27	75	46	6.5

Table 3: Model sizes (percentages of non-zero features in the resulting models). Bold for best, blue for best group.

science	lasso	orbit, space, contribute, funding, landing
	OMP	space, orbit, moon, planets, scientifically

Table 4: Largest positive weights in lasso and OMP for the science subset of 20NG.

set in a stratified manner by 80% for training and 20% for validation.

All the hyperparameters are tuned on the development dataset, using accuracy for evaluation. For lasso and ridge regularization, we choose λ from $\{10^{-2}, 10^{-1}, 1, 10, 10^{2}\}$. For elastic net, we perform grid search on the same set of values as ridge and lasso experiments for λ_{rid} and λ_{las}. For group lasso, OMP and GOMP regularizers, we perform grid search on the same set of parameters as ridge and lasso experiments. In the case we get the same accuracy on the development data, the model with the highest sparsity is selected. In GOMP we considered all individual features as separate groups of size one, along with the w2v groups. Last but not least, in both OMP and GOMP the maximum number of features, K(budget), is set to 2000.

5.3 Results

Table 2 reports the results of our experiments on the aforementioned datasets. The empirical results reveal the advantages of using OMP or GOMP for regularization in the text categorization task. The OMP regularizer performs systematically better than the baseline ones. More specifically, OMP outperforms the lasso, ridge and elastic net regularizers in all datasets, as regards to the accuracy. At the same time, the performance of OMP is quite close or even better to that of structured regularizers. Actually, in the case of electronics data, the model produced by OMP is the one with the highest accuracy. On the other hand, the proposed overlapping GOMP regularizer outperforms all the other regularizers in 3 out of 10 datasets.

Another important observation is how GOMP performs with different types of groups. GOMP only requires some "good" groups along with single features in order to achieve good accuracy. Smaller groups provided by LDA, LSI and w2v clusters provide a good solution and also fast computation, while others (GoW communities) can produce similar results with slower learning times. This phenomenon can be attributed to the different

	Dataset	CNN (20eps)	FastText (100eps)	Best OMP or GOMP	Best Lasso
20NG	science	0.935	0.958	0.964	**0.968**
	sports	0.924	0.935	0.951	**0.966**
	religion	**0.934**	0.898	0.902	**0.934**
	computer	0.885	0.867	0.902	**0.911**
Sentiment	vote	0.651	0.643	**0.687**	0.658
	movie	0.780	0.875	0.860	**0.900**
	books	0.742	0.787	**0.805**	0.800
	dvd	0.732	0.757	**0.820**	0.805
	electr.	0.760	0.800	**0.830**	0.820
	kitch.	0.805	0.845	0.830	**0.860**

Table 5: Comparison in test accuracy with state-of-the-art classifiers: CNN (Kim, 2014), FastText (Joulin et al., 2017) with no pre-trained vectors. The proposed OMP and GOMP algorithms produce the highest accurate model in 4 out of 10 datasets.

structure of groups. While LDA and LSI have a large number of groups with small number of features in them (1000 groups, 10 words per group), w2v clusters and GoW communities consist of smaller number of groups with larger number of words belonging to each group. Nevertheless, we have reached to the conclusion that the selection of groups is not crucial for the general performance of the proposed GOMP algorithm.

Table 3 shows the sparsity sizes of all the regularizers we tested. As it becomes apparent, both OMP and GOMP yield super-sparse models, with good generalization capabilities. More specifically, OMP produces sparse spaces similar to lasso, while GOMP keeps a significantly lower number of features compared to the other structured regularizers. In group regularization, GOMP achieves both best accuracy and sparsity in two datasets (vote & books), while group lasso only in one (sports).

In Table 4 we demonstrate the ability of OMP to produce more discriminative features compared to lasso by showing the largest weights and their respective term.

Finally, in Table 5 we compare state-of-the-art group lasso classifiers with deep learning architectures (Kim, 2014) with Dropout (Srivastava et al., 2014) for regularization and FastText (Joulin et al., 2017). We show that group lasso regularizers with simple logistic models remain very effective. Nevertheless, adding pre-trained vectors in the deep learning techniques and performing parameter tuning would definitely increase their performance against our models, but with a significant cost in time complexity.

5.4 Sparsity vs Accuracy

Figure 3 visualizes the accuracy vs. sparsity for all datasets and all classifiers. We do that in order to identify the best models, by both metrics. The desirable is for classifiers to belong in the top right corner, offering high accuracy and high sparsity at the same time. We observe that OMP and GOMP tend to belong in the right parts of the plot, having very high sparsity, often comparable to the aggressive lasso, even when they do not achieve the best accuracies.

5.5 Number of active features (atoms)

In both OMP and GOMP algorithms, the maximum desired number of active features (K, budget) was used as stopping criterion. For instance, by setting $K = 1000$, the proposed methods return the learned values that correspond to the first $\{100, 200, \ldots, 1000\}$ features, respectively. Thus, we exploit the feedforward feature selection structures of OMP and GOMP.

Figure 4 presents the number of active features versus accuracy in the development subsets of the 20NG dataset. It can be easily observed that after selecting 1000 active atoms, the accuracy stabilizes or even drops (overfitting problem). For instance, the best number of active features are: i) science: 700, ii) sports: 1100, iii) religion: 400 and iv) computer: 1500. The reason for selecting $K = 2000$ as the number of features to examine was to provide a sufficient number for OMP to reach a good accuracy while providing a super-sparse solution comparable to lasso.

5.6 Time complexity

Although certain types of group lasso regularizers perform well, they require a notable amount of time in the learning process.

OMP offers fast learning time, given the hyperparameter values and the number of atoms. For example, on the computer subset of the 20NG dataset, learning models with the best hyperparameter value(s) for lasso, ridge, and elastic net took 7, 1.4, and 0.8 seconds, respectively, on a 4-core 3.00GHz CPU. On the other hand, OMP requires only 4 seconds for training, making it even faster than lasso, while providing a sparser model.

GOMP can have very slow learning time when adding the features as groups individually. This is due to the large number of groups that GOMP needs to explore in order to extract the most "con-

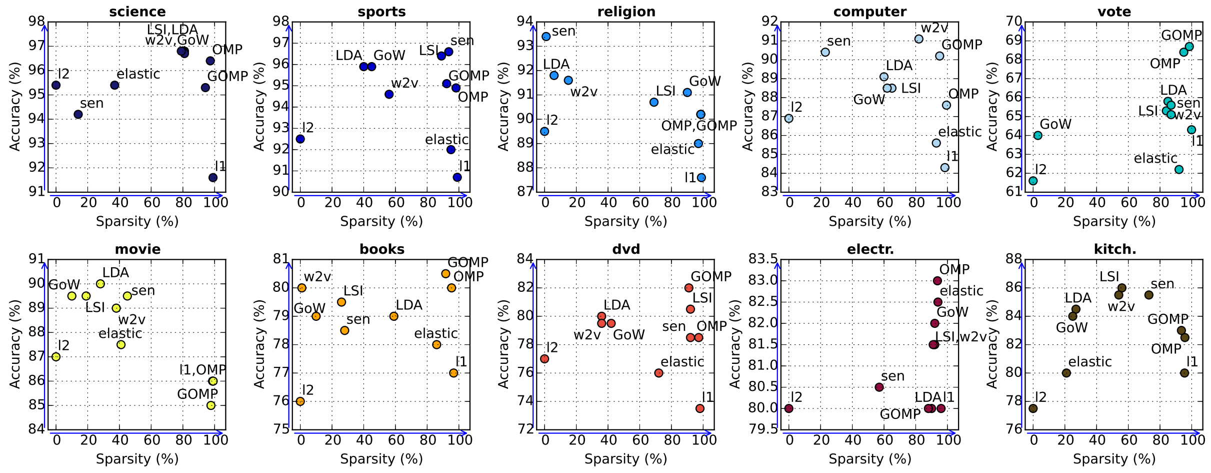
Figure 3: Accuracy vs sparsity on the test sets. Regularizers close to the top right corner are preferred.

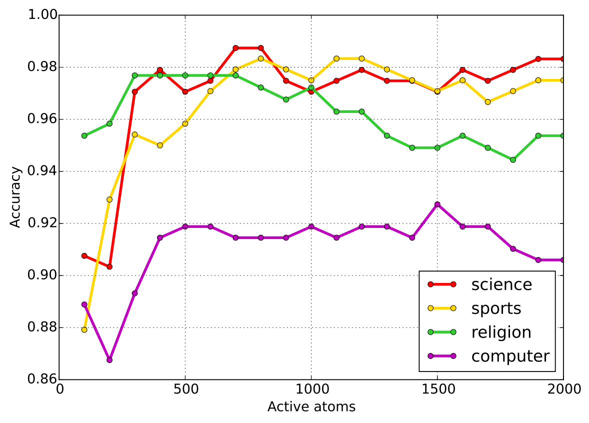
Figure 4: Accuracy vs. number of active atoms/features for OMP on 20NG data.

tributing" ones. If we consider GOMP without the individual features as groups, then the learning process becomes faster, with a clear decreasing effect on accuracy. In general, groups need to be well structured for GOMP to manage to surpass OMP and other state-of-the-art group lasso regularizers.

The advantages of the proposed methods are: (1) OMP requires no prior structural knowledge, (2) producing more discriminative features and (3) fast with relatively small number of dimensions.

Moreover, our implementation compared to the one of Lozano et al. (2011), provides the advantage of storing the weights and not having to recompute the whole matrices from scratch.

In the drawbacks of the methods: (1) OMP and GOMP are greedy algorithms, thus GOMP gets slow when we add the features as individual groups and (2) groups need to be "good".

6 Conclusion & Future Work

In this work, we introduced OMP and GOMP algorithms on the text classification task. An extension of the standard GOMP algorithm was also proposed, which is able to handle overlapping groups. The main advantages of both OMP and GOMP compared to other regularization schemes are their simplicity (greedy feedforward feature selection) and ability to produce accurate models with good generalization capabilities. We have shown that the proposed classifiers outperform standard baselines, as well as state-of-art structured regularizers in various datasets. Similar to Mosci et al. (2010); Yen et al. (2017); Xie et al. (2017), our empirical analysis validates that regularization remains a highly important topic, especially for deep learning models (Roth et al., 2017).

As mentioned previously, groups are not always specified in advance or hard to extract. Especially in environments involving text. To address this problem, we plan to extend our work by learning automatically the groups with Simultaneous Orthogonal Matching Pursuit (Szlam et al., 2012). Another interesting future direction would be to additionally penalize features inside the groups, similarly to sparse group lasso. Moreover, it would be highly interesting to examine the theoretical properties of overlapping GOMP. Finally, as shown in recent work by Roth et al. (2017), regularization remains an open topic for deep learning models.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments.

References

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. 2001. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159.

Stanley F. Chen and Ronald Rosenfeld. 2000. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. 2009. *The elements of statistical learning*, volume 2. Springer.

Arthur E. Hoerl and Robert W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. 2016. Japanese text normalization with encoder-decoder model. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 129–137.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. EMNLP '14, pages 1746–1751. ACL.

Yuwon Kim, Jinseog Kim, and Yongdai Kim. 2006. Blockwise sparse regression. *Statistica Sinica*, pages 375–390.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Aurelie C. Lozano, Grzegorz Swirszcz, and Naoki Abe. 2011. Group orthogonal matching pursuit for logistic regression. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 452–460.

Wei Lu, Hai Leong Chieu, and Jonathan Löfgren. 2016. A general regularization framework for domain adaptation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 950–954.

Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. 2017. Group sparse cnns for question classification with answer sets. *arXiv preprint arXiv:1710.02717*.

Stéphane G Mallat and Zhifeng Zhang. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415.

Lukas Meier, Sara Van De Geer, and Peter Bühlmann. 2008. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, NIPS '13, pages 3111–3119. Neural Information Processing Systems.

Sofia Mosci, Silvia Villa, Alessandro Verri, and Lorenzo Rosasco. 2010. A primal-dual algorithm for group sparse regularization with overlapping groups. In *Advances in Neural Information Processing Systems 23*, pages 2604–2612.

Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. 2011. Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*.

Bo Pang and Lilian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 271–278.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers*, pages 40–44.

Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. 2016. Linguistically regularized lstms for sentiment classification. *arXiv preprint arXiv:1611.03949*.

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. 2017. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems 30*, pages 2018–2028.

Volker Roth and Bernd Fischer. 2008. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *Proceedings of the 25th international conference on Machine learning*, pages 848–855.

François Rousseau and Michalis Vazirgiannis. 2013. Graph-of-word and tw-idf: New approach to ad hoc ir. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 59–68, New York, NY, USA. ACM.

Mark W. Schmidt, Glenn Fung, and Rómer Rosales. 2007. Fast optimization methods for L1 regularization: A comparative study and two new approaches. In *Proceedings of the 18th European Conference on Machine Learning*, pages 286–297.

Konstantinos Skianis, François Rousseau, and Michalis Vazirgiannis. 2016. Regularizing text categorization with clusters of words. In *EMNLP*, pages 1827–1837.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Grzegorz Swirszcz, Naoki Abe, and Aurelie C. Lozano. 2009. Grouped Orthogonal Matching Pursuit for Variable Selection and Prediction. In *Advances in Neural Information Processing Systems 22*, pages 1150–1158.

Arthur Szlam, Karol Gregor, and Yann LeCun. 2012. Fast approximations to structured sparse coding and applications to object classification. *Computer Vision–ECCV 2012*, pages 200–213.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of EMNLP*, pages 327–335.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Andre Nikolaevich Tikhonov and Vasili Iakovlevich Arsenin. 1977. *Solutions of ill-posed problems*. Winston.

Joel A Tropp. 2004. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242.

Joel A Tropp and Anna C Gilbert. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666.

Vladimir Naumovich Vapnik. 1991. Principles of Risk Minimization for Learning Theory. In *Advances in Neural Information Processing Systems 4*, pages 831–838.

Pengtao Xie, Aarti Singh, and Eric P. Xing. 2017. Uncorrelation and evenness: a new diversity-promoting regularizer. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3811–3820, International Convention Centre, Sydney, Australia. PMLR.

Ian En-Hsu Yen, Wei-Cheng Lee, Sung-En Chang, Arun Sai Suggala, Shou-De Lin, and Pradeep Ravikumar. 2017. Latent feature lasso. In *International Conference on Machine Learning*, pages 3949–3957.

Dani Yogatama and Noah A. Smith. 2014a. Linguistic structured sparsity in text categorization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 786–796.

Dani Yogatama and Noah A. Smith. 2014b. Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *ICML '14*, pages 656–664.

Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(1):49–67.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference*, pages 31–40.

Tong Zhang. 2009. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568.

Ye Zhang, Matthew Lease, and Byron C Wallace. 2017. Exploiting domain knowledge via grouped weight sharing with application to text categorization. *ACL*.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320.