Making the Most of Bag of Words: Sentence Regularization with Alternating Direction Method of Multipliers

Dani Yogatama DYOGATAMA@CS.CMU.EDU
Noah A. Smith NASMITH@CS.CMU.EDU
Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract

In many high-dimensional learning problems, only some parts of an observation are important to the prediction task; for example, the cues to correctly categorizing a document may lie in a handful of its sentences. We introduce a learning algorithm that exploits this intuition by encoding it in a regularizer. Specifically, we apply the sparse overlapping group lasso with one group for every bundle of features occurring together in a training-data sentence, leading to thousands to millions of overlapping groups. We show how to efficiently solve the resulting optimization challenge using the alternating directions method of multipliers. We find that the resulting method significantly outperforms competitive baselines (standard ridge, lasso, and elastic net regularizers) on a suite of real-world text categorization problems.

1. Introduction

When learning from high-dimensional data, we often capitalize on the insight that relatively few dimensions are relevant to the predictive task; this explains the success (at least in some settings) of *sparse* models such as the lasso (Tibshirani, 1996). For example, in categorizing text documents using "bag of words" representations, most words can typically be ignored (Forman, 2003). In genomics, where it is common to have millions of features, Kim & Xing (2008) and Wu et al. (2009), among others, have used sparse models to obtain better performance for the genome wide association mapping problem.

Another type of sparsity we might exploit comes from the structure of the data: some "parts" of an input may be more relevant to the task. In the case of text analysis, this idea was exploited by Yessenalina et al. (2010) and Tackstrom & McDonald (2011) using latent variable models that explicitly encode which sentences in a document are relevant to a polarity judgment (e.g., is the author's sentiment toward a film positive or negative?). Such models require sacrifices: convexity during parameter estimation and simplicity of prediction algorithms (compared to linear models).

We propose a different way to exploit the structure of the data that avoids these sacrifices. Building on the group lasso (Yuan & Lin, 2006), we instantiate groups of input features corresponding to sentences; a feature belongs to a separate group for every sentence it appears in. This diverges from past use of the group lasso for text modeling, in which feature *types* were grouped. For example, Martins et al. (2011b) had groups corresponding to word pair features, word shape features, part-of-speech features, and many more, for parsing.

In our approach, the model family is unchanged; there are no latent variables to reason about at prediction time. The structure in the training documents (here, sentence boundaries) is exploited only to encourage group behavior of features. Our algorithm does allow inspection, in a sense, of the sentences the model prefers, though only in the training data.

Our approach introduces a technical challenge, since a standard corpus may contain millions of sentences, each corresponding to a group that *overlaps* with other groups. We show how to use the alternating direction method of multipliers (Hestenes, 1969; Powell, 1969) to efficiently learn the parameters.

By experimenting on twelve text categorization tasks—including topic categorization, sentiment analysis, and forecasting—we demonstrate that our method consistently achieves more accurate models than lasso, ridge, and elastic net regularized baselines.

2. Background and Notation

We denote the feature vector to represent a document by $\mathbf{x} \in \mathbb{R}^V$, where V is the vocabulary size, and we represent

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

documents as vectors of word frequencies (i.e., "bags of words"). Each document is associated with a response (output) variable y. For simplicity and without loss of generality, we assume $y \in \{-1, 1\}$. The parameter vector that we want to learn is denoted by $\mathbf{w}$. We denote the loss function by $\mathcal{L}(\mathbf{x}, \mathbf{w}, y)$; in this work it is the log loss:

$$\mathcal{L}(\mathbf{x}, \mathbf{w}, y) = \log(1 + \exp(-y\mathbf{w}^\top \mathbf{x}))$$

The general framework can be extended to continuous responses (i.e., linear regression) and to other loss functions (e.g., SVMs' hinge loss).

The goal of the learning procedure is to estimate $\mathbf{w}$ for a given set of training documents $\{\mathbf{x}_d, y_d\}_{d=1}^D$ by minimizing the penalized training data loss:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \Omega(\mathbf{w}) + \sum_{d=1}^D \mathcal{L}(\mathbf{x}_d, \mathbf{w}, y_d)$$

where Ω is a regularization penalty to encourage models with small weight vectors. With a large number of features, as in text applications, regularization is crucial to avoid overfitting. In **ridge** regularization (Hoerl & Kennard, 1970), a standard method to which we compare the regularization discussed in §3, the penalty $\Omega_{rid}(\mathbf{w})$ is proportional to the squared ℓ_2-norm of $\mathbf{w}$:

$$\Omega_{rid}(\mathbf{w}) = \lambda \|\mathbf{w}\|_2^2 = \lambda \sum_j w_j^2,$$

where λ is a regularization hyperparameter that is tuned on development data or by cross-validation. In **lasso** regression (Tibshirani, 1996), the penalty $\Omega_{las}(\mathbf{w})$ is proportional to the ℓ_1-norm of $\mathbf{w}$:

$$\Omega_{las}(\mathbf{w}) = \lambda \|\mathbf{w}\|_1 = \lambda \sum_j |w_j|.$$

The lasso leads to *sparse* solutions, an attractive property for efficiency and (perhaps) interpretability.

The **group lasso** assumes that features can be binned into groups, and encourages all of the weights in a group to either be zero or nonzero using a $\ell_{1,2}$ norm (Yuan & Lin, 2006):

$$\Omega_{glas}(\mathbf{w}) = \sum_{g=1}^G \lambda_g \|\mathbf{w}_g\|_2,$$

where g indexes the groups and $\mathbf{w}_g$ is the subvector of $\mathbf{w}$ corresponding to the weights for features in group g. Typically the groups are non-overlapping, but this need not be the case (Jacob et al., 2009; Jenatton et al., 2011).

Often linear combinations of different regularizers are used; examples include the elastic net (which combines ridge and lasso; Zou & Hastie, 2005) and the sparse group lasso (which combines lasso and group lasso; Friedman et al., 2010).

3. Sentence Regularization

Considerable study has been devoted to structure in text, both for purposes of theoretical linguistics and for practical applications. All this work builds on the idea that more accurate interpretation can be obtained by explicitly representing rhetorical, semantic, or syntactic structures that relate word tokens to each other. Here we consider one very simple kind of structure that can easily be recovered with high accuracy in documents: sentences.

Our basic idea is to define, for every sentence in the training data, a **group** of the features that are present (i.e., nonzero) in that sentence. (In our models, these features are all word frequencies.) These groups, in turn, serve to define a group lasso regularization term, which we call the "sentence regularizer":

$$\Omega_{sen}(\mathbf{w}) = \sum_{d=1}^D \sum_{s=1}^{S_d} \lambda_{d,s} \|\mathbf{w}_{d,s}\|_2,$$

where d ranges over documents (as before) and s over sentences within a document. S_d is the number of sentences in document d. $\mathbf{w}_{d,s}$ corresponds to the subvector of $\mathbf{w}$ such that the corresponding features are present in sentence s of document d. The regularizer can take into account the length of the sentence by encoding it in $\lambda_{d,s}$. In the following, for simplicity and without loss of generality, we assume $\forall d, \forall s, \lambda_{d,s} = \lambda_{sen}$.

To gain an intuition for this regularizer, consider the case where we apply the penalty only for a single document, d_0, which happens (unrealistically) never to use the same word more than once (i.e., $\|\mathbf{x}_{d_0}\|_\infty = 1$). Because it instantiates group lasso, the sentence regularizer will encourage some groups to go to zero (especially groups whose sentences contain no words strongly associated with a label in the rest of the corpus). The effect is that only some sentences in d_0 will be selected as relevant (i.e., $\{s : \mathbf{w}_{d_0,s} \neq \mathbf{0}\}$), and the rest will have $\mathbf{w}_{d_0,s} = \mathbf{0}$ and therefore will have no effect on the prediction for d_0. Further, the words deemed not relevant in d_0 will have no effect on the prediction for other documents.

Of course, in typical documents, many words will occur in more than one sentence, and we create a group for every sentence in the training corpus. This means that our groups are heavily *overlapping*; a word that occurs in k sentences in the corpus will force its corresponding weight to associate with k groups. As a result, the regularizer mainly acts as a proxy to encourage group behavior of words appearing in the same sentences.

Comparison to latent variable models. Seen this way, we can draw connections between our model and latent variable models for sentiment analysis that explicitly "select"

relevant sentences (Yessenalina et al., 2010; Tackstrom & McDonald, 2011). Latent variables complicate inference, because prediction algorithms must reason about the additional variables. This sometimes leads to mixed inference problems (i.e., maximizing over y while marginalizing latent variables). Our method, by contrast, does not change the linear model family, so the prediction algorithm is unchanged. At inference time, there is no notion of "relevant" sentences.

More importantly, latent variable models lead to non-convex objective functions, so that learning methods' performance hinges on clever (or lucky) initialization (e.g., Yessenalina et al., 2010). Our approach maintains convexity of the objective function, allowing for familiar guarantees about the parameter estimate.

4. Learning

There has been much work on optimization with overlapping group lasso penalty (Jacob et al., 2009; Jenatton et al., 2011; Chen et al., 2011; Qin & Goldfarb, 2012; Yuan et al., 2013). The novel technical challenge of the sentence regularizer in §3 is the huge number of overlapping groups: one group for every sentence. Proximal methods (Duchi & Singer, 2009; Bach et al., 2011) offer one potential solution. Indeed, Martins et al. (2011a) introduced a proximal gradient algorithm for handling overlapping group lasso. Their algorithm applied proximal steps (one per group) sequentially, an approach we find prohibitive for cases with hundreds of thousands or millions of groups.

We propose instead to apply the alternating directions method of multipliers (ADMM; Hestenes, 1969; Powell, 1969). Goldstein & Osher (2009) first proposed ADMM for sparse modeling. For a full review of ADMM, see Boyd et al. (2010). The central idea in ADMM is to break the optimization problem down into subproblems, each depending on a subset of the dimensions of $\mathbf{w}$. Each subproblem p receives a "copy" of the subvector of $\mathbf{w}$ it depends on, denoted $\mathbf{v}_p$. We then encode constraints forcing each $\mathbf{v}_p$ to "agree" with the global solution $\mathbf{w}$.

In our setup, the dimensionality of $\mathbf{w}$ is the vocabulary size V. There is a subproblem for every sentence in the corpus; the vector $\mathbf{v}_{d,s}$ (for sentence s in document d) will have the same length as the sentence it corresponds to, with one dimension per word token. Therefore, the dimensionality of the concatenation of these copies, $\mathbf{v}$, is the length of the corpus, denoted N. Constraints are encoded in a matrix $\mathbf{M} \in \mathbb{R}^{N \times V}$, such that $\mathbf{M}[n, v] = 1$ iff token n is a word of type v and 0 otherwise. The constraint is therefore

$$\mathbf{v} = \mathbf{M}\mathbf{w}.$$

ADMM for overlapping group lasso only produces weakly sparse solutions,[1] for reasons we explain below. To achieve strong sparsity in the solution, which is desirable for high-dimensional data such as text, we couple the sentence regularizer with a classic lasso regularizer. Therefore, the objective function to be minimized by ADMM is:

$$\min_{\mathbf{w},\mathbf{v}} \Omega_{sen}(\mathbf{v}) + \Omega_{las}(\mathbf{w}) + \sum_{d=1}^{D} \mathcal{L}(\mathbf{x}_d, \mathbf{w}, y_d) \quad (1)$$
$$\text{s.t. } \mathbf{v} = \mathbf{M}\mathbf{w}$$

For brevity, we will henceforth write $\mathcal{L}(\mathbf{w})$ and hide the dependency on $\mathbf{x}$ and y. Note that we have overloaded notation somewhat; the sentence regularizer applied to $\mathbf{v}$ is given by:

$$\Omega_{sen}(\mathbf{v}) = \sum_{d=1}^{D} \sum_{s=1}^{S_d} \lambda_{d,s} \|\mathbf{v}_{d,s}\|_2.$$

Let $\mathbf{u}$ be the Lagrange variables. The augmented Lagrangian of Equation 1 is:

$$\Omega_{sen}(\mathbf{v}) + \Omega_{las}(\mathbf{w}) + \mathcal{L}(\mathbf{w}) + \mathbf{u}^\top(\mathbf{v} - \mathbf{M}\mathbf{w}) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{M}\mathbf{w}\|_2^2$$

Note the introduction of a quadratic penalty.

ADMM proceeds by updating each of $\mathbf{w}$, $\mathbf{v}$, and $\mathbf{u}$ by solving, in turn, the following problems:

$$\min_{\mathbf{w}} \Omega_{las}(\mathbf{w}) + \mathcal{L}(\mathbf{w}) - \mathbf{u}^\top\mathbf{M}\mathbf{w} + \frac{\rho}{2}\|\mathbf{v} - \mathbf{M}\mathbf{w}\|_2^2 \quad (2)$$
$$\cong \min_{\mathbf{w}} \Omega_{las}(\mathbf{w}) + \mathcal{L}(\mathbf{w}) + \frac{\rho}{2}\left\|\mathbf{M}\mathbf{w} - \left(\mathbf{v} + \frac{\mathbf{u}}{\rho}\right)\right\|_2^2$$
$$\min_{\mathbf{v}} \Omega_{sen}(\mathbf{v}) + \mathbf{u}^\top\mathbf{v} + \frac{\rho}{2}\|\mathbf{v} - \mathbf{M}\mathbf{w}\|_2^2 \quad (3)$$
$$\cong \min_{\mathbf{v}} \Omega_{sen}(\mathbf{v}) + \frac{\rho}{2}\left\|\mathbf{v} - \left(\mathbf{M}\mathbf{w} - \frac{\mathbf{u}}{\rho}\right)\right\|_2^2$$
$$\mathbf{u} = \mathbf{u} + \rho(\mathbf{v} - \mathbf{M}\mathbf{w}) \quad (4)$$

We consider each in turn.

Update for $\mathbf{w}$. In Equation 2, we fix $\mathbf{v}$ and $\mathbf{u}$ and update $\mathbf{w}$. We denote the element of $\mathbf{v}$ corresponding to nth token in the corpus by v_n, for $n \in \{1, \ldots, N\}$. We denote the frequency of type i in the corpus by N_i. Let $v_{i,n}$ denote the element of $\mathbf{v}$ corresponding to the nth token of type i for $n \in \{1, \ldots, N_i\}$. We index $\mathbf{u}$ similarly.

[1] Weakly sparse methods (e.g., ridge) do not drive the feature weights exactly to zero, whereas strongly sparse methods (e.g., lasso) result in exact zeroes.

Note that the quadratic term in Equation 2 can be rewritten as

$$\begin{aligned}
&\sum_{n=1}^{N}\left(w_{v_n}-(v_n+\tfrac{u_n}{\rho})\right)^2\\
&=\sum_{n=1}^{N} w_{v_n}^2-2w_{v_n}(v_n+\tfrac{u_n}{\rho})+(v_n+\tfrac{u_n}{\rho})^2\\
&=\sum_{i=1}^{V}\left(N_i w_i^2-2w_i\sum_{n=1}^{N_i}(v_{i,n}+\tfrac{u_{i,n}}{\rho})+\sum_{n=1}^{N_i}(v_{i,n}+\tfrac{u_{i,n}}{\rho})^2\right)\\
&=\sum_{i=1}^{V} N_i\left(w_i-\tfrac{1}{N_i}\sum_{n=1}^{N_i}(v_{i,n}+\tfrac{u_{i,n}}{\rho})\right)^2+\text{constant}(\mathbf{w})\\
&=\sum_{i=1}^{V} N_i\left(w_i-\mu_i\right)^2
\end{aligned}$$

where $\mu_i=\dfrac{1}{N_i}\displaystyle\sum_{n=1}^{N_i}(v_{i,n}+\tfrac{u_{i,n}}{\rho})$.

Intuitively, each word type is regularized towards a value near the mean of its corresponding copy variables. This is similar to having Gaussian priors with different means and variances for each word type. It now becomes clear why $\mathbf{w}$ will only be sparse in the limit (weakly sparse in practice) unless we add $\Omega_{las}(\mathbf{w})$ to the penalty, since the effective penalty is quadratic, as in ridge regression. This is the main reason to use sparse group lasso, if strong sparsity is required. (The reader may notice that when $\boldsymbol{\mu}=\mathbf{0}$ and $N_i=C$, this update is essentially equivalent to elastic net regression, which penalizes $\mathbf{w}$ with a linear combination of Ω_{ridge} and Ω_{las}.) For this update, we apply a proximal gradient method (Bach et al., 2011), since $\mathcal{L}(\mathbf{w})+\frac{\rho}{2}\sum_{i=1}^{V}N_i(w_i-\mu_i)^2$ is convex and continuously differentiable, and $\Omega_{las}(\mathbf{w})$ is a convex function whose proximal operator (Moreau, 1963) can be evaluated efficiently. The proximal operator for $\Omega_{las}(\mathbf{w})$ is the soft-thresholding operator:

$$[\text{prox}_{\Omega_{las},\lambda_{las}}(\mathbf{w})]_j=\begin{cases} w_j-\lambda_{las} & \text{if } w_j>\lambda_{las}\\ 0 & \text{if } |w_j|\le\lambda_{las}\\ w_j+\lambda_{las} & \text{if } w_j<-\lambda_{las}\end{cases}$$

Update for $\mathbf{v}$. Note that Equation 3 is the proximal operator of $\frac{1}{\rho}\Omega_{sen}$ applied to $\mathbf{M}\mathbf{w}-\frac{\mathbf{u}}{\rho}$. When applied to the collection of "copies" of the parameters, $\mathbf{v}$, Ω_{sen} no longer has overlapping groups. Hence we can separately solve for each $\mathbf{v}_{d,s}$:

$$\min_{\mathbf{v}_{d,s}}\|\mathbf{v}_{d,s}\|_2+\frac{\rho}{2}\left\|\mathbf{v}_{d,s}-\left(\mathbf{M}_{d,s}\mathbf{w}-\frac{\mathbf{u}_{d,s}}{\rho}\right)\right\|_2^2$$

with $\mathbf{M}_{d,s}$ defined as the rows of $\mathbf{M}$ corresponding to this sentence. Let $\mathbf{z}_{d,s}\triangleq\mathbf{M}_{d,s}\mathbf{w}-\frac{\mathbf{u}_{d,s}}{\rho}$. This problem can be solved by applying the proximal operator used in non-overlapping group lasso to each subvector:

$$\begin{aligned}
\mathbf{v}_{d,s}&=\text{prox}_{\Omega_{sen},\frac{\lambda_{sen}}{\rho}}(\mathbf{z}_{d,s})\\
&=\begin{cases}\mathbf{0} & \text{if } \|\mathbf{z_{d,s}}\|_2\le\frac{\lambda_{sen}}{\rho}\\ \dfrac{\|\mathbf{z}_{d,s}\|_2-\frac{\lambda_{sen}}{\rho}}{\|\mathbf{z}_{d,s}\|_2}\mathbf{z}_{d,s} & \text{otherwise.}\end{cases}
\end{aligned}\tag{5}$$

Note that this step effectively "selects" training-data sentences used to make predictions. The $\mathbf{v}$ variables can be inspected to see which training sentences have been identified as "relevant," as we will see later.

Update for $\mathbf{u}$. Equation 4 is a simple update of the dual variable $\mathbf{u}$.

Algorithm 1 shows our ADMM algorithm for sparse overlapping group lasso.

Algorithm 1 ADMM for sparse overlapping group lasso

Input: augmented Lagrangian variable ρ, regularization strengths λ_{sen} and λ_{las}
while stopping criterion not met **do**
  $\mathbf{w}=\arg\min_{\mathbf{w}}\Omega_{las}(\mathbf{w})+\mathcal{L}(\mathbf{w})+\frac{\rho}{2}\sum_{i=1}^{V}N_i(w_i-\mu_i)^2$
  for $d=1$ **to** D **do**
    for $s=1$ **to** S_d **do**
      $\mathbf{v}_{d,s}=\text{prox}_{\Omega_{sen},\frac{\lambda_{sen}}{\rho}}(\mathbf{z}_{d,s})$ {eq. 5; can be done in parallel}
    end for
  end for
  $\mathbf{u}=\mathbf{u}+\rho(\mathbf{v}-\mathbf{M}\mathbf{w})$ {can be done in parallel}
end while

Space and time efficiency. The learning algorithm is effective for large numbers of groups because each group operation and the $\mathbf{u}$ update (the second and third ADMM steps) can be done in *parallel*. The most expensive step is the minimization of $\mathbf{w}$. This is roughly as expensive as lasso or ridge methods since we can precompute $\boldsymbol{\mu}$, although we need to do the $\mathbf{w}$ minimization for every ADMM iteration.[2]

Our model requires storing of two parameter vectors during learning: $\mathbf{w}$ and $\mathbf{v}$. Although the size of $\mathbf{v}$ is N (the number of words in the training corpus), $\mathbf{v}$ is a sparse vector since most of the elements of $\mathbf{v}$ are driven to zero in the second ADMM step. Furthermore, $\mathbf{w}$ is also a sparse vector due to the Ω_{las} regularizer. The actual number of nonzero elements

[2] We minimize $\mathbf{w}$ to a relative convergence tolerance of 10^{-5}. The $\mathbf{w}$ minimization step need not be carried out to convergence at every iteration. Inexact ADMM (Boyd et al., 2010), as this method is known, might provide speedups.

requiring storage will, of course, depend on λ_{sen}, λ_{las}, ρ, and the dataset.

Convergence and stopping criteria. We can show that Algorithm 1 is guaranteed to converge by simply noting that both $\mathcal{L}(\mathbf{w}) + \Omega_{las}(\mathbf{w})$ and $\Omega_{sen}(\mathbf{v})$ are closed, proper, and convex functions of $\mathbf{w}$ and $\mathbf{v}$ respectively;[3] and the function $\Omega_{sen}(\mathbf{v}) + \Omega_{las}(\mathbf{w}) + \mathcal{L}(\mathbf{w}) + \mathbf{u}^\top(\mathbf{v} - \mathbf{M}\mathbf{w})$ has a saddle point. As a result, our problem satisfies the two assumptions required for ADMM convergence (Boyd et al., 2010). We can use the proof in Boyd et al. (2010) to show that Algorithm 1 has residual, objective, and dual variable convergence.

As noted there, ADMM is often slow to converge in practice, although tens of iterations are usually enough to obtain reasonably good solutions. We use relative changes in the ℓ_2 norm of the parameter vector $\mathbf{w}$ as our convergence criterion, and set the maximum number of iterations to 100. Other criteria such as primal and dual residuals convergence and performance on development data can also be used to determine convergence of Algorithm 1 in practice.

5. Experiments

To compare the sentence regularizer with other methods, we experiment with three text categorization problems: topic classification, sentiment analysis, and text-driven forecasting. In each case, we predict a binary label for a piece of text using a bag of words model.[4] These tasks are successively more difficult.

5.1. Data

We use publicly available datasets to evaluate our model described in more detail below.

Topic classification. We consider four binary categorization tasks from the 20 Newsgroups dataset.[5] Each task involves categorizing a document according to two closely related categories:

- `comp.sys`: `ibm.pc.hardware` vs. `mac.hardware`
- `rec.sport`: `baseball` vs. `hockey`
- `sci`: `med` vs. `space`
- `alt.atheism` vs. `soc.religion.christian`

[3] Notice that λ_{sen} and λ_{las} translate into bounds on the norms of $\mathbf{v}$ and $\mathbf{w}$ since there is a one-to-one correspondence between a regularization constant and the parameter-vector norm due to the primal and dual representation of the objective function.

[4] Of course, state-of-the-art approaches to these problems often use additional predictive features and representations (e.g., Yessenalina et al., 2010; Socher et al., 2013). Any feature that decomposes locally by sentence can be given the same treatment we give to word features. Our focus here is on a controlled comparison between regularizers, which is orthogonal to the engineering and discovery of features.

[5] http://qwone.com/~jason/20Newsgroups

Table 1. Descriptive statistics about the datasets. The number of sentences is equal to the number of groups.

	Dataset	D	# Dev.	# Test	V	# Sents.
20N	science	952	235	790	30,154	19,075
	sports	958	239	796	20,832	18,861
	religion	870	209	717	24,528	22,297
	comp.	929	239	777	20,868	13,969
Sentiment	books	1,600	200	200	21,641	14,249
	dvd	1,600	200	200	22,101	15,958
	music	1,600	200	200	17,283	12,172
	electr.	1,600	200	200	10,885	10,885
	movie	1,600	200	200	17,744	49,489
	vote	1,175	257	860	24,508	36,434
Fore.	science	3,207	280	539	42,702	638,068
	bill	37,850	7,341	6,571	10,001	2,526,063

Sentiment analysis. One task in sentiment analysis is predicting the polarity of a piece of text, i.e., whether the author is favorably inclined toward a (usually known) subject of discussion or proposition (Pang & Lee, 2008). Sentiment analysis, even at the coarse level of polarity we consider here, can be confused by negation, stylistic use of irony, and other linguistic phenomena. Our sentiment analysis datasets consist of four types of product reviews from Amazon.com with star ratings converted to binary labels (Blitzer et al., 2007),[6] movie reviews with similarly converted ratings (Pang & Lee, 2004; Zaidan & Eisner, 2008),[7] and floor speeches by U.S. Congressmen alongside "yea"/"nay" votes on the bill under discussion (Thomas et al., 2006).[8]

Text-driven forecasting. Forecasting from text requires identifying textual correlates of a response variable revealed in the future, most of which will be weak and many of which will be spurious (Kogan et al., 2009). We consider two such problems. The first one is predicting whether a scientific paper will be cited or not within three years of its publication (Yogatama et al., 2011); the dataset comes from the ACL Anthology and consists of research papers from the Association for Computational Linguistics and citation data

[6] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[7] http://www.cs.cornell.edu/~ainur/data.html

[8] http://www.cs.cornell.edu/~ainur/data.html

(Radev et al., 2009). The second task is predicting whether a legislative bill will be recommended by a Congressional committee (Yano et al., 2012).[9]

Table 1 summarizes statistics about the datasets used in our experiments. In total, we evaluate our method on twelve binary classification tasks.

5.2. Setup

In all our experiments, we use counts of unigrams as our features, plus an additional bias term which is not regularized. When explicit sentence boundaries are not given, we use MxTerminator (Reynar & Ratnaparkhi, 1997)[10] to segment documents into sentences. We compare our model with state-of-the-art methods for document classification: lasso, ridge, and elastic net regularization. Hyperparameters are tuned on a separate development dataset, using accuracy as the evaluation criterion. For lasso and ridge models, we choose λ from $\{10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. 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 our method, there are three hyperparameters to tune, λ_{sen}, λ_{las}, and ρ (see §4). We perform grid search on $\{10^{-1}, 1, 10, 10^2, 10^3\}$ for ρ and the same set of values as ridge and lasso experiments for λ_{sen} and λ_{las} If there is a tie on development data we choose the model with the smallest size. One drawback of our method compared to lasso and ridge models is that we have more hyperparameters to tune. Future work might explore the benefit of using better hyperparameter search methods (e.g., Gaussian processes; Bergstra et al., 2011) to reduce the computational cost of this search.

5.3. Results

Table 2 shows the results of our experiments on the twelve datasets. The results demonstrate the superiority of our model. It outperformed lasso on 11 out of 12 datasets, ridge on 11 out of 12 datasets, and elastic net on 9 out of 12 datasets. The improvements over lasso, ridge, and elastic net are statistically significant ($p < 0.01$, binomial test), aggregating across all datasets. Furthermore, notice that we were able to obtain a significantly smaller model (23% as large on average) compared to the ridge model. Lasso prunes more aggressively, but almost always performs worse. Elastic net obtained model of slightly bigger size compared to ours on average (25% as large as ridge on average). We also ran preliminary experiments using our method without the lasso term (using only Ω_{sen}), and qualitatively, it performed better than lasso and ridge methods, but slightly worse than elastic net on average. Recall that in this case the sentence regularizer downweights features in group structures, but is not able to completely discard them (i.e., it produces a weakly sparse solution). This suggests the value of combining the group lasso with a classical lasso regularizer (i.e., sparse group lasso).

We take this collection of results as support for the idea that observable structure in the data can be usefully exploited in the regularizer without introducing latent variables to represent the "relevance" of passages of a document to the task.[11]

5.4. Discussion

In terms of running time (wall clock), our model is slightly slower than standard regularizers. For example, for the sports dataset, learning models with the best hyperparameter value(s) for lasso, ridge, and elastic net took 27, 18, and 10 seconds, respectively, on an Intel Xeon CPU E5645 2.40 GHz machine with 8 processors and 24 GB RAM. Our model with the best hyperparameter values took 33 seconds to reach convergence. As mentioned previously, the major drawback is the need to do grid search for each of the hyperparameters: λ_{sen}, λ_{las}, and ρ, whereas lasso and ridge only has one hyperparameter and elastic net has two hyperparameters. However, note that this grid search can also be done in parallel, since they are not dependent on each other, so given enough processors our method is only marginally slower than standard regularizers.

Recall that, during learning, we make a copy in $\mathbf{v}$ of each weight in $\mathbf{w}$ for each corresponding word token. $\mathbf{w}$ is used to make predictions; it seeks to be a consensus among all of the $\mathbf{v}_{d,s}$. In practice, with many overlapping groups, the constraint $\mathbf{v} = \mathbf{M}\mathbf{w}$ is rarely satisfied exactly. Inspecting which $\mathbf{v}_{d,s}$ are nonzero can give some insight into what is learned, by showing which sentences are treated as "relevant" by the algorithm.

Tables 3 and 4 show training instances from the `comp.sys.ibm.pc.hardware` vs. `comp.sys.mac.hardware` task (20 Newsgroups) and the Amazon.com dvd reviews task, for the development-tuned hyperparameter values. We can see that in these particular cases, the learner selected informative sentences and removed uninformative ones. We also show the log-odds scores for removing each sentence. The log-odds score is defined here as the log of the model probability of the class label for an instance (document)

[9] `http://www.ark.cs.cmu.edu/bills`

[10] `ftp://ftp.cis.upenn.edu/pub/adwait/jmx`

[11] For the movie and vote sentiment analysis datasets, latent variable models in Yessenalina et al. (2010) achieved the best results of 92.50 and 77.67, respectively. However, the numbers are not directly comparable to ours since they used more features. They also exploited careful initialization to obtain these results. The model that used a similar set of features and random initialization achieved 87.22 and 78.84 classification accuracies.

Table 2. Classification accuracies and model sizes (percentages of nonzero features in the resulting models) on various datasets for each competing model. "m.f.c." is the most frequent class baseline. The improvements of our method over lasso, ridge, and elastic net are statistically significant ($p < 0.01$, binomial test) in aggregate.

Task	Dataset	Accuracy (%)					Model size (%)			
		m.f.c.	lasso	ridge	elastic	our method	lasso	ridge	elastic	our method
20N	science	50.13	90.63	91.90	91.65	**96.20**	1	100	34	12
	sports	50.13	91.08	93.34	93.71	**95.10**	2	100	15	3
	religion	55.51	90.52	92.47	92.47	**92.75**	.3	100	48	94
	computer	50.45	85.84	86.74	87.13	**90.86**	2	100	24	10
Sentiment	books	50.00	76.50	80.50	81.50	**84.00**	.6	100	9	6
	dvd	50.00	78.5	73.50	79.00	**81.50**	3	100	5	8
	music	50.00	73.00	72.00	73.50	**80.00**	5	100	14	14
	electronics	50.00	81.50	**86.00**	**86.00**	84.00	7	100	43	12
	movie	50.00	76.00	87.50	**89.00**	88.00	.3	100	14	5
	vote	58.37	73.14	72.79	72.79	**73.95**	2	100	44	6
Forecasting	science	50.28	64.00	66.79	66.23	**67.71**	31	100	43	99
	bill	87.40	88.36	87.70	**88.48**	88.11	7	100	7	8

using all sentences minus the log of the probability of the class label using all sentences except one. Intuitively, the scores indicate how much the sentence affects the model's decision. From the log-odds scores, we can see our model tends to make its decision mostly based on the sentences it "selects."

We observed that in some cases (e.g., religion, vote, etc.) the model selected most sentences, whereas in other cases (e.g., dvd, electronics, etc.) the model excluded many sentences. We believe that the flexibility of our model to include or exclude sentences through validation on development data contributes to the performance improvements.

In these experiments, we have used sentences to define groups, because sentences are easy to observe within documents. In future work, larger structures (paragraphs or sections) or smaller ones (clauses or phrases) might be used. Defining each document as a group has an intuitive connection with support vectors in support vector machines (Cortes & Vapnik, 1995), although further investigation is required to deterimine its effectiveness.

6. Conclusion

We introduced a new sparse overlapping group lasso regularizer for text modeling inspired by the structure inherent in linguistic data. We also showed how to efficiently perform learning for sparse group lasso with thousands to millions of overlapping groups using the alternating direction method of multipliers. We empirically demonstrated that our model consistently outperformed competing models on various datasets for various real-world document categorization tasks.

Acknowledgments

The authors thank André F. T. Martins for helpful discussions on ADMM and three anonymous reviewers for helpful feedback on an earlier draft of this paper. This research was supported in part by computing resources provided by a grant from the Pittsburgh Supercomputing Center, a Google research award, and the Intelligence Advanced Research Projects Activity via Department of Interior National Business Center contract number D12PC00347. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

Bach, Francis, Jenatton, Rodolphe, Mairal, Julien, and Obozinski, Guillaume. *Convex Optimization with Sparsity-Inducing Norms*. The MIT Press, 2011.

Bergstra, James, Bardenet, Remi, Bengio, Yoshua, and Kegl, Balazs. Algorithms for hyper-parameter optimization. In *Proc. of NIPS*, 2011.

Blitzer, John, Dredze, Mark, and Pereira, Fernando. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL*, 2007.

Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

Chen, Xi, Lin, Qihang, Kim, Seyoung, Carbonell, Jaime G., and

Table 3. An article from News 20 dataset categorized under `comp.sys.mac.hardware`. Each line is a sentence identified by the sentence segmenter. There are twelve sentences in this article. Selected sentences in the learner's copy variables are highlighted in **blue and bold**. We also display the color-coded log-odds scores, as discussed in the text (**sentence**, **elastic**, **ridge**, **lasso**) based on removing each sentence for each competing model. We only display scores that are greater than 10^{-3} in absolute values.

Sentence	Negative Positive
from : *anonymized*	
subject : accelerating the macplus ... ;)	(0.05)
lines : 15 we ' re about ready to take a bold step into the 90s around here by accelerating our rather large collection of stock macplus computers .	(0.07) (0.03) (0.02) (0.02)
yes indeed , difficult to comprehend why anyone would want to accelerate a macplus, but that's another story .	(0.06) (0.02) (0.02) (0.04)
suffuce it to say , we can get accelerators easier than new machines .	(0.01)
hey , i don ' t make the rules ...	(0.01)
anyway , on to the purpose of this post: i ' m looking for info on macplus acelerators .	(0.04) (0.01)
so far , i ' ve found some lit on the novy accelerator and the micrmac multispeed accelartor .	(0.02) (0.02) (0.02) (0.04)
both look acceptable , but i would like to hear from anyone who has tried these .	(–0.01)
also , if someone would recommend another accelerator for the macplus , i ' d like to hear about it .	(0.06) (0.03) (0.02) (0.06)
thanks for any time and effort you expend on this !	(–0.01) (–0.01) (–0.01)
karl	

Table 4. A review from Amazon dvd review dataset categorized as a positive review. Each line is a sentence identified by the sentence segmenter. There are five sentences in this article. Selected sentences in the learner's copy variables are highlighted in **blue and bold**. We also display the color-coded log-odds scores, as discussed in the text (**sentence**, **elastic**, **ridge**, **lasso**) based on removing each sentence for each competing model. We only display scores that are greater than 10^{-3} in absolute values.

Sentence	Negative Positive
this film is one big joke : you have all the basics elements of romance (love at first sight , great passion , etc .) and gangster flicks (brutality , dagerous machinations , the mysterious don , etc.) , but it is all done with the crudest humor .	(0.42) (0.22) (0.07) (0.48)
it ' s the kind of thing you either like viserally and immediately " get " or you don ' t .	(0.01) (0.01)
that is a matter of taste and expectations .	(0.01)
i enjoyed it and it took me back to the mid80s , when nicolson and turner were in their primes .	(0.02) (0.01)
the acting is very good , if a bit obviously tongue - in - cheek .	(0.01)

Xing, Eric P. Smoothing proximal gradient method for general structured sparse learning. In *Proc. of UAI*, 2011.

Cortes, Corinna and Vapnik, Vladimir. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Duchi, John and Singer, Yoram. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009.

Forman, George. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

Friedman, Jerome, Hastie, Trevor, and Tibshiran, Robert. A note on the group lasso and a sparse group lasso. Technical report, Stanford University, 2010.

Goldstein, Tom and Osher, Stanley. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2 (2):323–343, 2009.

Hestenes, Magnus R. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

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

Jacob, Laurent, Obozinski, Guillaume, and Vert, Jean-Philippe. Group lasso with overlap and graph lasso. In *Proc. of ICML*, 2009.

Jenatton, Rodolphe, Audibert, Jean-Yves, and Bach, Francis. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

Kim, Seyoung and Xing, Eric P. Feature selection via block-regularized regression. In *Proc. of UAI*, 2008.

Kogan, Shimon, Levin, Dimitry, Routledge, Bryan R., Sagi, Jacob S., and Smith, Noah A. Predicting risk from financial reports with regression. In *Proc. of HLT-NAACL*, 2009.

Martins, Andre F. T., Smith, Noah A., Aguiar, Pedro M. Q., and Figueiredo, Mario A. T. Online learning of structured predictors with multiple kernels. In *Proc. of AISTATS*, 2011a.

Martins, Andre F. T., Smith, Noah A., Aguiar, Pedro M. Q., and Figueiredo, Mario A. T. Structured sparsity in structured prediction. In *Proc. of EMNLP*, 2011b.

Moreau, J. J. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Ser. A Math*, 255: 2897–2899, 1963.

Pang, Bo and Lee, Lilian. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of ACL*, 2004.

Pang, Bo and Lee, Lilian. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

Powell, M. J. D. A method for nonlinear constraints in minimization problems. In Fletcher, R. (ed.), *Optimization*, pp. 283–298. Academic Press, 1969.

Qin, Zhiwei (Tony) and Goldfarb, Donald. Structured sparsity via alternating direction methods. *Journal of Machine Learning Research*, 13:1435–1468, 2012.

Radev, Dragomir R., Muthukrishnan, Pradeep, and Qazvinian, Vahed. The ACL anthology network corpus. In *Proc. of ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*, 2009.

Reynar, Jeffrey C. and Ratnaparkhi, Adwait. A maximum entropy approach to identifying sentence boundaries. In *Proc. of the Fifth Conference on Applied Natural Language Processing*, 1997.

Socher, Richard, Perelygin, Alex, Wu, Jean, Chuang, Jason, Manning, Chris, Ng, Andrew, and Potts, Chris. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*, 2013.

Tackstrom, Oscar and McDonald, Ryan. Discovering fine-grained sentiment with latent variable structured prediction models. In *Proc. of ECIR*, 2011.

Thomas, Matt, Pang, Bo, and Lee, Lilian. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proc. of EMNLP*, 2006.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society B*, 58(1):267–288, 1996.

Wu, Tong Tong, Chen, Yi Fang, Hastie, Trevor, Sobel, Eric, and Lange, Kenneth. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

Yano, Tae, Smith, Noah A., and Wilkerson, John D. Textual predictors of bill survival in congressional committees. In *Proc. of NAACL*, 2012.

Yessenalina, Ainur, Yue, Yisong, and Cardie, Claire. Multi-level structured models for document sentiment classification. In *Proc. of EMNLP*, 2010.

Yogatama, Dani, Heilman, Michael, O'Connor, Brendan, Dyer, Chris, Routledge, Bryan R., and Smith, Noah A. Predicting a scientific communitys response to an article. In *Proc. of EMNLP*, 2011.

Yuan, Lei, Liu, Jun, and Ye, Jieping. Efficient methods for overlapping group lasso. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2104–2116, 2013.

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

Zaidan, Omar F. and Eisner, Jason. Modeling annotators: a generative approach to learning from annotator rationales. In *Proc. of EMNLP*, 2008.

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