Introduction

Text is hard:
- High dimensionality of text → overfitting remains
- Not all words are useful → sparsity

Regularization:
- Critical for text classification, opinion mining, noisy text normalization
- Group lasso can fail to create sparse models
- Groups are not always available

Contribution:
- Apply OMP to text classification;
- Introduce overlapping GOMP moving from disjoint to overlapping groups;
- Analyze their efficiency in accuracy and sparsity (vs. group lasso & deep learning).

II. Orthogonal Matching Pursuit

Algorithm: Logistic Matching Pursuit

Input: \( x \in \mathbb{R}^d \), \( y \in \{-1,1\}^m \)

\( \{G_1, \ldots, G_L\} \) (groups), \( K \) (budget), \( \{\lambda_i\} \) (precision), \( \eta \)

Initialize: \( I = \emptyset \), \( \rho_{\theta} = y_i - 1 \)

1. While \( |I| < K \) do:

   2. \( \rho_{\theta} = \text{arg max} \{ \frac{\langle x, y_i \rangle}{\|x_{G_i}\|^2} \} \}

   3. Break if \( \rho_{\theta} \leq \eta \)

   4. Add \( i \) to \( I \)

   5. For \( i = 1 \) to \( L \)

   6. \( G_l = G_l \setminus \{i\} \)

   7. End for

   8. \( \theta = \text{arg min} \sum_{i \in I} \|x_{G_i} - y_i\|^2 \}

   9. Update active set

   10. \( k = 1 \) while

   11. End while

II. Structured Regularization

Where?
- Removing unnecessary words along with their weights
- Text normalization → machine learning problem (Ikeda, Shindo, and Matsumoto 2016)

Methods
- \( \ell_1, \ell_2 \): Elastic net regularization
- Group lasso (Yuan and Lin 2006)
- Linguistic structured regularization (Yogatama and Smith 2014)

III. Datasets & Setup

DATA

- Topic categorization on 20NG dataset
- Four binary classification tasks

Sensitivity analysis
- Floor speeches by U.S. Congressmen deciding “yea”/“nay” votes on the bill under discussion (Thomas, Pang, and Lee 2006)
- Movie reviews (Pang and Lee 2004)
- Product reviews from Amazon (Blitzer, Dredze, and Pereira 2007)

Settings
- Parameter tuning on development set
- Minibatch K-Means clustering on word2vec with max 2000 clusters.

IV. Results

Table: Accuracy in test subsets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>no reg</th>
<th>lasso</th>
<th>ridge</th>
<th>elastic</th>
<th>OMP</th>
<th>GOMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>20NG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>science</td>
<td>0.946</td>
<td>0.916</td>
<td>0.954</td>
<td>0.954</td>
<td>0.964</td>
<td>0.968</td>
</tr>
<tr>
<td>sports</td>
<td>0.908</td>
<td>0.907</td>
<td>0.925</td>
<td>0.920</td>
<td>0.949</td>
<td>0.959</td>
</tr>
<tr>
<td>religion</td>
<td>0.894</td>
<td>0.876</td>
<td>0.895</td>
<td>0.890</td>
<td>0.902</td>
<td>0.910</td>
</tr>
<tr>
<td>computer</td>
<td>0.846</td>
<td>0.843</td>
<td>0.866</td>
<td>0.856</td>
<td>0.876</td>
<td>0.891</td>
</tr>
<tr>
<td>movie</td>
<td>0.666</td>
<td>0.640</td>
<td>0.616</td>
<td>0.622</td>
<td>0.684</td>
<td>0.655</td>
</tr>
<tr>
<td>Sentiment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vote</td>
<td>0.865</td>
<td>0.860</td>
<td>0.870</td>
<td>0.875</td>
<td>0.860</td>
<td>0.890</td>
</tr>
<tr>
<td>movie</td>
<td>0.765</td>
<td>0.737</td>
<td>0.770</td>
<td>0.770</td>
<td>0.785</td>
<td>0.800</td>
</tr>
<tr>
<td>books</td>
<td>0.750</td>
<td>0.770</td>
<td>0.780</td>
<td>0.780</td>
<td>0.800</td>
<td>0.805</td>
</tr>
<tr>
<td>kitch</td>
<td>0.760</td>
<td>0.800</td>
<td>0.800</td>
<td>0.825</td>
<td>0.825</td>
<td>0.845</td>
</tr>
</tbody>
</table>

Table: Accuracy vs. number of active atoms/features for OMP.

V. Discussion & Future Work

- Group based regularizers better than the baseline ones.
- GOMP requires some “good” groups along with single features.

Conclusion

- Introduce OMP and GOMP for the text classification task
- Extending the standard GOMP algorithm was also proposed, which is able to handle overlapping groups
- Simple (greedy feedforward feature selection) → accurate models with high sparsity

Future work

- Examine the theoretical properties of overlapping GOMP
- Learning automatically the groups → Simultaneous OMP (Szlama, Gregory, and LeCun 2012)
- Sparse Group OMP