Regularizing Text Classification with Topics
Konstantinos Skianis, Michalis Vazirgiannis
LIX, École Polytechnique, France

Motivation

Text categorization is hard:
- high dimensionality
- prone to overfitting
- state-of-the-art structured regularization is slow due to overlapping clusters

Regularization is necessary:
- Critical for language modeling, structured prediction, and classification
- Prior on the feature weights
- Find the optimal weights: \( \theta = \arg\min_\theta \sum_i L(x_i, \theta y_i) + \lambda \Omega(\theta) \)

I. Structured Regularization

Group lasso: \( \Omega(\theta) = \lambda \sum_c | \theta_c | \) where \( \lambda > 0 \)

Objective: \( \Omega_{\text{full}}(\theta) + \Omega_{\text{full}}(v) + L(\theta) + u' (v - M\theta) + \frac{1}{2}\lambda |v-M\theta|^2 \)

Iterative update of \( \theta \) and \( v \):
\[
\begin{align*}
\min_{\theta} & \Omega_{\text{full}}(\theta) + L(\theta) + u' (v - M\theta) + \frac{1}{2}\lambda |v-M\theta|^2 \\
& \min_{v} \Omega_{\text{full}}(v) + u' v + \frac{1}{2}\lambda |v|^2
\end{align*}
\]

\( u = u + \lambda v (v - M\theta) \)

Algorithm ADMM

Input: augmented Lagrangian variable \( \rho \), \( \lambda \) and \( \lambda_k \)
1: \( \text{while update in weights not small do} \)
2: \( \theta = \arg\min \Omega_{\text{full}}(\theta) + L(\theta) + \frac{1}{2}\sum_i (\theta_i - \theta_i^{\text{old}})^2 \)
3: \( \text{for } g = 1 \rightarrow G \text{ do} \)
4: \( v = v + \rho_{x}(x^g - x^{g-1}) \)
5: \( \text{end for} \)
6: \( u = u + \lambda v (v - M\theta) \)
7: \( \text{end while} \)

STATISTICAL REGULARIZERS
- Network of features: \( \Omega_{\text{full}}(\theta) = \lambda \sum_c | \theta_c | M_{ij} \), where \( M = \alpha (I - P)^T (I - P) + \beta I \).
- Sentence Regularizer: \( \Omega_{\text{full}}(\theta) = \sum_{c=1}^C \sum_{i=1}^{|S_c|} | \theta_{c,i} | \)  

SEMANTIC REGULARIZERS:
- LDA regularizer
- LSI regularizer

GRAPHICAL REGULARIZERS
- Graph-of-words regularizer: Community detection on document collection graph
- Word2vec regularizer: Kmeans clustering on word2vec
- \( \Omega_{\text{full}}(\theta) = \sum_{c=1}^C | \theta_{c,i} | 2 \)

Figure: A Graph-of-words example.

II. Structured Regularization in NLP

III. Results

Table: Accuracy

<table>
<thead>
<tr>
<th>dataset</th>
<th>no reg.</th>
<th>lasso ridge elastic</th>
<th>group lasso</th>
<th>GoW word2vec</th>
</tr>
</thead>
<tbody>
<tr>
<td>science</td>
<td>0.946</td>
<td>0.916</td>
<td>0.954</td>
<td>0.960</td>
</tr>
<tr>
<td>sports</td>
<td>0.908</td>
<td>0.907</td>
<td>0.925</td>
<td>0.909</td>
</tr>
<tr>
<td>religion</td>
<td>0.894</td>
<td>0.876</td>
<td>0.895</td>
<td>0.918</td>
</tr>
<tr>
<td>computer</td>
<td>0.846</td>
<td>0.843</td>
<td>0.869</td>
<td>0.918</td>
</tr>
<tr>
<td>movie</td>
<td>0.860</td>
<td>0.864</td>
<td>0.866</td>
<td>0.918</td>
</tr>
<tr>
<td>books</td>
<td>0.750</td>
<td>0.770</td>
<td>0.780</td>
<td>0.790</td>
</tr>
<tr>
<td>dvd</td>
<td>0.765</td>
<td>0.775</td>
<td>0.790</td>
<td>0.800</td>
</tr>
<tr>
<td>elect.</td>
<td>0.790</td>
<td>0.800</td>
<td>0.825</td>
<td>0.805</td>
</tr>
<tr>
<td>kitch.</td>
<td>0.760</td>
<td>0.775</td>
<td>0.800</td>
<td>0.855</td>
</tr>
</tbody>
</table>

Table: Model size (— sparsity)

<table>
<thead>
<tr>
<th>dataset</th>
<th>no reg.</th>
<th>lasso ridge elastic</th>
<th>group lasso</th>
<th>GoW word2vec</th>
</tr>
</thead>
<tbody>
<tr>
<td>science</td>
<td>0.946</td>
<td>0.916</td>
<td>0.954</td>
<td>0.960</td>
</tr>
<tr>
<td>sports</td>
<td>0.908</td>
<td>0.907</td>
<td>0.925</td>
<td>0.909</td>
</tr>
<tr>
<td>religion</td>
<td>0.894</td>
<td>0.876</td>
<td>0.895</td>
<td>0.918</td>
</tr>
<tr>
<td>computer</td>
<td>0.846</td>
<td>0.843</td>
<td>0.869</td>
<td>0.918</td>
</tr>
<tr>
<td>movie</td>
<td>0.860</td>
<td>0.864</td>
<td>0.866</td>
<td>0.918</td>
</tr>
<tr>
<td>books</td>
<td>0.750</td>
<td>0.770</td>
<td>0.780</td>
<td>0.790</td>
</tr>
<tr>
<td>dvd</td>
<td>0.765</td>
<td>0.775</td>
<td>0.790</td>
<td>0.800</td>
</tr>
<tr>
<td>elect.</td>
<td>0.790</td>
<td>0.800</td>
<td>0.825</td>
<td>0.805</td>
</tr>
<tr>
<td>kitch.</td>
<td>0.760</td>
<td>0.775</td>
<td>0.800</td>
<td>0.855</td>
</tr>
</tbody>
</table>

IV. Regularizing Deep Learning

Sentence Classification with CNNs (Kim et al., EMNLP 2014)

Neuron Deletion (Morcos et al., ICLR 2018)

Grouped Weight Sharing (Zhang et al., ACL 2017)

V. Discussion & Future Work

- Superior proposed regularizers: more effective, more efficient and sparser
- GoW-based regularization although very fast, did not outperform the other methods
  - Overlapping community detection algorithms failed to identify “good” groups
  - Incorporating prior knowledge into neural models
  - Interpretable neurons are no more important
  - Networks which generalise better are harder to break

Conclusion

- Find and extract semantic and syntactic structures that lead to sparser feature spaces — faster learning times
- Linguistic prior knowledge in the data can be used to improve categorization performance for baseline bag-of-words models, by mining inherent structures
- No significant change in results with different loss functions as the proposed regularizers are not log loss specific

Future work

- Find better clusters in word2vec (+overlapping with GMM)
- Explore alternative regularization algorithms diverging from group-lasso
- Interpret neurons in a deeper level
- Different tasks like qa, chatbots etc.
- Study sparsity along with groups
- Learning the topics simultaneously

Table: Number of groups.

Table: Time (in seconds) for learning with best hyperparameters.