News Article Classification

(Multi-Label Learning: ML-KNN & BP-MLL)

Lauren Contard, Archit Datar, Yue Li, Bobby Lumpkin, Haihang Wu

The Ohio State University

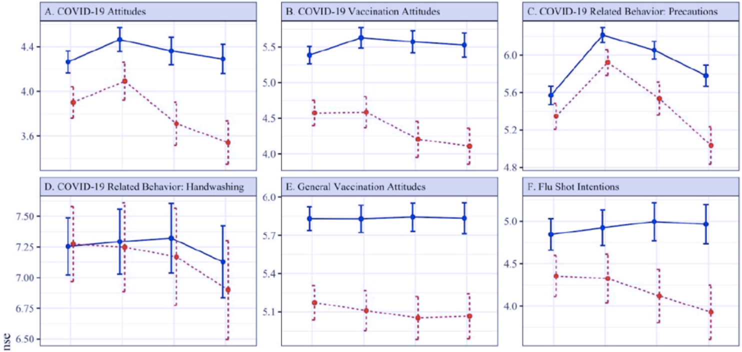
STAT 6500

Overview

Introduction and Problem Statement

Introduction

- People in the U.S. have shown polarized attitudes and behaviors in response to COVID-19.
- What could be some of the drivers? We aim to explore whether and how mainstream news media in the U.S. play a role in the polarization of the public's attitudes and behaviors.



Description of Datasets

- We analyzed news content about elite communication from 10 news media, including newspapers (e.g., New York Times) and cable network news (e.g., ABC news).
- We got the list of news articles from the GDELT database and used the Python package "Scrapy" to get the full text.
- We have 8588 news articles in total and randomly selected 290 paragraphs of them to label.
- We manually labeled our paragraphs into 15 categories. One paragraph usually has multiple labels.

Two Examples from the Dataset

"Meanwhile, Bottoms has repeatedly urged residents to stay home. On Friday, she tweeted coronavirus fatality statistics for the state: 'The numbers speak for themselves. PLEASE STAY HOME.'" (From USAtoday, 04/25/2020)

- Threats/impacts
- Responses/actions
- Severity
- Self-efficacy
- Public Health
- Negative

"You know, I wonder, I think there's been a lot of self-congratulations every day that we see in those briefings, frankly, about the testing in the United States, and we're doing so well, we're doing now more than South Korea did." (From abcnews, 03/29/2020)

- Responses/actions
- External-efficacy
- Public health
- Political evaluation
- Negative

Problem Statement

- We have a multi-label classification problem. We adapted two learning algorithms: k-NN and Neural Networks to handle multi-label data directly.

  RQ1: Which learning algorithm performs better in the multi-label classification problem: k-NN or neural networks??

- Our data has high dimensions.

  RQ2: How can we effectively implement linear and non-linear dimensions reduction in the multi-label classification problem?

- The sequence of features need to be considered in the classification.

  RQ3: Which type of the features perform better in the multi-label classification problem: with or without considering the order of words?

KNN Based Approaches

Binary Relevance (A Naive Approach)

- An intuitive approach to deal with the multilabel paradigm.

Binary Relevance (A Naive Approach)

- An intuitive approach to deal with the multilabel paradigm.
- Works by decomposing the multi-label learning task into a number of independent binary learning tasks (one per class label).

Binary Relevance (A Naive Approach)

- An intuitive approach to deal with the multilabel paradigm.
- Works by decomposing the multi-label learning task into a number of independent binary learning tasks (one per class label).

$L = \{l_1, l_2, l_3, l_4\}$

EX#	Label set
1	$\{l_1, l_2\}$
2	$\{l_2, l_4\}$
3	$\{l_1, l_3, l_4\}$
4	$\{l_4\}$

EX#	l_1
1	**true**
2	false
3	**true**
4	false

EX#	l_2
1	**true**
2	**true**
3	false
4	false

EX#	l_3
1	false
2	false
3	**true**
4	false

EX#	l_4
1	false
2	**true**
3	**true**
4	**true**

Binary Relevance (A Naive Approach)

- An intuitive approach to deal with the multilabel paradigm.
- Works by decomposing the multi-label learning task into a number of independent binary learning tasks (one per class label).

$L = \{l_1, l_2, l_3, l_4\}$

EX#	Label set
1	$\{l_1, l_2\}$
2	$\{l_2, l_4\}$
3	$\{l_1, l_3, l_4\}$
4	$\{l_4\}$

EX#	l_1
1	**true**
2	false
3	**true**
4	false

EX#	l_2
1	**true**
2	**true**
3	false
4	false

EX#	l_3
1	false
2	false
3	**true**
4	false

EX#	l_4
1	false
2	**true**
3	**true**
4	**true**

$\rightarrow$ Often criticized in the literature because of its label independence assumption.

Binary Relevance (A Naive Approach)

- An intuitive approach to deal with the multilabel paradigm.
- Works by decomposing the multi-label learning task into a number of independent binary learning tasks (one per class label).

$L = \{l_1, l_2, l_3, l_4\}$

EX#	Label set
1	$\{l_1, l_2\}$
2	$\{l_2, l_4\}$
3	$\{l_1, l_3, l_4\}$
4	$\{l_4\}$

EX#	l_1
1	**true**
2	false
3	**true**
4	false

EX#	l_2
1	**true**
2	**true**
3	false
4	false

EX#	l_3
1	false
2	false
3	**true**
4	false

EX#	l_4
1	false
2	**true**
3	**true**
4	**true**

→ Often criticized in the literature because of its label independence assumption.

→ We implement a KNN based binary relevance model and compare with a more novel adaptation: the ML-KNN model.

ML-KNN Algorithm: Overall Approach

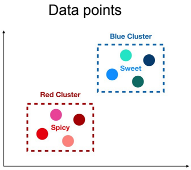
Overall Approach: This ML-KNN algorithm takes a parametric, Bayesian approach towards estimating the Bayes Optimal Classifier. As with the single-label algorithm, it does this using the K nearest neighbors of an instance. Namely...

- Given a test instance, t, $\vec{Y}_t$ is determined using the MAP estimate:

$$\begin{aligned}
\vec{y}_t(\ell) &= \underset{b\in\{0,1\}}{\operatorname{argmax}}\, \mathbb{P}\left(\mathrm{H}_b^\ell | E^\ell_{\vec{C}_t(\ell)}\right), \quad \ell \in \mathcal{Y} \\
&= \underset{b\in\{0,1\}}{\operatorname{argmax}} \frac{\mathbb{P}\left(\mathrm{H}_b^\ell\right) \cdot \mathbb{P}\left(E^\ell_{\vec{C}_{t(\ell)}} | \mathrm{H}_b^\ell\right)}{\mathbb{P}\left(E^\ell_{\vec{C}_t(\ell)}\right)} \\
&= \underset{b\in\{0,1\}}{\operatorname{argmax}}\, \mathbb{P}\left(\mathrm{H}_b^\ell\right) \cdot \mathbb{P}\left(E^\ell_{\vec{C}_{t(\ell)}} | \mathrm{H}_b^\ell\right)
\end{aligned}$$

- Where we take a Bayesian approach towards estimating the prior probabilities, $\mathbb{P}\left(\mathrm{H}_b^\ell\right)$, and conditional probabilities, $\mathbb{P}\left(E^\ell_{\vec{C}_{t(\ell)}} | \mathrm{H}_b^\ell\right)$.

ML-KNN Algorithm: Reasons for dimension reduction



Low dimensional space

	Reddish	Bluish
	1	0
	1	0
	1	0
	1	0
	0	1
	0	1
	0	1
	0	1

High dimensional space

	Red	Maroon	Pink	Flamingo	Blue	Turquoise	Seaweed	Ocean
	1	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0
	0	0	1	0	0	0	0	0
	0	0	0	1	0	0	0	0
	0	0	0	0	1	0	0	0
	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	1

Having a high dimensional feature space causes Euclidian distances between points to be fairly similar as the distance vector components are partitioned across many dimensions.

Nonlinear Dimension Reduction (ANN Autoencoder)

Nonlinear Dimension Reduction (ANN Autoencoder)

- PCA allows only for reductions in dimension due to linear representations.

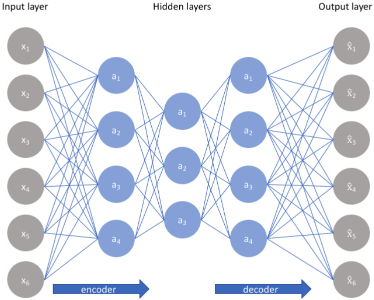
Nonlinear Dimension Reduction (ANN Autoencoder)

- PCA allows only for reductions in dimension due to linear representations.
- Autoencoders can learn data projections with suitable dimensionality and sparsity limitations that allow for nonlinear representations, as well.

Nonlinear Dimension Reduction (ANN Autoencoder)

- PCA allows only for reductions in dimension due to linear representations.
- Autoencoders can learn data projections with suitable dimensionality and sparsity limitations that allow for nonlinear representations, as well.

→ Train a feed forward neural network with a "bottleneck layer" to perform the identity mapping.

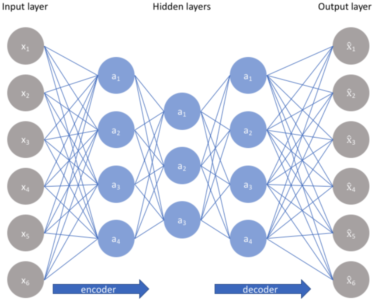


Nonlinear Dimension Reduction (ANN Autoencoder)

- PCA allows only for reductions in dimension due to linear representations.
- Autoencoders can learn data projections with suitable dimensionality and sparsity limitations that allow for nonlinear representations, as well.

→ Train a feed forward neural network with a "bottleneck layer" to perform the identity mapping.

→ Use bottleneck layer values as encodings of the data.



Threshold Function Learning

Perfect classification, using a constant threshold requires two conditions:

1. Predicted logit values for "true" labels be separated from predicted logit values for "false" labels.
2. This separation be around some constant value (usually either 0.5 or 0)

Learning a threshold function aims to relax the second condition.

$\rightarrow$ Namely, we fit a linear regression model to learn threshold values from the logit outputs of our models.

Results: Dimension Reduction

- Initial data contains 2094 features
- Using Principal Component Analysis, we were able to preserve 90 percent of variance using 165 principal components.
- We eliminated another 103 of these using the maximum absolute difference in means between positive and negative cases across all labels
- Leaving 62 principal components
- We also performed non-linear dimension reduction to reduce our data to 62 features using ANN auto encoding

Results: Binary Relevance KNN

- We first performed naive classification to compare with ML-KNN.
- We performed cross-validation on the number of neighbors k, and applied threshold learning.
- Below are the test Hamming loss values for data from both dimension reduction methods.

.

Method	PCA	Auto-Encoder
Initial train	0.139	0.128
Initial test	0.207	0.175
Cross-validation on *k*	0.204	0.181
Cross-validated parameter with original threshold	0.202	0.182
Threshold function learning	0.268	0.253

- The auto-encoder dimension reduction led to lower Hamming loss than PCA.
- Threshold function learning did not lead to better results.

Results: ML-KNN

- We next performed ML-KNN, using the same steps, again on both data sets.

Method	PCA	Auto-Encoder
Initial train	0.101	0.102
Initial test	0.196	0.182
Cross-validation on k & s	0.186	0.177
Cross-validated parameters with original threshold	0.196	0.181
Threshold function learning	0.223	0.210

- Again, classification was more successful for the data reduced by auto encoding, and threshold learning was not helpful

Comparison of KNN Methods

- We consistently achieved better classification with data reduced by the auto-encoding method, compared to PCA
- ML-KNN slightly outperformed binary relevance
- Best binary result: 0.181 Hamming loss
- Best ML-KNN result: 0.177 Hamming loss

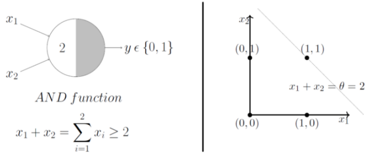
Neural Network Based Approaches

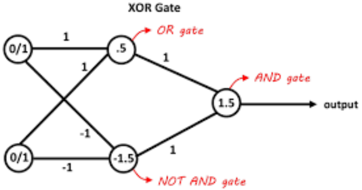
A Brief History

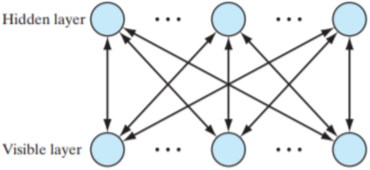
A Brief History

- Inspired by biological nervous systems, neural networks date back to the first half of the 20th century with works such as those by McCulloch and Pitts, which could model simple logical operations.

AND Function

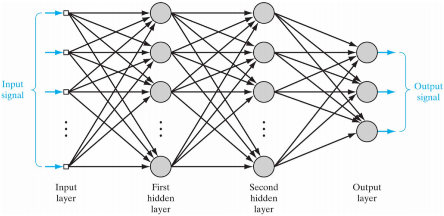






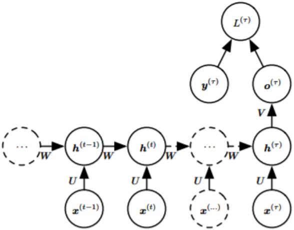
Feed Forward Networks

- Neurons in the first layer represent components of the input vectors.
- The output of the neuron in the next layer is determined by applying a non-linear "activation function" to a linear combination of the input components, plus a bias.



Recurrent Neural Networks (RNNs)

- RNNs are a popular adaptation for NLP problems.
- They utilize hidden unit connections with shared weights.
- Unfolding an RNN let's us visualize it like a feed forward network (see below).



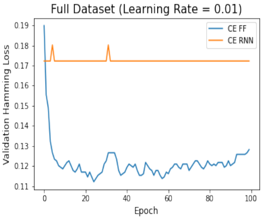
Naive vs Novel Approaches (Cross Entropy vs BPMLL)

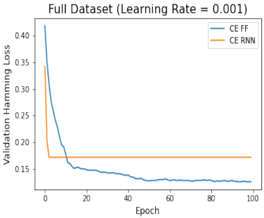
- By "naive" we refer to multilabel networks that utilize a cross entropy loss for training.
- By comparison, Zhang and Zhou bpmll proposed a novel loss, that emphasizes pairwise ranking accuracy.

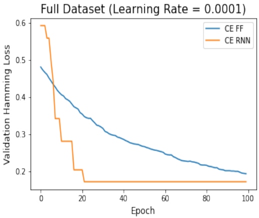
$$E = \sum_{i=1}^{m} E_i = \sum_{i=1}^{m} \frac{1}{|Y_i||\overline{Y}_i|} \sum_{(k,l) \in Y_i \times \overline{Y}_i} \exp(-(c_k^i - c_l^i))$$

so that the i^{th} error term is severely penalized if c_k^i is much smaller than c_l^i.

Artificial Neural Network Results: Full Dataset





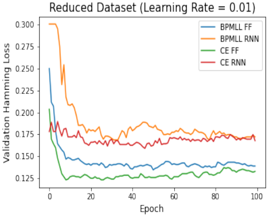


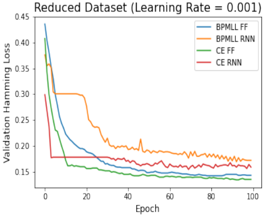
Learning Rate	CE FF	CE RNN
0.01	0.3020833333	0.2379807692
0.001	0.17868589740	0.1730769231
0.0001	0.2115384615	0.1770833333

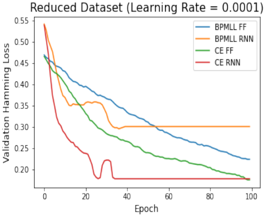
Table 1: Hamming Loss with Threshold Function Learning

CE FF outperform CE RNN in constant threshold but underperform in learned threshold; The effect of learning rate.

Artificial Neural Network Results: Reduced Dataset







Learning Rate	CE FF	BPMLL FF	CE RNN	BPMLL RNN
0.01	0.1520979021	0.145979021	0.2447552448	0.2194055944
0.001	0.1853146853	0.2578671329	0.1791958042	0.20454545
0.0001	0.2071678322	0.1844405594	0.1896853147	0.2132867133

Table 2: Hamming Loss with Threshold Function Learning

Same conclusion for RNN and FF; BPMLL shows NO better performance in hamming loss than cross entropy.

Discussion and Conclusions

Discussion and Conclusions: ANN Approaches

Discussion and Conclusions: ANN Approaches

1. The FF networks networks tend to outperform the RNNs.

Discussion and Conclusions: ANN Approaches

1. The FF networks networks tend to outperform the RNNs.
   $\rightarrow$ More parameters lead to overfitting in the bidirectional LSTM.

Discussion and Conclusions: ANN Approaches

1. The FF networks networks tend to outperform the RNNs.
   - $\rightarrow$ More parameters lead to overfitting in the bidirectional LSTM.
   - $\rightarrow$ Single directinal LSTM can't escape locally optimal solution.

Discussion and Conclusions: ANN Approaches

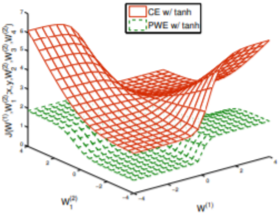
1. The FF networks networks tend to outperform the RNNs.
   - → More parameters lead to overfitting in the bidirectional LSTM.
   - → Single directinal LSTM can't escape locally optimal solution.
   - → Training on more data may improve.

Discussion and Conclusions: ANN Approaches

1. The FF networks networks tend to outperform the RNNs.
   - $\rightarrow$ More parameters lead to overfitting in the bidirectional LSTM.
   - $\rightarrow$ Single directinal LSTM can't escape locally optimal solution.
   - $\rightarrow$ Training on more data may improve.
2. The cross entropy networks tend to outperform the BPMLL networks.

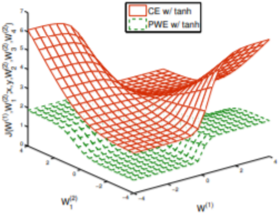
Discussion and Conclusions: ANN Approaches

1. The FF networks networks tend to outperform the RNNs.
   - → More parameters lead to overfitting in the bidirectional LSTM.
   - → Single directinal LSTM can't escape locally optimal solution.
   - → Training on more data may improve.
2. The cross entropy networks tend to outperform the BPMLL networks.
   - → The surface for the BPMLL loss with tanh activations has plateaus in which gradient descent can be very slow in comparison with the cross-entropy (Nam et al. [2014]). The case may be similar for ReLU activations.



Discussion and Conclusions: ANN Approaches

1. The FF networks networks tend to outperform the RNNs.
   - → More parameters lead to overfitting in the bidirectional LSTM.
   - → Single directinal LSTM can't escape locally optimal solution.
   - → Training on more data may improve.
2. The cross entropy networks tend to outperform the BPMLL networks.
   - → The surface for the BPMLL loss with tanh activations has plateaus in which gradient descent can be very slow in comparison with the cross-entropy (Nam et al. [2014]). The case may be similar for ReLU activations.



   - → While BPMLL is supposed to leverage correlations between labels, Nam et al. [2014] conjecture that these correlations also may cause overfitting.

Discussion and Conclusions: ANN Approaches (continued)

3. Models saw no improvement from using a learned threshold function.

Discussion and Conclusions: ANN Approaches (continued)

3. Models saw no improvement from using a learned threshold function.
   $\rightarrow$ Models could sufficiently separate predicted logits about 0.5.

Discussion and Conclusions: ANN Approaches (continued)

3. Models saw no improvement from using a learned threshold function.
   - → Models could sufficiently separate predicted logits about 0.5.
   - → This approach may be more useful for models that require extensive resources for sufficient training (improved less extensively trained models).

Discussion and Conclusions: ANN Approaches (continued)

3. Models saw no improvement from using a learned threshold function.
   - → Models could sufficiently separate predicted logits about 0.5.
   - → This approach may be more useful for models that require extensive resources for sufficient training (improved less extensively trained models).
4. The Feed Forward Networks were our best performing models when evaluated on our validation set, outperforming both the RNN models and the KNN based approaches.

References

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification — revisiting neural networks. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 437–452, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007. doi: 10.1016/j.patcog.2006.12.019.

Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006. doi: doi:10.1109/TKDE.2006.162.