Multi-label Text Classification: Brief Theory, Selected Algorithms, and Proposed Experiments

Nghia Duong-Trung

03.03.2025

1 Introduction

Multi-label classification is a type of classification where each instance can be associated with more than one label simultaneously. This is common in fields such as:

- News categorization (one article may belong to "Education" and "Economy").
- Medical diagnosis (one patient may have multiple conditions).
- Product tagging (one product may fit several categories).

2 Theoretical Foundation

2.1 Problem Definition

In standard single-label classification, each instance has exactly one label. In multi-label classification, each instance has a set of labels.

Given a dataset with n instances and L possible labels, each instance x_i is associated with:

$$\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iL}] \in \{0, 1\}^L$$

where $y_{ij} = 1$ indicates the j-th label applies to instance i, and $y_{ij} = 0$ otherwise.

The dataset is:

$$D = \{(x_i, \mathbf{y}_i)\}_{i=1}^n$$

2.2 Example Data Representation

```
Document: "The government passed a new economic policy."
Labels: [Politics, Economy]
```

This corresponds to:

$$\mathbf{y} = [1, 1, 0, \ldots, 0]$$

(where 1 represents assigned labels and 0 represents unassigned labels).

2.3 Loss Function: Binary Cross-Entropy

Multi-label classification often uses Binary Cross-Entropy (BCE) loss, applied independently for each label. For instance i and label j, the loss is:

$$\ell_{ij} = -\left(y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log(1 - \hat{y}_{ij})\right)$$

The per-instance loss sums across all labels:

$$\ell_i = \frac{1}{L} \sum_{j=1}^{L} \ell_{ij}$$

The overall loss for the dataset is:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \ell_i$$

This treats each label independently, which is suitable when label combinations do not follow strict rules.

2.4 Intuitive Explanation of BCE Loss

The binary cross-entropy loss encourages the predicted probability $\hat{y}_{ij}$ to be close to 1 for labels that apply and close to 0 for labels that do not apply. Each label contributes independently to the total loss, which simplifies optimization when the number of labels is large.

3 Algorithms for Multi-label Classification

3.1 Problem Transformation Methods

These methods convert the multi-label task into simpler problems.

3.1.1 Binary Relevance (BR)

- Train one binary classifier for each label.
- Predict each label independently.

For label j, train:

$$f_j : x \to y_j$$

3.1.2 Classifier Chains (CC)

- Train L classifiers sequentially.
- Each classifier considers the predictions of previous classifiers.

For label j:

$$f_j : (x, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{j-1}) \to y_j$$

This models label dependencies.

3.1.3 Label Powerset (LP)

- Treats each unique combination of labels as a single class.
- Converts the multi-label task into a multi-class problem.

3.2 Algorithm Adaptation Methods

3.2.1 Multi-label k-Nearest Neighbors (MLkNN)

- Extends kNN to estimate label probabilities based on neighbors.
- Assigns the most likely labels for each instance.

3.2.2 Multi-output Classifier

- Any base classifier is extended to predict a vector of labels.
- Example: Logistic Regression, Decision Trees.

3.2.3 Neural Networks with Sigmoid Outputs

- Neural network outputs L sigmoid probabilities, one per label.
- Loss: Binary cross-entropy applied to each label independently.

$$\hat{\mathbf{y}} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

3.2.4 Transformer-based Approaches

- Fine-tune pretrained transformers (like BERT) for multi-label tasks.
- Last-layer logits pass through sigmoid activation.

$$\hat{\mathbf{y}} = \sigma(\text{BERT}(x))$$

4 Experimental Setup

4.1 Datasets

The following datasets are commonly used in multi-label text classification experiments:

4.1.1 Reuters-21578

- URL: `https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`
- Description: News articles with multi-label topic categorization.

4.1.2 RCV1-v2

- URL: `https://trec.nist.gov/data/reuters/reuters.html`
- Description: Large-scale dataset with hierarchical topic codes for each news story.

4.2 Preprocessing

Common preprocessing steps:

- Lowercasing.
- Removing punctuation and stopwords.
- Tokenization.
- TF-IDF vectorization or BERT embeddings.

4.3 Models

We evaluate the following models:

- Binary Relevance (Logistic Regression).
- Classifier Chains (Random Forest).
- Multi-label kNN.
- BERT fine-tuned for multi-label classification.

5 Evaluation Metrics

5.1 Hamming Loss

$$\frac{1}{nL}\sum_{i=1}^{n}\sum_{j=1}^{L}\mathbf{1}\{\hat{y}_{ij} \neq y_{ij}\}$$

5.2 Micro-averaged F1-score

$$\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.3 Subset Accuracy

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{\hat{\mathbf{y}}_i = \mathbf{y}_i\}$$

6 Conclusion

This tutorial covered the theory, algorithms, and experimental setup for multi-label text classification, with a focus on both classical methods and modern neural approaches.

References

- https://scikit-learn.org/stable/modules/multiclass.html#multilabel-classification