Extreme Multi-Label Classification under Precision@k

Multi-label classification

\( \mathbf{z} = (z_1, z_2, \ldots, z_d) \in \{0, 1\}^d \) – result of offline prediction.

Marginal probability of a label:

\( \eta_j(x) = P(y_j = 1 | x) = \sum_{i=1}^{d} \mathbf{z}_i P(y|x) \)

Goal: find a classifier \( h(x) : X \rightarrow \mathbb{R}^m \) minimizing the expected loss:

\( L(h) = \mathbb{E}_{x,y} [\eta_j(h(x)) \cdot I(y_j = 1)] \)

The regret of a classifier \( h \) with respect to \( \ell \):

\( \text{reg}_\ell(h) = L(h) - L(h^*) = L(h) - L^* \)

Precision@k \((\mathcal{K})\): \( \mathcal{K} \) is a set of \( k \) labels predicted by \( h \) for \( x \):

\( \text{precision}_\ell(k)(y,x,h) = \frac{1}{k} \sum_j \eta_j(x) \)

Conditional risk for precision@k:

\( L_{\text{reg}}(h | x) = \mathbb{E} \left( \eta_j(y, x, h) \right) = 1 - \frac{1}{k} \sum_{i \in \mathcal{K}} \eta_j(x) \)

The optimal strategy for precision@k: \( k \) labels with the highest \( \eta_j(x) \)

Hierarchical Softmax

HSM [4] is a multi-class classification algorithm based on a label tree.

Each label \( y \) coded by \( z = (z_1, \ldots, z_d) \in \mathbb{C} \)

An internal node identified by a partial code \( z' = (z_1, \ldots, z_i) \)

The code does not have to be binary.

Factorize the marginal probabilities of labels using a chain rule.

\( \eta_j(x) = \prod_{i=1}^{d} \mathbf{P}(z_i | x^{-i}) \cdot x \)

- HSM uses logistic loss and a linear model for estimating \( \mathbf{P}(z_i | x^{-i}) \).
- A multi-class distribution: \( \sum \mathbf{P}(z_i = c | x^{-i}, x) = 1 \).

Multi-label data: Pick-one-label heuristic

- Tools like FastText [1] or Learned Trees [6], apply a pick-one-label heuristic to HSM to transform multi-label instances to multi-class ones.
- Randomly picking a positive label transforms the multi-label distribution to a multi-class distribution:

\( \hat{\eta}_j(x) = \sum_{i=1}^{d} I(y_i = 1) \mathbf{P}(y | x) \)

Inconsistent (non-zero regret) for label-wise logistic loss and precision@k:

\( \eta_j(x) \neq \mathbf{P}(y | x) \)

Given conditionally independent labels, \( \mathbf{P}(y | x) = \prod_{i=1}^{d} \mathbf{P}(y_i | x) \), HSM with pick-one-label heuristic is consistent for the precision@k loss.

Probabilistic Label Trees

PLTs [3] are a no-regret generalization of HSM to multi-label problems.

- Extended code \( z = (1, \ldots, z_d) \).
- Factorization of the marginal probability:

\( \eta_j(x) = \prod_{i=1}^{d} \mathbf{P}(z_i | x^{-i}, x) \)

- Different normalization than in HSM:

\( \sum \mathbf{P}(z_i = c | x^{-i}, x) = 1 \)

- PLTs applied to a multi-class distribution boil down to HSM.

Regret bounds

- Bound for the absolute difference between the true and the estimated marginal probability for label \( j \):

\( |\eta_j(x) - \hat{\eta}_j(x)| \leq \sum_{i=1}^{d} \mathbf{P}(z_i | x^{-i}) \cdot \left( 1 - \sqrt{\eta_j(x)} \right) \cdot \mathbf{P}(x^{-i}) \cdot \mathbf{P}(x) \)

- Bound for the regret with respect to precision@k:

\( \text{reg}_\ell(h | x) = \sum_{j \in \mathcal{K}} (|\eta_j(x) - \hat{\eta}_j(x)|) \leq 2 \max_{j \in \mathcal{K}} |\eta_j(x) - \hat{\eta}_j(x)| \)

Implementation (extremeText)

- Based on fastText.
- Tree structures: random, Huffman tree or build via top-down hierarchical balanced clustering.
- Linear models in the nodes.
- Online training with features embedding (hidden, dense representation).
- L2 regularization for all parameters of the model (for embeddings and nodes’ weights).
- Hidden representation obtained by weighted average of the feature vectors of proportion to the tf-idf scores of features.
- Depth first search prediction.
- Allows for fast online prediction.

Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metrics</th>
<th>fastText</th>
<th>Learned Trees</th>
<th>ExtremeText</th>
<th>EMIL-CNN</th>
<th>Wiki-500K</th>
<th>Amazon-670K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki-500K</td>
<td>N \text{test} = 191,032</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N \text{train} = 101,508</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d = 291,936</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m = 96,059</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P@5</td>
<td>79.17</td>
<td>69.12</td>
<td>57.02</td>
<td>41.02</td>
<td>16.59</td>
<td>30.39</td>
<td></td>
</tr>
<tr>
<td>P@1</td>
<td>72.14</td>
<td>61.12</td>
<td>49.54</td>
<td>36.45</td>
<td>8.61</td>
<td>21.58</td>
<td></td>
</tr>
<tr>
<td>P@5/100</td>
<td>72.14</td>
<td>61.12</td>
<td>49.54</td>
<td>36.45</td>
<td>8.61</td>
<td>21.58</td>
<td></td>
</tr>
</tbody>
</table>

| Amazon-670K | N \text{test} = 400,049 |
| N \text{train} = 358,085 |
| d = 13,000 |
| m = 62,097 |
| P@5 | 75.43 | 65.37 | 55.13 | 44.85 | 32.64 |
| P@1 | 72.47 | 61.37 | 50.13 | 39.64 | 28.64 |
| P@5/100 | 72.47 | 61.37 | 50.13 | 39.64 | 28.64 |

Acknowledgments: Kalina Jasinska is supported by the Polish National Science Center under grant no. 2017/25/N/ST6/00747.