EXTRACTING HIERARCHIES FROM DATA CLUSTERS FOR BETTER CLASSIFICATION

German Sapozhnikov (Saint-Petersburg State Polytechnic University) and Alexander Ulanov (HP Labs Russia)
ABOUT

– Motivation
– Problem
– Hierarchical Multi-Label classification
– State of the art
– Our solution, Algorithm
– Experiment results
– Conclusion and future work
MOTIVATION

– Why hierarchical classification can be better:
  • In flat classification, the number of training examples associated with each label is considerably less than the total number of examples
  • The computational complexity of training a multilabel classifier is strongly affected by the number of labels, each classifier in hierarchy deals with much smaller set of labels as compared to L(full set of labels)

– Sometimes datasets already have hierarchy but we are interested in cases when they don’t

– Large scale classification use cases:
  • WIPO IPC(International Patent Classification)
  • Legal head notes
  • News stories
  • Pascal challenge
PROBLEM

– **Problem**: Increase performance of large scale multilabel classification

– **Idea**: Transformation of a multilabel classification task with a large set of labels into a tree-shaped hierarchy of simpler multilabel classification tasks

– **Previous work**: “Enhancing Accuracy of Multilabel Classification by Extracting Hierarchies”, TIR’2011

– We present enhanced version of the algorithm
HIERARCHICAL MULTI LABEL CLASSIFICATION

- If number of categories for each document in classification task is more than one, we deal with multi-label classification.
- Classes may have hierarchical structure, we may use it for classification.
- On each layer classifier assigns one or more labels to a document.
- Each classifier deals with a small number of labels.

News article: Ron Paul beat Mitt Romney in 10 states!

Multi-label hierarchical classifier workflow

Categories:
- M1: Party
- M2: President elections
- l1: Sport
- l2: Democratic
- l3: Republican
- l4: USA
- l5: France
- l6: Russia

TMW 2012
STATE OF THE ART

- Methods that use predefined hierarchy
  - RCV1, WIPO, Wikipedia, DMOZ

- Two types of hierarchical classification [Sun et al., 01]
  - Big-bang approach
  - Top-down approach (the most popular)

- HOMER method, ECML/PKDD’08 [Tsoumakas et al., 08]
  - Automatically organizes labels into a tree-shaped hierarchy
  - In our algorithm, we use the same concept of hierarchy and meta-labels

- Pattern Recognition’11 [Brucker et al., 11]
  - Guessing the predefined hierarchy

- PASCAL challenge [Wang et al., 11]
  - Pascal challenge, hierarchy without meta-labels
OUR SOLUTION

We propose a predictive algorithm for extracting hierarchies for classification:

• Automatic generation of hierarchies for classification using clustering

• Use criteria that optimizes different measures: precision, recall, or F1
  – Optimize the hierarchy, not classifier technology
ALGORITHM: BUILDING TAXONOMIES LAYER BY LAYER

We use top-down approach
1. Use the given number of labels (classes)
2. Perform different clusterings of labels (varying number of clusters)
3. Predict, which clustering (\textit{partition}) will be the best for classification task (next slide)
4. Make this process recursive for clusters which size is more than 2. Cluster with one label becomes a leaf in the hierarchy. Cluster with two labels will be separated into two clusters in any case
ALGORITHM: PREDICTION FUNCTION (STEP1)

Predict, which partition will be the best for classification, taking into account their performance now and the fact that we will build some number of layers further.

**Now:** Make classification using clusters as meta-labels. As a result we get a performance measure that shows how good partition are.

<table>
<thead>
<tr>
<th>C</th>
<th>$P^T$</th>
<th>$R^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.99</td>
<td>0.95</td>
</tr>
<tr>
<td>3</td>
<td>0.96</td>
<td>0.90</td>
</tr>
<tr>
<td>4</td>
<td>0.92</td>
<td>0.88</td>
</tr>
</tbody>
</table>

Example: our measure is F1
- We want to predict, how good this partition is.
- We know $P, R, TP, FP, FN$ on this layer.
- To get the prediction for this partition we need to take into account that we will build some number of layers further (next slide).
ALGORITHM: PREDICTION FUNCTION (STEP 2)

Predict, the performance of the partition, taking into account it’s current performance and the fact that we will build some number of layers further.

Layer N: we are here

1, l₂, l₃, l₄, l₅, l₆

TP, FP, FN

Classification using clusters as classes

Layer N+1

TP*¹, FP*¹, FN*¹
P*¹, R*¹, F1*¹

Layer N+2

TP*², FP*², FN*², P*², R*², F1*²

Layer N+3

TP*result, FP*result, FN*result, P*result, R*result, F1*result

Layer N: we are here

1, l₂, l₃, l₄, l₅, l₆

C | Pᵀ | Rᵀ
---|---|---
2 | 0.99 | 0.95
3 | 0.96 | 0.90
4 | 0.92 | 0.88
How we compute prediction for one further possible partition of the cluster:

1. **TP**: The number of true positives will decrease with each next layer of the hierarchy.
2. **FN**: After estimating TP, we can easily calculate estimation of FN.
3. **FP**: Some FP documents will be filtered but some new will be added.

<table>
<thead>
<tr>
<th>C</th>
<th>$P_T^1$</th>
<th>$R_T^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.99</td>
<td>0.95</td>
</tr>
<tr>
<td>3</td>
<td>0.96</td>
<td>0.90</td>
</tr>
<tr>
<td>4</td>
<td>0.92</td>
<td>0.88</td>
</tr>
</tbody>
</table>

**Layer N**: we are here

**Layer N+1**

**Layer N+2**

**Layer N+3**

\[
\begin{align*}
TP_{N+2}^* &= TP \cdot R_T^2 = TP \cdot 0.95 \\
TP_{N+3}^* &= TP_{N+2}^* \cdot R_T^2 = TP \cdot 0.95 \cdot 0.95 \\
TP_{N+2}^{*2} &= (2 \cdot TP_{N+3}^* + TP_{N+2}^*) / 3 \approx TP \cdot 0.918 \\
FP_{N+2}^* &= FP \cdot R_T^2 + TP \cdot (1 - P_T^2) = FP \cdot 0.95 + TP \cdot 0.01
\end{align*}
\]
ALGORITHM: PREDICTION FUNCTION

Predict, which partition will be the best for classification, taking into account it’s current performance and the fact that we will build some number of layers further.

Example

1. We know the size of all classes and clusters
2. We know the result on this layer
3. We know prediction for all parts of the partition
   - We made classification and know everything about
   - We calculate prediction of TP, FN, FP for and
4. Using all information we can calculate prediction of TP, FP, FN for the partition
5. We can calculate prediction of the F1 for the partition

Stopping criteria:
If predictions for all clusterings are less than flat result on the current layer.
EXPERIMENTAL RESULTS

- Optimizing micro-F1

- F1 is harmonic mean of precision and recall. It was optimized by optimizing recall at the expense of precision

- We optimized micro-F1 for 5 datasets. For others result remains the same

- Classifier that we used in each node of the hierarchy: decision tree [Webb.99]

- All datasets were taken from http://mulan.sourceforge.net/

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>Ver. 2</th>
<th>Flat</th>
<th>Ver. 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Mediamill</td>
<td>0.53</td>
<td>0.55</td>
<td>0.52</td>
</tr>
<tr>
<td>Bibtex</td>
<td>0.39</td>
<td>0.65</td>
<td>0.28</td>
</tr>
<tr>
<td>Medical</td>
<td>0.83</td>
<td>0.86</td>
<td>0.79</td>
</tr>
<tr>
<td>Enron</td>
<td>0.83</td>
<td>0.55</td>
<td>0.46</td>
</tr>
<tr>
<td>Yeast</td>
<td>0.59</td>
<td>0.59</td>
<td>0.60</td>
</tr>
<tr>
<td>Genbase</td>
<td>0.98</td>
<td>1.00</td>
<td>0.96</td>
</tr>
<tr>
<td>Corel5k</td>
<td>0.09</td>
<td>0.18</td>
<td>0.06</td>
</tr>
<tr>
<td>CAL500</td>
<td>0.40</td>
<td>0.37</td>
<td>0.43</td>
</tr>
<tr>
<td>Scene</td>
<td>0.63</td>
<td>0.66</td>
<td>0.59</td>
</tr>
</tbody>
</table>

TMW 2012
CONCLUSION

– Results:
  • An algorithm for automatically extracting hierarchies of labels for classification
  • Algorithm is classifier independent. It enhances accuracy of multilabel classification optimizing a hierarchy structure, not classifiers
  • Main use case are datasets which don’t have predefined hierarchies
  • Algorithm was verified against 9 datasets

– Future work:
  • Use algorithm for large-scale tasks
  • Explore various classifiers to be used inside the hierarchy
  • Toolkit implementation