An Implementation of Hierarchical Multi-Label Classification System

Thanawut Ananpiriyakul, Piyapan Poomsirivilai, Peerapon Vateekul (Supervisor)

Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand
Thanawut.A@Student.Chula.ac.th, Piyapan.P@Student.Chula.ac.th, Peerapon.V@Chula.ac.th

Abstract

Hierarchical Multi-Label Classification (HMC) is a complex type of classification problem, where each example can be annotated into more than one label, and the labels are hierarchically organized. It has received a lot of attentions due to its need in broad domains of applications. There were many proposed HMC algorithms based on Support Vector Machine (SVM); however, it is surprising that none of them has ever been implemented and opened for the community. In this project, we present open-source HMC software built around our earlier algorithm called "HR-SVM". Moreover, there are many modifications in this version resulting in an improvement in terms of accuracy and induction time. The software was tested and showed promising results not only in the HMC domain, but also in the multi-label domain.

Experimental Result

Table 1. F_1-results on multi-label datasets*

Method / Dataset	SVM-BR (baseline)	R-SVM
emotions	0.240	**0.621 (+158)**
mediamill	0.028	**0.259 (+825)**
rcv1regions	0.125	**0.274 (+119)**
rcv1topics	0.258	**0.583 (+125)**
scene	0.768	**0.795 (+3)**
tmc2007	0.265	**0.618 (+133)**
yeast	0.299	**0.481 (+60)**

Table 2. F_1-results on hierarchical multi-label datasets*

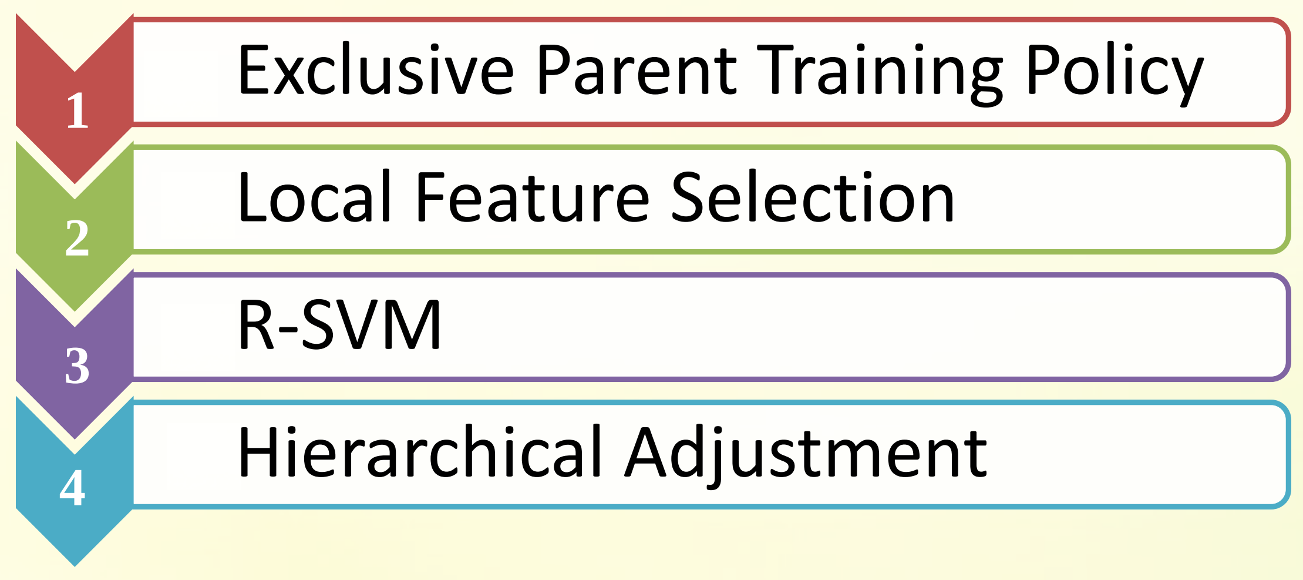
Method / Dataset	SVM-BR (baseline)	Clus-HMC	HR-SVM
D0_yeast_GO	**0.537**	0.398 (-26)	0.518 (-3)
D13_seq_GO	0.554	0.488 (-12)	**0.681 (+22)**
D14_exprindiv_GO	0.068	**0.215 (+216)**	0.145 (+113)
D15_scop_GO	0.799	0.789 (-1)	**0.801 (+0.2)**
D16_struc_GO	**0.592**	0.497 (-16)	0.582 (-1)
D17_interpro_GO	0.456	**0.582 (+27)**	0.415 (-9)

* The number in parentheses is a percentage improvement (%) comparing to baseline.
The boldface method is a winner on that dataset.

In the multi-label domain, the comparison methods are SVM-BR and R-SVM. The winner method is clearly "R-SVM" which can beat SVM-BR on all datasets. In the mediamill dataset, our method outperforms SVM-BR more than 800%!

In HMC domain, there are three competitors: SVM-BR , HR-SVM, and Clus-HMC. The result shows that our method outperforms Clus-HMC on 4 datasets. Comparing to SVM-BR, our system has lost at most 10%, while gained F_1 more than 100% on the D15_scop_GO dataset.

Proposed Framework



1. Exclusive Parent Training Policy (EPT)

For each class node, we propose a strategy to construct binary training dataset from ancestor classes: (1) all examples of 1st level superclass, (2) random 20% of examples of 2nd level superclass, and (3) random 10% of examples of 3rd level superclass.

2. Local Feature Selection (LFS)

The feature selection algorithm is F-Score (Yi-Wei Chen, Chih-Jen Lin). The advantage of F-Score is that it can measure over both continuous and categorical features.

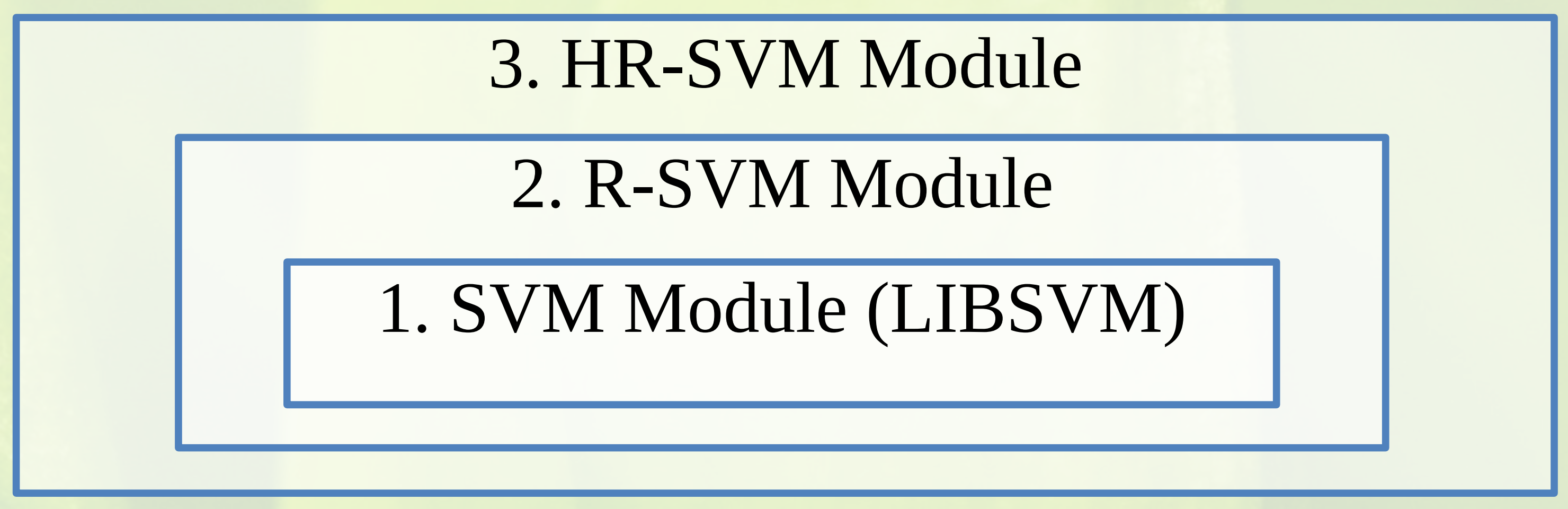
3. R-SVM

The original version of R-SVM is somehow slow since it uses all training examples. We propose to reduce the size of the calibration dataset by sampling only 50% of original. The result showed that this technique can cut time consumption of R-SVM by half without sacrificing accuracy of the original R-SVM algorithm.

4. Hierarchical Adjustment

We used a common strategy called "top-down approach" which classifies a testing example using classifiers from the top level to the bottom level.

System Implementation



There are three program levels: (1) SVM, (2) R-SVM, and (3) HR-SVM. Each of them serves on different types of classifications: (1) binary, (2) multi-label, and (3) hierarchical classification, consecutively. The core SVM engine is based on LIBSVM due to its promising performance.

The system was implemented in C++ programming language which is supported by various platforms including Microsoft Windows, Linux, and OS X. User can create batch command to execute the system easily.

Conclusion

SVM is one of the most popular algorithms to handle HMC task. But, there is no SVM tools for the HMC domain freely available. This project presents the first HMC tool using SVM. The implementation is based on our earlier work called "HR-SVM". There are three updates in this version including (1) more reliable training policy, (2) faster threshold adjustment, and (3) refined feature selection.

The experiments were conducted on both multi-label and HMC domains. The F_1-results showed that our system outperforms the baseline method on both domains. The software of our system is now available at *https://sites.google.com /site/hrsvmproject/*.