International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Enhancing Top-Down Classification Method with Metaclassification for Large-scale Dataset

Ankita Burungale[1],Dinesh Zende[2]

[1] Student ME-II,Department of Computer Engineering, VPCOE, Baramati, SavitribaiPhule Pune University, Baramati, Maharashtra, India

[2]Assistant professor, Department of Information Technology, VPCOE, Baramati, SavitribaiPhule Pune University, Baramati, Maharashtra, India

Abstract: ***Large-scale hierarchical classification has thousands of classes. The most commonly used method for multiclassification is one-versus rest, which is inappropriate due to computational complexity. So, Top down Method is used instead, but it is not perfect because of an error-propagation problem. The error-propagation means the document are wrongly rejected at higher level, it cannot passed down. Metaclassification Method solves error-propagation problem but it has higher complexity. To overcome this problem, enhancing top method is proposed which combines Top down and Metaclassification methods. It uses score of all base classifiers along root to leaf and checks whether predicted label is correct label or not.***

Keywords: Hierarchical text categorization, Metaclassification, Top down method, Meta level learning, Large-scale data

1. Introduction

Text categorization is fundamental task in data mining. The Information on internet is growing day by day, thus it comes to be very difficult to search required information and utilize this large information. The solution of this problem is to classify the information into topic where this topic arranged in hierarchy.

Recently, real-world applications have many thousands of classes. Most popular method for multiclassification is one versus rest. This method does not consider structural relationships among the categories. It trains the base classifier to check whether a document belongs to set of categories then resulting score is compared to the categories score to determine final category for document. In this technique, a single classifier is trained per class to distinguish that class from all other classes. Thus, one versus rest method is inappropriate for real word application.

Very often, Hierarchical classification is handled by a top-down method, firstly it determines whether a document belongs to root node if yes, then it checks whether document belong to nodes at next level. This process will be repeated until the document cannot be further classified into any sub tree or it reached at the leaf categories. This strategy has been adopted by most hierarchical classification methods due to its simplicity. Top-Down Method uses score-cut (S-cut) and rank-cut (R-cut) strategies. Its computation complexity is the logarithm to number of category.

The top down method has an error propagation problem so metaclassification method is used instead of this. It trains the base classifier using different classifier for each category and then predication of base classifier is used for training of metaclassification. This technique has large computational complexity.

Enhancing Top-Down method with metaclassification combines the metaclassification and the top-down method to improve the accuracy of normal top-down method. Top-Down Method has the blocking problem (error propagation). The blocking problem means documents are wrongly rejected by the classifiers at some higher-levels and cannot be passed down to the classifiers at the lower-levels. Proposed system will solves the problem of the error propagation by using the score of all the base classifier. This framework uses the score of all base classifier along the root to leaf into feature vector and then employ the metaclassifier to predict whether the corresponding label node is correct label or not.

There are two types of hierarchical classification tasks in real-world applications. One type is mandatory leaf-node classification, where only the leaf nodes are valid labels. In contrast, the other is non-mandatory leaf node classification where both the internal nodes and the leaf nodes are valid labels.

2. Literature Survey

In this section, top-down method and metaclassification method are discussed.

2.1 Top down Method

Many researchers are working on top down methods to improve classification accuracy. Dumais and Chen investigate Boolean and Multiplicative function for classifying test samples [2]. Boolean function sets threshold at higher level and only match second level category that has confidence score greater than threshold. Multiplicative function allows matching second level category even their confidence score is less than threshold. This technique works only for three level hierarchies.

Sun et al. solve blocking problem of the top-down method by using extended multiplicative, restricted voting and threshold reduction methods [3]. The extended multiplicative method is the extension of multiplicative method. It works on more than three level of hierarchy. It sends the sample down to lower level sub-tree, if products of two-classifier probability greater than predefined threshold. The restricted voting method gives chance to access documents before higher-level classifier rejects them. This method adds another channel to receive document from grandparent classifier. Here, hierarchy is modified, so its computational complexity is very high. Threshold reduction method uses lower threshold. Hence, more documents can be passed at lower level. However, there is possibility of document rejection at higher level.

H. Malik combines the advantages of flat and hierarchical technique [5]. This technique flats the original hierarchy to k^{th} level, prior to training of hierarchical classifiers (where k is a user-defined parameter). Flattening substitutes some categories by their descendant categories. Flatten hierarchy have less level therefore error propagation problem is solved. Here, hierarchy is flattened; therefore, its complexity is highly increased.

Bennett and Nguyen propose a metaclassification method to enhance top-down method named as refined expert [4]. They first form a tree of classifiers through the top down training, and then build metaclassifier, which is trained by predication from lower level nodes and cousin nodes. If probability from cousin node is high then sample belongs to sibling. This method uses top down and bottom up training thus its complexity is higher.

2.2 Metaclassification

Metaclassification is takes the outputs of the base classifiers as inputs to better learn target signals. Metaclassification is used to improve the accuracy of flat multiclass classification [6], [7].

Todorovski and Dzeroskihas developed a meta decision tree [6]. This method combines the predication of all base classifier that are trained using different learning algorithm. . The meta decision tree (MDT) decides which base classifier should be used to classify a test sample.

Kittler et al. has used majority vote rule, which assign category to test instances if category receives the major votes [7]. They trained the dissimilar classifiers either by using different input representations for the information,or using different parameters for the similar type of classifier (e.g. different value of k for KNN classifier; different value of weights for an MLP classifier), or using different classifiers totally. Majority rule will used to combine the output of dissimilar classifier.

3. Problem Definition

Enhancing Top-down Classification Method with Meta-classification for Large Scale Dataset aims to solve error propagation problem of normal top down method by combining top down and meta-classification method. To solve this problem, intended system uses scores of the all base classifier along a root-to-leaf path as the input and checks whether the leaf node is correct label or not. This system will reduce classification complexity using pruning method.

4. Methodology

Enhanced top down method combines top down and meta classification method. We will see top down method in first section and Enhanced top down method **(**Enhanced TD)in second section.

4.1 Top down Method

Dataset consist of hierarchy of classes *H*, which record all parent and child relations.

$H = \{(p, c) | p \text{ is a parent node, } c \text{ is one of its children}\}$ (1)

Where (*p, c)* is called a parent-child relation.Suppose*T, D* and *E* are the training, development and test sets respectively. Applying ScutTD consists of following three steps. RcutTD removes the second step.

In first step, train base-classifiers. One base classifier will be trained for each parent-child relation (*p, c*) of the hierarchy *H*, noted as f_c through the following training set,

$$T_{pc} = \left\{(x,y) \middle| \begin{matrix} x \in T_p, y = +1 \; if \; x \in T_c, \\ y = -1 \text{ otherwise} \end{matrix} \right\} \qquad (2)$$

Where T_p and T_c are the subset of training samples that belong to the parent node p and child node c.

In second step, determine the optimal thresholds for the base classifiers. Micro-F1 is taken as the criterion optimization target, which balances both precision and recall as follows,

$$t_c = \text{argmax } F_1(D_{pc}, f_c, t)$$

$$= \text{argmax } \frac{2P(D_{pc}, f_c, t)\, R(D_{pc}, f_c, t)}{P(D_{pc}, f_c, t) + R(D_{pc}, f_c, t)}$$

$$P(D_{pc}, f_c, t) = \frac{n_r}{|\{x \,|(x,y) \in D_{pc}, f_c(x) \geq t\}|} \qquad (3)$$

$$R(D_{pc}, f_c, t) = \frac{n_r}{|\{x \,|(x,y) \in D_{pc}, y = 1\}|}$$

$$n_r = |\{x \,|(x,y) \in D_{pc}, f_c(x) \geq t, y = 1\}|$$

Where,

- t_c and f_c are the local threshold and base classifier;
- D_{pc} is the local development subset which is similar with the T_{pc} defined by (1);
- *P* and *R* are the precision and recall;
- n_r is the number of correct predicted labels.

In third step, classify the test sample by following S-cut top down algorithm.

4.2 Enhanced Top-down method

To describe the proposed EnhancedTD, we first introduce the definition of meta-samples as follows,

$M(u, l, f*) = (M_x(u_x, l, f*), M_y(u_y, l, f*))$

$M_x(u_x, l, f*) = \{(n_i, f_{ni}(u_x)) | n_i \in p_l\}$ (4)

$M_y(u_y, l, f*) = \begin{cases} +1, l \in u_y \\ -1, l \notin u_y \end{cases}$

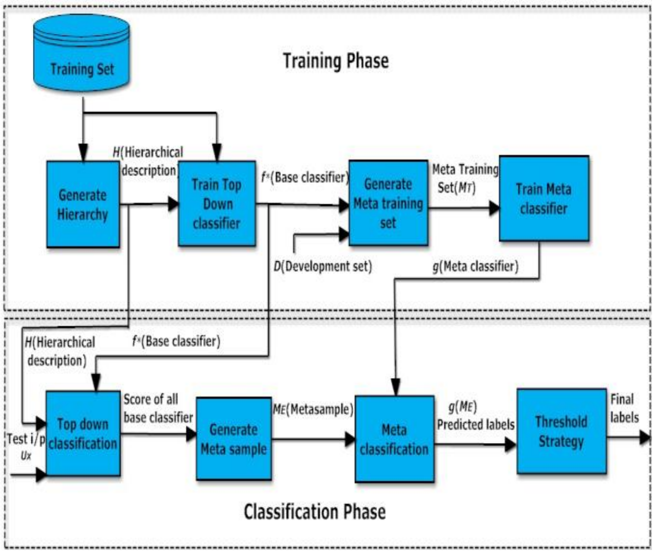


Figure 1: Workflow of Enhanced Top Down Method (EnhancedTD)

Where,

- M is the meta-mapping that consists of meta-input M_x and meta-output M_y;
- H is a hierarchy, $u = (u_x, u_y)$ is a base-sample where u_x is the input part and u_y is the label set;
- l is a leaf node (or a label), that is, a validate label for base-samples;
- $p_l = (n_0, n_1, . . . , n_k)$ is a path from the root to l where n_0 = root, $n_k = l$, $(n_i, n_{i+1}) \in H$, and $f*$ are base-classifiers.

Score cut top-down algorithm

Input : Test sample,
Hierarchical description,
Base-classifiers,
Thresholds (for ScutTD) ,
An integer parameter r (for RcutTD).

Output: set of predicated label y

1. Start from root node. Add root node in queue.
2. **Repeat step no. 3 and 4 until** queue is not empty.
3. Pop out the first item p from queue.
4. **if** p is leaf node then assign it to y.
   else Repeat step no. 4.1 and 4.2 for every childnode of p
   4.1 **ScutTD:-if** score of base classifier at child node is greater than threshold then add that child node in queue.
   4.2 **RcutTD:-if** score of base classifier at child node is ranked top r then add that child node in queue.

However, the above definition yields one meta-sample for each class, which may cause a problem of computational complexity on large-scale tasks. Hence, a method of selecting label candidates for each base-sample is employed so that only a small fraction of labels needs to be delivered into metaclassification. We note this selection method as $L(u_x, f*, H)$. Enhanced TD is based on the above two settings, and its workflow is described in Fig. 1. The training phase consists of three steps as follows,

i. Train base-classifiers $f*$ on a training data set T, which is the same with ScutTD.
ii. Construct a meta-training set with the base classifiers and a development set D, $M_T = \cup_{u \in D} \{M(u,l, f*,H) | l \in L(u_x,f*, H)\}$ (5)
iii. Train a meta-classifier g on meta-sample M_T.

The completely training phase requires the base-level training set T, development set D and the description of the hierarchy H. It produces a set of base-classifiers $f*$ and a meta-classifier g. The classifying phase also consists of three steps as follows,

i. Construct a group of meta-samples from a test base-sample u_x (its label u_y is unknown),

$M_E = \{M_x(u_x, l,f*) | l \in L(u_x,f*, H)\}$ (6)

ii. Present these meta-samples to the metaclassifier,

$$g(M_E) = \{g(M_x(u_x, l,f*)) | l \in L(u_x,f*, H)\} \quad (7)$$

$$= \{g_{u_x,l} | l \in L(u_x, f*, H)\}$$

iii. Interpret the predictions into base-level labels. The multilabel classification uses S-cut threshold strategy and single label classification uses R-cut threshold strategy.

The remained problems about how to implement meta-sample representation $M_x(u_x, l, f*)$ and selection of label candidates $L(u_x, f*, H)$ are solved in the next two subsections.

4.2.1 Representations of metasamples

In this subsection, the meta-samples will map into real numerical vectors that are used by meta-classifiers. The sparse vector is used to represent meta-samples through the following steps. First, convert the scores of the related base classifiers into a sparse vector. All the nodes except the root are numbered with integers, which is used as the dimensions of the sparse vector.

Second, sparse vector is extended with the additional features about the global attributes of the root to leaf paths in the hierarchy. This step is used to raise classification accuracy. These global attributes are helpful to determine whether a path is true or not. The following three additional features are used for the pilot experiments,

1. The average score of nodes along a path;
2. The minimum score of nodes along a path;
3. The fraction of nodes whose scores exceed the thresholds employed in ScutTD, named as passrate.

At the end, the values of meta features are transferred into a practical interval in order to enhance the training of meta-classifiers. Two types of conversion functions are used for pilot experiments. For the additional features, the following standard scaling function is used,

$$z_s = \frac{s - \mu_s}{\sigma_s} \tag{8}$$

Where s is the value of an additional feature, μ_s and σ_s are the corresponding mean and variance. For the basic features, the following sigmoid function is used,

$$z_s = \frac{1}{1+e^{-(s-\mu_s)}} \tag{9}$$

Where s is a score at a node n, and μ_s is the average score at node n.

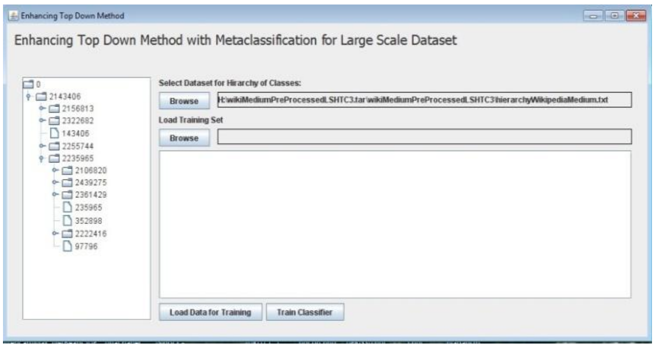
4.2.2 Selection of Label Candidates

The method used for selecting label candidates is similar to a classification method; both of them take samples as input and give the appropriate label. However, the method of selecting label should output more labels than a normal classifying method to provide a large coverage of the truly correct ones. To find such a loose classifying method, EnhancedTD will refer to RcutTD with a parameter r ≥ 2. RcutTD with r = 1 predicts one label per sample, thus it is normally used in single-labeled classification. RcutTD with r ≥ 2 predicts multiple labels per sample, which are taken as label candidates by enhanced top down method.

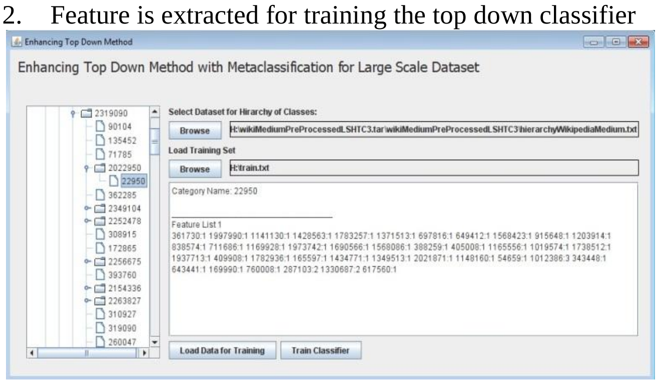
5. Result Discussion

5.1 Training phase

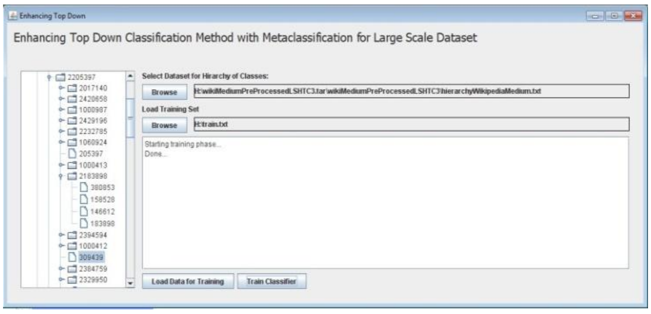
1. Hierarchy of parent child relation is built through hierarchical description given in dataset.



2. Feature is extracted for training the top down classifier



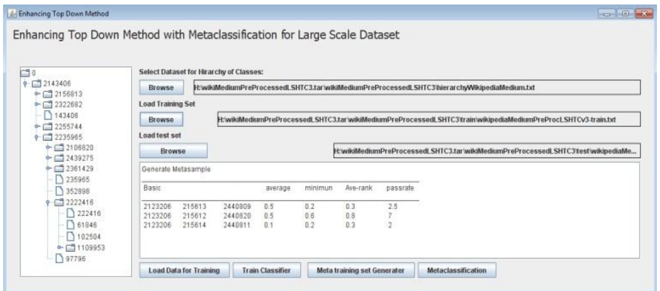
3. Top down classifier is trained using training set and hierarchical description.



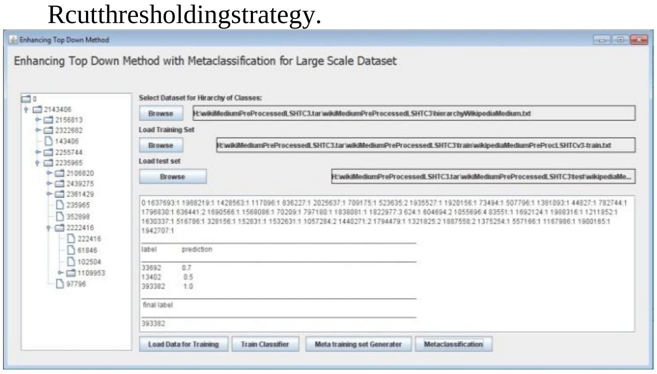
4. Meta training set generated with top down classifier. Metaclassifier is trained using metatraining set.

5.2 Testing phase

1. Test sample with unknown class is given. Metasample is constructed using the score of all base classifier along root to leaf.
2. Metaclassifier is applied on metasamples and generate set of labels. Here the score of all base classifier is considered so error propagation problem is solved.



3. Finally select the label from set of labels using Rcutthresholdingstrategy.



6. Conclusion

An Enhancing Top-Down Method with Metaclassification (EnhancedTD) will solve the error-propagation problem of the normal top-down methods, while retaining their capability for large-scale hierarchical classification. It will improve the classification accuracy of normal top-down method. It will take extra time for training and classification, so it is appropriate for most application where top down method is being used.

References

[1] Xiao-Lin Wang, Hai Zhao, and Bao-Liang Lu, Senior Member, IEEE, "A MetaTop-Down Method for Large-

Scale Hierarchical Classification, ” IEEE transaction on knowledge and data engineering , March 2014.
[2] S. Dumais and H. Chen, “Hierarchical Classification of Web Content,” Proc. 23rd Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 00), pp. 256-263, 2000.
[3] A. Sun, E.P. Lim, W.K. Ng, and J. Srivastava, “Blocking Reduction Strategies in Hierarchical Text Classification,” IEEE Trans. Knowledge and Data Engineering, vol. 16, no. 10, pp. 1305-1308, Oct. 2004.
[4] P.N. Bennett and N. Nguyen, “Refined Experts: Improving Classification in Large Taxonomies,” Proc. 32nd Intl ACM (SIGIR09), pp. 11-18, 2009.
[5] H. Malik, “Improving Hierarchical SVMS by Hierarchy Flattening and Lazy Classification,” Proc. ECIR Large-Scale Hierarchical Classification Workshop, 2010.
[6] Todorovski and S. Dzeroski, “Combining Classifiers with Meta Decision Trees,” Machine Learning, vol. 50, no. 3, pp. 223-249, 2003.
[7] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, “On Combining Classifiers,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226-239, Mar. 1998.

Author Profile

Ankita Burungale received the Bachelor degree (B.E.) in Computer engineering in 2013 from Rajarambapu Institute of Technology college of Engineering, Sakhrale, Shivaji University. Currently, She is pursuing Master’s degree in Computer Engineering at Vidya Pratishthan’s College of Engineering, BARAMATI, Pune University. Her current research interests include Data Mining and Information Retrieval.

Prof. Dinesh A. Zende has graduated in Computer Science & Engineering, in 2001, from Shivaji University, Kolhapur. He has completed Post-graduate degree from Walchand College of Engineering, Sangali (Shivaji University, Kolhapur).His research interests include Opearating Systems, Multicore Architecture, Image Processing application and Steganography.