Chapter 6

Compound Multi-label Classification Framework Based on Variable Precision Rough Set

With the diversification of the TC task, to construct multi-label classifier is often more in line with the needs of practical applications. However, in a multi-label classification task, each document often corresponds to more than one class label. In this chapter, we will construct a compound classification framework which may transform a multi-label classification task into several single label classifications and several simplified multi-label classifications. Rough Set (RS), as a mathematical tool dealing with vague and uncertain knowledge, can be appropriately applied in depicting data distribution and calculating extent of the region.

This chapter presents a compound multi-label classification framework based on Variable Precision Rough Set (VPRS), which partitions the document sample space, thus the multi-label classification task is decomposed into a number of simplified classification problems.
6.1 Reflection on Multi-label Text Classification

6.1.1 Problems in Application of Multi-label Text Classification

Text Classification (TC) techniques are now the most flourishing era, while it faces great challenges. Previous research works in text representation, feature selection and classifier construction have made great progresses. A lot of good algorithms emerged, which enhance TC results to a new level. However, with the rapid development of the Internet, the amount of texts explodes, new problems boom and new demands are constantly put forward.

The contents of TC are increasingly complex and diversified. The essence of TC is still covered by the effective organization and management of text messages. However, its "extension" has greatly enriched to deal with the objects including web pages, e-mail discussion groups, text message, instant message, Bulletin Board System (BBS) forums and so on. This makes the application of TC extend to hierarchical classification instead of simple plane classification, multi-label classification instead of single label classification. Algorithms of hierarchical classification and multi-label classification attract much research attentions.

However, let us look back and re-examine what we are faced with multi-label classification tasks: what about the constitution of classification objects? Does each document have more than one class label? The answer is negative. In fact, in most of the cases, only part of the document in multi-label classification relate to multiple labels, while the others only have a single label.

Many studies indicate that the more complex the TC task is, the worse the performance of classification. Compared with single-label TC task, it’s more difficult to get a satisfied result in multi-label TC problem. Commonly, in multi-label classification, all documents in corpus are placed under a uniform multi-label classification algorithm; as a result, it’s more likely to be inefficient and ineffective.
6.1.2 Compound Text Classification Framework

Based on above analysis, we present a compound multi-label classification framework, that is, partitioning the document sample space and then decomposing a multi-label TC task into several single-label TCs and several simplified multi-label TCs. Figure 6.1 shows this process which includes three phases: text representation, training classifier and classifying.

![Figure 6.1: Compound TC framework](image)

**Text representation**: includes preprocessing documents in training set (remove stop words); selecting features and representing documents in form of Vector Space Model (VSM).

**Training classifier**: partition sample space according to text distribution and training classifier in each space separately.

**Classifying**: represent new documents as the VSM form with feature word; compute the space that each document belongs to and classify each document by adopting corresponding classifier. That is, a new document should be classified by a single-label classifier if it belongs to the space of
certain single-label classifier, otherwise, it should be classified by certain multi-label classifier.

6.2 Compound Text Classification Framework Based on Variable Precision Rough Set

6.2.1 Variable Precision Rough Set

The original Rough Set (RS) theory is able to handle inconsistencies in the data, which occur when indiscernible objects are assigned to different classes\[^{[8]}\]. However, the values of condition and decision attributes are expected to be exact. Noisy or vague data are beyond the scope of RS theory. In many real word applications, the assumption of exact data is not fulfilled and some objects are mis-classified or condition attribute values are corrupted.

Even in noisy data, we can assume that most of the attribute values are correct. A decision table with some degree of noisy or inexact data is called probabilistic decision table in contrast to a deterministic decision table comprising completely precise data. When calculating relative reducts from a noisy decision system, it is possible to find a set of condition attributes $B \subset C$, which is sufficient to classify the majority of the objects correctly. Only a few objects would be mis-classified using the knowledge of $B$.

For example, for an attribute set $B \subset C$, a $B$-elementary set comprises 100 objects. 99 of them have the same decision class, only one is assigned to another class. RS theory does not accept $B$ as a relative reduct, even when there is only a slight inconsistency in the classification. Therefore, more attributes have to be added to $B$ to separate the misclassified object(s) from the others. As a result, relative reducts from noisy decision systems contain many condition attributes to cover the given classification for all objects of the positive region. Moreover, a high number of relative reducts can be found and these relative reducts are not stable when new objects are added to the
decision system.

To overcome these drawbacks, Ziarko developed an extension of the RS theory, VPRS model\(^{[91]}\). The basic idea of this model is to allow a small, pre-defined classification error. Generalized VPRS model is proposed by assigning asymmetric bounds to VPRS, which widens the application of VPRS\(^{[34]}\). Some fundamentals of the VPRS model are introduced in the following.

Generally, set \(X \subseteq Y\) couldn’t reflect how many elements in \(X\) belong to set \(Y\). In VPRS, a measurement is defined as \(C(X, Y)\).

\[
C(X, Y) = \begin{cases} 
1 - \frac{\text{card}(X \cap Y)}{\text{card}(X)} & \text{if } \text{card}(x) > 0 \\
0 & \text{if } \text{card}(x) = 0
\end{cases}
\]

(6.1)

It means that we would misclassify in the case of \(C(X, Y) \times 100\%\) when we try to classify the elements in set \(X\) to set \(Y\). Here, \(C(X, Y)\) should be regarded as relative classification error. Practically, the elements misclassified are \(C(X, Y) \times \text{card}(X)\) which absolute classification error is. Moreover, the relation of \(\beta\) majority inclusion defined as: \(Y_{\beta} \supseteq X\) if and only if \(C(X, Y) \leq \beta, (0 \leq \beta \leq 0.5)\).

We can further deduce the \(\beta\) lower approximation and the \(\beta\) upper approximation by generalizing relation of inclusion to the relation of \(\beta\) majority inclusion.

\[
L_{R_{\beta}}(X) = \{x \in X \mid C(x, X) \leq \beta\}
\]

(6.2)

\[
U_{R_{\beta}}(X) = \{x \in X \mid C(x, X) < 1 - \beta\}
\]

(6.3)

\(L_{R_{\beta}}(X)\) is the equivalence class set that includes the elements classified to \(X\) and classification error is less than \(\beta\). \(U_{R_{\beta}}(X)\) includes the elements which are impossible to be classified to the complementary set of \(X\) in the case that classification error is less than \(\beta\). With \(\beta = 0\), we have the case of classical
rough set. Accordingly, VPRS, an extension of rough set, inherits the characteristics and properties of classic RS and widens its application.

Asymmetry bound VPRS is a further extend of VPRS. It defines two real number $l$ and $\mu$, satisfying $0 \leq l < \mu \leq 1$. For an arbitrary subset $Y \subseteq U$, the lower approximation, upper approximation and boundary region of $Y$ is as follow.

$$L_{R_\beta}(X) = \{ x \in X | C(x, X) \leq l \}$$ (6.4)

$$U_{R_\mu}(X) = \{ x \in X | C(x, X) < \mu \}$$ (6.5)

$$BND_{R_{\mu, \beta}}(X) = \bigcup \{ x \in X | l < C(x, X) < \mu \}$$ (6.6)

Obviously, it is a VPRS when $l = \beta$ and $\mu = 1 - \beta$.

### 6.2.2 Algorithm for Partitioning Sample Space

We explain the main idea of algorithm by taking binary classification problem as example. Multi-class problem can be decomposed into several binary classification problems. Figure 6.2 exhibits the distribution of cases in 2-dimension space for a binary classification, where document belonging to positive class $C_+$ is labeled as “+” and document belonging to negative class $C_-$ is labeled as “o”.

![Figure 6.2: A binary classification problem in 2-dimension space](image)
The equivalence class of $C_+$ and $C_-$ could be constructed by $k$NN algorithm in the concept of VPRS. For a document $d$ in training set, we calculate the similarity between $d$ and each document in training set and selects the $k$ nearest neighbor of $d$, which make up of the neighbor region of $d$. If the $k$ documents in neighbor region of $d$ belong to the same class (i.e. class $C_+$), there didn’t exist uncertainty. However, there is rough uncertainty in neighbor region of $d$ if some documents in the $k$ documents belong to another class (i.e. class $C_-$). According to the concepts of VPRS, this kind of uncertainty could be defined by an improved majority inclusion relation showed as follow:

$$c(N_d, C_+) = 1 - \frac{|N_d \cap C_+|}{|N_d|}$$ (6.7)

Where, $d$ is a document in training set; $N_d$ is a neighbor region of document $d$; $|N_d|$ is the cardinality of $N_d$. The $POS_{\beta}C_+$, $BND_{\beta}C_+$ and $NEG_{\beta}C_+$ could be calculated. The partition is showed in Figure 6.3. Where, $POS_{\beta}C_+$ includes the documents belong to class $C_+$ in the miss-classification rate less than $\beta$; $NEG_{\beta}C_+$ includes the documents do not belong to $C_+$ in the miss-classification rate less than $\beta$; $BND_{\beta}C_+$ includes the documents which is hard to decide its class. The similar description is also proper for $C_-$. For every class, its space could be partitioned into $POS_{\beta}C_i$, $BND_{\beta}C_i$ and $NEG_{\beta}C_i$. Moreover, a Rocchio classifier could be trained in $POS_{\beta}C_i$. We generalize this strategy to multi-class classification problem described as follow: given corpus $U = \{d_1, \cdots, d_M\}$ and class label set $C = \{C_1, C_2, \cdots, C_N\}$, where, $d_i$ allows relate to more than one label. $\forall C_i \in C$, the space of $C_i$ is partitioned into $POS_{\beta}C_i$, $BND_{\beta}C_i$ and $NEG_{\beta}C_i$. A Rocchio classifier is trained for $C_i$. Consequently, we search the union of each $BND_{\beta}C_i$ where a multi-label classifier is constructed. Here, the classification algorithm LDA-SVM which is proposed in Chapter 4 is adopted.
Figure 6.3: Partition of two classes based on VPRS

The process of constructing single label classifier is described in Algorithm 6.1 and that of constructing multi-label classifier is showed in Algorithm 6.2.

**Algorithm 6.1**: Algorithm for Constructing Single Label Classifiers

**Input**: Parameter $k$-partition, $\beta$, $\lambda$, $\mu$, training set

**Output**: $N$ classifiers (Rocchio)

**Step1**: Search the $k$-partition nearest documents of each document in training set.

**Step2**: For $\forall C_i \in C$, construct the $R_\beta C_i$ and $\overline{R_\beta C_i}$ according to the concepts of VPRS.

**Step3**: Train a Rocchio classifier for class $C_i$ on the space of $R_\beta C_i$ (equal to $POS_\beta C_i$) as showed in the following formula:

$$
\overline{V}_{C_i} = \lambda \frac{1}{R_\beta C_i} \sum_{d \in R_\beta C_i} \frac{d}{\|d\|} - \mu \frac{1}{\overline{R_\beta C_i}} \sum_{d \in \overline{R_\beta C_i}} \frac{d}{\|d\|}
$$

(6.8)
\[ \bar{V}_C = \lambda \frac{1}{R_\beta C} \sum_{d \in \pi R_\beta C} \frac{\bar{d}}{||d||} - \mu \frac{1}{R_\beta C} \sum_{d \in \pi R_\beta C} \frac{\bar{d}}{||d||} \] (6.9)

**Algorithm 6.2:** Algorithm of Constructing Multi-label Classifiers

Input: BND\(_\beta\)C\(_1\), BND\(_\beta\)C\(_2\), ..., BND\(_\beta\)C\(_N\)

Output: M multi-label classifiers (LDA-SVM)

Step1: In BND\(_\beta\)C\(_1\), BND\(_\beta\)C\(_2\), ..., BND\(_\beta\)C\(_N\) \( \forall C_i, C_j \in C \) if BND\(_\beta\)C\(_i\) \( \cap \) BND\(_\beta\)C\(_j\) \( \neq \emptyset \), then combine the boundary region of C\(_i\) and C\(_j\) as BND\(_\beta\)C\(_i\) \( \cup \) BND\(_\beta\)C\(_j\).

Step2: \( \forall C_k \in C \) and \( k \neq i, k \neq j \), if BND\(_\beta\)C\(_k\) \( \cap \{ \text{BND\(_\beta\)C\(_i\) \( \cup \) BND\(_\beta\)C\(_j\)} \} \neq \emptyset \), then combine the boundary of C\(_k\) and the boundary of BND\(_\beta\)C\(_i\) \( \cup \) BND\(_\beta\)C\(_j\) as BND\(_\beta\)C\(_i\) \( \cup \) BND\(_\beta\)C\(_j\) \( \cup \) BND\(_\beta\)C\(_k\).

Step3: Repeat Step1 and Step2 until all boundary regions are combined. As a result, there are M unions of boundary region, noted as B\(_1\), B\(_2\), ..., B\(_M\). \( \forall B_j, j = 1,2,...,M \), construct a multi-classifier \( \mathcal{F}_j \) for B\(_j\).

**Algorithm 6.3:** Algorithm of Classification

Input: N Rocchio classifiers, M LDA-SVM classifiers, new document \( d \)

Output: class label of \( d \)

Step1: If satisfied \( \text{sim}(\bar{V}_C, d) \geq \text{sim}(\bar{V}_C, \bar{d}) \), then classify \( d \) in class C\(_+\). Jump to Step3.

Step2: Decide the boundary region B\(_i\) of \( d \) and then call classifier \( \mathcal{F}_i \) to classify document \( d \).

Step3: Output the label set of \( d \).
In Algorithm 6.3, \( \text{sim}(\mathbf{v}_c, \mathbf{d}) \geq \text{sim}(\mathbf{v}_c, \mathbf{d}) \) indicates that document \( d \) belongs to the region \( R_\beta C_+ \). We could decide that \( d \) belongs to class \( C_+ \) with high degree of belief. Otherwise, it indicates that \( d \) belongs to some boundary region \( \text{BND}_\beta C_+ \). We should call corresponding multi-label classifier to decide its label set.

### 6.2.3 Experiment and Analysis

Experiments on both Chinese corpus TanCorpV1.0 and English corpus Reuters-21578\(^{[33]} \) are performed to testify the algorithm above. Reuters-21578 is a standard corpus for text classification task\(^1 \). Here, a subset of Reuters-21578 is extracted by selecting 6 classes “Earn”, “Money-fx”, “Interest”, “Grain”, “Wheat” and “Corn” for this experiment. The distribution of each class in training set is showed in Figure 6.4.

\[\text{Figure 6.4: Distribution of 6 classes in Reuters-21578 in training set}\]

From the analysis of Chapter 3, we learn that there is high similarity between some classes in TanCorpV1.0. In fact, assigning more than one label for some documents is more reasonable. Consequently, the Chinese corpus in
this experiment is constructed by extracting part of TancorpV1.0 and relabeling the class information manually. The distribution of each class in training set is showed in Figure 6.5.

![Figure 6.5: Distribution of 6 classes in TanCorpV1.0](image)

In experiments, partitioning sample space is based on VPRS theory, single-label classifier adopt Rocchio algorithm and multi-label classifier adopt LDA-SVM algorithm proposed in chapter 4 where topic number K=100. Space calculation based on VPRS and Rocchio classifier is realized with C++ and multi-label classifiers are performed on the platform WEKA\textsuperscript{[51]}. Evaluation metric is F1-measure. We conduct 10 times experiments and get average results as follows.

Table 6.1: F1-measure of Rocchio classifiers on Reuters-21578

<table>
<thead>
<tr>
<th>Category</th>
<th>Num of document in testing set</th>
<th>F1-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earn</td>
<td>1087</td>
<td>0.96</td>
</tr>
<tr>
<td>Money</td>
<td>136</td>
<td>0.65</td>
</tr>
<tr>
<td>Interst</td>
<td>88</td>
<td>0.52</td>
</tr>
<tr>
<td>Grain</td>
<td>44</td>
<td>0.43</td>
</tr>
</tbody>
</table>
Table 6.2: F1-measure of multi-label classifiers on Reuters-21578

<table>
<thead>
<tr>
<th>Category</th>
<th>Num of document in testing set</th>
<th>F1-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>M+I</td>
<td>43</td>
<td>0.87</td>
</tr>
<tr>
<td>G+C</td>
<td>34</td>
<td>0.60</td>
</tr>
<tr>
<td>G+W</td>
<td>49</td>
<td>0.75</td>
</tr>
<tr>
<td>G+C+W</td>
<td>22</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 6.3: F1-measure of Rocchio classifiers on TanCorpV1.0

<table>
<thead>
<tr>
<th>Category</th>
<th>Num of document in testing set</th>
<th>F1-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sports</td>
<td>842</td>
<td>0.95</td>
</tr>
<tr>
<td>Entertainment</td>
<td>435</td>
<td>0.76</td>
</tr>
<tr>
<td>Art</td>
<td>149</td>
<td>0.56</td>
</tr>
<tr>
<td>Education</td>
<td>82</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table 6.4: F1-measure of multi-label classifiers on TanCorpV1.0

<table>
<thead>
<tr>
<th>Category</th>
<th>Num of document in testing set</th>
<th>F1-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edu. + Exam</td>
<td>38</td>
<td>0.89</td>
</tr>
<tr>
<td>Edu. + Exam + C</td>
<td>14</td>
<td>0.63</td>
</tr>
<tr>
<td>Edu. + C</td>
<td>30</td>
<td>0.73</td>
</tr>
<tr>
<td>Art + Enter.</td>
<td>15</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Figure 6.1 and Figure 6.3 show that the big classes have better performance for single-label classifier. For example, the F1-measure of the class “Earn” in Reuters-21578 and the class “Sports” in TanCorpV1.0 is more than 0.95. Small classes have worse performance for single-label classifier. For example, the F1-measure of the class “Grain” in Reuters-21578 and the class “Education” in TanCorpV1.0 is less than 0.5. This indicates that the single-label classifiers constructed in this framework are prone to be affected by the problem of unbalanced class, especially the small classes. Generally
speaking, the classification performance of single-label classifier on TanCorpV1.0 is better than that on Reuters-21578. The possible reason for it lies in the scale of the corpus.

Figure 6.2 and Figure 6.4 show that performances of multi-label classifiers on both corpora are good. The reason lies in the great reduction of the data scale. That is, the smaller the data scale is, the performance is more likely to be better. Multi-label classifiers in this experiment are only conducted on a subset of original corpus which has only no more than 3 labels (there are 6 labels in corpus). The classification objects are reduced also, for example, only less than 50 documents in boundary region are input into each multi-label classifier.

6.3 Conclusion

In consideration of the complex and diversified in multi-label classification task, a compound classification framework based on VPRS is raised to enhance the performance of classifiers in multi-label TC. It partitions the text space by computing the upper approximation and lower approximation. This decomposes a multi-label TC task into several single-label TCs and several multi-label TCs which have less labels than original task. That is, an unknown document should be classified by single-label classifier when it is partitioned into lower approximation space of some class. Otherwise, it should be classified by corresponding multi-label classifier.