Chapter 5

A Smoothing Latent Dirichlet Allocation Model for Text Classification Based on Tolerance Rough Set

As a kind of language model, Latent Dirichlet Allocation (LDA) should be smoothed also. In this chapter, we will first explain the reason for smoothing LDA and previous methods to solve this problem. Consequently, we propose a smoothing algorithm based on Tolerance Rough Set theory. It will be testified by experiments on both Chinese and English corpus.

5.1 Related Works

LDA is a document level language model which also involves smoothing problem. That is, the parameter $\beta$ needs to be smoothed. Otherwise, the generative probability will be zero which result in an invalid model if a new document includes new words. We could use the experience of other language models for reference. Accordingly, some smoothing strategies frequently-used are introduced in this part.
5.1.1 Smoothing Methods in Language Model

A language model is usually formulated as a probability distribution \( p(s) \) over strings \( s \) that attempts to reflect how frequently a string \( s \) occurs as a sentence. The most widely-used language models, by far, are n-gram language models. An enormous number of techniques has been proposed for smoothing n-gram models. Chen & Goodman empirically compare a variety of smoothing techniques\(^{[74]}\). Here, Lidstone smoothing and Jelinek-Mercer smoothing are introduced.

(1) Lidstone Smoothing

In statistics, Lidstone smoothing\(^{[27]}\) sometimes called additive smoothing, is a technique used to smooth categorical data.

Given a word series \( w_1w_2 \ldots w_n \), Lidstone smoothing technique is to pretend each n-gram occurs once more than it actually does\(^{[27,29,92]}\), yielding:

\[
P(w_i|w_1w_2 \ldots w_{i-1}) = \frac{N_c(w_1w_2 \ldots w_i)+\lambda}{N_c(w_1w_2 \ldots w_{i-1})+\lambda|V|}
\]  

(5.1)

Where, \( V \) is the vocabulary, the set of words being considered. \( \lambda (0 \leq \lambda \leq 1) \) is a parameter adjustable which assign non-zero probabilities to words which do not occur in the sample. It corresponds to no smoothing when \( \lambda = 0 \). It becomes frequently-used Laplacian smoothing when \( \lambda = 1 \).

From a Bayesian point of view, this result corresponds to the expected value of the posterior distribution, using a Dirichlet distribution with parameter \( \lambda \) as a prior.

(2) Jelinek-Mercer Smoothing

In general, it is useful to interpolate higher-order n-gram models with lower-order n-gram models, because when there is insufficient data to estimate a probability in the higher-order model, the lower-order model can often provide useful information. A general interpolated model is described by Jelinek and Mercer\(^{[13]}\). An elegant way of performing this interpolation is as follow\(^{[14]}\).
That is, the \( n \)th-order smoothed model is defined recursively as a linear interpolation between the \( n \)th-order maximum likelihood model and the \((n-1)\)th-order smoothed model. To end the recursion, we can take the smoothed 1st-order model to be the maximum likelihood distribution, or we can take the smoothed 0th-order model to be the uniform distribution.

\[
P_{\text{unif}}(w_i) = \frac{1}{|V|}
\]

The smoothing techniques above are mainly used in language models for the domains including speech recognition, optical character recognition, handwriting recognition, machine translation and so on. However, the smoothing techniques for language model of Text Classification (TC) are different from that of above domains. They mainly lie in two aspects: smoothing reason and smoothing object.

Smoothing reason is different. For the problem of smoothing n-gram language model, there is insufficient data to estimate a probability in the higher-order word series. The smoothing process is making use of the lower-order word series to estimate the probability of higher-order word series. In this situation, the probability of all sentences could be predicted. For the smoothing problem of TC, it is resulted from the difference between vocabularies of different classes. In a corpus, vocabulary of each class is a subset of whole vocabulary. Consequently, when a document is represented with global vocabulary, it is very likely to contain words that did not appear in its own. This phenomenon is due to variety of sparseness in text representation. It affects the performance of classification greatly.

Smoothing object is different. The aim of n-gram language model is to estimate the probability of higher-order word series by computing the probability of lower-order word series. However, only the 1th-order model
is used in TC. As a result, our smoothing objects are mainly words or other 1th-order terms.

5.1.2 Related Works in Smoothing Latent Dirichlet Allocation Model

LDA is a new model for collections of discrete data that provides full generative probabilistic semantics for documents\(^6\). Documents are modeled via a hidden Dirichlet random variable that specifies a probability distribution on a latent, low-dimensional topic space. The distribution over words of an unseen document is a continuous mixture over document space and a discrete mixture over all possible topics. It is of interest to use LDA in the generative framework for text classification. By using one LDA module for each class, we obtain a generative model for classification. But in the application of LDA, the large vocabulary size creates serious problems of sparseness. A new document may contain words that did not appear in any of the documents in the training corpus. Maximum likelihood method estimates of the multinomial parameters assign zero probability to such words, and thus zero probability to new documents.

This chapter focuses on the smoothing problem of LDA model applied to TC in generative frameworks. The standard approach to cope with this problem is to “smooth” the multinomial parameters, assigning positive probability to all vocabulary items whether or not they are observed in the training set\(^13\). To address this problem, the original paper applies variation inference methods to the extended model that includes Dirichlet smoothing on the multinomial parameter.

Data-driven smoothing strategy is provided by Li et al. in which probability mass is allocated from smoothing-data to latent variables by the intrinsic inference procedure of LDA\(^90\). In such a way, the arbitrariness of choosing latent variables’ priors for the multi-level graphical model is overcome. Following this data-driven strategy, two concrete methods, Laplacian smoothing and Jelinek-Mercer smoothing, are employed in LDA model.
A feature-enhanced smoothing method is brought in the idea that words not appeared in the training corpus can help to improve the classification performance\[11\]. The key point is fully considering the relativity between the new document and training corpus, and enhancing the document’s class feature by regarding the words not appeared in the training corpus.

5.2 Rough Set and Tolerance Rough Set

Rough Set (RS) is put forward by Poland mathematician Z. Pawlak. It is a mathematical tool that can effectively analyze and process incomplete, inconsistent, inaccurate data\[100\]. It gets widespread international concern as it has been successful applied in areas such as Knowledge Discovery (KD) in recent years. In this part, some concepts of RS are introduced including upper approximation and lower approximation, positive region and negative region, core and attributes reduction.

5.2.1 Information System and Decision Table

Let $U$ denote a finite and non-empty set called the universe, and let $R \subseteq U \times U$ denote an equivalence relation on $U$. The pair $apr = (U, R)$ is called an approximation space. The equivalence relation $R$ partitions the set $U$ into disjoint subsets. Such a partition of the universe is denoted by $U / R$. If two elements $x, y$ in $U$ belong to the same equivalence class, we say that $x$ and $y$ are indistinguishable. The equivalence classes of $R$ and the empty set $\emptyset$ are called the elementary or atomic sets in the approximation space $apr = (U, R)$.

Yao.Y.Y defined a set-based information system to be a quadruple\[99\],

$$S = (U, At, \{V_a | a \in At\}, \{f_a | a \in At\})$$ (5.4)

Where, $U$ is a nonempty set of objects, $At$ is a nonempty set of attributes; $V_a$ is a nonempty set of values for each attribute $a \in At$, $f_a : U \rightarrow 2^{V_a}$ is an information function for each attribute $a \in At$. 
An information system could be represented as another form $DT = (U, C \cup D, V, f)$. Where, $U$ is a nonempty set of objects; $V$ is a nonempty set of values for each attribute; $a \in At$, $f$ is an information function; $C \cup D$ is a nonempty limited set, that is, $A = C \cup D$ and meet $C \cap D = \emptyset$, $C$ is set of condition attributes, $D$ is set of decision attributes. This kind of information system is also named decision table. Table 5.1 is an example of information system or decision table.

Table 5.1: An information system

<table>
<thead>
<tr>
<th>Object</th>
<th>Height</th>
<th>Hair</th>
<th>Eyes</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>short</td>
<td>blond</td>
<td>blue</td>
<td>+</td>
</tr>
<tr>
<td>$O_2$</td>
<td>short</td>
<td>blond</td>
<td>brown</td>
<td>-</td>
</tr>
<tr>
<td>$O_3$</td>
<td>tall</td>
<td>red</td>
<td>blue</td>
<td>+</td>
</tr>
<tr>
<td>$O_4$</td>
<td>tall</td>
<td>dark</td>
<td>blue</td>
<td>-</td>
</tr>
<tr>
<td>$O_5$</td>
<td>tall</td>
<td>dark</td>
<td>blue</td>
<td>-</td>
</tr>
<tr>
<td>$O_6$</td>
<td>tall</td>
<td>blond</td>
<td>blue</td>
<td>+</td>
</tr>
<tr>
<td>$O_7$</td>
<td>short</td>
<td>blond</td>
<td>brown</td>
<td>-</td>
</tr>
<tr>
<td>$O_8$</td>
<td>short</td>
<td>blond</td>
<td>brown</td>
<td>-</td>
</tr>
</tbody>
</table>

The notion of information systems provides a convenient tool for the representation of objects in terms of their attribute values. If all information functions map an object to only singleton subsets of attribute values, we obtain a degenerate set-based information system commonly used in the Pawlak RS model. In this case, information functions can be expressed as $f_a : U \rightarrow V_a$. In the following discussions, we only consider this kind of information systems. We can describe relationships between objects through their attribute values. With respect to an attribute $a \in At$, a relation $R$ is given by $f(x, a) \in V_a$.

That is, two objects are considered to be indiscernible, in the view of single attribute $a$, if and only if they have exactly the same value. $R$ is an equivalence relation. The reflexivity, symmetry and transitivity of $R$ follow trivially from the properties of the relation $“=“$ between attribute values. For a subset of attributes $A \subseteq At$, this definition can be extended as follows:
CHAPTER 5. A SMOOTHING LDA MODEL FOR TC BASED ON TRS

\[ xR_a y \Leftrightarrow (\forall a \in A) f_a(x) = f_a(y) \]  

(5.5)

Where, in terms of all attributes in \( A \), \( x \) and \( y \) are indiscernible, if and only if they have the same value for every attribute in \( A \). The extended relation is still an equivalence relation.

The above discussion provides a convenient and practical method for constructing a binary relation, and in turn a Pawlak RS model. All other notions can be easily defined. For an element \( x \in U \), its equivalence class is given by:

\[ IND(R) = I_R(x) = \{(x, y) \in U \times U \mid \forall a \in R, f_a(x) = f_a(y)\} \]  

(5.6)

Where, \( I_R(x) \) is the equivalence class containing \( x \), \( IND(R) \) is an equivalent relation, in another word, an indiscernible relation. Moreover, it is a partition of \( U \), written as \( U / IND(R) \).

The equivalence relation and the induced equivalence classes may be regarded as the available information or knowledge about the objects under consideration.

5.2.2 Set Approximation

Given an arbitrary set \( X \subseteq U \), it may be impossible to describe \( X \) precisely using the equivalence classes of \( R \). That is, the available information is not sufficient to give a precise representation of \( X \). In this case, one may characterize \( X \) by a pair of lower and upper approximations:

\[ L_R(X) = \{x \in U \mid I_R(x) \subseteq X\} \]  

(5.7)

\[ U_R(X) = \{x \in U \mid I_R(x) \cap X \neq \emptyset\} \]  

(5.8)

Where, the lower approximation \( L_R(X) \) is the union of all the elementary sets which are subsets of \( X \). The upper approximation \( U_R(X) \) is the union of all the elementary sets which have a non-empty intersection with \( X \). An element in the lower approximation necessarily belongs to \( X \), while an element in the upper approximation possibly belongs to \( X \).

Based on the lower and upper approximations of a set \( X \subseteq U \), the universe
U can be divided into three disjoint regions, the positive region $POS_\rho(X) = L_\rho(X)$, the negative region $NEG_\rho(X) = U - U_\rho(X)$, and the boundary region $BND_\rho(X) = U_\rho(X) - L_\rho(X)$.

Figure 5.1: Positive, boundary and negative regions of a set X

Figure 5.1 illustrates the approximation of a set X, and the positive, negative and boundary regions. Each small rectangle represents an equivalence class. From this figure, we have the following observations. One can say with certainty that any element $x \in POS_\rho(X)$ belongs to X, and that any element $x \in NEG_\rho(X)$ does not belong to X. The upper approximation of a set X is the union of the positive and boundary regions, namely, $U_\rho(X) = POS_\rho(X) \cup BND_\rho(X)$. One cannot decide with certainty whether or not an element $x \in BND_\rho(X)$ belongs to X. For arbitrary element $x \in U_\rho(X)$, one can only conclude that $x$ possibly belongs to X. Equivalence classes are groups of objects which are indiscernible from each other, such as a group of objects in which all of the condition features are the same for each object.

5.2.3 Attribute Reduction

In information system, there often exists some condition attributes that do not provide any additional information about the objects in U. We should remove those attributes since the complexity and cost of the decision process can be reduced if those condition attributes are eliminated [52,53].
Assuming $P$ and $Q$ are equivalence relations in $U$, the important concept positive region $\text{Pos}_P(Q) = \bigcup_{x \in U/Q} P x$. A positive region contains all patterns in $U$ that can be classified in attribute set $Q$ using the information in attribute set $P$.

The degree of dependency $\gamma(C,D)$ between the condition attributes $C$ and the decision attributes $D$ is defined as:

$$\gamma(C,D) = \frac{\text{Card}(\text{Pos}(C,D))}{\text{Card}(U)}$$

(5.9)

Where, Card denotes the cardinality of set $U$.

The degree of dependency provides a measure of how important $C$ is in mapping the dataset examples into $D$. If $\gamma(C,D)=0$, then classification attribute $D$ is independent of the attributes in $C$, hence the decision attributes are of no use to this classification. If $\gamma(C,D)=1$, then $D$ is completely dependent on $C$, hence the attributes are indispensable.

Values $0 < \gamma(C,D) < 1$ denote partial dependency, which shows that only some of the attributes in $P$ may be useful, or that the dataset was flawed to begin with. In addition, the complement of dependency $\gamma(C,D)$ gives a measure of the contradictions in the selected subset of the dataset.

In information system $IS = \langle U, A, V, f \rangle$, $B \subseteq A$ and $a \in B$, attribute $a$ is redundant, if $I_a = I_{B \setminus \{a\}}$, vice versa, $a$ is necessary in $B$. $B$ is independent if $\forall a \in B$, $a$ is necessary, vice versa, $B$ is dependent. Assume $B' \subseteq B$, if $B'$ is independent and $I_{B'} = I_B'$, then $B'$ is a reduction of $B$. Reduction of $B$ may be more than one. All reduction of $B$ compose a set $\text{RED}(B)$. All the necessary relation in $B$ make up of the core of $B$, written as $\text{CORE}(B)$. The intersection of all reduction in attribute set is equal to core, i.e. for set $B$, $\text{CORE}(B) = \cap\text{RED}(B)$.

A decision table may have more than one reduct. Any one of them can be used to replace the original table. Finding all the reducts from a decision...
table is a NP-Hard\cite{45} problem. Fortunately, in many real applications it is usually unnecessary to find all of them; one is sufficient. A natural question is which reduct is the best if there are more than one reduct. The selection depends on the optimality criterion associated with the attributes. If it is possible to assign a cost function to attributes, the selection can be naturally based on the combined minimum cost criteria.

\section{5.2.4 Tolerance Rough Set}

The classical RS theory is based on equivalence relation that divides the universe of objects into disjoint classes\cite{62}. Practically, for some applications, the requirement for equivalent relation has showed to be too strict. And it must be extended. For example, let us consider a collection of scientific documents and keywords describing those documents. It is clear that each document can have several keywords and a keyword can be associated with many documents. Thus, in the universe of documents, keywords can form overlapping classes. By relaxing the relation $R$ to a tolerance relation, where transitivity property is not required, a generalized tolerance space is introduced \cite{86,57,37,98}. The RS theory can be useful for application to real-valued data. RS theory employs a similarity relation to minimize data as opposed to the indiscernible relation used in classical RS. This allows a relaxation in the way equivalence classes are considered. By employing this relaxation, the granularity of the rough equivalence classes has been reduced. This flexibility allows a blurring of the boundaries of the former rough or crisp equivalence classes and objects may now belong to more than one tolerance class.

An information system is an incomplete system if some attributes are not valued or we only know part of information about these attributes. To solve this problem, a method proposed by M. Kryszkiewicz is to assign a “NULL” to the elements non-valued. “NULL” represents a possibility of any value\cite{45}. Given an information system $IS = \langle U, A, \mathcal{V}, f \rangle$, we assign “*” represent each of the non-valued attributes which is a subset of attribute set, signed as $B \subseteq A$. The tolerance relation on $B$ is describes as follow:
\[ T(B) = \{(x, y) \in U \times U \mid \forall b \in B, b(x) = b(y) \lor b(x) = \ast \lor b(y) = \ast \} \] (5.10)

Where, \( B \subseteq A \). Obviously, \( T \) is reflexive and symmetry. But the transitivity property is not always meted. \( \forall y \in B \) and \( I_b(x) = \{y \in U \mid (x, y) \in T(B)\} \), \( I_b(x) \) is the tolerance class of object \( x \) on \( B \subseteq A \).

We could further define the upper approximation and lower approximation of \( X \) on \( B \subseteq A \) based on the definition of tolerance class.

Given an incomplete information system \( IS = (U, A, V, f) \), \( X \subseteq U \), \( B \subseteq A \), the upper approximation and lower approximation of \( X \) about \( B \) in the tolerance relation \( T \) is as follow:

\[ U_B(X) = \{x \in U \mid I_B(x) \cap X \neq \emptyset\} \] (5.10)

\[ L_B(X) = \{x \in U \mid I_B(x) \subseteq X\} \] (5.11)

### 5.3 A Smoothing Latent Dirichlet Allocation Model

#### Based on Tolerance Rough Set

##### 5.3.1 Problem of Smoothing Latent Dirichlet Allocation

As analyzed in 5.1.2, the difference between the vocabularies of each class results in sparseness. In this case, LDA model \((\alpha', \beta')\) learned from documents of class \( i \) could only compute generative probability of a new document according to the vocabulary of class \( i \). Unknown word in object documents will bring zero probability, as a result, the model is invalid.

LDA model \((\alpha', \beta')\) learned from documents of class \( i \) has two parameters \( \alpha' \) and \( \beta' \). The initial value of \( \alpha' \) could be manually assigned. However, \( \beta' \) which is related to vocabulary should be smoothed.

**Definition:** Assume that \( v_i \) is vocabulary of class \( c_i \), the
out-of-vocabulary (oov) words of class $c_i$ is represented as follow:

$$oov_i = \bigcup_{j \neq i} v_j - v_i$$

(5.12)

For the application of text classification, unknown words will increase greatly with growing in number of class. This may degrade the performance of model, especially for an unbalance corpus.

Blei designed a direct smoothing method, that is, setting an exchangeable prior Dirichlet distribution $Dir(\eta)$ for $\beta$ [6]. The smoothed model is showed in Figure 5.2. Bayes inference is performed on this smoothed model and the estimation of parameter $\beta'$ is as follow:

$$\beta'_j = \eta + \beta_j$$

(5.13)

**Figure 5.2: Smoothing LDA based on exchangeable Dirichlet prior distribution**

In fact, this process only adds a positive value on original estimate. Obviously, every word in $\beta'$ is not zero, as a result, zero probability is avoided in LDA model. Wherever, there are some disadvantages in this smoothing strategy.

It is likely to modify latent parameter $\beta$ arbitrarily. $\beta$ is related to latent variation $(\theta, z)$ and $\alpha$. Modification of $\beta$ may directly neglect the effect on other parameters.

The selection of exchangeable Dirichlet prior is in consideration of the
conductible of LDA model. It confines the distribution of smoothing method.

The assignment of parameter $\eta$ is according to experience. There is no principal for its assignment.

5.3.2 Smoothing Latent Dirichlet Allocation Model Based on Tolerance Rough Set

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of document and $T = \{t_1, t_2, \ldots, t_m\}$ be a set of index terms for $D$. In TRS, the tolerance space is defined over a universe of all index terms $U = T = \{t_1, t_2, \ldots, t_m\}$.

The idea is to capture conceptually related index terms into classes. For this purpose, the tolerance relation $R$ is determined as the co-occurrence of index terms in all documents from $D$. The choice of co-occurrence of index terms to define tolerance relation is motivated by its meaningful interpretation of the semantic relation in context and its relatively simple and efficient computation.

(1) Tolerance Class of Term

Let $f_D(t_i, t_j)$ denotes the number of documents in $D$ in which both term $t_i$ and term $t_j$ occurs. The uncertainty function $I$ with regards to threshold $\theta$ is defined as follow.

$$I_\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\} \quad (5.14)$$

Clearly, the above function satisfies conditions of being reflexive: $t_i \in I_\theta(t_i)$ and symmetric: $t_j \in I_\theta(t_i) \iff t_i \in I_\theta(t_j)$ for any $t_i, t_j \in T$. Thus, the tolerance relation $I \subseteq T \times T$ can be defined by means of function $I$:

$$t_i, t_j \iff t_j \in I_\theta(t_i) \quad (5.15)$$

Where, $I_\theta(t_i)$ is the tolerance class of index term $t_i$.

In context of TC, a tolerance class represents a concept that is characterized by terms it contains. By varying the threshold $\theta$ (e.g. relatively to the size of document collection), one can control the degree of
relatedness of words in tolerance classes (or in other words the preciseness of the concept represented by a tolerance class).

To measure degree of inclusion of one set in another, vague inclusion function is defined as:

$$v(X, Y) = \frac{|X \cap Y|}{|X|}$$ \hspace{1cm} (5.16)

It is clear that this function is monotonous with respect to the second argument. The membership function for it

$$\mu(t_x, X) = v(I_\phi(t_x), X) = \frac{|I_\phi(t_x) \cap X|}{|I_\phi(t_x)|}$$ \hspace{1cm} (5.17)

With the assumption that the set of index terms $T$ doesn't change in the application, all tolerance classes of terms are considered as structural subsets:

$$P(I_\phi(t_x)) = 1 \text{ for all } t_x \in T$$

Finally, the lower and upper approximations of any subset $X \subseteq T$ can be determined with the obtained tolerance $R = (T, I, v, P)$ as:

$$L_R(X) = \{ t_x \in T \mid v(I_\phi(t_x), X) = 1 \}$$ \hspace{1cm} (5.18)

$$U_R(X) = \{ t_x \in T \mid v(I_\phi(t_x), X) > 0 \}$$ \hspace{1cm} (5.19)

One interpretation of the given approximations can be as follows: if we treat $X$ as an concept described vaguely by index terms it contains, then $U_R(X)$ is the set of concepts that share some semantic meanings with $X$, while $L_R(X)$ is a "core" concept of $X$.

Another kind of representation is described as:

$$L_R(d_i) = \{ t_x \in T \mid I_\phi(t_x) \subseteq d_i \}$$ \hspace{1cm} (5.20)

$$U_R(d_i) = \{ t_x \in T \mid I_\phi(t_x) \cap d_i \neq \emptyset \}$$ \hspace{1cm} (5.21)

With TRS, the aim is enriching representation of document by taking into consideration not only terms actually occurring in document but also other
related terms with similar meanings. A "richer" representation of document can be acquired by representing document as set of tolerance classes of terms it contains. This is achieved by simply representing document with its upper approximation. Let \( d_i = \{t_{i1}, t_{i2}, \ldots, t_{ik} \} \) be a document in \( D \) and \( t_{i1}, t_{i2}, \ldots, t_{ik} \in T \) are index terms of \( d_i \):

\[
U_R(d_i) = \{ t_j \in T \mid v(I_\theta(t_j), X) > 0 \}
\]  

(5.22)  

(2) Algorithm of Producing Term Tolerance Class

Tolerance class of index term of documents is co-occurrence of index terms, which is described with matrix \( TOL = [tol_{i,j}]_{M \times N} \). Detailed algorithm is as follow.

**Algorithm 5.1:** Algorithm of producing term tolerance class

Input: TF (matrix of “document*term”) and \( \theta \) (co-occurrence threshold)

Output: \( TOL \) (binary matrix of term tolerance class)

Step1: Transform \( tf \) to the form of \( OC = [oc_{i,j}]_{N \times M} \) which is defined as follow:

\[
c_{i,j} = \begin{cases} 
1 & \text{if } tf_{i,j} > 0 \\
0 & \text{otherwise} 
\end{cases}
\]

(5.23)

Where, \( tf_{i,j} = 1 \) when term \( t_i \) appear in document \( d_i \).

Step2: Build co-occurrence matrix \( COC = [coc_{x,y}]_{M \times M} \) which is defined as:

\[
oc_{x,y} = Card(OC^x \ AND \ OC^y)
\]

(5.24)

Where, \( OC^x \), \( OC^y \) is the column vector in matrix \( OC \); \( Card \) is the cardinality of vector; \( coc_{x,y} \) is the co-occurrence times of term \( x, y \).

Step3: Given threshold \( \theta \), filter the cell in matrix \( COC \) and got the matrix of term tolerance class \( TOL \).
\[ TOL = [tol_{x,y}]_{\times M} \]

\[ tol_{x,y} = \begin{cases} 
1 & \text{cocl}_{x,y} > \theta \\
0 & \text{others} 
\end{cases} \quad (5.25) \]

Where, \( tol_{x,y} = 1 \) means that there exists tolerance relation between term \( x \) and \( y \).

(3) **Smoothing Latent Dirichlet Allocation Model Based on Tolerance Rough Set (TRS-LDA).**

By using one LDA module for each class, we obtain a generative model for classification. But in the application of LDA, the large vocabulary size creates serious problems of sparseness. A new document may contain words that did not appear in any of the documents in the training corpus. Consequently, LDA model must be smoothed. To avoid the disadvantage in classic LDA model, a strategy based on TRS is proposed in this chapter. Its main idea is compensating the out-of-vocabulary (oov) word list for class \( i \) and assigning corresponding value according to information coming from its tolerance class. In this case, the zero probability could be avoided.

First, the oov word list is added to class \( i \) and LDA model could be trained in extend vocabulary. The inference mechanism in LDA could be maturely applied, that is, parameter \( (\theta, z) \) and \( \alpha \) could be served in smoothing process with the changing of \( w \). This strategy avoids the arbitrary in modification parameters in model.

Secondly, we could adjust the scale of tolerance class to meet different situation. Different from the classic smoothing strategy, it could adopt various prior distributions more flexibly.

We revise the classic Laplacian smoothing to the form described as follow:
CHAPTER 5. A SMOOTHING LDA MODEL FOR TC BASED ON TRS

\[ P(w_t|c_i) = \frac{n_t(w_t) + \lambda}{n_t(c_i) + n_t|V|} \]  

\[ \lambda = \begin{cases} 
  tf_{ij} + 1 & \text{if } t_j \in d_i \\
  \varphi + \min_{k \in d_i, T_{ik}} TF_{ik} & \text{if } t_j \in U_R(d_i) \wedge t_j \notin d_i \\
  \varphi & \text{if } t_j \notin U_R(d_i) 
\end{cases} \]  

Where, \( tf_{ij} \) is the frequency of term \( t_j \) in document \( d_i \), \( \varphi, \rho \in [0, 1] \). In this chapter, \( \varphi = \rho = 0.2 \). That is, for an unknown word, we assign its prior value according to three cases: appearing in \( d_i \), being absent in \( d_i \) but appearing in \( U_R(d_i) \) and being absent in \( U_R(d_i) \).

We make use of the idea of Laplacian transform to add virtual vocabulary to each class and assign its value as Formula 5.27.

The smoothing strategy has different effect on big classes and small classes. Normally, there is much more vocabulary in big class, as a result, the virtual vocabulary added has little change on its original one, vice verse, for a small class. Consequently, a perfect smoothing algorithm could effectively improve the classification performance on small classes.

5.3.3 Experiments and Analysis

Experiments are performed on both Chinese corpus TanCorpV1.0 and English corpus 20Newsgroup. 20Newsgroup was collected by Ken Lang for text classification research\[32\]. All documents are distributed across 20 categories evenly (balanced), altogether 20,000 texts. TanCorpV1.0 which is collected by Dr. Tan Songbo includes 14,150 texts, distributing in 12 categories (unbalanced)\[72\].

In our experiments, the Micro-F1 and the Macro-F1 measures are used to evaluate the synthesis classification performance. We perform the comparison between our method and traditional LDA model. For the LDA model, the number of latent topics is an important factor which defines the granularity of the model. So, we evaluate models on different topic number.

(1) Experiments on Analysis Synthesis Performance on 20Newsgroup.
In Figure 5.3, the $X$-axis is the number of latent topics used in the two models and the $Y$-axis is the synthesis performance measure (Micro-F1 or Macro-F1).

**Figure 5.3(a): Classification results on TRS-LDA (Micro-F1 in 20Newsgroup)**

**Figure 5.3(b): Classification results on TRS-LDA (Macro-F1 in 20Newsgroup)**
From the Figure 5.3(a) and Figure 5.3(b), we can find:

1) On the whole, TRS-LDA model has an evident higher performance of about 6%~7% than LDA model both on Micro-F1 and Macro-F1. This occurs across all values of topic number.

2) When different latent topic number is selected, the behaviors of the two models are different. With the increase of topic number, the LDA change little. However, the TRS-LDA model improves little by little with the increase of topic number.

3) Micro-F1 and Macro-F1 have the very consistent results due to 20Newsgroup is a balance corpus.

(2) Experiments on Analysis Synthesis Performance on TanCorpV1.0:

In Figure 5.4, the X-axis is the number of latent topics used in the two models and the Y-axis is the synthesis performance measure (Micro-F1 or Macro-F1).

![Figure 5.4(a): Classification results on TRS-LDA (Micro-F1 in TanCorpV1.0)](image-url)
From the Figure 5.4(a) and Figure 5.4(b), we can find:

1) TRS-LDA model has an evident higher performance of about 7% than LDA model on Micro-F1 across all values of topic number.

2) TRS-LDA model has an evident higher performance of about 20% than LDA model on Macro-F1 across all values of topic number.

3) When different latent topic number is selected, the behaviors of the two models are different. With the increase of topic number, the Micro-F1 of LDA change little. However, that of TRS-LDA model improves little by little with the increase of topic number.

4) Micro-F1 and Macro-F1 have the very inconsistent results due to TanCorpV1.0 is an unbalanced corpus. We have the conclusion that TRS-LDA could improve the classification performance obviously on unbalanced corpus. For classification task in unbalanced dataset, worse result is often prone to small class result from its inefficient training. TRS-LDA could improve this phenomenon to some extend.
From the results of experiments on both 20Newgroup and TanCorpV1.0, we conclude:

- TRS-LDA can enhance the classification performance (Micro-F1 and Macro-F1) whether on balance or on unbalanced corpus.
- TRS-LDA can greatly raise the Macro-F1 on unbalanced corpus. Consequently, TRS-LDA has superiority on unbalance classification problem.

(3) Analysis on performance of each class on unbalanced corpus

Let us observe the effectiveness of TRS-LDA from the more refined granularity. Figure 5.5 gives the distribution of document amount in each class in TanCorpV1.0. The largest class includes nearly 3000 documents while the smallest class includes only 150 documents.

![Figure 5.5: Document distribution of TanCorpV1.0](image)

Figure 5.6 shows the F1 performance on each class on TanCorpV1.0, where the No. of class is on the sequence of class scale (up). We could find that F1 values of TRS-LDA model are always higher than those of LDA model. The increase of F1 values on big classes are less than those in small
classes. It indicates that TRS-LDA’s contribution lies in improving classification effectiveness on small classes, in the same time, not degrading the performance on big classes.

Figure 5.6: F1 value on each class of TanCorpV1.0

Figure 5.7: Recall values on each class of TanCorpV1.0
Figure 5.8: Precision values on each class of TanCorpV1.0

Figure 5.7 and Figure 5.8 shows separately the recall and precision metric on each class. We could observe that the tendency of recall is consistent with that of F1. On the contrary, the value of precision is turbulent greatly both on TRS-LDA model and on LDA model, especially on small classes. For big classes, the value of precision on TRS-LDA is near to that of LDA.

We summarize the performance of three metrics and have the conclusion: TRS-LDA enhances classification performance mainly by improving classification effectiveness on small classes. Effect of smoothing algorithm lies in relieving the over-fitting on small class.

5.4 Conclusion

In a view of the disadvantage in LDA model, that is, arbitrarily revising the variable in smooth process, a new strategy for smoothing based on Tolerance Rough Set is put forward. It constructs tolerant class in global vocabulary database firstly and then assigns value for unknown words in each class according to tolerance class. Experiments on both Chinese and
English corpus indicate that TRS-LDA can enhance the classification performance obviously whether for balance or for unbalanced dataset.