Information Modelling and Knowledge Bases XXXIII
M. Tropmann-Frick et al. (Eds.)

doi:10.3233/FAIA210489

Improving Maximum Entropy Model by GIS

Junichi KAWAMI [a] and Takao MIURA [a]
[a] *Dept.of Advanced Sciences, HOSEI University*
Kajinocho 3-7-2, Koganei, Tokyo, Japan
Email:junichi.kawami.8x@stu.hosei.ac.jp miurat@k.hosei.ac.jp

Abstract. Maximum Entropy Model (MEM)[1][4] estimates probability distribution functions, by which current state of knowledge is described in the context of prior data. Here we examine Generalized Iterative Scaling (GIS)[1] algorithm to determine optimum feature weights with feature selection during learning. Maximum Entropy principle[1] provides us with all the characteristics of the data given in advance and we could expect robust distribution against outlier. However it takes much time until convergence because the computation depends heavily on the number of classes. We introduce a novel approach random sampling of Monte Carlo method into GIS for improved computation.

Keywords. Natural Language Processing, Multiple Classification, Maximum Entropy Model, Monte Carlo method, Sampling

1. Introduction

Recently wide-spread internet allows us to analyse and extract what we could have and how we could do from the view point of both quantity and quality. Most of data are written in text and we should examine them with natural language processing (NLP). For example, very often we classify document d into one of given classes $C = \{c_1,...,c_m\}$, the problem is called *document classification*. Let D be a set of documents over words W and a document d in D, we consider d as a vector $[t_1,t_2,..,t_n]$ over words $W = \{w_1,w_2,...,w_n\}$, where t_j means frequency of w_j appeared in d. Note d is, in fact, a vector over W not a list. There have been many approaches proposed, but Maximum Entropy Model(MEM) works very well in the classification problem. As well know, we fact to data sparseness problems in NLP. MEM helps us to extract characteristic context by entropy and to make inferences on the basis of partial information.

Each word may carries several meanings. It is hard to identify interest words suitable for the current context of documents. N-grams or collocations mean a set of words to carry single semantics as a whole. We can separate them onto word to obtain the semantics. All these aspects cause hard tasks to solve classification problems correctly.

One of problems over MEM comes from how to estimate probabilities, GIS is one of algorithms for estimating the parameters of MEM. It helps us to compute these parameters empirically and approximately. However it takes much times until convergence because the computation depends heavily on the number of classes.

In this work, we propose a novel approach Generalized Iterative Scaling (GIS) based on random sampling improves computation. In fact, the marginal probability causes the heavy computation of the probability Summarization to all the classes, as integral calculation by random sampling. Compared to GIS which computes marginal probability with all classes during learning, it becomes faster to obtain approximation.

Our results contribute to NPL research focusing on the following points; (1) Our approach can improves efficiency of GIS algorithm with the help of sampling techniques and (2) We propose a sophisticated technique to introduce feature selection. By examining a collection of learning data, we *mine* effective functions in terms of association rules so that we improve classification dramatically and that we can complete feature selection automatically.

The rest of the paper is organized as follows. In section 2 we describe fundamental roles of document vectors, vector space model and document classification. In section 3, we describe Maximum Entropy Model and the model generation. Section 4 contains how to apply random sampling to the model calculation, and our sample generation in section 5. Section 6 contains some experimental results to see the effectiveness and we conclude this investigation in section 7.

2. Maximum Entropy Model

Conceptually MEM approach helps us to model all that is known in advance and few about what is unknown. In other words, we like to obtain probability model satisfying a set of constraints which represent ”evidence” and choose the *most uniform* distribution otherwise because the distribution carries the maximum entropy or the minimum commitment. One way to represent evidence is to encode characteristic facts as *features*. Any kind of contextual feature can be used in the model, and experimenters generally need to focus theirs efforts on deciding what features to use. The representation of the evidence discussed below, then determines the form p.

Given an input vector $\vec{x}$ of a document over words, we like to classify the $\vec{x}$, i.e., to estimate a class c to which $\vec{x}$ belongs. To build a classifier from the viewpoint of MEM, this means we like to estimate a class c' of the maximum probability $p(c'|\vec{x})$, i.e., $c' = \arg\max_{c} p(c|\vec{x})$.

2.1. Modeling by Maximum Entropy

In MEM, we assume a set of *features* f in advance to a word w in W and a class c in C which we intend to mention w is characteristic to c, i.e., w is a *feature word* of c. Given w, a feature is a function $f_w(\vec{x}, y)$ where $\vec{x}$ means a document vector and y a class:

$$f_{w,c}(\vec{x}, y) : W \times C \to \{0, 1\} \tag{1}$$

$$f_{w,c}(\vec{x}, y) = 1 \text{ if } w \in \vec{x} \text{ and } y = c,\ 0 \text{ otherwise} \tag{2}$$

Since features show characteristic aspects of our documents of interests, it could also be useful to describe classification. Here we assume each document may belong to one class as well as a word. To have conditional probability distribution function $p(c|\vec{x})$

given a class c and an input $\vec{x}$. We also assume q_w if we give a constraint " a document d containing a word w belongs to a class c" in terms of expects over features and their distribution. Then let q_w be a distribution of relative frequency over $W \times C$, and can be seen as a probability function of $(\vec{x}, y)$ as a constraint defined as :

$$E_p[f_w] = \sum_{\vec{x},y} q_w(\vec{x},y) f_w(\vec{x},y) = \frac{1}{N} \sum_{\vec{x},y} f_w(\vec{x},y) \tag{3}$$

Note N means the size of domains $W \times C$. Then we give an expect of the distribution p as our constraints of $w, \ldots$:

$$E_p[f_w] = \sum_{\vec{x},y} p_w(\vec{x},y) f_w(\vec{x},y) \tag{4}$$

The constraints wrt (w,c) can be described as:

$$E_p[f_w] = E_q[f_w] \tag{5}$$

$$\sum_{\vec{x},y} p_w(\vec{x},y) = 1 \tag{6}$$

Since the objective is to maximize entropy $H(p) = \sum p(\vec{x},c)\log(1/p(\vec{x},c))$ subject to the constraints above. To estimate the distribution p, we apply *Lagrange Multipliers* to our model by maximizing $L(p)$

$$\begin{aligned} L(p) = H(p) + \sum_{w} \lambda_w (E_p[f_w] - E_q[f_w]) \\ + \lambda_0 (\sum_{(\vec{x},y))} (p(\vec{x},y) - 1) \end{aligned} \tag{7}$$

Then we have the solution below:

$$\begin{aligned} p(\vec{x},y) &= exp\{ \sum_{w \in W} \lambda_w f_w(\vec{x},y)\}/Z \\ Z &= exp\{1 - \lambda_0\} \\ &= \sum_{\vec{x},y} exp\{ \sum_{w \in W} \lambda_w f_w(\vec{x},y)\} \end{aligned} \tag{8}$$

Note $p(y|\vec{x}) = p(\vec{x},y)/p(\vec{x}) = p(\vec{x},y)/\sum_y p(\vec{x},y)$.

Ratnaparkhi has examined several models : it is always possible to get such p in a unique manner and discussed how to do that. Once we obtain MEM, we could classify documents. Clearly the results depends heavily on both the selection of features and the parameters λ_w. There have been several algorithms such as *Generalized Iterative Scaling* (GIS) and *Improved Iterative Scaling* (IIS) proposed so far. However, all of them take much time to obtain the values.

2.2. Training parameters using GIS

To obtain model-parameters λs of MEM probability function, there have been proposed two useful algorithms, Generalized Iterative Scaling(GIS) and Improved Iterative Scaling(IIS) .Both algorithms work in an iterative scaling manner based on a gradient method. The parameters shows how important role the feature plays to classification task.

Let us illustrate how GIS works in a case of single classification in a Table 1:

Table 1. An outline of GIS algorithm

1. Assume all the feature $f_1,...,f_K$ are given in advance.
   And also assume q an initial distribution.
2. Let C and f_{K+1} be an auxiliary constant and a feature. $C = \max_{(\vec{x},y)} \sum_{j=1}^{K} f_j(\vec{x},y)$
3. Set $\omega_i^0 = 0.0, i = 1,...,K+1$
4. improve ω_i^k as follows where N is the size of traing data:
   $\omega_i^{K+1} = \omega_i^k + \log \frac{1}{C} \frac{E_q[f_j]}{E_p[f_j]}$
5. Repeat 4 until convegence.

We like to obtain our goal, a probability density function $p(\vec{x}|y;\omega)$. To do that, we have to estimate parameters ω. Note in 2 we define a constraint C and a feature f_{K+1} additionally to simplify the algorithm. Step 4 describes our constraints $E_q[f_j]$ in terms of the features:

$$E_p[f_j] = \sum_{\vec{x},y} q_j(\vec{x},y) f_j(\vec{x},y) = \frac{1}{N} \sum_{\vec{x},y} f_j(\vec{x},y) \tag{9}$$

Also $E_p[f_j]$ in step3 describes our constraint of probability distribution p in terms of the features.

$$E_p[f_j] = \sum_{i=1}^{N} \sum_{y \in Y(x_i)} p_j(\vec{x},y) f_j(\vec{x},y) \tag{10}$$

Similarly we repeat the whole process to improve ω_1 values until we get to $E_p[f_j] = E_q[f_j]$. Then we eventually obtain our model p. During GIS processes, it is impossible to avoid heavy computation. In fact, once we obtain $E_q[f_i]$ for initialization, we approximate $Ep[f_i]$ $O(|P(D)| \times |C|)$ times for each feature f_j .

3. Multiple classification and Feature Selection

By a word "multiple classification", we mean that an object belongs to a class softly. That is, we assume it belongs to a single class but we don't know explicitly, and we could have some knowledge with possibility by means of distribution over classes. We discuss multiple classification with MEM approach. To do that, we explore how to extract feature functions based on frequent patterns appeared in training data. We define our patterns as features, by which we can consider a set of words automatically as single feature so that we can construct MEM for multiple classification.

GIS helps us to compute parameters empirically and approximately such as feature weights. However it takes much time until convergence because the computation depends heavily on the number of classes. Multiple classification allows us to label a document multiple class, GIS plays critical role on classification task. In this investigation, we discuss how to apply MEM to multiple classification. Here let us discuss how to extend MEM, especially feature functions, and GIS algorithm.

3.1. Feature function for multiple classification

By a word ”multiple classification”, we mean that an object belongs to a class softly. That is, we assume it belongs to a single class but we don’t know explicitly, and we could have some knowledge with possibility by means of distribution over classes.

Since features show characteristic aspects of our documents of interests, it could also be useful to describe multiple classification. For example, a sentence *scientists who study viruses say they don’t know what a pandemic strain would look like* could belong to class ”health”. Similarly a word *pandemic* is characteristic to the class. However if class set C contain ”economy”, only *pandemic* could not assign a document to ’health” or ”economy”. In the sentence, we could get information that *pandemic* and *virus* could be characteristic to ”health”. In other words, for classification , characteristics to a class are not a word, but they are set of words at same time in a document, and the discovery of interesting associations and correlations between a set of words and classes helps us to assign documents to classes.

The association rule is an implication of the form $U \Rightarrow c$ where U is a set of words and c is a class. The rule $U \Rightarrow c$ holds in the document set D with *support*, where *support* is the percentage of documents in D that contain U and c. The rule $U \Rightarrow c$ has *confidence* in the documents set D, where *confidence* is the percentage of document in D containing U that also contain c. This is taken to be the conditional probability, $p(c|U)$. That is,

$$support_{U,c} = \frac{Frequency\ of\ documents\ containing\ U\ and\ c}{|D|} \tag{11}$$

$$confidence_{U,c} = \frac{Frequency\ of\ documents\ containing\ U\ and\ c}{Frequency\ of\ document\ containing\ U} = p(c|U) \tag{12}$$

a rule that satisfies both a minimum support threshold ($minsup_{U,c}$) and minimum confidence threshold ($minconf_{U,c}$) helps us to solve classification problems. The occurrence frequency of a set of words is the number of documents that contains the set of words. If the support of a set of words U satisfies a prespecified *minsup*, then U is a frequent set of words. We describe how to be corresponded rules to feature functions.

Given a document $\vec{x}$ and a class y over C, we extend the definition of a feature function $f(\vec{x}, y)$ as follows:

$f : P(W) \times C \rightarrow [0,1]$
$f_{U,c}(\vec{x}, y) = p_{U,c}$ if $U \subseteq \vec{x}$ and $y = c$, 0 otherwise

$p_{U,c} = p(c|U) = confidence_{U,c}$

The constraint wrt $(\vec{x}, y)$ can be described as:

$\sum_{y \in C} f_{U,c}(\vec{x}, y) = 1$

3.2. Feature Selection

Let us describe how to examine feature functions from learning data. To select feature function of MEM, we propose feature selection to use a *minsup* threshold to ensure the generation of a set of frequencies a set of words and a *minconf* threshold to ensure a set of correlations of a set of words. The discovery of interesting associations and correlations between a set of words and classes helps us to assign documents to classes.

Let us illustrate how the feature selection works in a case of multiple classification, when *minsup=0.3,minconf=0.4*.

Table 2. DB

Document	authority	virus	impact	pandemic	Class
x_1	1	1	0	1	economy
x_2	0	0	1	1	economy
x_3	0	1	0	1	health
x_4	0	1	0	0	health

Table 3. Frequency a set of words and classes in DB

A set of words	Frequency	economy	health
{authority}	1	1	0
{virus}	3	1	2
{impact}	1	1	0
{pandemic}	3	2	1
{authority, virus}	1	1	0
{authority, pandemic}	1	1	0
{virus, pandemic}	2	1	1
{authority, virus, pandemic}	1	1	0

Table 4. Confidence

A set of words	frequency	confidence	
		economy	health
{virus}	3	0.33	0.66
{pandemic}	3	0.66	0.33
{virus, pandemic}	2	0.5	0.5

Table 2 shows a DB, Table 3 shows frequencies of a set of words and classes at the same time in the DB and Table 4 shows *confidence* which equals to $p_{U,c}$ of each a set of words exceeding *minsup=0.3* and classes.

For all that have the same set of words, we select feature functions based on a set of words and classes which have confidence exceeding *minconf=0.4*. We show selected feature below:

$$f_{\{virus\},health}(\vec{x},y) = 1.0 \qquad f_{\{pandemic\},economy}(\vec{x},y) = 1.0$$

$$f_{\{virus,pandemic\},economy}(\vec{x},y) = 0.5 \qquad f_{\{virus,pandemic\},health}(\vec{x},y) = 0.5$$

In the selected feature functions, U helps us to assign documents to class c to follow rules ($U \Rightarrow c$).

3.3. GIS for multiple Classification

Here we have to discuss how to extend MEM, especially GIS algorithm. GIS provides us with the parameters λ_w to feature functions $f(\vec{x},y)$ to estimate $p(c|\vec{x})$. Note it takes heavy computation.

However, we see GIS is not efficient because of Z, a normalization term. In fact, Z is nothing but marginalization of each class so that we need the integral values. When we estimate $p(y)$ for each class y in such a way that $Z(\vec{x}) = \int_y p(\vec{x},y)dy$:

$$\begin{aligned} Z(\vec{x}) &= \sum_{y \in P(Y)} exp\{\phi(\vec{x},y)\} \\ &= \sum_{y \in P(Y)} \frac{exp\{\phi(\vec{x},y)\}}{p(y)} p(y) \\ &\approx \frac{1}{M} \sum_{k=1}^{M} \frac{exp\{\phi(\vec{x},y_k)\}}{p(y_k)} \quad y_k \sim p(y) \end{aligned} \tag{13}$$

As input $(\vec{x},y)$, we give training data (with class y). In this investigation, using random sampling, we approximate the computation appropriately and efficiently. That is, given a set of multi-class distribution Y , we take samples $y_1, y_2, ..., y_M$ where $y \sim p(Y)$. As p(Y), we assume our base probability as follows.

$$p(y_i) = \frac{FrequencyDistribution + 1}{|D| + |Y|} \tag{14}$$

To generate samples, we generate $0.0 \leq u \leq 1.0$ through uniform distribution and obtain v such that $\sum_{i=1}^{v} p(y_i) \leq u < \sum_{i=1}^{v} p(y_i)$. Since we assume all the classes are independent with each other, we take M times.

4. EXPERIMENTS

We show our experimental results to see how well the proposed approach works. We discuss our results for multiple classification using MEM. As the baseline, we compare normal GIS with our approach to focus on accuracy of classification, learning time, time which computes normalization term, and a rate of time to calculate denominator in learning.

4.1. Preliminaries

UCI KDD Archive contains datasets[6], as such Reuters-21578 Text Categorization Collection , for document classification. The dataset is composed of text and category labeled topic, people, place, and orgs. We examine the corpus containing documents labeled topic expect class of *earn* and *acq* in documents. Table 5 shows details of learning data and test data.

In feature selection, we select feature functions based on a set of words and classes satisfying $minsup = 0.15$, $minconf = 0.05$. Table 6 shows detail of features, Table 7 shows the number of feature functions for each of classes, $confidence$ is not equal to zero, and Table8 shows the number of documents of each class in both of learning data and test data documents topic as one piece of data in this experiment. We examine MEM by GIS based on random sampling and normal GIS. The number of update, feature function, learning data and test data of GIS based on random sampling has the same as the baseline.

Table 5. Details of corpas

	Learning data	Test data
File	reut2-000 reut2-001 reut2-002 reut2-003 reut2-004 reut2-005	reut2-014 reut2-015 reut2-016
Using Tag	Text：<BODY> Class:<TOPIC>	Text：<BODY> Class:<TOPIC>
The number of documents	776	334
The number of kind of class which appeared in data.	25	25

Table 6. Details of features

	Baseline	Proposed GIS
The number of feature functions	279 (Including correction function)	279 (Including correction function)
The number of kind of extracted class	10	10
The number of kind of sets of words	278	278
MinSuppot	0.04	0.04
MinConfidence	0.55	0.55

4.2. *Results*

In Tables 10, 11, 12, and 13, let us illustrate our results of accuracy, recall, precision and F-measure in both baseline and GIS based on random sampling as the number of sample is 5, 10, 15, 20, 25, 30, 35, 40, 45, 50. Table 9 shows results of learning time, time which computes normalization term, and a rate of time to calculate denominator in learning.

4.3. *Discussion*

4.3.1. *Classification*

Let us discuss what our results of classification means. In the case that the number of sample is 5, there are difference of results to accuracy, recall, precision and F-measure compared to the baseline with more six kinds of classes shown in Tables 10,11,12,13. In the case that the number of sample is 10 and 15, there are difference of results to accuracy, recall, precision and F-measure compared to the baseline with more two kinds of classes but the same results to MicroAve of accuracy, recall, precision, and F-measure. In the case that the number of sample is 20, 25, 30 and 35, there are difference of results to accuracy, recall, precision and F-measure compared to the baseline with more one kind of class. In the case that the number of sample is 40, 45, 50, we got result to recall of the same baseline.

Table 7. Class containing feature functions

Class	The number of feature functions
crude	163
trade	74
interest	14
coffee	12
money-fx	6
gnp	5
gold	1
sugar	1
ship	1
money-supply	1

Table 8. In learning data and test data the number of document to each of class

Class	The number of document	
	Learning data	Test data
alum	11	5
bop	12	4
cocoa	10	5
coffee	44	9
copper	12	9
cpi	18	8
crude	122	22
gnp	24	6
gold	34	11
grain	17	9
housing	9	0
interest	42	28
ipi	11	7
jobs	16	7
money-fx	53	31
money-supply	35	14
nat-gas	9	5
orange	10	6
reserves	15	2
retail	9	2
rubber	16	4
ship	60	8
sugar	39	19
trade	88	48
wpi	8	6

Table 9. Time which compute GIS

	BaseLine	The number of sample									
		5	10	15	20	25	30	35	40	45	50
Learning time(ms)	122567	29427	52498	77068	94323	120438	138685	162993	185359	206978	223334
Time which compute Z (ms)	98608	21096	41567	62794	79171	100953	118654	138708	161242	180388	194518
A rate of time to compute Z	0.805	0.717	0.791	0.815	0.839	0.838	0.856	0.851	0.869	0.871	0.871

Table 10. Accuracy

Class	BaseLine	The number of sample									
		5	10	15	20	25	30	35	40	45	50
nat-gas	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982
bop	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985
housing	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
ship	0.982	0.985	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982
retail	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993
crude	0.945	0.858	0.935	0.949	0.949	0.949	0.949	0.949	0.945	0.945	0.945
money-fx	0.902	0.913	0.905	0.905	0.902	0.902	0.902	0.902	0.902	0.902	0.902
gold	0.982	0.978	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982
interest	0.778	0.782	0.771	0.775	0.785	0.785	0.785	0.785	0.785	0.785	0.782
rubber	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985
copper	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967
grain	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967
cpi	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971
ipi	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975
jobs	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975	0.975
gnp	0.985	0.985	0.985	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982
reserves	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993
cocoa	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982
alum	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982
orange	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978	0.978
wpi	0.945	0.949	0.945	0.945	0.945	0.945	0.945	0.945	0.945	0.945	0.945
trade	0.909	0.884	0.913	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909
coffee	0.971	0.956	0.971	0.971	0.971	0.971	0.971	0.971	0.967	0.967	0.971
money-supply	0.942	0.949	0.953	0.942	0.942	0.942	0.942	0.942	0.942	0.942	0.942
sugar	0.993	0.964	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993	0.993
MicroAve	0.963	0.958	0.963	0.963	0.963	0.963	0.963	0.963	0.963	0.963	0.963

In the case that the number of sample is 40, we got result of 8.3% and 0.3% difference to accuracy, precision, recall and F-measure in classes of ”trade” and ”coffee”. In the case that the number of sample is more 50, we got result of the same baseline .

On the other hand, Table 13 shows parameter robustly update with GIS based on Monte Carlo method, so that the more less the number of samples decrease, the more the

Table 11. Recall

Class	BaseLine	The number of sample									
		5	10	15	20	25	30	35	40	45	50
nat-gas	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
bop	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
housing	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ship	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
retail	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
crude	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955	0.955
money-fx	0.290	0.258	0.258	0.258	0.290	0.290	0.290	0.290	0.290	0.290	0.290
gold	0.636	0.545	0.636	0.636	0.636	0.636	0.636	0.636	0.636	0.636	0.636
interest	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893	0.893
rubber	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
copper	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
grain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cpi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ipi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
jobs	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
gnp	0.667	0.500	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667
reserves	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cocoa	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
alum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
orange	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
wpi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
trade	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.979
coffee	0.778	0.667	0.889	0.889	0.889	0.889	0.889	0.889	0.778	0.778	0.778
money-supply	0.429	0.000	0.429	0.429	0.429	0.429	0.429	0.429	0.429	0.429	0.429
sugar	0.895	0.474	0.895	0.895	0.895	0.895	0.895	0.895	0.895	0.895	0.895
MicroAve	0.535	0.469	0.535	0.535	0.538	0.538	0.538	0.538	0.535	0.535	0.535

Table 12. Precision

Class	BaseLine	The number of sample									
		5	10	15	20	25	30	35	40	45	50
nat-gas	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
bop	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
housing	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ship	0.800	1.000	0.800	0.800	0.800	0.800	0.800	0.800	0.800	0.800	0.800
retail	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
crude	0.600	0.356	0.553	0.618	0.618	0.618	0.618	0.618	0.600	0.600	0.600
money-fx	0.643	0.889	0.727	0.727	0.643	0.643	0.643	0.643	0.643	0.643	0.643
gold	0.875	0.857	0.875	0.875	0.875	0.875	0.875	0.875	0.875	0.875	0.875
interest	0.301	0.305	0.294	0.298	0.309	0.309	0.309	0.309	0.309	0.309	0.305
rubber	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
copper	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
grain	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cpi	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ipi	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
jobs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gnp	0.667	0.750	0.667	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571
reserves	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cocoa	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
alum	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
orange	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
wpi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
trade	0.662	0.603	0.671	0.662	0.662	0.662	0.662	0.662	0.662	0.662	0.662
coffee	0.538	0.400	0.533	0.533	0.533	0.533	0.533	0.533	0.500	0.500	0.538
money-supply	0.429	NaN	0.545	0.429	0.429	0.429	0.429	0.429	0.429	0.429	0.429
sugar	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MicroAve	0.535	0.469	0.535	0.535	0.538	0.538	0.538	0.538	0.535	0.535	0.535

result is difficult. Our experiments show that decreasing the number of sample causes difference to result of baseline and GIS based on random sampling.

4.3.2. Learning Time

Let us discuss what our results of leaning time means. Table 9 shows the results of learning time and a rate of time to calculate denominator in learning. To all because baseline the computation of the probability Summarization to 25 kind of classes. In the case that samples are less than 20, each of time is less than learning time of the baseline, and is more than learning time of baseline in the case that samples are more than 25. On the other hand, it takes time to sample class by the inverse function method. We consider, if the number of samples equals to the number of classes while baseline appeared, learning time by GIS based on random sampling is more than learning time of baseline.

In the sampling, GIS based on random sampling without rejection, adopt all classes generated by sampling. Thus overhead hardly happens and improve GIS. Finally, the proposed GIS expects that the we expand dimension in probability distribution, the learning time decreases leaning time.

Table 13. F-measure

Class	BaseLine	The number of sample									
		5	10	15	20	25	30	35	40	45	50
nat-gas	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
bop	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
housing	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ship	0.615	0.667	0.615	0.615	0.615	0.615	0.615	0.615	0.615	0.615	0.615
retail	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
crude	0.737	0.519	0.700	0.750	0.750	0.750	0.750	0.750	0.737	0.737	0.737
money-fx	0.400	0.400	0.381	0.381	0.400	0.400	0.400	0.400	0.400	0.400	0.400
gold	0.737	0.667	0.737	0.737	0.737	0.737	0.737	0.737	0.737	0.737	0.737
interest	0.450	0.455	0.442	0.446	0.459	0.459	0.459	0.459	0.459	0.459	0.455
rubber	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
copper	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
grain	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cpi	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ipi	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
jobs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gnp	0.667	0.600	0.667	0.615	0.615	0.615	0.615	0.615	0.615	0.615	0.615
reserves	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cocoa	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
alum	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
orange	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
wpi	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
trade	0.790	0.746	0.797	0.790	0.790	0.790	0.790	0.790	0.790	0.790	0.790
coffee	0.636	0.500	0.667	0.667	0.667	0.667	0.667	0.667	0.609	0.609	0.636
money-supply	0.429	NaN	0.480	0.429	0.429	0.429	0.429	0.429	0.429	0.429	0.429
sugar	0.944	0.643	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944
MicorAve	0.535	0.469	0.535	0.535	0.538	0.538	0.538	0.538	0.535	0.535	0.535

5. CONCLUSION

In this work, we have proposed an approach of improved GIS based on random sampling by which the marginal probability causes the computation of the probability Summarization to all the classes.

Our experimental results showed that improved MEM by GIS takes advantages to the traditional GIS: we got 42% learning time more compared to the baseline, keeping the same results to precision recall and f0.3% difference to recall in two kind of class.

We expect to apply our approach to other kinds of languages in classification problem.

References

[1] Adwait Ratnaparkhi. ”A Simple Introduction to Maximum Entropy Models for Natural Language Processing.” 1997

[2] J,N,DARROCH and D,RATCLIFF ”GENERALIZED ITERATIVE SCALING FOR LOG-LINEAR MODELS.” 1972

[3] Bing Liu Wynne Hsu and Yiming Ma ”Integraing Classification and Association Rule Mining.” 1998

[4] Ambedkar Dukkipati, Abhay Kumar Yadav and M. Narasimha Murty ”Maximum entoropy model based Classication with Feature Selection.” 1972

[5] https://sites.google.com/site/kevinbouge/stopwords-lists

[6] https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html