A Machine Learning Approach to Reducing the Work of Experts in Article Selection from Database: A Case Study for Regulatory Relations of *S. cerevisiae* Genes in MEDLINE

Shin-ichi Usuzaka [1]
shin@ib.sci.yamaguchi-u.ac.jp

Kim Lan Sim [2]
klsim@ims.u-tokyo.ac.jp

Miyako Tanaka [3]
miyako@ube-k.ac.jp

Hiroshi Matsuno [1]
matsuno@sci.yamaguchi-u.ac.jp

Satoru Miyano [2]
miyano@ims.u-tokyo.ac.jp

[1] Faculty of Science, Yamaguchi University, 1677-1 Yoshida, Yamaguchi 753-8512, Japan
[2] Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1, Shirokanedai, Minatoku 108-8639, Japan
[3] Department of Business Adminstration, Ube National College of Technology, 2557, Tokiwadai, Ube 755-8555, Japan

Abstract

We consider the problem of selecting the articles of experts' interest from a literature database with the assistance of a machine learning system. For this purpose, we propose the rough reading strategy which combines the experts' knowledge with the machine learning system. For the articles converted through the rough reading strategy, we employ the learning system BONSAI and apply it for discovering rules which may reduce the work of experts in selecting the articles. Furthermore, we devise an algorithm which iterates the above procedure until almost all records of experts' interest are selected. Experimental results by using the articles from *Cell* show that almost all records of experts' interest are selected while reducing the works of experts drastically.

1 Introduction

Consider the following query to the database:

Find all records in the database MEDLINE with the following property:
(Q) *The record contains information which are useful for building the gene regulatory network of* Saccharomyces cerevisiae.

With a conventional information retrieval system such as Entrez and DBGET, it is very easy to select the set A of the records in MEDLINE with the keyword *S. cerevisiae*. In fact, A consists of more than 35,000 records. However, it is difficult to squeeze the records satisfying the above property (Q) out of the set A since the property (Q) is described in a rather ambiguous and technical way. Unless experts will read the contents of the records in A in an exhaustive way, it may be impossible to answer the above query correctly. Moreover, it is sometimes still difficult for an expert to decide only with the information given in a record, i.e., abstract, title, keywords, etc., if the article has the property (Q) or not even if the record is carefully read.

As a preliminary investigation, we have dealt with the articles in *Cell*. *Cell* has published 758 articles with a key word *S. cerevisiae*. An expert has read all these 758 records in MEDLINE carefully and classified them into *relevant records* and *irrelevant records* by a criterion that a record involves information about regulatory relations between genes of *S. cerevisiae* with which the gene regulatory network of *S. cerevisiae* can be drawn. The number of relevant records was 210 and it took a whole week for an expert to finish this work. Judging from this investigation, it can be considered feasible to accomplish this work for 35,000 records in a year but it requires a great amount of time and labor. It is a practically important problem to cope with such hard queries which essentially requires an exhaustive investigation by experts. In this situation, we need a systematic strategy for reducing the work of experts in the process of choosing relevant records from these 35,000 records. If we wish to make a complete set of relevant records, *all* records must be read by experts. However, in this paper, we allow some relevant records to be missed. But we require some guarantee that the number of missed records is small.

The purpose of this paper is to devise a strategy for combine the experts' knowledge with the machine learning system for discovering rules so that the work of experts shall be reduced. In the literature, intelligent information retrieval systems have been extensively studied together with natural language processing technology. Especially, intelligent information systems with the inductive learning ability have been shown successful in helping identify the most significant document index terms with various levels of relationship to their semantic significance [1, 2, 3, 4]. Basically, the system developed in the paper is on this line of research. If all the records in A could be read by experts, we could have the most correct set of records with the property (Q). Since this reading process is very expensive, our target is to reduce the work of experts while the number of records which satisfy the property (Q) but will not be read by experts shall be kept very small.

For this purpose, we consider a strategy which combines the experts' knowledge with the machine learning system. As the experts' knowledge, we introduce a notion of rough reading. When experts read an abstract of an article for such classification, they first read the abstract roughly and then, if it is considered as a candidate, they read the abstract carefully. The notion of rough reading is defined by abstracting this observation. Thus, this rough reading process converts an abstract to a text viewed with the experts' knowledge. For the abstracts converted through the rough reading strategy, we employ the machine learning system BONSAI [5] and apply it for discovering rules which may reduce the work of experts in selecting the records of interest from A. Our strategy is to automate this process with the help of experts. We formulate this strategy in the mathematical fashion. For experiments, we collected and classified 758 records from *Cell* by reading the whole documents in the records. Then, by using these 758 records, we made computational experiments to see the performance of this strategy. The results show that our strategy can reduce the work of experts. However, there are still many relevant records which have not been chosen. This is not satisfactory since we want to collect *all* records with the property (Q). Then we devise an algorithm which iterates the above procedure until almost all relevant records are collected. We analyze the algorithm with respect to convergence and prove the upper and lower bounds of the number of records which should be read by experts. Experimental results by using the records from *Cell* show that almost all relevant records are collected while reducing the work of experts drastically.

2 Pre-Reading Classification by Learning System

2.1 Rough Reading Strategy

By following the terminology of computational learning, we call a relevant record a *positive example* and an irrelevant record a *negative example*.

Let U be the set of records from which positive examples are to be extracted. For a subset S of U, we denote by S^+ and S^- the sets of positive and negative examples, respectively. For each $x \in U$, let $Word(x)$ be the set of words appearing in the record x. A *rough reading function* is a mapping $\rho : \bigcup_{x \in U} Word(x) \to \Sigma$, where Σ be a finite set of symbols called the *rough reading alphabet*. For each $x = w_1 w_2 \dots w_n$ in U with $n \geq 1$, let

$$R(x) = \rho(w_1)\rho(w_2) \cdots \rho(w_n) = a_1 a_2 \dots a_n,$$

where w_i $(1 \leq i \leq n)$ is a word in $Word(x)$ and $\rho(w_i) = a_i \in \Sigma$ for $1 \leq i \leq n$. We assume that the function $R : U \to \Sigma^*$ is a one-to-one mapping given *a priori*. Then, for a subset S of U, we call the set $R(S)$ the *rough reading image of S*. Since the rough reading function is one-to-one, we may confuse elements in U with the corresponding elements in $R(U)$ in the following discussion. We assume a machine learning system M that can produce a classification rule from two sets POS and NEG of strings with $POS \cap NEG = \emptyset$. The accuracy of a classification rule of M for the set POS (NEG) is denoted by p $(0 \leq p \leq 1)$ (n $(0 \leq n \leq 1)$). It means that $p|POS|$ strings in POS ($n|NEG|$ strings in NEG) are recognized as positive (negative) by the classification rule.

Experts choose a subset $T = T^+ \cup T^-$ as a *training set* for the machine learning system. We assume that the set T is divided by experts into two disjoint sets T^+ and T^-, a set of positive examples and a set of negative examples, respectively. The set $E = U - T$ remains for classification and we call it the *examining set*.

Since R is assumed to be one-to-one, the rough reading image $R(T)$ of T is also divided into two disjoint sets $R(T^+)$ and $R(T^-)$ which exactly correspond to the set T^+ and T^-, respectively. Then we input the sets $R(T^+)$ and $R(T^-)$ to the learning system M to find a classification rule B. Then by using the classification rule B, the set $R(E)$ is divided into two disjoint sets P' and N'. Then we obtain a partition $E = P \cup N$ of E, where $P = R^{-1}(P')$, $N = R^{-1}(N')$. A *rough reading strategy* is this process for dividing E into two disjoint sets P and N by using a training set T, a rough reading function R and a machine learning system M. Note that this strategy enables us to combine the experts' knowledge with the machine learning system.

Consider the set $T = T^+ \cup T^-$ and $E = U - T = E^+ \cup E^-$, where T is partitioned by experts into T^+ and T^- but the partition $E = E^+ \cup E^-$ is not known. Let p (n) denote the accuracy of the classification rule B for the set $R(T^+)$ ($R(T^-)$). If we could assume that the accuracy of the classification rule B for the set $R(E^+)$ ($R(E^-)$) is the same as p (n), we have the following observation[1] : By applying the classification rule of M with accuracies p and n, we can get $p|E^+| + (1-n)|E^-|$ examples selected as positive and $(1-p)|E^+| + n|E^-|$ examples selected as negative from E. Thus, from the inequality

$$\frac{p|E^+|}{p|E^+| + (1-n)|E^-|} > \frac{|E^+|}{|E^+| + |E^-|},$$

if $p + n > 1$ holds, we can conclude that the rough reading strategy reduces the task of experts.

2.2 Method for Constructing Rough Reading Alphabet and Function

The arguments in the previous section are based on the assumption that experts already have a rough reading alphabet Σ and a rough reading function ρ as the experts' knowledge. In order to perform experiments on a rough reading strategy for query (Q) in Introduction, we must have these Σ and ρ. In the following, we show the way how we make these Σ and ρ.

Experts can decide whether a given article is relevant or not by a quick view on the abstract. We consider that this work of experts consists of two processes. The first is a process of selecting important words related to the interest without taking the meaning carefully. For example, if the abstract contains several gene names and a word "suppressed", then the article is guessed to involve some gene regulatory relation between genes. Then in the next process, the guess will be checked by a careful reading of the abstract. Obviously both processes require experts' knowledge and ability. But the first process can be automated to some extent if the experts' knowledge about the importance of words is provided in advance.

In this section, we consider a rough reading alphabet $\Sigma = \{A, x, y, z, o\}$. Each character of Σ has the following meaning and numerical scores are assigned to x as 3, y as 2, z as 1 and o as 0.

[1] In fact, these accuracies on the examining set would be worse than the accuracies on the training set. We will take this fact into considerations later.

A: Gene Name, x: Very relevant, y: Relevant, z: Weakly relevant o: Not relevant

We should also consider a rough reading function for the set of records concerning *S. cerevisiae* from MEDLINE as a knowledge given *a priori.* However, in practice, instead of dealing with all words in the set, we use the records of *Cell.* That is, let U be the set of 758 records of *Cell* from MEDLINE. Then, let U^+ be the set of 210 relevant records and let $U^- = U - U^+$. Let $W_C = \bigcup_{x \in U} Word(x)$. For each word w in W_C, let $f_p(w)$ ($f_n(w)$) be the number of occurrences of the word w in the abstracts in U^+ (U^-). Let

$$\begin{aligned} \textit{frequency}(w) &= \frac{f_p(w) + f_n(w)}{\sum_{u \in W_C} f_p(u) + f_n(u)} \\ \textit{shift}(w) &= \frac{f_p(w)}{f_p(w) + f_n(w)}. \end{aligned}$$

Let φ and σ be fixed numbers with $0 \leq \varphi, \sigma \leq 1$ that will be used as thresholds for $\textit{frequency}(w)$ and $\textit{shift}(w)$, respectively. Let $W_C[\varphi, \sigma] = \{w \in W_C \mid \textit{frequency}(w) \geq \varphi \text{ and } \textit{shift}(w) \geq \sigma\}$. For our construction, we set $\varphi = 0.032$ and $\sigma = 0.65$. The set $W_C[\varphi, \sigma]$ consists of words which occur frequently and have a tendency to occur in the abstracts in U^+. Therefore these words are useful for discriminating U^+ and U^-.

All gene names are automatically classified as the category A and all words except for gene names which are not in $W_C[\varphi, \sigma]$ are classified as the category o. Then experts classify the words in $W_C[\varphi, \sigma]$ except for gene names into the categories of x, y and z according to the degree of relevance. The relationship between genes are being identified based on the words that are being used in the abstracts. It has been observed that certain deterministic words such as "whereas", "suppress/suppresses", "repression", etc. and gene names will invariably describe the gene relationships. On the contrary, most ancillary words carry no significant description of gene relationships. The extent of such significance is being exploited here in an organized manner whereby the different words used are being classified in a range of different degree of usefulness.

A few guidelines are followed in the process of this classification. Firstly, if there are no gene names in the whole paragraph, the abstract is ignored since we are only interested in gene relationships. Secondly, classification of the abstracts based on the words used are read as in the whole paragraph context. This is due to the fact that sometimes, the explanation in the latter sentences are actually continuation of the former. If the same word is found to occur several times in the same paragraph, the word is being designated (x, y, z, o) on separate accounts and an average score is calculated. For example, the word "catalytic" has 2 os, 2 ys, 2 xs and 1 z. The average score for this word would be $(2 \times 0) + (2 \times 2) + (2 \times 3) + (2 \times 1) = 12/7 = 1.7$. Therefore the final designation for the word "catalytic" should be y (score $= 2$). In the case where 1 z and 1 r are encountered, let's say for the word "WWW", the degree of importance can be set to the category z (0.5), since there is a possibility that the word "WWW" is used to express some relationships between genes. If this word is not important as a result, this word will be ignored during the process of learning. Correspondingly, in words that have extreme scores only (both xs and os in them), the average scores generated would indicate some degrees of importance for these words.

In this way we classify the words in W_C and obtain a rough reading function ρ_C for W_C. The distribution of the words in W_C is [(x, 47), (y, 66), (z, 95), (o, 6159), (A, 557)], where each of these pairs represents (category, the number of words). An abstract is converted to a text over Σ as in Table 1.

Then we define a rough reading function $\rho : \bigcup_{x \in U} Word(x) \to \Sigma$ by

$$\rho(w) = \begin{cases} A & \text{if } w \text{ is a gene name,} \\ o & \text{if } w \text{ is a gene name and not in } W_C, \\ \rho_C(w) & \text{otherwise.} \end{cases}$$

Table 1: Abstract conversion.

abstract	CYC7-H3 is a cis-dominant regulatory mutation that causes a 20-fold overproduction of yeast iso-2-cytochrome c. The CYC7-H3 mutation is an approximately 5 kb deletion with one breakpoint located in the 5' noncoding region of the CYC7 gene, approximately 200 base from the ATG initiation codon. The deletion apparently fuses a new regulatory region to the structural portion of the CYC7 locus. The CYC7-H3 deletion encompasses the RAD23 locus, which controls UV sensitivity and the ANP1 locus, which controls osmotic sensitivity. The gene cluster CYC7-RAD23-ANP1 displays striking similarity to the gene cluster CYC1-OSM1-RAD7, which controls, respectively, iso-1-cytochrome c, osmotic sensitivity and UV sensitivity. We suggest that these gene clusters are related by an ancient transpositional event.
conversion	AooxooooooooooooAooooooyoooooooooooAooooooooooyooooooooooooAzoAyooAzo ozyooAzoooyoooAoooooooAooooooyozyooooooooooooo

Table 2: Performance of Rough Reading Strategy.

	Training set		Examining set			
random seed	p	n	p	n	read record ratio (%)	positive record ratio (%)
0	0.85	0.96	0.68	0.84	30.8	68.4
1	0.95	0.88	0.66	0.84	29.7	65.8
2	0.90	0.90	0.51	0.89	22.4	51.1
3	0.95	0.84	0.50	0.90	20.9	50.0
4	0.95	0.84	0.76	0.81	35.0	75.8
5	0.95	0.92	0.51	0.87	23.3	51.1
6	0.90	0.86	0.63	0.80	31.7	63.2
7	0.85	0.90	0.77	0.82	34.4	76.8
8	0.85	0.90	0.55	0.84	26.7	54.7
9	0.90	0.94	0.54	0.86	24.7	53.7

2.3 Experiment with BONSAI

We use BONSAI [5] as a machine learning system. Given positive and negative examples of strings, BONSAI will find an alphabet indexing and a decision trees over regular patterns as a classification rule. An alphabet indexing is a classification of symbols into a smaller categories to reduce the size of the alphabet.

Let *POS* and *NEG* be the sets of all positive and negative examples, respectively. BONSAI chooses small samples *pos* and *neg* of positive and negative examples at random from *POS* and *NEG* for constructing an alphabet indexing and a decision tree. BONSAI is tuned by four parameters; window size, iteration number, maximum pattern length, and random number seed. We briefly summarize these parameters (refer to [5] in detail).

The *window size* is the number of strings chosen into *pos* (*neg*) from *POS* (*NEG*). BONSAI is repeated specified times by *iteration number* by changing pos and neg. BONSAI uses only regular patterns of the form xwy, where x and y are variables and w is a string. The upper bound of the length of w is specified by a parameter *maximum pattern length*. A random number used for generating the initial alphabet indexing which is called the *random number seed*. This random number seed is also used for choosing a small sample of positive and negative examples. *Index size* is the number of letters used by the alphabet indexing.

Experiments have been run to observe the effect of a rough reading strategy with BONSAI. We arbitrarily choose the training set T consisting of 20 positive examples (T^+) and 50 negative examples (T^-) in accordance with the ratio of 210 positive and 548 negative examples in the records of *Cell* (U). Table 2 shows the result. By changing the random number seed, we have made experiments ten times. Other parameters of BONSAI are set as widow size=10, iteration number=10, max pattern length=9, and index size=3.

Two columns p and n at the row of training set in the Table 2 indicate the accuracies described in 2.1 of the rule B obtained by BONSAI using the training set of 20+50 records. The other two columns p and n at the row of examining set indicate the same accuracies, but values at the column p (n) represent the ratio of the number of examples selected by the classification rule B and the number of records in E^+ (E^-) of examining set $E = U - T$.

At every rows in the columns p and n of examining set, we can see that $p + n > 1$ holds. This implies that the rough reading strategy with BONSAI works well. Furthermore, we can see the good performance of this strategy by comparing the "read record ratio" and "positive record ratio" in Table 2, where "read record ratio" is the percentage of records in the examining set E which have been read by experts and "positive record ratio" is the percentage of positive examples in the set E^+ found by experts. That is, "Nearly 70% (50%) of positive examples are found by experts, although they have read only about 30% (20%) of the set of whole records U."

We must note that BONSAI produces a classification rule which reflects not only the number of characters A, x, y, z, o but also the contexts of sentences to some extent by occurrence patterns of characters A, x, y, z, o.

3 Iterative Method for Collecting Almost All Positive Examples

The results in the previous section show that our strategy can reduce the work of experts. However, there are still many relevant records which have not been chosen. This is not a satisfactory situation since we want to collect *all* records with the property (Q). Then we devise an algorithm which iterates the procedure in Section 2 until almost all relevant records are collected. We analyze the algorithm with respect to convergence and prove the upper and lower bounds of the number of records which should be read by experts.

3.1 Algorithm

We define the following procedures.

- EXPERT(W): The procedure divide a set of examples W into two disjoint sets, the set of positive examples W^+ and the set of negative examples W^-. We must take the steps performed by the procedure EXPERT into consideration, since it reflects the work of experts which should be reduced. The number of steps is called the *cost* in the following arguments.

- BONSAI(Z^+, Z^-): From the sets of positive examples and negative examples, the procedure gets $B_{(x,y)}$ that represents a classification rule expressed as a pair of a decision tree and an alphabet indexing which selects $x|Z^+|$ elements in Z^+ truly as positive and $y|Z^-|$ elements in Z^- truly as negative.

- POSKAMO($S_i, B_{(x_i,y_i)}$): By applying the classification rule $B_{(x,y)}$ produced by the procedure BONSAI to the set S_i of examples, the procedure takes out a subset K_i of S_i as a collection of "probable positive examples."

A formal description of the algorithm [Iterative Algorithm] is given below. This algorithm gets all positive examples with the help of experts and a machine learning system which correspond to the procedures EXPERT and BONSAI as above, respectively, in theoretical sense.

[Iterative Algorithm]

INPUT: a set U of examples

OUTPUT: a set of positive examples in the set U

```
begin
  Selects a subset V ⊆ U arbitrarily;
  (V+, V−) ←EXPERT(V);
  B(x0,y0) ←BONSAI(V+, V−);
  S0 ← U − V;
  K0 ←POSKAMO(S0, B(x0,y0));
  (P0, N0) ←EXPERT(K0);
  if x0 = 1 return(P0 ∪ V+)
  else if y0 = 0 return((S0 − N0) ∪ V+))
  else
    begin
      i = 1;
      while (xi ≠ 1 and yi ≠ 0 )
        begin
          B(xi,yi) ←BONSAI(Pi−1, Ni−1);
          Si ← Si−1 − Ki−1;
          Ki ←POSKAMO(Si, B(xi,yi));
          (Pi, Ni) ←EXPERT(Ki);
          i ← i + 1;
        end
      if xi = 1
        return(⋃_{j=0}^{i} Pj ∪ V+)
      else
        return((S0 − ⋃_{j=0}^{i} Nj) ∪ V+)
    end
end
```

Now, we show the validness of the algorithm. The efficiency of the algorithm with respect to the cost of the procedure EXPERT is also demonstrated. First, the following lemma is needed for analysis.

Lemma 1

(1) Let $\{x_i\}_{i\geq 0}$ be a series with $0 \leq x_i \leq 1$. If $0 \leq x_i < 1$ for $0 \leq i \leq m-1$ and $x_m = 1$ for some $m \geq 1$, then $x_0 + \sum_{i=1}^{m} x_i \Pi_{j=1}^{i}(1 - x_{j-1}) = 1$.

(2) Let $\{y_i\}_{i\geq 0}$ be a series with $0 \leq y_i \leq 1$. If $0 < y_i \leq 1$ for $0 \leq i \leq m-1$ and $y_m = 0$ for some $m \geq 1$, then $\sum_{i=1}^{m}(1 - y_i) \Pi_{j=1}^{i} y_{j-1} = y_0$.

Proof. We can easily check that both of parts (1) and (2) of the lemma hold for $m = 1, 2$.

Then, we firstly show that the part (1) holds for $m \geq 3$. By setting $x_m = 1$, we can get

$$\begin{aligned} x_0 + \sum_{i=1}^{m} x_i \prod_{j=1}^{i}(1 - x_{j-1}) &= x_0 + \sum_{i=2}^{m} x_i \prod_{j=1}^{i-1}(1 - x_{j-1}) + \prod_{j=1}^{m}(1 - x_{j-1}) \\ &= x_0 + (1 - x_0)\{x_1 + \sum_{i=2}^{m-1} x_i \prod_{j=1}^{i-1}(1 - x_j) + \prod_{j=1}^{m-1}(1 - x_j)\}. \end{aligned}$$

The part (1) of the lemma is proved if we can see that the following equation

$$x_1 + \sum_{i=2}^{m-1} x_i \prod_{j=1}^{i-1}(1 - x_j) + \prod_{j=1}^{m-1}(1 - x_j) = 1$$

holds for arbitrary $m \geq 3$. It is clear that the equation holds for $m = 3$. We assume that the following equation holds for $m = k$ as an induction hypothesis;

$$x_1 + \sum_{i=2}^{k-1} x_i \prod_{j=1}^{i-1}(1 - x_j) + \prod_{j=1}^{k-1}(1 - x_j) = 1.$$

Then, in the case of $m = k + 1$, we can get

$$x_1 + \sum_{i=2}^{k} x_i \prod_{j=1}^{i-1}(1 - x_j) + \prod_{j=1}^{k}(1 - x_j) = x_1 + (1 - x_1)\{x_2 + \sum_{i=3}^{k} x_i \prod_{j=2}^{i-1}(1 - x_j) + \prod_{j=2}^{k}(1 - x_j)\}.$$

By rewriting a literal x_{i+1} to x_i for each $i \geq 1$ and the induction hypothesis above, we can see that the formula in braces { and } is equal to 1. Thus, the part (1) holds.

We secondly show that the part (2) of the lemma holds for $m \geq 3$. By setting $y_m = 0$, we can get

$$\sum_{i=1}^{m}(1 - y_i) \prod_{j=1}^{i} y_{i-1} = \prod_{i=1}^{m} y_{i-1} + \sum_{i=1}^{m-1}(1 - y_i) \prod_{j=1}^{i} y_{i-1} = y_0\{\prod_{i=2}^{m} y_{i-1} + \sum_{i=2}^{m-1}(1 - y_i) \prod_{j=2}^{i} y_{i-1} - y_1 + 1\}.$$

The lemma is proved if we can see tat the following equation

$$\prod_{i=2}^{m} y_{i-1} + \sum_{i=2}^{m-1}(1 - y_i) \prod_{j=2}^{i} y_{i-1} - y_1 = 0$$

holds for arbitrary $m \geq 3$. It is clear that the equation holds for $m = 3$. We assume that the following equation holds for $m = k$ as an induction hypothesis.

$$\prod_{i=2}^{k} y_{i-1} + \sum_{i=2}^{k-1}(1 - y_i) \prod_{j=2}^{i} y_{i-1} - y_1 = 0$$

Then, in the case of $m = k + 1$, we can get that

$$\begin{aligned} \prod_{i=2}^{k+1} y_{i-1} + \sum_{i=2}^{k}(1 - y_i) \prod_{j=2}^{i} y_{j-1} - y_1 &= \prod_{i=2}^{k} y_{i-1} + \sum_{i=3}^{k}(1 - y_i) \prod_{j=2}^{i} y_{j-1} + (1 - y_2)y_1 - y_1 \\ &= y_1\{\prod_{i=3}^{k+1} y_{i-1} + \sum_{i=3}^{k}(1 - y_i) \prod_{j=3}^{i} y_{j-1} - y_2\}. \end{aligned}$$

By rewriting a literal y_{i+1} to y_i for each $i \geq 1$ and the induction hypothesis above, we can see that the formula in braces { and } is equal to 1. This completes the proof of the lemma. □

Let U be a set of all examples. We call a set V a training set and call a set $U - V$ an examining set. We assume that the examining set is divided to two disjoint sets, P and N.

Theorem 1

(1) Iterative Algorithm terminates when the procedure BONSAI in the algorithm gets $B(1, y_0)$ (that is, $x_0 = 1$) ($B(x_0, 0)$ (that is, $y_0 = 0$)) by taking all positive examples in U. The cost totally spent by the procedure EXPERT from the beginning of the algorithm is $|V| + |P| + (1 - y_0)|N|$ ($|V| + x_0|P| + |N|$).

(2) For each $m \geq 1$, Iterative Algorithm terminates when the procedure BONSAI in the algorithm gets $B(1, y_m)$ (that is, $x_m = 1$) or $B(x_m, 0)$ (that is, $y_m = 0$) by taking all positive examples in U. The costs totally spent by the procedure EXPERT from the beginning of the algorithm are given as follows:

$$\text{Case } B(1, y_m)\text{:} \quad |V| + |P| + \{(1 - y_0) + \sum_{i=1}^{m}(1 - y_i) \prod_{j=1}^{i} y_{i-1}\}|N|$$

$$\text{Case } B(x_m, 0)\text{:} \quad |V| + \{x_0 + \sum_{i=1}^{m} x_i \prod_{j=1}^{i}(1 - x_{j-1})\}|P| + |N|$$

Proof. It is easy to see that the part (1) of the theorem holds. We then show the proof of the part (2).

Let $A_0^P = P$ and $A_0^N = N$ and let $B_{(x_i,y_i)}$ be a decision tree and an indexing gotten by BONSAI for each $i \geq 1$. For each $i \geq 1$, we define A_i^P and A_i^N as follows:

- **A_i^P:** a set whose elements are falsely selected from A_{i-1}^P as negative elements by the procedure POSKAMO using $B_{(x_{i-1},y_{i-1})}$ and $|A_i^P| = (1 - x_{i-1})|A_{i-1}^P|$.
- **A_i^N:** a set whose elements are truly selected from A_{i-1}^N as negative elements by the procedure POSKAMO using $B_{(x_{i-1},y_{i-1})}$ and $|A_i^N| = y_{i-1}|A_{i-1}^N|$.

Then, $A_{i-1}^P - A_i^P$ is a set of elements to be truly selected by the procedure POSKAMO using $B_{(x_{i-1},y_{i-1})}$ and the number of elements in the set is $x_{i-1}|A_{i-1}^P|$. Furthermore, $A_{i-1}^N - A_i^N$ is a set of elements to be falsely selected by the procedure POSKAMO using $B_{(x_{i-1},y_{i-1})}$ and the number of elements in the set is $(1 - y_{i-1})|A_{i-1}^N|$ [2].

The procedure EXPERT divides the set $(A_{i-1}^P - A_i^P) \cup (A_{i-1}^N - A_i^N)(= K_{i-1})$ into two disjoint sets $A_{i-1}^P - A_i^P(= P_{i-1})$ and $A_{i-1}^N - A_i^N(= N_{i-1})$, and the cost spent by the procedure is $x_i|A_{i-1}^P| + (1 - y_{i-1})|A_{i-1}^N|$. Thus, the sum of costs in all steps $i = 1 \ldots n$ can be given by

$$\sum_{i=0}^{n}\{x_i|A_i^P| + (1-y_i)|A_i^N|\} = \{x_0 + \sum_{i=1}^{n} x_i \prod_{j=1}^{i}(1-x_{j-1})\}|P| + \{(1-y_0) + \sum_{i=1}^{n}(1-y_i)\prod_{j=1}^{i} y_{i-1}\}|N|.$$

We can easily see that in the case $x_m = 1$ or in the case $y_m = 0$ $(m \geq 1)$, Iterative Algorithm takes all positive examples in $U - V$. From this fact and Lemma 1, it is clear that in each case when the procedure BONSAI gets $B_{(1,x_m)}$ or $B_{(x_m,0)}$, the cost spent by the procedure EXPERT is given by the formula in Theorem 1. □

Corollary 1

(1) The lower bound of sum of costs totally spent by the procedure EXPERT is given by

- $|V| + |P|$ if and only if $x_m = 1$ for some $m \geq 0$ and $y_0 = y_1 = \cdots = y_m = 1$, or
- $|V| + |N|$ if and only if $y_m = 0$ for some $m \geq 0$ and $x_0 = x_1 = \cdots = x_m = 0$.

(2) The upper bound of sum of costs totally spent by the procedure EXPERT is $|V| + |P| + |N|$ if and only if $x_m = 1$ and $y_m = 0$ for some $m \geq 0$.

Proof. (1) Since we can easily see that the following two inequalities hold;

$$0 \leq x_0 + \sum_{i=1}^{m} x_i \prod_{j=1}^{i}(1-x_{j-1}) \leq 1 \text{ and } 0 \leq (1-y_0) + \sum_{i=1}^{m}(1-y_i)\prod_{j=1}^{i} y_{i-1} \leq 1,$$

we can get the lower and upper bounds as in the corollary.

The lower bound $|V|+|P|$ is obtained when $(1-y_0)+\sum_{i=1}^{m}(1-y_i)\prod_{j=1}^{i} y_{i-1} = 0$. It is easily seen that the equation holds if and only if the procedure BONSAI gets $B_{(1,y_m)}$ and $y_0 = y_1 = \cdots y_m = 1$. On the other hand, the lower bound $|V|+|N|$ is obtained when $x_0+\sum_{i=1}^{m} x_i \prod_{j=1}^{i}(1-x_{j-1}) = 0$. It is easily seen that the equation holds if and only if the procedure BONSAI gets $B_{(x_m,0)}$ and $x_0 = x_1 = \cdots = x_m = 0$.
(2) It is obvious that the upper bound $|V|+|P|+|N|$ is obtained if and only if $x_m = 1$ and $y_m = 0$. □

[2] $S_i = A_i^P \cup A_i^N$

3.2 Experiments

Iterative Algorithm can get 100% positive examples if the condition $x_m = 1$ or $y_m = 0$ holds for some $m \geq 0$. However, in practice, we can not know whether or not the classification rule produced by the procedure BONSAI for a training set works well even on an examining set. Thus, in the following, we will observe the relationships between the ratios of the numbers of positive examples founded by experts and the numbers of examples (including negative examples) to be read by experts in order to give an affirmative answer of this question.

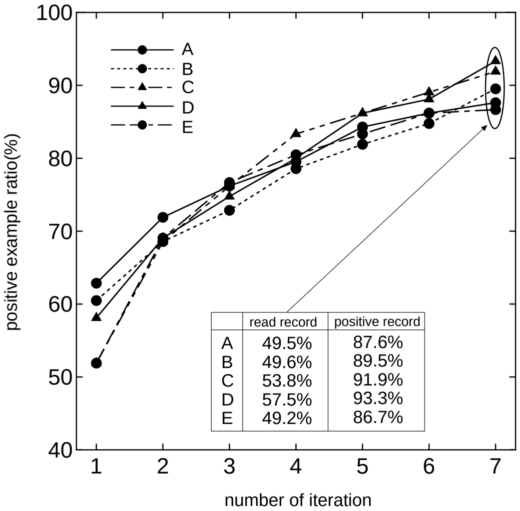
We choose randomly five pairs of sets of 20 positive examples and 50 negative examples from the rough reading image of 758 records in *Cell* as training sets. Thus, the examining set has 688 records.

We have applied BONSAI to each of these five training sets with the parameters iteration=10, max pattern length=9, indexing size=3, random seed number=10, and window size=10 initially. Since the random seed number parameter is set to 10, the procedure BONSAI gets ten rules classifying 20 positive examples and 50 negative examples. By applying the ten rules to the examining set, the procedure POSKAMO gets ten kinds sets each of which should be classified to the sets of positive examples and negative examples by the procedure EXPERT.

A problem of interest is how we should choose one from these ten kinds of sets. By our observations during the experiments, we find that the numbers of positive examples in these ten kinds of sets are not so distributed comparing with the wide range distribution of the number of these ten kinds of sets. On the other hand, it is obvious that we can not get reliable rule by BONSAI, if the training set is too small. From these two aspects, we decide that we should choose the smallest set having more than or equal to 20 records among these ten kinds of sets for the next iteration step.

We have repeated the steps of experiments seven times for each of these five training sets. During the experiments, if the number of positive or negative examples in the set gotten by the procedure POSKAMO is less than 10, we reset the window size parameter to 5.

Fig. 1 shows how the ratio of positive examples grows with the number of iterations on five training sets cases A∼E. We can see from the figure that nearly 90% positive examples are founded even if experts have read only half of whole records.



	read record	positive record
A	49.5%	87.6%
B	49.6%	89.5%
C	53.8%	91.9%
D	57.5%	93.3%
E	49.2%	86.7%

Figure 1: Performance of Iterative Method.

4 Conclusion

We have tried to extract the articles of the experts' interest from the literature database by a machine learning system. Our strategy to perform this is to express the knowledge of experts by a rough reading function and a rough reading alphabet. Furthermore, our important assumption is that even if the learning system uses only a little amount of records in the literature to produce a classification rule, the large amount of records to be remained can be classified by that rule efficiently. The result of experiment with BONSAI in 2.3 says that our technique based on these strategy and assumption works well.

Our next aim is to extract positive examples as many as possible. We notice that the records selected by BONSAI should be classified to positive P and negative examples N by experts in any case. This leads to the algorithm presented in this paper which iteratively identifies positive examples by using P and N to be classified by experts in the last step. By the experiment, we have shown that this algorithm enables us to select nearly 90% of positive examples while leaving half amount of records unread.

References

[1] Chen, H., Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms, *J. Am. Soc. Inf. Sci.*, 46(3):194–216, 1995.

[2] Chen, H., Shankaranarayanan, G., She, L., and Iyer, A., A machine learning approach to inductive query by examples: an experiment using relevance feedback, ID3, genetic algorithm, and simulated annealing, *J. Am. Soc. Inf. Sci.*, 49(8):693–705, 1998.

[3] Cortez, E.M., Park, S.C., and Kim, S., The hybrid application of an inductive learning method and a neural network for intelligent information retrieval, *Inf. Process. Manage*, 31(6):789–813, 1995.

[4] Lewis, D.D., Learning in intelligent information retrieval, *Proc. of Eighth International Workshop on Machine Learning*, 235–239, 1991.

[5] Shimozono, S., Shinohara, A., Shinohara, T., Miyano, S., Kuhara, S., and Arikawa, S., Knowledge acquisition from amino acid sequence by machine learning system BONSAI, *Trans. Inform. Process. Soc. Japan*, 35(10):2009–2018, 1994.