Selecting and Aggregating Features: Mining HTTP Log Files

Akiko AIZAWA
National Center for Science Information Systems
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
akiko@rd.nacsis.ac.jp

Abstract

In this paper, we first assume a classification hierarchy is given for features, and then calculate the comparative data losses of two reduction schemes, feature selection and feature aggregation, both are the methods for reducing the dimension of the original matrix for DSA. We also propose a new reduction procedure where the derived equations are used to decide which scheme should be employed for each feature group on the given feature hierarchy. The effectiveness of the proposed method is demonstrated through experiments using HTTP access logs and also a collection of full-text academic articles.

1 Introduction

Frequency matrices are common data representations in information systems which can be viewed as a special type of contingency tables whose cell (i, j) represents the number of occurrences of event j with object i. For this type of data, *dual scaling analysis* (DSA) [1]is known to provide an effective way to explore the underlying association structure among rows and among columns.

DSA is a statistical analysis method which adopts a mathematical technique called the singular value decomposition (SVD). In recent years, the usefulness of DSA or other SVD-based methods was demonstrated in various application domains including automatic construction of thesaurus [2], automatic indexing in information retrieval systems [3] or concept space visualization [4].

However, the computational cost prohibits the method to be applied to matrices with tens of thousands of dimensions, as is the case with URL access frequency matrices generated from HTTP log files or lexical tables generate from large textual corpus. The problem thus becomes how to reduce the dimension of the original data at the pre-processing stage, to make DSA feasible.

The issue is closely related to the feature subset selection problem in machine learning field [5] [6]. However, our problem has one interesting aspect that is not assumed in the conventional studies; a classification hierarchy is given for features, i.e. the URL-hierarchy or a thesaurus constructed by human is available. The existence of such classification hierarchy implies a possibility of reduction not only by eliminating non-significant features but also by substituting a group of features with a corresponding upper class feature. In this paper, we refer to the former type of reduction scheme by *feature selection* and the later type by *feature aggregation.*

In order to illustrate the effect of DSA and feature reduction, let us consider document by term matrix given as follows:

$$X = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \end{array} \begin{array}{c} \begin{array}{cccccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} \end{array} \\ \left[\begin{array}{cccccccccc} 1 & 3 & 4 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 5 & 20 & 54 & 2 & 0 & 0 & 3 & 5 & 0 & 0 \\ 0 & 45 & 3 & 0 & 55 & 1 & 1 & 1 & 1 & 1 \\ 3 & 9 & 5 & 2 & 2 & 0 & 0 & 0 & 0 & 1 \\ 55 & 3 & 36 & 48 & 4 & 0 & 2 & 0 & 2 & 1 \\ 1 & 3 & 1 & 1 & 1 & 56 & 34 & 33 & 21 & 3 \\ 0 & 0 & 8 & 0 & 6 & 30 & 4 & 5 & 33 & 0 \\ 1 & 0 & 3 & 0 & 0 & 0 & 12 & 18 & 19 & 18 \\ 5 & 2 & 2 & 0 & 0 & 20 & 5 & 29 & 0 & 30 \\ 0 & 2 & 0 & 2 & 1 & 10 & 5 & 9 & 9 & 0 \end{array}\right] \end{array}$$

Assume the first five terms (say $\{cat, dog, rabbit, tiger, bear\}$) are categorized into one group ($animals$) and the rest (say $\{car, train, bus, ship, truck\}$) to another ($vehicles$) (Fig. 1).

Aggregating these two groups of terms reduces the dimension of the matrix from 10 to 2, generating:

$$X_1 = \left[\begin{array}{cccccccccc} 9 & 81 & 103 & 21 & 146 & 7 & 14 & 4 & 9 & 5 \\ 1 & 8 & 5 & 1 & 5 & 147 & 72 & 67 & 114 & 33 \end{array}\right]^T,$$

where each cell represents the total frequency of terms in the same group. On the other hand, selecting the top two most frequent terms across documents yields:

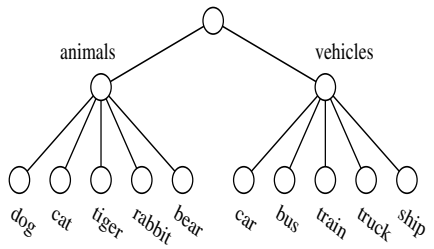


Figure 1: Example of Classification Hierarchy of Terms.

$$X_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 56 & 30 & 0 & 20 & 10 \\ 1 & 5 & 1 & 0 & 0 & 33 & 5 & 18 & 59 & 9 \end{bmatrix}^T.$$

Also, selecting the most frequent terms from each group produces:

$$X_3 = \begin{bmatrix} 4 & 54 & 3 & 5 & 36 & 1 & 8 & 3 & 2 & 0 \\ 1 & 5 & 1 & 0 & 0 & 33 & 5 & 18 & 59 & 9 \end{bmatrix}^T.$$

Table 1 shows the results of UPGMA clustering [7] where 10 documents are categorized into two groups using the distance calculated either from X, X_1, X_2 or X_3 either before or after dual scaling. It can be seen that (i)without DSA, neither of the methods able to identify expected clustering result $\{d_1,d_2,d_3,d_4,d_5\}$ $\{d_6,d_7,d_8,d_9,d_{10}\}$, and that (ii)with DSA, X and X_1 produce the same expected result while X_2 and X_3 still do not. Based on this, we can conclude that DSA with term aggregation works more successfully than term selection in this case. But this obviously depends on the frequency patterns of features in the original matrix and there may be cases where feature selection is more advantageous.

Table 1: Result of UPGMA clustering.

(a) before dual scaling

X :	$\{d_1,d_2,d_3,d_4,d_6,d_7,d_8,d_9,d_{10}\}$ $\{d_5\}$
X_1 :	$\{d_1,d_2,d_3,d_4,d_5,d_7,d_8,d_{10}\}$ $\{d_6,d_9\}$
X_2 :	$\{d_1,d_2,d_3,d_4,d_5,d_7,d_8,d_{10}\}$ $\{d_6,d_9\}$
X_3 :	$\{d_1,d_3,d_4,d_6,d_7,d_8,d_9,d_{10}\}$ $\{d_2,d_5\}$

(b) after dual scaling

X :	$\{d_1,d_2,d_3,d_4,d_5\}$ $\{d_6,d_7,d_8,d_9,d_{10}\}$
X_1 :	$\{d_1,d_2,d_3,d_4,d_5\}$ $\{d_6,d_7,d_8,d_9,d_{10}\}$
X_2 :	$\{d_1,d_2,d_8\}$ $\{d_3,d_4,d_5,d_6,d_7,d_8,d_{10}\}$
X_3 :	$\{d_1,d_2,d_3,d_4,d_5,d_7\}$ $\{d_6,d_8,d_9,d_{10}\}$

This paper aims at providing mathematically justified criteria to decide which reduction schemes, feature selection or feature aggregation, should be employed for a given feature group. This enables us to apply the same principle even for HTTP log data where the given URL hierarchy does not necessarily reflect the semantic structure of the target data.

The rest of the paper is organized as follows: Section 2 gives a mathematical description of the problem. Section 3 first provides equations for losses for selecting and aggregating features and then a new procedure is proposed to combine these two schemes so that the total loss value is minimized. Section 4 and 5 show the results of experimental studies using real-scale HTTP log files and also full-text survey articles. Section 6 includes some discussions and future issues.

2 Problem Description

2.1 Dual Scaling Analysis

Let X be a $m \times n$ matrix with each cell x_{ij} representing the frequency of event j $(1 \leq j \leq n)$ with object i $(1 \leq i \leq m)$. DSA can be viewed as a variation of principle axes methods which determines the optimal scoring for rows and columns so that the correlation between these two is maximized. Mathematically, DSA adopts a general matrix conversion procedure called *singular value decomposition*, after transforming X to Y given by:

$$y_{ij} = \frac{x_{ij}}{\sqrt{x_{i\cdot}}\sqrt{x_{\cdot j}}} - \frac{\sqrt{x_{i\cdot}}\sqrt{x_{\cdot j}}}{x_t}, \qquad (1)$$

with $x_{i\cdot}$ and $x_{\cdot j}$ being the sums of row i and column j respectively, and x_t the sum of all x_{ij}. The computation complexity of DSA is essentially the one needed for matrix inversion.

The above transformation, together with the nature of singular value decomposition, guarantees that adding or eliminating columns from X does not affect DSA results for the remaining columns so far as the sum of rows $(x_{i\cdot})$ and the total sum (x_t) are maintained. This property of DSA, which we utilize later in our theoretical development, is called *the principle of equivalent partitioning*.

2.2 Classification Hierarchy of Features

The representation of a *feature classification hierarchy* in this paper is simply a tree whose leaves correspond to *basic features*, features actually observed in the target data, and nodes to *feature groups*, abstract features defined by a group of basic features. A basic feature itself can be viewed as a special feature group with a single group member. In the following, we use the lower case f for basic features and the upper case F for feature groups. For example, in Figure 2, F_1 consists of features $\{f_3, f_4, f_5, f_6, f_7\}$, F_5 consists of a single feature $\{f_5\}$, and so on.

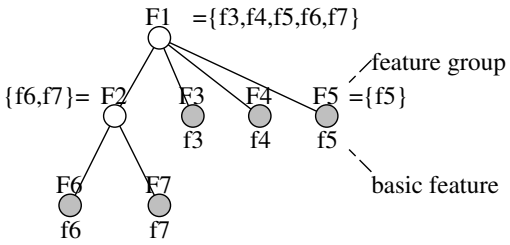


Figure 2: Classification hierarchy of features.

It should be noted that a classification hierarchy represents only *a priori* knowledge about possible resemblance of features and whether features categorized into the same group actually show similar patterns remains to be tested on the target data.

2.3 Feature Selection and Aggregation

Let $\vec{x}_{(j)}$ represent m dimensional vector corresponding to feature f_j, i.e. $\vec{x}_{(j)} = (x_{1j}, \cdots, x_{mj})^T$, the j'th column of X. Given a feature group F of size k, *feature selection* refers to the case where k_0 $(\leq k)$ of the k features is selected as representatives of F while the other $(k - k_0)$ features are discarded. On the other hand, *feature aggregation* refers to the case where the k feature vectors in T are replaced with a single newly generated vector $\vec{x}$ such that

$$\vec{x} = (\sum_{j=1}^{k} x_{1j}, \cdots, \sum_{j=1}^{k} x_{mj})^T. \qquad (2)$$

Thus in feature aggregation, only the total frequency of k features is maintained after reduction while all other information is discarded. Here, we assume feature aggregation occurs only at once and do not consider the situation when a subset of the group are aggregated.

3 Theoretical Development

3.1 Loss of Feature Selection

Assume features $\{f_{k+1}, \cdots, f_n\}$ are selected from the original matrix X, generating a reduced matrix $X' = (\vec{x}_{(k+1)}, \cdots, \vec{x}_{(n)})$. From the principle of equivalent partitioning, the reduction is, in terms of DSA, equivalent to substituting each of the eliminated features $\{f_1, \cdots, f_k\}$ with an identical vector $\vec{x}' = \{x'_1, \cdots, x'_m\}$ given by:

$$x'_i = \frac{1}{\alpha} \sum_{j=k+1}^{m} x_{ij}, \ \alpha = \frac{k \sum_{i=1}^{m} \sum_{j=k+1}^{n} x_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{k} x_{ij}}. \qquad (3)$$

Note that in the above equation, the sum of each rows and the total frequency are maintained. Using $\vec{x}'$, we define the loss of selecting $\{f_{k+1}, \cdots, f_n\}$, i.e. the loss of eliminating $\{f_1, \cdots, f_k\}$ as follows:

$$\begin{aligned} L_{sel}(f_{k+1}, \cdots, f_n) &= L_{eli}(f_1, \cdots, f_k) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{k} (x_{ij} - x'_i)^2. \end{aligned} \qquad (4)$$

Though there may exist other possible equivalent transformations for X', Eq.(3) is used as a basis in this paper because the calculation of the optimal transformation is obviously infeasible and the equation has good rationale as it approximates the missing features by the average of the remaining ones.

3.2 Loss of Feature Aggregation

Loss of feature aggregation on the other hand is calculated as follows: From the principle of equivalent partitioning, aggregating features $\{f_1, \cdots, f_k\}$ to a single feature given by Eq.(2) is, in terms of DSA, equivalent to substituting all the k features with identical feature vector $\vec{x}'' = \{x''_1, \cdots, x''_m\}$ such that:

$$x''_i = \frac{1}{k} \sum_{j=1}^{k} x_{ij}. \qquad (5)$$

Using $\vec{x}''$, we define the loss for feature aggregation as follows:

$$L_{agg}(f_1, \cdots, f_k) = \sum_{i=1}^{m} \sum_{j=1}^{k} (x_{ij} - x''_{ij})^2. \qquad (6)$$

When the feature group consists of only a single feature $(k = 1)$, the loss value equals zero.

Again, there may exist other better equivalent transformations in terms of the loss value, but we use Eq.(5) as our basis because of the simplicity of the calculation and semantic clearness; when aggregated, all the features in the group are averaged. The effectiveness of such approximation is examined through experiments in the next section.

Figure 3 depicts the difference of feature selection and aggregation where features $\{f_1, \cdots, f_k\}$ are reduced into a single feature.

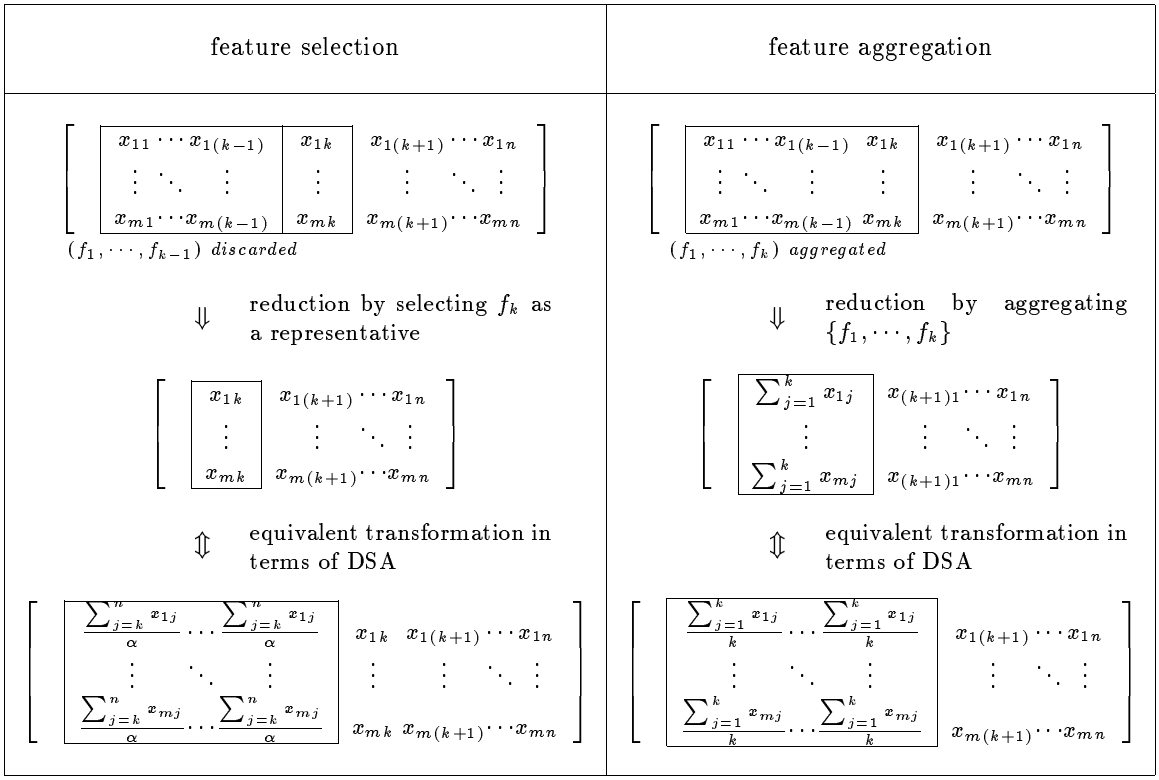


Figure 3: Reducing $\{f_1, \cdots, f_k\}$ into a single feature by selection and aggregation

3.3 Total Loss Calculation

Let $\mathcal{F}$ be a set of feature groups obtained by applying selection and aggregation. Every basic feature in X either (i) belongs to one and exactly one of the selected feature groups (including the feature itself), or (ii)otherwise. Then, the overall loss can be calculated as follows:

$$L_{total}(\mathcal{F}) = L_{sel}(\{f_j \mid f_j \in F, F \in \mathcal{F}\}) + \sum_{F \in \mathcal{F}} L_{agg}(\{f_j \mid f_j \in F\}). \quad (7)$$

3.4 Procedure for Feature Reduction

Assume n_0 be a specified number of features for reduction. The procedure we propose here is a sort of greedy algorithm which minimizes the total loss given by Eq. (7). While its execution, it maintains a list of sorted feature groups and assumes the top n_0 feature groups survive for selection. In our current implementation, the loss is optimized by the following steps :

- **step(1):**

  Initially, sort all the basic features in X according to their elimination loss given by Eq. (4). Select the top n_0 basic features and calculate the total loss of the selection.

- **step(2):**

  Chose one of the upper feature groups which does not appears on the list but has its members already listed. Calculate the total loss assuming its members are aggregated.

- **step(3):**

  If the loss value improves, eliminate all the group members and add the upper feature to the sorted list.

- **step(4):**

  Repeat step (2) and (3) until all the upper groups are tested.

Figure 4 illustrates the proposed reduction procedure.

Note that the procedure does not require much computation cost; the loss calculation at Step(2) is in proportional to the size of the matrix, and the iteration of Step(2)-(3) is no more than the number of non-basic features, which is relatively small as will be seen in Table 3.

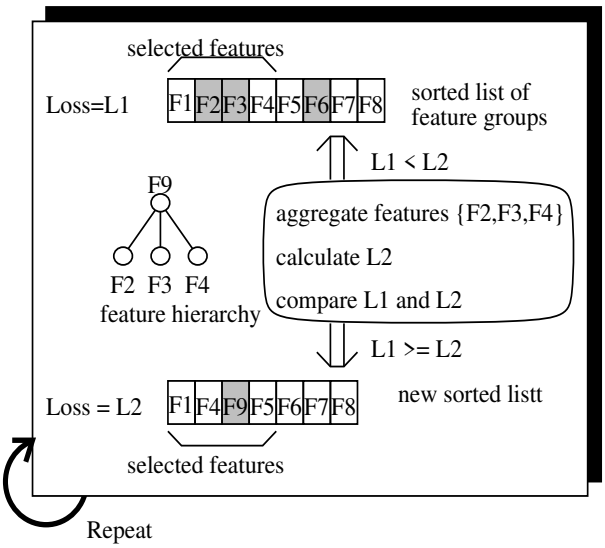


Figure 4: Proposed feature reduction procedure.

4 Experimental Results using HTTP Log data

4.1 Target Data

The information source used in our first experiment is HTTP access log obtained at well-known proxy server site in Japan. We select 12 client domains as objects and 2,400 URL's as attributes and figure out the total access count of each client to each URL to extract the original frequency matrix. The feature hierarchy is simply a directory of URL's derived from their notations (Fig. 5).

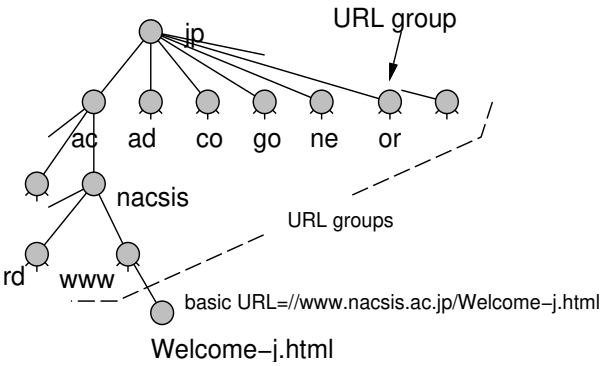


Figure 5: Example of URL hierarchy.

Although we use only limited size of data here, this is just for the evaluation purpose. In [13] [14], variations of the proposed method (with less computation overhead) are applied to HTTP log data with 460,000 URLs.

4.2 Method for Evaluation

The extracted frequency matrix is first preprocessed using either of (i)without normalization (RLOG), (ii)d_{ij} is set to 1 if there is any access from client(i) to URL(j) and 0 otherwise (UNIQ), (iii) each row is normalized to make the row's total equal to 1.0, i.e. $d_{ij} = f_{ij} / \sum_{j=1}^{2400} f_{ij}$ (FREQ).

Feature reduction is then performed after normalization to reduce the dimension of the matrix from 2,400 to 60. We adapt two reduction schemes; (a)the one we propose (*LRED, reduction-by-loss*), and (b)a naive selection where 60 most frequent URL's are selected (*FSEL, selection-by-frequency*). After the feature reduction, DSA is applied to each reduced matrix to calculate the similarities for every client pairs. Also as reference data, we directly apply DSA to the original (normalized but not reduced) matrix and calculate the similarities between the client pairs.

The performance of LRED and FSEL is evaluated by the correlation between these similarity value sets: *Pearson's product moment correlation coefficient* for values and *Spearman's rank correlation coefficient* for ranks. For both coefficients, the value 1.0 shows the most fittest.

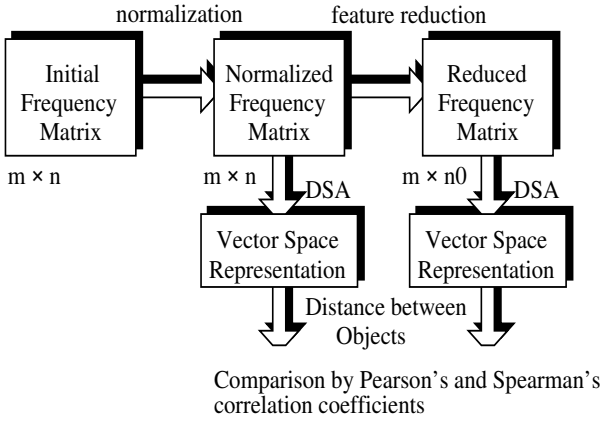


Figure 6: Evaluation method.

4.3 Results

The result is summarized in Table 2. For all the cases, LRED outperforms FSEL and establishes fairy good correlation with the reference data, which is sufficient for DSA. FSEL does not work with UNIQ since all the features it selects are **1**'s. For one-week data, the correlation becomes worse in most cases while LRED applied with FREQ still maintains good performance. This may be because when a client shows relatively low activities throughout the week, most information about the

client is lost after reduction. Such information loss can be prevented with FREQ since the values are normalized for each client.

Table 2: Performance comparison of LRED and FSEL.

(a) 20-weeks data

		Pearson	Spearman
RLOG	FSEL	0.87	0.74
	LRED	0.90	0.92
UNIQ	FSEL	–	–
	LRED	1.00	0.89
FREQ	FSEL	0.79	0.83
	LRED	0.92	0.94

(b) one-week data (20 weeks average)

		Pearson	Spearman
RLOG	FSEL	0.35	0.15
	LRED	0.53	0.39
UNIQ	FSEL	–	–
	LRED	0.83	0.83
FREQ	FSEL	0.73	0.77
	LRED	0.93	0.93

5 Experimental Results using Text Data

5.1 Target Data

The information sources we use in our second experiment are the survey articles in *Journal of Japanese Society for Artificial Intelligence* published during 1986-1995. The total 304 articles exist as full-text, digitized by OCR under NACSIS Corpus Project [8] [9].

We compare three different term hierarchies extracted from well known terminological dictionaries: (i)the one obtained from a technical term dictionary provided by Japanese Society for Artificial Intelligence [11] (*JSAI*), (ii)the one obtained from Technical Terms Dictionary of EDR Electronic Dictionary Version 1.5 [10] (*EDR*), and (iii)(i) expanded with synonymous term corpus automatically generated [12] (*JSAI**). Since JSAI is constructed by the society itself, we can expect it is the most suitable to characterize the target document set. On the other hand, EDR contains a large number of terms relating information processing in general and thus may not be so much focused on the subject domain. In JSAI*, each term in JSAI is expanded with a group of corresponding synonyms including various notation variations.

Technical terms defined in each dictionary constitute a set of basic terms. The occurrence of each basic term is counted for each article to make the initial frequency matrix $F = \{f_{ij}\}$. While counting, only simple normalization rules such as ignoring lower and upper case difference are applied for JSAI and EDR while in JSAI*, the occurrence counter is increased every time when one of the synonymous terms appears. Thus, the process can be considered as *enforced* aggregation where term aggregation is always applied at the bottom level. The difference between JSAI and JSAI* is, for example, in JSAI, the Japanese word "metafa(*metaphor* in English)" and English word "metaphor" (Figure 7) are considered different and thus assigned different frequencies. On the other hand, these two as well as other terms as "metaphorical expression", "in-yu(*metaphor*)", "hiyu(*metaphor*)" are considered to be the same in JSAI* and thus assigned the same frequency.

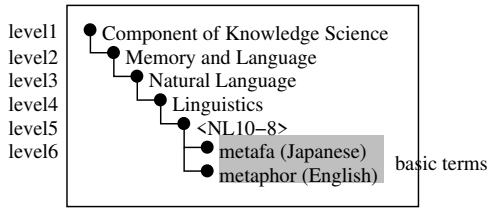


Figure 7: Example of terms defined by JSAI.

Table 3 shows the comparison of JSAI, EDR and JSAI*. It can be seen that EDR contains wide range of terms with less occurrences. Also, the average frequency per term for JSAI* is greater than the one for JSAI because of the term expansion effect. For reference, values without such expansion are also shown in the table by figures with brackets.

Table 3: Comparison of JSAI, EDR and JSAI*.

	JSAI	EDR	JSAI*
total number of basic terms	2,098	195,179	2,098 (10,265)
total number of term groups †	3,329	197,307	3,329 (13,594)
average depth of term hierarchy	6.0	7.7	6.0 (7.0)
number of terms actually observed	483	12,026	1,104 (3,196)
average frequency per observed term	0.56	0.25	1.29 (0.45)

† including basic terms

5.2 Method for Evaluation

Once the initial frequency matrix is obtained, each row of the matrix is normalized so that the total frequency for each document equals 1.0, i.e. $x_{ij} = f_{ij}/\sum_{j=1}^{n} f_{ij}$. Term reduction is then performed to reduce the size of the matrix to a specified number.

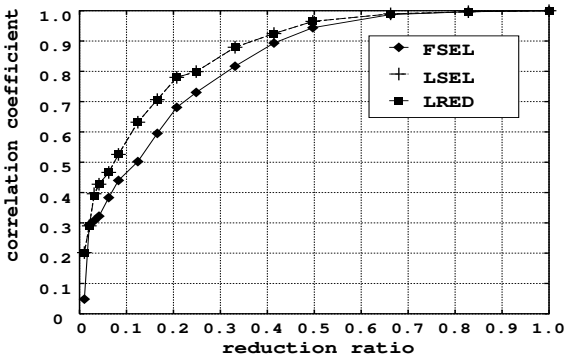
We compare the proposed reduction scheme (*LRED, reduction-by-loss*) with a simple selection without aggregation where basic terms are evaluated and selected using elimination loss given by Eq.(4) (*LSEL, selection-by-loss*), and also with a naive but commonly practiced method that selects the top frequent terms (*FSEL, selection-by-frequency*). We have also tested selection using cross entropy, but simple frequency worked better in our case. This may be because the authorized terminological sets do not include meaningless words such as stop words which cross entropy method can be the most effectively applied to.

After the reduction, DSA is applied to each reduced matrix to calculate similarities between every document pair. As reference, we also directly apply DSA to the original normalized (but not reduced) matrix and calculate similarities. The performance of LRED, LSEL and FSEL is evaluated by Pearson's product moment correlation coefficient and Spearman's rank correlation coefficient.

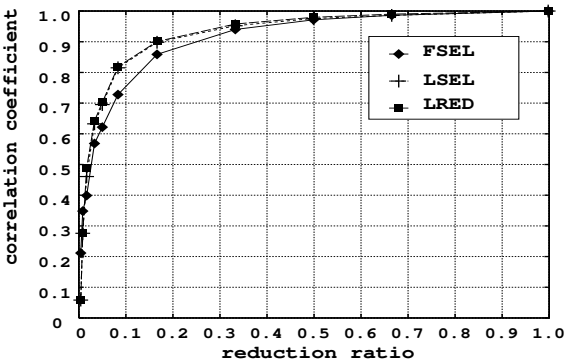
5.3 Results

Figure 8 compares the performance of FSEL, LSEL and LRED for each of JSAI, EDR, and JSAI*. The horizontal axis represents the ratio of selected terms to the total terms actually observed, and the vertical axis to the corresponding values of rank correlations. Results of distance correlations are omitted since these two showed exactly the same tendency.

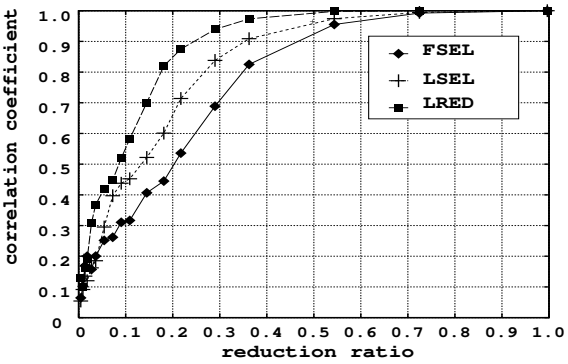
The experiment shows that LRED and LSEL outperform FSEL in all cases, justifying the use of the loss equation as a selection criteria. Also, LRED is better than LSEL, demonstrating that selection by loss and term aggregation both contribute to the performance improvement. Next, comparison across different term hierarchies suggests the aggregation effect depends on the kind of thesaurus adopted. Only small improvement is observed for JSAI and EDR since they do not contain much redundancy as JSAI*. Especially for JSAI, only several terms are aggregated which means most of the improvement is obtained by using the proposed loss equation for term selection. The result for JSAI* shows that 'enforced' aggregation to some extent works well. This can be explained as follows: the reduction scheme we have developed totally depends on statistical characteristics of terms, and thus can hardly deal with synonymous problem where the same concept is referred to with different terms, due to the language-difference, the use of acronyms and so on.



(a) performance of JSAI



(b) performance of EDR



(c) performance of JSAI*

Figure 8: Comparison of FSEL, LSEL and LRED for JSAI, EDR and JSAI*.

6 Conclusions

In this paper, we mathematically define the loss of two feature reduction schemes, feature selection and feature aggregation, and propose a method to combine these two assuming a feature hierarchy is given.

When the size of the feature dimension is large, conventional methods usually evaluate the significance of each feature independently at the preprocessing stage. On the other hand, the proposed method evaluates the dependencies of features in the same (and pre-determined) feature group by comparing the loss of different reduction schemes. Therefore, it can be expected that the method can effectively aggregate features if the given hierarchy represents meaningful clusters of features.

The same effect can also be expected with DSA itself, where the dependencies of all the features are considered in this case. However, except for the expensive computational cost, DSA has one problem inherent to principle axes methods: that is, the interpretation of the generated features (principle axes) is often difficult. Since the proposed method simply substitutes a group of features with a well-defined upper class feature, the method is sometimes preferable than DSA especially when is used as pre-processing filter for other data mining applications.

The objective of feature reduction in this paper is to maintain the information in the original data as much as possible. However, our experiments suggest that the proposed reduction scheme can also be used to overcome data sparseness problem, which remains for future study. Also, in our present study, the reduction scheme is applied only to the columns of the original matrix. Since DSA maintains duality between rows and columns, we look in to the possibility of extending the result to two-dimensional data reduction.

References

[1] Nishisato,S.: "Analysis of Categorical Data: Dual Scaling and It's Applications, " Asakura-Shoten (1982).

[2] Lebart,L., Salem,A. and Berry,L.: "Exploring Textual Data," Kluwer Academic Publishers (1998).

[3] Deerwester,S., Dumais,S.T., Furnas,G.W., Landauer,T.K. and Harshman,R. : "Indexing by Latent Semantic Analysis," Journal of the American Society of Information Science, Vol,41, No.6, pp.391-407 (1990).

[4] Sugimoto,M., Hori,K. and Ohsuga,S.: "A System for Visualizing Viewpoints and Its Application to Intelligent Activity Support," *IEEE Trans. System, Man and Cybernetics*, Vol.28C, No.1, pp.124-136 (1998).

[5] Mladenic,D.: "Feature Subset Selection in Text-Learning," 10th European Conference on Machine Learning (1998).

[6] Yang,Y. and Pedersen,J.O.: "A Comparative Study on Feature Selection in Text Categorization," Proc. of the 14th International Conference on Machine Learning, p.412-420 (1997).

[7] Kaufman,L. and Rousseeuw,P.J.: "Finding Groups in DATA," Wiley-Interscience (1990).

[8] Aihara,K. and Takasu,A.: "Domain Visualization Based on Authorized Documents," proceedings of the fourth International Conference on Information Systems, Analysis and Synthesis, p.391-398 (1998).

[9] Kageura,K., Koyama,T., Yoshioka,M., Takasu,A., Nozue,T. and Tsuji,K.: "NACSIS Corpus Project for IR and Terminological Research," NLPRS'97, p.493-496 (1997).

[10] EDR: "EDR Electronic Dictionary Version 1.5 Technical Guide," TR2-007, Japan Electronic Dictionary Research Institute (1996).

[11] Japan Society for Artificial Intelligence (ed.): "Jinko Chinou Handobukku (Handbook of Artificial Intelligence)," Ohm-sha (1990).

[12] Aizawa,A. and Kageura,K.: "An Approach to the Automatic Generation of Multilingual Keyword Clusters," COMPTERM'98 (1998).

[13] Aizawa,A.: "Analysis of Internet Domains by Extracting Information from HTTP Log files," Journal of IEICE, Vol.J81-DI, 1988-11 (1998).

[14] Aizawa, A.: "Reducing the Dimensions of Attributes by Selection and Aggregation (Abstract)," First International Conference, D-S'98 (Lecture Notes in Artificial Intelligence 1532), Springer, p.417-418 (1998).

[15] Aizawa, A.: "A Method for Dimension Reduction for Dual Scaling Analysis by Selecting and Aggregating Features," the Special Interest Group Notes of IPSJ, FI-52-4, (1998).