Formal Concept Analysis and Rough Set Theory in Clustering

Ho Tu Bao
Japan Advanced Institute of Science and Technology, Japan
National Institute of Information Technology, Vietnam

Abstract. This paper is concerned with the fundamental role of two mathematical theories in some clustering problems. Formal concept analysis provides the algebraic structure and properties of possible concepts from a given context, and rough set theory provides a mathematical tool to deal with imprecise and incomplete data. Based on these theories, we developed models and algorithms for solving three clustering problems: conceptual clustering, approximate conceptual clustering, and text clustering.

1 Formal Concept Analysis and Rough Set Theory

A theory of concept lattices has been studied under the name *formal concept analysis* (FCA) by Wille and his colleagues [1, 11]. Considers a *context* as a triple $(\mathcal{O}, \mathcal{D}, \mathcal{R})$ where $\mathcal{O}$ be a set of objects, $\mathcal{D}$ be a set of primitive descriptors and $\mathcal{R}$ be a binary relation between $\mathcal{O}$ and $\mathcal{D}$, i.e., $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{D}$ and $(o, d) \in \mathcal{R}$ is understood as the fact that object o has the descriptor d. For any object subset $X \subseteq \mathcal{O}$, the largest tuple common to all objects in X is denoted by $\lambda(X)$. For any tuple $S \in \mathcal{T}$, the set of all objects satisfying S is denoted by $\rho(S)$. A tuple S is *closed* if $\lambda(\rho(S)) = S$. Formally, a *concept* C in the classical view is a pair (X, S), $X \subseteq \mathcal{O}$ and $S \subseteq \mathcal{T}$, satisfying $\rho(S) = X$ and $\lambda(X) = S$. X and S are called *extent* and *intent* of C, respectively. Concept (X_2, S_2) is a *subconcept* of concept (X_1, S_1) if $X_2 \subseteq X_1$ which is equivalent to $S_2 \supseteq S_1$, and (X_1, S_1) is then a *superconcept* of (X_2, S_2).

It was shown that λ and ρ define a Galois connection between the power sets $\wp(\mathcal{O})$ and $\wp(\mathcal{D})$, i.e., they are two order-reversing one-to-one operators. As a consequence, the following properties hold which will be exploited in the learning process:

$$\text{if} \quad S_1 \subseteq S_2 \quad \text{then} \quad \rho(S_1) \supseteq \rho(S_2) \quad \text{and} \quad \lambda\rho(S_1) \subseteq \lambda\rho(S_2)$$
$$\text{if} \quad X_1 \subseteq X_2 \quad \text{then} \quad \lambda(X_1) \supseteq \lambda(X_2) \quad \text{and} \quad \rho\lambda(X_1) \subseteq \rho\lambda(X_2)$$
$$S \subseteq \lambda\rho(S), \quad X \subseteq \rho\lambda(X)$$
$$\rho\lambda\rho = \rho, \quad \lambda\rho\lambda = \lambda, \quad \lambda\rho(\lambda\rho(S)) = \lambda\rho(S)$$
$$\rho(\bigcup_j S_j) = \bigcap_j \rho(S_j), \quad \lambda(\bigcup_j X_j) = \bigcap_j \lambda(X_j)$$

The basic theorem in formal concept analysis [11] states that the set of all possible concepts from a context $(\mathcal{O}, \mathcal{D}, \mathcal{R})$ is a *complete lattice*[1] $\mathcal{L}$, called Galois lattice, in which infimum and supremum can be described as follows:

$$\bigwedge_{t \in T} (X_t, S_t) = (\bigcap_{t \in T} X_t, \lambda\rho(\bigcup_{t \in T} S_t)) \tag{1}$$

$$\bigvee_{t \in T} (X_t, S_t) = (\rho\lambda(\bigcup_{t \in T} X_t), \bigcap_{t \in T} S_t) \tag{2}$$

Rough set theory, a mathematical tool to deal with uncertainty introduced by Pawlak in early 1980s [10]. The starting point of this theory is the assumption that our "view" on elements of a set of objects $\mathcal{O}$ depends on some equivalence relation E on $\mathcal{O}$. An *approximation space* is a pair $(\mathcal{O}, E)$ consisting of $\mathcal{O}$ and an equivalence relation $E \subseteq \mathcal{O} \times \mathcal{O}$.

The key notion of the rough set theory is the *lower* and *upper approximations* of any subset $X \subseteq \mathcal{O}$ which consist of all objects *surely* and *possibly* belonging to X, respectively. The lower approximation $E_*(X)$ and the upper approximation $E^*(X)$ are defined by

$$E_*(X) = \{o \in \mathcal{O} : [o]_E \subseteq X\} \tag{3}$$

$$E^*(X) = \{o \in \mathcal{O} : [o]_E \cap X \neq \emptyset\} \tag{4}$$

where $[o]_E$ denotes the equivalence class of objects indiscernible with o with respect to the equivalence relation E.

2 FCA-based Conceptual Clustering

Conceptual clustering concerns mainly with symbolic data [9]. It does simultaneously two tasks: (i) *hierarchical clustering* (i.e., finding a hierarchy of useful subsets of unlabelled instances); and (ii) *characterization* (i.e., finding an intensional definition for each of these instance subsets). An important feature of conceptual clustering is that a partitioning of data is viewed as

[1] A lattice $\mathcal{L}$ is complete when each of its subsetf X has a least upper bound and a greatest lower bound in $\mathcal{L}$.

Table 1: Scheme of OSHAM conceptual clustering

Input	concept hierarchy H and an existing splittable concept C_k.
Result	H formed gradually.
Top-level	call OSHAM(root concept, $\emptyset$).

1. While C_k is still splittable, find a new subconcept of it that corresponds to the hypothesis minimizing the quality function $q(C_k)$ among η hypotheses generated by the following steps
   (a) Find a "good" attribute-value pair concerning the best cover of C_k.
   (b) Find a closed attribute-value subset S containing this attribute-value pair.
   (c) Form a subconcept C_{k_i} with the intent is S.
   (d) Evaluate the quality function with the new hypothesized subconcept.
   Form intersecting concepts corresponding to intersections of the extent of the new concept with the extent of existing concepts excluding its superconcepts.
2. If one of the following conditions holds then C_k is considered as unsplittable
   (a) There exist not any closed proper feature subset.
   (b) The local instances set C_k^r is too small.
   (c) The local instances set C_k^r is homogeneous enough.
3. Apply recursively the procedure to concepts generated in step 1.

'good' if and only if each cluster has a 'good' conceptual interpretation. In this sense, FCA is a good tool for conceptual clustering as it formalizes the duality between objects and their properties by Galois connections. Based on FCA, we have developed a conceptual clustering method OSHAM with some additional components to the concept representation by extent and intent. The key idea here to enrich the concept representation in FCA by adding several components based on the probabilistic and exemplar views on concepts that allow dealing better with typical or unclear cases in the region boundaries. The conceptual clustering method OSHAM to form a concept hierarchy in the framework of concept lattices is introduced and described in [2]. OSHAM searches to extract a good concept hierarchy by exploiting the structure of Galois lattice of concepts as the hypothesis space.

Instead of characterizing a concept only by its intent and extent, OSHAM represents each concept C_k in a concept hierarchy $\mathcal{H}$ by a 10-tuple

$$< l(C_k), f(C_k), s(C_k), i(C_k), e(C_k), d(C_k), p(C_k), d(C_k^r), p(C_k^r|C_k), q(C_k) > \quad (5)$$

where

- $l(C_k)$ is the level of C_k in $\mathcal{H}$;
- $f(C_k)$ is the list of direct superconcepts of C_k;
- $s(C_k)$ is the list of direct subconcepts of C_k;
- $i(C_k)$ is the intent of C_k (set of all common properties of instances of C_k);
- $e(C_k)$ is the extent of C_k (set of all instances satisfying properties of $i(C_k)$);
- $d(C_k)$ is the dispersion between instances of C_k;
- $p(C_k)$ is the occurrence probability of C_k;
- $d(C_k^r)$ is the dispersion of local instances of C_k which are not classified into subconcepts of C_k;
- $p(C_k^r|C_k)$ is the conditional probability of these unclassified instances of C_k;
- $q(C_k)$ is the quality estimation of splitting C_k into subconcepts C_{k_i}.

Table 1 represents the essential idea of algorithm OSHAM that allows discovering both disjoint and overlapping concepts depending on the user's interests by refining the condition 1.(a) and the intersection operation. In short, OSHAM combines the concept intent, hierarchical structure information, probabilistic estimations and the nearest neighbors of unknown instances. A experimental comparative evaluation of OSHAM is given in [2].

3 Approximate Conceptual Clustering

Kent[7] has pointed out common features between formal concept analysis and rough set theory, and formulated the *rough concept analysis* (RCA). For the sake of simplicity, we restrict ourselves here to present the basic idea of presenting approximate concepts in case of binary attributes where $\mathcal{D}$ is identical to the set $\mathcal{A}$ of all attributes a. Saying that a given formal context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ is not obtained completely and precisely means that the relation $\mathcal{R}$ is incomplete and imprecise. Let $(\mathcal{O}, E)$ be any approximation space on objects $\mathcal{O}$, we wish to approximate $\mathcal{R}$ in terms of E. The lower approximation $\mathcal{R}_{*E}$ and the upper approximation $\mathcal{R}^{*E}$ of $\mathcal{R}$ w.r.t. E can be defined element-wise as

$$\mathcal{R}_{*E}a = E_*(\mathcal{R}a) = \{o \in \mathcal{O} \mid [o]_E \subseteq \mathcal{R}a\} \tag{6}$$

$$\mathcal{R}^{*E}a = E^*(\mathcal{R}a) = \{o \in \mathcal{O} \mid [o]_E \cap \mathcal{R}a \neq \emptyset\} \tag{7}$$

The formal context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ can be then roughly approximated by two lower and upper formal contexts $(\mathcal{O}, \mathcal{A}, \mathcal{R}_{*E})$ and $(\mathcal{O}, \mathcal{A}, \mathcal{R}^{*E})$. These approximate contexts can be intuitively viewed as "truncated" and "filled up" contexts with respect to the equivalence relation E. Two formal context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ and $(\mathcal{O}, \mathcal{A}, \mathcal{R}')$ are *E-roughly equal* if they have the same lower and upper formal contexts, i.e., $\mathcal{R}_{*E} \equiv \mathcal{R}'_{*E}$ and $\mathcal{R}^{*E} \equiv \mathcal{R}'^{*E}$. A *rough formal context* in $(\mathcal{O}, E)$ is a collection of formal contexts of object set $\mathcal{O}$ and attribute set $\mathcal{A}$ which have the same lower and upper formal contexts (roughly equal formal contexts).

The rough extent of an attribute subset $S \subseteq \mathcal{A}$ w.r.t. $\mathcal{R}_{*E}$ and $\mathcal{R}^{*E}$ are defined as

$$\rho(S_{*E}) = \bigcap_{a \in S} \mathcal{R}_{*E}a \qquad \rho(S^{*E}) = \bigcap_{a \in S} \mathcal{R}^{*E}a \tag{8}$$

Now, any formal concept $(X, S) \in \mathcal{L}(\mathcal{O}, \mathcal{A}, \mathcal{R})$ can be approximated by $\mathcal{R}_{*E}$ and $\mathcal{R}^{*E}$. The *lower* and *upper E-approximation* of (X, S) are defined as

$$(X, S)_{*E} = (\rho(S_{*E}), \lambda\rho(S_{*E})) \in \mathcal{L}(\mathcal{O}, \mathcal{A}, \mathcal{R}_{*E}) \tag{9}$$

$$(X, S)^{*E} = (\rho(S^{*E}), \lambda\rho(S^{*E})) \in \mathcal{L}(\mathcal{O}, \mathcal{A}, \mathcal{R}^{*E}) \tag{10}$$

A *rough concept* of a formal concept $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ in $(\mathcal{O}, E)$ is the collection of concepts which have the same lower and upper E-approximations (roughly equal concepts). Note that approximate contexts of $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ in $(\mathcal{O}, E)$ vary according to the equivalence relation E. In [3] we introduce algorithm A-OSHAM for learning approximate concepts in the framework of rough concept analysis. Essentially, A-OSHAM induces a concept hierarchy in which each induced concept is associated with a pair of its lower and upper approximations. A-OSHAM generates concepts with their approximations recursively and gradually, once a level of the hierarchy is formed the procedure is repeated for each class.

4 Document clustering based on a Tolerance Rough Set Model

Given a set $\mathcal{D}$ of M full text documents. Our method of generating a hierarchical structure of this document collection consists of two phases. The first

Table 2: Scheme of A-OSHAM approximate conceptual clustering

Input	concept hierarchy H and an existing splittable concept C_k.
Result	H formed gradually.
Top-level	call A-OSHAM(root concept, $\emptyset$).
Variables	α is a given threshold.

1. Suppose that $C_{k_1}, ..., C_{k_n}$ are subconcepts of $C_k = (X_k, S_k)$ found so far. While C_k is still splittable, find a new subconcept $C_{k_{n+1}} = (X_{k_{n+1}}, S_{k_{n+1}})$ of C_k and its approximations by doing:
   (a) Find attribute a^* so that $\bigcup_{i=1}^{n} X_{k_i} \cup \rho(\{a^*\})$ is the largest cover of X_k.
   (b) Find the largest attribute set S containing a^* satisfying $\lambda\rho(S) = S$.
   (c) Form subconcept $C_{k_{n+1}}$ with $\rho(S_{k_{n+1}}) = S$ and $X_{k_{n+1}} = \rho(S)$.
   (d) Find a lower approximation and an upper approximation of $C_{k_{n+1}}$ with respect to a chosen equivalence relation E.

   From intersecting subconcepts corresponding to intersections of $\rho(S_{k_{n+1}})$ with extents of existing concepts on H excluding its superconcepts, and find their approximations.
2. Let $X_k^r = X_k \setminus \bigcup_{i=1}^{n+1} X_{k_i}$. If one of the following conditions holds then C_k is considered unsplittable:
   (a) There exist not any attribute set $S \subseteq S_k$ satisfying $\lambda\rho(S) = S$ in X_k.
   (b) $card(X_k^r) \leq \alpha$.
3. Apply A-OSHAM(C_{k_i}, H) to each C_{k_i} formed in the step 1.

phase extracts and maps each document into a set of terms, then enriches documents with their approximations by the proposed tolerance rough set model. The second phase groups documents by an agglomerative clustering method using the document approximations.

In the first phase each document d_j is mapped into a list of terms t_i each is assigned a weight that reflects its importance in the document. Denote by $f_{d_j}(t_i)$ the number of occurrences of term t_i in d_j (term frequency), and by $f_{\mathcal{D}}(t_i)$ the number of documents in $\mathcal{D}$ that term t_i occurs in (document frequency). The weights w_{ij} of terms t_i in documents d_j are first calculated

by

$$w_{ij} = \begin{cases} (1 + \log(f_{d_j}(t_i))) \times \log \frac{M}{f_{\mathcal{D}}(t_i)} & \text{if } t_i \in d_j, \\ 0 & \text{if } t_i \notin d_j \end{cases} \tag{11}$$

then normalized by vector length as $w_{ij} \leftarrow w_{ij}/\sqrt{\sum_{t_{hj} \in d_j} (w_{hj})^2}$. Each document d_j is represented by its r highest-weighted terms. A usual way is to fix a default value r common for all documents. We denote the document set by $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ where $d_j = (t_{1j}, w_{1j}; t_{2j}, w_{2j}; \ldots; t_{rj}, w_{rj})$ and $w_{ij} \in [0, 1]$. The set of all terms from $\mathcal{D}$ is denoted by $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$. In information retrieval, a query is given the form $Q = (q_1, w_{1q}; q_2, w_{2q}; \ldots; q_s, w_{sq})$ where $q_i \in \mathcal{T}$ and $w_{iq} \in [0, 1]$.

The *tolerance rough set model* (TRSM) aims to enrich the document representation in terms of semantics relatedness by creating tolerance classes of terms in $\mathcal{T}$ and approximations of subsets of documents. The model has the root from rough set models and its extensions [10]. The key idea is among three properties of an equivalence relation R in an universe U used in the original rough set model (reflexive: xRx; symmetric: $xRy \rightarrow yRx$; transitive: $xRy \wedge yRz \rightarrow xRz$ for $\forall x, y, z \in U$), the transitive property does not always hold in natural language processing, information retrieval, and consequently text data mining. In fact, words are better viewed as overlapping classes which can be generated by *tolerance relations* (requiring only reflexive and symmetric properties).

The key issue in formulating a TRSM to represent documents is the identification of tolerance classes of index terms. There are several ways to identify conceptually similar index terms, e.g., human experts, thesaurus, term co-occurrence, etc. We employ the co-occurrence of index terms in all documents from $\mathcal{D}$ to determine a tolerance relation and tolerance classes. The co-occurrence of index terms is chosen for the following reasons: (i) it gives a meaningful interpretation in the context of information retrieval about the dependency and the semantic relation of index terms, and (ii) it is relatively simple and computationally efficient. Note that the co-occurrence of index terms is not transitive and cannot be used automatically to identify equivalence classes. Denote by $f_{\mathcal{D}}(t_i, t_j)$ the number of documents in $\mathcal{D}$ in which two index terms t_i and t_j co-occur. We define an uncertainty function I depending on a threshold θ as

$$I_\theta(t_i) = \{t_j \mid f_{\mathcal{D}}(t_i, t_j) \geq \theta\} \cup \{t_i\} \tag{12}$$

It is clear that the function I_θ defined above satisfies the condition of $t_i \in I_\theta(t_i)$ and $t_j \in I_\theta(t_i)$ iff $t_i \in I_\theta(t_j)$ for any $t_i, t_j \in \mathcal{T}$, and so I_θ is both

Table 3: The TRSM nonhierarchical clustering algorithm

Input	The set $\mathcal{D}$ of documents and the number K of clusters.
Result	K clusters of $\mathcal{D}$ associated with cluster membership of each document.

1. Determine the initial representatives $R_1, R_2, ..., R_K$ of clusters $C_1, C_2, ..., C_K$ as K randomly selected documents in $\mathcal{D}$.
2. For each $d_j \in \mathcal{D}$, calculate the similarity $S(\mathcal{U}(\mathcal{R}, d_j), R_k)$ between its upper approximation $\mathcal{U}(\mathcal{R}, d_j)$ and the cluster representative $R_k, k = 1, ..., K$. If this similarity is greater than a given threshold, assign d_j to C_k and take this similarity value as the cluster membership $m(d_j)$ of d_j in C_k.
3. For each cluster C_k, re-determine its representative R_k.
4. Repeat steps 2 and 3 until there is little or no change in cluster membership during a pass through $\mathcal{D}$.
5. Denote by d_u an unclassified document after steps 2, 3, 4 and by NN(d_u) its nearest neighbor document (with non-zero similarity) in formed clusters. Assign d_u into the cluster that contains NN(d_u), and determine the cluster membership of d_u in this cluster as the product $m(d_u) = m(\text{NN}(d_u)) \times S(\mathcal{U}(\mathcal{R}, d_u), \mathcal{U}(\mathcal{R}, \text{NN}(d_u)))$. Re-determine the representatives R_k, for $k = 1, ..., K$.

reflexive and symmetric. This function corresponds to a tolerance relation $\mathcal{I} \subseteq \mathcal{T} \times \mathcal{T}$ that $t_i \mathcal{I} t_j$ iff $t_j \in I_\theta(t_i)$, and $I_\theta(t_i)$ is the tolerance class of index term t_i.

A vague inclusion function ν, which determines how much X is included in Y, is defined as

$$\nu(X, Y) = \frac{|X \cap Y|}{|X|} \tag{13}$$

This function is clearly monotonous with respect to the second argument. Using this function the membership function, introduced by Pawlak [10], a similar notion as that in fuzzy sets, μ for $t_i \in \mathcal{T}, X \subseteq \mathcal{T}$ can be defined as

$$\mu(t_i, X) = \nu(I_\theta(t_i), X) = \frac{|I_\theta(t_i) \cap X|}{|I_\theta(t_i)|} \tag{14}$$

With these definitions we can define a tolerance space as $\mathcal{R} = (\mathcal{T}, I, \nu, P)$ in which the *lower approximation* $\mathcal{L}(\mathcal{R}, X)$ and the *upper approximation*

Table 4: TRSM-based hierarchical agglomerative clustering algorithm

---

Input A collection of M documents $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$
Result Hierarchical structure of $\mathcal{D}$

Given: a collection of M documents $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$
a similarity measure $sim : \mathcal{P}(\mathcal{D}) \times \mathcal{P}(\mathcal{D}) \rightarrow R$
for $j = 1$ **to** M **do**
$C_j = \{d_j\}$ **end**
$H = \{C_1, C_2, , \ldots, C_M\}$
$i = M + 1$
while $|H| > 1$
$(C_{n_1}, C_{n_2}) = \text{argmax}_{(C_u, C_v) \in H \times H} sim(\mathcal{U}(\mathcal{R}, C_u), \mathcal{U}(\mathcal{R}, C_v)$
$C_i = C_{n_1} \cup C_{n_2}$
$H = (H \setminus \{C_{n_1}, C_{n_2}\}) \cup \{C_i\}$
$i = i + 1$

$\mathcal{U}(\mathcal{R}, X)$ in $\mathcal{R}$ of any subset $X \subseteq \mathcal{T}$ can be defined as

$$\mathcal{L}(\mathcal{R}, X) = \{t_i \in \mathcal{T} \mid \nu(I_\theta(t_i), X) = 1\} \tag{15}$$

$$\mathcal{U}(\mathcal{R}, X) = \{t_i \in \mathcal{T} \mid \nu(I_\theta(t_i), X) > 0\} \tag{16}$$

The term-weighting method defined by Eq. (11) is extended to define weights for terms in the upper approximation $\mathcal{U}(\mathcal{R}, d_j)$ of d_j. It ensures that each term in the upper approximation of d_j but not in d_j has a weight smaller than the weight of any term in d_j.

$$w_{ij} = \begin{cases} (1 + \log(f_{d_j}(t_i))) \times \log \frac{M}{f_{\mathcal{D}}(t_i)} & \text{if } t_i \in d_j, \\ \min_{t_h \in d_j} w_{hj} \times \frac{\log(M/f_{\mathcal{D}}(t_i))}{1 + \log(M/f_{\mathcal{D}}(t_i))} & \text{if } t_i \in \mathcal{U}(\mathcal{R}, d_j) \setminus d_j \\ 0 & \text{if } t_i \notin \mathcal{U}(\mathcal{R}, d_j) \end{cases} \tag{17}$$

The vector length normalization is then applied to the upper approximation $\mathcal{U}(\mathcal{R}, d_j)$ of d_j. Note that the normalization is done when considering a given set of index terms.

Figure 3 and Figure 4 describe two general TRSM-based nonhierarchical and hierarchical clustering algorithm. The TRSM-based nonhierarchical clustering algorithm can be considered as a reallocation clustering method to form K clusters of a collection $\mathcal{D}$ of M documents. The main point of the TRSM-based hierarchical clustering algorithm is at each merging step it uses upper

approximations of documents in finding two closest clusters to merge. Several variants of agglomerative clustering can be applied, such single-link or complete-link clustering. As documents are represented as length-normalized vectors and when cosine similarity measure is used, an efficient alternative is to employ the group-average agglomerative clustering. The group-average clustering can avoid the elongated and straggling clusters produced by single-link clustering, and can avoid the high cost of complete link clustering. In fact, it allows using cluster representatives to calculate the similarity between two clusters instead of averaging similarities of all document pairs each belong to one cluster [8]. In such a case, the complexity of computing average similarity would be $O(N^2)$. Careful evaluation and validation of clustering quality are given in [5] and [6]. The results show that tolerance rough set model and TRSM-based clustering algorithms can be used to improve the effectiveness and efficiency in information retrieval and text analysis.

References

[1] Ganter, B., Wille, R., *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, 1999.

[2] Ho, T.B., Discovering and Using Knowledge From Unsupervised Data. *Decision Support Systems*, Elsevier Science, Vol. 21, No. 1, 1997, 27–41.

[3] Ho, T.B., "Two approaches to the representation and interpretation of concepts in concept lattices", *Information Modelling and Knowledge Bases XI*, IOS Press, 2000, 12-25.

[4] Ho, T. B. and Funakoshi K., "Information retrieval using rough sets", *Journal of Japanese Society for Artificial Intelligence*, Vol. 13, N. 3, 1998, 424–433.

[5] Ho, T. B. and Nguyen, N.B., "Nonhierarchical Document Clustering by Tolerance Rough Set Model", *International Journal of Intelligent Systems*, Vol. 17 (2002), No. 2, 199–212.

[6] Kawasaki, S, Nguyen, N.B., and Ho, T. B., "Hierarchical Document Clustering Based on Tolerance Rough Set Model", *Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, September 2000, Lyon. Lecture Notes in Artificial Intelligence 1910, Springer, 458–463

[7] Kent, R.E., "Rough concept analysis", *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994, 248–255.

[8] Manning, C. D. and Schutze, H., *Foundations of Statistical Natural Language Processing*, The MIT Press, 1999.

[9] Michalski R.S. and Stepp R.E., "Learning from observation: Conceptual learning", *Machine Learning: An Artificial Intelligence Approach*, Vol. 1, R. S. Michalski, J. G. Carbonelle, T. M. Michell (Eds.), Morgan Kaufmann, 1983, 331–363.

[10] Pawlak, Z., *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, 1991.

[11] Wille, R., "Restructuring lattice theory: An approach based on hierarchies of concepts", *Ordered Sets*, I. Rival (Ed.), Reidel, 1982, 445–470.