

From Contingency Tables to Taxonomies

F. A. El-Mouadib

Department of Computer Science Garyounis University Benghazi, Libya

ABSTRACT

Databases are a very useful source of different forms of knowledge, one of which is taxonomies. Classification of objects, which is a form of acquiring knowledge, involves formation of classes and construction of one or more taxonomies that include those classes.

In this paper, we demonstrate how contingency tables can be used as a base to build one-level taxonomy elements. In fact only a subset of them, which approximate equivalence relations, can be used to construct taxonomies. The approximate equivalence relations can be found by the use of a set of lambda (λ) measures of association. The process of forming multi-level taxonomy relies on approximate equivalence relations and guided by partition utility functions.

Key Words: Approximate equivalence relations; Concept hierarchy; Contingency tables; Equivalent hierarchy elements; Knowledge discovery.

INTRODUCTION

Knowledge Discovery in Databases (KDD) (database mining) is both feasible and practical. It has recently been an active research area in many business and scientific domains. Knowledge discovery, according to Frawley et al (1991), has been defined as the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. KDD has been applied to many fields such as: Medicine, Agriculture, Insurance, Engineering, Military, Space Science, and many other fields.

Knowledge comes in different forms such as rules, contingency tables, equations, concepts and taxonomies. Classification of objects involves the construction of taxonomy of classes. The classes usually have a hierarchical organization in which subclasses possess the discriminating features of their super-classes, and classes, which are the siblings in the hierarchy, are mutually exclusive with respect to the presence or absence of some set of features. Taxonomy, as defined by Klosgen and Żytkow (1995), is a hierarchical system of selected subsets of a domain, typically arranged in a tree, which is exhaustive, and disjoint. In this paper, we limit our attention to construct binary taxonomy trees. At each node of the tree, only a binary split of the data is allowed. In our method, we start with contingency tables, as basic building blocks, and simple tests, which distinguish special kind of knowledge, one-level taxonomies can be build. In order to find one-level taxonomy, a systematic search is performed for pairs of attributes whose association is "strong enough", which approximate equivalence relation. If such approximately equivalent relations are found at a particular node of the tree, then one-level taxonomy is derived and used to guide the split of the data at that

particular node. Unlike the ID3 system Quinlan (1990); our system is not confined to split the data in each node on only one attribute. Given our algorithm's simplicity, it is suitable for data mining applications. Indeed, in the process of building a taxonomy tree, our method does not use any similarity/dissimilarity measures but instead it relies only on approximating equivalence relations and on Partition Utility (PU) functions. It should be emphasized that our method works in a strict sense unsupervised fashion, i.e., no a priori model of the data is assumed (neither the number of classes is prespecified nor any assumptions about data distribution are made).

The Search for Approximate Equivalence Relations:

Let us begin with a pair of attributes measured on a nominal scale. A 2D-contingency table represents equivalence relation between the two attributes or classifications (**A** and **B**) involved if and only if population is concentrated in cells no two of which are in the same row or column of the table. In real world databases pairs of attributes, which are represented by such tables, are rare. Therefore, our algorithm relies only on approximate equivalence relations in the process of forming taxonomy. In order to find approximate equivalence relations, we use a set of lambda measures (λ_b , λ_a and λ) of association introduced by (Goodman & Kruskal, 1954; Bishop, Fienberg, & Holland, 1975) gives general treatment to a number of measures of association.

The measure λ_b is indeterminate if and only if the entire population lies in one column of the contingency table, otherwise $0 \leq \lambda_b \leq 1$ inclusive; $\lambda_b = 0$ if and only if knowledge of the **A** classification is of no help in predicting the **B** classification; $\lambda_b = 1$ if and only if knowledge of the **A** classification completely specifies the **B** classification; if **A** and **B** are independent and λ_b is determinate then $\lambda_b = 0$; λ_b is unchanged by the permutation of rows and columns. Analogously to λ_b , the measure λ_a can be defined. Clearly, the measure λ can be defined in the same way as λ_b . Both measures have analogous properties. The computation of λ_a , λ_b and λ is extremely simple. In particular, in terms of cell counts of the contingency table,

$$\lambda = \frac{\sum_a n_{am} + \sum_b n_{mb} - n_{.m} - n_{m.}}{2n - (n_{.m} + n_{m.})}$$

$$\lambda_b = \frac{\sum_a n_{am} - n_{.m}}{n - n_{.m}}$$

$$\lambda_a = \frac{\sum_b n_{mb} - n_{m.}}{n - n_{m.}}$$

where n is the size of the population, n_{am} is the maximum cell count in row a , n_{mb} is the maximum cell count in column b , $n_{.m} = \max_b \{ \sum_a n_{ab} \}$ is the maximum of columns totals in the contingency table, $n_{m.} = \max_a \{ \sum_b n_{ab} \}$ is the maximum of row totals in the contingency table, $\sum_a n_{am}$ is the total of rows maximums in the contingency table, $\sum_b n_{mb}$ is the total of

columns maximums in the contingency table. A detailed discussion the set of λ measures is given in(El-Mouadib,1997).

For attributes measured on an ordinal scale, association between them can be represented by probability for like and unlike orders (positive or negative association). The issue of (approximate) equivalence is not tied to the notion of order. The set of λ measures can be used to evaluate if approximate equivalence is present between any ordered attributes. Usually, clusters present in data described by ordered attributes reflect the ordered structure of the data: say, “small” values of an attribute appear in one cluster and “medium and large” values in another, not “small and large” in one and “medium” in another. In such case, one may want to have clusters reflecting such ordered structure of data. For ordered data, our method uses a measure of closeness to equivalence based on the probabilities of like and unlike orders. Our measure gamma (γ_{mod}) is a modified one than the gamma (γ) measure that was proposed by(Goodman & Kruskal,1954) for ordered attributes. Details of γ_{mod} measure are given in(El-Mouadib & Koronacki ,1999).

In summary, in order to form one-level taxonomies, a systematic search is performed for pairs of attributes whose strength of association is above some controllable threshold value. In our method, a threshold can be set for λ_a and λ_b , and any pair of attributes whose λ_a or λ_b is above that threshold is considered conducive to one-level taxonomy. If such a pair is found, its contingency table is then aggregated into a two-by-two table by merging values of the two attributes involved. Aggregation of values should be performed in such a way that the resulting 2×2 table has the largest possible value of λ . If λ proves sufficiently close to one, the approximate equivalence described by the 2×2 table provides a natural basis for a binary split (i.e., creation of a hierarchy element). If an attribute is nominal, aggregation of its values is preceded by their rearrangement guided by correspondence analysis (rearrangement by correspondence analysis is a standard statistical tool: see, e.g.,(Mardia, Kent, & Bibby,1979) for an illustrative example).

Loosely speaking, correspondence analysis of rows of the contingency table (the same applies to columns) may enable one to rearrange the rows in such a way that the least distance rows are adjacent after rearrangement and, generally, the order in which rearranged rows appear in the table reflect the distances between the rows.

After rows and columns rearrangement, or without it if the attributes are ordered, aggregation can be easily done automatically by merging rows 1 to a into one aggregated level of attribute A and rows $a + 1$ to α into another level of A , and merging columns 1 to b into one aggregated level of attribute B and columns $b + 1$ to β into another. Running over all possible pairs (a, b) , one finds a 2×2 table with the largest value of λ . If the maximum value of λ is above a pre specified threshold, the table is retained for further analysis.

Partition Utility:

In our taxonomy formations process (El-Mouadib , 1997), the search for approximate equivalence relations may leads to the creation of many one-level taxonomy (hierarchy

element¹). The one-level taxonomies can be merged into one one-level taxonomy if they have the same data range and shares common descriptors. Still, after merging, several one-level taxonomies may remain. In order to choose one of them to be the next node to be added to the taxonomy tree, we use the partition utility function (Fisher & Hapanyengwi,1993; Fisher ,1996). The taxonomy formation algorithm picks the one-level taxonomy with the greatest partition utility value as the one, which guides the split at this level of the tree. The partition utility function based on the Gini index, is given by (analogous formula can be given for the PU function based on entropy):

$$PU = \frac{1}{K} \sum_{k=1}^K CU(C_k)$$

$$CU(C_k) = P(C_k) \sum_{i=1}^I \sum_{j=1}^J [P(A_i = V_{ij} | C_k)^2 - P(A_i = V_{ij})^2]$$

where:

$k = 1, \dots, K$ and K is the number of categories in a partition (default=2).

$i = 1, \dots, I$ and I is the number of attributes in the given data.

$j = 1, \dots, J$ and J is the number of values in attribute A_i

Taxonomy Formation Algorithm:

Now, let us Give A Summary of Our Taxonomy Formation Algorithm. A Brief Description of Each Procedure Follows:

Main Algorithm:

Taxonomy formation

Iterate the following steps until there are no more regularity

Discover regularities.

Create hierarchy elements.

Merge similar hierarchy elements.

Partition evaluation.

Choose the appropriate partition.

Build one level of the taxonomy tree.

Split the data.

The main algorithm governs the over all process of building a multi-leveled taxonomy tree. The discovered regularities are in the form of 2-D contingency tables. The search fro regularities is conducted in an exhaustive manner. Each contingency table is examined by λ_a and/or λ_b measures (for nominal data values) to qualify it as regularity if test result exceeds

¹ A hierarchy element is a simple tree structure comprised of 3 classes (the root and two children). The root class is labeled with the description of the class of records in which the equivalence holds. Each of the children is labeled with the descriptors (i.e., statements such as "Attribute name=value") that hold for that child for the given equivalence.

some pre-specified threshold. For ordered data values, the modified gamma (γ_{mod}) measure can be used as well as the measures λ_a and/or λ_b .

As it has been mentioned earlier that correspondence analysis is a standard statistical tool (Mardia, et al.,1979; Krzanowski,1988), for illustration and examples), which guides the rearrangement of rows and/or columns of a contingency table in preparation to the aggregation step. This step is performed only once for each given 2-D contingency table (regularity). The correspondence analysis step produces a 2-D table of the same order as the given 2-D contingency table in which similar rows and/or columns are adjacent.

In the aggregation step, a given 2-D contingency table of higher order is aggregated into a 2×2 table by merging values of the two attributes involved. The aggregation of values in the given 2-D contingency table into a 2×2 table can be visualized as moving a plus sign (+) along the rows and columns of the 2-D contingency table. The plus sign divides the given 2-D contingency table into four groups of cells and the content of each group will be added up into one of the 2×2 table cells. Testing each produced 2×2 table if it has the largest possible value of λ .

A regularity that represents an equivalence relation (or an approximation of it) is used to create a hierarchy element (one-level taxonomy).

After hierarchy elements have been created, the next step in the taxonomy formation process is to merge equivalent hierarchy elements².

All the one-level taxonomies (some of them may have been obtained by merging) are evaluated (according to the partition utility function). The taxonomy formation algorithm picks the most appropriate one-level taxonomy (the one with the greatest partition utility value) to be the next node to be added to the taxonomy tree.

In our taxonomy formation algorithm, only binary splits are allowed. The placement of the one-level taxonomy on the taxonomy tree is done in breath-first fashion. The taxonomy tree is build one level at time.

At each node of the taxonomy tree, the data is partitioned into sub-populations (left and right) according the chosen one-level taxonomy.

The chosen one-level taxonomy is considered to be the condition on which the data is split.

EXPERIMENTS AND RESULTS

Here, we demonstrate the results obtained by our taxonomy formation algorithm for two examples. To examine if the algorithm works as desired, we have used real-life examples with data labeled and coming from known classes. Of course, class labels are disregarded (removed) when building a taxonomy tree.

The first example is the well known small soybean database, (see, e.g., UCI repository of Merz & Murphy, 1996), which have been extensively studied [2, 5, 14, 15, 16], consists of 47

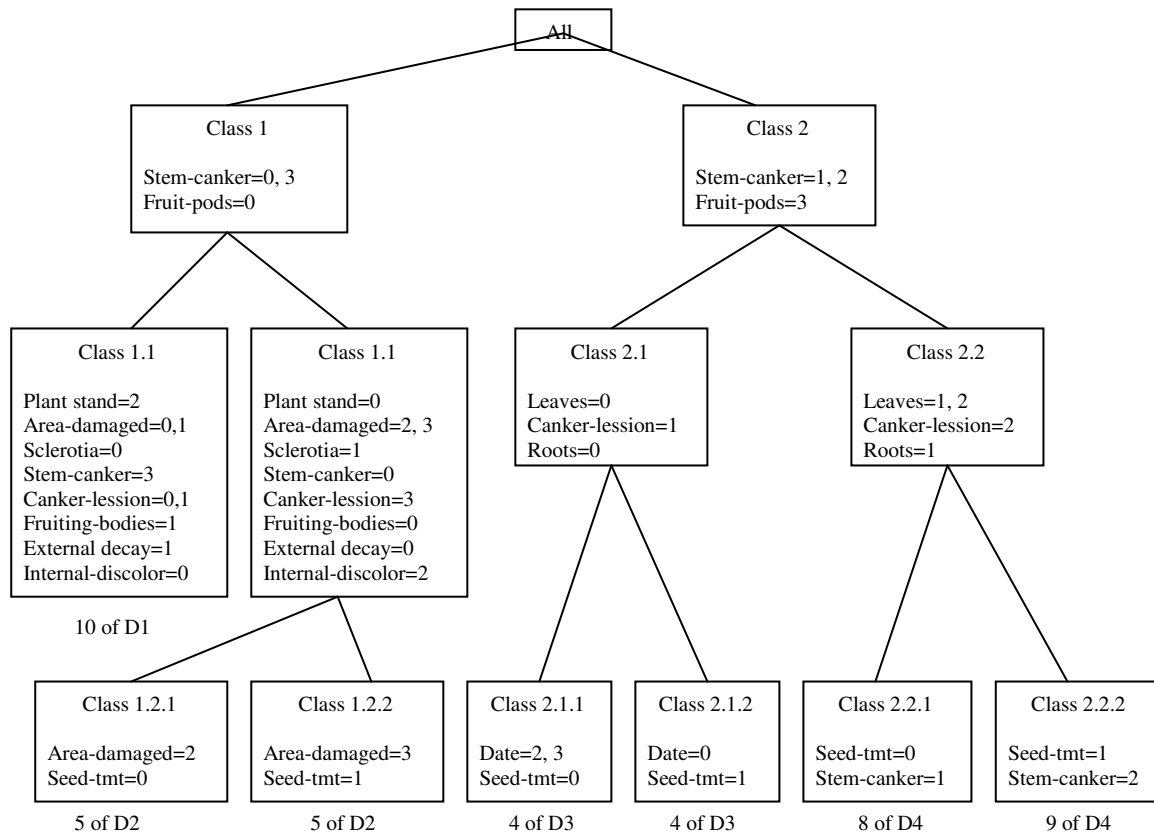
² Two hierarchy elements are equivalent if they have the same range of records in the root and have common descriptors in each of the children.

instances. Each instance was described along 35 attributes. Four categories (classes) of soybean disease were present in the data.

The following table depicts the data specifications.

Disease name	No. of instances	Disease label
Diaporthe Stem Canker	10	D1
Charcoal Rot	10	D2
Rhizoctonia Root Rot	10	D3
Phytophthora Rot	17	D4

Of course, class labels are disregarded (removed) when building a taxonomy tree. All the attributes are nominal and so the set of lambda measures is used to form taxonomy. The final taxonomy tree is as follows:



Out of the 47 instances, 45 of them have been correctly classified and only two instances have been lost during the process of building the taxonomy tree due to the acceptance of approximation. The threshold for the measures λ_a and λ_b was 0.90 and for λ was 0.894. The results of this experiment are; Class 1.1 contains 10 instances and all of them belong to D1. Each of Class 1.2.1 and Class 1.2.2 has 5 instances and all of them belong to D2. Each of Class 2.1.1 and Class 2.1.2 has 4 instances and all of them belong to D3. Class 2.2.1 contains 8 instances and Class 2.2.2 contains 9 instances and all of them belong to D4.

The second example is the famous Iris data (see, e.g., UCI repository of Merz & Murphy, 1996). The data set consists of 150 instances. Each record has 4 numeric attributes and the fifth one is the class attribute. The four attributes represent measurements of an Iris plant; they are of continuous type, and they are: Sepal Length (SL) in cm, Sepal Width (SW) in cm, Petal Length (PL) in cm, and Petal Width (PW) in cm. Each class has 50 instances and the classes are Iris Setosa, Iris Versicolour, and Iris Virginica.

The entire data had to be discretized preceding the taxonomy formation process. In the first version of this experiment, we have discretized the data values according to:

$$h = 2.603(IQ)n^{-1/3} \equiv \tilde{h}_{os}$$

And we have obtained the following results:

Attribute name	Min Value	Max Value	Bin Width	Number of Bins(n_b)
SL	4.3	7.90	0.590	7
SW	2.00	4.40	0.200	12
PL	1.00	6.90	1.670	4
PW	0.10	2.50	0.690	4

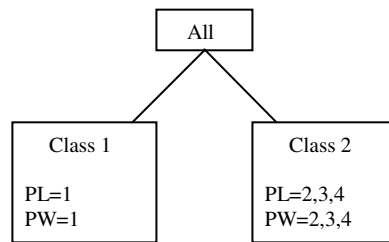
The taxonomy formation process has produced the following approximate equivalence relation for PL and PW with $\lambda_b = 0.74$. We have label discretized attribute's levels from 1 to n_b)

		PW			
		1	2	3	4
PL	1	50	0	0	0
	2	0	24	12	0
	3	0	1	41	5
	4	0	0	13	4

After aggregation, the taxonomy formation process has produced the following equivalence relation with λ equal to 1.00.

		PW	
		1	2,3,4
PL	1	50	0
	2,3,4	0	100

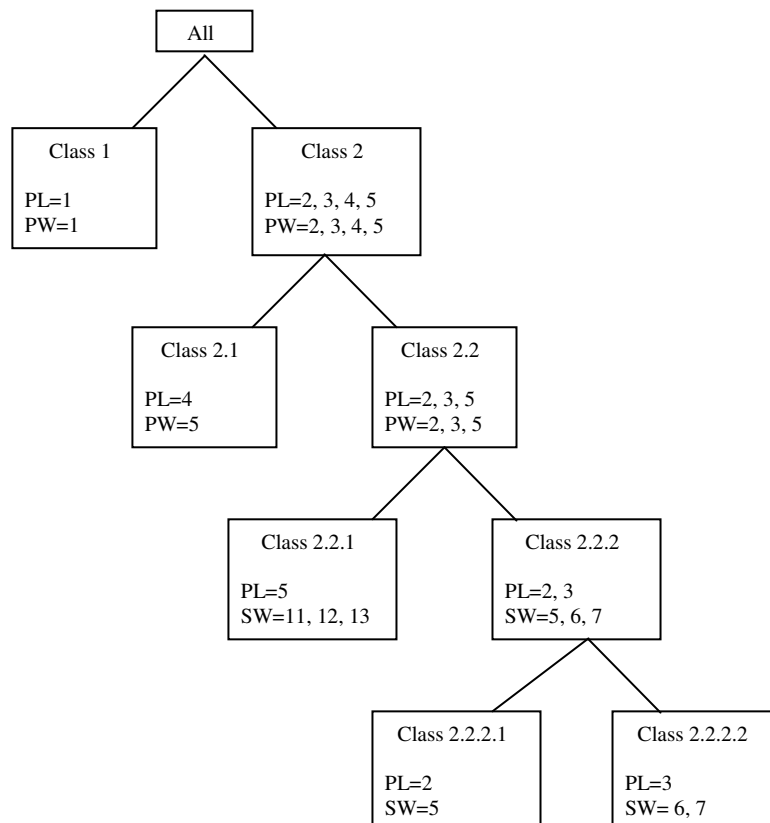
A one-level taxonomy is constructed as follows:



At this level, no approximate equivalence relation has been found (Class1 comprises of all cases of Iris Setosa, while Class2 consists of all cases of Iris Versicolour and Iris Virginica). In the second version of this experiment, bin width has been decreased to $h=2 (IQ)n^{-1/3}$, which lead to the following discretization results:

Attribute name	Min Value	Max Value	Bin Width	Number of Bins(n_b)
SL	4.3	7.90	0.590	8
SW	2.00	4.40	0.200	13
PL	1.00	6.90	1.670	5
PW	0.10	2.50	0.690	5

The taxonomy process has produced the following taxonomy tree:



The results of the above taxonomy tree are summarized in the following table:

Class	No. of instances	Classification
Class1	50	Iris Setosa
Class2.1	40	Iris Virginica
Class2.2.1	6	Iris Virginica
Class2.2.2.1	6	Iris Versicolour
Class2.2.2.2	43	Iris Versicolour
	4	Iris Virginica

Out of the 150 instances, 149 of them have been correctly classified and only one instance has been lost during the process of building the taxonomy tree due to the approximation. The lowest threshold for λ_b was 0.83 and for λ was 0.9166.

Concluding remarks:

In addition to the two examples above, the taxonomy formation process was examined on three more databases from the UCI repository of (Merz & Murphy, 1996): the large soybean, virus, and crabs data. In all the examples results obtained were encouraging and of similar “quality” as those for the two examples discussed.

The obtained results have proven the validity of the use of contingency tables as basic building blocks for taxonomy formation. In fact only a subset of them, which approximate equivalence relations, can be used to construct taxonomies. The results we have obtained from all the experiments that we have conducted are truly encouraging in the field of knowledge discovery in databases. In conclusion, the author would like to make a number of remarks to be considered in future research:

1. Finding some criterion to deal with missing data values.
2. Finding some method to deal with sparse data.
3. Finding an improved approach to discretization, which would be less “ad hoc”?

REFERENCES

- Bishop, Y., Fienberg, S., & Holland, P. (1975). *Discrete multivariate analysis theory and practice*. USA: The MIT Press Cambridge.
- Biswas, G., Weinberg, J., Yang, Q., & Koller, G. (1991). *Conceptual clustering and exploratory data analysis*. Evanston: Proceedings of the 8th International Workshop on Machine Learning.
- El-Mouadib, F., & Koronacki, J., (1999). *On taxonomy formation by approximate equivalence relations*. Ustron: Intelligent Information Systems VIII, Proceedings of the Workshop held in Ustron/Wisa, Poland.
- El-Mouadib, F., (1997). *Taxonomy Formation*. Zakopane: Intelligent Information Systems VI, Proceeding of the workshop held in Zakopane, Poland.

- Fisher, D. (1987). Knowledge acquisition via incremental concept clustering. *Machine Learning*, 2(9),39–172.
- Fisher, D., & Hapanyengwi, G. (1993). Database Management and analysis tools of machine induction. *Journal of Intelligent Information Systems*, 2, 5-38.
- Frawley, W. J., Piatetsky-Shapiro, G., & Matheus C. J. (1991). Knowledge discovery in databases: an overview. In G. Piatetsky-Shapiro & W. J. Frawley (eds.). *Knowledge Discovery in Databases*(pp. 1–27). California: AAAI/MIT Press.
- Goodman,L., & Kruskal, W.(1979). *Measure of association for cross classification: Springer series in statistics*. New York: Springer-Verlag.
- Klosgen, W., & Żytkow, J. (1995). Knowledge discovery in databases terminology. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (eds.). *Advances In Knowledge Discovery and Data Mining*. (pp. 574–592). California: AAAI Press.
- Krzanowski, W. J. (1988). Principles of multivariate analysis, a user's perspective. *Oxford Statistical Science*, Series-3, 24 – 32 & 86 – 104.58
- Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). *Multivariate analysis*. Academic Press.
- Merz, C. J., & Murphy, P. M. (1996). *UCI Repository of machine learning databases*, irving, CA. California: University of California.
- Quinlan, J. R. (1990). Induction of Decision Trees, Readings in machine learning. In J. W. Shavlik, & T. G. Dietterch (eds.). *Morgan Kufmann* (pp. 57–69). Bosten: Kluwer Academic Publishers.
- Troxel M., Swarm K., Zembowicz, R., & Żytkow J. (1994). Concept hierarchies: a restricted form of knowledge derived from regularities. In Z. Raś, & M. Zemankova (eds.). *In Proceeding of The seventh International Symposium on Methodologies for Intelligent Systems*. (pp. 437-447).
- Zembowicz, R., & Żytkow, J. (1995). From contingency tables to various forms of knowledge in databases. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (eds.). *Advances in Knowledge Discovery and Data Mining*. (pp. 329–349).). California: AAAI Press.
- Żytkow, J., & Zembowicz R. (1997). Contingency tables as the foundation for concepts, concept hierarchies and rules: the 49er system approach. *Fundamenta Informaticae*. 30, 383–399.