The Taxonomy-Based Learning: Algorithms and applications

Jiawei Hong and Xiaonan Tan
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 89-65

July 10, 1989

The Taxonomy-Based Learning: Algorithms and applications

Jiawei Hong and Xiaonan Tan
Department of Computer and Information Science
university of Massachusetts at Amherst

July 10, 1989

Abstract

This paper presents a formal model of taxonomy tree and an efficient algorithm for taxonomy-based machine learning. The application of taxonomy tree to Chinese phonetics is reported. By using the taxonomy tree algorithm, a computer can partition all Chinese consonants into different groups, which matches the classical taxonomy of Chinese consonants. The concepts of labials, alveolars, velars, prepalatals,... in Chinese phonetics have been rediscovered by a computer.

1 Introduction

Similarity-Based Learning method has received a great deal of attention recently. This machine learning method involves the comparison of several instances of a concept in order to find features shared among them and differences between them. Common features are assumed to define a useful concept [1][2].

By our common knowledge, "animal" is a concept, cat, dog, horse, chicken... are all its instances, "cat" is another concept whose instances include white cat, black cat,... and so on. It is by the common features that people form concepts. Similar to human learning, the concern of similarity-based learning in inductive concept acquisition has been the determination of characteristic descriptions, which represent concepts by summarizing the properties that hold true for all instances of the concept. Characteristic descriptions

are typically encoded as a single conjunction of maximally specific features. One of the most important problems in similarity-based learning is how to detect similarity and differences between examples to reveal regularities [2].

In this paper we present a mathematical model, by which we can describe what is a concept in a computational way and thereby can design an algorithm to discover some new concepts using a computer.

To achieve this goal, we may define a distance function $d(a, b)$ of two things a and b. The value of $d(a, b)$ may be chosen as the total number of features at which a and b do not agree. If this distance is small, we may put a and b together.

If at the same time, $d(b, c)$ is also small, we may put c into the same class, then both a and c are in the same class. However, we can not guarantee that $d(a, c)$ would be small. Based on $d(a, b)$, we can define a sophisticated distance function $D(a, b)$ such that if a and b are in the same class (in the extension of a concept) but c is not in the class (not in the extension of the concept), then we have $D(a, b) < D(a, c) = D(b, c)$, i.e., the distance between two items in the extension of a concept is always smaller than the distance between any item in the extension and any item not in the extension.

Based on this distance function, we can define a complete taxonomy tree and prove that there is always such a unique tree. In this complete taxonomy tree, each leaf is a concrete item, each inner node corresponds to a concept.

We give an efficient algorithm to find the unique taxonomy tree.

As an application, we use this concept learning method to Chinese phonetics. Using this algorithm, a computer can partition all consonants into 9 groups. A computer has formed concepts corresponding to labials, alveolars, prepalatals, velars... in Chinese phonetics.

Compared with the classical taxonomy of Chinese consonants which has been established and improved for thousand of years, the result obtained by computer is very reasonable. The complete taxonomy tree for Chinese vowels is also obtained.

The taxonomy tree depends on the distance function. There are many different ways to choose a distance function. Some features may added, some may ignored, and some may weighted. Our results simply depend on the information that whether or not a combination of a consonant and a vowel exists in a standard Chinese dictionary. If we could consider the places of articulation as additional features, the result obtained can be improved further.

2 The Taxonomy Tree

It is very convenient to describe the taxonomy by a tree structure. Figure 1 shows part of the tree represents animal taxonomy.

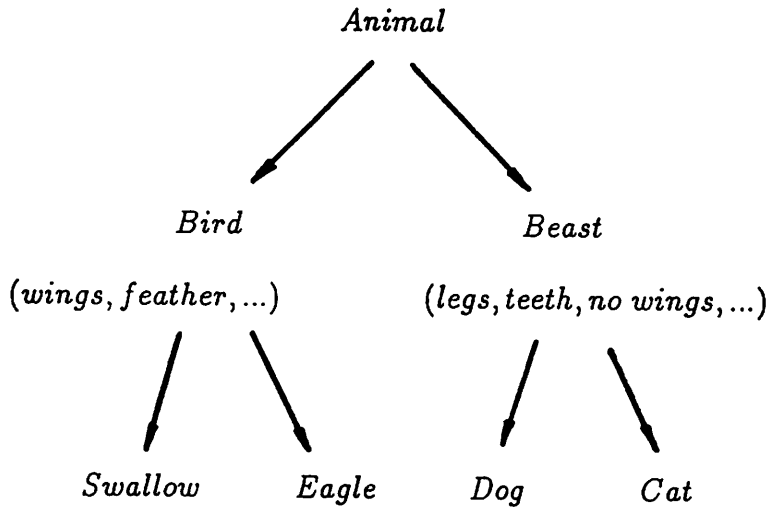

*Fig.*1

Formally, we give the following definitions:

Definition 1 *A taxonomy tree of size n is a labelled tree with n leaves, each inner node has at least two sons and labelled by a concept, each leaf is labelled with a different item.*

We consider the distance between two items.

Definition 2 *If for any pair of items p, q, we define a non-negative number* $d(p,q)$ *satisfying*

$$d(p,p) = 0$$

$$d(p,q) = d(q,p)$$

$$d(p,q) \leq d(p,r) + d(r,q)$$

then we say $d(p,q)$ *is the distance between p and q.*

For example, we can measure the distance between two items by the total number of features in which they are different from each other.

For a set of items $S = \{a_i\}$, we can define a function $D(a, b)$ of $a, b \in S$ as follows:

$$D(a, b) = \quad Min \quad \{Max \quad \{d(a_i, a_{i+1})\}\}$$

In other words, $D(a, b)$ is the minimum distance d such that we can arrive at b from a by a sequence of jumps from an item to another: $a = a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow ... \rightarrow a_n = b$, and $d(a_i, a_{i+1}) \leq d$ for each $i = 0, 1, ..., n-1$.

Now we examine the matter from a different angle: each item is a vertex in a complete edge weighted graph, each edge (a, b) is weighted by a number $d(a, b)$, the length of the edge. Then the number $D(a, b)$ has the following properties:

1. If we remove all edges longer than $D(a, b)$, then a and b are still connected, and

2. If we remove all edges longer than or equal to $D(a, b)$, then a and b are disconnected.

It is not difficult to prove that this is a distance function. Furthermore, we have the following triangular inequality

$$D(a, b) \leq Max\{D(a, c), D(c, b)\}$$

Definition 3 *A complete taxonomy tree is a taxonomy tree which has the following properties:*

1. If a, b are items in the same subtree and c is not in this subtree then

$$D(a, b) < D(a, c) = D(b, c).$$

2. If a node has three subtrees and items a, b, c are on these three subtrees respectively, then

$$D(a, b) = D(b, c) = D(a, c).$$

Theorem 1 *Assume that a distance function d is defined on a set of n items, then there is always a unique (up to isomorphism) complete taxonomy tree which can be found by the following algorithm:*

Consider the indirected graph of n items. For each pair of items (i, j) there is an edge with weight $d(i, j)$, the length of edge (i, j).

Remove the edges which have the largest weight until the graph becomes disconnected. Then draw a root and several sons under the root. Each son corresponds to a connected components. Repeat this for all connected components.

Proof. Assume that items a, b are in the same subtree and c is not in this subtree, then at some point of the above procedure, all edges longer than or equal to a number, say l, have been deleted, a and b are in the same connected component, but c is not in this component. Therefore by our definition, $D(a,b) < l$, $l \leq D(a,c)$ and $l \leq D(c,b)$. By the triangular inequality, we have property 1.

Assume that there is a node which has three subtrees containing items a, b, c respectively, then at some point of our procedure, there is a connected component (corresponding to the node). The largest weight of the remaining edges in this component is l. By deleting all edges of weight l in this component, it is not connected any more: it has three connected components containing items a, b, c respectively. Then by our definition, $D(a,b) = D(b,c) = D(c,a) = l$, we have property 2. Therefore there exists at least one complete taxonomy tree.

Assume that we delete the edges one by one according to their weights until the graph is disconnected after the deleting of all edges of weight l. At this moment, the total number of connected components is, say, $c > 1$. Assume also that there are c_1 sons under the root of a complete taxonomy tree and the distance $D(a,b)$ between any two items a and b in different subtrees under the root is l_1.

Then first, we claim that $l = l_1$. Clearly, since after the removal of all edges whose weight is greater than l, the graph is still connected, we have $l_1 \leq l$. But if $l_1 < l$, then the removal of all edges whose weight is greater than or equal to l will disconnect the graph. Therefore there are two items a and b such that $D(a,b) \geq l > l_1$, a contradiction.

Secondly, we claim $c = c_1$. By the definition of the complete taxonomy tree, we know that after the removal of all edges whose weight is greater than or equal to l, the graph is divided into exactly c connected components. Since $l = l_1$, we have $c = c_1$. Using the same argument, we can show that all complete taxonomy trees are isomorphic to each other inductively.

There are n^2 edges. After the removal of some edges, we use the depth first search to find the connected components, the time needed is $O(n)$. Therefore the time complexity is $O(n^3)$. **QED.**

This method is top-down. Now we give a bottom up method. We assume there is no edge in the graph of n items at the beginning. Then we add the shortest edge(s) to it. Then add the next shortest edge(s) to it,....

The bottom up taxonomy algorithm:

begin

Sort the triples $(d(i,j),i,j)$ by their first arguments. Put them in Q;

While Q is not empty **do**

begin

Select and remove all edges in Q that have the minimum value $d(i,j)$;

If these edges connect different subtrees then put all subtrees under a common root (to form a new larger subtree);

end

end

For each new added edge (a,b), we should find the roots of those two subtrees containing a and b respectively. The time is proportional to the depth of the taxonomy tree. Therefore the total time complexity is $O(n^2d)$, where d is the depth of the taxonomy tree.

By the union-find algorithm, we can reduce the complexity to $O(n^2log^*n)$. Since the best sorting algorithm needs time $O(n^2log\ n)$, the total complexity of the algorithm is $O(n^2log\ n)$.

To show the application of the taxonomy-based learning, we choose Chinese phonetics as an example.

3 Chinese Consonants Classified According to the Place of Articulation

Phonetics is an essential subject in the study of spoken language. The study on Chinese phonetics has a long history. Phoneticians have done much research on it and obtained many valuable and important results. The consonant classification is one of them [3][4].

In traditional Chinese phonetics, there are 21 consonantal initials and 35 (simple or compound) vowels. According to their place of articulation, consonants in Chinese may be classified into six categories. They are labials, denti-alveolars, prepalatals, post-alveolars, alveolars and velars. Labial refers to the interaction of both lips, labio-dental to that of the lower lip and the upper teeth and the rest indicate characteristic locations which the tongue either touches or appro- aches. They are classified and arranged in this way for the following reasons:

1) The consonants of each category have their homorganic nature in articulation.

2) In the constitution of syllables, certain sets of initials occur before certain sets of finals.

3) It is more convenient to compare the consonants of each category with those in other Chinese dialects.

The Chinese consonants are tabulated in Tab.1.

Labials:	b p m f
Denti-alveolars:	z c s
Alveolars:	d t n l
Prepalatals:	zh ch sh r
Post-alveolars	j q x
Velars:	g k h

*Tab.*1.

Except these 21 consonants, we have two other consonants, y and w, which can be combined with Chinese vowels. Therefore we have altogether 23 consonants.

In the following, we apply our algorithm to find the complete taxonomy tree and compare our result with the above classical one.

First of all, we should find a distance function d. A syllable in standard Chinese is generally made up of an initial (a consonant) and a final (a simple or compound vowel). We look at the "Xin Hua" Dictionary (a standard Chinese dictionary) and make a table of syllable-formation (See Table 2) [4]. Each consonant corresponds to a column, and each simple or compound vowel corresponds to a row. If the ith initial can go with the jth final, we put 1 in the crossing entry of row i and column j, otherwise, we put 0 in it. For example, "ma" (mather) is a Chinese word in this dictionary, where "m" is a consonant and "a" is a vowel, then we put a 1 in the crossing entry of the row "a" and column "m".

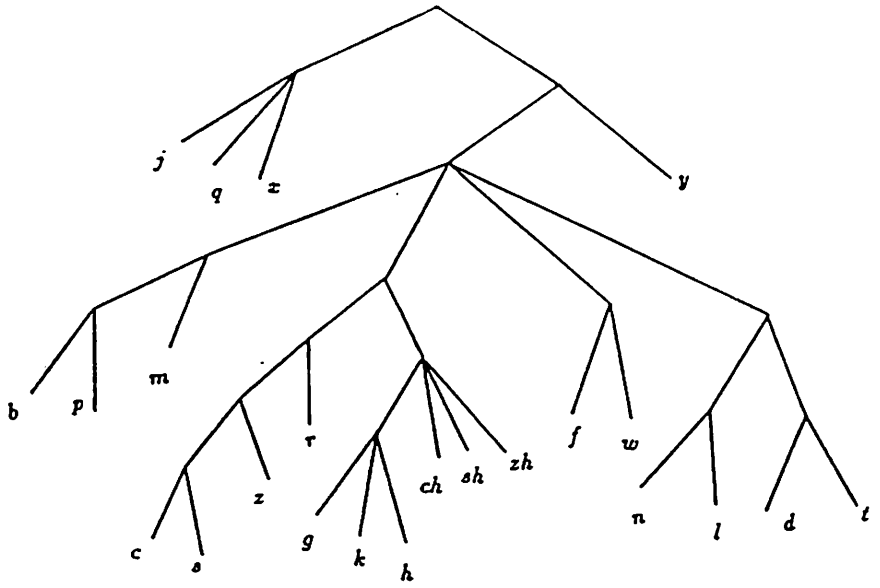

Fig. 2.

	b	c	ch	d	f	g	h	j	k	l	m	n	p	q	r	s	sh	t	w	x	y	z	zh
a	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1
ai	1	1	1	1	0	1	1	0	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1
an	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1
ang	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1
ao	1	1	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1
e	0	1	1	1	0	1	1	0	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1
ei	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	0	0	1	1
en	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	1
eng	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
i	1	0	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0	1	1	0	0
î	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1
ia	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0
ian	1	0	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0	1	0	0	0
iang	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0
iao	1	0	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0	1	0	0	0
ie	1	0	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0	1	0	0	0
in	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1	1	0	0
ing	1	0	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0	1	1	0	0
iong	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
iu	0	0	0	1	0	0	0	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0
o	1	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0
ong	0	1	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	1
ou	0	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1
u	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1
ü	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	1	1	0	0
ua	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1
uan	0	1	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	1	1
üan	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
uai	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1
uang	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1
üe	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	1	1	0	0
ui	0	1	1	1	0	1	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1
un	0	1	1	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1
ün	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
uo	0	1	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	1	1

Tab. 2.

The distance $d(b,c)$ of two consonants b and c is defined as the total number of places where these two columns are different. For example in the table, $d(b,c) = 16$. The distance between Chinese consonants can be easily computed.

Using the taxonomy algorithm, we can find the complete taxonomy tree. Figure 2 is the complete taxonomy tree of Chinese consonants. A similar result is obtained for Chinese vowels.

Therefore the consonants can be roughly partitioned into 7 roups and the vowels can be partioned into 5 roups.

4 Two Dimensional Taxonomy

We refine the taxonomy by partition each group according to each member's spellability with
groups in another dimension. For example, the spellability of the consonant group c, s, z, r, ch, sh, zh, g, k, h with vowel groups can be represented by Tab. 3.

	c	s	z	r	ch	sh	zh	g	k	h
$\{i,..\}$										
$\{o\}$										
$\{a,..\}$	1	1	1	1	1	1	1	1	1	1
$\{i\}$	1	1	1	1	1	1	1			
$\{ua,..\}$					1	1	1	1	1	1

Tab.3.

The distance matrix is therefore

	c	s	z	r	ch	sh	zh	g	k	h
c	0	0	0	0	1	1	1	1	1	1
s	0	0	0	0	1	1	1	1	1	1
z	0	0	0	0	1	1	1	1	1	1
r	0	0	0	0	1	1	1	1	1	1
ch	1	1	1	1	0	0	0	1	1	1
sh	1	1	1	1	0	0	0	1	1	1
zh	1	1	1	1	0	0	0	1	1	1
g	1	1	1	1	1	1	1	0	0	0
k	1	1	1	1	1	1	1	0	0	0
h	1	1	1	1	1	1	1	0	0	0

*Tab.*4.

By the same algorithm, we partition this group into 3 subgroup $\{c, s, z, r\}$, $\{ch, sh, zh\}$, $\{g, k, h\}$. The other consonant groups will not be decomposed. The final result is as follows:

{ *c*, *s*, *z*, *r* },
{ *ch*, *sh*, *zh* },
{ *g*, *k*, *h* },
{ *j*, *q*, *x* },
{ *b*, *p*, *m* },
{ *f*, *w* },
{ *d*, *t* },
{ *n*, *l* },
{ *y* }.

The vowel groups will be decomposed as follows:
{ *a*, *ai*, *an*, *ang*, *ao*, *e*, *ei*, *en*, *eng*, *ou*, *u*,
uan, *un*, *ong*, *ui*, *uo* } $\Longrightarrow$
{ *a*, *ai*, *an*, *ang*, *ao*, *e*, *ei*, *en*, *eng*, *ou*, *u* },
{ *uan*, *un*, *ong*, *ui*, *uo* }.

{ *ia*, *iang*, *iong*, *üan*, *ün*, *ü*, *üe*, *i*, *ian*,
iao, *ie*, *in*, *ing*, *iu*, } $\Longrightarrow$
{ *ia*, *iang*, *iong*, *üan*, *ün*, *ü*, *üe*, },
{ *i*, *ian*, *iao*, *ie*, *in*, *ing*, *in* }.

The final partition of Chinese vowels is as follows:
{ *a*, *ai*, *an*, *ang*, *ao*, *e*, *ei*, *en*, *eng*, *ou*, *u* },
{ *un* *uan*, *ong*, *ui*, *uo* },
{ *ia*, *iang*, *iong*, *üan*, *ün*, *ü*, *üe* },
{ *i*, *ian*, *iao*, *ie*, *in*, *ing*, *in* },
{ $\hat{i}$},
{ *ua*, *uai*, *uang* },
{ *o* }.

We choose one element as the representative for the group, then the Chinese spelling rule can be represented by the following table:

	c	ch	g	f	b	d	n	j	y
a	1	1	1	1	1	1	1		1
un	1	1	1			1	1		1
ia							1	1	1
i					1	1	1	1	1
$\hat{i}$	1	1							
ua		1	1						
o				1	1		1		1

*Tab.*5.

Compare Tab. 2 with Tab. 5, we notice that there are 23 "missing"s. A missing of a combination does not mean that it is not possible. It only means that it happens there is no such a combination in the standard Chinese, but it is possible that this combination appears in other Chinese dialect.

5 Conclusion and Future Research

A formal model of taxonomy tree and an efficient algorithm for machine concept learning have been presented. As an example, by using this algorithm we rediscovered some important concepts in Chinese phonetics. Under the new taxonomy discovered by our algorithm, the rules of spellability of Chinese consonants and vowels appear very concise. The new taxonomy basically mat- ches the classical rules. Some novel parts suggest some idea to phonecticians.

This method can also be used to form concepts in many other fields. A computer might discover concepts like line, point, square, circle,..., by comparing many pictures, if we knew what were those fundamental features that could be used to distinguish or identify these pictures. This method might be particularly useful to natural language study. We may input many papers and documents into the computer, and let the computer to discover the concepts like verb, noun, pronoun, and so on, by comparing the context sensitivities between words. More and more subconcepts might be discovered, and a refined grammatical system of rules might be established.

References

[1] A. P. Danylik, "The Use of Explanations for Similarity-Based Learning", *Proc. of IJCAI 87* , pp.274-276, August, 1987.

[2] K. S. Murray, "Multiple Convergence: An Approach to Disjunctive Concept Acquisition", *Proc. of IJCAI 87* , pp.297-300, August, 1987.

[3] D. M. Dow, "An Introduction to The Pronunciation of Chinese", *Edinburgh* , November, 1972.

[4] "Xin Hua Dictionary", *The Commercial Press*, Beijing, China.