A Computational Best-Examples Model

Jianping Zhang[1]
Department of Computer Science
Utah State University
Logan, Utah 84322-4205 USA
Email: jianping@zhang.cs.usu.edu

Abstract

In the past, several machine learning algorithms were developed based on the exemplar view. However, none of the algorithms implemented the best-examples model in which the concept representation is restricted to exemplars that are typical of the concept. This paper describes a computational best-examples model and empirical evaluations on the algorithm. In this algorithm, typicalities of instances are first measured, then typical instances are selected to store as concept descriptions. The algorithm is also able to handle irrelevant attributes by learning attribute relevancies for each concept. The experimental results empirically showed that the best-examples model recorded lower storage requirements and higher classification accuracies than three other algorithms on several domains.

1. Introduction

Smith and Medin (1981) proposed the exemplar view for concept representation and category classification. Specifically, two cognitive models of the exemplar view, the proximity model and the best-examples model, were took up. In the proximity model, each concept is represented by all of its instances that have been encountered. The best-examples model assumes that the representation is restricted to exemplars that are typical of the concept. It seems impossible for an adult to remember all instances for each concept. The best-examples model strongly supports human concept formation. People tend to remember those most often encountered instances and forget those rarely encountered instances. Concepts involved in real world applications usually possess graded structures (Barsalou, 1985). Instead of being equivalent, instances of a concept may be characterized by a degree of typicality in representing the concept. Typical instances of a concept better characterize the concept than atypical instances. Typical instances represent the central tendency of a concept, so concepts described by typical instances are more human understandable than those described by atypical instances and also easier for human to capture the basic principles underlying these concepts.

In the past, several machine learning algorithms were developed based on the exemplar view, these learning algorithms are called instance-based learning algorithms, e.g., Protos (Bareiss, et al., 1990), IBL (Aha, et al., 1991), and Each (Salzberg, 1991). Although all these algorithms restricted the number of stored instances, none of them truly implemented the idea of the best-examples model. These algorithms selected misclassified instances which were proved to be near-boundary instances by Aha et. al (1991). Salzberg (1991) developed a method which assigned a weight to each stored instance. In his approach, typical instances got smaller weights than near-boundary instances, so they played more important role than near-boundary instances. However, this approach did not restrict stored instances to typical instances.

This paper presents a computational best-examples model developed from the cognitive best-examples model proposed in (Smith and Medin, 1981). Several problems were addressed in the computational best-examples model. First, an algorithm was developed to measure typicalities of instances. Second, an approach was designed to learn the weights of attributes for each class. Finally, an algorithm was proposed to select typical instances of a concept to store in memory. The computational model has been implemented and tested on both artificial and practical domains, and compared with three different instance-based learning algorithms: storing all instances, storing only incorrectly classified instances, and storing near-boundary instances. The empirical results showed that the

[1] This research was supported by the Department of Computer Science at Utah State University and the Utah State University Faculty Research Grant SCS-11107. The author would like to thank Steven Salzberg for providing the datasets of the malignant tumor classification and diabetes in Pima Indians.

computational best-examples model recorded lower storage requirements and higher classification accuracies than previous instance-based algorithms.

2. Learning Attribute Weights

Relevancies of attributes have a great impact on the performance of instance-based learning algorithms. Not all attributes chosen to describe a problem are relevant, even they do, the degrees of their relevancies differ. Different concepts in a problem may have different set of relevant attributes. For instance, an attribute that well distinguishes Concept1 from Concept2 may not do well to distinguish Concept2 from Concept3. In our model, the relevancies of attributes not only affect the classification of an instance, but also the typicality measured for each instance.

Both Aha (1989) and Salzberg (1991) assigned a weight to each attribute as its relevancy. Aha (1989) also assigned a different weight to the same attribute for different concepts. This is the approach used in our algorithm, but the weights were computed differently. In both (Aha, 1989) and (Salzberg, 1991), weights are computed incrementally. That is, each time a new instance was seen, the weights of attributes were modified based on the classification of the new instance made by the current descriptions. Weights were calculated during the process of instance selection in their algorithms. In our model, instances are selected according to their typicalities. Weights of attributes are used in measuring typicalities of instances, so we need the weights before selecting instances. Therefore, Aha's and Salzberg's methods cannot be used in our model. We use a statistical method to calculate the weights of attributes.

In our method, the weight of the attribute A with respect to the concept C is computed based on the difference of the distribution of the positive examples of C on all values of A and the distribution of the negative examples of C on all values of A. If the two distributions are very similar, the attribute A does not distinguish the concept C from other concepts well. In this situation, the difference of the two distributions is very small so the attribute gets a low weight (close to 0). If the two distributions do not intersect each other, the attribute A completely distinguishes C from other concepts. The difference of the two distributions in this situation reaches the maximum value so the attribute gets the largest weight. Generally, a more relevant attribute has a less intersection and a larger difference between the two distributions so it gets a larger weight.

Specifically, the weight of the attribute A which takes a value from {0, 1, ..., n} with respect to the concept C is computed by the following formula:

$$\frac{1}{2}\sum_{i=0}^{n} ABS\left(\frac{|\{e|A(e)=i \wedge e\in P\}|}{|P|} - \frac{|\{e|A(e)=i \wedge e\in N\}|}{|N|}\right)$$

where P and N are the sets of positive and negative examples of the concept C, respectively. $|\{e|A(e)=i \wedge e\in P\}|$ and $|\{e|A(e)=i \wedge e\in N\}|$ are the numbers of positive and negative examples whose value of the attribute A is i, respectively. The weight ranges from 0 to 1. if $\frac{|\{e|A(e)=i \wedge e\in P\}|}{|P|} = \frac{|\{e|A(e)=i \wedge e\in N\}|}{|N|}$ for all i ($0 \le i \le n$), The weight is 0 with respect to the concept C. IF one of $\frac{|\{e|A(e)=i \wedge e\in P\}|}{|P|}$ and $\frac{|\{e|A(e)=i \wedge e\in N\}|}{|N|}$ is 0 for all i ($0 \le i \le n$), The weight equals to 1 and the attribute A completely distinguishes C from other concepts.

3. Measuring Instance Typicalities

In our model, the typicality of an instance is measured based on its family resemblance (Rosch and Mervis, 1975), where family resemblance is defined as an instance's average similarity to other concept instances (intra-concept similarity) and its average similarity to instances of contrast concepts (inter-concept similarity). The more similar an instance is to other concept instances and the less similar it is to instances of contrast concepts, the higher its family resemblance, and the more typical it is of its concept. In other words, typical instances have higher intra-concept similarity and lower inter-concept similarity than atypical instances. The typicality of an instance is measured as the ratio of its intra-concept similarity to its inter-concept similarity. Thus, a larger intra-concept similarity implies a larger typicality, and a larger inter-concept similarity implies a smaller typicality. Generally, the typicalities of typical instances are much larger than 1, boundary instances have typicalities close to 1, and the instances with typicalities less than 1 are either noise or exceptions.

The intra-concept similarity of an instance of a concept C is computed as the average of the similarities of the instance to all other instances of C with respect to C, and the inter-concept similarity of an instance of a concept C is computed as the average of the similarities of the instance to all instances of contrast concepts (negative examples of C) with respect to C. The similarity of instances e^1 to e^2 with respect to C $sim(C, e^1, e^2)$ is the opposite of the distance of e^1 to e^2 with respect to C:

$$sim(C, e^1, e^2) = 1 - dis(C, e^1, e^2)$$

$dis(C, e^1, e^2)$ is computed by measuring the weighted Euclidean distance of the instance e^1 to the instance e^2. Specifically,

$$dis(C,e^1,e^2)=\frac{\sqrt{\sum_{i=1}^{m} W(i,C) * \left(\frac{e^1_i - e^2_i}{max_i - min_i}\right)^2}}{\sqrt{\sum_{i=1}^{m} W(i,C)}}$$

where e^j_i (j = 1, 2) is the value of the ith attribute on example e^j, max_i and min_i are respectively the maximum and minimum values of the ith attribute, and m is the number of attributes. W(i, C) is the weight of the attribute i with respect to the concept C. When the ith attribute is symbolic-valued, $e^1_i - e^2_i = 1$ if they are different, $e^1_i - e^2_i = 0$ otherwise. For missing values, $e^1_i - e^2_i = 0.5$. The distance of a linear attribute is normalized to the range of 0 to 1. The distance between two instances is also normalized to the range of 0 to 1.

4. Selecting Typical Instances

In the sections 2 and 3, we discussed the algorithms for computing weights of attributes and typicalities of instances. In this section, we shall first introduce the method for instance selection and classification, then present the complete instance-based algorithm in the computational best-examples model. The nearest neighbor algorithm stores all instances as concept descriptions. Aha et al. (1991) and Salzberg (1991) developed storage reduction instance-based learning algorithms in which only incorrectly classified instances were stored. Aha et al. (1991) empirically demonstrated that their storage reduction algorithm IB2 significantly reduced the storage requirements, and only slightly degraded classification accuracies. As indicated by Aha et al (1991), majority of stored instances by IB2 were near-boundary instances.

Similar to many IBL algorithms, the instance-based learning algorithm in our model stores a subset of training instances in its memory, and uses a distance measure to decide the distance between new instances and those stored. New instances are classified according to their closest neighbor's classification. The distance measure used is the one introduced in the section 3 with respect to the concept to which the stored instance belongs. Each time a new instance is incorrectly classified, our algorithm does not store the incorrectly classified instance itself, instead it stores the most typical instance which correctly classifies the new instance. That is, the algorithm finds the most typical instance such that after the instance is stored into the memory, the new instance can be correctly classified.

Similar to *Each* (Salzberg, 1991) and PEBLS (Cost and Salzberg, 1991), each stored instance is associated with a weight. The weight is used in measuring the distance between a new instance and the stored instance. The distance between a stored instance X of a concept C and a new instance Y is:

$$D(C, X, Y) = W_X * dis(C, X, Y)$$

where dis(C, X, Y) is the distance measure introduced in section 3, W_X is the weight of X, and C is the concept to which X belongs. Each stored instance covers an area in the instance space. The area covered by an instance depends on the distribution of all stored instances and the weight assigned to the instance. Generally, the smaller the weight of an instance, the larger the area covered by the instance. By changing the weight, one can change the area that the instance covers. Detailed discussion about the issue can be found in (Cost and Salzberg, 1991). The weight of an instance in our algorithm is simply the reciprocal of its typicality. The rationale for this is that a typical instance is more reliable than a boundary instance and should cover a larger area. Namely, it should have a smaller weight. An exceptional case should cover only a small area so it should have a large weight.

Specifically, our computational best-examples model is described as follows:

1. Compute weights of all attributes with respect to each concept,
2. Compute typicalities for all instances,
3. CD = null,
4. pick up the most typical incorrectly classified instance x, find the most typical instance y which correctly classifies x,
5. compute the weight of y: $weight(y) = \frac{1}{typicaity(y)}$,
6. add y to CD,
7. repeat the step 4, 5 and 6 until all instances are correctly classified.

We have implemented the algorithm in a system TIBL (Typical-Instance-Base Learning). To compare with other instance-based learning algorithms, we have also implemented three other instance-based learning algorithms, BIBL (Boundary-Instance-Based Learning), SRIBL (Storage Reduction Instance-Based Learning), and IBL (Instance-Based Learning). BIBL algorithm stores the lest typical instances, that is, exceptional and boundary instances. This algorithm repeats the process of finding the incorrectly classified instance with the smallest typicality and storing it until all instances are correctly classified. SRIBL is similar to IB2 (Aha, et al., 1991). It repeats the process of finding an incorrectly classified instance and storing it until all instances are correctly covered.

IBL is the 1-nearest neighbor algorithm and stores all training instances.

5. Empirical Evaluation

To empirically evaluate the typical-instance-based learning algorithm, we have conducted two kinds of experiments with TIBL. The first kind of experiments was designed to evaluate the algorithm in comparison with other instance-based learning algorithms, while the second kind of experiments was to evaluate the effect of learning attribute relevancies. The performance was evaluated on two aspects: classification accuracy and storage requirement. Classification accuracy was measured as the percentage of correct classifications made by the concept description on a set of randomly selected test instances. Storage requirement was measured by the number of instances stored in descriptions. All results reported in this section were averaged over 10 trials.

We applied the four instance-based learning algorithms: TIBL (Typical-Instance-Based Learning), BIBL (Boundary-Instance-Based Learning), SRIBL (Storage Reduction Instance-Based Learning), and IBL (Instance-Based Learning) to five domains: classification of n-of-m concept, classification of congressional voting recording, malignant tumor classification, diagnosis of diabetes in Pima Indians, and diagnosis of heart disease. In these experiments, TIBL was applied without attribute weight learning. Table 1 summaries the characteristics of the five domains and table 2 reports the experimental results of the four different instance-based learning algorithms on the five domains. Test sets were disjoint with training sets except for the n-of-m concept on which test set was the whole instance space. In table 2, ACC and #ins represent accuracy and the number of instances, respectively.

Domain	Training Set Size	Test Set Size	Number of Attributes
n-of-m	400	1024	10
Voting	200	235	16
Tumor	150	219	9
Diabetes	200	568	8
Heart	100	203	13

Table 1: Summary of Domain Characteristics

	TIBL		BIBL		SRIBL		IBL	
Domains	ACC(%)	#ins	ACC(%)	#ins	ACC(%)	#ins	ACC(%)	#ins
n-of-m	99.5	10.8	76.0	182.8	80.3	219.4	85.5	400.0
Voting	90.4	31.6	92.0	59.5	92.4	51.9	93.4	200.0
Tumor	93.1	19.5	90.4	29.4	91.2	28.8	93.7	150.0
Diabetes	70.2	105.6	66.5	106.9	65.5	105.3	69.9	200.0
Heart	82.0	33.7	73.9	46.6	75.6	45.2	77.8	100.0

Table 2: Experimental Results of 4 IBL Algorithms on 5 Domains

The n-of-m concept is an artificial domain and contains 10 binary attributes and 2 concepts, C1 and C2. If 5 or more of the 10 attributes of an instance are 1, then the instance belongs to C1, otherwise it belongs to C2. TIBL significantly improved both accuracy and storage requirements over BIBL, SRIBL and IBL. The reason for such a large improvement is that the n-of-m concept has a very clear graded structure. When the two most typical instances, 1111111111 and 0000000000, appeared in the training set, they were the only two instances chosen by TIBL, 1111111111 for C1 and 0000000000 for C2. These two instances were weighted differently, 1111111111 had a slightly smaller weight than 0000000000 so that 1111111111 covered larger area than 0000000000. The concept C1 did cover a larger area than C2. 100% accuracy was achieved by these descriptions. Following is an example of such descriptions.

1111111111: weight = 0.483
0000000000: weight = 0.523

The congressional Voting database contains the voting records of the members of the United States House of Representatives during the second session of 1984. It is described by 16 binary attributes and has 288 missing values among its 435 instances. TIBL's classification accuracy is slightly lower than BIBL's, SRIBL's and IBL's, but TIBL saved much fewer instances. An interesting result is that almost all descriptions generated by TIBL included only one or two instances with very high typicalities plus a number of instances with very low typicalities. Very

few instances with medium typicalities (1.2 to 3) were included in the descriptions. This is because that these instances were correctly classified by the typical instances stored. The typical instances of a description represented the central tendency, while the instances with low typicalities were exceptions which could not be correctly classified by any typical instances. The lower TIBL accuracy may be due to the fact that the test set included some exceptions which were not correctly classified by the typical instances stored.

The malignant tumor classification domain includes a set of 369 breast cancer patients, of which 201 have no malignancy and the remainder have confirmed malignancies (Wolberg and Mangasarian, 1989). The problem is to determine whether the tumors were benign or malignant from these cancer patients. Each patient is described by nine real-valued features. Mangasarian et al. (1989) applied a new linear programming technique to this domain, and good results have been achieved. Although the accuracy of IBL was slightly better than TIBL, it stored about 7 times more instances. TIBL outperformed both BIBL and SRIBL in terms of both accuracy and storage requirement, but the accuracy improvement is not significant. BIBL and SRIBL performed similarly. Similar to those obtained in the congressional voting records, the concept descriptions generated consisted of a few typical instances and a number of exceptional instances. These exceptional instances can be removed without degrading the accuracy.

The Diabetes in Pima Indians data set contains 768 instances, of which 500 (65%) have no diabetes, and 268 are diabetes patients. Each instance is described by 8 linear attributes. The problem is to diagnose who has diabetes and who has no. The accuracy of TIBL was consistently better than those of BIBL and SRIBL. The storage requirement of TIBL is about the same as those of BIBL and SRIBL. TIBL performed equally well as IBL in accuracy, while it reduced the storage by half.

The heart disease data set contains 303 instances, each instance is represented as 13 numeric attributes plus a classification: presence or absence of heart disease. 164 of the 303 instances have no heart disease. The goal is to learn to distinguish presence of heart disease from absence. Excellent results were obtained by TIBL on this domain. TIBL's classification accuracy was over 80% and higher than previously published results. Aha et al. (1991) reported 75.7% accuracy for standard nearest neighbor and 78% for a variant of NN that discards apparently noisy instances. They also reported that the C4 decision tree learning algorithm (Quinlan, 1987) achieved 75.5% accuracy. In our experiments, TIBL showed a significant accuracy improvement over the other three methods BIBL, SRIBL and IBL. It stored fewer instances than BIBL and SRIBL.

TIBL reduced the storage requirements dramatically on the datasets on which high accuracy were achieved by learning systems, e.g.,. congressional voting records and malignant tumor. This result was partially caused by the fact that high quality datasets enabled our algorithm to better distinguish typical instances from atypical ones. Another reason was that instances in high quality datasets are very concentrated and constitute few peaks which are well represented by a few typical instances.

To evaluate the effect of attribute weight learning, TIBL has been run on two domains, n-of-m concept and congressional voting, with and without attribute weight learning. The congressional voting dataset was the same as the one used in the experiments reported above. The n-of-m concept was modified by adding 5 irrelevant attributes. Table 3 presents the experimental results. Descriptions of n-of-m were tested on 2000 examples and descriptions of congressional voting records were tested 335 and 235 examples for the training sizes 100 and 200, respectively.

Domain	Training Set Size	With Attribute Relevancy		No Attribute Relevancy	
		ACC	#ins	ACC	#ins
n-of-m	200	95.6%	12.4	85.3%	60.4
	400	99.4%	7.5	89.0%	96.3
Voting	100	90.1%	12.6	88.7%	14.2
	200	91.3%	27.8	90.4%	31.6

Table 3: Experimental Results with and Without Learning Attribute Relevancies

Significant improvements on both classification accuracy and storage requirement were achieved on the domain of n-of-m concept. Although the improvements on congressional voting records were minor, they were stable. Improvement on accuracy was obtained on 19 of the 20 trials made over the two training set sizes and the improvement on storage requirement was observed for all 20 trials. These improvements were due to the attribute weight learning. Attribute weights not only helped TIBL in classifying new instances, but also in identifying typical instances, because attribute weights were used to compute typicalities of instances. For example, the typicalities of the most typical instances of n-of-m concept were around 1.25 without using attribute weights and were around 2.5 with using attribute

weights. The typicalities of the most typical instances of congressional voting records were around 2.8 without using attribute weights and were around 4.0 with using attribute weights.

6. Summary and Future Work

The main contribution of the work described in this paper is the development of a computational best-examples model from the cognitive best-examples model proposed by Smith and Medin (1981). Three algorithms, attribute weight learning algorithm, instance typicality measuring algorithm and instance selection algorithm, were developed in this computational model. This model was empirically evaluated and compared with other instance-based learning algorithms. The results confirmed that the best-examples model can be adopted in developing instance-based learning systems. The results showed that when concepts have graded structures instances-based learning systems developed best-examples model may outperform other instance-based learning systems. The computational model may also help cognitive researchers to better understand the best-examples model.

One of the limitations of the computational model is the way to compute the distance of instances when attributes are symbolic-valued. In this case, distance in TIBL is computed by counting the attribute values that match. As indicated in (Cost and Salzberg, 1991), this approach for computing distance may not perform well when the domains are complex. In the future, we shall implement a more complicated method called Value Difference Metric (VDM) (Stanfill and Waltz, 1986; Cost and Salzberg, 1991) which takes into account the overall similarity of classification of all instances for each possible value of each attribute. In this method, a matrix defining the distance between all values of an attribute is derived statistically, based on the examples in the training set. Other future work includes developing a method for learning weights of linear attributes, especially continuous attributes. The problem of classifying new instances with degrees of membership should be addressed in the future too.

References

Aha, D., "Incremental, instance-based learning of independent and graded concept descriptions," Proceedings of the Sixth International Workshop on Machine Learning, Ithaca, NY, 1989.

Aha, D. and Kibler, D., "Noise-tolerant instance-based learning algorithms," Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI, 1989.

Aha, D., Kibler, D, and Albert, M., "Instance-Based Learning Algorithm," *Machine Learning* 6, 1991

Bareiss, E. R., Porter, B. W., and Wier, C. C., "Protos: An Exemplar-Based Learning Apprentice," *Machine Learning: An Artificial Intelligence Approach V III,* 1990.

Barsalou, L., "Ideals, central tendency, and frequency of instantiation as determinants of graded structure in categories," in Journal of Experimental Psychology: Learning, Memory and Cognition, 11. 1985.

Cost, S., and Salzberg, S., "Q weighted nearest neighbor algorithm for learning with symbolic features," Technique report, Department of Computer Science, The Johns Hopkins University, 1991.

Mangasarian, O., Setiono, R., and Wolberg, W., "Pattern recognition via linear programming: theory and application to medical diagnosis," Technical Report #878, Computer Science Department, University of Wisconsin-Madison, 1989.

Quinlan, J. R., "Simplifying decision trees." In *International Journal of Man-Machine Studies*, vol. 27, 1987.

Rosch, E. and Mervis, C. B., "Family Resemblances: Studies in the Internal Structure of Categories." In *Cognitive Psychology*, vol. 7, 1975.

Salzberg, S., "A nearest hyperrectangle learning method," Machine Learning, 6:3, 1991.

Smith, E. E., Medin, D. L., *Categories and Concepts.* Harvard University Press, 1981.

Stanfill, C. and Waltz, D., "Toward memory-based reasoning," Communications of the ACM, 29:12, 1986.

Wolberg, W. and Mangasarian, O., "Multisurface method of pattern separation applied to breast cytology diagnosis," Manuscript, Department of Surgery, Clinical Science Center, University of Wisconsin, Madison, WI, 1989.