P93-27

WORKING NOTES

AAAI
FALL SYMPOSIUM SERIES

Symposium:
Machine Learning in Computer Vision: What, Why, and How?

Program Committee:
Kevin Bowyer, University of South Florida, Cochair
Lawrence Hall, University of South Florida, Cochair
Chris Brown, University of Rochester
Bruce Draper, University of Massachusetts
Tom Mitchell, Carnegie-Mellon University
Dean Pomerleau, Carnegie-Mellon University
Larry Rendell, University of Illinois

OCTOBER 22, 23, 24, 1993

SHERATON IMPERIAL HOTEL & CONVENTION
CENTER, RALEIGH, NORTH CAROLINA

1993 Fall Symposium Series

Machine Learning in Computer Vision: What, Why, and How?

Working Notes

(Distribution limited to symposium attendees. Not for citation.)

October 22 - 24, 1993
Sheraton Imperial Hotel & Convention Center
Raleigh, North Carolina

Sponsored by the
American Association for Artificial Intelligence

Table of Contents

Late Papers:

THE PRAX APPROACH TO LEARNING A LARGE NUMBER OF TEXTURE CONCEPTS

J. Bala, R. Michalski, and J. Wnek
Center for Artificial Intelligence
George Mason University
Fairfax, VA 22030
{bala, michalsk, jwnek}@aic.gmu.edu

Abstract

This paper describes an approach, called PRAX, to learning descriptions of a large number of texture concepts from texture samples. The learning process consists of two phases: 1) learning descriptions of a selected subset of texture classes, called *pr*incipal ***axes*** (briefly, *praxes*), and 2) learning descriptions of other classes (*non-prax* classes), by relating them to the praxes. Descriptions of non-prax classes are expressed in terms of the similarities to praxes, and thus the second phase represents a form of analogical learning. While the first phase is done as a one-step learning process, the second phase is performed as an incremental learning process. The method was applied to learning texture concepts from texture samples, and illustrated by an experiment on learning 24 texture classes, using a subset of 8 classes to learn praxes. After acquiring all texture descriptions from samples taken from a training area, the implemented program, PRAX-2, recognized texture samples from the testing area without a single error.

Introduction

Most research on concept learning from examples concentrates on algorithms for generating concept descriptions of a relatively small number of classes. In conventional methods, when the number of classes if growing, their descriptions become increasingly complex. For example, Figure 1 depicts the growth of the complexity of class descriptions (measured by the number of rules) with the number of classes and the number of training examples. The results were obtained from texture data using eight attributes per example and applying the AQ rule learning method [Michalski et al., 1983]. While increasingly complex descriptions are usually needed to cover more training examples, the predictive accuracy of such descriptions on new examples may actually decrease. This is due to the so-called overfitting effect (Bergadano et al., 1992). This effect may be particularly pronounced in the case of learning texture descriptions from texture samples, because of the highly disjunctive nature of such descriptions.

In some computer vision applications, the number of classes may be very large, and they may not be known entirely in advance. Therefore, in such situations, the learning method must be able to learn incrementally new classes. Such a *class-incremental* mode is different from the conventional *event-incremental* mode, in which examples of classes are supplied incrementally, but the set of classes remains unchanged.

This paper presents a learning method that is specifically oriented toward learning descriptions of a large number of classes in a class-incremental mode. The learning process consists of two phases. In Phase 1, symbolic descriptions of a selected subset of classes, called *pr*incipal ***axes*** (briefly, *praxes*) are learned from concept examples (here, samples of textures). The descriptions are expressed as a set of rules. In Phase 2, the system incrementally learns descriptions of other classes (*non-prax* classes). These descriptions are expressed in terms of the similarities to praxes, and thus the second phase represents a form of analogical learning. To utilize a uniform representation, the prax descriptions are also transformed into a set of similarities to the original symbolic descriptions.

The basic idea of the method is illustrated in Figure 2. Suppose that the system already learned the concepts of "orange" (Des1) and "lemon" (Des2). A new concept "grapefruit" can be learned in terms of basic properties (Des3'), the same way as the previous concepts (orange and lemon), or in terms of similarities (and/or dissimilarities) between the new concept ("grapefruit") and previously learned concepts (Des3").

Method

As mentioned earlier, the underlying idea of the PRAX approach is to determine descriptions of a selected class of basic concepts called principal axes (praxes or PXs), and then describe all classes in terms of relations to the prax descriptions. An early version of the method was described in (Michalski et al., 1993).

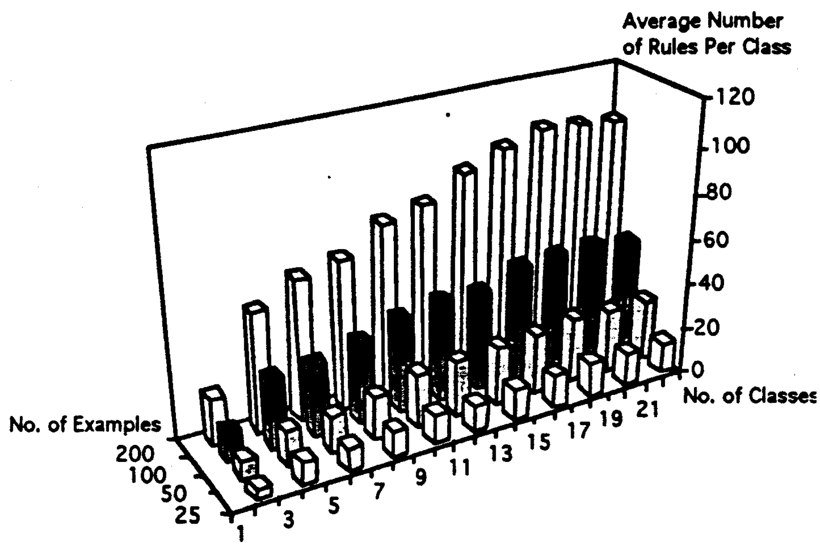

Figure 1: Average number of rules per class

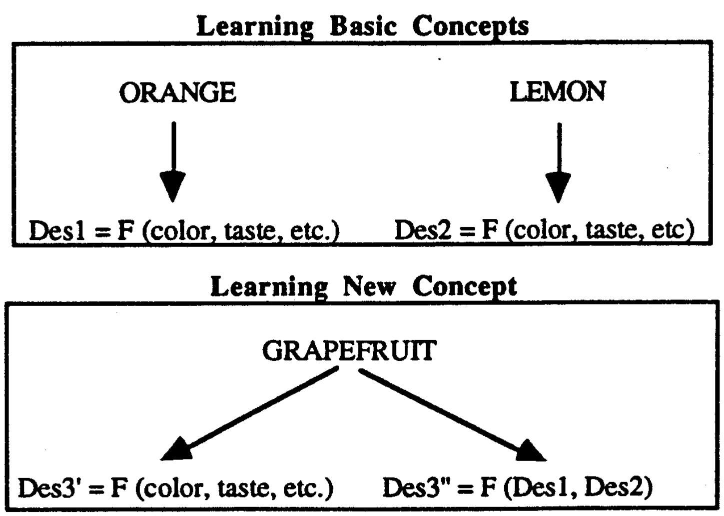

Figure 2: Two different ways of learning concept "grapefruit"

Prax descriptions are learned using the AQ-type rule learning program (specifically, AQ-15). The program learns descriminant descriptions of praxes from given examples (Michalski, 1983). Specifically, the AQ program is used to generate a The concept descriptions are represented in VL_1, which is a simple version of the Variable-Valued Logic System, In the application of the learning method to texture recognition, a concept description characterizes a single texture class. The description (also called a cover) is in disjunctive normal form, and is equivalent to a set of rules. Below is an example of a cover generated by the AQ program for some texture class:

[x1=10..54] & [x3=18..54] & [x5=11..17] & [x6=6] *or*
[x3=18..53] & [x4=16..54] & [x6=0..6] & [x8=5..12]

The above cover consists of two disjuncts (rules). Each rule describes one conjunctive set of conditions for classifying a texture sample to the given class. Attributes x1 to x8 represent certain measurements of a texture sample (in experiments presented in this paper x1 is the Laplacian edge operator, x2 is the frequency spot, x3 is the horizontal edge operator, x4 is the vertical edge operator, x5 is the horizontal V-shape operator, x6 is the vertical V-shape operator, x7 is the vertical line operator, and x8 is the horizontal line operator). For example, suppose that a vector of values of eight attributes characterizing an unknown texture sample is <20, 10, 25, 17, 1, 4, 30, 6>. Such a vector, called an event, satisfies the second rule (disjunct), because the attribute values specified in the event are within the ranges indicated by the rule (e.g., x3=25 is within the range 18..53; x5=1 satisfies the second rule because there is no condition in it for x5),

Prax descriptions could be viewed as constructed intermediate attributes. Therefore, the PRAX method can be viewed as a form of constructive induction (Michalski, 1978). Once the prax descriptions have been determined, all concept descriptions are related to them. Specifically, given examples of some concept, the system determines a similarity vector (SV) for that concept, in which each component represents the average degree of similarity between the concept examples and PX. A degree of similarity is obtained by calculating the distance in the attribute space from an example of a concept to a single rule in prax description. The method uses a non-linear distance metric to calculate values of new attributes. The distance metric is based on the idea of flexible matching. In flexible matching, the degree of closeness between the example and the concept is determined, instead of a binary decision as used in strict matching. Specifically, the match of an example E to a disjunct D is computed by the following formula:

$$MATCH(E,D) = \prod_{i}\left(1 - \frac{dis(E_i,D_i)}{\max_i - \min_i}\right)$$

where E_i is the value of the i-th attribute of example E, D_i is the condition involving the i-th attribute in D, $\max_i$ and $\min_i$ are the maximum and minimum values of the i-th attribute, and m is the number of attributes. The term $dis(E_i, D_i)$ depends on the type of the attribute involved in the condition. An attribute can be one of two types: nominal and linear. In a nominal condition, the referent in a condition is a single value or an internal disjunction of values, e.g., [color = red v blue v green]. The distance is 1, if such a condition is satisfied by an example, and 0 if it is not satisfied. In a linear condition, the referent is a range of values, or an internal disjunction of ranges, e.g., [weight = 1..3 v 6..9]. A satisfied condition returns the value 1 for distance. If the condition is not satisfied, the distance between an example and the condition is the absolute of a difference between the value of the example and the nearest end-point of the interval of the condition (normalized by the distance between the farthest value and the condition). For example, if the domain of x is [0 .. 10], the value of x for the example E is $E_x=4$ and the condition is [x = 7 .. 9], then

$$\text{dis}(E_x, \text{condition}) = \frac{7 - 4}{10 - 0} = \frac{3}{10}$$

The flexible match method as described above is used in generating the similarity vector (SV) description, i.e. the concept description expressed in the new representation space. The SV description is obtained by applying the flexible matching process to the examples of the new concept and the previously learned Principal Axes. Entries in the SV vector of a given class represent average flexible matches (normalized to range 0 to 100) for all examples of that class to PXs.

In experimental testing of the method on the problem of learning descriptions of a large number of visual textures, PRAX significantly outperformed the k-NN classifier often used for such problems [Bala et al., 1992].

PRAX-2

The method described here (Figure 3) extends the initial PRAX method by making it more space-efficient. This is accomplished by reducing the number of PXs in the changed representation. The selection or deletion of a given PX is based on its discriminatory power, measured as the standard deviation of its values through all classes. In experiments with 24 texture classes depicted in Figure 5 (100 training examples per class and 100 testing examples per class) the number of PXs generated from the initial 8 classes was reduced from 170 to 17. Thus, all 24 classes were recognized using only 17 PXs (rules). Figure 4 shows examples of one PX expressed as the conjunction of attribute conditions and one of a class description (SV) expressed as the vector of 17 similarity measures.

The ability of the method to describe many classes while using a small set of rules, is a promising result obtained in the initial experiments. The main strength of the method lies in a problem-relevant transformation of the descriptor space. The new descriptors form generalized sub-spaces of the initial training space.

PRAX-2 does not have the mechanism to decide how to choose basic concepts. Choosing the minimal subset of concepts to be used for principal axes generation is crucial for method optimization. The new version of the method (PRAX-3) with automatic derivation of minimal subsets of concepts is being currently developed.

Given:

P - set of principal axes found by PRAX $\{PX_1, PX_2, ..., PX_M\}$
M - number of principal axes
K - number of training classes
ϑ - maximum discriminant standard deviation

Do:

For each $PX_i \in P$
    For each class k
        Calculate average match to examples of class k (S_{ik})
For each $PX_i \in P$
    Calculate a standard deviation $\sigma_i = \sigma\{S_{ik},\ k=1,.., K\}$

For each $PX_i \in P$
    If $\sigma_i < \vartheta$ then remove PX_i from P

Figure 3: Algorithm for finding a minimal set of principal axes

PX == rule =>[x1=8..21] & [x3=15..22] & [x4=24..42] & [x5=19..37] & [x6=28..36] & [x7=28..36] & [x8=12..25]

SV(class C16) => [5.1, 57, 21, 51, 41.2, 6.2, 0.4 6.3, 0.4, 9.6, 95, 94, 89, 93, 87, 28, 0.3]

Values in the SV vector represent a normalized (range 0 to 100) average match of examples of the C16 class to 17 PXs

Figure 4: Examples of a PX and a class description

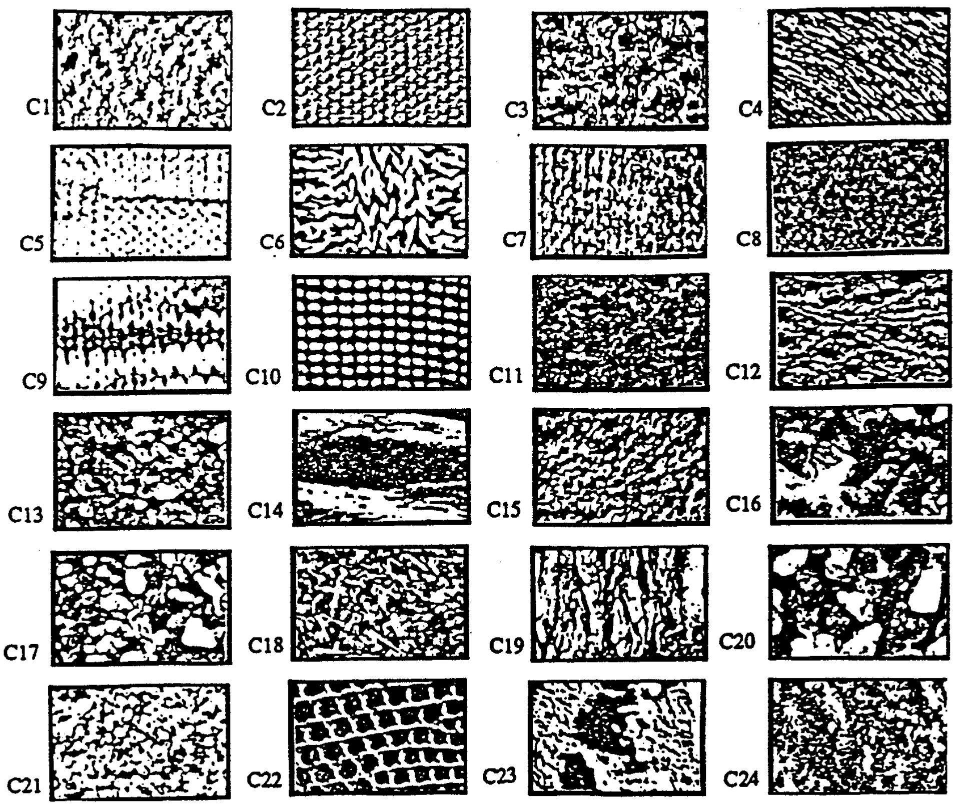

Figure 5: Texture classes (C1 to C8 used to learn praxes)

Acknowledgment

This research has been done in the George Mason University Center for Artificial Intelligence and has been supported in part by the Defense Advanced Research Projects Agency under the grant administered by the Office of Naval Research No. N00014-87-K-0874 and No. N00014-91-J-1854, in part by the Office of Naval Research under grants No. N00014-88-K-0397, No. N00014-88-K-0226 and No. N00014-91-J-1351, and in part by the National Science Foundation Grant No. IRI-9020266.

References

Bala, J., R. Michalski and J. Wnek, "The Principal Axes Method for Constructive Induction," *The Ninth International Conference on Machine Learning*, Aberdeen, Scotland, 1992.

Bergadano, F., S. Matwin, R. S. Michalski and J. Zhang, "Learning Two-tiered Descriptions of Flexible Concepts: The POSEIDON System," *Machine Learning*, Vol. 8, pp. 5-43, 1992.

Michalski, R. S., "A Theory and Methodology of Inductive Learning," *Artificial Intelligence*, Vol. 20, pp. 111-116, 1983.

Michalski, R.S., "Pattern Recognition as Knowledge-Guided Computer Induction," Report No. 927, Department of Computer Science, University of Illinois, Urbana, June 1978. (Published under the title "Pattern Recognition as Rule-Guided Inductive Inference," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 4, pp. 349-361, July 1980).

Michalski, R., Bala J., Pachowicz P. "GMU RESEARCH ON LEARNING IN VISION: Initial Results." *Proceedings of the 1993 DARPA Image Understanding Workshop*, Washington DC, 1993.