1995

Image Classification using Version Spaces

Edit J. Kaminsky
University of New Orleans, ejbourge@uno.edu

Willie G. Brown
Jackson State University

Image Classification using Version Spaces

Edit J. Kaminsky
Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148
tel: (504) 286-5616 fax: (504) 286-3950
ejbee@uno.edu

Willie G. Brown
Department of Computer Science
Jackson State University
Jackson, MS 39217

Abstract

This paper presents the candidate elimination implementation of the version space strategy for classification of photographic data. It is shown that very accurate classification is easily achieved and that only a small number of training samples are needed to generate the rules.

1 Introduction

Vast amounts of remotely sensed data are collected to gather information about natural resources by scanning the surface of the earth. Low and high resolution sensors are currently in use and airborne photographic data is also collected. Identification, classification, and interpretation of information contained in these data are performed by experienced and skilled personnel for tasks such as environmental monitoring and disaster relief. The research reported here focuses on the utilization of a modified version space strategy to classify remotely sensed data according to land use/cover type as an effort to use Artificial Intelligence (AI) techniques to automate image classification. The candidate elimination implementation of the version space method for concept acquisition is applied to classification of airborne photographic data. The version space approach is shown to learn rules for describing land use/cover classes in terms of pixel attributes.

Two key enhancements to the basic version space algorithm are described in this paper: 1) a method for using partially-learned concepts as rules for classification and 2) a method for handling disjunctive descriptions. Test results over several data sets, including both spectral and textural information, show that the version space system provides highly accurate classification results that are only slightly inferior to those of neural networks [1]. However, the rules produced by the version space method are symbolic and easily understood by humans. Also, the version space classifier requires fewer training instances to learn a set of rules and needs to process the training set far fewer times than the neural classifier. No work on image classification using version spaces, other than that of the current authors [2, 3], has been reported.

2 The Photographic Data

An aerial photograph taken with a Zeiss color infrared camera and digitized to about one-meter (1 m) resolution using an Iconic digital scanner which filtered the original into three bands for false color generation is used. The data corresponds to an area of about 1 Km by 1 Km from Southeastern Louisiana that includes a distilling, storage, and pumping facility for natural gas. It is located across from Pilot Town in the Garden Isle district of the Mississippi River. The five classes in the photographic image are: grass, forest, stagnant water, flowing water, and urban. All values are normalized to the range (0.0, 1.0) before processing.

3 Texture

Image texture describes the primitives that compose the texture and the spatial dependence or interaction between them. Haralick [4] presented the Spatial Gray Level Dependence (SGLD) procedure for extracting 14 features from blocks of digital image data. This method was selected to create a data set that includes textural information which is combined with the original spectral data. The texture information in the SGLD method is specified by co-occurrence matrices. A principal components transformation is performed on the set of SGLD features and the most prominent component from each of the three bands is used.

4 Version Space Classifiers

Mitchell [5] posed concept acquisition as a search problem and presented a version space strategy for representing and searching the space of all possible descriptions for a given concept. The problem is presented formally as follows:

- *Given:*
  1. A language in which to describe instances
  2. A language in which to describe generalizations
  3. A matching predicate that matches generalizations to instances
  4. A set of positive and negative training instances of a target generalization to be learned.
- *Determine:*
  Generalizations within the provided language that are consistent with the presented training instances (*i.e.*, plausible descriptions of the target generalization).

Learning a description for a concept consists of finding the correct description by candidate elimination. Examining pre-classified training instances, one at a time, allows the elimination from further consideration of all descriptions which either describe a non-member or do not describe a member. After each new instance is processed, the remaining descriptions constitute the *version* of the description *space* that correctly classifies all of the instances examined so far. Only the correct description will remain after a discriminating sequence of instances has been observed. Two boundary sets, S and G, can be used to concisely represent a given version space. The set S always contains the most specific descriptions still plausible, while the G set always contains the most general remaining candidates.

With the version space implementation of our image classifier, rules are learned independently for each class, all attributes for our classification problem are linear (i.e., attributes can take linearly ordered sets of mutually exclusive values), and the land use/cover classes can be specified by k-disjunctive normal form (k-DNF) concepts.

Feature vectors are presented and processed by candidate elimination until one of several termination conditions occurs. The ideal termination condition is when the version space converges to the single correct description for pixels of the target class. However, several factors related to the training data may prevent complete convergence. First, there may not be enough training instances. In this case, the version space reaches a state that is consistent with all training instances but that still contains more than one description. Second, there may be no pure conjunctive description that describes all members of a given class and excludes all non-members. This condition is detected when none of the descriptions in the G set describes a positive instance. Some form of disjunction is necessary in this case. Third, there may be no pure conjunctive description that excludes all non-members of the target class. This condition may be due to noise in the training data or insufficient expressiveness of the description language. Since the instance space here is linear-attribute based, and any range can be described, a conflict of the third type must be the result of noise. Specifically, the problem arises when a version space reaches a state that classifies a negative instance as positive. If this happens, some instance has caused the S set description to become general enough to include the negative instance. Then, if the classes are separable, one of the instances must be classified incorrectly. In our work, disjunctive concepts are handled similar to Murray's "multiple convergence" [6]. An additional version space is created whenever a positive instance is not included in any of the current version spaces. When the termination condition is reached, the S set descriptions of the final set of version spaces constitute a disjunctive description of members of the target class. When noise is detected in this task, Haussler's ideas [7] are implemented. To resolve conflicts when rules for two or more classes claim the same instance, the rule with the strongest claim prevails. Rule R1 has a stronger claim to an instance than rule R2 if the percentage of class 1 instances claimed by R1 is higher than the percentage of class 2 instances claimed by R2. When no rule claims an instance, the distance of the instance from each rule is computed and the rule that is closest to the instance claims it.

5 Results

Our evaluation of classifier performance is based primarily on classification accuracy in the form of confusion tables which provide information about the accuracy of classifiers and insight into the confusion between classes. The number of instances that need to be processed to generate the version space or rules is also an important measure.

5.1 Accuracy

Table 1: Accuracy of the RGB VS classifier (%).

Class	Grass	FW	Urban	SW	Forest
Grass	**92.39**	0.00	0.00	0.05	7.56
FW	13.32	**83.22**	3.38	0.08	0.00
Urban	2.51	0.00	**97.49**	0.00	0.00
SW	0.43	3.14	0.00	**95.69**	0.75
Forest	0.11	0.00	0.00	2.80	**97.10**

6.40% incorrect overall

Test results show that the version space system exhibits excellent performance, only slightly inferior to that of neural networks [1]. Table 1 shows the confusion table for a version space classifier using RGB features only. Table 2 is the confusion table for the version space classifier based on spectral and textural values (one principal component per band).

Table 2: Accuracy of Textural VS Classifier (%).

Class	Grass	FW	Urban	SW	Forest
Grass	**99.82**	0.00	0.00	0.00	0.18
FW	16.70	**82.49**	0.40	0,40	0.01
Urban	16.94	0.00	**83.06**	0.00	0.00
SW	0.62	2.83	0.00	**94.56**	1.99
Forest	7.42	0.00	0.00	3.58	**89.00**

6.64% incorrect overall

Notice there is a 7.5 % increase in classification accuracy of the grass class when principal component textures are added, but a 14% decrease in accuracy for the urban pixels, so whether to incorporate or exclude textures for classification with version spaces depends on the particular classes. Comparing our results to those of the neural systems presented in [1] we notice that the version space strategy provides an alternative to neural networks to accurately classify images by land type/cover.

5.2 Speed

The number of instances processed while generating the version space is important because reduction in number means less ground truth is needed, thus reducing expenses and human involvement. When pre-computed textures alone were used, only 99 instances had to be processed to generate the version space and corresponding rules. When both spectral and textural values were used, 460 instances were needed. For the sliding window, where a neighborhood of pixels is considered when classifying the central pixel (see [1]), only 169 instances had to be processed. It must be stated, too, that given that the training sets are small, it is important to constitute it with the "best" samples; the training set will influence the classification results slightly.

As an example of the rules generated by the version space, for class "grass" we have: The pixel belongs to class "grass" if:

$0.121 < red \leq 0.412$

$0.195 < green \leq 0.455$

$0.077 < blue \leq 0.325$

6 Conclusions

The candidate elimination implementation of the version space strategy was applied to the classification of high resolution filtered photographic data, and shown to classify data quickly and accurately and provide useful information in the form of symbolic classification rules. Few training instances are needed to learn a set of rules and the training set needs to be processed only once. The version space classifier with pure RGB values yields accuracies as high as 98 % (depending on the training set), while the classifier based on spectral and textural values (one principal component per band) yields 95 % accuracy, but a 7.5 % increase in accuracy is achieved for the class "grass" when principal component textures are added. Very few training instances are needed to generate all classification rules.

References

[1] E. J. Kaminsky and H. S. Barad, "Image classification using textural neural networks," in *SIP'95 Proceedings of the 1995 Signal and Image Processing Conference (in print)*, (Las Vegas, Nevada), November 20–23, 1995.

[2] E. J. Kaminsky, D. Miller, and S. Rana, "Neural network classification of remote sensing data," *Computers and Geosciences*, vol. 21, pp. 377–386, April 1995.

[3] W. Brown, H. Barad, and E. Kaminsky, "A comparison of neural networks and version spaces for classification of remotely sensed spatial data," in *NASA/American Soc. Eng. Education (ASEE Summer Faculty Fellowship Program 1994* (J. E. Miller and A. Joyce, eds.), Sept. 1994.

[4] R. M. Haralick and K. Shanmugam, "Combined spectral and spatial processing of ERTS imagery data," *Remote Sensing of Environment*, vol. 3, no. 13, pp. 3–13, 1974.

[5] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, pp. 203–226, 1982.

[6] K. S. Murray, "Multiple convergence: An approach to disjunctive concept acquisition," in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 1, pp. 297–300, IEEE, 1987.

[7] D. Haussler, "Quantifying inductive bias: Ai learning algorithms and valiant's learning framework," *Artificial Intelligence*, vol. 36, pp. 177–221, 1988.