Proc. Fifteenth International Conference on Pattern Recognition (ICPR 2000), 3-8 September 2000, Barcelona, Spain.

Feature Learning for Recognition With Bayesian Networks

Justus H. Piater and Roderic A. Grupen
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{piater|grupen}@cs.umass.edu

Abstract

Many realistic visual recognition tasks are "open" in the sense that the number and nature of the categories to be learned are not initially known, and there is no closed set of training images available to the system. We argue that open recognition tasks require incremental learning methods, and feature sets that are capable of expressing distinctions at any level of specificity or generality. We describe progress toward such a system that is based on an infinite combinatorial feature space. Feature primitives can be composed into increasingly complex and specific compound features. Distinctive features are learned incrementally, and are incorporated into dynamically updated Bayesian network classifiers. Experimental results illustrate the applicability and potential of our approach.

1. Introduction

During the past decade, considerable progress has been made in the area of machine recognition [3, 8, 2, 4]. Increasingly impressive recognition results are reported on large databases of various objects. While this success is truly remarkable, we observe that most work in this field shares certain characteristics: First, there exist a set of training images, which are complete at the outset and unchanged during the learning phase. In fact, many current recognition algorithms critically depend on the accessibility of the training set in its entirety. Second, most algorithms are based on descriptions of objects or classes in isolation, without regard to objects belonging to other classes, as opposed to descriptions of distinctions between objects. Thus, the properties of these descriptions largely predetermine the capabilities of the algorithm to generalize and to recognize minute distinctions. The assumption behind this design is that in terms of the description employed by the algorithm, all objects within a class are more similar to each other than to objects belonging to other classes.

Task domains that share these two characteristics we call *closed*. Are practically occurring recognition problems closed domains? We argue that many realistic visual recognition problems constitute *open* task domains: To the agent, the number and nature of visual categories or object classes is not initially known. A given class may contain dissimilar objects, some of which may be very similar to – but distinguishable from – objects belonging to other classes. There is no fixed set of training images that perfectly describes the classes and is observable in its entirety. Most existing algorithms for visual recognition are not well suited for open task domains. Extending our previous work [6], this paper presents a framework for visual learning that constitutes progress toward this goal.

2. Features

In order to learn distinctions at various levels of detail which are initially unknown, a very large feature space is required, along with a method of generating distinctive features from this space. We generate such a feature space by defining *primitive* features that can be combined into *compound* features according to a set of rules, described below. The size of a feature, i.e. the number of primitive features contained in a (compound) feature, provides a partial order of the feature space. Our feature learning procedure generates candidate features by randomly sampling from this space in a manner that prefers small features, and considers larger features as required (Section 4).

Primitive features are local appearance descriptors represented as vectors of local filter responses. The filters are oriented derivatives of 2-D Gaussian functions, with orientations chosen such that they form a steerable basis [1]. Here, the steerability property permits the efficient computation of filter responses at any orientation, given $d + 1$ measured filter responses for the dth derivative at specific orientations. We exploit this property to achieve rotational invariance at negligible computational overhead.

This work was supported in part by the National Science Foundation under grants CISE/CDA-9703217, IRI-9704530 and IRI-9503687, by the Air Force Research Labs, IFTD (via DARPA) under grant F30602-97-2-0032, and by Hugin Expert A/S through a low-cost PhD license of their Bayesian network library.

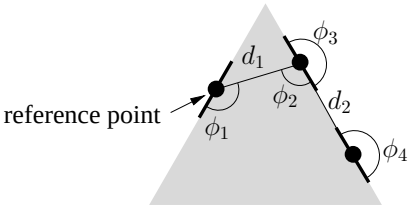

Figure 1. A geometric feature of order 3, composed of three primitives. The feature is defined by the angles ϕ and the distances d. Each primitive is either an edgel or a texel.

Our current system employs two types of primitive features [6]: An *edgel* is given by the two first-derivative Gaussian basis functions (G_x and G_y), encoding the local intensity gradient. A *texel* is represented as an 18-vector containing the responses to the basis filters of the first three derivatives at two scales. This encodes a local texture signature [7].

Primitive features by themselves are not very discriminative. However, spatial combinations of these can express a wide range of shape and texture characteristics at various degrees of specificity or generality. We employ the following four complementary types of feature composition:

- *Geometric* relations are given by the relative angles and distances between the participating lower-order features (Figure 1). Geometric features are useful for representing e.g. corners, angles, and collinearity.
- *Topological* relations here refer to relaxed geometric relationships between component features that allow some degree of variability in angles and distances. Topological compound features are more robust to viewpoint changes than are geometric features, at the expense of specificity.
- *Conjunctive* features assert the presence of all component features without making any statement about their geometric or topological relationship.
- *Disjunctive* features are considered to be present in a scene if at least one component feature is detected. This can express statements such as "If I see a dial *or* a number pad, I may be looking at a telephone."

Features are computed at various scales, generated by successively subsampling images by factor two. This achieves some degree of scale invariance. In addition, many features are insensitive to minor changes in scale (see for example Figure 1).

Each feature f is present at a pixel location p to a degree $s_f(p) \in [0, 1]$, which is computed by correlating the vector of applicable filter responses at p with the pattern vector defining f. A feature is present in an image I to the degree $s = \max_{p \in I} s_f(p)$. For more detail on our features, see our earlier work [6].

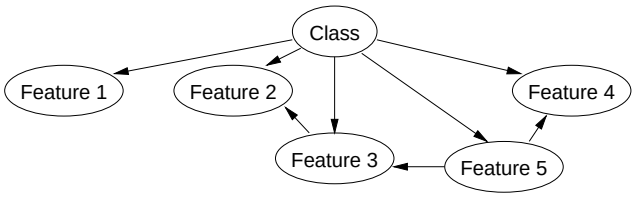

Figure 2. A Bayesian network for one class. A network such as this is created for each class.

3. Bayesian networks for recognition

In a Bayesian network, each node represents a random variable. The network structure specifies a set of conditional independence statements: The variable represented by a node is conditionally independent of its non-descendants in the graph, given the values of the variables represented by its parent nodes. In our scenario, each class is modeled as a separate Bayes net. The presence of an object is modeled as a discrete random variable with two states, *true* and *false*. The presence of an object gives rise to observable features, which are represented by random variables whose distributions are conditioned on the presence of an object of this class. Feature observations are entered into the net by setting the corresponding feature nodes to the observed values. Then, updated posterior probabilities are propagated throughout the Bayes net [5].

If some features are not independent, corresponding arcs must be inserted between the appropriate feature nodes. For example, in Figure 2, Feature 3 may be a geometric composition with Feature 2, which is also in the feature set. Then, the presence of Feature 3 in an image implies the presence of Feature 2. Thus, in the Bayes net there is an arc from node 3 to node 2. An analogous argument holds for topological and conjunctive features, such as Feature 5 in Figure 2, which combines Features 3 and 4. In the case of disjunctive features, the direction of the argument (and that of the additional arrows) is reversed.

Discretizing features. Recall from Section 2 that the feature variables are continuous. Since most theory on belief propagation in Bayesian networks applies to discrete random variables only, we split each feature variable into two bins, corresponding to "present" and "not present". The threshold is determined individually for each feature variable such that its discriminative power is maximized. The discriminative power of a feature variable given a threshold is measured in terms of the Kolmogorov-Smirnoff distance (KSD). The KSD between two conditional distributions of a random variable is the difference between the cumulative probabilities at a given value of this variable under the two conditions. This separates the instances of the two conditions optimally, in the Bayesian sense, using a single cut-point.

Recognition. Conventionally, one would compute all feature values, propagate evidence, and report, say, the class with the highest posterior probability as the recognition result. Instead, we compute features one by one and update the Bayesian network after incorporating each feature. Features are processed in decreasing order of informativeness, defined by the mutual information between a feature and the class nodes, i.e. its potential to reduce the entropy in the class random variables. In practice, only a fraction of all features are computed because the entropies of the class variables diminish before all features have been queried. This phenomenon suggests a straightforward, but very effective *forgetting* procedure: We delete any features that cease to be used during recognition.

4. Adaptive feature generation

For simplicity, we restrict the following discussion to an incremental supervised-learning scenario: The learning system is presented with training images one at a time, along with the true class label. In addition, we assume that the agent can retrieve random *example images* of known classes. This assumption is realistic in many scenarios. For example, the agent may direct its gaze at known locations, or may interact with a human teacher.

Initially, the agent does not know about any objects or features. When it is presented with the first object, it simply remembers the correct answer given by the teacher. When it is shown the second object, it will offer the same answer as its best guess.

When the agent gives a wrong answer, it needs to learn a new feature to discriminate this object category from the mistaken category (or categories). This is done by random sampling, with a bias for structurally simple features. We employ the following heuristic procedure, where each step is iterated up to a constant number of times:

1. Pick a random feature from some other Bayes net (corresponding to another class) that is not yet part of this Bayes net (corresponding to the true class). This promotes the usage of general features that are characteristic of more than one class.
2. Sample a new feature directly from the misrecognized image by either picking two points and turning them into a geometric compound of two edgels, or by picking one point and measuring a texel feature vector.
3. Pick a random feature that is already part of this Bayes net and expand it geometrically by picking an additional image point close-by.
4. Pick two random features from this Bayes net and combine them topologically.
5. Similarly to step 4, but combine conjunctively.
6. Similarly to step 4, but combine disjunctively.

After each new feature is generated, it is evaluated on a small set of example images (retrieved from the environment as noted above) that include examples of the true class and the mistaken class(es). If the feature has any discriminative power, it is then added to the Bayes net of the true class using the conditional probabilities estimated on the example images. If the image is now recognized correctly by the expanded Bayes net, the feature learning procedure stops; if not, the feature is removed from the net, and the learning procedure continues. Note that it is possible for this procedure to terminate without success.

During operation of the learning system, an instance list of all classes encountered and features queried is maintained. Periodically, all feature cutpoints, class priors and conditional probabilities in the Bayes nets are updated according to this list.

Expert learning. Feature learning does not have to stop after learning a training set perfectly. The system can continue to search for better features. We train our system to develop better features by requiring a minimum KSD of all features that are used during a recognition procedure. If a feature does not meet this requirement, the system has to learn a new and better feature. We raise the minimum KSD at consecutive stages of expert learning, until the system fails to find adequate features. As a consequence, fewer (but superior) features will be queried while recognizing a given image, and many of the inferior features will become obsolete and can be deleted.

5. Experiments

To illustrate that our algorithm is able to produce discriminative features, we performed pilot experiments on two example tasks (Figure 3). In the COIL task, the images of the first four objects of the COIL-20 database[1] were split into two disjoint sets such that no two neighboring viewpoints were represented in the same set. As a result, each image set contained 36 images, spaced 10 degrees apart on the viewing sphere, at constant elevation. We performed a 2-fold cross-validation using these two sets. In the PLYM task, there were eight geometric objects on 15 artificially rendered images each, covering a small section of the viewing sphere[2]. We performed a 10-fold stratified cross-validation on this data set, with random subdivision of the 15 images of each class into 10 subsets of 1 or 2 images each.

The results of the experiments are summarized in Table 1. The first two columns indicate the number of expert learning stages, and the average number of features queried per recognition. The remaining columns give the proportions of correct, false, and ambiguous answers. While the recognition results fall short of current machine recognition

[1] http://www.cs.columbia.edu/CAVE/coil-20.html

[2] http://www.cis.plym.ac.uk/cis/levi/UoP_CIS_3D_Archive/8obj_set.tar

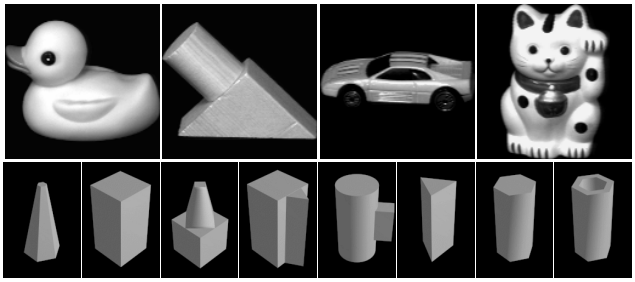

Figure 3. Objects of the COIL task (top) and the PLYM task (bottom).

Table 1. Summary of experimental results.

Task	#f.	Training Set:			Test Set:		
		cor.	wrg.	oth.	cor.	wrg.	oth.
COIL	44	0.98	0.02		0.81	0.19	
1	36	0.85	0.11	0.04	0.73	0.23	0.05
2	23	0.97	0.03		0.83	0.16	0.01
3	11	0.83	0.14	0.03	0.67	0.27	0.06
PLYM	19	1.00			0.72	0.28	
1	21	1.00			0.76	0.21	0.03
5	13	0.95	0.03	0.02	0.71	0.09	0.20

technology, they were achieved by an uncommitted visual system designed for open tasks, with a strong bias toward few and simple features that had access only to a small number of random training views at any given time during an incremental training procedure.

In accord with our biased search strategy, most learned features were isolated texels and simple geometric compounds of edgels and/or texels. Smaller numbers of the other compound types of features were also found. In most cases, the training set was not learned perfectly. This is because our system currently gives up after 10 iterations through the training set. Clearly, more effective techniques for finding distinctive features are desirable.

As the minimum KSD required of a feature is increased during feature upgrade, it is increasingly difficult for the system to find appropriate features in order to learn the training set perfectly. However, feature upgrade has the desired effect of decreasing the number of features queried during recognition, and where the training set is learned well, it also tends to reduce the number of false recognitions while marginally increasing the correct recognition rate on the test set.

6. Conclusions

Bayesian networks are not widely used in computer vision applications. Commonly encountered difficulties include the specification of the network structure and the conditional probability tables. For our application, we gave principled solutions to both problems based on known dependencies and an instance list. This list facilitates maximum reuse of acquired information. If the list is truncated by dropping the oldest instances, the learning system will smoothly adapt to a changing environment.

We have presented a framework for progressive learning of open visual recognition tasks. It is general enough to incorporate any type of localized image property as feature primitives, and a variety of means for composing them into higher-order features. Our method successfully learns features for object discrimination. Moreover, we showed how to learn improved features by increasing the minimum KSD required of features during recognition. If such features can be found, this results in a reduction of the number of features queried, a reduction of false answers, and a slight increase in correct answers.

One limitation of our system is the undirected search for features in images that is only guided by a few simple heuristics. Ideally, a system would learn systematic strategies for discovering useful features. Another critical limitation of our current system is the restricted expressiveness of our feature space that encodes only high-contrast edge, corner and texture information. A more complete model should at least encode color and blob-type features. In addition, sophisticated recognition requires higher-level features such as qualitative ("Gestalt") features (e.g. parallelism, symmetry, continuity, closure) and multiplicity (a triangle has three corners; a bicycle wheel has many spokes). These are areas of future research.

References

[1] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891–906, 1991.

[2] B. W. Mel. Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.

[3] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int. J. Computer Vision*, 14:5–24, 1995.

[4] R. C. Nelson and A. Selinger. A cubist approach to object recognition. In *Int. Conf. on Computer Vision*, 1998.

[5] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 1988.

[6] J. H. Piater and R. A. Grupen. Toward learning visual discrimination strategies. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 410–415. IEEE Computer Society, June 1999.

[7] R. P. N. Rao and D. H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78:461–505, 1995.

[8] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Fourth Europ. Conf. on Computer Vision*, Cambridge, UK, Apr. 1996.