A Probabilistic Approach to Real-time Construction of Memories

Brendan Burns, Paul Cohen
University of Massachusetts
Amherst, Massachusetts, 01002.
{bburns,cohen}@cs.umass.edu

December 14, 2001

Abstract

We outline a probabilistic technique for unsupervised construction of object models. We demonstrate that using this technique an agent can construct models which distinguish the different objects observed by the agent. In contrast to previous model based object recognition techniques which supplied the models used for recognition, this probabilistic technique performs unsupervised clustering. The agent develops a set of models in an unsupervised manner based solely upon its previous experience. We perform recognition on time series, rather than individual images, allowing an agent to recognize an object by learning its dynamics. Results of experiments exploring this technique in simulation and the real world are presented.

1 Introduction

Our motivation in exploring object memory and recognition stems from a desire to give an autonomous agent the ability to build symbolic representations of its environment through its interactions. To do this, an agent must be able to recognize physical objects in this environment. Persistent memories allow an agent to realize that two temporally distinct experiences are in fact experiences with a similar or identical object. This knowledge allows an agent to group and generalize experiences with similar objects. Rather than attempting to answer the instantaneous question "what object is this?" we are interested in recognizing the presence of similar objects in temporally distinct experiences. We do not care so much about the traditional "object recognition" task. That is we are not concerned if the agent can say "That thing I just passed was a chair" instead we focus on the agent's ability to say "Both of those experiences were with object A." Finally it is a requirement of our work that an agent build its memory with little to no outside supervision. This unsupervised aspect is critical both because it allows the agent to develop representations that it sees as significant and because it allows for models of objects which go beyond what any programmer might hand-code into a system.

The goals above mean that our agent's memory must have certain properties. It needs to incrementally build its memory of objects through its experiences. This building process must include adding new models as well merging of existing models as more information becomes available through experience. The building process must also occur in an online fashion, since the agent exploring its environment needs to immediately take advantage of the new information it has learned.

Ultimately these memories are models of the physical dynamics of a given object. The model's probabilities represent the probability of a shift in the object's appearance, as well as the probability that an object may take on a particular appearance at all. In this way the agent's models move beyond recognition and into dynamical representation. Thus, our models facilitate hypothesis construction by the agent about an object's future appearance.

2 Using Object Memories

2.1 Overview

The following are two tasks in the development of an object memory system: The process by which models are used to recognize objects and the process by which these models are constructed. The recognition of objects serves model creation by establishing whether the current experience is a part of a current model or needs to be added as a new model. This section describes our approach to the object recognition task which models the dynamics of an object's appearance through time.

The general description of our recognition task is: Given a time series of snapshots or data-points for an object $[S]$ and a set of object models $\{M_1 \dots M_n\}$, determine M_i such that $argmax(P(M_i|S))$, e.g. the model selected is the model with the highest probability of representing the time series.

2.2 Generalizing Instantaneous Experiences

To represent an experience with an object as a discrete time series it is necessary to distill our agent's experiences with real world entities into snapshots. This discretization serves as a generalization of a particular view of an object into a class of views. An object and an agent's view yield a two dimensional image i. Creating a snapshot s_n from this image i is a matter of applying a generalizing function $g(i) = s_n$ which maps a specific image of an object to a general snapshot class. We call the range of this generalizing function $g()$ the *snapshot vocabulary* since it describes the set of snapshots which an agent may possibly view.

The process of constructing a snapshot time series involves moving through the world and at each time step t, recording the output of the function $g(i_t)$, where i_t is our agent's view at time t. Thus a time series is written as the set $S = \{s_1, \dots, s_n\}$ where each s_n is in the range of $g()$

2.3 Object Models

An *object model* encapsulates the probabilities of snapshots in a time series. An object model M_i is two functions: First $f_{M_i} : (S, S) \to \Re$ such that $f_{M_i}(s_k, s_n) = P(s_k \to s_n|M_i)$, that is the probability that s_n follows s_k in a time series S, given that the object described in the model M_i generated S. This is the *transition probability* of two snapshots s_k and s_n. Second $f'_{M_i} : S \to \Re$ such that $f_{M_i}(s_k) = P(s_k|M_i)$, the probability that a snapshot s_k is observed in a time series given that the time series was generated by the model M_i. f'_{M_i} is called the *appearance probability* of a snapshot s_k.

2.4 Recognizing a Sequence

The first task for an agent with a set of object models $M_1 \dots M_n$ and a new time series S is to find M_i such that M_i had the highest probability given a time series of snapshots S, explicitly written as $P(M_i|S)$. From Bayes rule we can state that $P(M_i|S) = \frac{P(M_i)P(S|M_i)}{P(S)}$. Assuming uniform priors over all time series, this expression can be simplified to $P(M_i|S) = P(M_i)P(S|M_i)$. We can easily calculate the probability of the time series S given a model M_i as follows:

$$\begin{aligned}
P(S|M_i) &= P(s_1 \to s_2 \to \dots \to s_n|M_i) \\
&= P(s_1|M_i)P(s_1 \to s_2|M_i) \dots \\
&\quad P(s_{n-1} \to s_n|M_i) \\
&= f'_{M_i}(s_1)f_{M_i}(s_1, s_2) \dots \\
&\quad f_{M_i}(s_{n-1}, s_n)
\end{aligned}$$

To finish the calculation we only need to estimate the probability of the model. For the recognition task the probabilities of a model $M_i, P(M_i)$ is $\frac{1}{n}$ when there are n models.

Thus given a model M_i and a time series S, we can calculate $P(M_i|S)$ for every model M_i and simply select the highest probability model. The object whose model has the highest probability is deemed the object that generated the time series, and the time series is recognized as that particular object.

3 Building Memories

3.1 Overview

Our agent's memory is simply a set of models. How is such a set assembled? More importantly how can such a memory be dynamic? Initially model based techniques utilized hand coded models created for the agent. However, providing such hand coded models is ultimately detrimental to an exploratory agent. With hand-coding the agent's ability to distinguish objects is limited to the models encoded a priori. Further, a set of hand-coded models is necessarily static since it has no ability to incorporate new models without outside intervention.

An agent in use in a truly exploratory setting must be able to learn after it has been placed in its environment. Placing the burden of model construction on a human implementor is an unnecessary impediment. Instead the agent is provided with the ability to dynamically modify its models through experience. Even if we know the objects which we can expect to encounter we have no way of knowing a priori, if the distinctions in object models we create are useful to our robotic agent.

In the following section we discuss a probabilistic technique for constructing object models. First we describe the manner in which a times series may be transformed into a model, then we discuss the ways in which models are altered through time and experience.

3.2 Constructing Memories

Given a time series of snapshots S_k, memory construction produces a model M_j that represents the probabilities observed in S_k.

3.2.1 Transition Probabilities

Transition probabilities can be estimated from the transitions observed in the time series. Let $|S_k|$ denoted the number of snapshots in the time series S_k, then for each transition $s_m \rightarrow s_n$, the transition probability $P(s_m \rightarrow s_n|S_k)$ is estimated by:

$$P(s_n \rightarrow s_m|S_k) = \frac{\#\text{ times } s_n \rightarrow s_m\text{occurs in } S_k}{\#\text{ of transitions in } S_k}$$

$$P(s_n \rightarrow s_m|S_k) = \frac{Count(s_n, s_m, S_k)}{|S_k| - 1}$$

$Count(s_m, s_n, S_k)$ returns the number of times the transition $s_m \rightarrow s_n$ is observed in the time series S_k. The transition probability is simply the number of times a given transition, in this case $s_m \rightarrow s_n$, is observed in the set of series, divided by the total number of transitions observed in the series.

3.2.2 Appearance Probabilities

The appearance probability of a snapshot s_n in a single time series S_k is estimated by:

$$P(s_n|S_k) = \frac{Count(s_n, S_i)}{|S_k|}$$

$Count(s_n, S_i)$ returns the number of times the snapshot s_n appears in the time series S_i.

3.2.3 Unknown Snapshots

In the absence of any observations (e.g. if $Count(s_m, s_n, S_i) = 0$ or $Count(s_n, S_i) = 0$) the probabilities are given as a small non-zero constant. If this were not the case an unobserved transition would immediately remove what might otherwise be an easily recognized sequence by rendering its probability to zero.

3.2.4 Summary

Thus from any observed time series we can estimate f_{M_i} and f'_{M_i} [2.3]. Given an time series we can construct a new model of the object that generated the series. In the next section these estimations are generalized to allow time series to be incorporated into existing models.

3.3 Augmenting Memories

If our agent recognizes a time series of snapshots S_k as belonging to a model M_j then it makes sense that it should use this new information to update and improve its models. The process of augmenting object models is a generalization of the model construction technique outlined above. Generalizing the transition probability estimation to a set of time series $S_1 \dots S_n$ which match a model M_k, the probability of a transition $s_n \rightarrow s_m$ is estimated by:

$$P(s_n \rightarrow s_m | M_k) = \frac{\Sigma_{i=1}^{n} Count(s_n, s_m, S_i)}{\Sigma_{i=1}^{n}(|S_i| - 1)}$$

This *generalized transition probability* is the sum of the number of times a particular transition occurs in each series divided by the total number of transitions in the set of series.

The generalized appearance probability for a model M_k made up of series $S_1 \dots S_n$ is estimated by:

$$P(s_n | M_k) = \frac{\Sigma_{i=1}^{n} Count(s_n, S_i)}{\Sigma_{i=1}^{n}|S_i|}$$

The *generalized appearance probability* is the number of times a snapshot, s_n, is observed in a set of series, divided by the total number of snapshots observed in the set of series.

3.4 Adding Memories

Given a set of models $M_1 \dots M_n$ and a time series S, memory addition is the process which decides whether to add the series as part of an existing model or to add a new model created by the series S. We solve this decision problem by selecting the alternative which results in the most probable set of models given all observed time series [4].

First, define the probability of a model M_k given that we know it has recognized a set of time series $S_1 \dots S_n$. We will denote this probability as $P(M_k | S_1 \dots S_n)$. From Bayes' rule we know that this is equivalent to $P(M_k)P(S_1 \dots S_n | M_k)/P(S_1 \dots S_n)$. Noting that the probability of each series S_i is one since we have already observed each sequence allows us to simplify the expression to:

$$\begin{aligned} P(M_k | S_1 \dots S_n) &= P(M_k)P(S_1 \dots S_n | M_k) \\ P(M_k | S_1 \dots S_n) &= P(M_k)P(S_1 | M_k) \dots P(S_n | M_k) \\ P(M_k | S_1 \dots S_n) &= P(M_k)\Pi_{i=1}^{n} P(S_i | M_k) \end{aligned}$$

We must now generalize this probability to a set of models $M_1 \dots M_k$. Define the set $S_{j1} \dots S_{jn}$ as the set of time series which are classified as belonging to the model M_j. Generalizing the equation above is simply a matter of taking the product of the probabilities for each particular model. Thus the probability of the set of models is estimated by:

$$P(M_1 \dots M_k | S_{11} \dots S_{kn}) = \Pi_{j=1}^{k} \Pi_{i=1}^{n} P(M_j | S_{ji})$$

With this formula in mind, then it is simply a case of determining whether the set of models which has the latest time series S added as a separate models or the set of models which has S added to an existing models is more probable. Whichever set of models is more probable is kept, and the system moves on to the next experience.

4 Results

Results for this algorithm are presented below. The algorithm was tasked with two separate experiments. The first experiment was designed to test the quality of object recognition using stored memories. The second experiment was designed to test the agent's ability to develop memories autonomously.

4.1 Experimental Setup

Each experiment was run in two test-beds: a three-dimensional simulated environment and a real-world robotic platform. The three-dimensional simulated environment involved a camera circling a point in three-space while constantly keeping its camera focused upon that point. In an effort to make the simulated environment more like the real world, the experiments were run with the center of the camera's attention was located randomly within a half-meter of the object being observed.

In our first experiment the objects being observed were a cube, a pyramid and a wedge (fig. 1). In an attempt to see how our techniques scaled up to larger numbers of objects we ran a second set of experiments with sixteen objects made up out of a collection of cubes (fig 2). The task of object recognition in this domain was complicated by the fact that all the objects to be recognized are identical to one or more other objects when viewed from some angles. For example the sillohette of the wedge appears identical to the pyramid when viewed from one angle, but when it is turned ninety degrees it takes on the same appearance as the cube. Shared views would be fatal to any object recognition techniques which do not represent the possibility that an object has multiple appearances from different perspectives such as traditional single image approaches. However since this technique models the dynamics of an object rather than its particular view at a particular time it is able to cope with such challenges.

The robotic environment consisted of a Pioneer II robot in a play-pen which had three dimensional objects placed inside of it. The robot had a low-level controller that caused it to approach and circle an object while maintaining the focus of its camera upon the object.

Between the two test-beds the only difference was the origination of the pixelized view of the world. In the first case it was a three dimensional simulation, in the second a digital camera. All of the visual analysis, image extraction, snapshot creation and object recognition was performed by the same program. We believe that the results in both platforms are highly comparable. The only significant difference is the decrease (although not elimination) of noise in the simulated environment.

4.2 Experiment 1

The first experiment consisted of providing the agent with an object memory constructed from single brief (60 seconds) experiences with each object in the set to be recognized and a single experience with an unknown object which the agent was required to classify as one of the objects in memory or unknown. The experiment's focus was evaluate the recognition ability of an agent who has already developed at least a portion of its memory. The agent's performance was evaluated based upon the accuracy of its recognition.

4.2.1 Results and Discussion

With the first set of objects (cube, wedge, pyramid) the overall accuracy of the recognition algorithm was 97%. With the larger set of sixteen objects the overall accuracy of the recognition algorithm was 80% In both cases these averages were over multiple instances of each object. Four instances of the first set of three objects and ten instances of the second set of sixteen objects.

The random focus in these experiences meant that at times large portions (up to 50%) of the objects were outside of the agent's view. This resulted in previously unseen snapshots which created transition probabilities not present in the object models. It is significant to see that the algorithm is robust to this level of noise in its observations. Additionally when the algorithm made mistakes it made them by misclassifying objects which share common snapshots. For example in the smaller set of objects, the wedge was mistaken for the cube and the pyramid was mistaken for the wedge, but the cube was never mistaken for the pyramid.

We believe the decline in accuracy between the first and second set of objects is due to the increase in simularity between objects in the set. For every object in the set of sixteen there was another object which varied from it by only placement of a single cube (less than 25% of its total volume) as a result there are many viewpoints from which the objects are absolutely identical. Also the second set of objects is significantly more complex than the first set which makes the recognition task more difficult since the process of generalizing viewpoints into snapshots can eliminate important details.

In the robotic domain the overall accuracy was 75%. This was on a limited number of trials (20) due to the intricacies of dealing with a real robotic platform interacting dynamically with its environment. The snapshot time series captured by the robot had similarities to the random centered time series since in many views the objects were partially outside of the camera's frame. The robot is also incapable of making a perfect circle of a point so the low-level controller's behavior resulted in views which differed in scale and orientation across experiences. Like the

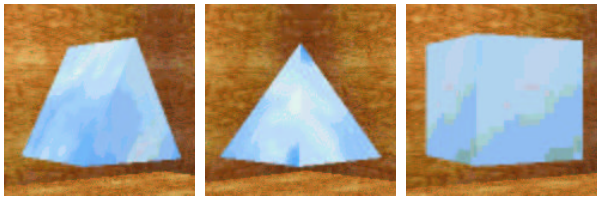

Figure 1: The three objects in the first simulation experiment

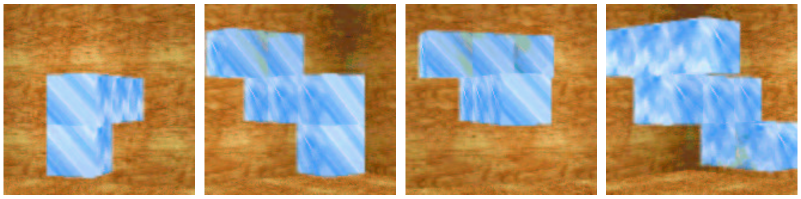

Figure 2: Four of the sixteen objects in the second simulation experiment

second set of objects, the decrease in accuracy with the real robot is no doubt also caused by the increased complexity of the objects being viewed.

4.3 Experiment 2

The second experiment was designed to test the agent's abilities to develop memories. The agent was exposed to a series of objects with no information about any of the objects. It was given the choice to recognize the series as a known object and incorporate this new experience into a known object model or add the experience as a new model. The agent started with no memories and was scored both upon the accuracy of the models which it learned and the number of models learned relative to the real number of objects seen.

4.3.1 Results and Discussion

In the simulated environment when exposed to a selection of memories with focus centered on the object, the system constructed only three memories where each memory corresponded to one of the objects it experienced.

In the simulated environment with twelve experiences of the first set of objects, the system constructed four memories resulting in the duplication of a model of the pyramid in memory. This addition did not greatly hinder the accuracy of the memory, if recognizing an instance of the pyramid as either model is considered "correct" and recognizing an instance of another object as either pyramid model is is considered "incorrect". This results point to the need for a model merging component to be added to the system. Since the classification of pyramids was nearly evenly split between the two models, a model merging operation would likely solve this problem and not reduce accuracy.

With the second set of objects the system constructed 40 memories when exposed to 160 experiences, ten with each object. This means that there were an excess of 24 memories created. These memories were nearly exclusively caused by varying numbers of duplicate memories of five of the sixteen objects. The five objects which were duplicated were the same ones which system had the most difficulty recognizing when performing the pure recognition task described above.

In the real world data set, the memory construction created seven models for the three objects (two men, two giraffes and three dogs). Although this is not a perfect result, it does show that it is capturing a consistency between experiences, even if it is unable to find the single model which captures all experiences. Again a model merging component would be useful since the identifications of the other experiences were evenly spread across the various models.

These results are preliminary but they show the ability of a probabilistic system to autonomously build and use models of object dynamics for the purpose of recognition in simulation and the real world.

5 Related Work

Statistical and probability based recognition has existed in the pattern recognition community for a long time ([1], [8]). Often these techniques have been applied to object recognition tasks ([?], [?]). Although our approach to probability based recognition shares some similar traits it is different in some fundamental ways. Unlike much of the work in pattern recognition the method described above is unsupervised. It autonomously builds clusters to perform recognition without any information about the actual quantity or distribution of classes. More importantly, instead of modeling objects as a distribution of values in a finite vector representation, we are modeling objects as first order Markov chains. Thus we are not representing an object as the probability of a particular sillohette, but rather the probability of a particular sillohette given the current sillohette. This distinction is important because it is what makes our model represent the dynamics of an object and not just its appearance. Finally, pattern recognition largely performs recognition on fixed length vectors of values. For our technique, there is no required length of experience needed to perform recognition.

There has been a number of earlier works which have dealt three-dimensional object recognition. [2] gives a good overview of various model based approaches to object recognition. One of the first attempts to infuse probability into the process of object recognition is [3] where probabilistic rankings were used to reduce the size of the model search space.

Much of this earlier vision work focused upon the construction of monolithic object representations. Lately people have discovered the usefulness of using snapshot based representations [5]. [9] developed models which were a set of snapshots and performed recognition through interpolation between views. Other work has also focused upon breaking up an object into its constituent parts for recognition ([7]) but used two dimensional images and extracted features which were consistent across multiple views.

Recently work has also begun to explore unsupervised model construction. [6] showed work on the autonomous development of three dimensional memories of objects. This work also dealt with single two dimensional images and built line segment models of objects to use to classify future experiences. Object recognition was performed through extraction of these line segment features and look up in an associative database of feature object pairs.

The techniques described above all were developed with image processing as the ultimate goal of the system. This differs from our desire to facilitate the representation of an environment by an exploratory agent.

The BCD algorithm [4] should also be mentioned since it showed a general method of using probability theory to autonomously cluster time series of discrete data. These clusters are analogous to our memories and in fact the BCD algorithm influenced the development of our model manipulation. Unlike BCD which reconsiders the total number of models at every step, this system simplifies (and speeds) itself by only considering the choice between adding an experience as part of an existing model or as a new model. As was mentioned earlier, some form of incremental merging, possibly not after every experience, would no doubt improve performance.

6 Conclusions and Future Work

We have described a system for the autonomous and online development of an agent's memory of objects which models the dynamics of the object's appearance. We have demonstrated the agent's ability to dynamically construct its own models and utilize these models for classification of its experiences. We have also demonstrated the effectiveness of the models for the pure recognition task. This work is significant because it demonstrates a technique which can develop memories in an unsupervised fashion. Such an ability is necessary for the autonomous exploration of unknown areas as well as critical for the unsupervised development of symbolic information about an agent's world.

There are several ways in which this system could be improved. Currently our snapshots are one dimensional points representing information about the shape class of the object. A snapshot which incorporated knowledge of the agent's activities such as rotational velocity and camera angle could provide more accurate representations of each object's dynamics. Also the prior probabilities of all of the observed models are fixed at $\frac{1}{n}$ for a memory with n models. These probabilities could instead be adjusted to represent the observed frequency of objects. This would in effect place a bias on existing models and limit the rate at which new models are formed. In addition to merging, this might prevent the creation of extraneous models.

Also we are interested in exploring other methods of performing recognition. Instead of using a probability distribution as a model to calculate the probability of a sequence we could compare the distribution in the model to the distribution implicit in the unknown sequence. There are several well known methods for comparing probability distributions and it would be interesting to explore the accuracy that they would achieve.

In addition to these improvements to the object recognition system, our work going forward will be to utilize such a system in an agent performing "active exploration." That is, an agent which utilizes the object recognition probabilities it receives from its memory to make decisions about how and when to interact with the surrounding world in order to maximize its ability to classify its interactions. This active exploration is the beginning of the steps which will allow an agent to become truly unsupervised and pro-active in its exploration and modeling of an environment. Such independent model construction is the ultimate goal of this line of research.

References

[1] P. Devijer and J. Kittler, editors. *Pattern Recognition Theory and Applications.* NATO/ASI, 1986.

[2] S. Edelman. Computational theories of object recognition, 1997.

[3] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3), March 1987.

[4] P. Cohen M. Ramoni, P. Sebastiani. Bayesian clustering by dynamics. *Machine Learning*, 2001.

[5] H. Bulthoff M. Tarr. Image based object recognition in man, monkey and machine. In H. Bulthoff M. Tarr, editor, *Object Recognition in Man, Monkey and Machine.* MIT Press, Cambridge, MA, 1998.

[6] R. Nelson. Three-dimensional recognition via twostage associative memory, 1995.

[7] R. Nelson and A. Selinger. A cubist approach to object recognition. In *Internation Conference on Computer Vision*, New Delhi, India, 1998. Narosa Publishing House.

[8] Jurgen Schurmann. *Pattern Classification: A unified view of statistical and neural approaches.* Wiley Interscience, 1996.

[9] S. Ullman. Three-dimensional object recognition based on a combination of views. In H. Bulthoff M. Tarr, editor, *Object Recognition in Man, Monkey and Machine.* MIT Press, Cambridge, MA, 1998.