Unsupervised Clustering of Robot Activities: A Bayesian Approach

Paul Cohen, Marco Ramoni, Paola Sebastiani and John Warwick

University of Massachusetts in Amherst, MA, USA, The Open University, UK and Imperial College, UK

Abstract

Our goal is for robots to learn conceptual systems sufficient for natural language and planning. The learning should be autonomous, without supervision. The first steps in building a conceptual system are to say some things are alike and others are different, based on how an agent interacts with them, and to organize similar things into classes or clusters. We use the BCD algorithm for clustering episodes experienced by our robots. The clusters contain episodes with similar dynamics, described by Markov chains.

1 Introduction

Picture yourself in a room illuminated only by a computer screen, onto which bit strings flash, two or three a second, for a few seconds at a stretch. After a few moments of this, the screen goes blank, until it starts up again with another sequence of bit strings. Your task is to make sense of the bit strings. Being human, you wonder what they mean. You try to work out how they might refer to your own experiences. You ask where they came from and what kind of process generated them. But your questions lead nowhere, so lacking a stronger model, you decide to analyze the bit strings as tokens generated by a very simple machine, a Markov chain. You treat each unique string as a token and tabulate the transition probabilities between them. And then you cluster the sequences of tokens into groups with similar transition probabilities. At the end of the day, you have your answer: Although the machine has generated several hundreds of sequences of bit strings, none identical, only five clusters emerge. Whatever process is generating the data, at some level of abstraction it seems to be doing only five qualitatively different things.

We have just described the world as experienced by a Pioneer 1 robot. As it does things in the laboratory — pushing a toy cup, passing another, picking up a block — its perceptual system produces propositions such as `(OBJECT A RED)` and `(APPROACH ROBOT A)`. By design, we know what these propositions mean but the robot does not. They might as well be structures of gensyms. Two or three times a second, the perceptual system produces a set of propositions to describe the current state. Every few seconds, the robot stops doing whatever it was doing and starts something else. In the current work, we mark these *episode boundaries* for the robot, although we are developing an algorithm to find episode boundaries automatically. Episodes, then, are time series representations, grounded in sensory data, of robot activities. The problem for our robot, and the focus of much of our research, is to learn enough about its activities and the objects in its environment to support planning and natural language dialog with humans [2, 3, 12, 11, 14, 16]. This paper describes an essential early step in the robot's conceptual development: Clustering episodes by their dynamics. Once the robot has identified clusters — once it knows that these episodes are similar and those are not — it can search for explanations, in particular, it can look for attributes of episodes that predict cluster membership.

We have developed a Bayesian algorithm for clustering episodes described by logical proposition. In this work, dynamics are captured in first-order Markov chains (see [14, 12] for other approaches) and the clustering algorithms puts together episodes that are likely to be generated by the same process. Although a Markov chain is a very simple descriptions of a dynamic process, the algorithm, called Bayesian Clustering by Dynamics (BCD), has been applied successfully to cluster robot experiences based on sensory inputs [17, 19], simulated war games [18], as well as the behavior of stocks in market and the fugues of Bach.

2 Bayesian Clustering by Dynamics

The BCD algorithm is easily sketched: Given time series of tokens that represent states, construct a transition probability table for each series, then measure the similarity between each pair of tables to decide which tables try to cluster first, and finally group similar tables into clusters if their grouping increases a scoring metric. The clusters found by the BCD algorithm have the interesting property that they comprise a clustering with maximum posterior probability.

2.1 Estimating Markov Chains

Suppose we observe a time series $S = (x_0, x_1, x_2, ..., x_{i-1}, x_i, ..)$, where each x_i is one of the states $1, ..., s$ of a variable X. In the current work, x_i is a set of propositions generated by the robot's perceptual system at time i, such as `((MOVING-FORWARD R) (IS-RED A))`. The process generating the sequence S is a (first order) Markov chain if the conditional probability that the variable X visits its state j at time t, given the sequence $(x_0, x_1, x_2, ..., x_{t-1})$, is only a function of the state visited at time $t-1$. Hence, we write $p(X_t = j|(x_0, x_1, x_2, ..., x_{t-1})) = p(X_t = j|x_{t-1})$, where X_t denotes the variable X at time t.

Markov chains can be represented as a probability distribution over the possible initial states of the chain and a table $P = (p_{ij})$ of transition probabilities, where $p_{ij} = p(X_t = j|X_{t-1} = i)$ is the probability of visiting state j given the current state i. Given a time series generated from a Markov chain, we might estimate the probabilities of state transitions $X_t = j|X_{t-1} = i$ from the data as $p_{ij} = n_{ij}/n_i$, where $n_i = \sum_j n_{ij}$ and n_{ij} is the frequency of the transitions $X_t = j|X_{t-1} = i$ observed in the time series. Instead we prefer a Bayesian estimate in which prior information about transition probabilities can be taken into account. The probability $\hat{p}_{ij}$ is estimated as $\hat{p}_{ij} = (\alpha_{ij} + n_{ij})/(\alpha_i + n_i)$, where $\alpha_i = \sum_j \alpha_{ij}$ and the so called prior hyper-parameter α_{ij} can be thought of as the prior frequency of the transition $X_t = j|X_{t-1} = i$, thus encoding prior knowledge about the process [17]. This estimate is a posterior probability in the sense of being estimated from prior information α_{ij} about the transition $X_t = j | X_{t-1} = i$ and the observed frequency n_{ij} of the transition. Thus, α_i and n_i are the numbers of times the variable X visits state i in a process consisting of α and n transitions, respectively.

2.2 Clustering

The story so far has the robot engaging with its environment in episodes of a few seconds duration. Each episode is a time series S_i of sets of propositions, and each time series is transformed into a Markov chain as described above. Now, given the set of Markov chains, the BCD algorithm is ready to cluster the series in the set $S = (S_k)$. The BCD algorithm is *agglomerative*, which means that, initially, there is one cluster for each Markov chain, then pairs of Markov chains are merged, iteratively. Merging two Markov chains yields another Markov chain and so on until a stopping criterion is met. Both the decision about whether to group Markov chains and the stopping criterion are based on the posterior probability of the clustering, that is, the probability of the clustering conditional on the data. Two Markov chains are merged if doing so increases the posterior probability of the clustering, and the algorithm stops when the posterior probability of the clustering cannot be improved. In fact, BCD performs a hill-climbing search through the space of clusterings, so it yields a locally-maximum posterior probability clustering. More precisely, it produces a partition — a division of the episodes into mutually exclusive and exhaustive subsets. BCD's task is to find a maximum posterior probability partition of Markov chains. Said in yet another way, BCD solves a Bayesian model selection problem, where the model it seeks is the most probable partition of Markov chains given the data.

The number of possible partitions grows exponentially with the number of Markov chains, so BCD cannot evaluate them all in its search for the most probable partition given the data. A heuristic method is required to make the search feasible. A good heuristic is merge or agglomerate similar Markov chains. What makes two Markov chains similar? The measure of similarity that BCD uses is therefore an average of the Kulback-Liebler distances between row conditional distributions. Iteratively, BCD computes the set of pairwise distances between the transition probability tables, sorts the generated distances, merges the two closest Markov chains and evaluates the result. Note that the similarity measure is only a heuristic guide for the search process rather than a grouping criterion. The evaluation asks whether the resulting model M_c, in which two Markov chains are merged and replaced by the resulting Markov chain, is more probable than the model M_s in which these Markov chains are different, given the data S. If the probability $p(M_c|S)$ is larger than $p(M_s|S)$, BCD updates the set of Markov chains by replacing the two Markov chains with the cluster resulting from their merging. Then, BCD updates the set of ordered distances by removing all the ordered pairs involving the merged Markov chains, and by adding the distances between the new Markov chain and the remaining Markov chains in the set. The procedure repeats on the new set of Markov chains. If the probability $p(M_c|S)$ is not larger than $p(M_s|S)$, BCD tries to merge the second best, the third best, and so on, until the set of pairs is empty and, in this case, returns the most probable partition found so far. The rationale for this search is that merging similar Markov chains first should result in better models and increase the posterior probability sooner. Empirical evaluations of the methods in simulated data appear to support this intuition [17]. Further details are in [17] and [18].

3 Clustering Robot Experiences

The robot in these experiments is a Pioneer 1 platform. All told, the robot generates roughly 40 time series of real-valued sensor data, sampled at 10Hz. In previous work we showed how to cluster episodes described by raw sensor data [19, 12, 16, 14, 15]. In this work, we cluster time series of propositions returned by a rudimentary perceptual system. One reason is that we overload the clustering algorithm when we force it to both resolve the ambiguity in sensory data — the job of perception — and find similarities in time series. The robot's perceptual system is a work in progress, but for our most recent experiments, it could describe the state of the world with up to 47 propositions at any instant. The state space for the robot may therefore be represented as a bit string of length 47. Each unique bit string represents a unique combination of propositions; for example, `LEFT-OF A B, IS-RED A, IS-ORANGE B` is one such string, in which three bits — corresponding to these propositions — have the value 1 and the rest have value 0. Although the state space is 2^{47}, in practice the robot encounters only a few of these states, and in this experiment encountered only 40. Because the robot encountered only 40 unique states, BCD can represent each episode as a transition probability table of size 40×40. For example, in the first ten steps of an episode, one sees six unique states, described in Table 1. [1]

Episodes in the experiment were "set pieces" in which the robot executed a simple program in an environment controlled by us, such as moving forward past one object and bumping into another. Each episode lasted between two and eight seconds. Three replications, with different starting locations for the robot and objects in its environment, were run for each of the scenarios described in Table 2. Of course, the robot cannot group and differentiate these episodes based on what *we* call them, it must do so based on its perceptions during the episodes. BCD produced a partition of six clusters for these episodes that is displayed in Table 3. The numbers in parentheses refer to the replications of episodes; for instance, the first cluster contains all three replicates of `PUSH-C` and `APPROACH-C`, whereas the second cluster contains one of the three replicates of `PASS-RIGHT-C`. Clusters 3 and 4 each contain the replicates of just a single activity, whereas Clusters 5 and 6 contain several activities.

How should we evaluate this partition? Let us note, first, that it was produced by a single run of BCD, with no effort to tune the α parameter, or clean up the perceptual data, or to "help" BCD in any way. BCD did not produce 14 clusters (corresponding to the 14 scenarios in Table 2), but instead grouped some activities together. For example, Cluster 4 contains four activities in which the robot moved toward object A. Sometimes it stopped short of the object, sometimes it passed the object (on the left or right), and sometimes it pushed the object. It is not surprising, nor particularly disappointing, that BCD grouped these activi-

[1] Notice that some states are physically impossible; for example, in the fifth state in Table 1, the robot is apparently receding from an object and stopped. The perceptual system is imperfect and has no "common sense" about the world, so it not infrequently constructs impossible state descriptions.

```
((STOP R) (IS-RED A))
((STOP R) (IS-RED A))
((APPROACH A R) (STOP R) (IS-RED A))
((STOP R) (IS-RED A))
((RECEDE A R) (STOP R) (IS-RED A))
((IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
```

Table 1: An example of propositions returned by the perceptual system of the robot.

APPROACH-A	APPROACH-C
PASS-RIGHT-A	PASS-RIGHT-C
PASS-RIGHT-A-THEN-PUSH-C	
PASS-RIGHT-C-THEN-PASS-RIGHT-A	
PASS-LEFT-A	PASS-LEFT-C
PASS-LEFT-A-THEN-PUSH-C	
PASS-LEFT-A-THEN-PASS-LEFT-C	
PASS-LEFT-A-THEN-PASS-RIGHT-C	
PASS-LEFT-C-THEN-PASS-RIGHT-A	
PUSH-A	PUSH-C

Table 2: Scenarios used in the experiment.

Cluster 1	PUSH-C (1 2 3)
	APPROACH-C (1 2 3)
Cluster 2	PASS-LEFT-C (1 2 3)
	PASS-RIGHT-C (1)
Cluster 3	PASS-RIGHT-A-THEN-PUSH-C (1 2 3)
Cluster 4	PASS-RIGHT-C-THEN-PASS-RIGHT-A (1 2 3)
Cluster 5	APPROACH-A (1 2 3)
	PASS-RIGHT-A (1 2 3)
	PASS-LEFT-A (1 2 3)
	PUSH-A (1 2 3)
	PASS-LEFT-C-THEN-PASS-RIGHT-A (2 3)
Cluster 6	PASS-LEFT-C-THEN-PASS-RIGHT-A (1)
	PASS-RIGHT-C (2 3)
	PASS-LEFT-A-THEN-PUSH-C (1 2 3)
	PASS-LEFT-A-THEN-PASS-LEFT-C (1 2 3)
	PASS-LEFT-A-THEN-PASS-RIGHT-C (1 2 3)

Table 3: Clusters produced by the BCD algorithm.

ties together, as they share similar dynamics: they all begin with approaching the object.

The disappointment is that BCD included PASS-LEFT-C-THEN-PASS-RIGHT-A (2 3) in Cluster 5, where they clearly do not belong. The story for Cluster 6 is similar: Three of the activities (and nine of the episodes) involve passing A on the left and then interacting with C, but one episode, PASS-LEFT-C-THEN-PASS-RIGHT-A (1) doesn't belong. The remaining two episodes, PASS-RIGHT-C (1 2), have similar dynamics to the latter phase of PASS-LEFT-A-THEN-PASS-RIGHT-C (1 2 3), so grouping them in Cluster 6 is not incorrect. As to Cluster 1, pushing C involves first approaching it, so grouping these activities together makes sense. Lastly, Cluster 2 is "pure" but for the inclusion of PASS-RIGHT-C (1). In sum, we would have been happier had the activities in Clusters 5 and 6 not been grouped together, and we have identified four episodes (out of 42) that clearly do not belong in the clusters to which they were assigned, but on the whole, the partition above is satisfactory.

In a followup analysis, we ablated the state descriptions and re-ran BCD, to see how much the partition depended on particular propositions in the state descriptions. For example, we removed the propositions IS-RED X and IS-ORANGE X and re-wrote the affected state descriptions (so the state ((APPROACH R A)(IS-RED A)) becomes (APPROACH R A)). Similarly, we removed the propositions MOVING-BACKWARD R and MOVING-FORWARD R. In these analyses, BCD did not produce partitions of the episodes identical with the one above, but many of the clusters' substructures were maintained. For example, after removing propositions about color, BCD grouped together the elements in Clusters 1 and 3, above. It also formed a new cluster from PASS-RIGHT-C (1 2 3) and PASS-LEFT-A (1 2 3) episodes, which seems odd, although part of the explanation is that the robot identifies objects A and C by color, so with color terms gone, it confuses activities with objects A and C.

We have used BCD in other experiments with the robot, with similar results — it groups together instances of activities, and groups activities that share components such as approaching an object — and we have also used it to cluster simulated engagements in a wargame simulator [18], as well as time series of financial instruments, and Bach's fugues. In all these cases, we have been pleased with the results, but we recognize the need for more objective evaluation criteria. Generally it is difficult to say whether a partition of episodes is correct: Even if a gold standard partition exists for a given problem (and it doesn't in these experiments), we would need a metric that accounts for partially matching the gold standard. A different approach would be to assess whether a partition leads to good or poor results when it is used for some purpose, such as classification or another predictive task. But as yet, we have not put the robot's clusters to use in the life of the robot.

4 Related Work

At first glance it may appear that BCD and Hidden Markov Models (HMMs) are similar technologies, and indeed we have used HMMs for some robot learning tasks [4], but they are quite different. A HMM is a state machine with transition probabilities between states and each state has a probability distribution over the tokens it emits [13]. HMMs are trained with time series (univariate or multivariate, continuous or categorical), which means the probability distributions within states and the transition probabilities between states are estimated. One must specify the number of states in advance, although the algorithm in [4] dynamically splits HMM states in accordance with a minimum description length principle. Might one use HMM technology to find maximum likelihood partitions of time series, as BCD does? This problem would have to be transformed into one of finding the probabilities of emitting tokens and transition probabilities in a model of a specific number of states. An obvious choice would be to fit one HMM with n states to each episode with n unique states, then cluster the HMMs, but it is unclear what advantage this holds over our current method of estimating Markov chains and clustering them. Another idea is to have each state represent a cluster of Markov chains. The difficulty is that, *within* an HMM state, one can only model the *marginal* probabilities of tokens, not the conditional probability of a token given the previous token. These conditional probabilities are modeled as state transition probabilities in HMMs, which means that an episode must be modeled as a sequence of HMM states, not as a single state.

The Bayesian modeling-based approach used in BCD is similar to that used, for example, by Raftery [5] to cluster static data. Recent work [20] attempted to extend the idea to dynamic processes without, however, succeeding in finding a closed form solution as the one we have identified.An important attribute of our method is its heuristic search, which makes the algorithm feasible. BCD is similar in some respects to other algorithms for clustering time series developed in our lab. To assess the dissimilarity of a pair of multivariate, real-valued time series, Oates applies dynamic time warping to force one series to fit the other as well as possible; the residual lack of fit is a measure of dissimilarity, and with this, Oates can cluster episodes [12]. Rosenstein solves the problem by first detecting events in time series, then measuring the root mean squared difference between values in two series in a window around an event [14, 15]. It is worth noting that these methods and BCD handle time very differently. In Rosenstein's method, two time series are compared moment by moment for a fixed interval. In Oates's approach, one series is stretched and compressed within intervals to make it fit the other as well as possible. The former method keeps time rigid, the latter makes time elastic. We are beginning a study of the relative strengths and weaknesses of these methods on several kinds of time series.

5 Conclusion

Our goal is for robots to learn conceptual systems sufficient for natural language and planning, without supervision. By conceptual systems, we mean organizations of concepts that denote activities and objects in the robots' world; and knowledge about how activities unfold, and the roles that objects play in activities. With Lakoff and others [9, 8, 10, 7, 21, 6, 22, 1] we assert that the first step in building a conceptual system is to say some things are alike and others are different, based on how we interact with them, and to organize similar things into classes or clusters. BCD organizes the robot's activities into clusters.The next step, currently underway, is to organize the objects in the robot's environment into clusters based on their roles in activities. Once we have clusters of activities and objects, we will apply standard classification algorithms such as `c4.5` to find the attributes of activities and objects that predict cluster membership.

Acknowledgments

This research is supported by DARPA/AFOSR under contracts Nos F49620-97-1-0485 and N66001-96-C-8504. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA/AFOSR or the U.S. Government.

References

[1] J. A. Coelho and R. A. Grupen. A control basis for learning multifingered grasps. *Journal of Robotic Systems*, 14(7):545–557, 1997.

[2] P. R. Cohen. Dynamic maps as representations of verbs. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence*, pages 45–149, 1998.

[3] P. R. Cohen, M. S. Atkin, T. Oates, and C. R. Beal. Neo: Learning conceptual knowledge by sensorimotor-interaction with an environment. In *Proceedings of the First International Conference on Autonomous Agents*, pages 170–177, 1997.

[4] L. Firoiu, T. Oates, and P. R. Cohen. Learning regular languages from positive evidence. In *Proceedings of the Twentieth Annual Meeting of the Cognitive Science Society*, pages 350–355, 1998.

[5] C. Fraley and A. E. Raftery. How many clusters? Which clustering methods? Answers via model-based cluster analysis. Technical Report 329, Department of Statistics, University of Washington, 1998.

[6] D. Gentner and A. B. Markman. Structural alignment in comparison: No difference without similarity. *Psychological Science*, 5:152–158, 1994.

[7] G. J. Gibbs and H. L. Colston. The cognitive psychological reality of image schemas and their transforms. *Cognitive Linguistics*, 6–4:347–378, 1995.

[8] M. Johnson. *The Body in the Mind*. University of Chicago Press, 1987.

[9] G. Lakoff. *Women, Fire, and Dangerous Things*. University of Chicago Press, 1984.

[10] J. M. Mandler. How to build a baby: II. Conceptual primitives. *Psychological Review*, 99(4):587–604, 1992.

[11] T. Oates and P. R. Cohen. Learning planning operators with conditional and probabilistic effects. In *Proceedings of the AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems*, pages 86–94, 1996.

[12] T. Oates, M. D. Schmill, and P. R. Cohen. Identifying qualitatively different experiences: Experiments with a mobile robot. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[13] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[14] M. Rosenstein and P. R. Cohen. Concepts from time series. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 739–745. AAAI Press, 1998.

[15] M. T. Rosenstein and P. R. Cohen. Continuous categories for a mobile robot. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999.

[16] M. D. Schmill, T. Oates, and P. R. Cohen. Learned models for continuous planning. In *Proceedings of Uncertainty 99: The Seventh International Workshop on Artificial Intelligence and Statistics*, pages 278–282, 1999.

[17] P. Sebastiani, M. Ramoni, and P. Cohen. Bayesian analysis of sensory inputs of a mobile robot. In *Proceedings of the 5th Workshop on Case Studies in Bayesian Statistics*. 1999. Submitted.

[18] P. Sebastiani, M. Ramoni, P. Cohen, J. Warwick, and J. Davis. Discovering dynamics using Bayesian clustering. In *Proceedings of the Third International Symposium on Intelligent Data Analysis*, pages 199–209. Springer, New York, NY, 1999.

[19] P. Sebastiani, M. Ramoni, and P.R. Cohen. Unsupervised classification of sensory input in a mobile robot. In *Proceedings of the IJCAI Workshop on Neural, Symbolic and Reinforcement Methods for Sequence Learning*. 1999. To appear.

[20] P. Smyth. Probabilistic model-based clustering of multivariate and sequential data. In *Proceedings of Artificial Intelligence and Statistics 1999*, pages 299–304. Morgan Kaufman, San Mateo CA, 1999.

[21] E. S. Spelke, K. Breinlinger, J. Macomber, and K. Jacobson. Origins of knowledge. *Psychological Review*, 99:605–632, 1992.

[22] E. Thelen and L. Smith. *A Dynamic Systems Approach to the Development of Cognition and Action*. MIT Press, Cambridge, MA, 1994.