Bayesian Clustering of Sensory Inputs by Dynamics

Paola Sebastiani, Marco Ramoni and Paul Cohen

Computer Science Technical Report 00-13

Experimental Knowledge Systems Laboratory
Computer Science Department
140 Governors Drive
University of Massachusetts
Amherst, MA 01003-4610

Abstract

This paper introduces a Bayesian method for unsupervised clustering of dynamic processes and applies it to the abstraction of sensory inputs of a mobile robot. The method starts by transforming the sensory inputs into Markov chains and then applies an agglomerative clustering procedure to discover the most probable set of clusters capturing the robot's experiences. To increase efficiency, the method uses an entropy-based heuristic search strategy.

TR: 00-13

TR Title: Bayesian Clustering of Sensory Inputs by Dynamics

Authors: Paola Sebastiani, Marco Ramoni, Paul R. Cohen

Computer Science Department
140 Governors Drive
University of Massachusetts
Amherst, MA 01003

Date: March, 2000

Keywords: Discovery, Machine Learning, Robotics

ABSTRACT

This paper introduces a Bayesian method for unsupervised clustering of dynamic processes and applies it to the abstraction of sensory inputs of a mobile robot. The method starts by transforming the sensory inputs into Markov chains and then applies an agglomerative clustering procedure to discover the most probable set of clusters capturing the robot's experiences. To increase efficiency, the method uses an entropy-based heuristic search strategy.

Bayesian Clustering of Sensory Inputs by Dynamics

Paola Sebastiani
Statistics Department
The Open University
p.sebastiani@open.ac.uk

Marco Ramoni
Knowledge Media Institute
The Open University
m.ramoni@open.ac.uk

Paul Cohen
Department of Computer Science
University of Massachusetts, Amherst
cohen@cs.umass.edu

Abstract

This paper introduces a Bayesian method for unsupervised clustering of dynamic processes and applies it to the abstraction of sensory inputs of a mobile robot. The method starts by transforming the sensory inputs into Markov chains and then applies an agglomerative clustering procedure to discover the most probable set of clusters capturing the robot's experiences. To increase efficiency, the method uses an entropy-based heuristic search strategy.

1. Introduction

This paper presents a Bayesian algorithm for *clustering by dynamics*. Suppose one has a batch of univariate time series generated by one or more unknown processes, and the processes have characteristic dynamics. Clustering by dynamics is the problem of grouping time series into clusters so that the elements of each cluster have similar dynamics. Suppose a batch contains a time series of stride length for every episode in which a person moves on foot from one place to another. Clustering by dynamics might find clusters corresponding to "ambling," "striding," "running," and "pushing a shopping cart," because the dynamics of stride length are different in these processes. Similarly, pathologies of the heart can be characterized by the patterns of sistolic and diastolic phases; dance steps, hand gestures and facial expressions can be characterized by the dynamics of movement of body parts [Johansson, 1973]; economic states such as recession can be characterized by the dynamics of economic indicators; syntactic categories can be categorized by the dynamics of word transitions [Charniak, 1993]; and so on.

The goal of this work is to enable mobile robots to learn the dynamics of their activities. Our algorithm learns Markov chain (MC) representations of the dynamics in time series and then clusters these time series by their dynamics to learn prototype experiences. For example, our robot has learned prototype experiences that correspond to passing an object and moving toward an object. It is important to the goals of our project that the robot's learning should be *unsupervised*, which means we do not tell our algorithm which Markov chains, clusters and prototypes to learn.

A MC represents a dynamic process as a transition probability matrix. For each experience the robot has, we construct one such matrix for each sensor. Each row in the matrix represents a state of the sensor, and the columns represent the probabilities of transition from that state to each other state of the sensor on the next time step. The result is a set of conditional probability distributions, one for each state of the sensor, that can be learned from the past experiences of the agent. After k experiences, the robot has learned k transition matrices for each sensor. Next, a Bayesian clustering algorithm groups experiences that produce similar transition probability matrices. Each group is then characterized by its average or prototypical dynamics. The learned model of dynamics enables the agent to classify its current experience by computing the probability of an experience being in a particular cluster given sensor readings, and to predict future experiences, conditional on current input and cluster membership.

While there are similarities between this problem and learning Hidden Markov Models (HMMs), this problem is different and somewhat simpler. An HMM has one probability distribution for the symbols emitted by each state, and also a matrix of probabilities of transitions between states [Rabiner, 1989]. In our problem we fit a fully observable Markov model to each episode and then we search for a partition of these models into clusters that maximizes the likelihood of the data. Thus our algorithm is more closely related to other approaches to clustering by dynamics, such as [Smyth, 1997; Rosenstein and Cohen, 1998], than it is to HMMs.

A Bayesian approach is particularly well suited to clustering by dynamics because it frames the learning process as continuous updating rather than a batch analysis of data. Furthermore, a Bayesian approach provides a principled way to integrate prior and current evidence. As our robot gains more experience (i.e., as its "prior" knowledge increases) it requires proportionately more evidence to modify or discount its prior conclusions.

The rest of the paper is organized as follows. After reviewing background material on MCs, we describe how to induce the transition probability matrix of a MC from sensor readings, and then describe a Bayesian clustering algorithm to sequentially merge similar MCs induced by episodes.

2. The Robot Platform

The Pioneer 1 robot is a small platform with two drive wheels and a trailing caster, and a two degree of freedom paddle gripper. For sensors the Pioneer 1 has shaft encoders, stall sensors, five forward pointing and two side pointing sonars, bump sensors, a pair of IR sensors at the front and back of its gripper, and a simple vision system that reports the location and size of color-coded objects. Our configuration of the Pioneer 1 has roughly forty sensors, though the values returned by some are derived from others.

3. Markov Chains

The dynamics of a sequence of sensory values can be modeled by a Markov Chain (MC). The sensor X is regarded as a random variable taking values $1, 2, ..., s$. The process generating the sequence $x = (x_0, x_1, x_2, ..., x_{i-1}, x_i, ..)$ is a MC if $p(X = x_t|(x_0, x_1, x_2, ..., x_{t-1})) = p(X = x_t|x_{t-1})$ for any x_t in x. Let X_t be the variable representing the sensor values at time t, then X_t is conditionally independent of $X_0, X_1, ..., X_{t-2}$ given X_{t-1}. The assumption of conditional independence allows us to represent a MC by a vector of probabilities $p_0 = (p_{01}, p_{02}, ..., p_{0s})$, denoting the distribution of X_0 (the initial state of the chain) and a matrix of transition probabilities:

$$P = (p_{ij}) = \begin{array}{|c|cccc|} \hline & & & X_t & \\ X_{t-1} & 1 & 2 & \cdots & s \\ \hline 1 & p_{11} & p_{12} & \cdots & p_{1s} \\ 2 & p_{21} & p_{22} & \cdots & p_{2s} \\ \vdots & & & \cdots & \\ s & p_{s1} & p_{s2} & \cdots & p_{ss} \\ \hline \end{array}$$

where $p_{ij} = p(X_t = j|X_{t-1} = i)$. By using the Chapman-Kolmogorov Equations [Ross, 1996], the expected value of X_t is $p_0 P^t$ which, for increasing values of t, gives the average sequence.

4. Discovering Markov Chains

During its interaction with the world, the robot records the values of about 40 sensors every 1/10 of a second. In an extended period of wandering around the laboratory, the robot will engage in several different activities — moving toward an object, losing sight of an object, bumping into something — and these activities will have different sensory signatures. Because we insist that the robot's learning is unsupervised, we do not tell the robot which activities it is engaging in, or even that it has switched from one activity to another. Instead we define a simple event marker — simultaneous change in three sensors — and we define an *episode* as the period between event markers. For each episode in each sensor, we build a transition matrix and then we cluster transition matrices with similar dynamics.

4.1 Learning A Markov Chain

Suppose the robot has generated a sequence of values from the sensor X for one episode. This sequences can be summarized into a $s \times s$ contingency table that contains the frequencies of transitions $n_{ij} = n(X_{t-1} = i \rightarrow X_t = j)$. These counts are used to estimate the transition probabilities p_{ij} characterizing the dynamic process that generated the data.

An intuitive way to estimate p_{ij} is to use the relative frequencies of transitions n_{ij}/n_i. In this way, the probability of the transition $X_{t-1} = i \rightarrow X_t = j$, that we will denote as $i \rightarrow j$, is estimated as the ratio between the number n_{ij} of times the transition has been observed and all observations on the variable in state i, that is, $n_i = \sum_j n_{ij}$. This estimate is a function of the data only and there may be other sources of information about the process. Furthermore, this method estimates the transition probability p_{ij} as 0 whenever $n_{ij} = 0$. Thus, when the chain is observed over a relatively short time interval, or a transition probability is small, it is very easy to conclude that some transition is impossible. A Bayesian estimation of p_{ij} overcomes this problem as well as using any prior knowledge about the process. This is achieved by augmenting the observed frequencies n_{ij} by *hyperparameters* α_{ij} that encode the prior knowledge about the process in terms of imaginary counts of a sample of size α. The *Bayesian estimate* of p_{ij} is

$$\hat{p}_{ij} = \frac{\alpha_{ij} + n_{ij}}{\alpha_i + n_i} \qquad (1)$$

where $\alpha_i = \sum_j \alpha_{ij}$. By writing Equation 1 as

$$\hat{p}_{ij} = \frac{\alpha_{ij}}{\alpha_i} \frac{\alpha_i}{\alpha_i + n_i} + \frac{n_{ij}}{n_i} \frac{n_i}{\alpha_i + n_i} \qquad (2)$$

we see that $\hat{p}_{ij}$ is an average of the estimate n_{ij}/n_i and of the quantity α_{ij}/α_i with weights that depend on α_i and the sample size n_i. Rewriting of Equation 1 as 2 shows that α_{ij}/α_i is the estimate of p_{ij} when the data set does not contain transitions from the state i — and hence $n_{ij} = 0$ for all j — and it is therefore called the *prior* estimate of p_{ij} while $\hat{p}_{ij}$ is called the *posterior estimate.* It can be shown that the variance of the prior estimate α_{ij}/α_i is given by $(\alpha_{ij}/\alpha_i)(1 - \alpha_{ij}/\alpha_i)/(\alpha_i + 1)$ and, for fixed α_{ij}/α_i, the variance is a decreasing function of α_i. Since small variance implies a large precision about the estimate, α_i will be called the *local precision* about the conditional distribution $X_t|X_{t-1} = i$ and it indicates the level of confidence about the prior specification. The quantity $\alpha = \sum_i \alpha_i$ is the *global* precision, as it accounts for the level of precision of all the s conditional distributions.