CLUSTERING FINITE DISCRETE MARKOV CHAINS

Greg Ridgeway, University of Washington and Steven Altschuler, Microsoft Corp.
Greg Ridgeway, Box 354322, University of Washington, Seattle, WA 98195-4322

Keywords: clustering, Markov chain, mixture modeling

Abstract

In problem situations where observations consist of a sequence of events, Markov models often prove useful. However, when there is suspected heterogeneity among the Markov transition kernels generating the observed sequences, more refined methods become necessary. In this paper we describe a probabilistic method for clustering Markov processes with a pre-specified number of clusters. We derive a Gibbs sampler and a computationally efficient hybrid MCMC-constrained EM algorithm.

Introduction

Consider an s-state discrete Markov process (Ross [1993]) where the transition matrix for the process is unknown. Further assume that a dataset of N such processes, possibly of different length, exists in which each process came from one of m transition matrices and an associated initial state distribution. Therefore, we should be able to cluster together those processes that share the same underlying Markov transition structure. In this problem setting we do not know the elements of the m transition matrices and their associated initial state distributions, the proportion of processes in each cluster, nor the cluster membership of each process. Since the cluster membership is an unobservable or latent variable, closed form maximum likelihood estimators are unobtainable.

Ridgeway [1997] describes an application to modeling user traversal of web sites. Although unobservable, site analysts might believe that certain classes of customers visit their site, developers, investors, etc. In order to learn how users traverse their site, to improve site design, and for collaborative filtering, the analyst needs to consider the heterogeneity of the population when clustering users.

Likelihood and posterior distribution

Let ${}_{(\ell)}P_{ij}$ be the (i,j) element of the ℓ^{th} probability transition matrix, or the probability that a process in cluster ℓ would transition from state i to state j. Also let ${}_{(\ell)}p_i$ be the i^{th} element of the initial state distribution of processes from cluster ℓ. For each of the N Markov processes, indexed by a superscript (k), we observe an initial state, i_0, and the number of times the process transitioned from state i to state j, n_{ij}. Lastly, $\delta_\ell^{(k)}$ is the unobserved 0/1 indicator that process k belongs to cluster ℓ. Therefore, the likelihood function is

$$f(\underline{n} \mid \underline{p}, \underline{P}, \underline{\delta}) = \prod_{k=1}^{N} \prod_{l=1}^{m} \left({}_{(l)}p_{i_0^{(k)}} \prod_{i=1}^{s} \prod_{j=1}^{s} {}_{(l)}P_{ij}^{n_{ij}^{(k)}} \right)^{\delta_l^{(k)}}$$

In the absence of prior information, specifying uninformative priors in straightforward. The rows of every cluster's probability transition matrix and every cluster's initial state distribution all receive uninformative, s dimensional Dirichlet priors. Let $\underline{\alpha}$ be vector of length m of the mixture proportions so that $\underline{\delta}^{(k)}$ will be distributed multinomial$(1,\underline{\alpha})$. Lastly, we assign an uninformative Dirichlet hyperprior to $\underline{\alpha}$. Therefore, the posterior distribution of the unknown model parameters follows by Bayes' Theorem.

$$f(\underline{p}, \underline{P}, \underline{\delta}, \underline{\alpha} \mid \underline{n}) \propto$$

$$\prod_{k=1}^{N} \prod_{l=1}^{m} \left({}_{(l)}p_{i_0^{(k)}} \prod_{i=1}^{s} \prod_{j=1}^{s} P_{ij}^{n_{ij}^{(k)}} \right)^{\delta_l^{(k)}} \bullet \prod_{k=1}^{N} \prod_{l=1}^{m} \alpha_l^{\delta_l^{(k)}}$$

Assuming that the first order Markov assumption is correct, this distribution captures all of the information about the process clustering that is contained in the data. However, this distribution is rather complex and all of the usual distribution summary values (mean, variance, etc.) are extremely difficult to extract. Appealing to a Markov Chain Monte Carlo approach (Hastings [1970], Gelman, *et al* [1995]) to sample from this distribution can avoid this problem with a degree of computational cost.

In this paper we use a Gibbs sampling algorithm that partitions the parameters into blocks for which sampling from the conditional distribution of any block given the remaining blocks is easy. Each row of every cluster's probability transition matrix, each cluster's initial state distribution, the mixture proportions, and each $\underline{\delta}^{(k)}$ form the blocks. The Gibbs sampling algorithm draws updates for each block in turn conditional on the current values of the other blocks, denoted by a superscript minus.

$$f({}_{(l)}p|{}_{(l)}p^{-}, \underline{n}) \propto \prod_{k=1}^{N} {}_{(l)}p_{i_0^{(k)}}^{\delta_l^{(k)}} = \prod_{k=1}^{N} \prod_{i=1}^{s} {}_{(l)}p_i^{\delta_l^{(k)} \bullet \mathrm{I}(i_0^{(k)}=i)} = \prod_{i=1}^{s} {}_{(l)}p_i^{\sum_{k=1}^{N} \delta_l^{(k)} \bullet \mathrm{I}(i_0^{(k)}=i)}$$

$$\equiv \text{Dirichlet}\left(1 + \sum_{k=1}^{N} \delta_l^{(k)} \, \mathrm{I}(i_0^{(k)} = 1), \; \ldots, \; 1 + \sum_{k=1}^{N} \delta_l^{(k)} \, \mathrm{I}(i_0^{(k)} = s) \right)$$

$$f(\underline{{}_{(l)}P_{i\bullet}} \mid \underline{{}_{(l)}P_{i\bullet}^{-}}, \underline{n}) \propto \prod_{k=1}^{N} \left(\prod_{j=1}^{s} {}_{(l)}P_{ij}^{n_{ij}^{(k)}} \right)^{\delta_l^{(k)}} = \prod_{j=1}^{s} {}_{(l)}P_{ij}^{\sum_{k=1}^{N} \delta_l^{(k)} n_{ij}^{(k)}}$$

$$\equiv \text{Dirichlet}\left(1 + \sum_{k=1}^{N} \delta_l^{(k)} n_{i1}^{(k)}, \ldots, 1 + \sum_{k=1}^{N} \delta_l^{(k)} n_{is}^{(k)} \right)$$

$$
\begin{aligned}
f(\underline{\alpha} \mid \underline{\alpha^{-}}, \underline{n}) &\propto \prod_{l=1}^{m} \alpha_l^{\sum_{k=1}^{N} \delta_l^{(k)}} \\
&\equiv \text{Dirichlet}\left(1+\sum_{k=1}^{N}\delta_1^{(k)},\ldots,1+\sum_{k=1}^{N}\delta_m^{(k)}\right) \\
f(\delta^{(k)} \mid \delta^{(k)-}, \underline{n}) &\propto \prod_{l=1}^{m}\left(\alpha_l \bullet {}_{(l)}p_{i_0^{(k)}} \prod_{i=1}^{s}\prod_{j=1}^{s} {}_{(l)}P_{ij}^{n_{ij}^{(k)}}\right)^{\delta_l^{(k)}} \\
&\equiv \text{Mult}\left(1, \tfrac{1}{z}\left[\alpha_1 \bullet {}_{(1)}p_{i_0^{(k)}} \prod_{i=1}^{s}\prod_{j=1}^{s} {}_{(1)}P_{ij}^{n_{ij}^{(k)}}, \ldots, \alpha_m \bullet {}_{(m)}p_{i_0^{(k)}} \prod_{i=1}^{s}\prod_{j=1}^{s} {}_{(m)}P_{ij}^{n_{ij}^{(k)}}\right]\right)
\end{aligned}
$$

Where Z is the appropriate normalizing constant.

These distributions have a rather intuitive interpretation as well. The row updates come from a distribution where the expected value is approximately the MLE for the row if the cluster assignments, δ, were known. The vector α come from a distribution where the expected value is approximately the mixture proportions if, again, the cluster assignments were known. Lastly, the cluster assignments are drawn such that the probability of each cluster is proportional to the mixture probability times the likelihood of the observation coming from the associated transition matrix.

As with all MCMC implementations of parameter estimation for mixture models, this method can suffer from the "label switching" problem. The posterior density for a particular labeling of the clusters is equal for any other permutation of the labels. If the clusters are "far apart" then it is *unlikely* that label switching would occur. However, with weak data or clusters that are very close label switching can be common place. In normal mixture models constraints are often imposed to insure identifiability. However, constraints such as these alter the posterior distribution and in a problem such as this one might not even be possible. A more appropriate method would detect switches in the labels and make corrections. Stephens [1996] proposes such a method.

Furthermore, MCMC algorithms tend to be slow to converge. Here we propose a hybrid MCMC-constrained EM algorithm that has shown substantial computational improvement.

A Constrained EM algorithm

The block conditional distributions shown in the previous section have a particularly nice feature. All of the probability parameters depend only on the cluster assignments and the observable data. The reassigning of processes to clusters then depends only upon the probabilities. This observation leads to the following algorithm.

1. Randomly assign the processes to clusters.
2. Rather than sampling from a Dirichlet to update the probability estimates, estimate the probabilities using the expected value of the block conditional.
3. Reassign each process to the cluster that most likely generated it. The vector of probabilities for a process belonging to each cluster is exactly the multinomial probability parameter for the $\underline{\delta}^{(k)}$ block conditional.
4. If none of the processes have been assigned to a different cluster then stop. Otherwise, go to step 2.

Hartigan's k-means algorithm (Hartigan, *et al* [1979], Forgy [1967]) is analogous to this hard cluster assignment formulation for normal mixture models.

The constrained EM approach lacks accuracy and detail but has the advantage of speed. The Gibbs sampler on the other hand can be used to compute arbitrary functionals of the distribution but takes several orders of magnitude longer to iterate to reasonable accuracy. Naturally a hybrid algorithm may be useful to borrow from the strengths and diminish the affect of the weaknesses of both algorithms. A hybrid algorithm iterates the constrained EM algorithm to convergence. The cluster assignments from the constrained EM algorithm provide initial assignments for the Gibbs sampler. Then, with little or no burn-in, the Gibbs algorithm runs until it obtains decent estimates for the posterior means and variance of the parameters.

Summary

Analysis of sequences of events in which homogeneity of transition probabilities is suspect might benefit from this method. This algorithm not only segments the sequences but also gives interpretable results from which the analyst can readily draw application specific conclusions.

References

Gelman, Carlin, Stern, and Rubin [1995]. *Bayesian Data Analysis*, Chapman & Hall.

Hastings, W.K. [1970]. "Monte Carlo sampling methods using Markov chains and their applications." *Biometrika* 57:97-109.

Hartigan, J.A. and Wong, M.A. [1979]. "A k-means clustering algorithm." *Applied Statistics* 28, 100-108.

Forgy, E. [1965]. "Cluster Analysis of multivariate data: efficiency vs. interpretability of classifications." *Biometrics* 21:768.

Ridgeway, Greg. [1997] "Finite discrete Markov process clustering." Technical Report MSR-TR-97-24, Microsoft Research.

Ross, Sheldon M. [1993]. *Probability Models 5th Edition*, Academic Press.

Stephens, Matthew [1996]. "Dealing with the multi-modal distributions of mixture model parameters." At http://www.stats.ox.ac.uk/~stephens/identify.ps.