Simultaneous Clustering on Representation Expansion for Learning Multimodel MDPs

Trevor Campbell **Robert H. Klein** **Alborz Geramifard** **Jonathan P. How**

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
77 Massachusetts Ave., Cambridge, MA 02139
{tdjc,rhklein,agf,jhow}@mit.edu

Abstract

This paper addresses the problem of model learning in a Markov decision process (MDP) that exhibits an underlying multiple model structure. In particular, each observed episode from the MDP has a latent classification that determines from which of an unknown number of models it was generated, and the goal is to determine both the number and the parameterization of the underlying models. The main challenge in solving this problem arises from the coupling between the separation of observations into groupings and the selection of a low-dimensional representation for each group. Present approaches to multiple model learning involve computationally expensive probabilistic inference over Bayesian nonparametric models. We propose Simultaneous Clustering on Representation Expansion (SCORE), an iterative scheme based on classical clustering and adaptive linear representations, which addresses this codependence in an efficient manner and guarantees convergence to a local optimum in model error. Both a batch and an incremental version of SCORE are presented. Empirical results on simulated domains demonstrate the advantages of SCORE when compared to contemporary techniques with respect to both sample and time complexity.

Keywords: clustering, representation expansion, multiple model learning

Acknowledgements

This work was supported by ONR MURI Grant N000141110688.

1 Introduction

A key component of model-based Reinforcement Learning (RL) techniques is to build an accurate model of a Markov Decision Process (MDP) from observed interactions [1, 2]. This paper considers the problem of model learning in a system that exhibits an underlying multiple model structure, where the number of underlying models is unknown. In particular, each of the latent models contains a complete description of the transition dynamics and reward of the MDP, and each observed episode (or "trajectory") has a latent classification that determines from which model it was generated. An example of such a scenario is a pursuit task, where an agent seeks to capture a target whose movement model can be neutral (e.g. random walk) or defensive (e.g. avoiding the agent), and where the agent does not know about these behaviors a priori. Such missions are of great interest to the autonomous planning community. For example, in mobile robot path planning problems the models can describe various types of terrain, or for aircraft routing problems the models can describe the evolution of various weather patterns.

There are two major challenges posed by this problem formulation. The first is that of *simultaneous model distinction and learning*: In order to learn the models, the algorithm must first separate the trajectories into groupings based on their latent classifications, but in order to separate the trajectories into such groupings, the algorithm must have a good parameterization of the models. The second is that *model distinction depends on the representation*: Since learning exact models is infeasible, a lower-dimensional representation is required, but the ability to separate trajectories into groupings depends on the chosen representation. Past representation expansion approaches [3, 4] do not take advantage of multiple-model structure, leading to high sample complexity and suboptimal policies that arise from averaging over all the underlying models. Multiple-model approaches [5, 6] assume rigid model structure and a known number of models. Bayesian nonparametric approaches [7, 8] can infer both the number of models and the representation, but these approaches do not scale well to typical RL problem sizes.

The main contribution of this paper is the simultaneous clustering on representation expansion (SCORE) architecture, which solves these problems using a combination of clustering and linear representation adaptation.

2 Background

A Markov decision process (MDP) is defined as tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ is a finite set of states, and $\mathcal{A}$ is a finite set of actions, P describes transition dynamics, R is a reward function, and γ is a discount factor. In this paper, we focus on learning given a fixed policy; consequently, $\mathcal{A}$ is removed from consideration. This situation occurs as a subcomponent of many approximate planning algorithms, such as the policy evaluation step of policy iteration [4]. Further, we make the assumption that the only uncertain component in the MRP model is an unknown scalar field $f : \mathcal{S} \to \mathbb{R}$: However, it is straightforward to extend the work presented here to the case of learning P and R without any partial knowledge. In this work, we make use of linear function approximation [4, 9], where $\phi : \mathcal{S} \to \mathbb{R}^m$ maps each state to m feature values. This reduces the goal of model learning to learning f_Φ, where

$$\Phi = \begin{bmatrix} \phi(s_1) & \cdots & \phi(s_{|\mathcal{S}|}) \end{bmatrix}^T, \Phi f_\Phi \approx f = \begin{bmatrix} f(s_1) & \cdots & f(s_{|\mathcal{S}|}) \end{bmatrix}^T. \tag{1}$$

Let the set of observed episodes be $\{y_i\}_{i=1}^N$, where $y_i = \{(s_{i1}, f_{i1}), \ldots, (s_{iT_i}, f_{iT_i})\}$, f_{ij} is a noisy unbiased estimate of $f(s_{ij})$, and $\phi(s_{ij}) \equiv \phi_{ij}$ for brevity. Then the problem of minimizing the sum of $|f_{ij} - {f_\Phi}^T \phi_{ij}|^2$ over all the observed samples for a single model is solved via

$$f_\Phi = \left[\sum_{i,j} \phi_{ij} \phi_{ij}^T \right]^{-1} \left[\sum_{i,j} \phi_{ij} f_{ij} \right], \tag{2}$$

where the sums are over the range $j \in \{1, \ldots, T_i\} \forall\, i \in \{1, \ldots, N\}$. To replace f with learning P and R (as mentioned earlier), simply replace f with R directly (as they are both $|\mathcal{S}|$-dimensional vectors), and add P by considering feature prediction errors and swapping vector norms for Frobenius norms where required. For a more thorough discussion of single model-based RL with a linear representation, the reader is directed to the analysis by Parr et al. [9]. Feature adaptation algorithms (such as Batch iFDD[2], which is used in this work) increase the feature dimension m (add a column to Φ) based on the resulting predictive errors $f_{ij} - {f_\Phi}^T \phi_{ij}$ in order to better capture the model.

3 Simultaneous Clustering On Representation Expansion (SCORE)

3.1 Problem Formulation

The problem considered in the present work extends the traditional MRP learning scenario by allowing an unknown number of models to be responsible for generating the observations. More precisely, suppose there is a set $\mathcal{F} = \{f_k\}_{k=1}^{|\mathcal{F}|}$ of unknown cardinality $|\mathcal{F}|$ of scalar fields defined on $\mathcal{S}$, each associated with an MRP. Given a collection of N observed trajectories, each trajectory having been generated by one of the MRPs, the goal is to infer the set $\mathcal{F}$, i.e. both $|\mathcal{F}|$ and $f_k \;\forall\; k = 1, \ldots, |\mathcal{F}|$. As it is often intractable to learn the full representation of f for even just a single system, a linear feature representation of the system is used to reduce the problem to learning $\mathcal{F}_\Phi = \{f_{\Phi k}\}_{k=1}^{|\mathcal{F}_\Phi|}$. However, the capability

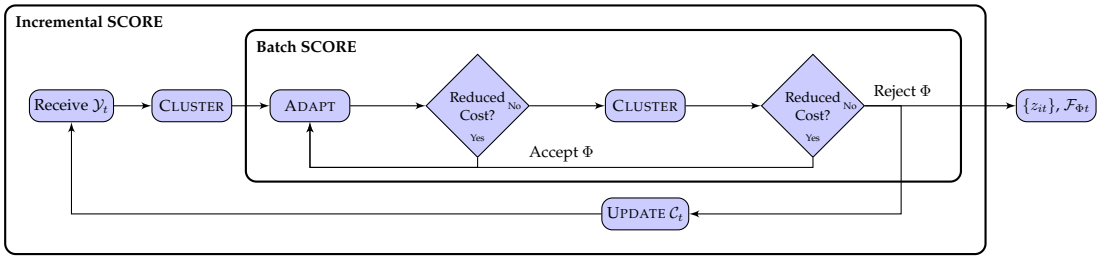


Figure 1: Algorithm block diagram.

to distinguish MRPs based on observed data is intimately linked to the particular chosen linear representation; thus, Φ itself must also be inferred in order to find the best grouping of the trajectories.

Based on the standard formulation of MRP model learning, the natural extension to the case with multiple models is a minimization of the sum of the squared predictive errors over all the trajectories from each MRP with respect to $f_{\Phi k}$. Let $z_i \in \{1, \ldots, |\mathcal{F}_\Phi|\}$ be the label denoting the index of the MRP from which episode y_i was generated. Then the squared predictive error of trajectory i in MRP k is

$$\delta_y^2(i,k) = \sum_{j=1}^{T_i} (f_{ij} - f_{\Phi k}^T \phi_{ij})^2. \tag{3}$$

Finally, define $|\Phi|$ to be the number of features (columns) in Φ, then the overall goal of multimodel learning is to solve the optimization problem

$$\min_{\Phi, \mathcal{F}_\Phi, \{z_i\}_{i=1}^N} \lambda|\mathcal{F}_\Phi| + \eta|\Phi| + \sum_{k=1}^{|\mathcal{F}_\Phi|} \left[\nu \|f_{\Phi k}\|^2 + \sum_{i:z_i=k} \delta_y^2(i,k) \right], \tag{4}$$

where $\lambda|\mathcal{F}_\Phi|$ (with $\lambda > 0$) is a cost penalty on the complexity of the learned model (based on past literature in the classical limits of Bayesian nonparametric models [10]), $\eta|\Phi|$ (with $\eta > 0$) is a cost penalty on the complexity of the representation, and $\nu\|f_{\Phi k}\|^2$ (with $\nu > 0$) is a regularization term. The optimization problem (4) is a mixed integer nonlinear program, with complexity that is nonlinear in N for both exact and heuristic methods [11], even with fixed $|\mathcal{F}_\Phi|$ and $|\Phi|$. As the size of the dataset grows, finding a good approximate solution with standard methods becomes intractable. Thus the focus of this paper is to present a tractable algorithm for solving (4) for large (and possibly increasing with time) N.

3.2 Batch SCORE

The batch SCORE algorithm, shown in Figure 1, is an iterative algorithm for minimizing the objective (4). It starts by using the CLUSTER algorithm (5) to find an initial clustering of the data given an initial representation Φ. Then, the ADAPT algorithm expands the representation using the predictive error for each observation with respect to its assigned cluster. If the expansion results in a decrease in (4), the expansion is accepted and the loop begins again. If it does not (due to the η penalty), the CLUSTER algorithm is run in an attempt to build a new clustering in the new representation. If that clustering yields an overall reduction in (4), the expansion is accepted and the loop begins again. If (4) is still not reduced, the algorithm terminates and returns $\mathcal{F}_\Phi$, Φ, and $\{z_i\}_{i=1}^N$.

CLUSTER

$$f_{\Phi k} \leftarrow (\nu I + \sum_{i,j:z_i=k} \phi_{ij}\phi_{ij}^T)^{-1} \sum_{i,j:z_i=k} f_{ij}\phi_{ij} \tag{5}$$

$$z_i \leftarrow \operatorname*{argmin}_k \begin{cases} \delta_y^2(i,k) & k \in \{1,\ldots,K\} \\ \lambda + \nu\|f_{\Phi k}\|^2 + \delta_y^2(i,k) & k = K+1 \end{cases} \tag{6}$$

ADAPT

$$\theta^* \leftarrow \text{iFDD[2]} \tag{7}$$

$$\Phi \leftarrow \begin{bmatrix} \Phi & \theta^* \end{bmatrix} \tag{8}$$

The parameters of Batch SCORE are $\lambda, \eta, \nu \in \mathbb{R}_+$, where λ, η are as in (4), and ν is the regularization parameter. K is defined as the number of clusters currently instantiated by the algorithm, and a new cluster is only created when the label update selects $K+1$ as the minimum cost index; thus, increasing λ reduces the number of clusters created. Note that $\delta_y^2(i, K+1)$ is equal to the distance from an observation y_i to the model $f_{\Phi(K+1)}$ found by using only the data from that observation; this term is required to account for the fact that the best possible new model $f_{\Phi(K+1)}$ for y_i has a nonzero error (due to a possibly poor representation). The per-iteration complexity of batch SCORE is dominated by the $O(D|\Phi|^2 + K|\Phi|^3)$ model update step in (5), where $D = \sum_{i=1}^N T_i$ is the total number of transitions observed. The CLUSTER step may be viewed as modified version of the DP-means [10] algorithm; this close relationship guarantees that Batch SCORE finds a local minimum in (4), and terminates in a finite number of iterations (Theorem 3.1).

Theorem 3.1. *Batch SCORE monotonically decreases the objective in (4), finds a local optimum, and terminates in a finite number of iterations (proof omitted for brevity).*

3.3 Incremental SCORE

While simple to implement and theoretically promising, the batch algorithm becomes computationally intractable as N grows. Incremental SCORE fixes this issue by processing smaller batches of size $N_t \ll N$ in a sequence $t = 1, 2, \ldots$ and retaining learned information between batches, thus reducing the per-iteration complexity to $O(D_t|\Phi|^2 + K|\Phi|^3)$ (where $D_t \ll D$). This learned information is transferred by retaining the old models from previous batches, $f'_{\Phi k}$, and the representation used to create them, Φ'. Define the error between the old and new models (in the old subspace) to be

$$\delta_M^2(k) = \|\Sigma^{\frac{1}{2}}\Phi'\left(f'_{\Phi k} - (\Phi'^T\Sigma\Phi')^{-1}\Phi'^T\Sigma\Phi f_{\Phi k}\right)\|_2^2$$

where Σ is a diagonal matrix of the sample approximation of the stationary distribution of the Markov system. Further, define $\mathcal{C}_t = \{f'_{\Phi k}, w_{k(t-\Delta t_k)}, \Delta t_k\}_{k=1}^{|\mathcal{C}_t|}$ to be the set of old models $f'_{\Phi k}$, weights $w_{k(t-\Delta t_k)}$, and ages Δt_k. Define n_{kt} to be the sum over all transitions in all trajectories assigned to cluster k in batch t. Finally, to transfer information about the representation, errors are computed in iFDD using all transitions observed in *all* received batches of data (these cumulative errors are updated incrementally after each batch is processed). The steps of Incremental SCORE, shown in the architecture in Figure 1, are as follows:

CLUSTER

$$\gamma_{kt} \leftarrow \begin{cases} \left(\left(w_{k(t-\Delta t_k)}\right)^{-1} + \Delta t_k \tau\right)^{-1} & k \le |\mathcal{C}_t| \\ 0 & k > |\mathcal{C}_t| \end{cases}$$

$$A \equiv \Phi^T\Sigma\Phi', \quad B \equiv \Phi'^T\Sigma\Phi' \tag{9}$$

$$f_{\Phi kt} \leftarrow (\gamma_{kt}AB^{-1}A^T + \nu I + \sum_{i,j:z_{it}=k}\phi_{ij}\phi_{ij}^T)^{-1}(\gamma_{kt}Af'_{\Phi k} + \sum_{i,j:z_{it}=k} f_{ij}\phi_{ij})$$

$$z_{it} \leftarrow \underset{k}{\operatorname{argmin}} \begin{cases} \delta_y^2(i,k) & n_{kt} > 0 \\ \nu\|f_{\Phi k}\|^2 + \delta_y^2(i,k) + \gamma_{kt}\delta_M^2(k) & n_{kt} = 0, k \le |\mathcal{C}_t| \\ \lambda + \nu\|f_{\Phi k}\|^2 + \delta_y^2(i,k) & n_{kt} = 0, k > |\mathcal{C}_t| \end{cases} \tag{10}$$

ADAPT

$$\theta^* \leftarrow \text{iFDD (see [2])} \tag{11}$$

$$\Phi \leftarrow \begin{bmatrix} \Phi & \theta^* \end{bmatrix} \tag{12}$$

UPDATE $\mathcal{C}_t$

$$w_{kt} \leftarrow \gamma_{kt} + n_{kt}$$

$$\Delta t_k \leftarrow \begin{cases} 1 & n_{kt} > 0 \\ \Delta t_k + 1 & n_{kt} = 0 \end{cases} \tag{13}$$

The parameter $\tau > 0$ controls how quickly the weight of old information decays (increasing τ causes old information to decay faster). The parameter γ_{kt} is set at the beginning of each CLUSTER step, and controls the weight on prior information in (9). During the label assignment step, the incremental algorithm can "revive" an old cluster; here, $\delta_M^2(k)$ is the error between $f_{\Phi kt}$ and $f_{\Phi k(t-\Delta t_k)}$ after using only observation y_{it} to compute the model update in (9). This algorithm may be seen as a modified version of Dynamic Means [12], and due to this connection, incremental SCORE is guaranteed to converge in a finite number of iterations (Theorem 3.2).

Theorem 3.2. *For each batch $\mathcal{Y}_t$, the inner loop of Incremental SCORE monotonically decreases*

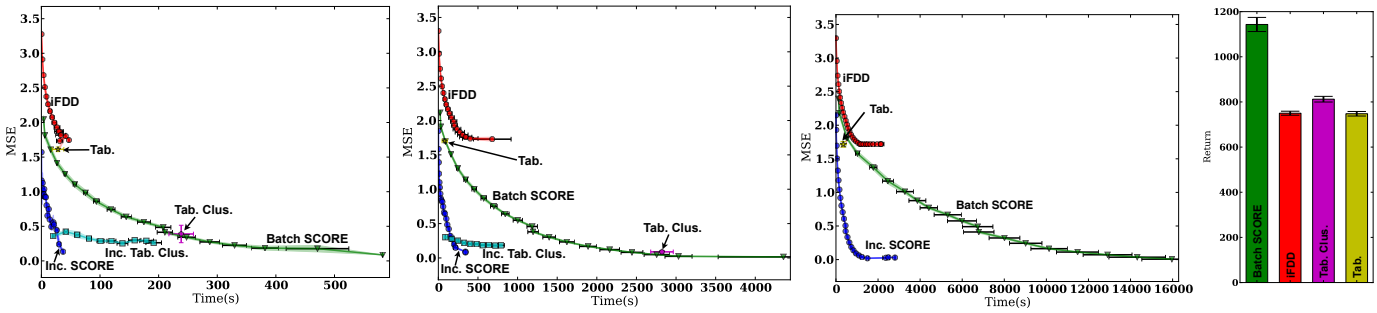
$$\eta|\Phi| + \sum_{k:n_{kt}>0}\lambda\left[\Delta t_k = 0\right] + \gamma_{kt}\delta_M^2(k) + \nu\|f_{\Phi kt}\|^2 + \sum_{i:z_{it}=k}\delta_y^2(i,k), \tag{14}$$

finds a local optimum, and terminates in a finite number of iterations (proof omitted for brevity).

4 Experimental Results

In this simulation experiment, we used SCORE to capture the effects of thermal updrafts on an unmanned aerial vehicle(UAV). The domain was a two-dimensional 25×25 grid world, and the UAV could take actions selected from $\{\uparrow, \downarrow, \leftarrow, \rightarrow, \cdot\}$. Each state s had a mean updraft strength selected from $\mu_k(s) = \{0, 1, 5, 10\}$, with $K = 3$ different latent mean updraft strength fields, and $k \in \{1, \ldots, K\}$. At the start of each episode, a label $z \sim$ UNIFORM$(1, K)$ was sampled, and in each step t the UAV was given an altitude boost of $r_t \sim \mathcal{N}(\mu_z(s_t), \sigma = 0.5)$. The goal of the experiment was: 1) To learn the latent updraft fields $\mu_k(s)$ from a set of training data (collected with a uniform random policy); 2) To create a $Q(s,a)$ function for each learned field; and 3) To take actions by keeping track of the likelihood of each latent field given altitude boost observations, weighting the different $Q(s,a)$ functions accordingly, and acting greedily with respect to the weighted $Q(s,a)$ function (a Q_{MDP} approach[13]). We compared six approaches: batch/incremental SCORE (Batch SCORE/Inc. SCORE), batch/incremental clustering with a tabular representation (Tab. Clus./Inc. Tab. Clus.), no clustering with iFDD (iFDD), and no clustering with a tabular representation (Tab.). The initial representation for the approaches with feature expansion consisted of an indicator function for each value of the two axes in the plane (i.e. 50 features for the 25×25 grid). The tabular representation had an indicator function feature for each grid cell (i.e. 625 features for the 25×25 grid).

Figures 2(a)-2(c) show the mean squared error (MSE) of altitude boost prediction (with 95 % confidence intervals) computed with respect to the sample stationary distribution. The X-axis represents the clock time in seconds. Each datapoint in Figures 2(a)-2(c) represents one expansion for iFDD, one round of expansion and clustering for SCORE, or one round of clustering for Tab. Clus.. Note that Tab. Clus. and Inc. Tab. Clus. failed to run on the large dataset (Figure 2(c)) due to memory limitations.



(a) 200 Observations (b) 1000 Observations (c) 5000 Observations (d) Mean Return

Figure 2: (a)-(c): Model error of batch SCORE (Batch SCORE), incremental SCORE (Inc. SCORE), batch clustering with a tabular representation (Tab. Clus.), incremental clustering with a tabular representation (Inc. Tab. Clus.), no clustering with a tabular representation (Tab.), and no clustering with feature expansion (iFDD). (d): Mean Q_{MDP} return using a training dataset of 200 observations.

From Figures 2(a)-2(c), it is clear that feature expansion or clustering alone is inadequate. Both Tab. and iFDD converged quickly but to a poor solution, due to an inability to discover the underlying latent models. Tab. Clus. yielded a lower MSE but required significantly more time as the size of the observations grew. Batch SCORE yielded the lowest final modeling error of the batch algorithms (especially when data was scarce), demonstrating its capability to both generalize and discover underlying models. The incremental versions of Tab. Clus. and SCORE had similar final MSEs to their batch counterparts, but required significantly less computation time; this reduction was magnified as the size of the dataset increased.

The models learned using the 200 observation training dataset were then used in a planning exercise, where the UAV altitude boost was used as the reward signal; a comparison of the cumulative return is shown in Figure 2(d) with 95% confidence intervals. The incremental algorithms (not shown) did not perform as well as their batch counterparts on this dataset; the sample stationary distribution for the small individual batches was not reflective of the true stationary distribution, and thus their MSEs were not reflective of their true predictive accuracy over the whole state space. For such small datasets, batch algorithms are preferred. The performance using models generated from the Tab. and iFDD algorithms is approximately the same, in agreement with Figures 2(a)-2(c); using these models, the agent effectively planned on an average of all the true updraft fields, so that the optimal actions to take become less clear. Tab. Clus. performed slightly better, as the planning agent was able to identify which of its underlying learned models it was experiencing; however, the lack of data prevented it from learning accurate models. Batch SCORE significantly outperformed the other batch algorithms, as it simultaneously identified the underlying multiple model structure and constructed individual models which generalized well using scarce data.

5 Conclusion

This work addressed the problem of model learning in a system that exhibits an underlying multiple model structure using the Simultaneous Clustering on Representation Expansion (SCORE) algorithm. SCORE addresses the codependence between representation selection and observation clustering, and guarantees convergence to a local optimum in model error. Empirical results demonstrated the advantage of this approach when compared to standard techniques with respect to both sample and time complexity.

References

[1] R. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the 7th International Conference on Machine Learning*, pages 216–224, 1990.

[2] A. Geramifard, T. J. Walsh, N. Roy, and J. How. Batch iFDD: A Scalable Matching Pursuit Algorithm for Solving MDPs. In *Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Bellevue, Washington, USA, 2013. AUAI Press.

[3] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An Analytic Solution to Discrete Bayesian Reinforcement Learning. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[4] R. S. Sutton, C. Szepesvári, A. Geramifard, and M. Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 528–536, 2008.

[5] M. Haruno, D. M. Wolpert, and M. Kawato. MOSAIC Model for Sensorimotor Learning and Control. *Neural Computation*, 13(10):2201–2220, 2001.

[6] D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.

[7] E. B. Fox. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena*. PhD thesis, Massachusetts Institute of Technology, 2009.

[8] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy. A Bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31(4):383–400, 2011.

[9] R. Parr, L. Li, G. Taylor, C. Painter-Wakefield, and M. L. Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 752–759, New York, NY, USA, 2008. ACM.

[10] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via Bayesian nonparametrics. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012.

[11] D. Arthur, B. Manthey, and H. Röglin. k-means has polynomial smoothed complexity. In *Proceedings of the 50th Symposium on Foundations of Computer Science*, 2009.

[12] T. Campbell, M. Liu, B. Kulis, and J. How. Dynamic clustering via asymptotics of the dependent dirichlet process. *arXiv ePrint 1305.6659*, 2013.

[13] M. Littman, A. Cassandra, and L. Kaelbling. Learning policies for partially observable environments: scaling up. In *International Conference on Machine Learning (ICML)*, pages 362–370, San Francisco, CA, 1995.