The Adaptive k-Meteorologists Problem and Its Application to Structure Learning and Feature Selection in Reinforcement Learning

Carlos Diuk CDIUK@CS.RUTGERS.EDU
Lihong Li LIHONG@CS.RUTGERS.EDU
Bethany R. Leffler BLEFFLER@CS.RUTGERS.EDU
RL^3 Laboratory, Department of Computer Science, Rutgers University, Piscataway, NJ USA 08854

Abstract

The purpose of this paper is three-fold. First, we formalize and study a problem of learning probabilistic concepts in the recently proposed KWIK framework. We give details of an algorithm, known as the Adaptive k-Meteorologists Algorithm, analyze its sample-complexity upper bound, and give a *matching* lower bound. Second, this algorithm is used to create a new reinforcement-learning algorithm for factored-state problems that enjoys significant improvement over the previous state-of-the-art algorithm. Finally, we apply the Adaptive k-Meteorologists Algorithm to remove a limiting assumption in an existing reinforcement-learning algorithm. The effectiveness of our approaches is demonstrated empirically in a couple benchmark domains as well as a robotics navigation problem.

1. Introduction

Imagine that you just moved to a new town that has multiple (k) radio and TV stations. Each morning, you tune in to one of the stations to find out what the weather will be like. Which of the k different meteorologists making predictions every morning is the most trustworthy? Let us imagine that, to decide on the best meteorologist, each morning for the first M days you tune in to all k stations and write down the probability that each meteorologist assigns to the chances of rain. Then, every evening you write down a 1 if it rained, and a 0 if it didn't. Can this data be used to determine who is the best meteorologist?

In the example above, each meteorologist is allowed to predict the chances of rain, rather than the binary outcome of whether it will rain. Such predictions are termed *probabilistic concepts* (Kearns & Schapire, 1994; Yamanishi, 1992). They extend the notion of deterministic concepts by allowing an instance or example to belong to a class with certain probability. While Kearns and Schapire (1994) study *PAC*-learning of probabilistic concepts, this paper considers learning in the recently proposed *KWIK* framework (Li et al., 2008).

The first contribution of this paper is to formalize two KWIK learning problems, known as the *k-Meteorologists* and the *Adaptive k-Meteorologists*, expand an algorithmic idea introduced by Li et al. (2008), and give a polynomial sample-complexity upper bound for the resulting algorithms. Furthermore, a new matching lower bound is given indicating the optimality of our algorithms.

The second and third contributions are to demonstrate how the algorithm for the (Adaptive) k-Meteorologists Problem can be applied to two important problems in reinforcement learning: structure learning of factored problems, and feature selection in a robot application.

First, we consider the problem of learning in a factored-state Markov decision process (MDP) where the transition dynamics are represented by a *Dynamic Bayes Network or DBN* (Dean & Kanazawa, 1989; Boutilier et al., 1999) for which we do not know the structure but have an upper bound on its in-degree. Strehl et al. (2007) proposed SLF-$R_{\max}$, the first algorithm to solve this problem that is PAC-MDP (Strehl et al., 2006b). Based on the solution to the Adaptive k-Meteorologists Problem, we develop a new PAC-MDP algorithm known as Met-$R_{\max}$ which improves SLF-$R_{\max}$'s sample complexity for structure discovery from $\tilde{O}(n^{2D})$ to $\tilde{O}(n^D)$, where n is the number of factors in the DBN and D the maximum in-degree. Empirical experiments in the Stocks-trading domain (Strehl et al., 2007) and the System Administrator domain (Guestrin et al., 2003) demonstrate the superior sample efficiency of our new approach.

Second, we present the problem of a robot that has to decide which of multiple sensory inputs is relevant for

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009.

capturing an environment's transition dynamics. Leffler et al. (2007) introduced the RAM-$R_{\max}$ algorithm, which assumes that a classifier is provided that permits clustering different states according to their effects on transition dynamics, and leverages that knowledge to learn very efficiently. In this paper, we eliminate this assumption and introduce an algorithm called SCRAM-$R_{\max}$ that uses our solution to the k-Meteorologists problem to enable the agent to learn the relevant sensory input from experience while acting online. In the example we consider, should the robot cluster states according to its camera readings of surface color, surface texture, its IR sensor reading, or some combination of them? In robot-navigation environments, the number of possible state classifiers could be as large as the number of sensors on the robot. As in the structure-learning problem, we replace the requirement that a single structure has to be provided as input by the assumption that a whole family of possible structures is available to be considered, and use the k-Meteorologists to choose the structure that best models the environment's dynamics.

2. KWIK Learning Probabilistic Concepts

Probabilistic concepts are a useful generalization of deterministic concepts and are able to capture uncertainty in many real-life problems, such as the weather broadcasting example described in the previous section. Formally, a probabilistic concept h is a function that maps an input space X to the output space $Y = [0, 1]$; $h : X \mapsto Y$. In the meteorologist example, every $x \in X$ corresponds to the features that can be used to predict chances of rain, and $h(x)$ indicates the probability that x is in the concept, namely, the chances that it will rain on that day. The hypothesis class H is a set of probabilistic concepts: $H \subseteq (X \to Y)$.

2.1. The KWIK Model

Using tools from statistical learning theory, Kearns and Schapire (1994) study how to learn probabilistic concepts in the PAC model (Valiant, 1984). Below, we formulate a related problem in the recently proposed KWIK model (Li et al., 2008). KWIK stands for Knows What It Knows. It is a computational supervised-learning model that requires self-awareness of prediction errors and is useful in learning problems such as reinforcement learning and active learning where active exploration can impact the training examples the learner is exposed to.

Two parties are involved in this learning process. The *learner* runs a learning algorithm and makes predictions; while the *environment*, which represents an instance of a KWIK learning problem, provides the learner with inputs and observations. A KWIK "run" proceeds as follows:

- The hypothesis class H, accuracy parameter ϵ, and confidence parameter δ are known to both the learner and the environment.
- The environment selects a target concept $h^* \in H$.
- For *timestep* $t = 1, 2, 3, \ldots$,
  - The environment selects an input $x_t \in X$ in an *arbitrary* way and informs the learner. The target value $y_t = h^*(x_t)$ is unknown to the learner.
  - The learner predicts an output $\hat{y}_t \in Y \cup \{\perp\}$ where $\perp$ indicates that the learner is unable to make a good prediction of y_t.
  - If $\hat{y}_t = \perp$, the learner makes a stochastic observation $z_t \in Z = \{0, 1\}$ of the output y_t: $z_t = 1$ with probability y_t and 0 otherwise.

We say that H is *KWIK-learnable* if there exists an algorithm $\mathcal{A}$ with the following property: for any $0 < \epsilon, \delta < 1$, two requirements are satisfied with probability at least $1-\delta$ in a whole run of $\mathcal{A}$ according to the KWIK protocol above:

1. (*Accuracy Requirement*) If $\hat{y}_t \neq \perp$, it must be ϵ-accurate: $|\hat{y}_t - y_t| < \epsilon$;
2. (*Sample Complexity Requirement*) The total number of $\perp$s predicted during the whole run, denoted $\zeta(\epsilon, \delta)$, is bounded by a function polynomial in $1/\epsilon$ and $1/\delta$.

We call $\zeta(\epsilon, \delta)$ a *sample complexity* of $\mathcal{A}$. Furthermore, H is *efficiently KWIK-learnable* if the per-timestep time complexity of $\mathcal{A}$ is polynomial in $1/\epsilon$ and $1/\delta$.

2.2. The (Adaptive) k-Meteorologists Problems

The (Adaptive) k-Meteorologists Problems consider efficient learning of probabilistic concepts in the KWIK framework. In the k-Meteorologist Problem, the learner is given a finite set of k probabilistic concepts: $H = \{h_1, h_2, \ldots, h_k\}$, where $h_i : X \to Y$ for all $i = 1, \ldots, k$. The task of KWIK-learning a target concept $h^* \in H$ can be understood as one of identifying the true but unknown concept from a set of k candidates following the learning process defined formally in the previous subsection.

A related problem is extensively studied in expert algorithms (see, *e.g.*, Cesa-Bianchi et al. (1997)), where the learner always makes a prediction $\hat{y}_t \in [0, 1]$ based on the predictions of all meteorologists. The goal of the learner is to make predictions so that the number of mistakes she makes (more generally, the loss she suffers) is not much larger than the best meteorologist or the best combinations of the meteorologist, in hindsight. In contrast, a KWIK algorithm must output $\perp$ ("I don't know") when its prediction may be incorrect. This feature is essential in the reinforcement-learning problems we will consider.

In some learning problems, the candidate concepts, h_i, are not provided as input. Instead, they have to be learned by the learner itself. This motivates a more general version of the k-Meteorologists Problem, which we term as the Adaptive k-Meteorologists Problem. Here, the learner is given k classes of hypotheses, $H_1, \ldots, H_k$, and also provided with k sub-algorithms, $\mathcal{A}_1, \ldots, \mathcal{A}_k$, for KWIK-learning these classes. The goal of the learner is to make use of these sub-algorithms to KWIK-learn the *union* of these hypothesis classes: $H = H_1 \cup \cdots \cup H_k$.

2.3. Solution

The k-Meteorologists Problem is a special case of the Adaptive k-Meteorologists Problem where every hypothesis class H_i contains exactly one hypothesis: $H_i = \{h_i\}$. For the sake of simplicity, we start with the simpler k-Meteorologists Problem to explain the intuition of our algorithm, and then provide detailed pseudo-code descriptions for the adaptive version.

The major challenge in the k-Meteorologists Problem is that the learner only observes stochastic binary labels while she is required to make predictions about the label probabilities. A natural idea is to get sufficient labels for the same input x and then estimate $\Pr(z = 1|x)$ by the relative frequency. But since inputs may be drawn adversarially, this approach must have a sample complexity of $\Omega(|X|)$.

Here, we expand an idea outlined by Li et al. (2008) to avoid the dependence on the size of X. Suppose $z_t \in \{0, 1\}$ is the label acquired in timestep t. Define the squared error of meteorologist h_i to be $e_t = (h_i(x_t) - z_t)^2$. We may maintain cumulative squared prediction errors for individual meteorologists. It can be shown that the target probabilistic concept, h^*, will have the smallest squared error *on average*. If any concept h_i has a much larger cumulative error than another concept h_j, it follows that $h_i \neq h^*$ with high probability.

Algorithm 1 provides a solution to the Adaptive k-Meteorologists Problem, in which the additional parameter m will be specified in Theorem 1. Essentially, the algorithm runs all the k sub-algorithms simultaneously and does all $\binom{k}{2}$ pairwise comparisons among the k probabilistic concepts. If any probabilistic concept returns $\perp$, the algorithm outputs $\perp$ and obtains a stochastic observation z_t to allow the sub-algorithms to learn (Lines 7–9). Now suppose no probabilistic concept returns $\perp$. If the set of predictions is consistent then an accurate prediction can be made (Line 12) although the algorithm does not know which concept is h^*. Otherwise, the algorithm outputs $\perp$ and then acquires a label which contributes to distinguishing at least one pair of meteorologists (Lines 15–21). A candidate concept is removed if there is statistically significant evidence that it is worse than another concept (Line 19).

Algorithm 1 The Adaptive k-Meteorologists Algorithm.

1: Input: ϵ, δ, m, $H_1, \ldots, H_k$, $\mathcal{A}_1, \ldots, \mathcal{A}_k$.
2: Run each subalgorithm $\mathcal{A}_i$ with parameters $\frac{\epsilon}{8}$ and $\frac{\delta}{k+1}$.
3: $R \leftarrow \{1, 2, \ldots, k\}$.
4: $c_{ij} \leftarrow 0$ and $\Delta_{ij} \leftarrow 0$ for all $1 \leq i < j \leq n$.
5: **for** $t = 1, 2, 3, \ldots$ **do**
6: Obtain x_t and run each $\mathcal{A}_i$ to get its prediction, $\hat{y}_{ti}$.
7: **if** $\hat{y}_{ti} = \perp$ for some $i \in R$ **then**
8: Let $\hat{y}_t = \perp$ and observe $z_t \in \mathbb{Z}$.
9: Send z_t to all subalgorithms $\mathcal{A}_i$ with $\hat{y}_{ti} = \perp$.
10: **else**
11: **if** $|\hat{y}_{ti} - \hat{y}_{tj}| \leq \epsilon$ for all $i, j \in R$ **then**
12: Let $\hat{y}_t = (\max_{i \in R} \hat{y}_{ti} + \min_{i \in R} \hat{y}_{ti})/2$.
13: **else**
14: Let $\hat{y}_t = \perp$ and observe z_t.
15: **for all** $i, j \in R$ such that $|\hat{y}_{ti} - \hat{y}_{tj}| \geq \frac{\epsilon}{2}$ **do**
16: $c_{ij} \leftarrow c_{ij} + 1$.
17: $\Delta_{ij} \leftarrow \Delta_{ij} + (\hat{y}_{ti} - z_t)^2 - (\hat{y}_{tj} - z_t)^2$.
18: **if** $c_{ij} \geq m$ **then**
19: $R \leftarrow R \setminus \{I\}$ where $I = i$ if $\Delta_{ij} > 0$ and $I = j$ otherwise.
20: **end if**
21: **end for**
22: **end if**
23: **end if**
24: **end for**

2.4. Analysis

This section gives *matching* upper and lower sample-complexity bounds for Algorithm 1. We only give proof sketches here, but complete details are found in Li (2009).

Observe that every $\perp$ output by Algorithm 1 is either from some sub-algorithm (Line 8) or from the main algorithm when it gets inconsistent predictions from different probabilistic concepts (Line 14). Thus, the sample complexity of Algorithm 1 is at least the sum of the sample complexities of those sub-algorithms plus the additional $\perp$s required to figure out the true h^* among the k candidates. The following theorem formalizes this observation:

Theorem 1 *Let $\zeta_i(\cdot, \cdot)$ be a sample complexity of sub-algorithm $\mathcal{A}_i$. By setting $m = O\left(\frac{1}{\epsilon^2} \ln \frac{k}{\delta}\right)$, the sample complexity of Algorithm 1 is at most*

$$\zeta^*(\epsilon, \delta) = O\left(\frac{k}{\epsilon^2} \ln \frac{k}{\delta}\right) + \sum_{i=1}^{k} \zeta_i\left(\frac{\epsilon}{8}, \frac{\delta}{k+1}\right).$$

Proof: (sketch) The proof has four steps. First, we show that the squared error of the target hypothesis must be the smallest *on average*. Second, if some hypothesis is $\frac{\epsilon}{8}$-accurate (as required by Line 2 in Algorithm 1), its average squared error is still very close to the average squared

error of the predictions of h^*. Third, by setting m appropriately (as given in the theorem statement), we can guarantee that only sub-optimal hypotheses are eliminated in Line 19 with high probability, by Hoeffding's inequality. Finally, the condition in Line 15 guarantees that the total number of $\perp$s outputted in Line 14 is bounded by the first term in the desired bound of the theorem. □

Theorem 1 indicates that the additional sample complexity introduced by Algorithm 1, compared to the unavoidable term, $\sum_i \zeta_i$, is on the order of $\frac{k}{\epsilon^2} \ln \frac{k}{\delta}$. The following theorem gives a matching lower bound (modulo constants), implying the optimality of Algorithm 1 in this sense.

Theorem 2 *A sample-complexity lower bound for the k-Meteorologists Problem is*

$$\zeta_* (\epsilon, \delta) = \Omega \left(\frac{k}{\epsilon^2} \ln \frac{k}{\delta} \right).$$

Proof: (sketch) The proof is through a reduction from 2-armed bandits (Mannor & Tsitsiklis, 2004) to the k-Meteorologists Problem. The idea is to construct input–observation pairs in the KWIK run so that the first $k-1$ hypotheses, $h_1, \ldots, h_{k-1}$, have to be eliminated one by one before the target hypothesis, $h^* = h_k$, is discovered. Each elimination of h_i (for $i < k$) can be turned into identifying a sub-optimal arm in a 2-armed bandit problem, which requires $\Omega(\frac{1}{\epsilon^2} \ln \frac{1}{\delta})$ sample complexity (Mannor & Tsitsiklis, 2004). Based on this lower bound, we may prove this theorem by requiring that the total failure probability in solving the k-Meteorologists Problem is δ. □

3. Structure Learning in Factored-state MDPs

Efficient RL algorithms for flat MDPs (Kearns & Singh, 2002; Brafman & Tennenholtz, 2002; Kakade, 2003; Strehl et al., 2006a; Strehl et al., 2006b)) have superlinear or quadratic dependence on the number of states, rendering them unsatisfactory for large-scale problems. In many real-life problems of interest, however, special structures can be leveraged to avoid an explicit dependence on the number of states. In this section, we consider problems where the dynamics and reward structures of the MDP can be succinctly modelled as a DBN. Efficient RL algorithms like factored E^3 (Kearns & Koller, 1999) and Factored-$R_{\max}$ (Strehl, 2007) have a sample complexity that is independent of the number of states, but assume the structure of the DBN is *known*.

In this section, we introduce Met-$R_{\max}$, an algorithm for factored-state MDPs whose transition dynamics are represented by a DBN with *unknown* structure but known maximum in-degree D. Met-$R_{\max}$ shows a significant improvement in the sample complexity of structure discovery over SLF-$R_{\max}$, the state-of-the-art algorithm proposed by Strehl et al. (2007). Empirical evidence is provided in Section 3.3. The assumption that D is known is a common one in the structure-discovery literature for Bayes networks (*e.g.*, Abbeel et al. (2006)), and is used by both SLF-$R_{\max}$ and Met-$R_{\max}$. It is an open question how to discover DBN structure efficiently under weaker assumptions.

3.1. Met-$R_{\max}$

The Met-$R_{\max}$ algorithm follows the same structure as SLF-$R_{\max}$ but replaces one of its subroutines by the Adaptive k-Meteorologists Algorithm. We refer the reader to Strehl et al. (2007) for details of SLF-$R_{\max}$ and we just note that the example they provide for an "admissible algorithm for the Structure-Learning problem" (Strehl et al., 2007) is replaced in Met-$R_{\max}$ by the Adaptive k-Meteorologists Algorithm. We only explain here how we use the Adaptive k-Meteorologists as an admissible structure learner.

Let us assume the transition dynamics of a factored-state MDP are represented by a DBN with n binary factors and maximum in-degree D. For any given factor f_i and action a, we must consider as possible parents all $\binom{n}{D}$ subsets of factors. Each parent set itself specifies a hypothesis sub-class (corresponding to H_i in Adaptive k-Meteorologists), and every hypothesis in this sub-class can be KWIK-learned (Li et al., 2008; Li, 2009). We thus simply initialize Algorithm 1 with $k = \binom{n}{D}$ "meteorologists", each trying to predict the outcome of f_i under action a based on the corresponding parent subset.

3.2. Analysis

The sample complexity of the state-of-the-art structure-discovery algorithm, due to Strehl et al. (2007), is

$$O \left(\frac{k^2}{\epsilon^2} \ln \frac{k}{\delta} \right),$$

where $k = \binom{n}{D} = O(n^D)$ is the number of possible parents in the DBN. Hence, Theorem 1 suggests the improvement of our algorithm is on the order of $O(n^D)$ which is substantial if n or D is large. Section 3.3 provides empirical evidence showing superiority of Met-$R_{\max}$ over SLF-$R_{\max}$.

Using the theoretical tools of Strehl et al. (2007), we can show that Met-$R_{\max}$ is PAC-MDP with a sample complexity of exploration (Kakade, 2003)

$$\kappa = O \left(\frac{n^{D+3} AD}{\epsilon^3 (1-\gamma)^6} \ln \frac{nA}{\delta} \ln \frac{1}{\epsilon(1-\gamma)} \right).$$

That is, if we run Met-$R_{\max}$ for infinitely many timesteps, then with probability at least $1-\delta$, the non-stationary policy[1] in the visited states is ϵ-optimal except for at most

[1] The learner's action-selection policy changes over time as the

κ many timesteps. Therefore, Met-$R_{\max}$ is a provably sample-efficient reinforcement-learning algorithm that is able to efficiently explore the environment (modelled as a DBN) and discover the structure of its dynamics. Interested readers are referred to complete details and extensions studied by Li (2009).

3.3. Experiments

We present experiments in two domains from the literature that are commonly represented as factored-state MDPs: Stock trading (Strehl et al., 2007) and System Administrator (Guestrin et al., 2003).

Exact planning (that is, finding an optimal policy) in an MDP modelled as a DBN is computationally hard although efficient and effective approximate methods have been developed. Since the present paper focuses on sample-complexity issues, we chose value iteration to compute optimal value functions and policies in our experiments.

3.3.1. Stock-Trading domain

This domain was introduced by Strehl et al. (2007) to illustrate their approach. We use the same domain and the same parameters to show how Met-$R_{\max}$ solves the same problem but improves performance, in correspondence with the new bound presented in section 3.2.

As in Strehl et al. (2007), we ran experiments on an instance of the domain consisting of 3 sectors and 2 stocks per sector. We used the same parameters used by the authors ($m = 10$ for Factored-$R_{\max}$, $m = 20$ and $\epsilon_1 = 0.2$ for SLF-$R_{\max}$). The parameter m indicates the number of samples required by each meteorologist before it assumes its prediction is known. A parameter search over a coarse grid was conducted for Met-$R_{\max}$, and the results presented use $m = 7$, which is the smallest value of m for which the algorithm converges to an optimal policy. Each experiment was repeated 10 times and the results averaged with confidence intervals plotted.

Figure 1 shows the reward accumulated by each agent per step of experience. As expected, the fastest algorithm to converge to a near-optimal policy and maximize reward is Factored-$R_{\max}$, which receives the DBN structure as input. SLF-$R_{\max}$ converges to a near-optimal policy after a number of additional steps, presumably needed to infer the underlying structure. Met-$R_{\max}$, converges to the same policy significantly faster. In fact, the performance of Met-$R_{\max}$ (where the structure has to be learned) and that of Factored-$R_{\max}$ (where the structure is given as input) is within margin of error in this particular example.

learner experiences state transitions. It is thus a non-stationary policy rather than a stationary policy (Puterman, 1994).

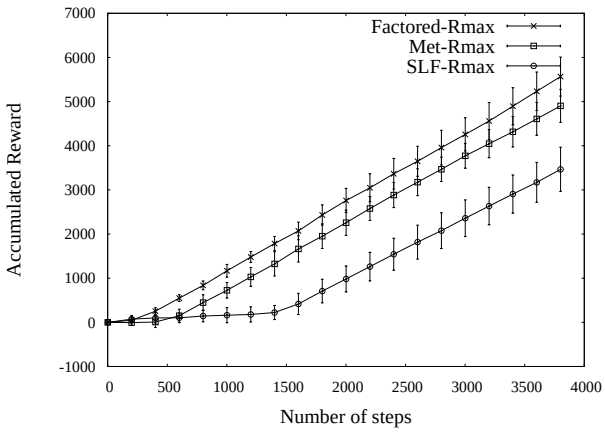


Figure 1. Cumulative reward on each timestep for the three algorithms on the Stock-trading domain: Factored-$R_{\max}$ was given the DBN structure, while Met-$R_{\max}$ and SLF-$R_{\max}$ were not.

3.3.2. System Administrator

As a second empirical example, we applied Met-$R_{\max}$ to the System Administrator problem introduced by Guestrin et al. (2003). Once again, we compare Factored-$R_{\max}$, where the structure of the DBN is provided as input, against both Met-$R_{\max}$ and SLF-$R_{\max}$.

We used the bidirectional ring topology instance of the problem with 8 machines. The state is represented by 8 binary factors, representing whether or not each of the machines is running. The probability that a machine is running at time $t + 1$ depends on whether itself and its two neighbors are running at time t, so the in-degree of the DBN representation for this problem is 3. There are 9 possible actions: reboot the i-th machine, or do nothing. If machine i is down at time t and reboot(i) is executed, it will be running at time $t + 1$ with probability 1. If the machine and both its neighbors are running, there is a 0.05 probability that it will fail at the next timestep. Each neighbor that is failing at time t increases the probability of failure at time $t + 1$ by 0.3. For example, if machine i is running but both neighbors are down, there is a 0.65 chance that machine i will be down at the next timestep. Each machine that is running at time t accrues a reward of $+1$, and if the action taken is reboot there is a penalty of -1.

A parameter search was performed for the three algorithms, and the results shown correspond to $m = 30$ for Factored-$R_{\max}$, $m = 30$ and $\epsilon_1 = 0.2$ for SLF-$R_{\max}$, and $m = 50$ for Met-$R_{\max}$.

Figure 2 shows the results. As expected, Factored-$R_{\max}$was the fastest. Similar to the Stock-trading results, Met-$R_{\max}$was able to discover the underlying DBN structure like SLF-$R_{\max}$ but at a much faster rate.

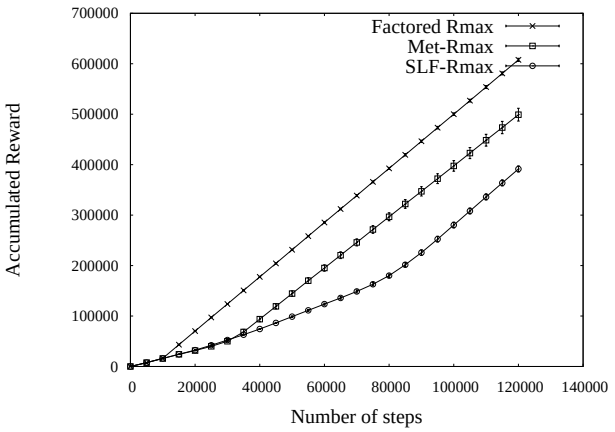


Figure 2. Cumulative reward on each timestep for the three algorithms on the SysAdmin domain: Factored-$R_{\max}$ (structure given), Met-$R_{\max}$ and SLF-$R_{\max}$.

4. Robot Sensory Feature Selection

Consider a robot-navigation domain, where the robot needs to traverse different types of terrains that affect its dynamics. The robot might be equipped with multiple sensors, each providing different types of perceptual inputs, but only some of those inputs are relevant in determining the terrain features that affect movement. The problem becomes a feature-selection problem: if the robot pays attention to all available features, the state space becomes too big. It needs to select which features to pay attention to, and those features need to be the correct ones. The essence of our approach in this section is that we will assign one "meteorologist" to each of the robot's perceptual inputs, or small set of inputs, and let the Adaptive k-Meteorologists Algorithm select features automatically.

4.1. SCRAM-$R_{\max}$

As noted earlier, Leffler et al. (2007) introduced the Relocatable Action Models $R_{\max}$ (RAM-$R_{\max}$) algorithm, which assumes that a classifier provided as input permits clustering different states according to their effects on transition dynamics. In our case, we will simply assume that the maximum number D of potentially relevant sensory features is provided, and we construct as many classifiers as combinations of D sensors there are.

The Adaptive k-Meteorologists Algorithm will then simultaneously learn the correct subset of features and the right predictions necessary to build an accurate action model for each terrain. The resulting algorithm is called *Sensor Choosing for Relocatable Action Models-*$R_{\max}$, or *SCRAM-*$R_{\max}$ for short.

SCRAM-$R_{\max}$ uses the same exploration policy as $R_{\max}$ and the same action-selection policy as RAM-$R_{\max}$. The only difference is that instead of providing a single state clustering function (a classifier) as input, we initialize the algorithm with k possible classifiers, each classifying states according to the input of a different sensor or subsets of sensors. When asked to predict a transition outcome based on a given state s and action a, if any of the k classifiers responds $\perp$, we assume an optimistic outcome and guide exploration towards taking action a. The observed outcome is provided to the k classifiers as an example. After enough experience is gathered, the most accurate classifier according to the Adaptive k-Meteorologists algorithm will be used.

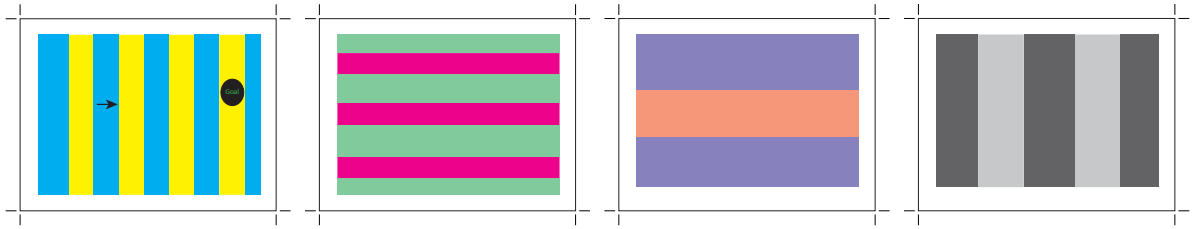
4.2. Artificial-Dynamics Experiment

An experiment using a LEGO Mindstorm NXT® robot was performed to demonstrate the benefits of learning accurate classifiers using SCRAM-$R_{\max}$. The robot was made up of two wheels and a caster, and placed on a flat surface with an overhead camera. Instead of setting up real different terrains, we artificially designated different areas of the surface as having different dynamics. The overhead camera detects the robot's position on the surface, and informs it to a controller that decides how the robot moves based on the action it chose. The robot is also provided with artificial sensors, so that each sensor breaks the surface into terrains a different way.

4.2.1. Experimental Setup

Experiments in this section used four different classifiers, as shown in Figure 3. Classifier 1 (Figure 3(a)) decomposes the surface into the correct terrains, the ones actually matching the artificial dynamics we designed: when the agent is in a state labeled as blue, actions a_0, a_1, a_2, and a_3 result in the robot going left, right, forward and backward, respectively. In a state labeled as yellow, the resulting behaviors are right, left, backward and forward, respectively. The other three classifiers break the surface into terrains that do not match the ones determining the dynamics. The expectation is that the Adaptive k-Meteorologists Algorithm will be able to detect the correct classifier while learning the system's dynamics.

We compared an agent running the SCRAM-$R_{\max}$ algorithm against three RAM-$R_{\max}$ agents. Each of the RAM-$R_{\max}$ agents had a different terrain classifier as input. One used Classifier 1, the correct one; another one used Classifier 2, an incorrect one; and the last one used a classifier that combined the all four of the given classifiers (Figure 4), breaking the surface into lots of terrains. All algorithms were given the goal location and had the parameters $m = 10$ and $\gamma = 1$. The rewards were 1 for reaching the goal, -1 for going out of bounds, and -0.025 for every



(a) Classifier 1 (b) Classifier 2 (c) Classifier 3 (d) Classifier 4.

Figure 3. Several different classifiers of the artificial-dynamics environment given to the SCRAM-$R_{\max}$ and RAM-$R_{\max}$ algorithms. (a)The actual classifier used to determine the robot's dynamics. The arrowhead indicates the start state of the agent and the ellipse indicates the goal location.(b),(c), and (d) show incorrect classifiers that were given as input to the different algorithms.

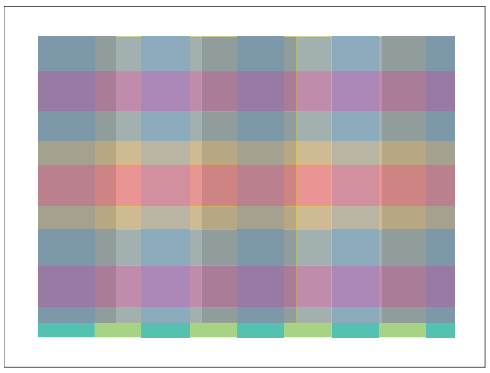


Figure 4. Combination of all features from Figure 3. By taking the product of all features as a different cluster, we would get one cluster for each different color in this figure.

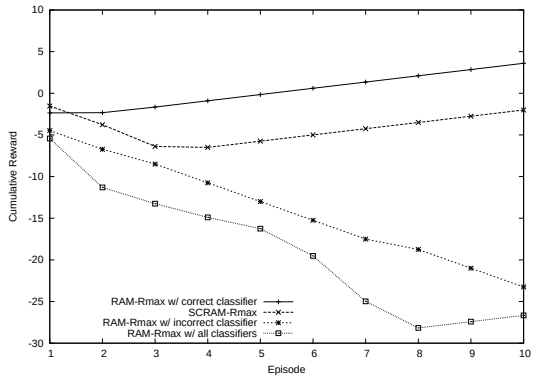


Figure 5. Cumulative Reward in the artificial-dynamics environment.

other state-action pair. Since a RAM-$R_{\max}$ agent with the incorrect model could run indefinitely, the episode would "time-out" if the agent had seen more than m experiences in each type and took more than 50 steps. If a time-out occurred, the episode would terminate and the agent would receive a reward of -1.

4.2.2. Results

Figure 5 shows the various agents' performances. As expected, the RAM-$R_{\max}$ agent with the correct classifier (Classifier 1) performed the best, learning a seemingly optimal policy—receiving 0.75 for each episode. The SCRAM-$R_{\max}$ agent received the second best cumulative reward by converging to the same policy after learning that Classifier 1 was the most accurate clustering function. Next in performance was the RAM-$R_{\max}$ agent with the combined classifier. While its cumulative reward was at one point below -28, the agent eventually learned the optimal policy and began receiving positive rewards. The agent that performed the worst was the RAM-$R_{\max}$ with the incorrect classifier. This agent learned a very noisy dynamics model and was not able to reach the goal. In fact, without a proper dynamics model, the agent was often not able to go out of bounds to end the run, and frequently timed-out.

5. Conclusions

In this paper, we formalize and expand an existing idea for solving the (Adaptive) k-Meteorologists Problems to learn probabilistic concepts in the KWIK model (Li et al., 2008). We give a complete algorithm, analyze its sample-complexity upper bound, and provide a matching lower bound. This algorithm is then used in two new reinforcement-learning algorithms for structure discovery and feature selection while actively exploring an unknown environment.

The first algorithm, Met-$R_{\max}$, is highly efficient at structure learning and improves significantly on the previous state-of-the-art algorithm, SLF-$R_{\max}$. The second algorithm, SCRAM-$R_{\max}$, is able to discover relevant sensory input and thus removes a limiting assumption in its predecessor, RAM-$R_{\max}$. Superiority of both algorithms were demonstrated in either benchmark problems and a robot navigation problem.

Finally, we note that the Adaptive k-Meteorologist Algorithm is very general and can be applied to KWIK-learn

many functions. This includes not only probabilistic concepts, but also real-valued functions. For instance, its capability of finding relevant sensory inputs can be combined with a recently developed approach to KWIK-learning Gaussian distributions (Brunskill et al., 2008), resulting in a new, efficient algorithm in continuous-state problems.

Acknowledgements

We thank Michael Littman and the anonymous reviewers for discussions and suggestions that improved the paper. Carlos Diuk, Lihong Li, and Bethany Leffler were partially funded by DARPA IPTO FA8750-05-2-0249, NSF DGE 0549115, and NSF IIS-0713435, respectively.

References

Abbeel, P., Koller, D., & Ng, A. Y. (2006). Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, *7*, 1743–1788.

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, *11*, 1–94.

Brafman, R. I., & Tennenholtz, M. (2002). R-max—a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, *3*, 213–231.

Brunskill, E., Leffler, B. R., Li, L., Littman, M. L., & Roy, N. (2008). CORL: A continuous-state offset-dynamics reinforcement learner. *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI-08)*.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., & Warmuth, M. K. (1997). How to use expert advice. *Journal of the ACM*, *44*, 427–485.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*, 142–150.

Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, *19*, 399–468.

Kakade, S. (2003). *On the sample complexity of reinforcement learning*. Doctoral dissertation, Gatsby Computational Neuroscience Unit, University College London, UK.

Kearns, M. J., & Koller, D. (1999). Efficient reinforcement learning in factored MDPs. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)* (pp. 740–747).

Kearns, M. J., & Schapire, R. E. (1994). Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, *48*, 464–497.

Kearns, M. J., & Singh, S. P. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, *49*, 209–232.

Leffler, B. R., Littman, M. L., & Edmunds, T. (2007). Efficient reinforcement learning with relocatable action models. *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)* (pp. 572–577).

Li, L. (2009). *A unifying framework for computational reinforcement learning theory*. Doctoral dissertation, Department of Computer Science, Rutgers University, New Brunswick, NJ.

Li, L., Littman, M. L., & Walsh, T. J. (2008). Knows what it knows: A framework for self-aware learning. *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML-08)* (pp. 568–575).

Mannor, S., & Tsitsiklis, J. N. (2004). The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, *5*, 623–648.

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. New York: Wiley-Interscience.

Strehl, A. L. (2007). Model-based reinforcement learning in factored-state MDPs. *Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning* (pp. 103–110).

Strehl, A. L., Diuk, C., & Littman, M. L. (2007). Efficient structure learning in factored-state MDPs. *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07)* (pp. 645–650).

Strehl, A. L., Li, L., & Littman, M. L. (2006a). Incremental model-based learners with formal learning-time guarantees. *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI-06)* (pp. 485–493).

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., & Littman, M. L. (2006b). PAC model-free reinforcement learning. *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML-06)* (pp. 881–888).

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134–1142.

Yamanishi, K. (1992). A learning criterion for stochastic rules. *Machine Learning*, *9*, 165–203.