

Story Similarity Measures for Drama Management with TTD-MDPs

Joshua K. Jones
College of Computing
Georgia Tech
Atlanta, GA, USA 30332
jkj@cc.gatech.edu

Charles L. Isbell
College of Computing
Georgia Tech
Atlanta, GA, USA 30332
isbell@cc.gatech.edu

ABSTRACT

In interactive drama, whether for entertainment or training purposes, there is a need to balance the enforcement of authorial intent with player autonomy. A promising approach to this problem is the incorporation of an intelligent Drama Manager (DM) into the simulated environment. The DM can intervene in the story as it progresses in order to (more or less gently) guide the player in an appropriate direction. Framing drama management as the selection of an optimal probabilistic policy in a Targeted Trajectory Distribution Markov Decision Process (TTD-MDP) describing the simulation has been shown to be an effective technique for drama management with a focus on replayability. One of the challenges of drama management is providing a means for the author to express desired story outcomes. In the case of TTD-MDP-based drama management, this is generally understood to involve defining a *distance measure over trajectories through the story*. While this is a central issue in the practical deployment of TTD-MDP-based DMs, it has not been systematically studied to date. In this paper, we present the results of experiments with distance measures in this context, as well as lessons learned. This paper’s main contribution is presenting empirically-founded practical advice for those wishing to actually deploy drama management based on TTD-MDPs on how best to construct a similarity measure over story trajectories. We also validate the effectiveness of the local probabilistic policy optimization technique used to solve TTD-MDPs in a regular but extremely large synthetic domain.

Categories and Subject Descriptors

I.2.1 [Artificial Intelligence]: Applications and Expert Systems—*games*; G.3 [Probability and Statistics]: *Markov Processes*; J.5 [Arts and Humanities]: *fine arts*

General Terms

Algorithms

Keywords

Drama Management, Interactive Drama, TTD-MDPs

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*
Copyright © 2014, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

1. INTRODUCTION

In computer-based interactive drama, whether it be for the purposes of training via simulation or for entertainment, there is a difficult balance to be struck. On one hand, there is a desire to give the player/trainee maximum autonomy within the simulated environment – this is central to engagement and a sense of agency, important in both training and entertainment settings. However, there is also a need to enforce some constraints based on authorial intent, so that the experience is coherent and productive, in terms of providing a desired set of training experiences or inducing a dramatic arc with good narrative properties. This tension between enforcing constraints and allowing player autonomy has been recognized in the literature [6, 8]. Drama Managers (DMs) are particularly focused on balancing these competing desiderata in interactive drama. The idea is to provide an agent, the DM, with a set of actions (sometimes called *interventions*) that can be taken in the simulated environment with the intent of “nudging” the player to take some desired subsequent actions. These interventions typically take the form of non-player character (NPC) actions or changes in the environment itself (e.g. changes in the weather conditions), and may range from extremely subtle (a gentle hint from a friend) to quite heavy-handed (a rockslide blocks all passages but one).

The goal is both to avoid obvious “choose your own adventure” settings, where the narrative has a fixed progression except at predefined branch points, but also to avoid allowing the story to collapse into meaninglessness – though it is difficult or impossible to completely preclude the latter in the case of a player truly determined to derail the experience. Further, we would like an element of replayability, in the sense that, even if a player takes the same set of actions in two separate sessions with the game/simulation, the outcomes may be different due to some elements of non-determinism. These are precisely the criteria addressed by Targeted Trajectory Distribution Markov Decision Process (TTD-MDP)-based DMs. TTD-MDPs are concerned with *trajectories* through a story (game/simulation), which are generally defined as sequences of major plot points encountered from the beginning to the end of one interactive story. Thus, a story trajectory is an abstract way to describe one instance of a story created by a player interacting with the game or simulation. As shown by Bhat, *et. al.* [2], framing a drama management problem as a TTD-MDP allows for the efficient, local computation of a probabilistic policy over partial story trajectories and DM interventions that, over time and to the extent possible, will cause the distri-

bution of trajectories through the story to match a desired distribution over trajectories as specified by the author.

The difficulty then becomes the specification of this distribution over story trajectories. In any reasonably complex game, the number of conceivable trajectories is easily massive enough to preclude a complete enumeration. The idea proposed in past work on TTD-MDPs [2, 8, 10] is to allow the author to specify exemplars of good (and possibly bad) trajectories. Along with the specification of a parameterized distribution type (e.g. mixture of Gaussians) that can incorporate the exemplars (e.g. as means) and some way to quantify similarity between story trajectories, this is sufficient to induce a distribution over the complete space of trajectories. Past work has focused on making computation of the probabilistic policy efficient, local and optimal, but has largely deferred the question of how to select “good” similarity measures for story trajectories. A good measure must satisfy at least two criteria: first, it should be semantically correct in terms of capturing story similarity in a way that makes sense within the specific domain, and second, it should satisfy constraints that produce natural and reasonable results when used in the context of TTD-MDP probabilistic policy computation. In this work, we begin to directly address the open question of the characteristics that make a good similarity measure for use with TTD-MDPs. We are at this point primarily focused on the second criterion, using a straightforward game with simple rules and interactions but high dimensionality (large state and action spaces) to explore and gain insight about story trajectory similarity measures. In this paper, we present the results of some of these experiments, as well as lessons learned.

This paper’s main contribution is presenting empirically-founded practical advice for those wishing to actually deploy drama management based on TTD-MDPs on how best to construct a similarity measure over story trajectories. We also validate the effectiveness of the local probabilistic policy optimization technique used by TTD-MDPs in a regular but extremely large synthetic domain. Secondly, the paper highlights the need for further research into best practices for the construction of these similarity measures, from both a technical and story semantics perspective, necessary work for TTD-MDPs to be practically deployed that has been largely deferred until now.

2. TTD-MDPs

Here, we will describe the basic formulation of the TTD-MDP problem, and direct the reader interested in the details of the optimal local solution of the probabilistic policy to Bhat, *et. al.* [2]. TTD-MDPs are an extension of Markov Decision Processes (MDPs). An MDP is defined by a tuple $\langle S, A, P, R \rangle$, where S is a set of states, A is a set of actions that can be taken in those states, $P : S \times A \times S \rightarrow [0, 1]$ is a transition function that describes the probability of transitioning from one state to another when a particular action is taken, and $R : S \rightarrow \mathbb{R}$ is a reward function mapping states to reward values. The goal in an MDP, then, is to compute an optimal *policy*, where a policy $\pi : S \rightarrow A$ indicates an action to take in each state. An optimal policy, often denoted π^* , guarantees that an agent that behaves as it dictates will receive the maximum possible long-term expected reward.

The problem with MDPs in the context of interactive drama is that they do not afford maximum replayability: because the policies are deterministic, if the player takes

the same set of actions in multiple games, the outcomes will be exactly the same (barring other sources of nondeterminism in the environment). As a way to address this concern, TTD-MDPs allow for the specification of a distribution over story trajectories. Rather than learning a deterministic policy to maximize long term reward, the goal becomes selecting a probabilistic policy that, over time, is expected to match the specified distribution of trajectories as closely as possible. Formally, a TTD-MDP is also defined, relative to an underlying MDP, by a tuple $\langle T, A, P, P(T) \rangle$. Here, T is the set of all possible (complete and partial) trajectories of states in the underlying MDP given the dynamics of the interactive drama. A complete trajectory means that the game has finished, while a partial trajectory represents a game in progress. The game is assumed to have finite, bounded duration, and thus, T is finite, as are all of its elements. A and P are respectively the set of actions and the transition function from the underlying MDP. $P(T)$ is the desired, target distribution over complete trajectories. The solution to a TTD-MDP is a non-deterministic policy $\pi : T \rightarrow P(A)$, yielding a probability distribution $P(A)$ over actions for each trajectory in T . As is somewhat apparent based on our definition of TTD-MDPs relative to an underlying MDP, any finite-length discrete-time MDP can be converted to a TTD-MDP. We will not detail it here, but previously published work [2] gives a detailed algorithm for the efficient, local and optimal solution of TTD-MDPs for a probabilistic policy. Note that, based on the details of the transition model P , it is not necessarily possible to select a policy for an arbitrary TTD-MDP that will approach the target distribution over time. For example, if the target distribution requires *never* entering some state s' from s , but the transition model has some non-zero probability of entering s' for every action one can take in s , then it is impossible to avoid s' . However, the algorithm of Bhat *et. al.* [2] does guarantee a policy that minimizes the global KL-divergence from the target distribution.

3. DISTANCES IN STORY TRAJECTORY SPACE

The story trajectories constituting T in a TTD-MDP will form a *trajectory tree*, with a root that is the start state of the underlying MDP, and leaves that represent final state(s) terminating trajectories. For example, consider the 3x3 grid world depicted in Figure 1. In this world, there are two possible actions in most states – move up, or move right. Once the agent, which starts in state (0,0), advances two spaces either up or right, it can no longer take the action that would result in it moving off the board. (2,2), in the upper right-hand corner of the board, is the final state. Figure 1 also depicts T for the TTD-MDP that arises from this grid world example. Each path from the root, (0,0) to a leaf node labeled with (2,2) represents a distinct trajectory in T . In this small example, it is not an onerous burden to ask that the author simply manually assign probabilities to each of the six possible trajectories. However, as problem size grows (in terms of the state and action space of the underlying MDP), this very quickly becomes intractable due to exponential growth, as the number of trajectories will be on the order of d^a , where d is the average number of moves needed to reach a terminal state from the start state, and a is the average number of actions allowed in each state.

(0,2)	(1,2)	(2,2)
(0,1)	(1,1)	(2,1)
(0,0)	(1,0)	(2,0)

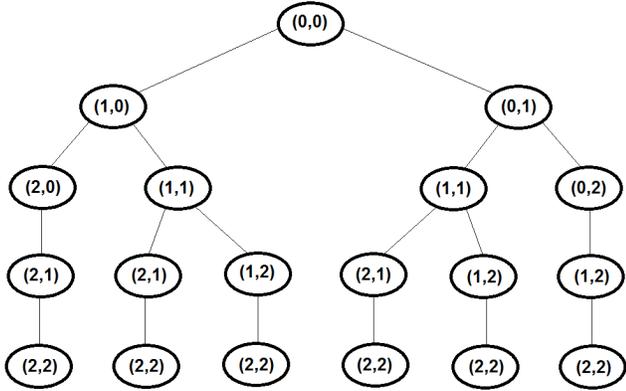


Figure 1: A simple 3x3 grid world, and the trajectory tree depicting T , the set of all possible trajectories through that world, in the associated TTD-MDP.

Thus, for realistic interactive dramas, it will not be possible to ask the author to enumerate a probability distribution over all trajectories in T . Instead, we wish to allow the author to specify some set of *exemplar* trajectories which are considered “good” stories. Then, we build a story trajectory space based on some distance measure over story trajectories. Finally, we can induce a distribution over all possible story trajectories within the space created by the distance measure by using some probability distribution (often a mixture of Gaussians, with one Gaussian centered on each exemplar), to distribute mass over all complete trajectories. This distribution of mass over exemplars in the trajectory space is illustrated in Figure 2. Given this complete distribution over all trajectories in T , we can apply the local, optimal method of Bhat, *et. al.* to solve for a probabilistic policy in an online fashion.

4. RELATED RESEARCH

The topic of interactive drama management, first proposed by Laurel [4], has been widely recognized as important over the past several years, and as such has been an active area of research. Roberts & Isbell [9] give a detailed overview of a broad range of different approaches to drama management, identifying strengths and weaknesses of each. Their analysis is based on a set of explicitly defined desiderata that the authors identify for drama managers, including: speed, coordination between non-player characters, replayability, authorial control, player autonomy, ease of authoring, adaptability to a player’s unique traits, affordance for theoretical analysis, subtlety of the DM’s interventions, and measurability of the quality of experiences produced. TTD-MDPs are particularly focused on replayability, and the method scores well on most of these criteria, with the possible noted exceptions of adaptability to player type and ease of authoring. A mechanism by which to address the issue of authorial burden, suggested by the work

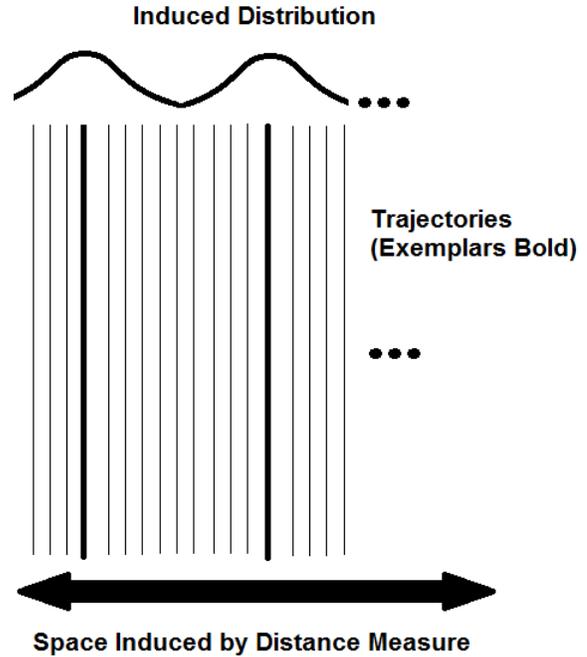


Figure 2: An illustration of the complete distribution over T , the set of all possible trajectories for an environment much larger than that of Figure 1, as induced by exemplars within a story trajectory space, relative to a distance measure and based on a mixture of Gaussians model.

of Si, Marsella & Pynadath [11], is to allow for authoring by the specification of examples of “good” stories. In the work of Si, *et al.*, agent policies are deterministic, and thus the problem of inducing target distributions from the exemplars does not arise. In the case of TTD-MDPs, authoring means specifying a target distribution over story trajectories. As such, we wish to allow the author to specify exemplars that represent “good” complete stories, and then use these to induce a target distribution. For this task, the difficulty then becomes specifying a distance measure over story trajectories that allows a complete distribution to be inferred from the exemplars, a nontrivial task in its own right. Work on applying TTD-MDPs to the problem of designing museum tours [8] does propose a two-valued distance measure that relies on the Levenshtein (edit) distance between the sequences of rooms visited in a compared pair of tours and the relative number of congested rooms visited by the two tours. This work is of interest in that it chooses a semantically founded distance measure in a domain that is potentially of practical interest. However, no study has yet performed a systematic comparison of different distance measure alternatives in the context of TTD-MDPs.

The lineage of TTD-MDPs themselves originates in the formulation of drama management devised by Bates [1]. This formulation was later reconstrued as a search problem by Weyhrauch [12], and then as a reinforcement learning problem by Nelson, *et. al.* [7]. In order to provide for varied experience, TTD-MDPs then shifted from deterministic to probabilistic policies. The use of probabilistic policies has appeared in other research, including Isbell, *et. al.*’s Cobot [3] and Littman’s work on Markov games [5].

5. EXPERIMENTS

5.1 Synthetic Domain

We have experimented with a TTD-MDP-based DM in a synthetic environment called HyperWorld. This environment is high dimensional – states are described by 10 features. An agent, called the “hero”, is controlled by a simulated player, and is expected to move from the origin state (0,0...0) to the goal state (9,9...9). The hero has 10 actions available in most states. Specifically, each of the hero’s actions will increment one of the state features. In states where some of the features are already at ‘9’, the hero will have fewer than 10 viable actions as some state dimensions can no longer be incremented. The DM is able to intervene by making suggestions to the player through an NPC called the “sage”. These hint-based interventions are not guaranteed to succeed. We use a synthetic player in these experiments that, with 98% probability, will move in the suggested dimension (if possible), and with 2% probability, will move in the subsequent dimension (again, if possible). In cases where either the suggested dimension or the subsequent dimension cannot be incremented any further (i.e. is already ‘9’), the remaining dimension will be incremented by the synthetic player with 100% probability. If both the suggested dimension and the subsequent dimension are already at ‘9’, the intervention is not viable and nothing will occur. This domain allows us to obtain readily interpretable results, while incorporating enough complexity to challenge the TTD-MDP policy selection algorithm. In these experiments, we provide the DM with an accurate model of the synthetic player’s probabilistic decision making process, contingent on the various hint-interventions that can be produced by the DM. In practice, if this model is inaccurate, approximation of the target distribution will degrade.

5.2 Experimental Setup

The target trajectory distribution was varied in each of the experimental setups described here, and in each case, 1000 simulations were run to get a sense of the empirically resulting distribution over trajectories that the DM was able to achieve. After the baseline experiment, which uses a uniform distribution over exemplars, placing zero probability mass elsewhere (and which does not require the specification of a distance function over pairs of trajectories), we test using a mixture of Gaussians distribution, with three alternative distance measures. Here, the Gaussian distributions are over *distances* from each exemplar. This means that we would like the DM to find a policy such that the measured distances of actual trajectories from our exemplars fall into a mixture of Gaussians distribution with the specified parameters. In order to distribute the probability mass of the Gaussians properly over trajectories with varying distances from the exemplars, we use a sampled Gaussian kernel. This approach requires that the distance measures range over integer values. For two of the three measures described here, this is quite natural. For the other, we specifically address the issue below. Based on the dynamics of the domain, it may or may not be possible to match the target distribution closely. For example, if our hero ignored advice and behaved randomly, no amount of cajoling from the DM would result in an approximation of the desired distribution. With a more tractable player, we should be able to do better. In these first three experiments, we fit a single Gaussian to our

exemplar with a mean of zero (which will likely always be the case), and a variance of 0.05, giving us a very peaked distribution around the exemplar.

Our distance measures will only be defined for (partial) trajectories with equal length. Thus, when working with a partial trajectory, we will compare against prefixes of the exemplars with the same length. Allowing distance comparisons between prefixes with unequal length is conceivable, but introduces very large complications without obvious benefit in this domain. A key value in computing these distance measures is the absolute difference, summed over dimensions (features), between any two corresponding states in the trajectories. That is, we will compute this difference between the first states in each of the compared trajectories, again between the second states in each, and so on until the last pair of states. If the trajectories contain ‘N’ states, we will in this way generate ‘N’ difference values. For the domain used here, each of these differences will be in the range $[0, num_dimensions * (dimension_max - dimension_min)] = [0, 90]$. We then normalize these distance scores by dividing by two. This normalization of the difference is reasonable because any time a “wrong” dimension is incremented in the actual trajectory, this causes a cumulative difference of two – one for the wrong dimension that got incremented, and one for the “right” dimension that did not get incremented. Dividing out this “double counting” of errors helps to keep a more directly accurate measure of the difference between individual states. Our distance measures, then, vary in the way that they combine these individual normalized differences between corresponding pairs of states in the compared trajectories to produce a single distance value.

- *total-difference* – This measure simply adds each of the individual normalized differences. Notice that the maximum value of this measure is dependent upon the length of the (partial) trajectories compared.
- *average-difference* – This measure can be computed by dividing the total-difference measure by the length of the trajectories being compared. That is, it is the average of the individual normalized differences. Here, we may not naturally generate integral values. Because we always want to prefer less error to more error to the extent possible given the expressivity of the measure, we round division up rather than down. This prevents treating a small error late in a trajectory as equal to no error, for instance.
- *max-difference* – This measure is equal to the maximum individual normalized difference between any corresponding pair of states in the two trajectories.

Note that there is a requirement on the domain that is imposed by this type of distance measure – it must be possible to compute a meaningful distance value between all possible pairs of story states. In many cases, for featurized state representations, this will mean computing a difference value for each feature in the state representation, and then summing these values (possibly with some reweighting) to compute an overall difference between story states. We believe that defining a distance over individual story states will be possible in many domains of practical interest. If it is not possible for a given domain, some other distance

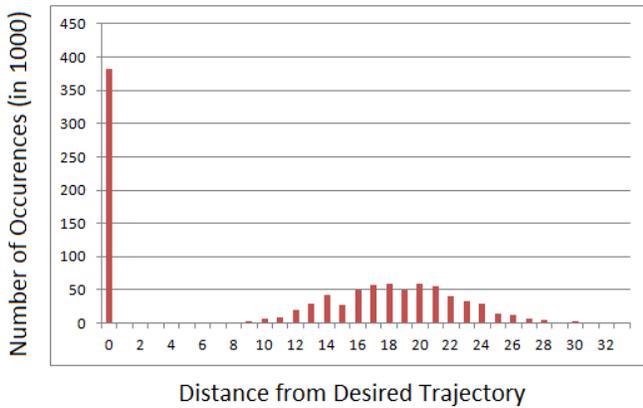


Figure 3: Actual story trajectory distribution achieved by the DM for a single example trajectory, with a uniform target distribution.

between trajectories (such as counting unit error for each non-matching state in a trajectory) must be used. However, notice that such distance measures will be substantially less informative in terms of the relative similarity of story states, and thus the capability of the DM to keep the unfolding trajectory “close” to an exemplar will also be more limited.

5.3 Single Exemplar Results

In the first set of experiments, we specify a single desired path – specifically, a path where each ‘even’ dimension (0,2,4,...,8) is incremented, completely and in order, before the odd dimensions, which are then incremented completely and in order. We first establish a baseline by using a uniform distribution, and then test each of the three distance measures described above in this setting.

As a baseline, we used a uniform distribution over exemplars, which in the case of a single exemplar means that all probability mass is placed directly on the one specified trajectory. In order to graph results, we need to pick a way to measure distances between trajectories, even though no such measure is needed to induce the distribution in this case. Here, we have chosen the ‘max-difference’ measure, as it is the most compact. Results are shown in Figure 3. Clearly, the interventions are successful in causing a large number of actual trajectories followed through the simulation to match the desired trajectory (those with a distance of zero).

Notice also that once a trajectory deviates from the desired path (due to synthetic player nondeterminism), the partial trajectory is in a zero probability mass region of the trajectory space. This means that the DM does not have any preference over ensuing steps in the deviating trajectory – they are all equally bad in some sense. Thus, there is no principled guidance from the DM after any point of random deviation. Notice that, as expected due to the central limit theorem, the distances of these deviating trajectories take a roughly Gaussian distribution. While this behavior does illustrate some benefits of using the DM to influence player behavior, it is also somewhat unsatisfying. We would like the DM to realize that things can always get worse – even when deviation from a desired trajectory occurs, we would still like to keep the ensuing trajectory as close as

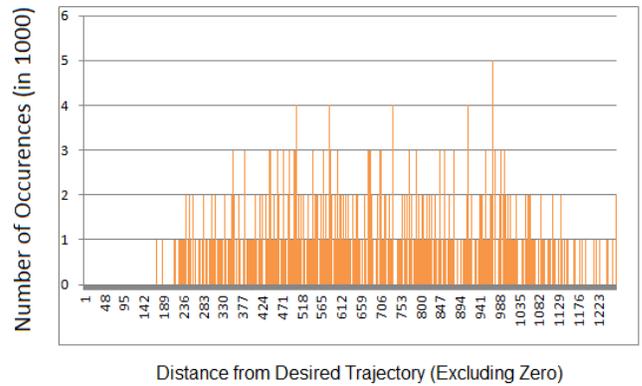


Figure 4: Detail view of the actual story trajectory distribution achieved by the DM for a single example trajectory, with a Gaussian target distribution, using the total-difference measure. We exclude 411 trajectories at a zero distance for readability.

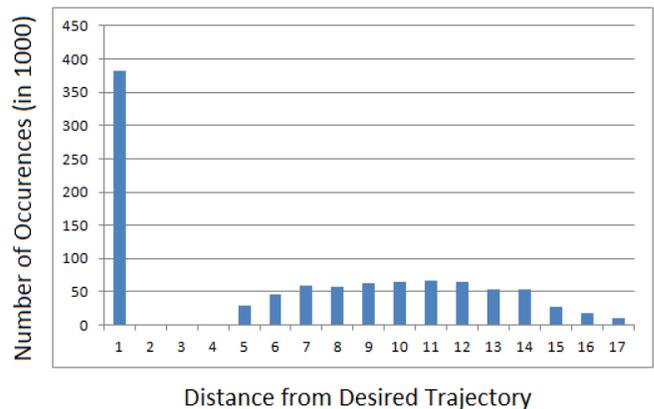


Figure 5: Actual distribution achieved for a single example trajectory, with a Gaussian target distribution, using the average-difference measure.

possible to a nearby target trajectory. In the following three experiments, we kept the same, single target trajectory, but instead used a mixture of Gaussians model (here, a single Gaussian for a single trajectory) along with each of the three distance measures described above. The intent of using a mixture of Gaussians is to provide indication to the DM of preference for trajectories close to the exemplars in the space induced by the distance measure. In each case, we graph the distance from the evens-before-odds exemplar *according to the distance measure used to induce the distribution*. This means that the absolute values of the distances graphed are not directly comparable. However, we are less concerned with the exact distance values achieved (except, perhaps, for counting exact matches with the exemplar, which are comparable across the distance measures), and instead are mostly concerned with the distribution of trajectories.

Results for these three experiments are shown in Figures 4-6. In Figure 4, we omit 411 trajectories that fell at a distance of zero from the exemplar, because including them forces scaling that makes the graph unreadable.



Figure 6: Actual distribution achieved for a single example trajectory, with a Gaussian target distribution, using the max-difference measure.

5.3.1 Analysis

First, a preliminary observation about the reasonableness of the results. When the hero has freedom to do so, it will choose to deviate from the sage’s instructions with 2% probability. If the perfect trajectory is to be followed, the hero will need to choose to follow the sage’s advice until dimensions 0, 2, 4, 6 and 8 are fully incremented, in order. At this point, the remainder of the simulation runs in lockstep, because the hero no longer has an option to deviate from the sage’s advice. For instance, once dimension 8 is fully incremented and the sage advises the hero to advance along dimension 1, there is no opportunity for the hero to instead choose to increment dimension 2, as it is already fully incremented. So, assuming that the sage advises the hero to move according to the exemplar 100% of the time (which in practice it will not, in order to try and match the specified distribution), we would expect the hero to follow the evens-before-odds exemplar trajectory $0.98^{9*5} \approx 0.403$, or about 40.3% of the time. All of the distance measures do yield results in line with this expectation (Figures 4 & 6). However, they vary substantially in the distribution of trajectories that deviate from the exemplar.

Overall, a quick examination of the figures provides clear evidence that max-difference is the superior distance measure for use with a mixture of Gaussians model, based on the resulting distribution over story trajectories. When max-difference is used, we see that as desired, deviating trajectories are kept closer to the specified trajectory, as the DM can now continue to select good interventions even after a deviation occurs. This distance measure yields a reasonable number of zero-distance trajectories, as described above, and also produces a distribution over distances that appears to be an approximately Gaussian distribution, as our target distribution specifies. It *does* appear to be a substantially less-peaked distribution than our target distribution, which is also reasonable given the non-determinism of the hero (simulated player) in this domain.

So, given that we see desired behavior only when max-difference is used, how can we explain the undesirable behavior that we get from total-difference and average-difference? In the case of total-difference, the problem is that a small error is compounded at each step – and in this domain, there

is no way to directly reverse the error. This means that if a random misstep is made, and an odd dimension is incremented early, this error will be carried forward to each new state added to the trajectory (and thus the cumulative error count of total-difference will continue to grow) until the mistakenly incremented odd dimension *would* have been incremented on the desired evens-first trajectory. This means that, in this domain, under total-difference, any small error quickly and necessarily becomes compounded, pushing the distance from the desired trajectory into the low-probability “tail” of the Gaussian. Once this happens, the preference between trajectories essentially becomes uniform, and the DM begins to provide random advice. Thus, we see a substantial spike of trajectories at zero distance (matching the desired trajectory), and then a random smattering of trajectories at a fairly large distance, since a minor deviation will quickly result in essentially a random walk to the goal. In domains with reversible actions, the total-difference measure might be a viable, even preferable option, since it would put pressure on the DM to cause undesirable actions to be reversed, in a way that max-difference would not.

Looking at the results for average-difference, they appear similar to those obtained for the uniform distribution baseline. Though the absolute values of the parameters of the resulting distributions (e.g. the means of the non-zero distance trajectories’ distribution) cannot be compared directly because they are computed according to different measures, in both cases we see a large impulse of the expected magnitude at zero, and roughly Gaussian-looking distributions centered at some distance removed from the impulse containing the deviating trajectories. This is because of information loss when discretizing the raw average values. For example, the difference between a cumulative difference of one (a single error) vs. three (a single error on one step followed by another on the next) becomes impossible to capture with average-difference once the trajectory length grows to four – both will be mapped to a distance of one. While the policy of rounding up preserves differences between zero error and non-zero error at any trajectory length (maintaining the magnitude of the impulse at zero), differences in non-zero error quickly become insubstantial. Thus, the discrete average-difference measure behaves very much like the baseline, offering little to no useful guidance once deviation occurs. This distance measure could somewhat be more useful if not discretized, though with a Gaussian distribution there will in any case be little discrimination between values with differences of small magnitude once the vicinity of the mean is departed. Further, the allocation of probability mass over trajectories becomes much more difficult and involved with a real-valued distance measure, and doing so is left as future work.

5.4 Multiple Exemplar Results

Finally, we wish to see how the system will behave when we specify more than one target trajectory. In this second set of experiments, we provide two desired trajectories – evens first, as described above, and odds first, which is very similar but asks that the player first move the hero along each odd dimension and then along each even dimension. We once again use the mixture of Gaussians distribution and the distance measure described above. Results are depicted in Figures 7-9.

The careful reader will notice some asymmetries in the

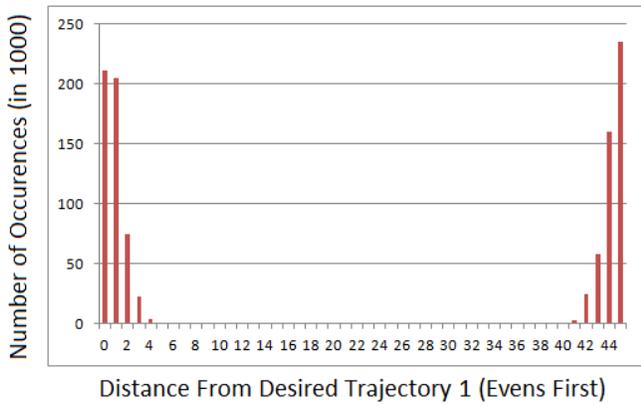


Figure 7: Actual distribution achieved for two example trajectories, using a Gaussian target distribution. Distance shown is from the first of the two target trajectories.

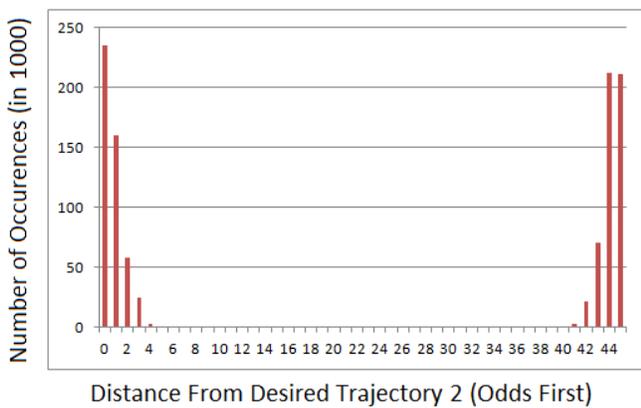


Figure 8: Actual distribution achieved for two example trajectories, using a Gaussian target distribution. Distance shown is from the second of the two target trajectories.

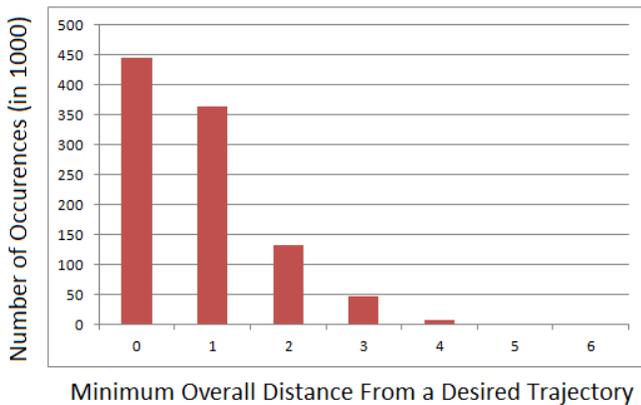


Figure 9: Actual distribution achieved for two example trajectories, using a Gaussian target distribution. Distance shown is from the closest specified trajectory.

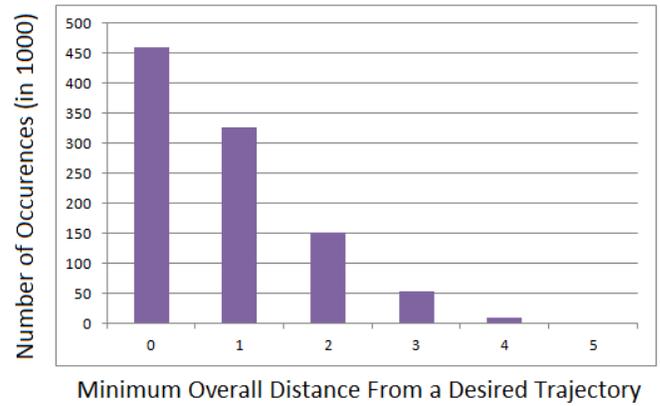


Figure 10: Actual distribution achieved for four example trajectories, using a Gaussian target distribution. Distance shown is from the closest example trajectory.

distributions around each of the two specified trajectories. Apart from the effects of nondeterminism, this is due to different boundary conditions affecting the two paths – for instance, there is no subsequent dimension to be incremented when DM guidance indicates that the final dimension (9) should be selected. These effects create some asymmetries in the evens first vs. odds first trajectories. Overall, we can see that the DM is quite successful both in targeting the desired trajectories, and in approximating the specified mixture of Gaussians distribution around the two specified trajectories, even in this large, high dimensional and nondeterministic domain.

As an additional experiment to further examine the effects of adding more exemplars, we also tried adding two more trajectories to those of the runs depicted in Figures 7-9. The first of the two new trajectories first increments every even dimension, in order, by one, and then repeats until each even dimension has been fully incremented (in this case, each even dimension is first incremented to one, then to two, etc). Then the process is repeated for the odd dimensions. The second of the two new trajectories is the same, except with odd dimensions first, followed by even dimensions. Results of this run are depicted in Figure 10.

Once again, we see that the DM is successful in achieving a good distribution of trajectory lengths around the exemplars. In fact, adding more desirable trajectories allows the DM to do slightly better – since early deviations by the hero can sometimes be compensated for by “switching” the intended trajectory to one for which the error is an appropriate move.

Additionally, notice that this domain is quite large: the complete game tree has a depth of 90 moves, and a branching factor of 10, with a size on the order of 10^{90} . Yet the local optimization procedure is able to produce decisions according to an optimal probabilistic policy using time and space that is quite reasonable for a modest personal computer. For example, the 1000 trials in the case of four desired trajectories ran in a wall time of very close to 15 minutes, or $900/1000 = .9$ minutes/trial. Given that each trial consists of 90 moves, this means that the DM is taking less than half a second to compute the appropriate intervention at each step (accounting for the overhead of other needed com-

putations at each game step). This is quite reasonable for practical deployment on machines with limited power.

6. LESSONS LEARNED

Based on these experiments, it is clear that there is a substantial interaction between (1) the selected distance measure over story trajectories, (2) the distribution type selected, (3) how the mass of the selected distribution is allocated across discrete trajectories, and (4) the mechanics of the domain.

Part of the contribution of this work is identifying these interacting dimensions of variation that would be interesting to explore systematically in future work (only (1) is empirically addressed here), and which should be explicitly considered when designing a new trajectory distance measure. By analyzing these interactions within the context of our preliminary experiments in the HyperWorld domain, we can begin to provide some guidance about the characteristics needed for distance measures to provide desired behavior in various kinds of domains.

First, we can say that, for story domains with transitions that cannot be directly reversed by player actions – which is likely to be a common class of domains – it is very important that the distance measure selected does not compound earlier errors (e.g. by summing carried forward error over states in a trajectory) as the trajectory length grows. Distance measures that do compound errors with trajectory length will exhibit poor behavior when used with any distribution that places mass on trajectories that asymptotically approaches zero as distance from an exemplar increases. Because this characteristic will hold for any reasonable probability distribution chosen, the use of an error-compounding trajectory distance measure is unlikely to be a good choice for any story domain with non-reversible state transitions. However, though it is not empirically studied here, it does appear reasonable that such distance measures may have value in story domains where state transitions can be reversed, essentially “undoing” deviations. There may be additional complications in handling such domains, such as defining distance measures that allow trajectories with unequal lengths to be compared, and we leave the matter for future research, but do wish to note it here.

Next, we can conclude that averaging errors across trajectory length is unlikely to be a viable option in most situations. The difficulties that arise when averaging, due to the decreasing significance of small variations in error values as trajectory lengths grow, may be ameliorated if alternatives for dimension (3) can be developed. However, at this time it is not clear that any substantial benefit would be achieved by doing so, and given that viable alternatives exist, taking on the additional complexity does not immediately appear worthwhile.

Finally, we can determine that, for domains like HyperWorld, the max-difference distance measure is both a clear winner relative to the other measures tested, and in fact appears to yield very good behavior from an objective standpoint. More generally, it appears that, at least for domains with non-reversible transitions, and without introducing substantial complications to allow for the comparison of trajectories with unequal lengths, distance measures like max-difference that do not compound errors as trajectories grow are the best choice.

7. CONCLUSIONS

In this paper, we have described the first set of experiments exploring the use of different kinds of distance measures over story trajectories for Drama Management using TTD-MDPs. Further, we have demonstrated empirically the effectiveness of the local optimization technique of Bhat *et. al.* [2] in a substantially larger domain than used in any previously reported experiment. Based on these initial experiments with distance measures, we have presented a set of recommendations for the selection of distance measures based on domain characteristics and other design choices. Further, we identify these domain characteristics and design choices as dimensions of variation for future experiments. At present, we have identified only one domain characteristic that appears particularly significant to the choice of trajectory distance measure: the reversibility of story state transitions. Another useful direction for future work is to identify more classes of domain mechanics that have a substantial impact on the choice of trajectory distance measure.

8. REFERENCES

- [1] J. Bates. Virtual reality, art and entertainment. *Presence*, 1(1):133–138, 1992.
- [2] S. Bhat, D. L. Roberts, M. J. Nelson, C. L. Isbell, and M. Mateas. A globally optimal algorithm for TTD-MDPs. In *AAMAS-07*, page 199. ACM, 2007.
- [3] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone. A social reinforcement learning agent. In *AGENTS-01*, pages 377–384. ACM, 2001.
- [4] B. K. Laurel. *Toward the design of a computer-based interactive fantasy system*. PhD thesis, Ohio State University, 1986.
- [5] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML-94*, volume 94, pages 157–163, 1994.
- [6] B. Magerko and J. E. Laird. Mediating the tension between plot and interaction. In *AAAI Workshop Series: Challenges in Game AI*, pages 108–112, 2004.
- [7] M. J. Nelson, D. L. Roberts, C. L. Isbell Jr, and M. Mateas. Reinforcement learning for declarative optimization-based drama management. In *AAMAS-06*, pages 775–782. ACM, 2006.
- [8] D. L. Roberts, A. S. Cantino, and C. L. Isbell Jr. Player autonomy versus designer intent: A case study of interactive tour guides. In *AIIDE-07*, pages 95–97, 2007.
- [9] D. L. Roberts and C. L. Isbell. A survey and qualitative analysis of recent advances in drama management. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, 4(2):61–75, 2008.
- [10] D. L. Roberts, M. J. Nelson, C. L. Isbell, M. Mateas, and M. L. Littman. Targeting specific distributions of trajectories in MDPs. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1213. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [11] M. Si, S. C. Marsella, and D. V. Pynadath. Thespian: Using multi-agent fitting to craft interactive drama. In *AAMAS-05*, pages 21–28. ACM, 2005.
- [12] P. Weyhrauch. *Guiding interactive drama*. PhD thesis, Carnegie Mellon University, 1997.