Using Divergence-to-Go to Explore 2D Mazes

Arslan Siddiqui

Abstract: Reinforcement learning is a complex task that is computationally expensive and it becomes even more challenging in complex continuous environments. Traditional stochastic methods like epsilon-greedy are unable to explore the large state-action space in a uniform manner. In this paper, we use divergence-to-go as a method for guided exploration of the state-action space. It is based on an information-theoretic approach to perform a guided search and measures the cumulative divergence associated with each state-action pair. The performance of divergence to go is measured against random search and we demonstrate the divergence-to-go outperforms it by a significant margin.

***Keywords* - Reinforcement learning, divergence-to-go, information-theoretic learning**

1. INTRODUCTION

Reinforcement Learning (RL) is used to make an agent learn how to interact with an environment such that a desirable goal is achieved. The main objective is to optimize the limited number of actions of an agent with respect to a reward signal that is present in the environment. RL can be used to find solutions for Markov Decision Processes (MDPs). MDPs can be used to model certain stochastic control problems. Generally, MDPs with only a finite (and usually small) number of actions are considered. A popular approach taken today is to select a random action with a small probability which can be decreased over time. However, in practical situations, with the enormity of the state-action space, a discretization of the action space is not useful unless it is done very finely. In such cases, a directed approach helps, such that the space is explored as efficiently and evenly as possible.

An autonomous system need to know the uncertainty present in the environment currently. When there is complete knowledge of the environment, no exploration is required.

Quantifying uncertainty is not a trivial task as it is a local attribute and varies from one state to another.

Information-theoretic descriptors can be used to ascertain uncertainty and one such measure that can be employed in entropy. Entropy has an inherent issue that it is unable to differentiate between uncertainty due to lack of knowledge and uncertainty due to stochasticity in the world. To overcome this problem, we can employ divergence as measure to quantify the agent's current knowledge and knowledge that it will possess in a future state.

2. BACKGROUND INFORMATION

In model-based RL, the agents estimates a model of state transition probabilities by undertaking an action and observing the changes in the environment. Each time an agent goes to state *y* from state x by taking action *a,* we can update our belief distribution P(y|x,a) and we can compute the divergence between two successive states as $D[P_{new}(y|x,a)||P_{old}(y|x,a)]$. Divergence-to-go is defined as discounted sum of divergences for the transitions over time.

$$dtg(x,a) = E[\sum_{t=0}^{\infty} \gamma^t D(x_t)]$$

As time increases, dtg values will decrease and tend to zero. The agent chooses an action that maximises divergence which leads to efficient exploration of the space.

A. *Markov Decision Process*

In Reinforcement Learning, an agent decides the best action to select based on its current state. The sequential nature of this problem can be represented as a Markov Decision Process. It consists of a possible set of states **S**, a possible set of actions **A**, a real valued reward function **R(s,a)** and the transition model **P(s`|s,a)**.

A state is a token representation of the current environment that the agent is in. The transition model gives an action's effect in any given state. Particularly, P(s'|s,a) defines a transition from state s to s’ by taking the action a. A(s) defines the number of actions that an agent can take in state s. Reward, given by, R(s), specifies the reward for being in state s and R(s’,a,s) defines the reward for choosing action a and transitioning from state s to s`.

Typically, only a small fraction of states possess a non-zero reward, there has to be mechanism for assigning rewards to states that indirectly led to the state with the reward. To facilitate this, we assign a value to each state equal to the average of the sum of the rewards after visiting that state:

$$V^{\pi}(x) = E[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | X_o = x]$$

The value of each state depends on the policy
π : S -> A, which assigns an action to each state in X.

It can become computationally intractable to calculate value functions of MDPs because we have to calculate the expected value over all future iterations. We can use tools of Dynamic Programming to solve this problem.

B. Information Theoretic Learning

Information Theoretic Learning uses descriptors from information theory like entropy and divergence, which can be estimated directly from the data to substitute the conventional statistical descriptors of variance and covariance. It finds uses in adaptation of linear and non-linear filters and also in unsupervised and supervised learning applications. The foundation of ITL is built upon Renyi’s (differential) alpha-order entropy, which is given by:

$$H_{\alpha}(X) = \frac{1}{1-\alpha} log \int p^{\alpha}(x) dx$$

Here, p is a probability distribution function belonging to L**α**. The benefit of using Renyi’s entropy is that it can be estimated pairwise from samples, parametrically. For alpha = 2, the equation becomes Renyi’s quadratic entropy given by :

$$H_2(X) = -log \int p^2(x) dx$$

Kernel density estimation (Parzen-Rosenblatt window estimation) can be used to estimate the PDF of a random variable in a non-parametric manner. The most commonly used kernel function is the gaussian which is represented as:

$$G_{\sigma} = \frac{1}{\sqrt{2\pi}\sigma} exp(\frac{\|x-y\|^2}{2\sigma^2})$$

When we plug in the gaussian kernel in the kernel density estimation, it becomes:

$$p_X(x) = \frac{1}{N\sigma} \sum_{i=1}^{N} G_{\sigma}(\frac{x - x_i}{\sigma})$$

Using this we can estimate the probability density present in Reni’s quadratic entropy.

$$H_2(X) = -log(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_j - x_i))$$

This allows is to compute the entropy using pairwise samples. The argument of the log in the previous equation is called the information potential is represented by *V.* The analogy is drawn from physics in the sense that the pairwise distance between the samples is always positive and is inversely proportional to the squared distance between them. We can consider the manifestation of a particle field for each particle in the width of the field defined by the Gaussian Kernel. This represents the potential energy which particles abide to in the real world but in our case, the potential energy abides to law determined by the Gaussian Kernel.

Another term that is of importance to us is the cross information potential which can be calculated as:

$$H_2(X) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_f(i) - x_g(j))^2$$

Where, f(x) and g(x) are distribution of the same random variable x. It measures the marginal forces that each each datapoint in either distribution exerts onto each other.

3. TRANSITION MODEL

To estimate the transition model for a set of points, we first have to draw from intuition that points close in space have similar transition probability density functions. We can use a similarity measure that gives a weight to each data point that we use to estimate the distribution. We can use the radial basis function to fulfill this need. We also have to make the assumption that transition distributions, through the state space, vary smoothly about the initial point of transition. With this in mind, we can estimate the transition pdf (x -> y) by using the following equation:

$$p(y|x) = \sum_{i} s(x_i, x) k_\sigma(y_i - x_i, y - x)$$

We normalise the similarity function so that the pdf satisfies the property of integrating to 1.

Given two state reward transition distributions p and q, the Euclidean and Cauchy schwarz divergences can be computed by observing that both probability distributions can be written in terms information potential and cross information potential terms.

$$D_{euc} = V_p + V_q - 2V_c$$
$$D_{cs} = log \frac{V_p V_q}{V_c^2}$$

The information potential terms can be estimated as:

$$V_p = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} s_p(x_i, x) s_p(x_j, x) G_{\sigma\sqrt{2}}(q(x_i) - p(x_i), q(x_j) - p(x_j))$$
$$V_q = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} s_q(x_i, x) s_q(x_j, x) G_{\sigma\sqrt{2}}(q(x_i) - p(x_i), q(x_j) - p(x_j))$$
$$V_c = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} s_p(x_i, x) s_q(x_j, x) G_{\sigma\sqrt{2}}(q(x_i) - p(x_i), q(x_j) - p(x_j))$$

We can calculate the values using kernel matrices and similarity vectors to calculate the values as

$$V = s^T K s$$

4. DIVERGENCE-TO-GO

Divergence to go is a dynamic programming framework and we have to solve with methods analogous to those in reinforcement learning. We map state action pairs to divergence to go values in the case of continuous action spaces. We choose Kernel Temporal Difference algorithm and compute divergence-to go using the product kernel over state-action pair x`,

$$\delta_t = D + \gamma \max_a (dtg_{t+1}(\overline{x}`)) - dtg_t(\overline{x})$$

$$dtg_t(\overline{x}) = D_o + \alpha \sum_{j=1}^{t} \delta_j k(\overline{x}, \overline{x}_j)$$

Where D is the divergence measure, alpha is the learning rate and Do is the initial value of dtg.

We compute divergence according to the equations specified in the previous section, and for each divergence computation, we use N nearest neighbours to the current state-action pair as samples. We arrange them in ascending order and split them equally and then compute the divergence between the two sets of samples. This speeds up computation as there is only a limited amount of samples from which to calculate divergence. Gamma is the discount factor is used to specify the number of steps into the future over which we're computing the sum of divergences.

The initial value of DTG, Do, controls the rate of exploration in the sense that a small value of Do results in the dtg policy exploring small areas of state space until the estimated divergence to go for the actions in that area are below Do. While a large value of Do results in a higher rate of exploration before focusing on areas of high divergence. A principle to assign Do can be generalised to:

$$D_o = \frac{D_{max}}{1 - \gamma}$$

As training time -> infinity, dtg ->0, thus it's not suitable for learning a policy. We have to take into account the state o the current model as well.

The model state can be quantified by either by a global approach, i.e. how the transition model acts for each and every state and action or by a local approach which summarises how the model acts at a certain state or for a state action pair.

Since the transition model does not have easily accessible parameters which can be used to summarize the model in addition to it being really difficult to summarize the global output of the model, we use the local approach. This allows us to use ITL quantities to summarize transitions.

We can represent a transition using the transition PDF’s mean and entropy.

We thus introduce the mean kernel and and entropy kernel, both being gaussian kernels. Then the positive definite kernel becomes, k((x,a,h,m),(x‘a’,m’,h,’)) where m and h stand for mean and entropy, the dtg update the becomes:

$$dtg(x_n, a, m, h) = \eta \sum_{j}^{n} \delta_j k((x, a, m, h), (x_j, a_j, m_j, h_j))$$

$$\delta_i = D_i + \gamma \underset{a}{max}\, dtg(x_{n+1}, a_{n+1}, m_{n+1}, h_{n+1}) - dtg(x_n, a_n)$$

5. EXPERIMENTAL SETUP

To perform the experiment we have designed 4 different mazes that the agent has to navigate through. Each maze is of size 20 x 20 and the blue boundary represents walls through which an agent will not be able to pass. Each discrete state in the action space can be expressed in cartesian coordinates as a tuple varying from 0 to 20.

For quantifying the performance of divergence to go as a tool for maze exploration. At the start, the agent is present at top-left most state. Then at each successive iteration, the action which results in the highest divergence is chosen and the agent transitions to the state. We consider the maze to be fully explored when 95% of the state space has been explored. The kernel size and transition model kernel size (along x and y axes) is varied according to the following values:

σ_s = [0.25,0.5,0.75]

σ_x = [0.1,0.25,0.5,0.75,1,1.25,1.5]

σ_y = [0.1,0.25,0.5,0.75,1,1.25,1.5]

To quantify the performance of divergence to go we use two main metrics, the number of steps taken to explore 95% of the state space and the rate of exploration which can be defined as the number of steps taken per unique state visited. The ideal value of rate of exploration should be 1 and cannot be less than it. Rate of exploration takes into account the explorable volume of the maze and puts the the number steps taken for exploration in the proper context. Maze 1 has an explorable volume of 272 states, maze 2 has an explorable volume of 256 states, maze 3 has 238 explorable states and maze 4 has 248 explorable states. We perform the training by choosing α = 0.01 and γ = 2

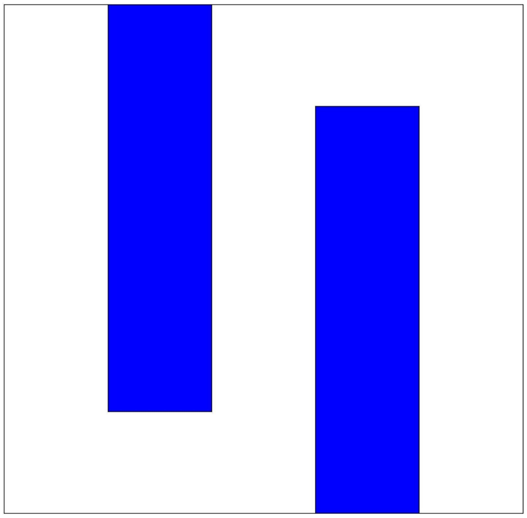

Fig 1. 1st maze

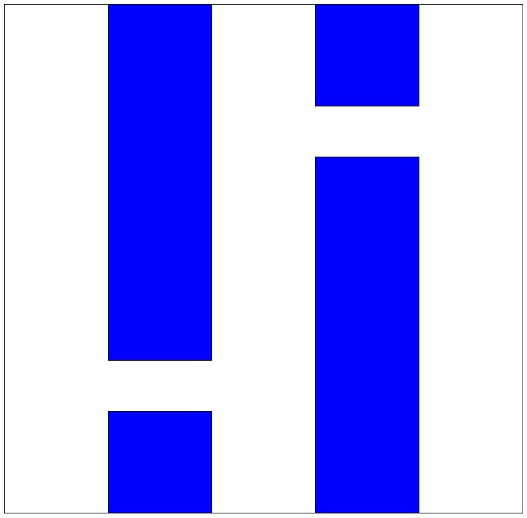

Fig 2. 2nd maze

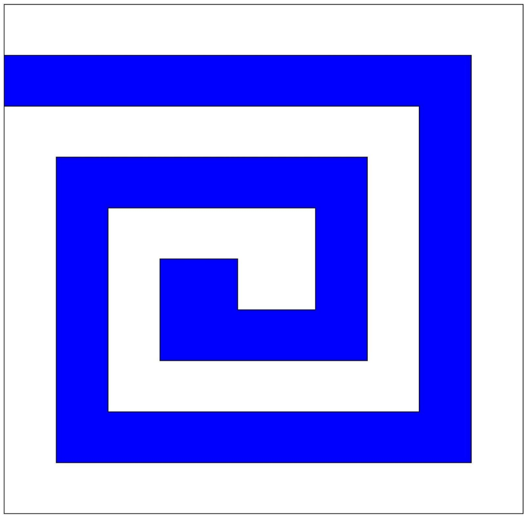

Fig 3. 3rd Maze

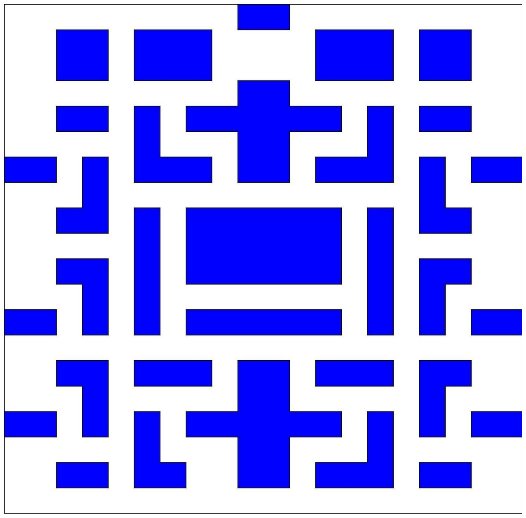

Fig 4. 4th maze

To observe the performance of divergence to go,we run 10 monte carlo trials for each combination of similarity and transition kernel size. This makes it a total of 1470 trials per maze. And to observe the performance of random policy, we run 100 monte carlo simulations for each maze and observe the average number of iterations as well as rate of exploration. The action space available to the agent is up,right,down and left.

6. RESULTS

	Average Steps	Min	Max	Rate
Maze 1	7895.77	1995	19085	30.60
Maze 2	9252.09	1902	30966	38.07
Maze 3	22797.71	3070	84845	100.87
Maze 4	7355.58	3328	19938	31.16

Table 13

Table 13 shows the values for average steps, minimum steps, maximum steps taken to explore 95% of the maze and also the rate of exploration for the 4 mazes by random search..

The results of the experiments have been noted in tables 1-13. We will analyse each maze and compare the results of divergence to go with random search.

A. Maze 1

The first maze is a very simple example and there is freedom for the agent to move freely. For divergence to go, the minimum value of rate of exploration = 2.81, i.e. 2.81 steps per state and the average number of steps taken = 725.4. The value is observed at similarity kernel size = 0.75, σ_x = 1, σ_y = 0.5. The minimum value for kernel size = 0.25 is 3.14 and is observed at σ_x = 1.25, σ_y = 0.25, and the minimum value of rate of exploration for kernel size = 0.5 is 3.13 and is observed at σ_x = 1.5, σ_y = 0.75.

In comparison, we observe that radom policy takes an average of 7895.77 steps to achieve an equivalent degree of exploration and the rate of exploration comes out to 30.60.

This means that divergence to go is approximately 11 times faster than random search for the first maze.

B. Maze 2

The second maze is a bit more constrained in the sense that there are 2 tiny tunnels which have to be navigated through to access unexplored space.

For divergence to go, the minimum value of rate of exploration = 3.12, i.e. and the average number of steps taken to explore the maze = 759.9. The value is observed at similarity kernel size = 0.5, σ_x = 0.25, σ_y = 0.75. The minimum rate observed for kernel size = 0.25 is 3.28 and is observed at σ_x = 1.25, σ_y = 1, and the minimum value of rate of exploration for kernel size = 0.75 is 3.41 and is observed at σ_x = 0.5, σ_y = 1.25.

In comparison, we observe that radom policy takes an average of 9252.09 to achieve an equivalent degree of exploration and the rate of exploration comes out to 38.07.

This means that divergence to go is approximately 12 times faster than random search for the second maze

C. Maze 3

The third maze is complex and requires a highly directed path for efficient exploration.

For divergence to go, the minimum value of rate of exploration = 4.6 and the average number of steps taken = 1040.5. The value is observed at similarity kernel size = 0.75, σ_x = 0.75, σ_y = 0.75. The minimum value for kernel size = 0.25 is 4.92

Width 0.25	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	1049.5	4.06	1102	4.27	1160.3	4.49	1072.9	4.15	1254.2	4.86	899	3.48	1032	4.0
0.25	1028.1	3.98	1067.4	4.13	990.1	3.83	940.6	3.64	1065.9	4.13	1159.6	4.49	829.1	3.21
0.5	1148	4.44	949.5	3.6	819.4	3.17	846.1	3.27	1090.9	4.22	1084.6	4.2	933.3	3.61
0.75	1208.2	4.68	874.7	3.39	898	3.48	1071.4	4.15	980.1	3.79	1122.7	4.35	833.8	3.23
1	845.7	3.277	1042.5	4.04	1122	4.34	1018.9	3.94	969.9	3.75	1019.2	3.95	888.3	3.44
1.25	984.3	3.81	**811.1***	**3.14***	1122.2	4.34	950.4	3.68	824.2	3.19	1027.3	3.98	1119.7	4.33
1.5	1001.9	3.88	986.2	3.82	931.8	3.61	1114.8	4.32	1149.8	4.45	934	3.62	888.4	3.44

Table 1

Width 0.50	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	958.3	3.71	1058	4.10	1174	4.55	848.5	3.28	884.9	3.42	1169.5	4.53	1183.3	4.58
0.25	850.3	3.29	1046.5	4.05	967.6	3.75	1110.8	4.30	1104.7	4.281	926.4	3.59	883.9	3.42
0.5	959.4	3.71	965.9	3.74	1172.8	4.54	970.1	3.76	981.9	3.80	1180	4.57	1135.9	4.40
0.75	1180.3	4.57	1209.4	4.68	893.9	3.46	971.2	3.76	895.6	3.47	1015.2	3.93	993.7	3.85
1	886.2	3.43	987.5	3.82	1052.9	4.07	1030	3.99	862.8	3.44	860.1	3.33	1023.0	3.96
1.25	1069.8	4.14	1528	5.92	954.2	3.69	987.6	3.82	1082.4	4.19	950.3	3.68	1056.9	4.09
1.5	1041.2	4.03	1033.1	4.00	1071.2	4.15	**808.2***	**3.13***	875.4	3.39	981.8	3.80	848.7	3.28

Table 2

Width 0.75	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	1365.3	5.29	1107.5	4.29	1198.3	4.64	1017.9	3.94	986	3.82	844.6	3.27	941	3.64
0.25	1141.9	4.42	989.5	3.48	1096.5	4.25	898	3.48	1041.6	4.03	1151.2	4.46	959.8	3.72
0.5	1045.5	4.05	1171.8	4.54	1037.3	4.02	870.5	3.37	788.5	3.05	1187	4.6	917.6	3.55
0.75	1035.3	4.01	1029.7	3.99	827	3.2	1015.4	3.93	851.4	3.3	871	3.37	1046	4.05
1	1012.7	3.92	1040.5	4.03	**725.4***	**2.81***	882.8	3.42	873.5	3.38	1104.6	4.28	925.9	3.58
1.25	1110.1	4.30	1164.1	4.51	1176.0	4.55	852.6	3.30	935.7	3.62	938.1	3.63	866.1	3.35
1.5	974.9	3.77	891.7	3.45	960	3.72	1019.3	3.95	940.7	3.64	892.9	3.46	858.7	3.32

Table 3

Tables 1, 2 and 3 show the values of average steps and rate of exploration for the first maze, for 3 different size of similarity kernels as well as different kernel sizes of transition pdf estimation. The top row represents variation along the y-axis and the first column shows the variation along the x-axis. The similarity kernel size is shown at the top left corner. The underlined values represent the combination of kernel size for which divergence computation was very tiny and the values with asterisk show minimum observed values of average steps and rate of exploration.

Width 0.25	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	829.1	3.41	1042.1	4.28	1079.8	4.44	1090	4.48	958.3	3.94	975.6	4.01	1073	4.41
0.25	953.4	3.92	865.3	3.56	862.5	3.54	995	4.09	983.6	4.04	955.6	3.93	1011.1	4.16
0.5	826.2	3.4	1333.1	5.48	821.7	3.38	887.3	3.65	951.6	3.91	873.9	3.59	926.9	3.81
0.75	1111.4	4.57	1029.3	4.23	1046.7	4.30	1000.8	4.11	1078.7	4.43	883.2	3.63	964.4	3.96
1	873.5	3.59	941.7	3.87	1067.4	4.39	973.5	4.00	957.2	3.93	944.1	3.88	828.3	3.4
1.25	1162.2	4.78	1047.9	4.31	926.6	3.81	950.6	3.91	**798***	**3.28***	929.9	3.82	911.3	3.75
1.5	950	3.90	1079.3	4.44	1017.5	4.18	921.3	3.79	909.7	3.74	1012.3	4.16	893.1	3.67

Table 4

Width 0.50	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	1121.4	4.61	1008.5	4.15	828.7	3.41	1140.4	4.69	858.1	3.53	1066	4.38	865.1	3.56
0.25	1205.5	4.96	991.8	4.08	1117.2	4.59	**759.9***	**3.12***	1043.2	4.29	1009.1	4.15	1084.2	4.46
0.5	1009.7	4.15	1054.1	4.33	893.8	3.67	957.6	3.94	1064.3	4.37	1094.1	4.5	1121.7	4.61
0.75	919.3	3.78	892.9	3.67	874.7	3.59	970.4	3.99	879.8	3.62	910.3	3.74	1060.1	4.36
1	1006.5	4.14	984.6	4.05	866.4	3.56	1146.1	4.71	947.9	3.90	973.4	4.00	905.3	3.72
1.25	905.6	3.72	1041.5	4.28	1080.2	4.44	1074.6	4.42	1084	4.46	931.4	3.83	1015.1	4.17
1.5	947.3	3.89	825.1	3.39	1230.3	5.06	955.7	3.93	880.9	3.62	945.8	3.89	961.6	3.95

Table 5

Width 0.75	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	957.4	3.93	1200.3	4.93	896	3.68	991.8	4.08	965.4	3.97	1033.8	4.25	1056.8	4.34
0.25	1124.6	4.62	897.3	3.69	1005.9	4.13	1056.5	4.34	869.9	3.57	1018.5	4.19	1047.5	4.31
0.5	906.1	3.72	999.2	4.11	1004.5	4.14	1153.4	4.74	1026.4	4.22	**829***	**3.41***	876	3.60
0.75	996.8	4.10	1037.6	4.26	1094.2	4.50	1050.9	4.32	944	3.88	1070	4.40	894.8	3.68
1	111.3	4.58	1057.7	4.35	1046.6	4.38	926.8	3.81	981.1	4.03	832.6	3.42	976.7	4.01
1.25	1109.5	4.56	1080.5	4.44	1071.5	4.40	953.7	3.92	1199.1	4.93	1184.6	4.87	925	3.80
1.5	1033.6	4.25	1038.1	4.27	957.2	3.93	972.5	4.00	889.5	3.66	1044.3	4.29	877.4	3.61

Table 6

Tables 4, 5 and 6 show the values of average steps and rate of exploration for the second maze maze, for 3 different size of similarity kernels as well as different kernel sizes of transition pdf estimation. The top row represents variation along the y-axis and the first column shows the variation along the x-axis. The similarity kernel size is shown at the top left corner. The underlined values represent the combination of kernel size for which divergence computation was very tiny and the values with asterisk show minimum observed values of average steps and rate of exploration.

Width 0.25	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	1187.5	5.25	2551.9	11.29	2725.1	12.05	2734.5	12.09	2650.9	11.72	2395.4	10.59	2936.4	12.99
0.25	2832.7	12.53	1568.2	6.938	1852.3	8.19	2695.8	11.92	2664.3	11.7	2426.2	10.75	2736.7	12.10
0.5	2673.4	11.82	1923.0	8.50	1160.8	5.13	1291.4	5.71	1438.9	6.36	1994.1	8.28	2584.1	11.43
0.75	2971.5	13.14	2826.9	12.5	**1112***	**4.92***	1570.7	6.95	1189.2	5.26	1516.8	6.71	1686.8	7.46
1	2786.2	12.32	2893.0	12.8	1452.5	6.42	1180.3	5.22	1272.1	5.62	1196.5	5.29	2010.1	8.89
1.25	2431.2	10.75	2751.6	12.17	2009.6	8.89	1365.6	6.04	1126.6	4.98	1180.3	5.22	1176.2	5.20
1.5	2689.9	11.90	2719.1	12.03	2647.8	11.71	2019.3	8.93	1249.1	5.52	1281.7	5.67	1175.3	5.20

Table 7

Width 0.50	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	1116	4.93	2908.7	12.87	3187.2	14.10	2827.9	12.51	2328.6	10.3	2754.6	12.18	2811.2	12.43
0.25	2590.4	11.46	1094.9	4.84	1791.8	7.92	2786.1	12.32	2730.2	12.08	2503.3	11.06	2461.3	10.89
0.5	2621.9	11.60	1773.3	7.84	**1050.7***	**4.64***	1177.3	5.20	1369.2	6.05	1964.3	8.69	2361.5	10.44
0.75	2625.3	11.61	2885.1	12.76	1280.7	5.66	1099.3	4.86	1420.1	6.28	1348.3	5.96	2208.5	9.77
1	2394.9	10.59	2717.3	12.02	1579.9	6.99	1628.2	7.20	1209.8	5.35	1284.8	5.68	1357.5	6
1.25	2914.6	12.89	2583.4	11.43	1873.0	8.28	1352.1	5.98	1330.8	5.88	1067.6	4.72	1096.9	4.85
1.5	2889	12.78	2655.4	11.74	2726.1	12.06	1826.3	8.08	1638.7	7.25	1205	5.33	1071	4.73

Table 8

Width 0.75	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	1358.8	5.99	2504.6	11.08	2731.2	12.08	2830.7	12.52	2561.9	11.33	2754.6	12.18	3122.4	13.81
0.25	2587.2	11.44	1165.2	5.15	1948.7	8.62	2763.2	12.22	2476.3	10.95	2598.1	11.49	2929.7	12.96
0.5	2989.3	13.22	1909.4	8.44	1280.6	5.66	1175.9	5.20	1815.3	8.03	1800.7	7.96	2040.3	9.02
0.75	2425.9	10.73	2710.4	11.99	1307.3	5.78	**1040.5***	**4.60***	1251.1	5.53	1384.2	6.12	1880	8.31
1	2654.4	11.7	2683.2	11.87	1530.4	6.77	1124.9	4.97	1192.8	5.27	1124.3	4.97	1295.2	5.73
1.25	3099.4	13.71	2460	10.88	1980.5	8.76	1252.1	5.54	1186	5.24	1169.8	5.17	1114.9	4.93
1.5	2826.5	12.5	2380.4	10.53	2568.1	11.36	1628.7	7.2	1246.5	5.51	1287.2	5.69	1221.9	5.4

Table 9

Tables 7, 8 and 9 show the values of average steps and rate of exploration for the third maze maze, for 3 different size of similarity kernels as well as different kernel sizes of transition pdf estimation. The top row represents variation along the y-axis and the first column shows the variation along the x-axis. The similarity kernel size is shown at the top left corner. The underlined values represent the combination of kernel size for which divergence computation was very tiny and the values with asterisk show minimum observed values of average steps and rate of exploration.

Width 0.25	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	733	3.1	724.6	3.07	880.8	3.73	738.5	3.12	714.7	3.02	753.9	3.19	690.2	2.92
0.25	799.8	3.38	726.3	3.07	785.5	3.32	727.9	3.08	780.7	3.30	751.9	3.18	762.2	3.22
0.5	725.5	3.07	758.6	3.21	787.4	3.33	688	2.91	743.6	3.15	788.9	3.34	710.9	3.01
0.75	828.5	3.51	782	3.31	755.8	3.2	737.3	3.12	733.8	3.1	**672.5***	**2.84***	791.5	3.35
1	835.1	3.53	679.2	2.87	819.6	3.47	779.6	3.30	682	2.88	701.7	2.97	772.3	3.27
1.25	744.5	3.15	874.6	3.7	818.9	3.46	723.9	3.06	713.7	3.02	742.7	3.14	763.7	3.23
1.5	781.6	3.31	714.6	3.02	717.2	3.03	745.7	3.15	744.3	3.15	766.8	3.24	744.6	3.15

Table 10

Width 0.50	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	750.8	3.18	784.5	3.34	**666.7***	**2.82***	726.6	3.07	688.9	2.91	788	3.33	744.5	3.15
0.25	725.3	3.07	715.2	3.03	790.6	3.35	728.4	3.08	712	3.01	786.6	3.33	754.7	3.19
0.5	667.7	2.82	794.6	3.36	733.2	3.1	732	3.01	680.3	2.88	807.5	3.42	816.7	3.46
0.75	769.8	3.26	820.8	3.47	740.3	3.13	732.6	3.1	766.6	3.24	709.2	3	789.6	3.34
1	717.2	3.03	750.7	3.18	724.6	3.07	773	3.27	758.4	3.21	788.4	3.34	776.5	3.29
1.25	695.8	2.94	795.3	3.36	699	2.96	709.3	3	751	3.18	704.8	2.98	749.9	3.17
1.5	799.8	3.38	745.7	3.15	723.4	3.06	780.3	3.30	820.5	3.47	757.8	3.21	701	2.97

Table 11

Width 0.75	0.1		0.25		0.5		0.75		1		1.25		1.5	
	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate	Avg	Rate
0.1	715.8	3.03	740.8	3.13	802.6	3.40	837.7	3.54	739.2	3.13	727	3.08	736.2	3.11
0.25	681.3	2.88	690.9	2.92	805.8	3.41	714.2	3.02	677.3	2.86	773.5	3.27	775.5	3.28
0.5	673.7	2.85	735.9	3.11	743.8	3.11	713	3.02	729.7	3.09	700.8	2.96	685.6	2.9
0.75	723.3	3.06	729.9	3.09	758.2	3.21	669.5	2.83	723.4	3.06	700.2	2.96	833.4	3.53
1	713.4	3.02	779.3	3.3	721.8	3.05	**634.3***	**2.68***	729.1	3.08	683.6	2.89	758.4	3.21
1.25	735.7	3.11	762.4	3.23	756.1	3.23	767.8	3.25	699.7	2.96	751.7	3.18	799.6	3.38
1.5	841.9	3.56	739.5	3.13	695.6	2.94	666.5	2.82	739.3	3.13	726.6	3.07	819.6	3.47

Table 12

Tables 10, 11 and 12 show the values of average steps and rate of exploration for the fourth maze, for 3 different size of similarity kernels as well as different kernel sizes of transition pdf estimation. The top row represents variation along the y-axis and the first column shows the variation along the x-axis. The similarity kernel size is shown at the top left corner. The underlined values represent the combination of kernel size for which divergence computation was very tiny and the values with asterisk show minimum observed values of average steps and rate of exploration.

and is observed at σ_x = 0.75, σ_y = 0.5, and the minimum value for kernel size = 0.5 is 4.64 and is observed at σ_x = 0.5, σ_y = 0.5.

In comparison, we observe that radom policy takes a staggering average of 22797.71 steps to achieve an equivalent degree of exploration and the rate of exploration comes out to to be 100.87 steps per visited state.

This means that divergence to go is approximately 22 times faster than random search for the third maze.

D. Maze 4

The fourth maze under consideration resembles the maze used in PAC-MAN. It doesn't require a highly guided approach to explore thoroughly but has a number of narrow corridors and turns that will benefit from guided exploration.

For divergence to go, the minimum value of rate of exploration = 2.68 and the average number of steps taken = 634.3. The value is observed at similarity kernel size = 0.75, σ_x = 1, σ_y = 0.75. The minimum rate for kernel size = 0.25 is 2.84 and is observed at σ_x = 0.75, σ_y = 1.25, and the minimum value for kernel size = 0.5 is 2.82 and is observed at σ_x = 0.1, σ_y = 0.5.

In comparison, we observe that radom policy takes an average of 7355.58 steps to achieve an equivalent degree of exploration and the rate of exploration comes out to 31.16.

This means that divergence to go is approximately 12 times faster than random search for the fourth maze.

Further we can make the observation that even with suboptimal selection of similarity and transition kernel size, we achieve an improvement of minimum 2-4 times and the average steps taken to explore the mazes by dtg are consistently under the minimum number of steps taken by random choice.

7. CONCLUSION

Divergence-to-go is an approach based on ITL descriptors which guides exploration through the use of divergence and entropy. It can be applied to discrete or continuous action spaces. It explores the maze faster by choosing actions which maximise divergence so that unexplored parts of the state space are visited first. Dtg handily outperforms random choice in every metric. There are still a few problems that exist within the first being optimal kernel width. Since, dtg is based on kernel density estimation it is of utmost importance to choose the size properly. Another problem is that if the scales of similarity and kernel sizes do not match, then the divergence computed is very tiny which has the potential to become computationally intractable.

REFERENCES

[1] M. Emigh, J. Principe, “Directed Exploration using Divergence-to-Go”

[2] J. Principe, “Information Theoretic Learning - Renyi’s Entropy and Kernel Perspectives”

[3] D. Xu, “Energy, Entropy, and Information Potential for Neural Computation”

[4] J. Principe, D. Xu, “Information-Theoretic Learning using Renyi’s Quadratic Entropy”

[5] M. P. Deisenroth, *Efficient reinforcement learning using gaussian processes*. KIT Scientific Publishing, 2010, vol. 9.

[6] R. Bellman, *Dynamic Programming*, 1st ed.

[7] W. Liu, P. P. Pokharel, and J. C. Prı́ncipe, “Correntropy: properties and applications in non-gaussian signal processing,” IEEE Transactions on Signal Processing, vol. 55, no. 11, pp. 5286–5298, 2007.

[8] J. Bae, L. S. Giraldo, P. Chhatbar, J. Francis, J. Sanchez, and J. Principe,“Stochastic kernel temporal difference for reinforcement learning,” in Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on. IEEE, 2011, pp. 1–6.

[9] H. Hasselt, M. Wiering “Reinforcement Learning in Continuous Action Spaces”, ADPRL 2007

[10]R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. The MIT press, Cambridge MA, A Bradford Book, 1998.