Unsupervised Representation Learning by Latent Plans

Anonymous Authors[1]

Abstract

This paper introduces **Plan2Vec**, an unsupervised representation learning objective inspired by value-based reinforcement learning methods. We show that even without access to actions, we can learn plannable representations that inform long-range structures, purely passively from high-dimensional sequential datasets without supervision. The network learns by playing an "Imagined Planning Game" on the graph formed by the dataset, using a local metric function trained contrastively from context. We show that the global metric on this learned embedding can be used to plan with O(1) complexity by linear interpolation.This is an exponential speed-up critical for planning on any learned representation that contains non-trivial global structure.

1. Introduction

Much of self-supervised or unsupervised learning from sequential data are concerned with learning from structures available within a single frame of observation (Kingma & Welling, 2013; Hjelm et al., 2018; Oord et al., 2018), or sequences in a narrow spatiotemporal-window (Perozzi et al., 2014; Caron et al., 2018) As a result, such learning objectives usually place only local constraints on the embedding.

In addition, it is often unclear in an unsupervised learning setting what construes a "good feature". The usual fall-back is to evaluate the learned features on a set of classification tasks (Guo et al., 2018), or showing samples that activates a particular filter channel (Caron et al., 2018). In reality, when such representations are used on an agent trying to master a certain task in the real-world, the embedding devolves into nothing more than a lossy dimensionality reduction of the original input, lacking otherwise informative structure beyond what's inside each image. This problem becomes more pronounced when the latent configuration space underlying the observations are complex and high-dimensional. Take humanoid for example, without search heuristics involving a long-range metric, classical planning algorithms that take advantage of only local constraints would slow down exponentially in such cases, eventually falling to a halt (Chua et al., 2018).

Meanwhile, value-based reinforcement learning algorithms, in particular universal value function approximators (**UVFA**, see Schaul et al. (2015)) aim to learn a goal-conditioned value function $V(s, g)$ over all state-goal pairs. This is equivalent to learning a global and long-range distance metric. Besides being useful as an informative heuristics for planning, such learned value function can be used as linear features to represent temporal abstraction over actions (Sutton & Tanner, 2005) in addition to the observed state space. When combined with a closed-form policy trained in-tandem where the gradient/reward signal for the policy comes entirely from the learned value function approximator, such methods are able to learn highly complex maneuvers on non-trivial topology with or without explicit forward planning (Peng et al., 2018; Pong et al., 2018).

Motivated by these observations, we propose to learn a plannable representation with *no supervision*, by introducing sample-based value iteration with a planning policy as the sole learning objective. The theoretical difficulty is threefold:

1. Standard formulation of reinforcement learning require *substantial human supervision* in the form of meticulously shaped dense **rewards**.
2. Reinforcement learning is **active**. It requires interaction with a environment between optimization phases to receive on-policy trajectories that eventually reaches optimality.
3. In order to plan on a continuous state and action space, one usually need to learn a close-form behavior **policy**, or a forward model of the environment.

The main contribution of this paper is that we solve all three problems, by formulating learning the global structure of a data manifold as learning a planning agent that tries to master an imagined "reaching game" on a dataset. To solve *point 1*, we make human supervision for designing the reward unecessary by using a local metric function trained contrastively by local context as the reward function. To solve *point 2*, we remove the need for either action data,

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

or a model of the world, by planning entirely in the latent configuration space on a graph. To solve *point 3*, we formulate the policy as a planning network that uses the global metric being learned as a planning heuristic. We show that on low dimentional state space we can boostrap a global metric by doing value iteration on planned paths. In addition we show preliminary results on deformable object manipulation, where the data comprise of different rope configurations.

2. Learning Representation by Latent Planning

Our objective is to find a way to learn a representation that has sensible global structure that makes non-trivial planning computationally feasible in the latent space. In this section, we will describe a method that casts learning such a representation as an *imagined "planning game"* that the network plays, using only unsupervised temporal sequence data. Different from (Watter et al., 2015; Banijamali et al., 2017) and similiar to (Kurutach et al., 2018), our method does not rely on dynamics of the underlying environment in the form of sampled action data, and neither do we learned a forward model. This is because such local and detailed information often distracts from long-range planning. Instead, our imagined game occurs on a graph where disjoint temporal sequences are connected via a local metric function trained using a contrastive loss similar to (Sermanet et al., 2017; Mikolov et al., 2013). Our network then optimizes the embedding of this graph by learning a policy for navigating this graph that plans using this embedding as a planning heuristic.

In the next sections, we will overview how our method "connects the dots" by learning a local metric function, and then extrapolating these local knowledge of the dataset to a global embedding via value-iteration. During each game play, we sample two random samples x_0 and x_g from the dataset. We formulate the task as trying to reach the target x_g from x_0 by hopping through intermediate datapoints $x_{[1:g]}$. In a typical sequential dataset in a continuous sample space, points from different sequences are rarely identical. To connect the dots, and construct a connected graph on which planning could happen, we first learn a local metric function contrastively from the local context within each sequence. We show that the local metric function learned this way generalize well.

Then we formulate the imagined "planning game" and the planning agent. The agent samples in the dataset using the global metric that we are trying to learn as the heuristic. The reward simply measures how many hops that the agent has made in reaching the target. By focusing on observation data without action samples, we avoid learning a forward dynamic model, which often distracts from the long-range planning objective.

Our method builds upon prior works in unsupervised representation with contrastive losses and value-based reinforcement learning methods. In particular, we will overview methods that learn a local distance metric between pairs of images, and value iteration under a standard Markov decision process (MDP) formalism.

3. Contrastive Losses and Local Metric

Our key observation is that in sequential dataset, the temporal sequence is usual optimal over shorter temporal span, and suboptimal over longer ranges. Learning a local metric function has the advantage that the model only need to memorize datapairs in a reduced neighborhood, leading to improved generalization.

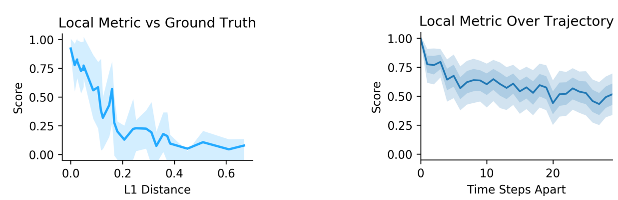

Figure 1. Monte Carlo sampled data offers good local learning signal, but over long-term, the behavior policy responsible for sampling is usually sub-optimal. As a result the distance over the trajectory is usually not well-behaved. **Left**: prediction of metric versus ground-truth distance, learned from image-input. **Right**: same local metric function evaluated over complete sequences. One can see that over longer range, the signal start to contain much less contrast.

Algorithm 1 Local Metric Learning with Contrastive Loss

Require: set of observation sequences $\{\tau = x_{[0:T]}\}$

Initialize f_ϕ

Sample x_t, x^+_{t+1} where $x_t, x_{t+1} \in \tau_i, y^+ = 1$

Sample x_t, x^- where $x^- \sim \tau_j$ where $x \notin \tau_j, y^- = 0$

for each epoch **do**

  minimize $\|f_\phi(x, x^*) - y^*\|_2$ for $x, x^\pm, y^\pm$

end for

We can then use this local metric function to connect disjoint trajectories in the dataset into a connected graph, and then use it as a good supervision signal as the reward for learning the planning agent.

3.1. Organizing the Gloabl Structure of the Latent Space By Planning

We formulate value iteration under the Markov decision process (MDP) formalism. An MDP is usually parameterized as the tuple $\langle S, A, P, R\rangle$ where S and A are the set of state and actions. $P(s'|s, a)$ is the transition function of the

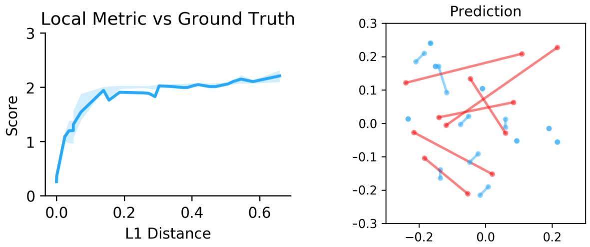

Figure 2. **Left**: Contrastively trained local metric funcition. Y-axis is the output of the metric function (score). x-axis is the ground-truth distance between the two input samples. Trained from state-space inputs. **Right**: input data pairs. Color red indicates the metric function considers the pair to be *far-apart*, blue indicate that the pair are in a small neighborhood.

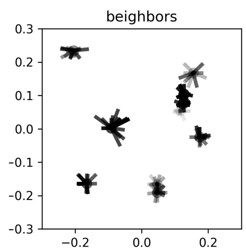

Figure 3. Plot showing clusters of neighbors that our local metric function learns. Each cluster orginates from a single datapoint. Each radial line indicates one neighbor. Most neighbors are from trajectories different from the one for the point itself.

environment. $R(s, a, s')$ is the reward. An agent is usually represented by the policy distribution $\pi(a|s)$.

Take the reward function for a specific task $R_{\mathcal{T}}$ and a policy π, we can learn a state-action value function $Q_\pi : S \times A \rightarrow \mathbb{R}$ that returns the expected future value for executing action a at state s, conditioned on the policy.

In sample-based Q-learning with deep neural network function approximators, we minimize the sample-based bellman-residual

$$\delta = \left\| V(s) - \mathcal{B}V \right\| \tag{1}$$

where the bellman operator is defined as

$$\mathcal{B}V = R(s_t, a_t, s_{t+1}) + \gamma \max_a V(s_{t+1}). \tag{2}$$

Samples takes the form of the tuple $\langle s_t, a_t, r_t, s_{t+1} \rangle$, and usually importance sampled from a replay buffer.

Because value-iteration is on-policy, we clear the replay buffer after a few epochs. We also use high-sight experience re-labeling to insert positive reaching examples from the trajectories to improve the rate of learning.

Because game plays occur entirely inside the imagined task on the dataset, we can sampling multiple next points for

Algorithm 2 Unsupervised Learning by Latent Plans

Require: planning horizon H
Require: set of observation sequences $S = \{\tau = x_{[0:T]}\}$
Require: local metric function $\phi(x, x') \Rightarrow \mathbb{R}^+$
Require: reward function $r(x, x_g) = \mathbf{1}_{N(x_g, \epsilon)} - 1$
Initialize global embedding $\varphi(x, x') \Rightarrow \mathbb{R}^+$
repeat
  sample $x_0, x_g \in S$ as *start* and *goal*
  repeat {h=0, h++}
    find *set* $\mathbf{n} = \{x' \text{ s.t. } \phi(x_0, x') \in N(1, \epsilon)\}$
    find $x^* = \arg\min_{x \in \mathbf{n}} \varphi(x, x_g)$
    compute $r_t = r(x^*, x_g)$
    add $\langle x, x^*, r_t, x_g \rangle$ to buffer B
  until r = 0 or h = H
  Sample $\langle x, x', r, x_g \rangle$ from B
  minimize $\delta = \left\| V_\varphi(x, x_g), r + V_\varphi(x', x_g) \right\|_2$
until convergence

the value function replay. This is directly related to soft-Q learning in that the hard maximization operation is now replaced by a boltzmann sampling, where the distance function is treated as an energy model. A temperature constant regulates the sampling of this behavior policy

We experimented with three different types of reward.

- using the ground-truth distance as the reward. This case serves as a control, to validate that other parts of the algorithm is working.
- binary reward where the reward is 0 if the next planed step is within the neighborhood predicted by the local-metric, 1 otherwise.
- using the value of the learned local-metric function as the reward.

Reward	**Planning Success Rate**
ground truth r	$96.8\% \pm 2$
local metric r	$96.6\% \pm 1$
binary reward	$95.8\% \pm 1$

We found that all three methods learns well. The fact that we can use a binary reward instead of a local-metric learned by contrastive regress, means that we could potentially use more conceptual local information for training.

Figure 6. Examples of rope pairs that are connected (positive, *left*), and not connected (negative, *right*). These samples are found in the dataset by a trained local-metric function.

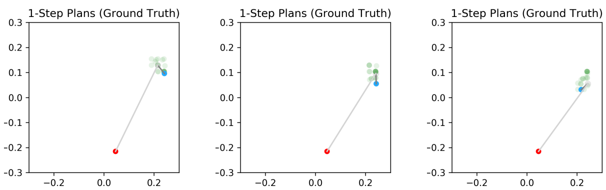

Figure 4. Planning steps learned via value iteration. Red dot is the goal position, blue dot is the plannd next step (1-step) using the global metric function. Green dots are the neighbors inferred via the local metric function. Gray dot is the current position of the agent.

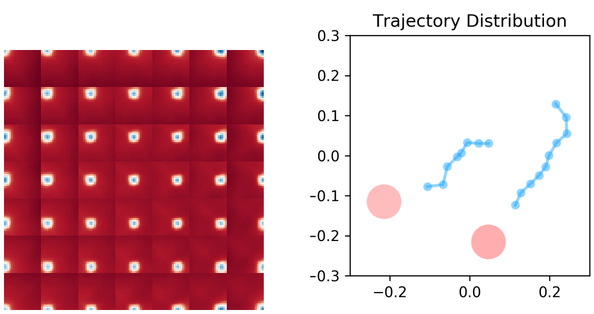

Figure 5. The learned value function and the planned trajectories. **Left**: 7x7 Map of the value function. Each plaque (out of 7x7) is a 21x21 map of the state space. Value goes from zero (blue) to red. Each plaque corresponds to a different goal position, visible as the peak (blue) of the plaque. **Right**: Two planned trajectories from the left value function. Showing the planned trajectories succesfully reaching towards the goals.

Our result shows that value iteration is able to learn a global embedding without supervision on a 2-dimensional robot navigation domain. The planning agent is able to reach the goal position within 10 steps of planning. We believe these results show great promise for extending our method to more complex global topology.

4. Discussions

The work most similar to us from the manifold learning community is DeepWalk (Perozzi et al., 2014). DeepWalk uses a random policy to sample short trajectories from a social graph. Then uses skip-gram (Mikolov et al., 2013) to learn a node embedding from its context in those trajectories. This contextural embedding objective resembles the contrastive embedding loss we use to supervise our local metric function. Despite of these similarities, DeepWalk does not formulate a learned policy, and falls under the category of representation learning algorithms that only learns from a localized context. Our key contribution is to cast unsupervised learning as learning a representation for a planning agent.

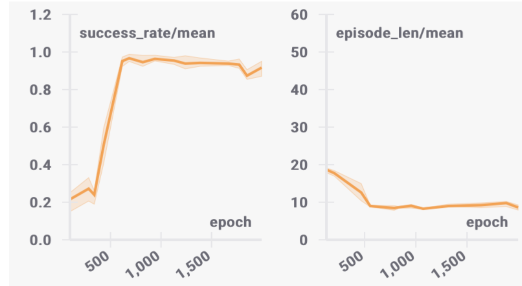

Figure 7. Learning curve of the planning agent. Showing 96.7% planning success rate with the learned value function (**left**), and average 10 steps in reaching the goal position during planning.

In manifold learning, locally linear embedding (LLE) similar to DeepWalk in that the local linear embedding could be considered a “strongger” version of skip-gram, whereas linear contributions of each neighbor is preserved. However, similar VAE, LLE enforces global structure, and prevent volumn collapse via addition of a volume regularization term globally. This is similar to the variational prior in an VAE, both lack meaningful alignment with planning semantics.

The similarity between DeepWalk and diffusion map literature is apparent. Compared with these, our method explicitly trains a policy which generate on-policy trajectors that are optimum. One can argue that diffusion maps are more closely related to the soft version of our algorithm (see Alg.3). The hard sampling version (see Alg.2) is more akin to Walker’s Q-learning in that the operator contains an optimum, or when temperature is zero in the diffusion map, where equilibrium will take infinitely long.

5. Appendix A: Soft-Value Iteration

References

Banijamali, E., Shu, R., Ghavamzadeh, M., Bui, H., and Ghodsi, A. Robust locally-linear controllable embedding. *arXiv preprint arXiv:1710.05373*, 2017.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. *arXiv preprint arXiv:1807.05520*, 3, 2018.

Algorithm 3 Soft Planning Alternative (difference in red)

Require: planning horizon H
Require: set of observation sequences $S = \{\tau = x_{[0:T]}\}$
Require: local metric function $\phi(x, x') \Rightarrow \mathbb{R}^+$
Require: reward function $r(x, x_g) = \mathbf{1}_{N(x_g, \epsilon)} - 1$

```
Initialize global embedding φ(x, x') ⇒ ℝ⁺
repeat
    sample x_0, x_g ∈ S as start and goal
    repeat {h=0, h++}
        find set n = {x' s.t. ϕ(x_0, x') ∈ N(1, ε)}
        find d_i, x_i = soft max_{x∈n} φ(x, x_g)
        compute r_t^i = r(x_i, x_g)
        add ⟨x, {x_i, d_i, r_i}, x_g⟩ to buffer B
    until r = 0 or h = H
    Sample ⟨x, {x'_i, d_i, r_i}, x_g⟩ from B
    minimize δ = ‖V_φ(x, x_g), Σ (e^{d_i/T} / Z) [r_i + V_φ(x'_i, x_g)]‖_2
until convergence
```


Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pp. 4759–4770, 2018.

Guo, Z. D., Azar, M. G., Piot, B., Pires, B. A., Pohlen, T., and Munos, R. Neural predictive belief representations. *arXiv preprint arXiv:1811.06407*, 2018.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kurutach, T., Tamar, A., Yang, G., Russell, S. J., and Abbeel, P. Learning plannable representations with causal infogan. In *Advances in Neural Information Processing Systems*, pp. 8747–8758, 2018.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):143, 2018.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710. ACM, 2014.

Pong, V., Gu, S., Dalal, M., and Levine, S. Temporal difference models: Model-free deep rl for model-based control. *arXiv preprint arXiv:1802.09081*, 2018.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *International Conference on Machine Learning*, pp. 1312–1320, 2015.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., and Levine, S. Time-contrastive networks: Self-supervised learning from pixels. 2017.

Sutton, R. S. and Tanner, B. Temporal-difference networks. In *Advances in neural information processing systems*, pp. 1377–1384, 2005.

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.