Learning POMDP Models with Similarity Space Regularization: a Linear Gaussian Case Study

Yujie Yang YANGYJ21@MAILS.TSINGHUA.EDU.CN
School of Vehicle and Mobility, Tsinghua University, Beijing, China

Jianyu Chen JIANYUCHEN@TSINGHUA.EDU.CN
Institute of Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
Shanghai Qizhi Institute, Shanghai, China

Shengbo Eben Li LISHBO@TSINGHUA.EDU.CN
School of Vehicle and Mobility, Tsinghua University, Beijing, China

Editors: R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, M. Kochenderfer

Abstract

Partially observable Markov decision process (POMDP) is a principled framework for sequential decision making and control under uncertainty. Classical POMDP methods assume known system models, while in real-world applications, the true models are usually unknown. Recent researches propose learning POMDP models from the observation sequences rolled out by the true system using maximum likelihood estimation (MLE). However, we find that such methods usually fail to find a desirable solution. This paper makes a profound study of the POMDP model learning problem, focusing on the linear Gaussian case. We show the objective of MLE is a high-order polynomial function, which makes it easy to get stuck in local optima. We then prove that the global optimal models are not unique and constitute a similarity space of the true model. Based on this view, we propose Similarity Space Regularization (SimReg), an algorithm that smooths out the local optima but keeps all the global optima. Experiments show that given only a biased prior model, our algorithm achieves a higher log-likelihood, more accurate observation reconstruction and state estimation compared with the MLE-based method.

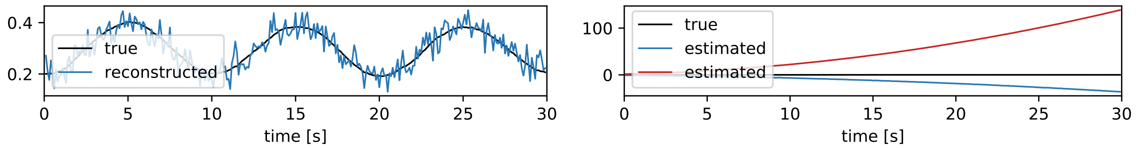
Keywords: partially observable Markov decision process, model learning, maximum likelihood estimation, similarity space

1. Introduction

Many sequential decision making and control problems are faced with uncertainty, such as robotics Thrun (2002), computer games Vinyals et al. (2019) and autonomous driving Brechtel et al. (2014). These problems can be formulated as partially observable Markov decision process (POMDP) Kaelbling et al. (1998), which is composed of a stochastic state transition model and a stochastic observation model. In POMDP, the system model plays an important role in finding a good policy, which can be used for state estimation, belief state inference, as well as obtaining the optimal policy through planning Silver and Veness (2010); Somani et al. (2013); Kurniawati and Yadav (2016). Classical POMDP methods assume known models Smallwood and Sondik (1973); Kaelbling et al. (1998); Pineau et al. (2003); Smith and Simmons (2004). However, an accurate model is often unknown in real-world problems.

Recent advances in machine learning enable learning the POMDP models from only the observation sequences rolled out by the true systems. In problems with simple models, such as linear Gaussian models and low-dimensional discrete state space models, the posterior distribution of the state has closed-form solutions. In these cases, the expectation-maximization (EM) algorithm Dempster et al. (1977) can be used for model learning Roweis and Ghahramani (1999); Ghahramani (2001); Schön et al. (2011), which iteratively computes the expectation of the log-likelihood and calculates parameters that maximize the expectation. In problems with high-dimensional nonlinear models, the posterior distribution of the state is intractable and Variational inference (VI) Kingma and Welling (2013) is usually adopted to learn the posterior distribution of the state and the parameters of the POMDP models Bayer and Osendorfer (2014); Chung et al. (2015); Krishnan et al. (2015); Karl et al. (2017); Krishnan et al. (2017); Ha and Schmidhuber (2018).

Both EM algorithm and VI-based algorithms follow the maximum likelihood estimation (MLE) framework, which tries to find the optimal model by maximizing the likelihood of observation sequences. However, as we discover in our experiments, such methods often fail to learn a desirable model. Figure 1 shows some example experiment results for learning a vehicle model using MLE (More details can be found in Section 4). Figure 1(a) shows the mean of observation distribution reconstructed by the learned model (in blue), compared with that rolled out by the true system (in black). The reconstructed observations are very noisy, indicating that the model does not learn reasonable system dynamics to filter out the noises. Figure 1(b) shows the state estimation results of the learned models (blue and red) compared with the true state (black). The two estimated curves are generated by models trained on the same data set but under two different random seeds. The state estimations significantly diverge from the true values and behave totally differently. The above phenomenons motivate us to raise the following questions:



(a) Mean of observation distribution

(b) State

Figure 1: Evaluations of the learned models using MLE.

1. Why is MLE unable to learn a good model?
2. How to formally define a good model? Is it uniquely defined by the true system that generates the data?
3. How can we design an algorithm for learning a good model?

To answer the above questions, we make a profound study of the model learning problem in POMDP. We focus on linear Gaussian systems in this paper for clearer intuitions and more tractable theoretical analysis. Our contributions are listed as follows:

- We point out that MLE is very likely to learn a local optimal model even in linear Gaussian systems. We explain this by deriving the objective function in analytical form, which turns out to be a high-order polynomial of the model parameters.

- We show that the global optimal models are not unique and they constitute a similarity space containing the true model. This indicates that the goal for learning is not necessarily to find the true model, but to find an arbitrary model in this similarity space.
- We propose an algorithm called Similarity Space Regularization (SimReg), which smooths out the local optima but keeps the global optima. Experiments show that given only a biased prior model, our algorithm achieves a higher likelihood, more accurate observation reconstruction and state estimation compared with MLE.

2. Preliminary

2.1. Linear Gaussian system

A linear Gaussian system is described by the following equations,

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t, \\ y_t &= Cx_t + v_t, \end{aligned} \tag{1}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^p$ and $y_t \in \mathbb{R}^q$ are the state, control and observation of the system at time step t. $w_t \in \mathbb{R}^n$ and $v_t \in \mathbb{R}^q$ are the state noise and observation noise at time step t. We assume that w_t and v_t are i.i.d. and follow zero-mean Gaussian distributions $\mathcal{N}(0, W)$ and $\mathcal{N}(0, V)$, respectively. $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{q\times n}$ are state matrix, control matrix and observation matrix. We denote $\theta = \{A, B, C, W, V\}$ as unknown parameters to be learned.

2.2. Maximum likelihood estimation

Maximum likelihood estimation (MLE) is a method for estimating the parameters of a probabilistic model given observation data. In the POMDP model learning problem, the objective of MLE is to find the optimal parameter θ^* that maximizes the log-likelihood of observation sequences,

$$\theta^* = \arg\max_{\theta} \mathbb{E}[\log p(y_{0:T}; \theta)], \tag{2}$$

where $y_{0:T} = \{y_0, y_1, \ldots, y_T\}$ is the observation sequence from time step 0 to T. In the rest of this paper, we omit the parameter θ in the log-probability for simplicity.

2.3. Kalman filter

Kalman filter is widely used for state estimation in linear Gaussian systems with known model. It has two steps, predict and update. In the predict step, the state estimate and estimate covariance are predicted using the estimation in the last time step,

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1} + Bu_{t-1}, \tag{3}$$

$$\Sigma_{t|t-1} = A\Sigma_{t-1}A^T + W, \tag{4}$$

where $\hat{x}_{t|t-1}$ and $\Sigma_{t|t-1}$ are the predicted state estimate and estimate covariance. $\hat{x}_{t-1}$ is the estimation of time step $t-1$. In the update step, the state estimate and estimate covariance are updated using the new observation y_t,

$$\hat{x}_t = \hat{x}_{t|t-1} + L_t(y_t - C\hat{x}_{t|t-1}), \tag{5}$$

$$\Sigma_t = (I - L_t C)\Sigma_{t|t-1}, \tag{6}$$

where

$$L_t = \Sigma_{t|t-1} C^T (C \Sigma_{t|t-1} C^T + V)^{-1} \tag{7}$$

is the Kalman gain.

Under certain conditions, the Kalman filter converges to a linear time-invariant filter as time goes infinity. In this case, the estimate covariance and the Kalman gain converge to constant matrices Σ and L. The steady-state estimate covariance Σ can be obtained by solving the Discrete Algebraic Riccati Equation (DARE):

$$\Sigma = A(\Sigma - \Sigma C^T (C\Sigma C^T + V)^{-1} C\Sigma) A^T + W. \tag{8}$$

The steady-state Kalman gain is:

$$L = \Sigma C^T (C\Sigma C^T + V)^{-1}. \tag{9}$$

3. Method

3.1. Local optima of maximum likelihood estimation

In a linear Gaussian system, we can derive the analytical form of the log-probability in (2). Use the sequential property of the observation sequence, we can write the log-probability in a summation form,

$$\log p(y_{0:T}) = \sum_{t=0}^{T} \log p(y_t | y_{0:t-1}). \tag{10}$$

According to Kalman filter,

$$p(y_t|y_{0:t-1}) = \mathcal{N}(y_t | C\hat{x}_{t|t-1}, C\Sigma_{t|t-1}C^T + V). \tag{11}$$

For simplicity of the following derivations, we consider the case where the steady-state Kalman filter is used. Then the estimation covariance and Kalman gain are both constant matrices. Furthermore, we consider the case where the state and observation are single-dimensional. Note that the derivations can be extended straightforwardly to the generic cases. We have

$$p(y_t|y_{0:t-1}) = \mathcal{N}(y_t | C\hat{x}_{t|t-1}, C^2\Sigma + V). \tag{12}$$

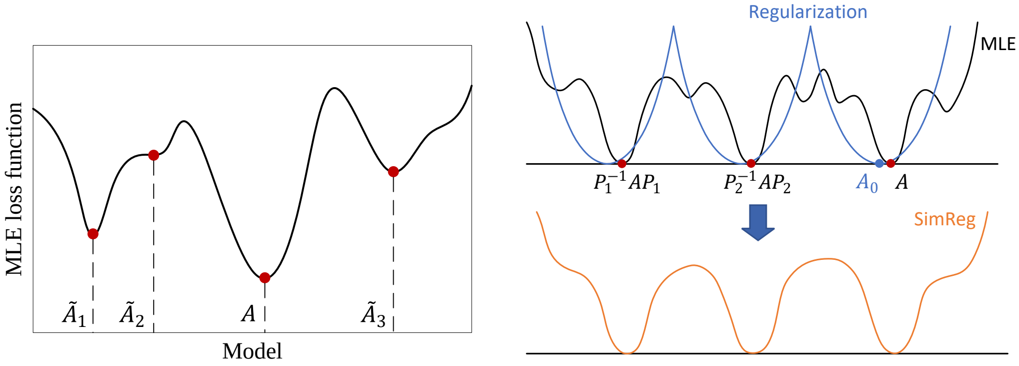
According to the probability density function of Gaussian distribution,

$$\log p(y_t|y_{0:t-1}) = -\frac{1}{2}\log 2\pi (C^2\Sigma + V) - \frac{(y_t - C\hat{x}_{t|t-1})^2}{2(C^2\Sigma + V)}. \tag{13}$$

According to Kalman filter,

$$\begin{aligned}
\hat{x}_{t|t-1} &= A\hat{x}_{t-1} + Bu_{t-1} \\
&= A\left(\hat{x}_{t-1|t-2} + L(y_{t-1} - C\hat{x}_{t-1|t-2})\right) + Bu_{t-1} \\
&= A(I - LC)\hat{x}_{t-1|t-2} + ALy_{t-1} + Bu_{t-1} \\
&= A^t (I - LC)^t \hat{x}_{0|-1} + \sum_{\tau=0}^{t-1} A^\tau (I - LC)^\tau (ALy_{t-1-\tau} + Bu_{t-1-\tau}).
\end{aligned} \tag{14}$$

(14) is a high-order polynomial of model parameters. Take the state matrix A as an example. (14) is a t^{th} order polynomial of A. Then (13) is a $2t^{\text{th}}$ order polynomial of A. When gradient-based algorithm is used to maximize (10), it finds the parameters where the gradient is zero. This results in a root finding problem of $(2T-1)^{\text{th}}$ order polynomial, which has $2T-1$ solutions in the complex field. Among these solutions, some might be the global optima of the objective function and correspond to the optimal model(s) including the true model, while others are local optima or saddle points corresponding to sub-optimal models, as shown in Figure 2(a). When T is large, there are multiple such local optima and MLE is likely to get stuck in one of them.



(a) (b)

Figure 2: Loss functions of MLE and SimReg. (a) The loss function of MLE has multiple local optima. (b) SimReg smooths out local optima near all global optima.

In particular, we discover in our experiments that the state estimation of the learned model tends to be very noisy (as in Figure 1(a)). This is because it learns a small observation noise and large state noise. By doing so, the model simply 'trust' the observation data to increase the log-likelihood. This makes the model easier to get stuck in a local optimum.

3.2. Optimal solutions in similarity space

To avoid bad local optima and obtain the global optima, we need to first answer how many global optima does the MLE objective have and what properties do they have. Clearly, the global optimal solution should maximize the likelihood of observations. When the number of observation data approaches infinity, the global optimal model(s) will generate the same observation distributions as the true system. The following theorem tells us that the global optimal model is not uniquely the true model, but instead the entire similarity space containing the true model.

Theorem 1 *Consider two observable linear Gaussian system models $\mathcal{M}$ and $\tilde{\mathcal{M}}$ with parameters $\theta = \{A, B, C, W, V\}$ and $\tilde{\theta} = \{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{W}, \tilde{V}\}$, respectively. If $\forall x_0$ and control sequence, $\exists \tilde{x}_0$, s.t. the observation sequence generated by $\tilde{\mathcal{M}}$ from initial state $\tilde{x}_0$ follows the same distribution as that generated by $\mathcal{M}$ from initial state x_0, then there exists an invertible matrix P, s.t. $A = P\tilde{A}P^{-1}, B = P\tilde{B}, C = \tilde{C}P^{-1}$.*

Proof Let the controls be zero. The mean of observation at time step t

$$\mu_{y,t} = C\mu_{x,t} = CA^t x_0. \tag{15}$$

Since the distributions of observations generated by the two systems are the same,

$$CA^t x_0 = \tilde{C}\tilde{A}^t \tilde{x}_0, \forall t \geq 0. \tag{16}$$

When $t = 0, 1, \ldots, n-1$, write (16) in matrix form,

$$U_o x_0 = \tilde{U}_o \tilde{x}_0, \tag{17}$$

where

$$U_o = \begin{bmatrix} C \\ CA \\ \cdots \\ CA^{n-1} \end{bmatrix} \tag{18}$$

is the observability matrix of the system. Since the two systems are both observable, the ranks of U_o and $\tilde{U}_o$ are both n. Thus, we can select n linearly independent rows from U_o and obtain an $n \times n$ invertible matrix, which we denote as Q_i. The corresponding rows in $\tilde{U}_o$ also forms an $n \times n$ matrix, which we denote as $\tilde{Q}$. Then we have

$$\begin{aligned} Q_i x_0 &= \tilde{Q}\tilde{x}_0 \\ x_0 &= Q_i^{-1}\tilde{Q}\tilde{x}_0. \end{aligned} \tag{19}$$

Similarly,

$$\tilde{x}_0 = \tilde{Q}_i^{-1} Q x_0. \tag{20}$$

Thus,

$$x_0 = Q_i^{-1}\tilde{Q}\tilde{Q}_i^{-1} Q x_0. \tag{21}$$

The above equation holds $\forall x_0$. Thus, $Q_i^{-1}\tilde{Q}\tilde{Q}_i^{-1}Q = I$. $Q_i^{-1}\tilde{Q}$ and $\tilde{Q}_i^{-1}Q$ are invertible matrices. Let $Q_i^{-1}\tilde{Q} = P$, then $x_0 = P\tilde{x}_0, \tilde{x}_0 = P^{-1}x_0$.

Consider (16) when $t = 0$,

$$\begin{aligned} Cx_0 &= \tilde{C}\tilde{x}_0 \\ Cx_0 &= \tilde{C}P^{-1}x_0 \\ C &= \tilde{C}P^{-1}. \end{aligned} \tag{22}$$

When $t = 1, 2, \ldots, n$, write (16) in matrix form,

$$\begin{aligned} U_o A x_0 &= \tilde{U}_o \tilde{A} \tilde{x}_0 \\ U_o A x_0 &= \tilde{U}_o \tilde{A} P^{-1} x_0 \\ U_o A &= \tilde{U}_o \tilde{A} P^{-1} \\ A &= P\tilde{A}P^{-1}. \end{aligned} \tag{23}$$

Consider a control sequence in which $u_0 \neq 0$ and $u_t = 0, \forall t \geq 1$. Let the initial state $x_0 = 0$, then the mean of observation at time step t

$$\mu_{y,t} = C\mu_{x,t} = CA^{t-1}Bu_0. \tag{24}$$

Thus,

$$CA^{t-1}Bu_0 = \tilde{C}\tilde{A}^{t-1}\tilde{B}u_0, \forall t \geq 1 \tag{25}$$

When $t = 1, 2, \ldots, n$, write (25) in matrix form,

$$U_o B u_0 = \tilde{U}_o \tilde{B} u_0. \tag{26}$$

The above equation holds $\forall u_0$. Thus,

$$\begin{aligned} U_o B &= \tilde{U}_o \tilde{B} \\ B &= P\tilde{B}. \end{aligned} \tag{27}$$

■

Theorem 1 shows that the global optimal models are similar to the true model and they constitute a similarity space. The following Corollary tells us that these global optimal models are also optimal in respect to control and state estimation.

Corollary 2 *If the learned model is similar to the true model, then it gives the optimal solution for state estimation and control in linear quadratic Gaussian (LQG) case.*

In fact, the invertible matrix P is a linear mapping from the learned state to the true state. Thus, the optimal state estimate

$$\hat{x}_t = P\hat{\tilde{x}}_t, \tag{28}$$

where $\hat{\tilde{x}}_t$ is the state estimate of the learned model. The optimal control of LQG

$$u_t^* = -K_t\hat{x}_t = -K_t P\hat{\tilde{x}}_t, \tag{29}$$

where K_t is the feedback gain.

3.3. Learning POMDP model with similarity space regularization

Based on the above analysis, our learning objective should be to find a solution in the similarity space containing the true model. However, without further information about the true model, there is no way for us to find such a solution. Fortunately, we can provide reasonable prior models for a lot of problems. For example, most physical and mechanical systems (including cars, robots, etc.) have well-studied structures of their dynamic models. The downside is that these prior models are usually inaccurate with potentially large bias.

Assuming access to only a biased prior model, we propose the Similarity Space Regularization (SimReg) algorithm, which smooths the local optima but keeps the global optima and learns a model in the vicinity of the similarity space of the true model. Assume that we have a biased prior knowledge of the state matrix,

$$A_0 = A + \delta A, \tag{30}$$

where A is the true state matrix and δA is a deviation. We then try to find a solution close to the similarity space of the prior model. Assume that

$$\|\tilde{A}\tilde{x} - P^{-1}A_0P\tilde{x}\| \leq \varepsilon, \forall \tilde{x}, \tag{31}$$

where $\tilde{A}$ is the learned state matrix, P is an invertible matrix, $\tilde{x}$ is a state in the learned state space and ε is a small constant. According to the definition of operate norm we have:

$$\|\tilde{A} - P^{-1}A_0P\| = \max_{\|\tilde{x}\|=1} \|(\tilde{A} - P^{-1}A_0P)\tilde{x}\| \leq \varepsilon \tag{32}$$

According to the triangular inequality,

$$\begin{aligned}\|\tilde{A} - P^{-1}AP\| &\leq \|\tilde{A} - P^{-1}A_0P\| + \|P^{-1}\delta AP\| \\ &\leq \varepsilon + \|P^{-1}\|\|\delta A\|\|P\| \\ &= \varepsilon + \mathrm{cond}(P)\|\delta A\|.\end{aligned} \tag{33}$$

(33) shows that if the deviation of prior model $\|\delta A\|$ and the condition number of P are small enough, the learned model will be close to the similarity space of the true model, which is an optimal solution according to Corollary 2. Note that

$$\frac{\|\tilde{A}\tilde{x} - P^{-1}A_0P\tilde{x}\|}{\|P^{-1}\|} = \frac{\|P^{-1}(P\tilde{A}\tilde{x} - A_0P\tilde{x})\|}{\|P^{-1}\|} \leq \|P\tilde{A}\tilde{x} - A_0P\tilde{x}\|. \tag{34}$$

Thus,

$$\|P\tilde{A}\tilde{x} - A_0P\tilde{x}\| \leq \frac{\varepsilon}{\|P^{-1}\|} \Rightarrow \|\tilde{A}\tilde{x} - P^{-1}A_0P\tilde{x}\| \leq \varepsilon. \tag{35}$$

Thus, we can make $\|P\tilde{A}\tilde{x} - A_0P\tilde{x}\|$ small in order to make $\|\tilde{A}\tilde{x} - P^{-1}A_0P\tilde{x}\|$ small. Based on this view, we propose the following objective function:

$$J(\Theta)_{SimReg} = \mathbb{E}\left[-\log p(y_{0:T};\tilde{\theta}) + \alpha\sum_{t=0}^{T}\|P\tilde{A}\hat{x}_t - A_0P\hat{x}_t\|_2^2\right], \tag{36}$$

where $\Theta = \{\tilde{\theta}, P\}$ is the parameters to be learned. The first term in the expectation is the original MLE loss. The second term is a regularization that makes the learned model close to the similarity space of the prior model. α is a weight for balancing these two terms. $\hat{x}_t$ is computed using Kalman filter with the learned model. The optimal parameters are solved by minimize (36) using gradient-based optimization algorithms.

An intuitive illustration of SimReg is shown in Figure 2(b). The loss function of MLE has multiple global optima and local optima. The models corresponding to global optima are similar to the true model A. The regularization term is small near all global optima and large elsewhere. It thus smooths out local optima near all global optima and helps the algorithm find a better solution.

4. Experiments

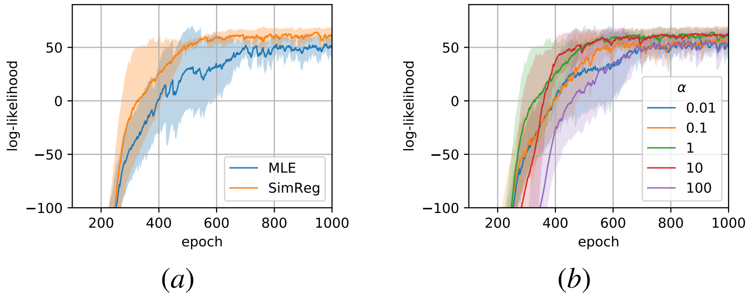
We evaluate SimReg on a vehicle lateral dynamics model. The state is composed of lateral position y, heading angle φ, lateral velocity v and yaw rate ω. The control is front wheel angle δ. The observation includes y and φ. The system model can be written as the following equations,

$$\begin{bmatrix}\dot{y}\\ \dot{\varphi}\\ \dot{v}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix}0 & u & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & \frac{k_1+k_2}{mu} & \frac{ak_1-bk_2}{mu} - u\\ 0 & 0 & \frac{ak_1-bk_2}{I_{zz}u} & \frac{a^2k_1+b^2k_2}{I_{zz}u}\end{bmatrix}\begin{bmatrix}y\\ \varphi\\ v\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ 0\\ -\frac{k_1}{m}\\ -\frac{ak_1}{I_{zz}}\end{bmatrix}\delta + w, \tag{37}$$

$$\begin{bmatrix}y\\ \varphi\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}y\\ \varphi\\ v\\ \omega\end{bmatrix} + v. \tag{38}$$

where u is the longitudinal velocity, k_1 and k_2 are lateral stiffness of the front and rear axles, a and b are distances from the center of gravity to the front and rear axles, I_{zz} is the yaw moment of inertia. w and v are noises that follow zero-mean diagonal Gaussian distributions. We discretize (37) using forward Euler method with time step 0.1s.

The training data are collected by applying actions generated using Gaussian processes with Radial basis function (RBF) kernel. The length scales of RBF kernels are randomly selected from $[5, 20]$ for each sequence. The actions are multiplied by 0.01 as the amplitude. The prior model A_0 is generated by randomly perturbing the diagonal elements of the true model A. We use Adam Kingma and Ba (2015) as the optimization algorithm for learning the parameters. We compare SimReg with MLE. The training curves of log-likelihood are shown in Figure 3(a). SimReg outperforms MLE on both sample efficiency and asymptotic performance. This indicates that MLE gets stuck in local optima while SimReg finds a better solution. The training curve of SimReg is also smoother, indicating that the regularization stabilizes the learning process.



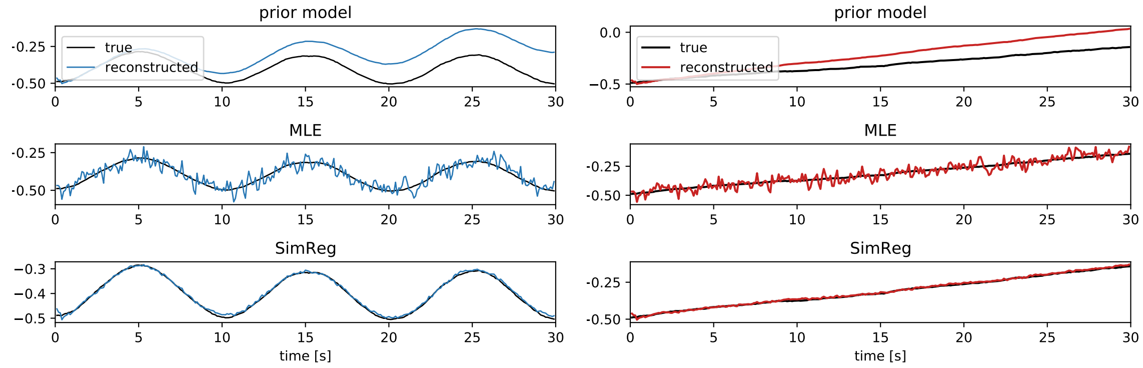
(a) (b)

Figure 3: Training curves of log-likelihood. The solid lines correspond to the mean and the shaded regions correspond to 95% confidence interval over three runs.

We compare the reconstructed observations of different models, as shown in Figure 4. The reconstruction of the prior model is biased since it deviates from the true model. The reconstruction of MLE is very noisy, indicating that the model is a bad local optimum. The reconstruction of SimReg is unbiased and smooth, indicating that the algorithm finds a near-optimal solution. Here the SimReg model is trained with regularization weight $\alpha = 1$ and deviation rate of 0.0049, which is the L2-norm of δA divided by the L2-norm of A.

We then compare the state estimation performances of the models, as shown in Figure 5. The estimation of SimReg is multiplied by P, which is explicitly learned. The state estimation of MLE diverges from the true state, indicating that the model learned an irrelevant state space and is not capable of state estimation. The state estimation of SimReg is close to the true state, indicating that the learned model is close to the similarity space of the true model.

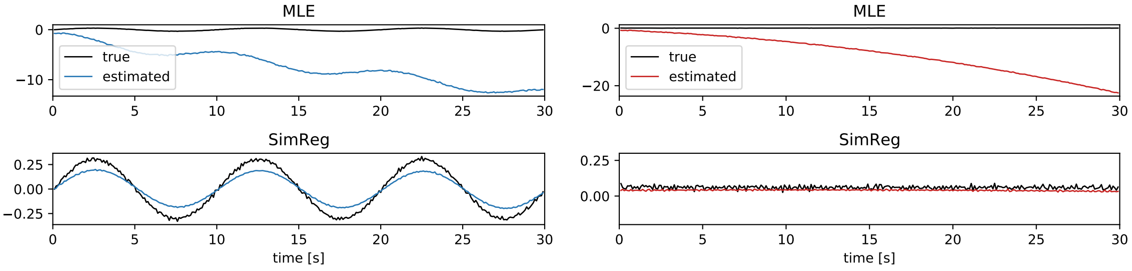
Finally, we compare the training log-likelihood under different values of α, as shown in Figure 3(b). We also compare the mean squared error (MSE) of the mean of reconstructed observation distribution, as shown in Table 1. $\alpha = 0$ corresponds to MLE. Results show that either too small or too large values of α lead to low log-likelihood and large observation reconstruction error. When α is too small, the regularization term has little effect on the objective function thus cannot smooth out the local optima. When α is too large, the penalty of the deviation from the prior model becomes too large, which causes the learned model to deviate from optimal solutions.



(*a*) sinusoidal control

(*b*) constant control

Figure 4: Mean of reconstructed observation distribution.



(*a*) sinusoidal control

(*b*) constant control

Figure 5: State Estimation.

Table 1: Observation reconstruction error under different values of α.

α	0	0.01	0.1	1	10	100
MSE (10^{-4})	14.121	5.587	4.161	1.255	0.627	2.901

5. Conclusion

In this paper, we study the POMDP model learning problem in the case of linear Gaussian systems. We show that MLE is likely to get stuck in local optima because its objective function is a high-order polynomial. We then prove that the optimal solutions of the problem are not unique and they constitute a similarity space of the true system model. Based on this view, we propose Similarity Space Regularization (SimReg), an algorithm that learns a near-optimal solution by regularizing the model to the vicinity of the similarity space corresponding to the global optimal models. Finally, we evaluate our algorithm on a vehicle lateral dynamics model. Results show that our algorithm achieves a higher log-likelihood, more accurate observation reconstruction and state estimation compared with the MLE-based method. In the future, we will extend our analysis and algorithm to general high-dimensional non-linear systems.

References

Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. In *Advances in Neural Information Processing Systems*, 2014.

S. Brechtel, T. Gindele, and R. Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 392–399, 2014. ISBN 2153-0017. doi: 10.1109/ITSC.2014.6957722.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pages 2980–2988, 2015.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–22, 1977. ISSN 0035-9246.

Zoubin Ghahramani. *An introduction to hidden Markov models and Bayesian networks*, pages 9–41. World Scientific, 2001.

David Ha and Jürgen Schmidhuber. World models. In *Advances in Neural Information Processing Systems*, 2018.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998. doi: 10.1016/s0004-3702(98) 00023-x.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, page arXiv:1605.06432, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *28th International Conference on Computational Linguistics*, 2013.

Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. ISBN 2374-3468.

Rahul G. Krishnan, Uri Shalit, and David A. Sontag. Deep kalman filters. *ArXiv*, abs/1511.05121, 2015.

Hanna Kurniawati and Vinay Yadav. *An online POMDP solver for uncertainty planning in dynamic environment*, pages 611–629. Springer, 2016.

J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *International Joint Conference on Artificial Intelligence*, pages 1025–1030, 2003.

Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999. ISSN 0899-7667.

Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011. ISSN 0005-1098.

David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in neural information processing systems*, pages 2164–2172, 2010.

Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973. ISSN 0030-364X.

Trey Smith and Reid Simmons. Heuristic search value iteration for pomdps. In *20th Conference on Uncertainty in Artificial Intelligence*, 2004.

Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. In *Advances in neural information processing systems*, pages 1772–1780, 2013.

Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. ISSN 0001-0782.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, and Petko Georgiev. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. ISSN 1476-4687.