Robust Learning from Demonstration Using Leveraged Gaussian Processes and Sparse-Constrained Optimization

Sungjoon Choi, Kyungjae Lee, and Songhwai Oh

Abstract— **In this paper, we propose a novel method for robust learning from demonstration using leveraged Gaussian process regression. While existing learning from demonstration (LfD) algorithms assume that demonstrations are given from skillful experts, the proposed method alleviates such assumption by allowing demonstrations from casual or novice users. To learn from demonstrations of mixed quality, we present a sparse-constrained leveraged optimization algorithm using proximal linearized minimization. The proposed sparse constrained leverage optimization algorithm is successfully applied to sensory field reconstruction and direct policy learning for planar navigation problems. In experiments, the proposed sparse-constrained method outperforms existing LfD methods.**

I. Introduction

The problem of learning a mapping between the world state and actions, which is often referred to as a policy search, lies at the heart of many robotics applications [1]. As robots are becoming more prevalent in our daily lives, the necessity for more flexible and adaptive robot programming increases significantly. One can naturally expect that non-robotics experts will have opportunities to have interactions with robots, and, in this regard, having abilities for a robot to effectively learn from demonstrations would greatly increase the quality of robotic applications. This procedure of learning from external demonstrations is often referred to as learning from demonstration.

Learning from demonstration (LfD) focuses on the problem of learning a policy from *trajectories* provided by an expert, where *trajectories* are defined as sequences of state-action pairs [1]. This LfD approach differs from exploration-based policy learning methods, such as reinforcement learning (RL) [2] or guided policy learning (GPL) [3], in that the resulting policy is properly defined only in previously encountered states [1]. While RL and GPL have advantages in that they can learn a policy even for previously unvisited states, these methods heavily depends upon the prior knowledge such as an accurate dynamic model. Moreover, exploring unseen state space with a random policy can be dangerous in many robotics applications.

In this paper, we aim to find a policy from external demonstrations within the LfD framework. However, we intentionally omitted the term *experts* in that we do not assume all examples follow the experts' strategy. In particular, we model the expert's policy as a random process, a Gaussian process, and examples are collected from a multiple correlated random processes. Furthermore, by assuming that the number of erroneous examples is sparse, we propose a sparse constrained leverage optimization method based on model selection for Gaussian process regression.

Gaussian process regression is a nonparametric Bayesian regression method and has been widely used to model a policy function within both LfD and RL [4]–[7] due to its high expressiveness. Furthermore, GPR can provide both the prediction at an unseen input and the uncertainty about the prediction, which was used to change the mode between exploitation and exploration in [4].

Most of existing work on LfD assumes that the external demonstrations are trustworthy and dependable [4], [5]. In practice, however, we cannot assume that all demonstrations are beneficial to the learning process as it is often expensive to collect perfectly optimal behaviors. Moreover, within the LfD framework, it is natural to assume that the *proficiency* of each individual demonstrator varies and the quality of each demonstration gets declined as more demonstrators contribute to the data acquisition process.

The main contributions of this paper are twofold. First, we model multiple sources of demonstrations, i.e., demonstrations from *experts* and *novices*, as a correlated random process under the leveraged Gaussian process framework [8]. Second, we present a sparse constrained leverage optimization method by combining a model selection method for Gaussian process regression with proximal linearized minimization, a sparse optimization method [9].

The remainder of this paper is organized as follows. In Section II, a leveraged Gaussian process with a newly proposed smooth leveraged kernel function is presented and the positive semidefiniteness of the kernel function is proven. A sparse constrained leverage optimization method is proposed in Section III. Learning from demonstration experiments and results are given in Section IV.

II. Leveraged Gaussian Process

A leveraged Gaussian process was proposed in [8] to incorporate not only positive training data but also negative training data into a single regression framework. Figure 1 illustrates the regression results of the ordinary Gaussian process regression (GPR) with a squared exponential kernel function and leveraged GPR given five training data. The regression result of the leveraged GPR, shown in the red line, tends to *drift away* from the second training data from

This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2009348) and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.B0101-15-0307, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)).

S. Choi, K. Lee and S. Oh are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 151-744, Korea (e-mail: {sungjoon.choi, kyungjae.lee, songhwai.oh}@cpslab.snu.ac.kr).

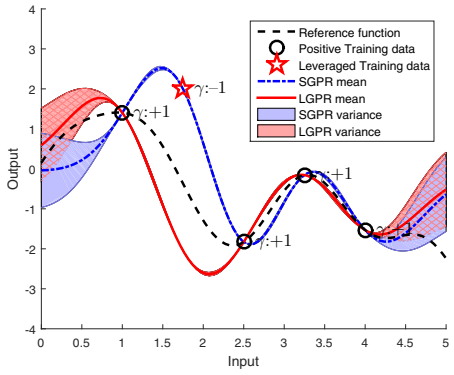


Fig. 1: Stationary GPR (SGPR) and leveraged GRP (LGPR) results. The reference function and 4 sampled training data are shown as a black dashed line and circles, respectively. The leveraged training data is shown as a red star. Mean functions for stationary GPR which treats leveraged training data as a positive training data and leveraged GPR which treats leveraged training data as a negative training data are shown as a blue dotted dash line and a red plane line, respectively

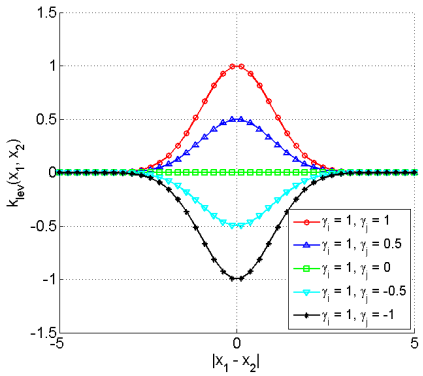


Fig. 2: A leveraged kernel function with different values of γ_i and γ_j.

the left, which works as a negative training example due to its leverage value of -1, while *anchor* to the rest of positive training data.

A. Leveraged Kernel Function

A kernel function is undoubtedly the most crucial ingredient in Gaussian process regression, as it encodes our assumptions about the function which we wish to learn. Intuitively speaking, a kernel function can be interpreted as a non-negative measure defined in the input space which estimates how correlated two function values at different inputs are.

In [8], a leveraged kernel function (1) was proposed by introducing a leverage parameter γ at each training sample, which varies from -1 to $+1$:

$$k_{lev}(\mathbf{x}_i, \mathbf{x}_j) = (1 - |\gamma_i - \gamma_j|)\, k_{SE}(\mathbf{x}_i, \mathbf{x}_j), \qquad (1)$$

where $k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = g^2 \exp\left(-\frac{1}{2}\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{l^2}\right)$, γ_i and γ_j are leverage parameters of the i-th and j-th inputs, respectively, and $\theta = \{g^2, l^2\}$ is a set of hyperparameters of a Gaussian process. For test (unseen) inputs, γ can be set to one. This leverage parameter γ has an important meaning in that if γ is -1, the corresponding training sample works as a *negative* training sample while when γ is $+1$ the corresponding training sample works identical to the ordinary *positive* training sample, which is depicted in Figure 1.

The shape of the leveraged kernel function is illustrated in Figure 2. We can see that between positive examples whose leverages are both $+1$ (and negative examples whose leverages are both -1), the kernel function works as an ordinary squared-exponential kernel function as the first term in (1) becomes 1. However, between positive and negative samples, the correlation decreases as the distance between inputs decreases as the first term in (1) approaches -1. This property makes a Gaussian process with a leveraged kernel function incorporate both positive and negative examples in a single regression framework by assigning the leverage values of positive data to $+1$ and negative data to -1. Furthermore, the training sample with leverage 0 will not affect the regression result at an unseen test input whose leverage is 1.

The positive semidefiniteness of (1) was shown using Bochner's Theorem [10], which states that if an isotropic kernel function has non-negative Fourier transform coefficients, it satisfies positive semidefiniteness [8].

Theorem 1 (Bochner's Theorem): Let f be a bounded continuous function on $\mathbb{R}^d$. Then, f is positive semidefinite if and only if it is the (inverse) Fourier transform of a nonnegative and finite Borel measure μ, i.e.,

$$f(\mathbf{x}) = \int_{\mathbb{R}^d} e^{i\mathbf{w}^T\mathbf{x}} \mu(d\mathbf{w}).$$

Here, we propose a new smooth leveraged (SL) kernel function

$$k_{SL}(\mathbf{x}_i, \mathbf{x}_j) = \cos\left(\frac{\pi}{2}(\gamma_i - \gamma_j)\right) k_{\text{PSD}}(\mathbf{x}_i, \mathbf{x}_j), \qquad (2)$$

where $k_{\text{PSD}}(\mathbf{x}_i, \mathbf{x}_j)$ is any positive semidefinite kernel function, and γ_i and γ_j are leverage parameters of the i-th and j-th inputs, respectively. It is assumed that $\gamma_i \in [-1, 1]$ for all i. In this paper, we use a popular squared exponential (SE) kernel function

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = g^2 \exp\left(-\frac{1}{2}\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{l^2}\right) + w^2 \delta_{\mathbf{x}_i, \mathbf{x}_j} \qquad (3)$$

for $k_{\text{PSD}}(\mathbf{x}_i, \mathbf{x}_j)$, where $\theta = \{g^2, l^2, w^2\}$ is a set of hyperparameters of a Gaussian process and $\delta_{\mathbf{x}_i, \mathbf{x}_j}$ is a Dirac delta function whose output is 1 if $\mathbf{x}_i = \mathbf{x}_j$ and 0 otherwise. First, we show the positive semidefiniteness of the smooth leveraged kernel function (2).

Proposition 1: The smooth leveraged kernel function (2) is a positive semidefinite function.

Proof: It is known that the tensor product of two PSD kernel functions, defined over different spaces satisfies the PSD condition [11]. In other words, if $k(\mathbf{x_1}, \mathbf{x}'_1)$

and $k(\mathbf{x_2}, \mathbf{x_2'})$ satisfy the PSD condition over different spaces $\mathcal{X}_1$ and $\mathcal{X}_2$, then the tensor product $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x_1}, \mathbf{x_1'})k(\mathbf{x_2}, \mathbf{x_2'})$ is also a PSD kernel function on the product space $\mathcal{X}_1 \times \mathcal{X}_2$. These properties can easily be verified using the definition of the PSD [11].

Using this property, (2) can be decomposed into two parts where the first term is

$$k(\gamma_i, \gamma_j) = \cos\left(\frac{\pi}{2}(\gamma_i - \gamma_j)\right) \tag{4}$$

and the second term is any positive semidefinite kernel. Consequently, proving (4) to be PSD with respect to every γ_i and γ_j in $[-1, 1]$ will complete the proof.

Theorem 1 states that showing the non-negativeness of a Fourier transform of an isotropic kernel, i.e. $k(x, x') = k(|x - x'|)$, is equivalent to showing the positive semidefiniteness. Thus, showing the Fourier transform of

$$k(t = \gamma_i - \gamma_j) = \cos\left(\frac{\pi}{2}t\right) \tag{5}$$

is non-negative will complete the proof. The Fourier transform of $\cos\left(\frac{\pi}{2}t\right)$ is $\pi[\delta(w - \frac{1}{2}\pi) + \delta(w + \frac{1}{2}\pi)]$ which is clearly non-negative for all w. Hence, by Theorem 1, the kernel (5) is positive semidefinite. ■

Comparing the smooth leveraged kernel function (2) with the original leveraged kernel function (1), we can see that (2) can be differentiated everywhere in its domain $[-1, 1]$, whereas (1) is not differentiable when $\gamma_i = \gamma_j$. However, it shares the most crucial property of the leveraged kernel function in that training example with $\gamma = -1$ works as a negative training example as $k_{SL}(\mathbf{x}_i, \mathbf{x}_j) = -k_{SE}(\mathbf{x}_i, \mathbf{x}_j)$ when $\gamma_i = 1$ and $\gamma_j = -1$. In other words, if the leverage parameters are all -1 or $+1$, which was the case in [8], leveraged Gaussian processes with (1) and (2) will work identically.

III. Leverage Optimization

In this section, we focus on the problem of finding the leverage of each training sample, which we will refer to as leverage optimization.

A. Problem Formulation

As the leverage of each training sample indicates how reliable each data point is, the procedure of finding the leverage values of the training data can be regarded as selecting a proper subset, or equivalently, selecting an erroneous subset, out of the corrupted training set. While leverage parameters can vary between -1 to 1, we restrict the range of leverage parameters to be within 0 to 1 as training data with 0 leverage will not affect the resulting regressor.

The proposed leverage optimization problem is shown below.

$$\underset{\gamma}{\text{maximize}} \quad L(\mathbf{y}|\,\gamma, \mathbf{X}), \tag{6}$$

where $L(\cdot)$ is the marginal log likelihood of a Gaussian process. $\mathbf{X}$ and $\mathbf{y}$ are collections of n inputs and outputs, respectively, from the training data, and γ is a vector of n leverage parameters. The intuition behind this optimization is that we will treat leverage parameters γ as hyperparameters of a Gaussian process and find the optimal values of leverage parameters using the maximum likelihood estimation (MLE) method

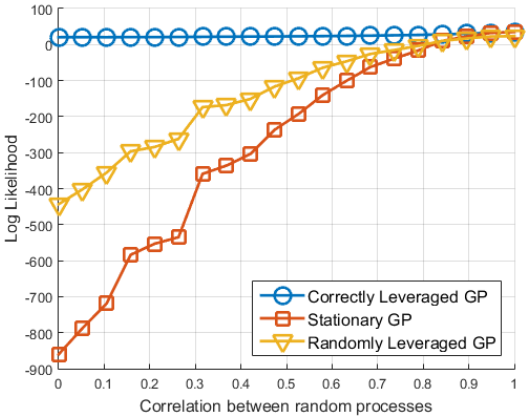


Fig. 3: The marginal log likelihood of a leveraged Gaussian process at different correlations.

First, we show that the maximum likelihood estimation (MLE) is suitable for the leverage optimization. Let us assume that our training data come from two correlated Gaussian processes. Then, we can compute the log likelihood of the training data by assuming that the whole training data came from a single stationary Gaussian process which is shown with red squares in Figure 3, where the x-axis indicates the correlation between two random processes and the y-axis indicates the average log likelihood of 100 independent trials. On the other hand, if we know the correlation between two random processes, we can computed the log likelihood of the whole training data with a leveraged Gaussian process by modifying the leverage parameter of each process. The log likelihood of this setting is shown with blue circles in Figure 3. The yellow inverted triangles indicate the highest log likelihood of 100 randomly chosen leverage parameters. The results clearly tells us that finding an appropriate (or correct) leverage will dramatically increase the overall log likelihood (note the log scale).

However, there is one remaining problem. Unlike hyperparameters of a kernel function which contains only a small number of parameters, the number of leverage parameters equals to the number of training data which makes the correct leverage optimization demanding. In other words, the number of optimization variables is too many compared to the number of training data. In order to handle this problem, we propose a sparse-constrained leveraged optimization method by assuming that the number of outliers or trajectories from novices is sparse compared to inliers or trajectories from experts. Then, the problem can be formulated as the following optimization problem with a sparsity constraint:

$$\begin{aligned} \underset{\gamma}{\text{maximize}} \quad & L(\mathbf{y}|\,\mathbf{X}, \gamma) \\ \text{subject to} \quad & \|\gamma\ - \mathbf{1}_n\|_0 < \delta, \end{aligned} \tag{7}$$

where $\mathbf{X}$ and $\mathbf{y}$ are input and output training data, respectively, $\gamma \in \mathbb{R}^n$ is a leverage vector, $\mathbf{1}_n$ is a column vector filled with 1, δ is the maximum number of non-zero elements,

and $L(\cdot)$ is the marginal log likelihood function.

Due to the l_0-norm constraint, the exact optimization of (7) is demanding, in fact, it is a NP-hard problem. In the following sections, we introduce a sparse-constrained optimization method using proximal linearized minimization (PLM) [9].

B. Derivatives

The derivative of the marginal log-likelihood using the smooth leveraged kernel function (2) can be analytically computed as follows:

$$\frac{\partial \mathcal{L}}{\partial \gamma_k} = \frac{1}{2} y^T K^{-1} \frac{\partial K}{\partial \gamma_k} K^{-1} y - \frac{1}{2} \mathrm{tr}\left(K^{-1} \frac{\partial K}{\partial \gamma_k} \right), \quad (8)$$

where $\frac{\partial K}{\partial \gamma_k}$ is a derivative of kernel matrix K with respect to γ_k whose (i,j) element is

$$\left[\frac{\partial K}{\partial \gamma_k} \right]_{ij} = \frac{\partial k(\mathbf{x}_i, \mathbf{x}_j, \gamma_i, \gamma_j)}{\partial \gamma_k} = \begin{cases} -\frac{\pi}{2} \sin(\frac{\pi}{2}(\gamma_k - \gamma_j)) k_{SE}(\mathbf{x}_i, \mathbf{x}_j) & \text{if } i = k, \\ \frac{\pi}{2} \sin(\frac{\pi}{2}(\gamma_i - \gamma_k)) k_{SE}(\mathbf{x}_i, \mathbf{x}_j) & \text{if } j = k, \\ 0 & \text{otherwise.} \end{cases}$$

C. Proximal Linearized Minimization (PLM)

Directly handling the l_0-norm constraint in (7) is practically intractable as it is non-convex and non-differentiable. To make the optimization problem computationally tractable, we first relax the non-convex l_0-norm constraint in (7) using the l_1-norm constraint as follows:

$$\begin{aligned} \underset{\gamma}{\text{maximize}} \quad & L(\mathbf{y}|\,\mathbf{X}, \gamma) \\ \text{subject to} \quad & \|\gamma \ - \mathbf{1}_n\|_1 < \delta, \end{aligned} \quad (9)$$

where δ ensures the level of sparsity. Then, we can make (9) to an unconstrained optimization problem as follows:

$$\underset{\gamma}{\text{minimize}} \quad - L(\mathbf{y}|\mathbf{X}, \bar{\gamma} + \mathbf{1}_n) + \lambda \|\bar{\gamma}\|_1, \quad (10)$$

where $\bar{\gamma} = \gamma \ - \mathbf{1}_n$ and λ is a sparsity coefficient.

The unconstrained optimization problem (10) can be solved via proximal linearized minimization (PLM) as the log-likelihood term $L(\cdot)$ is differentiable with respect to $\bar{\gamma}$ as shown in Section III-B and $\|\cdot\|_1$ is convex whose proximal mapping can be computed inexpensively [9] using a soft-thresholding function, which is defined as

$$\mathrm{prox}^{\lambda\|\gamma - 1\|_1}(\gamma) = \begin{cases} \gamma - \lambda & \text{if } \gamma > 1 + \lambda, \\ \gamma + \lambda & \text{if } \gamma < 1 - \lambda, \\ 1 & \text{otherwise.} \end{cases}$$

Using PLM, the update rule for solving (9) is

$$\bar{\gamma}^{k+1} \in \mathrm{prox}_{1/t}^{\lambda\|\bar{\gamma}\|_1} \left(\bar{\gamma}^k - t \nabla_{\bar{\gamma}} \left(-L(\mathbf{y}|\,\mathbf{X}, \bar{\gamma}^k + \mathbf{1}_n) \right) \right), \quad (11)$$

where $\mathrm{prox}(\cdot)$ is a proximal mapping and t is a step size.

IV. Experiments

In this section, we validate the performance of the leveraged optimization with two different experiments: sensor field reconstruction and robust policy learning for a planar navigation problem.

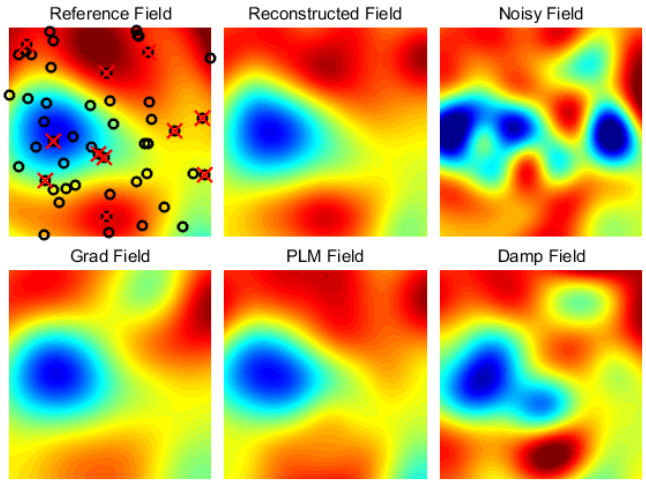


Fig. 4: Sensory field reconstruction. (Reference Field) A reference field. The locations of inliers and outliers are shown in black circles and red crosses, respectively. (Reconstructed Field) A field reconstructed using only inliers. (Noisy Field) A field reconstructed using both inliers and outliers. (Grad Field) A field reconstructed by leverage optimization without constraints. (PLM Field) A field reconstructed by leverage optimization with the l_1-norm sparse constraint using PLM. (Damp Field) A field reconstructed by leverage optimization with the l_2-norm sparse constraint.

A. Sensory Field Reconstruction

We applied the sparse constrained leverage optimization method using proximal linearized minimization (PLM) to a regression problem of reconstructing the sensory field. Specifically, we assume that we are given sensory measurements, locations and sensor values, in a two dimensional plane, where a small portion of measurements (outliers) came from different sensory fields. In other words, we do not assume specific probability model for outliers. In all experiments, the soft threshold value for a proximal linearized minimization (PLM) is fixed. We would like to note that one can regard locations and sensory values as 2-dimensional states and 1-dimensional control outputs, respectively, which will make the sensory field reconstruction as a policy learning process.

The top left figure in Figure 4 indicates the reference sensory field, the original field we aim to reconstruct. The correctly measured locations (inliers) and outliers are depicted with black circles and red crosses, respectively. The top middle and top right figures show the reconstructed field using Gaussian process regression with only inliers (middle) and both inliers and outliers (right). The bottom left, middle, and right figures illustrate reconstructed fields from the leveraged Gaussian processes, where the leverages are computed with l_2-regularized leverage optimization (right), l_1-regularized leverage optimization (middle), and leverage optimization without regularization (left). We can clearly see that the reconstruction performance of the proposed sparse-constrained leverage optimization outperforms the compared methods.

In particular, the optimization without regularization tends to over-optimize the leverage parameters which leads the regression result to be overly smooth as shown in the bottom left figure of Figure 4. On the other hand, l_2-regularized leverage optimization tends not to reduce the leverage of outliers as shown in the bottom right figure of Figure 4.

The quantitative results are shown in Figure 5. The average reconstruction errors of three leverage optimization methods at different outlier rates are depicted in Figure 5(a), where the averages are computed from 150 independent runs. In all cases, the sparse-constrained leverage optimization outperformed other optimization methods. This is mainly due to the fact that the sparse constrained leverage optimization was able to correctly reduce leverage parameters of outliers without reducing the leverages of inliers as illustrated in Figure 5(b) and 5(c). The average sparsity level of each optimization method is shown in Figure 5(b). Even though we do not adjust the soft threshold value for each outlier rate, the leverage optimization with PLM was able to automatically increase the sparsity level as the outlier rate increases which leads to superior performance in detecting the outliers as shown in Figure 5(c).

B. Robust Policy Learning

In this experiment, we validate that the proposed sparse-constrained leverage optimization can effectively be used in the learning from demonstration framework when trajectories with mixed qualities are given. The experimental setting is as follows: first, we randomly generate a 2-dimensional reward map using a Gaussian process realization. From the reward map, optimal trajectories with random initial points are collected using an exhaustive search by discretizing the control space of directional and angular velocities where an agent is assumed to follow the unicycle dynamics. This setting is similar to planar navigation experiments described in [12].

In particular, each trajectory consists of 5 state-action pairs and a total of 50 trajectories are collected. The training data for robust learning from demonstration consist of both optimal trajectories and a small portion of outlier trajectories, where the outliers are collected from optimal trajectories of *different* reward maps. We fix the total number of training data to 50 and vary the outlier ratio to test the performance of robust learning from demonstration.

The proximal linearized minimization method is again used to solve the sparse-constrained optimization problem. We compared the proposed robust learning from demonstration with the sparse-constrained leverage optimization method (RLfD-l_1) against the proposed method with two different leverage optimization methods, a gradient based method with no regularization (RLfD-No) and l_2-norm regularized optimization method (RLfD-l_2). Two existing learning from demonstration methods, a Gaussian process (LfD-GP) [6] and k-nearest neighborhood algorithm (LfD-kNN) [13], are used as baseline methods. The robust Gaussian process regression method (LfD-RGP) [14] is also compared to the proposed method. Furthermore, LfD using Gaussian process regression with optimal trajectories (LfD-Ref) is presented as a reference. Excluding LfD-Ref, for all methods, trajectories with mixed qualities are given as training data.

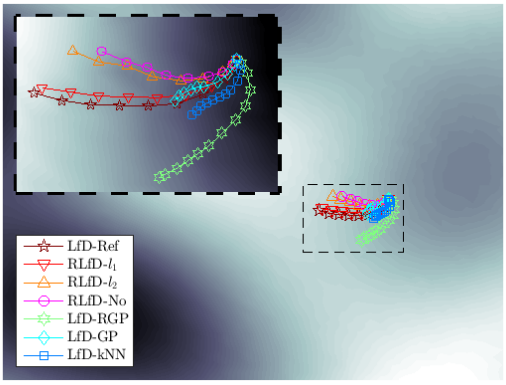


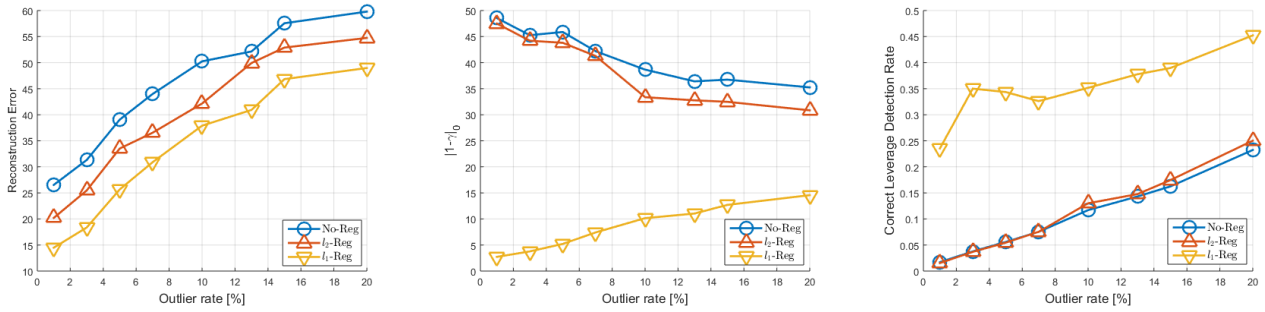
Fig. 7: A reward map and resulting trajectories of learned policies from different learning from demonstration algorithms. An enlarged view of trajectories are shown in the top-right corner.

The expected reward of each policy is shown in Figure 6(a), where the expected reward is computed as the average of discounted sum of rewards starting from 100 random locations in the reward map, following the policy function learned from each LfD method. When the outlier rate is small, expected rewards computed by different algorithms do not show significant differences. However, even with a moderate outlier rate of 5%, the propose robust learning from demonstration with sparse constrained leverage optimization (RLfD-l_1) outperforms other LfD methods. The sparsity level and outlier detection ratio are shown in Figure 6(b) and 6(c), where the proposed sparse constrained optimization shows the highest outlier detection rate. However, the performance gap of the outlier detection rate between the l_1-norm constrained method and others is reduced, compared to the field reconstruction problem as shown in Figure 5(c). This is mainly because the change in the field does not directly change the resulting trajectories. In other words, two optimal trajectories sampled from completely different fields could behave similarly in certain regions and this increases the ambiguity of finding correct leverage values of each trajectory.

Figure 7 depicts a reward map and resulting trajectories of learned policies from different algorithms, where the trajectory from the proposed method (RLfD-l_1) shows the most similar behavior to that of the policy learned from optimal trajectories (LfD-Ref).

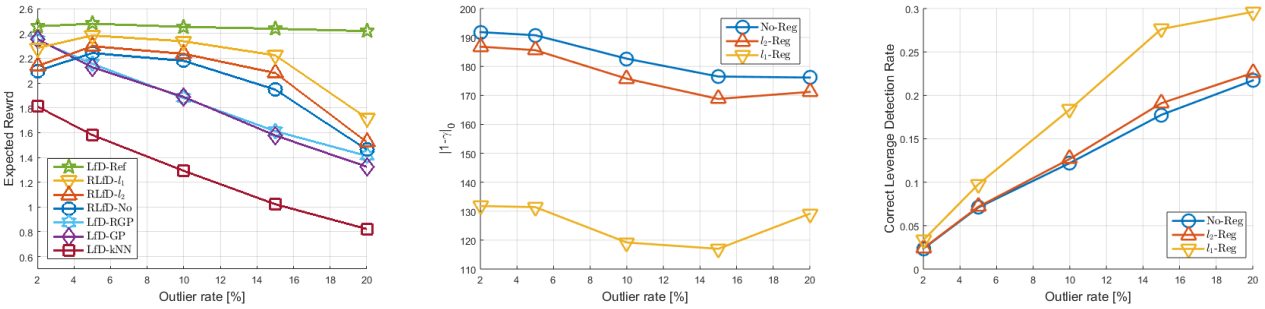
V. Conclusion

The robust learning from demonstration algorithm using a sparse constrained leverage optimization method is proposed in this paper. While many existing learning from demonstration (LfD) algorithms assume that demonstrations are given from skillful experts, such assumption is alleviated in the proposed method by allowing demonstrations to contain a small portion of demonstrations from casual or novice users.



(a) (b) (c)

Fig. 5: (a) Reconstruction errors at different outlier levels of three leverage optimization methods: leverage optimization without regularization (No-Reg), with l_2-regularization (l_2-Reg), and l_1-regularization (l_1-Reg). (b) Sparsity levels of different leverage optimization methods. (c) Outlier detection ratio of different leverage optimization methods.



(a) (b) (c)

Fig. 6: (a) Expected rewards of policies learned by different learning from demonstration algorithms: the proposed robust learning from demonstration with sparse-constrained leverage optimization method (RLfD-1_1), leverage optimization without regularization (RLfD-No), leverage optimization with l_2-norm regularization (RLfD-l_2), two existing LfD methods using a Gaussian process (LfD-GP) and kNN algorithm (LfD-kNN), robust Gaussian processes (LfD-RGP), and LfD using Gaussian process regression with optimal trajectories (LfD-Ref). (b) Sparsity levels of different leverage optimization methods. (c) Outlier detection ratios of different leverage optimization methods.

In particular, we present a sparse-constrained leveraged optimization method using proximal linearized minimization. The proposed robust learning from demonstration methods are successfully applied a sensory field reconstruction problem and a direct policy learning problem. In all experiments, the proposed sparse-constrained method has outperformed existing LfD methods.

REFERENCES

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[3] S. Levine and V. Koltun, "Guided policy search," in *Proc. of the International Conference of Machine Learning (ICML)*, 2013.

[4] K. Gräve, J. Stückler, and S. Behnke, "Learning motion skills from expert demonstrations and own experience using Gaussian process regression," in *International Symposium on Robotics and German Conference on Robotics*, 2010.

[5] S. P. Chatzis, D. Korkinof, and Y. Demiris, "A nonparametric bayesian approach toward robot learning by demonstration," *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 789–802, 2012.

[6] M. Schneider, "Learning from demonstration with Gaussian processes," Ph.D. dissertation, Master-thesis, University of Applied Sciences Ravensburg-Weingarten, 2009. 1.1, 1.2. 4, B, 2009.

[7] S. Choi, E. Kim, and S. Oh, "Real-time navigation in crowded dynamic environments using Gaussian process motion control," in *Proc. of the International Conference of Robotics and Automation (ICRA)*, 2014.

[8] S. Choi, E. Kim, K. Lee, and S. Oh, "Leveraged non-stationary Gaussian process regression for autonomous robot navigation," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

[9] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[10] S. Bochner, *Harmonic analysis and the theory of probability*. Courier Dover Publications, 2012.

[11] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 1.

[12] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proc. of the International Conference of Machine Learning (ICML)*, 2012.

[13] D. C. Bentivegna and C. G. Atkeson, "Learning from observation using primitives," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2001.

[14] E. Kim, S. Choi, and S. Oh, "A robust autoregressive Gaussian process motion model using l1-norm based low-rank kernel matrix approximation," in *Proc. of the International Conference of Intelligent Robots and Systems (IROS)*. IEEE, 2014.