SUPPLEMENTARY MATERIAL

Nonlinear Inverse Reinforcement Learning with Gaussian Processes

Anonymous Author(s)
Affiliation
Address
email

This supplement presents a derivation of the partial derivatives of the GPIRL log likelihood in Equation 4 of the paper with respect to each variable in the optimization, as well as additional details regarding the warped kernel function in Section 6 and a restart technique that was used to avoid local optima. We decompose the log likelihood into three parts:

$$\log P(\mathcal{D}, \mathbf{u}, \boldsymbol{\theta}|\mathbf{X_u}) = \overbrace{\underbrace{\log P(\mathcal{D}|\mathbf{r} = \mathbf{K}_{\mathbf{r},\mathbf{u}}^{\mathrm{T}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u})}_{\text{IRL log likelihood}}}^{\mathcal{L}_\mathcal{D}} + \overbrace{\underbrace{\log P(\mathbf{u}|\boldsymbol{\theta}, \mathbf{X_u})}_{\text{GP marginal}}}^{\mathcal{L}_\mathcal{G}} + \overbrace{\underbrace{\log P(\boldsymbol{\theta}|\mathbf{X_u})}_{\text{hyperparameter prior}}}^{\mathcal{L}_\mathcal{H}}$$

The terms $\mathcal{L}_\mathcal{G}$ and $\mathcal{L}_\mathcal{H}$ are denoted in Equation 4 by the expression "$\log P(\mathbf{u}, \boldsymbol{\theta}|\mathbf{X_u})$." In Appendix A, we derive the partial derivatives of $\mathcal{L}_\mathcal{D}$ with respect to the reward function $\mathbf{r}$, while in Appendix B we derive the partial derivatives of $\mathcal{L}_\mathcal{G}$ and $\mathbf{r}$ with respect to the hyperparameters $\boldsymbol{\theta}$ and the inducing points $\mathbf{u}$. Appendix C derives the partial derivatives for the hyperparameter priors $\mathcal{L}_\mathcal{H}$. Appendix D derives the derivatives of the warped kernel function discussed in Section 6 of the paper and describes the priors used on each warp parameter. Finally, Appendix E describes a simple restart procedure we used to avoid local optima, which is particularly useful when using the warped kernel.

A Derivatives of the IRL Log Likelihood

To find the derivatives of the IRL log likelihood in Equation 1 of the paper, we first rewrite the IRL log likelihood in terms of only the reward and value functions:

$$\begin{aligned}
\mathcal{L}_\mathcal{D} &= \sum_i \sum_t \left(\mathbf{Q}^{\mathbf{r}}_{s_{i,t}a_{i,t}} - \mathbf{V}^{\mathbf{r}}_{s_{i,t}} \right) \\
&= \sum_i \sum_t \left(\mathbf{r}_{s_{i,t}a_{i,t}} - \mathbf{V}^{\mathbf{r}}_{s_{i,t}} + \sum_{s'} \gamma \mathcal{T}_{s'}^{s_{i,t}a_{i,t}} \mathbf{V}^{\mathbf{r}}_{s'} \right)
\end{aligned}$$

Differentiating with respect to the reward, we obtain the following:

$$\frac{\partial \mathcal{L}_\mathcal{D}}{\partial \mathbf{r}} = \sum_i \sum_t \frac{\partial \mathbf{r}_{s_{i,t}a_{i,t}}}{\partial \mathbf{r}} - \sum_i \sum_t \frac{\partial \mathbf{V}^{\mathbf{r}}_{s_{i,t}}}{\partial \mathbf{r}} + \sum_i \sum_t \sum_{s'} \gamma \mathcal{T}_{s'}^{s_{i,t}a_{i,t}} \frac{\partial \mathbf{V}^{\mathbf{r}}_{s'}}{\partial \mathbf{r}}$$

The first term in the summation is simply the empirical visitation count of each state-action pair, denoted $\hat{\mu}$ and given by $\hat{\mu}_{sa} = \sum_i \sum_t 1_{s_{i,t}=s \wedge a_{i,t}=a}$. From [2], we further have that $\frac{\partial \mathbf{V}^{\mathbf{r}}_s}{\partial \mathbf{r}} = E\left[\mu|s\right]$,

the expected visitation count of each state-action pair when starting from state s and following the optimal stochastic policy. The partial derivatives of $\mathcal{L}_\mathcal{D}$ are then given by

$$\begin{aligned}\frac{\partial \mathcal{L}_\mathcal{D}}{\partial \mathbf{r}} &= \hat{\mu} - \sum_i \sum_t E\left[\mu | s_{i,t}\right] + \sum_i \sum_t \sum_{s'} \gamma \mathcal{T}_{s'}^{s_{i,t} a_{i,t}} E\left[\mu | s'\right] \\ &= \hat{\mu} - \sum_s \hat{\nu}_s E\left[\mu | s\right]\end{aligned}$$

where $\hat{\nu}_s = \sum_a \hat{\mu}_{sa} - \sum_i \sum_t \gamma \mathcal{T}_s^{s_{i,t} a_{i,t}}$. We can compute $\tilde{\mu} = \sum_s \hat{\nu}_s E\left[\mu | s\right]$ efficiently for any vector $\hat{\nu}$ using a simple iterative algorithm described in [2]:

Algorithm 1 Iterative estimation of $\tilde{\mu} = \sum_s \hat{\nu}_s E\left[\mu | s\right]$

```
μ̃ ← 0
while not converged do
    for all s' ∈ S and a' ∈ A do
        μ̃'_{s'a'} ← π(a'|s') [ν̂_{s'} + Σ_s Σ_a γ T_{s'}^{sa} μ̃_{sa}]
    end for
    μ̃ ← μ̃'
end while
```

Intuitively, the algorithm repeatedly updates the state-action visitation frequencies $\tilde{\mu}$ by distributing the current probability mass to successor states according to the transition function $\mathcal{T}$, and then distributing the mass in each state into actions according to the optimal stochastic policy π, given by $\pi(s|a) = \exp\left(\mathbf{Q}_{sa}^\mathbf{r} - \mathbf{V}_s^\mathbf{r}\right)$. The value function $\mathbf{V}^\mathbf{r}$ is obtained by repeatedly applying the modified Bellman backup operator, given by

$$\mathbf{V}_s^\mathbf{r} = \log \sum_{a \in \mathcal{A}} \exp\left(\mathbf{r}_s + \gamma \sum_{s'} \mathcal{T}_{s'}^{sa} \mathbf{V}_{s'}^\mathbf{r}\right)$$

The final derivative is given by $\frac{\partial \mathcal{L}_\mathcal{D}}{\partial \mathbf{r}} = \hat{\mu} - \tilde{\mu}$.

B Derivatives of the GP Marginal Likelihood

The GP marginal likelihood, given in Equation 3 in the paper, consists of the fitting term $-\frac{1}{2}\mathbf{u}^\mathrm{T}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$ and the normalizing term $-\frac{1}{2}\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|$. To obtain partial derivatives with respect to a particular hyperparameter $\boldsymbol{\theta}_j$, we follow Rasmussen and Williams [1]:

$$\begin{aligned}\frac{\partial \mathcal{L}_\mathcal{G}}{\partial \boldsymbol{\theta}_j} &= \frac{1}{2}\mathbf{u}^\mathrm{T}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \boldsymbol{\theta}_j}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \boldsymbol{\theta}_j}\right) \\ &= \frac{1}{2}\mathrm{tr}\left((\alpha\alpha^\mathrm{T} - \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1})\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \boldsymbol{\theta}_j}\right)\end{aligned}$$

where $\alpha = \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$. The partial derivatives with respect to the inducing points $\mathbf{u}$ are simply given by $\frac{\partial \mathcal{L}_\mathcal{G}}{\partial \mathbf{u}} = -\alpha$. We must also compute the contribution of the IRL term $\mathcal{L}_\mathcal{D}$ to the gradient with respect to $\boldsymbol{\theta}$ and $\mathbf{u}$. Since we have the gradient of $\mathcal{L}_\mathcal{D}$ with respect to $\mathbf{r}$, it remains to compute the partial derivatives of $\mathbf{r}$, given $\mathbf{r} = \mathbf{K}_{\mathbf{r},\mathbf{u}}^\mathrm{T}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$:

$$\begin{aligned}\frac{\partial \mathbf{r}}{\partial \boldsymbol{\theta}_j} &= \frac{\partial \mathbf{K}_{\mathbf{r},\mathbf{u}}^\mathrm{T}}{\partial \boldsymbol{\theta}_j}\alpha - \mathbf{K}_{\mathbf{r},\mathbf{u}}^\mathrm{T}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \boldsymbol{\theta}_j}\alpha \\ \frac{\partial \mathbf{r}}{\partial \mathbf{u}} &= \mathbf{K}_{\mathbf{r},\mathbf{u}}^\mathrm{T}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\end{aligned}$$

We must also compute the partial derivatives of the kernel matrices $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ and $\mathbf{K}_{\mathbf{r},\mathbf{u}}$ with respect to the hyperparameters $\boldsymbol{\theta}$ by differentiating the regularized kernel function in Equation 5 of the paper:

$$\frac{\partial k}{\partial \beta}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\beta} k(\mathbf{x}_i, \mathbf{x}_j)$$
$$\frac{\partial k}{\partial \lambda_k}(\mathbf{x}_i, \mathbf{x}_j) = \left(-\frac{1}{2}(x_{ik} - x_{ij})^2 - 1_{i \neq j}\sigma^2 \right) k(\mathbf{x}_i, \mathbf{x}_j)$$

where λ_k denotes the k^{th} diagonal entry in $\boldsymbol{\Lambda}$. The final gradient of the log likelihood is

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log P(\mathcal{D}|\mathbf{u}, \boldsymbol{\theta}, \mathbf{X}_\mathbf{u}) = \frac{\partial \mathcal{L}_\mathcal{D}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathcal{L}_\mathcal{G}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathcal{L}_\mathcal{H}}{\partial \boldsymbol{\theta}}$$
$$\frac{\partial}{\partial \mathbf{u}} \log P(\mathcal{D}|\mathbf{u}, \boldsymbol{\theta}, \mathbf{X}_\mathbf{u}) = \frac{\partial \mathcal{L}_\mathcal{D}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{u}} + \frac{\partial \mathcal{L}_\mathcal{G}}{\partial \mathbf{u}}$$

The prior gradients $\frac{\partial \mathcal{L}_\mathcal{H}}{\partial \boldsymbol{\theta}}$ are given in the next section.

C Derivatives of Hyperparameter Priors

The general hyperparameter prior described in Section 4 of the paper consists of the inverse covariance term $-\frac{1}{2}\text{tr}(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2})$ and the sparsity-inducing penalty $\phi(\boldsymbol{\Lambda}) = \sum_i \log(\boldsymbol{\Lambda}_{ii} + 1)$. The partial derivatives of the covariance term are given by

$$\frac{\partial}{\partial \boldsymbol{\theta}_j}\left[-\frac{1}{2}\text{tr}(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}) \right] = \text{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2} \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \boldsymbol{\theta}_j} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \right) = \text{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-3} \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \boldsymbol{\theta}_j} \right)$$

The derivatives of the penalty term with respect to the diagonal entries of $\boldsymbol{\Lambda}$ are simply

$$\frac{\partial \phi}{\partial \lambda_k} = \frac{1}{\sum_i \boldsymbol{\Lambda}_{ii} + 1}$$

The partial derivatives of the log prior with respect to the hyperparameters are therefore given by

$$\frac{\partial \mathcal{L}_\mathcal{H}}{\partial \beta} = \text{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-3} \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \beta} \right)$$
$$\frac{\partial \mathcal{L}_\mathcal{H}}{\partial \lambda_k} = \text{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-3} \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_k} \right) - \frac{1}{\sum_i \boldsymbol{\Lambda}_{ii} + 1}$$

D Details of the Warped Kernel Function

Described in Section 6 of the paper, the warped kernel is an alternative kernel function that can be used to learn piecewise constant rewards. The warped kernel function transforms the feature coordinates according to a parameterized sigmoid, centered at $\mathbf{m}$ and scaled by $\boldsymbol{\ell}$. The warp also affects the contribution of noise to the expected distance between two points along each coordinate k, so that it is no longer given by $2\sigma^2$, but instead approximated to first order by $\sigma^2(w_k^\sigma(x_{ik}) + w_k^\sigma(x_{jk}))$, where $w_k^\sigma(x_{ik}) = \frac{\partial w_k}{\partial x_{ik}} + s_k$ and s_k is a learned parameter that prevents degeneracies in the "tails" of the sigmoid. To derive w_k^σ, we differentiate w_k with respect to x_{ik}:

$$w_k^\sigma(x_{ik}) = \frac{\partial w_k}{\partial x_{ik}} + s_k = \left(\frac{1}{\ell_k} \right) \left(\frac{1}{e^{z_k} + 2 + e^{-z_k}} \right) + s_k$$

where $z_k = -\frac{x_{ik}-m_k}{\ell_k}$. To optimize the warp parameters, we add them to $\boldsymbol{\theta}$. During optimization, we must compute the partial derivatives of the kernel function with respect to each hyperpameter. For each of the warp parameters $p_k \in \{m_k, \ell_k, s_k\}$, these partial derivatives are given by

$$\frac{\partial k}{\partial p_k}(\mathbf{x}_i, \mathbf{x}_j) = -\frac{1}{2}\lambda_k \left[(w_k(x_{ik}) - w_k(x_{jk})) \left(\frac{\partial w_k}{\partial p_k}(x_{ik}) - \frac{\partial w_k}{\partial p_k}(x_{jk}) \right) + 1_{i \neq j} \sigma^2 \left(\frac{\partial w_k^\sigma}{\partial p_k}(x_{ik}) + \frac{\partial w_k^\sigma}{\partial p_k}(x_{jk}) \right) \right] k(\mathbf{x}_i, \mathbf{x}_j)$$

The derivatives of w_k and w_k^σ can be computed by first defining two intermediate variables:

$$h = \frac{1}{e^{z_k} + 2 + e^{-z_k}} \qquad g = \frac{2}{e^{2z_k} + 3e^{z_k} + 3 + e^{-z_k}}$$

Note that, even as $\ell_k \to 0$, h and g remain numerically stable. The derivatives are now given by

$$\frac{\partial w_k}{\partial m_k}(x_{ik}) = -\frac{h}{\ell_k} \qquad \frac{\partial w_k}{\partial \ell_k}(x_{ik}) = \frac{h z_k}{\ell_k} \qquad \frac{\partial w_k}{\partial s_k}(x_{ik}) = 0$$
$$\frac{\partial w_k^\sigma}{\partial m_k}(x_{ik}) = \frac{h - g}{\ell_k^2} \qquad \frac{\partial w_k^\sigma}{\partial \ell_k}(x_{ik}) = \frac{g z_k + h(z_k - 1)}{\ell_k^2} \qquad \frac{\partial w_k^\sigma}{\partial s_k}(x_{ik}) = 1$$

We also place a prior on each warp parameter. Since the features in our examples are positive, we use a gamma prior on $\mathbf{m}$ with shape parameter $a = 2$ and scale $b = 2$. To encourage sharp, narrow sigmoids, we use unit variance Gaussian priors on $\boldsymbol{\ell}$ and $\mathbf{s}$. The derivatives are given by

$$\frac{\partial \mathcal{L}_\mathcal{H}}{\partial s_k} = \text{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-3} \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial s_k} \right) - s_k$$
$$\frac{\partial \mathcal{L}_\mathcal{H}}{\partial \ell_k} = \text{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-3} \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \ell_k} \right) - \ell_k$$
$$\frac{\partial \mathcal{L}_\mathcal{H}}{\partial m_k} = \text{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-3} \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial m_k} \right) + \frac{a-1}{m_k} - \frac{1}{b}$$

where the first term in each sum is the derivative of the inverse covariance term from Appendix C.

E Avoiding Local Optima

The standard regularized kernel presented in Equation 5 of the paper is often able to produce a good solution with a single run of the L-BFGS optimization procedure, though a few random restarts can provide minor improvement. The warped kernel is more susceptible to local optima, and random restarts are necessary to obtain good results on more complex examples. First, we run each optimization 5 times, each time with a random initial setting for $\mathbf{u}$. When using the warped kernel, we then perform 5 more restarts, where $\mathbf{u}$ is initialized to the final value of the best run so far (the one with the highest likelihood) and the sigmoid centers $\mathbf{m}$ are resampled at random from their prior gamma distribution, while all other parameters are reset to their initial values. With the standard kernel, only one such restart is used, since there are no sigmoid centers.

References

[1] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[2] B. D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.