Variational Inference for Inverse Reinforcement Learning with Gaussian Processes

Paulius Dilkas

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Masters project proposal

Date of submission placed here

Contents

1 Introduction

Inverse reinforcement learning (IRL)—a problem proposed by Russell in 1998 [57]—asks us to find a reward function for a Markov decision process that best explains a set of given demonstrations. IRL is important because reward functions can be hard to define manually [1, 3], and rewards are not entirely specific to a given environment, allowing one to reuse the same reward structure in previously unseen environments [3, 30, 35]. Moreover, IRL has seen a wide array of applications in autonomous vehicle control [31, 33] and learning to predict another agent's behaviour [9, 70, 77, 78, 79]. Most approaches in the literature (see Section 3) make a convenient yet unjustified assumption that the reward function can be expressed as a linear combination of features. One proven way to abandon this assumption is by representing the reward function as a Gaussian process [30, 35, 48].

2 Statement of the Problem

Definition 1. A *Markov decision process* (MDP) is a set $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, r\}$, where $\mathcal{S}$ and $\mathcal{A}$ are sets of states and actions, respectively; $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a function defined so that $\mathcal{T}(s, a, s')$ is the probability of moving to state s' after taking action a in state s; $\gamma \in [0, 1)$ is the discount factor (with higher γ values, it makes little difference whether a reward is received now or later, while with lower γ values the future becomes gradually less and less important); and $r : \mathcal{S} \to \mathbb{R}$ is the reward function.

In *inverse reinforcement learning*, one is presented with an MDP without a reward function $\mathcal{M} \setminus \{r\}$ and a set of expert demonstrations $\mathcal{D} = \{\zeta_i\}_{i=1}^{N}$, where each demonstration $\zeta_i = \{(s_{i,0}, a_{i,0}), \ldots, (s_{i,T}, a_{i,T})\}$ is a multiset of state-action pairs representing the actions taken by the expert during a particular recorded session. Each state is also characterised by a number of features. The goal of IRL is then to find r such that the optimal policy under r

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \middle| \pi\right]$$

matches the actions in $\mathcal{D}$.

Following previous work on GP IRL [35, 30], we use a maximum entropy IRL model [77], under which we have that

$$P(a|s) \propto \exp(Q_r(s, a)),$$

where

$$Q_r(s, a) = r(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') V_r(s'), \tag{1}$$

and $V_r(s)$ is a "soft" version of the Bellman backup operator, which can be obtained by repeatedly applying the following equation until convergence: [35, 36]

$$V_r(s) = \log \sum_{a \in \mathcal{A}} \exp\left(r(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') V_r(s')\right).$$

The likelihood of the data can then be written down as [30, 35]

$$p(\mathcal{D}|r) = \prod_{i=1}^{N}\prod_{t=1}^{T} p(a_{i,t}|s_{i,t}) = \exp\left(\sum_{i=1}^{N}\sum_{t=1}^{T} Q_r(s_{i,t}, a_{i,t}) - V_r(s_{i,t})\right). \tag{2}$$

However, a reward function learned by maximising this likelihood is not transferable to new situations [30, 35]. One needs to model the reward structure in a way that would allow reward predictions for previously unseen states.

One way to model rewards without assumptions of linearity is with a *Gaussian process* (GP). A GP is a collection of random variables, any finite combination of which has a joint Gaussian distribution [54]. We write $r \sim \mathcal{GP}(0, k_{\boldsymbol{\lambda}})$ to say that r is a GP with mean 0 and covariance function $k_{\boldsymbol{\lambda}}$, which uses a vector of hyperparameters $\boldsymbol{\lambda}$. Covariance functions take two state feature vectors as input and quantify how similar the two states are, in a sense that we would expect them to have similar rewards.

As training a GP with n data points has a time complexity of $\mathcal{O}(n^3)$ [54], numerous approximation methods have been suggested, many of which select a subset of data called *inducing points* and focus most of the training effort on them [39]. Let $\mathbf{X_u}$ be the matrix of features at inducing states, $\mathbf{u}$ the rewards at those states, and $\mathbf{r}$ a vector with $r(\mathcal{S})$ as elements. Then the full joint probability distribution can be factorised as

$$p(\mathcal{D}, \boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}, \mathbf{r}) = p(\mathbf{X_u}) \times p(\boldsymbol{\lambda}|\mathbf{X_u}) \times p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u}) \times p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) \times p(\mathcal{D}|r). \tag{3}$$

Here $p(\mathbf{X_u})$ and $p(\boldsymbol{\lambda}|\mathbf{X_u})$ are customisable priors,

$$\begin{aligned} p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u}) &= \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{K_{u,u}}) \\ &= \frac{1}{(2\pi)^{m/2}|\mathbf{K_{u,u}}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{u}^\intercal \mathbf{K_{u,u}^{-1}}\mathbf{u}\right) \\ &= \exp\left(-\frac{1}{2}\mathbf{u}^\intercal \mathbf{K_{u,u}^{-1}}\mathbf{u} - \frac{1}{2}\log|\mathbf{K_{u,u}}| - \frac{m}{2}\log 2\pi\right) \end{aligned} \tag{4}$$

is the GP prior [54], where $m \in \mathbb{N}$ is the number of inducing points. The GP posterior is a multivariate Gaussian [35]

$$p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) = \mathcal{N}(\mathbf{r}; \mathbf{K_{r,u}^\intercal K_{u,u}^{-1}}\mathbf{u}, \mathbf{K_{r,r}} - \mathbf{K_{r,u}^\intercal K_{u,u}^{-1}}\mathbf{K_{r,u}}), \tag{5}$$

and $p(\mathcal{D}|r)$ is as in (2). The matrices such as $\mathbf{K_{r,u}}$ are called *covariance matrices* and are defined as $[\mathbf{K_{r,u}}]_{i,j} = k_{\boldsymbol{\lambda}}(\mathbf{x_{r},}_i, \mathbf{x_{u},}_j)$, where $\mathbf{x_{r},}_i$ and $\mathbf{x_{u},}_j$ denote feature vectors for the ith state in $\mathcal{S}$ and the jth state in $\mathbf{X_u}$, respectively [30].

Given this model, data $\mathcal{D}$, and inducing feature matrix $\mathbf{X_u}$, our goal is then to find optimal values of hyperparameters $\boldsymbol{\lambda}$, inducing rewards $\mathbf{u}$, and the rewards for all relevant states $\mathbf{r}$. While the previous paper to consider this IRL model computed maximum likelihood estimates for $\boldsymbol{\lambda}$ and $\mathbf{u}$, and made an assumption that $\mathbf{r}$ in (5) has zero variance [35], we aim to avoid this assumption and use variational inference to approximate the full posterior distribution $p(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}|\mathcal{D}, \mathbf{X_u})$. *Variational inference* (VI) is an approximation technique

for probability densities [8]. Let $q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r})$ be our approximating family of probability distributions for $p(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}|\mathcal{D}, \mathbf{X_u})$ with its own hyperparameter vector $\boldsymbol{\nu}$. Then the job of VI algorithms is to optimise $\boldsymbol{\nu}$ in order to minimise the *Kullback-Leibler* (KL) divergence between the original probability distribution and our approximation. KL divergence (asymmetrically) measures how different the two distributions are, and in this case can be defined as [8]

$$\begin{aligned} D_{\mathrm{KL}}(q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r})||p(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}|\mathcal{D}, \mathbf{X_u})) &= \mathbb{E}_{(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})\sim q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})}[\log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}) - \log p(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}|\mathcal{D}, \mathbf{X_u})] \\ &= \mathbb{E}_{(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})\sim q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})}[\log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}) - \log p(\mathcal{D}, \boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}, \mathbf{r})] \\ &\quad + \mathbb{E}_{(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})\sim q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})}[\log p(\mathcal{D}, \mathbf{X_u})]. \end{aligned}$$

The last term is both hard to compute and constant w.r.t. $q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r})$ [8], so we can remove it from our optimisation objective. The negation of what remains is often called the *evidence lower bound* (ELBO) and is defined as[1] [7, 8]

$$\begin{aligned} \mathcal{L}(\boldsymbol{\nu}) &= \mathbb{E}_{(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})\sim q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})}\left[\log \frac{p(\mathcal{D}, \boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}, \mathbf{r})}{q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r})}\right] \\ &= \iiint q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}) \log \frac{p(\mathcal{D}, \boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}, \mathbf{r})}{q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r})}\, d\boldsymbol{\lambda}\, d\mathbf{u}\, d\mathbf{r}. \end{aligned} \tag{6}$$

By considering full probability distributions instead of point estimates—as long as the approximations are able to capture important features of the posterior—our predictions are likely to be more accurate and rely on fewer assumptions. Moreover, we hope to make use of various recent advancements in VI for both time complexity and approximation distribution fit (see Section 3), making the resulting algorithm competitive both in terms of running time and model fit.

3 Literature Survey

As mentioned in the introduction, most IRL algorithms assume that the reward function can be represented as a linear combination of features. This assumption originated in one of the earliest papers on the topic by Ng and Russell [44], which introduced several linear programming approaches to the problem. The authors also noticed that often multiple reward functions can explain the same behaviour, and suggested heuristics for reward functions that are "far away" from reward functions that do not fit the data.

A few years later, Abbeel and Ng [1] developed an algorithm for the same formulation of the problem with a guarantee to converge quickly. Neu and Szepesvári [43] identified a weakness in Abbeel and Ng's approach: the algorithm requires features to be "appropriately" scaled, and optimal scaling may not be known. Instead, they suggest a way to combine IRL with *apprenticeship learning*, i.e., a supervised learning task for optimal *policy* recovery (whereas IRL focuses on recovering the reward function).

[1] Throughout the proposal, all integrals should be interpreted as Lebesgue integrals over the entire sample space.

Ramachandran and Amir [51] were the first to formulate IRL in terms of Bayesian learning. While the model is easily interpretable and able to handle experts that make mistakes, the algorithm can only handle small state spaces and requires Monte Carlo Markov Chain (MCMC) sampling for inference.

Ziebart et al. [77] keep the linearity assumption, but introduce an influential idea: resolving the ambiguity when multiple reward functions explain the data by appealing to the maximum entropy principle.

Choi and Kim [14] extend the Bayesian model to learn good features as well as the reward function, trying to overcome the limitation of linearity. However, the approach is quite limiting: all features are assumed to have Boolean values, and the algorithm simply learns their conjunctions.

Levine et al. [35] are the first to suggest a way to learn nonlinear reward functions without harsh restrictions on the problem domain by using GPs. We base our work primarily on their paper, and the weaknesses we hope to address have already been covered in the previous section. A recent extension to their work by Jin et al. [30] aims to harness the power of deep learning by using several layers of GPs, making the model less dependent on being provided good features. They also use VI, but with a few simplifying assumptions: deterministic training conditional for the reward vector, and fully independent training conditional for the latent state (see Section 3.1 for more details).

3.1 Variational Inference

Variational inference has seen a recent increase in interest among academics, with different approaches focusing on different goals: better time complexity, handling a wider variety of models, making approximations more accurate, and using more complex function approximation techniques (such as neural networks) to infer local latent variable values without having to calculate them individually for each data point [75]. As our IRL model is based on a GP, we will begin by reviewing some of the VI approaches applied specifically to GP regression. Based on a recent review of scalable GPs [75], we will concentrate on *stochastic variational sparse approximations*, as they have achieved modelling accuracy close to that of the full GP with no approximations, while providing a time complexity of $\mathcal{O}(m^3)$. Below we provide a short overview of various assumptions that have been used in approximating *sparse* GPs (i.e., GPs that use inducing points), following on a paper by Quiñonero-Candela and Rasmussen [49].

Subset of data ($\mathcal{O}(m^3)$) is a baseline method of simply using a subset of data points.

Subset of regressors ($\mathcal{O}(nm^2)$) [62, 63, 71] is a degenerate approximation that uses a weight for each inducing point. A GP is called *degenerate* if the covariance function has a finite number of non-zero eigenvalues, restricting the prior distribution to only a finite number of linearly independent functions [49].

Authors, year	Inducing points	Hyperparameters	Complexity
Titsias, 2009 [67]	variational	variational	$\mathcal{O}(nm^2)$
Hensman et al., 2013 [22]	fixed	variational	$\mathcal{O}(m^3)$
Gal et al., 2014 [19]	variational	variational	$\mathcal{O}(nm^2)$
Hoang et al., 2015 [24]	fixed	fixed	$\mathcal{O}(m^3)$
Cheng and Boots, 2017 [12]	variational	variational	$\mathcal{O}(nm_\alpha + nm_\beta^2)$
Hensman et al., 2017 [21]	fixed	variational	$\mathcal{O}(nm)$
Peng et al., 2017 [45]	variational	variational	$\mathcal{O}(m^3)$

Table 1: Summary of relevant VI approximations to GPs. For both inducing points and hyperparameters, 'fixed' means 'chosen before the algorithm starts' and 'variational' means 'included amongst the variational parameters'. Hyperparameters m_α and m_β refer to the number of bases used to represent the GP's mean and covariance, respectively.

Deterministic training conditional ($\mathcal{O}(nm^2)$) [60] approximation imposes a zero-variance normal distribution for $\mathbf{r}|\mathbf{u}$, resulting in the same mean but different variance predictions compared to the subset of regressors.

Fully independent training conditional ($\mathcal{O}(nm^2)$) [64] has the assumption that the GP values are independent of each other when conditioned on the inducing values:

$$q(\mathbf{r}|\mathbf{u}) = \prod_{i=1}^{n} p(r_i|\mathbf{u}).$$

Partially independent training conditional ($\mathcal{O}(nm^2)$) [68, 59] approximates the same distribution as the fully independent training conditional, but considers a block diagonal rather than a diagonal covariance matrix.

Transduction tailors the predictive distribution to specific test inputs [49]. As we are not too concerned about a specific set of test inputs in the IRL setting, transduction is of limited interest to us.

Augmentation [53] aims to improve predictive accuracy by adding each test input to the inducing points.

Nyström approximation ($\mathcal{O}(nm^2)$) [72] approximates the prior covariance of $\mathbf{r}$, but can lead to negative predictive variances.

Relevance vector machine ($\mathcal{O}(m^3)$) [66] is a degenerative approximation supporting a limited range of covariance functions.

Table 1 further summarises some of the recent and/or influential GP approximation approaches that might be suitable for our GP IRL model. In order to derive a reliable ELBO, the inducing points should be either fixed or modelled by a probability distribution (none of the approaches do that), while hyperparameters cannot be variational (the reason will be explained in Section 4).

- We would also like to avoid having the hyperparameters fixed, as learning them efficiently would introduce a separate problem. As the approach by Hoang et al. does not seem to be easily extendable to modelled hyperparameters, it is unsuitable to our needs.
- (TODO: this will have comments about other papers as well and will be restructured into a paragraph) The variational Fourier features (VFF) algorithm by Hensman et al. [21] is only defined for Matérn kernels, which is likely to be too restrictive for our situation (the original paper on using GPs for IRL [35] used the automatic relevance detection kernel that has weights controlling how important each feature is). While extending VFF to support a flexible class of kernels defined by Wilson and Adams [73] is an interesting and promising avenue of work, it is likely to be beyond the scope of this project.
- The derivation of the ELBO in the work of Peng et al. [45] relies primarily on the fact that the evidence for a GP is a Gaussian, while in our case the evidence can be defined in several ways depending on the model, but must always involve $\mathcal{D}$, making the main idea of the paper inapplicable to IRL.

Since we expect $p(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r} | \mathcal{D}, \mathbf{X}_\mathbf{u})$ to be highly irregular, we would like our approximation to be capable of representing a wide range of possible probability distributions. The primary way to represent complex posteriors in VI is by using *normalising flows*, i.e., a collection of invertible functions—parametrised by additional variational parameters—that are applied to latent variables [55]. Unfortunately, this parametrisation also means that the gradient of the joint probability distribution w.r.t. variational parameters $\nabla_{\boldsymbol{\nu}} p(\mathcal{D}, \boldsymbol{\lambda}, \mathbf{X}_\mathbf{u}, \mathbf{u}, \mathbf{r})$ is no longer zero, making an analytic expression for ELBO impossible using the usual methods. While it may be possible to have an approximating distribution for the flow parameters, it is uncertain how such an algorithm would behave, as it would have to perform optimisation in a space with significantly more dimensions.

4 Proposed Approach

In order to properly investigate the difference between variational inference and maximum likelihood estimation for the model, we keep other parts of the model the same. Namely, we set the covariance function to a version of the automatic relevance detection kernel [35, 42]

$$k_{\boldsymbol{\lambda}}(\mathbf{x}_i, \mathbf{x}_j) = \lambda_0 \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\intercal \boldsymbol{\Lambda}(\mathbf{x}_i - \mathbf{x}_j) - \mathbb{1}[i \neq j]\sigma^2 \operatorname{Tr}[\boldsymbol{\Lambda}]\right),$$

where λ_0 is the overall "scale" factor for how similar or distant the states are,

$$\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$$

is a diagonal matrix that determines how relevant each feature is (where d denotes the number of features), $\mathbb{1}$ is defined as

$$\mathbb{1}[b] = \begin{cases} 1 & \text{if } b \text{ is true} \\ 0 & \text{otherwise,} \end{cases}$$

and σ^2 is set to $10^{-2}/2$ (as the original paper noted that the value makes little difference to the performance of the algorithm [35]). Our vector of hyperparameters for the covariance function is then $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_d)^\intercal$. Similarly, we keep the expression for the prior of $\boldsymbol{\lambda}$:

$$p(\boldsymbol{\lambda}|\mathbf{X_u}) = \exp\left(-\frac{1}{2}\operatorname{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}] - \sum_{i=1}^{d}\log(\lambda_i + 1)\right). \tag{7}$$

Next, we can rewrite the posterior by using the chain rule and Bayes' theorem in order to get a better sense of what we are trying to approximate:

$$\begin{aligned} p(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}|\mathcal{D}, \mathbf{X_u}) &= p(\boldsymbol{\lambda}|\mathbf{X_u}, \mathcal{D}) \times p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathcal{D}) \times p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}, \mathcal{D}) \\ &\propto p(\boldsymbol{\lambda}|\mathbf{X_u}, \mathcal{D}) \times p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathcal{D}) \times p(\mathcal{D}|r) \times p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) \\ &\propto p(\boldsymbol{\lambda}|\mathbf{X_u}, \mathcal{D}) \times p(\mathcal{D}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) \times p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u}) \times p(\mathcal{D}|r) \times p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) \\ &\propto p(\mathcal{D}|\boldsymbol{\lambda}, \mathbf{X_u}) \times p(\boldsymbol{\lambda}|\mathbf{X_u}) \times p(\mathcal{D}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) \times p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u}) \times p(\mathcal{D}|r) \times p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) \end{aligned}$$

Note that now there are only two unknown probability distributions: $p(\mathcal{D}|\boldsymbol{\lambda}, \mathbf{X_u})$ and $p(\mathcal{D}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u})$, which can be computed as follows:

$$\begin{aligned} p(\mathcal{D}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) &= \int p(\mathcal{D}|r) \times p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u})\, d\mathbf{r}, \\ p(\mathcal{D}|\boldsymbol{\lambda}, \mathbf{X_u}) &= \iint p(\mathcal{D}|r) \times p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) \times p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u})\, d\mathbf{u}\, d\mathbf{r}. \end{aligned}$$

This suggests the following form for the approximation:

$$q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r}) = q(\boldsymbol{\lambda}) \times q(\mathbf{u}|\boldsymbol{\lambda}) \times q(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{u}). \tag{8}$$

4.1 The Structure of the Approximating Distribution

At this point we are forced to make assumptions about the approximate posterior in order to arrive at an implementable solution. Time permitting, ways to relax the assumptions may be investigated towards the end of the project.

First, as is common in the literature for applying VI to GPs [12, 22, 24, 67], we simply set

$$q(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{u}) = p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}). \tag{9}$$

We can make a similarly justified choice for $q(\mathbf{u}|\boldsymbol{\lambda})$:

$$q(\mathbf{u}|\boldsymbol{\lambda}) = q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S}) \tag{10}$$

for some variational parameters $\mathbf{m} \in \mathbb{R}^m$ and a positive semi-definite matrix $\mathbf{S} \in \mathbb{R}^{m \times m}$. [12, 21, 23].

In order to have a reasonable way of calculating/approximating the ELBO (see Section 4.4), we need to have an approximating distribution for $\boldsymbol{\lambda}$, but unfortunately all the papers in Table 1 either fix it or treat it as a variational parameter. Hence we make our first assumption without justification from previous literature:

$$q(\boldsymbol{\lambda}) = \prod_{i=0}^{d} q(\lambda_i). \tag{11}$$

We want to restrict λ_0 to be positive so that $k_{\boldsymbol{\lambda}}$ would produce non-negative values and not become trivial. Similarly, we want that $\lambda_i \geq 0$ for $i = 1, \ldots, d$ so that $\boldsymbol{\Lambda}$ is a positive-definite matrix. Considering the possible distributions for all $d + 1$ variables, we would like the mean to be flexible (i.e., not tied to zero, like with the exponential distribution), and the tails to converge to zero as the value of the random variable moves away from the mean. We might want to support some right skew, but the distribution should be close to symmetric with at least some parameter values. This limits our choice of distributions quite significantly, and we decide to go with the gamma distribution as it is fairly flexible and commonly used [27]. We then define the probability density functions as

$$q(\lambda_i) = \Gamma(\lambda_i; \alpha_i, \beta_i) = \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} \lambda_i^{\alpha_i - 1} e^{-\beta_i \lambda_i}, \quad i = 0, \ldots, d, \tag{12}$$

where $\alpha_i > 0$ and $\beta_i > 0$ are parameters of the distribution, and $\Gamma(\cdot)$ is the gamma function. This gives us our vector of variational parameters $\boldsymbol{\nu} = (\mathbf{m}, \mathbf{S}, \alpha_0, \beta_0, \ldots, \alpha_d, \beta_d)^\intercal$.

4.2 Evidence Lower Bound

In this section we derive and simplify the ELBO for this (now fully specified) model. In order to derive the ELBO, let us go back to (6) and write[2]

$$\mathcal{L}(\boldsymbol{\nu}) = \mathbb{E}[\log p(\mathcal{D}, \boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}, \mathbf{r})] - \mathbb{E}[\log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}, \mathbf{u}, \mathbf{r})].$$

By plugging in (3) and (8), we get

$$\begin{aligned}\mathcal{L}(\boldsymbol{\nu}) = \mathbb{E}[&\log p(\mathbf{X_u}) + \log p(\boldsymbol{\lambda}|\mathbf{X_u}) + \log p(\mathbf{u}|\boldsymbol{\lambda}, \mathbf{X_u}) + \log p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u}) + \log p(\mathcal{D}|r)] \\ &- \mathbb{E}[\log q(\boldsymbol{\lambda}) + \log q(\mathbf{u}) + \log q(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{u})].\end{aligned}$$

Note that $\mathbb{E}[\log p(\mathbf{X_u})]$ is just a constant, so we can simply drop it from the expression. Furthermore, since $q(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{u}) = p(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{X_u}, \mathbf{u})$, they cancel each other out. Then we can

[2]At this point, we will drop the subscript denoting which variables the expectation is taken over. Also note that throughout the derivation equality is taken to mean "equality up to an additive constant".

substitute various terms with their definitions to get

$$\begin{aligned}\mathcal{L}(\boldsymbol{\nu}) = \mathbb{E}\left[-\frac{1}{2}\operatorname{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}] - \sum_{i=1}^{d}\log(\lambda_i+1)\right] + \mathbb{E}[\log\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{K}_{\mathbf{u},\mathbf{u}})] \\ + \mathbb{E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T} Q_r(s_{i,t},a_{i,t}) - V_r(s_{i,t})\right] \quad - \sum_{i=0}^{d}\mathbb{E}\left[\log\left(\frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)}\lambda_i^{\alpha_i-1}e^{-\beta_i\lambda_i}\right)\right] \\ - \mathbb{E}[\log\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})].\end{aligned}$$

We can simplify the last term by noting that $-\mathbb{E}[\log\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})] = \frac{1}{2}\log|\mathbf{S}|$ up to an additive constant that depends on the number of dimensions of the distribution [2]. Also note that we cannot use this fact for $\mathbb{E}[\log\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{K}_{\mathbf{u},\mathbf{u}})]$ because $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ depends on $\boldsymbol{\lambda}$. We also plug in the definitions of $\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{K}_{\mathbf{u},\mathbf{u}})$ and Q_r, and get:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\nu}) = \frac{1}{2}\log|\mathbf{S}| + \mathbb{E}\left[-\frac{1}{2}\operatorname{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}] - \sum_{i=1}^{d}\log(\lambda_i+1)\right] \\ + \mathbb{E}\left[-\frac{1}{2}\mathbf{u}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}| - \frac{m}{2}\log 2\pi\right] \\ + \mathbb{E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T} r(s_{i,t}) - V_r(s_{i,t}) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')V_r(s')\right] \\ - \sum_{i=0}^{d}\mathbb{E}[\alpha_i\log\beta_i - \log\Gamma(\alpha_i) + (\alpha_i-1)\log\lambda_i - \beta_i\lambda_i]\end{aligned}$$

Now we can remove $\mathbb{E}\left[-\frac{m}{2}\log 2\pi\right]$ since it is constant w.r.t. both the variational parameters and the variables the expectation is over, and move constants (or variational parameters) independent of the approximated variables outside of the expectations. Also note that $\mathbb{E}[\lambda_i] = \alpha_i/\beta_i$ and $\mathbb{E}[\log\lambda_i] = \psi(\alpha_i) - \log\beta_i$, where ψ is the digamma function defined as $\psi(x) = \frac{d}{dx}\log\Gamma(x)$ [7]. Moreover, we can simplify $\sum_{i=1}^{N}\sum_{t=1}^{T} r(s_{i,t})$ by choosing to represent all visited states in $\mathbf{r} = (r_1,\ldots,r_k)^\intercal$, and defining a new vector $\mathbf{t} = (t_1,\ldots,t_k)^\intercal$, where t_i is the number of times the state associated with the reward r_i has been visited across all demonstrations. Then

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T} r(s_{i,t})\right] &= \mathbb{E}[\mathbf{t}^\intercal\mathbf{r}] = \mathbf{t}^\intercal\mathbb{E}[\mathbf{r}] = \mathbf{t}^\intercal\mathbb{E}_{(\boldsymbol{\lambda},\mathbf{u})\sim q(\boldsymbol{\lambda})q(\mathbf{u})}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}] \\ &= \mathbf{t}^\intercal\mathbb{E}_{\boldsymbol{\lambda}\sim q(\boldsymbol{\lambda})}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}] = \mathbf{t}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m}.\end{aligned}$$

Finally, as $\mathbf{u}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$ is a function of $\mathbf{u}$ and $\boldsymbol{\lambda}$, we can take the expectation of $\mathbf{u}$, leaving the expectation of $\boldsymbol{\lambda}$:

$$\mathbb{E}[\mathbf{u}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}] = \mathbb{E}[\operatorname{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{S}] + \mathbf{m}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}] = \operatorname{Tr}[\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{S}] + \mathbf{m}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m}.$$

This allows us to simplify $\mathcal{L}(\boldsymbol{\nu})$ to the following:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\nu}) = &\frac{1}{2}\log|\mathbf{S}| - \frac{1}{2}\operatorname{Tr}[\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}]] - \sum_{i=1}^{d}\mathbb{E}[\log(\lambda_i+1)] - \frac{1}{2}\operatorname{Tr}[\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{S}] \\ &- \frac{1}{2}\mathbf{m}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m} - \frac{1}{2}\mathbb{E}[\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|] + \sum_{i=0}^{d}\alpha_i - \log\beta_i + \log\Gamma(\alpha_i) + (1-\alpha_i)\psi(\alpha_i) \\ &+ \mathbf{t}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m} - \sum_{i=1}^{N}\sum_{t=1}^{T}\mathbb{E}[V_r(s_{i,t})] - \gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\mathbb{E}[V_r(s')].\end{aligned}$$

4.3 Derivatives

4.3.1 $\partial/\partial\mathbf{m}$

We begin by removing terms independent of $\mathbf{m}$:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{m}} = &-\frac{1}{2}\frac{\partial}{\partial\mathbf{m}}[\mathbf{m}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m}] + \frac{\partial}{\partial\mathbf{m}}[\mathbf{t}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m}] \\ &- \sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\partial}{\partial\mathbf{m}}\mathbb{E}[V_r(s_{i,t})] - \gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\frac{\partial}{\partial\mathbf{m}}\mathbb{E}[V_r(s')].\end{aligned}$$

Here

$$\begin{aligned}\frac{\partial}{\partial\mathbf{m}}[\mathbf{m}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m}] &= (\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}] + \mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]^\intercal)\mathbf{m}, \\ \frac{\partial}{\partial\mathbf{m}}[\mathbf{t}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{m}] &= \mathbf{t}^\intercal\mathbb{E}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}],\end{aligned}$$

and

$$\begin{aligned}\frac{\partial}{\partial\mathbf{m}}\mathbb{E}[V_r(s)] &= \frac{\partial}{\partial\mathbf{m}}\iiint V_r(s)p(\mathbf{r}|\boldsymbol{\lambda},\mathbf{X}_{\mathbf{u}},\mathbf{u})\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})q(\boldsymbol{\lambda})\,d\mathbf{r}\,d\mathbf{u}\,d\boldsymbol{\lambda} \\ &= \iiint V_r(s)p(\mathbf{r}|\boldsymbol{\lambda},\mathbf{X}_{\mathbf{u}},\mathbf{u})\frac{\partial}{\partial\mathbf{m}}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})q(\boldsymbol{\lambda})\,d\mathbf{r}\,d\mathbf{u}\,d\boldsymbol{\lambda},\end{aligned} \tag{13}$$

where

$$\begin{aligned}\frac{\partial}{\partial\mathbf{m}}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S}) &= \mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})\frac{\partial}{\partial\mathbf{m}}\left[-\frac{1}{2}(\mathbf{u}-\mathbf{m})^\intercal\mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})\right] \\ &= \mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})\left(-\frac{1}{2}\right)(\mathbf{S}^{-1}+\mathbf{S}^{-\intercal})(\mathbf{u}-\mathbf{m})\frac{\partial}{\partial\mathbf{m}}[\mathbf{u}-\mathbf{m}] \\ &= \mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})\frac{1}{2}(\mathbf{S}^{-1}+\mathbf{S}^{-\intercal})(\mathbf{u}-\mathbf{m}).\end{aligned}$$

Substituting it back into (13) gives

$$\begin{aligned}\frac{\partial}{\partial \mathbf{m}}\mathbb{E}[V_r(s)] &= \frac{1}{2}\iiint V_r(s)(\mathbf{S}^{-1}+\mathbf{S}^{-\intercal})(\mathbf{u}-\mathbf{m})p(\mathbf{r}|\boldsymbol{\lambda},\mathbf{X}_{\mathbf{u}},\mathbf{u})\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})q(\boldsymbol{\lambda})\,d\mathbf{r}\,d\mathbf{u}\,d\boldsymbol{\lambda}\\ &= \frac{1}{2}\mathbb{E}[V_r(s)(\mathbf{S}^{-1}+\mathbf{S}^{-\intercal})(\mathbf{u}-\mathbf{m})].\end{aligned}$$

Hence

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{m}} =& -\frac{1}{2}(\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]+\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]^{\intercal})\mathbf{m}+\mathbf{t}^{\intercal}\mathbb{E}[\mathbf{K}_{\mathbf{r},\mathbf{u}}^{\intercal}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\\ &-\frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbb{E}[V_r(s_{i,t})(\mathbf{S}^{-1}+\mathbf{S}^{-\intercal})(\mathbf{u}-\mathbf{m})]\\ &-\gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\mathbb{E}[V_r(s')(\mathbf{S}^{-1}+\mathbf{S}^{-\intercal})(\mathbf{u}-\mathbf{m})].\end{aligned}$$

4.3.2 $\partial/\partial\mathbf{S}$

Similarly to the previous section,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{S}} =& \frac{1}{2}\frac{\partial}{\partial \mathbf{S}}\log|\mathbf{S}|-\frac{1}{2}\frac{\partial}{\partial \mathbf{S}}\operatorname{Tr}[\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{S}]\\ &-\sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\partial}{\partial \mathbf{S}}\mathbb{E}[V_r(s_{i,t})]-\gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\frac{\partial}{\partial \mathbf{S}}\mathbb{E}[V_r(s')],\end{aligned}$$

where

$$\frac{\partial}{\partial \mathbf{S}}\log|\mathbf{S}| = \mathbf{S}^{-\intercal},$$

and

$$\frac{\partial}{\partial \mathbf{S}}\operatorname{Tr}[\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]\mathbf{S}] = \mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]^{\intercal}$$

by *The Matrix Cookbook* [46]. Then

$$\frac{\partial}{\partial \mathbf{S}}\mathbb{E}[V_r(s)] = \iiint V_r(s)q(\mathbf{r})\frac{\partial}{\partial \mathbf{S}}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})q(\boldsymbol{\lambda})\,d\mathbf{r}\,d\mathbf{u}\,d\boldsymbol{\lambda},$$

where

$$\begin{aligned}\frac{\partial}{\partial \mathbf{S}}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S}) &= \frac{\partial}{\partial \mathbf{S}}\left[\frac{1}{(2\pi)^{m/2}|\mathbf{S}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{u}-\mathbf{m})^{\intercal}\mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})\right)\right]\\ &= \frac{\partial}{\partial \mathbf{S}}\left[\frac{1}{(2\pi)^{m/2}|\mathbf{S}|^{1/2}}\right]\exp\left(-\frac{1}{2}(\mathbf{u}-\mathbf{m})^{\intercal}\mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})\right)\\ &+\frac{1}{(2\pi)^{m/2}|\mathbf{S}|^{1/2}}\frac{\partial}{\partial \mathbf{S}}\left[\exp\left(-\frac{1}{2}(\mathbf{u}-\mathbf{m})^{\intercal}\mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})\right)\right]\\ &= \frac{1}{(2\pi)^{m/2}}\frac{\partial}{\partial \mathbf{S}}\left[\frac{1}{|\mathbf{S}|^{1/2}}\right]\exp\left(-\frac{1}{2}(\mathbf{u}-\mathbf{m})^{\intercal}\mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})\right)\\ &-\frac{1}{2}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})\frac{\partial}{\partial \mathbf{S}}[(\mathbf{u}-\mathbf{m})^{\intercal}\mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})].\end{aligned}$$

The two remaining derivatives can be taken with the help of *The Matrix Cookbook* [46]:

$$\frac{\partial}{\partial \mathbf{S}}\left[\frac{1}{|\mathbf{S}|^{1/2}}\right] = -\frac{1}{2}|\mathbf{S}|^{-3/2}\frac{\partial|\mathbf{S}|}{\partial \mathbf{S}} = -\frac{1}{2}|\mathbf{S}|^{-3/2}|\mathbf{S}|\mathbf{S}^{-\intercal} = -\frac{1}{2|\mathbf{S}|^{1/2}}\mathbf{S}^{-\intercal},$$

$$\frac{\partial}{\partial \mathbf{S}}[(\mathbf{u}-\mathbf{m})^\intercal \mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})] = -\mathbf{S}^{-\intercal}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\intercal\mathbf{S}^{-\intercal}.$$

Plugging them back in gives

$$\frac{\partial}{\partial \mathbf{S}}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S}) = -\frac{1}{2}\mathbf{S}^{-\intercal}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S}) + \frac{1}{2}\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})\mathbf{S}^{-\intercal}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\intercal\mathbf{S}^{-\intercal},$$

and

$$\begin{aligned}\frac{\partial}{\partial \mathbf{S}}\mathbb{E}[V_r(s)] &= \frac{1}{2}\iiint V_r(s)(\mathbf{S}^{-\intercal}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\intercal\mathbf{S}^{-\intercal} - \mathbf{S}^{-\intercal})q(\mathbf{r})\mathcal{N}(\mathbf{u};\mathbf{m},\mathbf{S})q(\boldsymbol{\lambda})\,d\mathbf{r}\,d\mathbf{u}\,d\boldsymbol{\lambda} \\ &= \frac{1}{2}\mathbb{E}[V_r(s)(\mathbf{S}^{-\intercal}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\intercal\mathbf{S}^{-\intercal} - \mathbf{S}^{-\intercal})].\end{aligned}$$

Therefore

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{S}} = &\frac{1}{2}\mathbf{S}^{-\intercal} - \frac{1}{2}\mathbb{E}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}]^\intercal - \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbb{E}[V_r(s_{i,t})(\mathbf{S}^{-\intercal}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\intercal\mathbf{S}^{-\intercal} - \mathbf{S}^{-\intercal})] \\ &- \gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\mathbb{E}[V_r(s')(\mathbf{S}^{-\intercal}(\mathbf{u}-\mathbf{m})(\mathbf{u}-\mathbf{m})^\intercal\mathbf{S}^{-\intercal} - \mathbf{S}^{-\intercal})].\end{aligned}$$

4.3.3 $\partial/\partial\alpha_j$

We begin in the usual way:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \alpha_j} = &-\frac{1}{2}\frac{\partial}{\partial\alpha_j}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}]] - \frac{1}{2}\frac{\partial}{\partial\alpha_j}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{S}]] - \frac{1}{2}\frac{\partial}{\partial\alpha_j}\mathbb{E}[\mathbf{m}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}] - \frac{1}{2}\frac{\partial}{\partial\alpha_j}\mathbb{E}[\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|] \\ &+ \frac{\partial}{\partial\alpha_j}\mathbb{E}[\mathbf{t}^\intercal\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}] + \frac{\partial}{\partial\alpha_j}[\alpha_j + \log\Gamma(\alpha_j) + (1-\alpha_j)\psi(\alpha_j)] \\ &- \sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\partial}{\partial\alpha_j}\mathbb{E}[V_r(s_{i,t})] - \gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\frac{\partial}{\partial\alpha_j}\mathbb{E}[V_r(s')].\end{aligned}$$

First,

$$\frac{\partial}{\partial\alpha_j}[\alpha_j + \log\Gamma(\alpha_j) + (1-\alpha_j)\psi(\alpha_j)] = 1 + \psi(\alpha_j) - \psi(\alpha_j) + (1-\alpha_j)\psi'(\alpha_j) = 1 + (1-\alpha_j)\psi'(\alpha_j)$$

by the definition of ψ. The remaining terms can all be treated in the same way, as they all contain expectations of scalar functions that are independent of α_j, and α_j only occurs in $\Gamma(\lambda_j;\alpha_j,\beta_j)$. Thus we can work with an abstract function as follows:

$$\begin{aligned}\frac{\partial}{\partial\alpha_j}\mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})] &= \frac{\partial}{\partial\alpha_j}\iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})q(\boldsymbol{\lambda})q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u} \\ &= \iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})q(\lambda_0)\cdots q(\lambda_{j-1})\frac{\partial}{\partial\alpha_j}\left[\frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)}\lambda_j^{\alpha_j-1}\right]e^{-\beta_j\lambda_j} \\ &\qquad q(\lambda_{j+1})\cdots q(\lambda_d)q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u}.\end{aligned}$$

Then

$$\begin{aligned}
\frac{\partial}{\partial\alpha_j}\left[\frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)}\lambda_j^{\alpha_j-1}\right] &= \frac{\frac{\partial}{\partial\alpha_j}[\beta_j^{\alpha_j}\lambda_j^{\alpha_j-1}]\Gamma(\alpha_j)-\beta_j^{\alpha_j}\lambda_j^{\alpha_j-1}\Gamma'(\alpha_j)}{(\Gamma(\alpha_j))^2}\\
&= \frac{\beta_j^{\alpha_j}\lambda_j^{\alpha_j-1}\frac{\partial}{\partial\alpha_j}[\alpha_j\log\beta_j+(\alpha_j-1)\log\lambda_j]\Gamma(\alpha_j)-\beta_j^{\alpha_j}\lambda_j^{\alpha_j-1}\Gamma'(\alpha_j)}{(\Gamma(\alpha_j))^2}\\
&= \frac{\beta_j^{\alpha_j}\lambda_j^{\alpha_j-1}(\log\beta_j+\log\lambda_i)\Gamma(\alpha_j)-\beta_j^{\alpha_j}\lambda_j^{\alpha_j-1}\Gamma'(\alpha_j)}{(\Gamma(\alpha_j))^2}\\
&= \frac{\beta_j^{\alpha_j}\lambda_j^{\alpha_j-1}}{\Gamma(\alpha_j)}\left(\log\beta_j+\log\lambda_j-\frac{\Gamma'(\alpha_j)}{\Gamma(\alpha_j)}\right),
\end{aligned}$$

which means that

$$\begin{aligned}
\frac{\partial}{\partial\alpha_j}\mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})] &= \iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})q(\lambda_0)\cdots q(\lambda_{j-1})\frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)}\lambda_j^{\alpha_j-1}e^{-\beta_j\lambda_j}\left(\log\beta_j+\log\lambda_j-\frac{\Gamma'(\alpha_j)}{\Gamma(\alpha_j)}\right)\\
&\qquad q(\lambda_{j+1})\cdots q(\lambda_d)q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u}\\
&= \iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})\left(\log\beta_j+\log\lambda_j-\frac{\Gamma'(\alpha_j)}{\Gamma(\alpha_j)}\right)q(\boldsymbol{\lambda})q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u}\\
&= \mathbb{E}\left[f(k_{\boldsymbol{\lambda}},\mathbf{r})\left(\log\beta_j+\log\lambda_j-\frac{\Gamma'(\alpha_j)}{\Gamma(\alpha_j)}\right)\right]\\
&= \left(\log\beta_j-\frac{\Gamma'(\alpha_j)}{\Gamma(\alpha_j)}\right)\mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})]+\mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})\log\lambda_j].
\end{aligned}$$

With these results in mind, we can simplify the initial expression to

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\alpha_j} &= 1+(1-\alpha_j)\psi'(\alpha_j)+\left(\log\beta_j-\frac{\Gamma'(\alpha_j)}{\Gamma(\alpha_j)}\right)\Bigg(-\frac{1}{2}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}]]-\frac{1}{2}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{S}]]\\
&\quad-\frac{1}{2}\mathbb{E}[\mathbf{m}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}]-\frac{1}{2}\mathbb{E}[\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|]+\mathbb{E}[\mathbf{t}^\intercal\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}]\\
&\quad-\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbb{E}[V_r(s_{i,t})]-\gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\mathbb{E}[V_r(s')]\Bigg)\\
&\quad-\frac{1}{2}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}]\log\lambda_j]-\frac{1}{2}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{S}]\log\lambda_j]-\frac{1}{2}\mathbb{E}[\mathbf{m}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}\log\lambda_j]\\
&\quad-\frac{1}{2}\mathbb{E}[\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|\log\lambda_j]+\mathbb{E}[\mathbf{t}^\intercal\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}\log\lambda_j]\\
&\quad-\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbb{E}[V_r(s_{i,t})\log\lambda_j]-\gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\mathbb{E}[V_r(s')\log\lambda_j].
\end{aligned}$$

4.3.4 $\partial/\partial\beta_j$

Finally,

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\beta_j} =& -\frac{1}{2}\frac{\partial}{\partial\beta_j}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}]] - \frac{1}{2}\frac{\partial}{\partial\beta_j}\mathbb{E}[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{S}]] - \frac{1}{2}\frac{\partial}{\partial\beta_j}\mathbb{E}[\mathbf{m}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}] \\ & - \frac{1}{2}\frac{\partial}{\partial\beta_j}\mathbb{E}[\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|] + \frac{\partial}{\partial\beta_j}\mathbb{E}[\mathbf{t}^\intercal\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}] - \frac{\partial}{\partial\beta_j}[\log\beta_j] \\ & - \sum_{i=1}^{N}\sum_{t=1}^{T}\frac{\partial}{\partial\beta_j}\mathbb{E}[V_r(s_{i,t})] - \gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\frac{\partial}{\partial\beta_j}\mathbb{E}[V_r(s')].\end{aligned}$$

Similarly to the previous section, we can handle all derivatives of expectations in the same way:

$$\begin{aligned}\frac{\partial}{\partial\beta_j}\mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})] &= \frac{\partial}{\partial\beta_j}\iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})q(\boldsymbol{\lambda})q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u} \\ &= \iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})q(\lambda_0)\cdots q(\lambda_{j-1})\frac{\lambda_j^{\alpha_j-1}}{\Gamma(\alpha_j)}\frac{\partial}{\partial\beta_j}[\beta_j^{\alpha_j}e^{-\beta_j\lambda_j}] \\ &\qquad q(\lambda_{j+1})\cdots q(\lambda_d)q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u}.\end{aligned}$$

Since

$$\frac{\partial}{\partial\beta_j}[\beta_j^{\alpha_j}e^{-\beta_j\lambda_j}] = \alpha_j\beta_j^{\alpha_j-1}e^{-\beta_j\lambda_j} - \beta_j^{\alpha_j}e^{-\beta_j\lambda_j}\lambda_j = \beta_j^{\alpha_j}e^{-\beta_j\lambda_j}\left(\frac{\alpha_j}{\beta_j} - \lambda_j\right),$$

we have that

$$\begin{aligned}\frac{\partial}{\partial\beta_j}\mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})] =& \iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})q(\lambda_0)\cdots q(\lambda_{j-1})\frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)}\lambda_j^{\alpha_j-1}e^{-\beta_j\lambda_j}\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right) \\ &\qquad q(\lambda_{j+1})\cdots q(\lambda_d)q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u} \\ =& \iiint f(k_{\boldsymbol{\lambda}},\mathbf{r})\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)q(\boldsymbol{\lambda})q(\mathbf{r})q(\mathbf{u})\,d\boldsymbol{\lambda}\,d\mathbf{r}\,d\mathbf{u} \\ =& \mathbb{E}\left[f(k_{\boldsymbol{\lambda}},\mathbf{r})\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right] = \frac{\alpha_j}{\beta_j}\mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})] - \mathbb{E}[f(k_{\boldsymbol{\lambda}},\mathbf{r})\lambda_j].\end{aligned}$$

This gives us the final expression of $\frac{\partial\mathcal{L}}{\partial\beta_j}$:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\beta_j} =& -\frac{1}{\beta_j} - \frac{1}{2}\mathbb{E}\left[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-2}]\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right] - \frac{1}{2}\mathbb{E}\left[\mathrm{Tr}[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{S}]\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right] \\ & - \frac{1}{2}\mathbb{E}\left[\mathbf{m}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right] - \frac{1}{2}\mathbb{E}\left[\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right] \\ & + \mathbb{E}\left[\mathbf{t}^\intercal\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right] \\ & - \sum_{i=1}^{N}\sum_{t=1}^{T}\mathbb{E}\left[V_r(s_{i,t})\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right] - \gamma\sum_{s'\in\mathcal{S}}\mathcal{T}(s_{i,t},a_{i,t},s')\mathbb{E}\left[V_r(s')\left(\frac{\alpha_j}{\beta_j}-\lambda_j\right)\right].\end{aligned}$$

4.4 Variational Inference Algorithms

The typical way to optimise a quantity (the ELBO, in this case) involves computing its gradient [8]. Due to terms involving $\mathcal{D}$ that are computed by solving an MDP and do not resemble a typical probability distribution, we turn our attention to the *black box variational inference* [52] paper that suggests a convenient trick:

$$\nabla_{\boldsymbol{\nu}} \mathcal{L} = \mathbb{E}_{(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})\sim q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})}[\nabla_{\boldsymbol{\nu}} \log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})(\log p(\mathcal{D},\boldsymbol{\lambda},\mathbf{X}_{\mathbf{u}},\mathbf{u},\mathbf{r}) - \log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r}))].$$

With this trick in mind, we only need to evaluate $\log p(\mathcal{D},\boldsymbol{\lambda},\mathbf{X}_{\mathbf{u}},\mathbf{u},\mathbf{r})$ and take the gradient of $\log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})$. Following the same paper, $\nabla_{\boldsymbol{\nu}}\mathcal{L}$ then has an unbiased estimate

$$\nabla_{\boldsymbol{\nu}} \mathcal{L} \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}_s,\mathbf{u}_s,\mathbf{r}_s)(\log p(\mathcal{D},\boldsymbol{\lambda}_s,\mathbf{X}_{\mathbf{u}},\mathbf{u}_s,\mathbf{r}_s) - \log q_{\boldsymbol{\nu}}(\boldsymbol{\lambda}_s,\mathbf{u}_s,\mathbf{r}_s))$$

computed by drawing S Monte Carlo samples $(\boldsymbol{\lambda}_s,\mathbf{u}_s,\mathbf{r}_s) \sim q_{\boldsymbol{\nu}}(\boldsymbol{\lambda},\mathbf{u},\mathbf{r})$.

5 Work Plan

1. Prove all applications of the dominated convergence theorem.
2. Implement the algorithm as a more specialised version of Algorithm 2 in the black box VI paper.
3. ¡Evaluation¿
4. Experiment with various improvements mentioned in the black box VI paper, other VI papers, especially the ones about GPs.

5.1 Evaluation

Against what? GPIRL and DGP-IRL.

For what problems? GPIRL paper has a big toolkit, but with MATLAB. Binary world from DGP-IRL would be too difficult.

- Object world. Fix one, vary the other: number of colours, number of examples.
- Highway driving behaviour: both synthetic and human performance.

Using what metrics? Visualise the (true and learned) rewards as shades of grey. Also measure expected value difference in four situations:

- discrete features

- discrete feature transfer
- continuous features
- continuous feature transfer

6 Notes on papers (to be removed)

6.1 Miscellaneous

(Directed) similarity between MDPs using restricted Boltzmann machines [10]

Chapter 6 on distance measures [41]

The PhD thesis behind maximum causal entropy [76]

6.2 Gaussian Processes

Simple introduction to GPs for time-series modelling [56]

GPs over graphs instead of vectors (haven't actually read) [69]

Another introduction from physics (skimmed through) [29]

Learning a GP from very little data [47]

One GP for multiple correlated output variables [6]

Kernels for categorical and count data [58]

Scalability/Approximations thesis [28]

6.3 Interpretability

Learning latent factors [38]

The behaviour of Reddit users [16]

6.4 Inverse Reinforcement Learning

One of the first papers on the topic [44]

Bayesian setting [51]

Learning optimal composite features [14]

A different take on IRL with GPs [48]

IRL for large state spaces (haven't read) [11]

Multiple reward functions [13]

A recent survey [3]

Some not-very-successful method [43]

6.4.1 Multiple Strategies

EM clustering [4]

Structured priors [17]

There are more, but I haven't gotten to them yet.

6.5 Variational Inference

Part IV on probabilities and inference [40]

Stochastic VI [26]

Structured stochastic VI [25]

Another review of recent advances [75]

Tighter ELBOs are not necessarily better [50]

For details on Lebesgue's dominated convergence theorem [15]

Still looking for the relevant version of the theorem [32]

Approximation as a multivariate Gaussian (haven't read) [65]

How black box VI can be even more black box [37]

Evaluating VI [74]

6.5.1 for GPs

Linear VI for GPs [12]

Sparse VI for GP [21]

Stochastic VI for sparse spectral GPs (no inducing points, so not that relevant) [23]

Sparse GPs 2 [19]

The 3 papers I need to focus on:

SVI for sparse GPs [22]

distributed 1 [24]

distributed 2 (haven't read) [45]

The paper at the core of the 3 approaches (approximates posterior) [67]

Generalized version? [61]

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In Carla E. Brodley, editor, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004. Briefly mentioned in the literature review.

[2] Nabil A. Ahmed and Digambar V. Gokhale. Entropy expressions and their estimators for multivariate distributions. *IEEE Trans. Information Theory*, 35(3):688–692, 1989. The entropy of a multivariate Gaussian distribution.

[3] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *CoRR*, abs/1806.06877, 2018.

[4] Monica Babes, Vukosi N. Marivate, Kaushik Subramanian, and Michael L. Littman. Apprenticeship learning about multiple intentions. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 897–904. Omnipress, 2011.

[5] Francis R. Bach and David M. Blei, editors. *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015.

[6] Ilias Bilionis, Nicholas Zabaras, Bledar A. Konomi, and Guang Lin. Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification. *J. Comput. Physics*, 241:212–239, 2013.

[7] Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007. Describes VI, KL divergence, and properties of the gamma distribution.

[8] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. A recent review of VI.

[9] Kenneth D. Bogert and Prashant Doshi. Multi-robot inverse reinforcement learning under occlusion with estimation of state transitions. *Artif. Intell.*, 263:46–73, 2018. Using IRL to predict positions of patrolling robots.

[10] H. Bou Ammar, E. Eaton, M.E. Taylor, D.C. Mocanu, K. Driessens, G. Weiss, and K.P. Tuyls. An automated measure of MDP similarity for transfer in reinforcement learning. In *Proceedings of the MLIS-2014 collocated with The Twenty-Eighth AAAI Conference on Artificial Intelligence, 27-28 July 2014, Quebec City, Canada*, pages 31–37, 2014.

[11] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 182–189. JMLR.org, 2011.

[12] Ching-An Cheng and Byron Boots. Variational inference for Gaussian process models with linear complexity. In Guyon et al. [20], pages 5190–5200.

[13] Jaedeug Choi and Kee-Eung Kim. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 314–322, 2012.

[14] Jaedeug Choi and Kee-Eung Kim. Bayesian nonparametric feature construction for inverse reinforcement learning. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 1287–1293. IJCAI/AAAI, 2013. First attempt at learning rewards that are not linear combinations of features by constructing new features.

[15] E. Çınlar. *Probability and Stochastics*. Graduate Texts in Mathematics. Springer New York, 2011.

[16] Sanmay Das and Allen Lavoie. The effects of feedback on human behavior in social media: an inverse reinforcement learning model. In Ana L. C. Bazzan, Michael N. Huhns,

Alessio Lomuscio, and Paul Scerri, editors, *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 653–660. IFAAMAS/ACM, 2014.

[17] Christos Dimitrakakis and Constantin A. Rothkopf. Bayesian multitask inverse reinforcement learning. In Scott Sanner and Marcus Hutter, editors, *Recent Advances in Reinforcement Learning - 9th European Workshop, EWRL 2011, Athens, Greece, September 9-11, 2011, Revised Selected Papers*, volume 7188 of *Lecture Notes in Computer Science*, pages 273–284. Springer, 2011.

[18] Jennifer G. Dy and Andreas Krause, editors. *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2018.

[19] Yarin Gal, Mark van der Wilk, and Carl E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3257–3265, 2014.

[20] Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017.

[21] James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18:151:1–151:52, 2017.

[22] James Hensman, Nicoló Fusi, and Neil D. Lawrence. Gaussian processes for big data. In Ann Nicholson and Padhraic Smyth, editors, *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press, 2013.

[23] Quang Minh Hoang, Trong Nghia Hoang, and Kian Hsiang Low. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 2007–2014. AAAI Press, 2017.

[24] Trong Nghia Hoang, Quang Minh Hoang, and Bryan Kian Hsiang Low. A unifying framework of anytime sparse gaussian process regression models with stochastic variational inference for big data. In Bach and Blei [5], pages 569–578.

[25] Matthew D. Hoffman and David M. Blei. Structured stochastic variational inference. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San*

Diego, California, USA, May 9-12, 2015, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015.

[26] Matthew D. Hoffman, David M. Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[27] Robert V. Hogg, Joseph W. McKean, and Allen T. Craig. *Introduction to Mathematical Statistics*. What's New in Statistics Series. Pearson, 2018. Used as a reference for the gamma distribution.

[28] Hanna Hultin. Evaluation of massively scalable Gaussian processes. Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, June 2017.

[29] David J.C. MacKay. Introduction to Gaussian processes. 168, 01 1998.

[30] Ming Jin, Andreas C. Damianou, Pieter Abbeel, and Costas J. Spanos. Inverse reinforcement learning via deep Gaussian process. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. This paper builds on top of the original GPIRL paper to add an extra layer of latent variables.

[31] Beomjoon Kim and Joelle Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *I. J. Social Robotics*, 8(1):51–66, 2016. An example of how IRL can be used in socially adaptive robot path planning.

[32] O. Knill. *Probability Theory and Stochastic Processes with Applications (Second Edition)*. World Scientific Publishing Company Pte Limited, 2017.

[33] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *I. J. Robotics Res.*, 35(11):1289–1307, 2016. An example of how IRL can be used in socially adaptive robot path planning.

[34] Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors. *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*. MIT Press, 2001.

[35] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 19–27, 2011. First paper to tackle rewards that cannot be expressed as a linear combination of features.

[36] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Supplementary material: Nonlinear inverse reinforcement learning with Gaussian processes. `http://graphics.`

stanford.edu/projects/gpirl/gpirl_supplement.pdf, December 2011. Contains derivations of likelihood partial derivatives and additional details about the implementation.

[37] Yingzhen Li and Qiang Liu. Wild variational approximations. In *NIPS workshop on advances in approximate Bayesian inference*, 2016.

[38] Yunzhu Li, Jiaming Song, and Stefano Ermon. InfoGAIL: Interpretable imitation learning from visual demonstrations. In Guyon et al. [20], pages 3815–3825.

[39] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian process meets big data: A review of scalable GPs. *CoRR*, abs/1807.01065, 2018. A recent review of scalable GPs.

[40] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2003.

[41] B. McCune, J.B. Grace, and D.L. Urban. *Analysis of Ecological Communities*. MjM Software Design, 2002.

[42] R.M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer New York, 2012. The first mention of automatic relevance detection/determination kernels.

[43] Gergely Neu and Csaba Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In Ronald Parr and Linda C. van der Gaag, editors, *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, pages 295–302. AUAI Press, 2007.

[44] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 663–670. Morgan Kaufmann, 2000. An influential early approach to the IRL problem.

[45] Hao Peng, Shandian Zhe, Xiao Zhang, and Yuan Qi. Asynchronous distributed variational Gaussian process for regression. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2788–2797. PMLR, 2017.

[46] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008. Contains the formula for the expectation of a quadratic form and many useful derivatives.

[47] John C. Platt, Christopher J. C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a Gaussian process prior for automatically generating music playlists. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances*

in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada], pages 1425–1432. MIT Press, 2001.

[48] Qifeng Qiao and Peter A. Beling. Inverse reinforcement learning with Gaussian process. *CoRR*, abs/1208.2112, 2012. An alternative formulation of IRL with GPs.

[49] Joaquin Quiñonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005. A review of sparse approximations for GP regression.

[50] Tom Rainforth, Adam R. Kosiorek, Tuan Anh Le, Chris J. Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In Dy and Krause [18], pages 4274–4282.

[51] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2586–2591, 2007. Mentioned in the literature survey.

[52] Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 814–822. JMLR.org, 2014. A basis for any algorithm I can come up with.

[53] Carl E. Rasmussen. Reduced rank Gaussian process learning. *Unpublished manuscript*, 2002. The first time augmentation was suggested.

[54] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. The main book on GPs.

[55] Danilo J. Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Bach and Blei [5], pages 1530–1538. How to use VI to approximate distorted posterior distributions.

[56] Stephen J. Roberts, Matt Osborne, Mark Ebden, Steve Reece, Neale Gibson, and Suzanne Aigrain. Gaussian processes for time-series modelling. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 371 1984:20110550, 2013.

[57] Stuart J. Russell. Learning agents for uncertain environments (extended abstract). In Peter L. Bartlett and Yishay Mansour, editors, *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998.*, pages 101–103. ACM, 1998. The first paper (talk) that defines inverse reinforcement learning.

[58] Terrance Savitsky, Marina Vannucci, and Naijun Sha. Variable selection for nonparametric Gaussian process priors: Models and computational strategies. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 26(1):130, 2011.

[59] Anton Schwaighofer and Volker Tresp. Transductive and inductive methods for approximate Gaussian process regression. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 953–960. MIT Press, 2002. The second paper on the partially independent training conditional approximation.

[60] Matthias W. Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS 2003, Key West, Florida, USA, January 3-6, 2003*. Society for Artificial Intelligence and Statistics, 2003. The paper on the deterministic training conditional GP approximation.

[61] Rishit Sheth, Yuyang Wang, and Roni Khardon. Sparse variational inference for generalized GP models. In Bach and Blei [5], pages 1302–1311.

[62] Bernhard W Silverman. Some aspects of the spline smoothing approach to nonparametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–52, 1985. First paper about the subset of regressors approximation.

[63] Alexander J. Smola and Peter L. Bartlett. Sparse greedy Gaussian process regression. In Leen et al. [34], pages 619–625. The third paper about the subset of regressors approximation.

[64] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 1257–1264, 2005. The paper on the fully independent training conditional GP approximation.

[65] Linda S. L. Tan and David J. Nott. Gaussian variational approximation with sparse precision matrices. *Statistics and Computing*, 28(2):259–275, 2018.

[66] Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001. The relevance vector machine as an approximation for a GP.

[67] Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In David A. Van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*, volume 5 of *JMLR Proceedings*, pages 567–574. JMLR.org, 2009.

[68] Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000. The first paper on the partially independent training conditional approximation.

[69] Arun Venkitaraman, Saikat Chatterjee, and Peter Händel. Gaussian processes over graphs. *CoRR*, abs/1803.05776, 2018.

[70] Adam Vogel, Deepak Ramachandran, Rakesh Gupta, and Antoine Raux. Improving hybrid vehicle fuel efficiency using inverse reinforcement learning. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* AAAI Press, 2012. Using IRL to predict where the driver is going.

[71] Grace Wahba, Xiwu Lin, Fangyu Gao, Dong Xiang, Ronald Klein, and Barbara E. Klein. The bias-variance tradeoff and the randomized GACV. In Michael J. Kearns, Sara A. Solla, and David A. Cohn, editors, *Advances in Neural Information Processing Systems 11, [NIPS Conference, Denver, Colorado, USA, November 30 - December 5, 1998]*, pages 620–626. The MIT Press, 1998. The second paper about the subset of regressors approximation.

[72] Christopher K. I. Williams and Matthias W. Seeger. Using the Nyström method to speed up kernel machines. In Leen et al. [34], pages 682–688. The paper on Nyström approximation.

[73] Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1067–1075, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. A class of kernels generated from a mixture of Gaussian spectral densities.

[74] Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In Dy and Krause [18], pages 5577–5586.

[75] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *CoRR*, abs/1711.05597, 2017. Another recent review, focusing more on comparing algorithms.

[76] Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy.* PhD thesis, Pittsburgh, PA, USA, 2010. AAI3438449.

[77] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008. An influential idea: maximum entropy IRL.

[78] Brian D. Ziebart, Andrew L. Maas, Anind K. Dey, and J. Andrew Bagnell. Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. In Hee Yong Youn and We-Duke Cho, editors, *UbiComp 2008: Ubiquitous Computing, 10th International Conference, UbiComp 2008, Seoul, Korea, September 21-24, 2008, Proceedings*, volume 344 of *ACM International Conference Proceeding Series*, pages 322–331. ACM, 2008. Using IRL to predict the behaviour of taxi drivers.

[79] Brian D. Ziebart, Nathan D. Ratliff, Garratt Gallagher, Christoph Mertz, Kevin M. Peterson, James A. Bagnell, Martial Hebert, Anind K. Dey, and Siddhartha S. Srinivasa. Planning-based prediction for pedestrians. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, MO, USA*, pages 3931–3936. IEEE, 2009. Using IRL to predict the movement of pedestrians.