Expected Information Maximization: Using the I-Projection for Mixture Density Estimation

Dichteschätzung für Mischverteilungen mittels der Informationsprojektion
Master-Thesis von Philipp Becker aus Groß-Umstadt
März 2019

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Expected Information Maximization: Using the I-Projection for Mixture Density Estimation
Dichteschätzung für Mischverteilungen mittels der Informationsprojektion

Vorgelegte Master-Thesis von Philipp Becker aus Groß-Umstadt

1. Gutachten: Prof. Jan Peters Ph.D.
2. Gutachten: M.Sc. Oleg Arenz
3. Gutachten: Prof. Dr. techn. Gerhard Neumann

Tag der Einreichung:

Erklärung zur Abschlussarbeit gemäß § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Philipp Becker, die vorliegende Master-Thesis ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Datum / Date: ____________________

Unterschrift / Signature: ______________________________

Abstract

Latent variable models in general and mixture models in particular are popular parametric families in probabilistic modeling and density estimation. The Expectation Maximization (EM) algorithm [Dempster et al., 1977] is a well-established approach for fitting them to samples. EM maximizes the likelihood of the data given the model, which can lead to solutions that average over modes that can not be represented by the model and can yield significant probability mass in regions where there is no data to support it. In the context of robotics and autonomous systems, averaging over modes can lead to hazardous behavior. In this work, we introduce Expected Information Maximization (EIM), a novel approach to fit latent variable models to samples. EIM aims at finding the information-projection (I-projection), which ignores modes it cannot represent. We combine recent advances in variational inference [Arenz et al., 2018] with density ratio estimation [Sugiyama et al., 2012a] to obtain an upper bound objective that can be optimized using an EM-like procedure. The objective is derived for marginal and conditional latent variable models. Additionally, we provide efficient implementations of EIM for Gaussian mixture models and mixtures of experts by exploiting similarities to information theoretic policy search methods [Deisenroth et al., 2013]. Introducing density ratio estimation relates EIM to Generative Adversarial Networks(GANs) [Goodfellow et al., 2014]. Yet our approach is not adversarial and aims at scenarios where a tractable model is required. We analyze the exact connection between EIM and GANs, as well as other related work. In our experiments, we show the benefits of the I-projection and demonstrate that our approach outperforms existing methods capable of finding it.

Zusammenfassung

Latente Variablenmodelle und insbesondere Mischmodelle sind beliebte parametrische Familien in der probabilistischen Modellierung und Dichteschätzung. Der Expectation Maximization (EM)-Algorithmus [Dempster et al., 1977] ist ein etablierter Ansatz, um diese Modelle von Daten zu lernen. EM maximiert die Wahrscheinlichkeit das die Daten mittels des Modells erzeugt wurden. Dies kann zu Lösungen führen, welche über Modi die nicht dargestellt werden können, mitteln und eine signifikante Wahrscheinlichkeitsdichte in Regionen aufweisen in denen die Daten dies nicht rechtfertigen. Im Kontext von Robotik und autonomen Systemen kann dies zu gefährlichen Verhalten führen. In dieser Arbeit stellen wir Expected Information Maximization (EIM) vor, einen neuen Ansatz latente Variablenmodelle von Daten zu lernen. EIM arbeitet mit der Informationsprojektion, welche Modi ignoriert die nicht dargestellt werden können. Wir kombinieren aktuelle Verfahren der Variationsinferenz [Arenz et al., 2018] und der Schätzung von Dichteverhältnissen [Sugiyama et al., 2012a], um eine obere Schranke zu erhalten, welche wir in einem EM-artigen Verfahren optimieren. Wir leiten diese Schranke für marginale und bedingte latente Variablenmodelle her. Darüber hinaus, leiten wir effiziente Implementierungen für Gaußsche Mischmodelle und Mixtures of Experts her. Hierzu nutzen wir Ähnlichkeiten zu informationstheoretischer Strategiesuche [Deisenroth et al., 2013]. Die Nutzung von Methoden zu Schätzung von Dichteverhältnissen verbindet EIM und Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. Unser Ansatz ist jedoch nicht gegnerisch und zielt auf Szenarien ab in denen ein berechenbares Modell erforderlich ist. Wir analysieren den genauen Zusammenhang zwischen EIM und GANs sowie anderen verwandten Arbeiten. In unseren Experimenten zeigen wir den Nutzen der Informationsprojektion und, dass unser Ansatz besser dazu geeignet ist diese zu finden als bestehende Methoden.

Acknowledgments

I thank my supervisors Oleg Arenz and Gerhard Neumann for their advice, patience, and input in the many discussion that took place while writing this thesis.

My gratitude also goes to my family, especially my parents, for their support during my studies over the last years.

Contents

Figures and Tables

List of Figures

List of Tables

1 Introduction

Learning generative models from and estimating the probability density of data are common and important tasks in machine learning.

Mixture models are commonly used as a class of parametric distributions. By combining simpler distributions, one can construct arbitrary rich models which are still intuitive and efficient to work with. Furthermore, the model complexity can be controlled in an intuitive manner by adapting the number of components. Arguably, the most popular mixture model is the Gaussian mixture model (GMM), consisting of a categorical mixture distribution and Gaussian components. If a conditional distribution should be learned, mixtures of experts have proven to be a reasonable extension to GMMs [Jacobs et al., 1991; Yuksel et al., 2012].

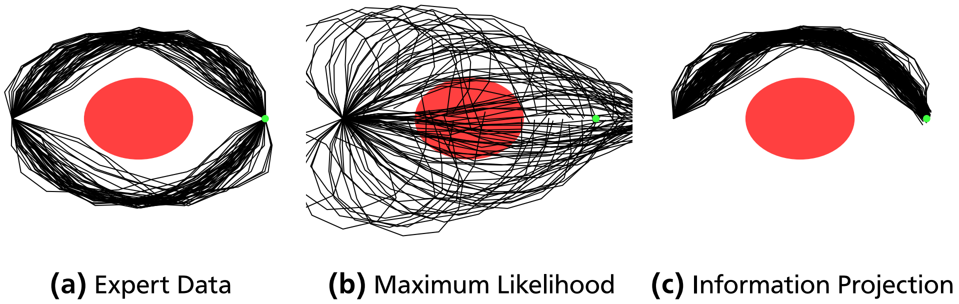
The most common approach to fit mixture models to data is the Expectation Maximization algorithm [Dempster et al., 1977]. This iterative scheme works by maximizing the likelihood of the training data under the model. Yet, maximizing the likelihood for mixture models is prone to over-fitting as well as premature convergence to spurious local optima. Additionally, the maximum likelihood objective forces the model to assign probability density to every value for which the density of the target distribution is non-zero. Thus, in the common case where the model is not rich enough to perfectly fit the target distribution, it averages over modes it cannot represent and may assign the majority of the model's density to regions where there is no data to justify it.

When modeling behavior, especially in the context of autonomous systems and robotics, such averaging behavior can have catastrophic consequences. Averaging over the behavior from multiple experts might cause a robot or an autonomous car to enter regions of their state space that should be avoided. See Figure 1.1 for an example.

Computing the information-projection (I-projection) of the model onto the data distribution provides an alternative. In the unusual case that the model is rich enough to fit the data, both, maximizing the likelihood and computing the I-projection, yield the same solution. However, in the more common case of a model that is not rich enough, the I-projection ignores the modes it cannot represent instead of averaging over them. While the I-projection may ignore some parts of the data, there is also no density where it is not justified by the data, resulting in a safer, more robust behavior. Thus, the I-projection may be the more reasonable choice in the context of autonomous systems and robotics.

Yet, to the best of our knowledge, the only methods capable of finding the I-projection solely based on samples are based on Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. Extensions of the original approach are capable of minimizing arbitrary f-divergences [Nowozin et al., 2016; Poole et al., 2016; Uehara et al., 2016] of which the I-projection is a special case. A key feature of GANs is that they do not require the model density to be tractable which allows using powerful models. Such models can be used to learn high dimensional distributions, e.g., over images. Yet, assuming an intractable model also prevents GANs from utilizing information about the model structure during training. Additionally, their adversarial objective makes training GANs particularly hard.

Recently, Arenz et al. [2018] introduced Variational Inference by Policy Search (VIPS), a novel method for variational inference. Their approach minimizes an upper bound objective in an EM-like procedure. Yet, like all variational inference approaches, they assume access to the unnormalized density of the target distribution. Based on the VIPS objective, we propose Expected Information Maximization (EIM),



(a) Expert Data **(b)** Maximum Likelihood **(c)** Information Projection

Figure 1.1.: We recorded expert data of a 10-link planar robot, tasked with reaching the small green dot, without colliding with an obstacle, i.e., the large red circle. The expert data has two modes in joint space, reaching under and over the obstacle. We fit a single Gaussian to the expert data by maximizing the likelihood and finding the I-projection. The maximum likelihood solution averages over both modes in the expert data, and thus fails to reach the point and collides with the obstacle. The I-projection solution, on the other hand, focuses on one mode and manages to solve the task. Yet, the I-projection solution also ignores the second mode.

a novel approach capable of finding the I-projection between the model and the target distribution solely based on samples from the target distribution. In order to work with this upper bound objective based on samples, we use density ratio estimation [Sugiyama et al., 2012a] to approximate quantities we are unable to compute due to the lack of access to the target distribution. The usage of density ratio estimation relates our approach and GANs.

Yet, while GANs focus on learning models with intractable densities over high dimensional data, we, on the other hand, focus on rather low dimensional scenarios where a tractable model is needed or desired. In this case, one can exploit knowledge about the model density and structure during learning.

Additionally, we derive a similar upper bound objective for conditional distributions to enable learning conditional latent variable models, which increases the number of potential applications of our approach dramatically.

Based on those general derivations we introduce EIM for Gaussian mixture models and mixtures of experts. In the conditional case, the distribution might depend linearly or nonlinearly on the conditioning variable. For both cases, we present efficient implementations of EIM. Similar to [Arenz et al., 2018], we exploit similarities to information theoretic policy search [Peters et al., 2010; Deisenroth et al., 2013; Abdolmaleki et al., 2015], a class of reinforcement learning algorithms, to efficiently realize the updates of the model distributions. Finally, we compare our approach to existing related work on a qualitative level by means of an in-depth analysis and on a quantitative level by means of experiments. Those experiments show the benefits of our approach over GANs as well as EM and further demonstrate the usefulness of the I-projection objective.

2 Preliminaries

In this section, we introduce mixture models, the class of probabilistic models we use throughout this work, as well as the KL divergence, and in particular, the information projection, which will serve as our objective. Additionally, we review several established concepts that are relevant to the derivations of our approach or are conceptually related to it. We work with the same upper bound objective as Variational Inference by Policy Search (VIPS) [Arenz et al., 2018], a recent variational inference approach. Both, VIPS and our approach use ideas from the classical EM algorithm [Dempster et al., 1977] to optimize this upper bound. As we assume only samples are available of the target distribution, we need to employ density ratio estimation [Sugiyama et al., 2012a] to make the bound computable. During the implementation of our approach, we exploit similarities with information theoretic policy search [Deisenroth et al., 2013]. Generative Adversarial Networks [Goodfellow et al., 2014] are not only the only other approach capable of computing the I-projection based on samples, but they are also related to our approach by the used density ratio estimation techniques.

2.1 Mixture Models and EM

Introducing latent variables is a common way to model complex distributions. If carefully chosen, the latent variables can significantly simplify the model structure. Such models are referred to as latent variable models. While being higher dimensional, the joint distribution over observed and latent variables is often more tractable and easier to handle than the marginal over the observed variables.

A popular class of latent variable models are mixture models. The main idea is to split the responsibility of modeling the data between several usually relatively simple components, e.g., Gaussians. The components are then combined using a mixture distribution, assigning each component a weight. Mixture models have several appealing properties. They can model arbitrary complex distributions while still being easy to work with. For many common operations, such as computing the density, and algorithms, including the one presented in this work, computations can be performed independently for the single components and the mixture distribution. Additionally, adapting the number of components provides an intuitive way of controlling model complexity.

Mixtures of experts [Jacobs et al., 1991] are a natural extension of mixture models to conditional distributions. For those models both the single components as well as the mixture distribution depend on a conditioning variable. The weighting distribution is often realized as a softmax and referred to as gating. Mixtures of experts are a commonly used model class [Yuksel et al., 2012] that share many benefits of general mixture models.

In the following, we formally introduce the types of mixture models used as parametric families throughout this work and review on established approaches to learn them from data, in particular, the Expectation Maximization (EM) algorithm [Dempster et al., 1977].

2.1.1 Gaussian Mixture Models and Mixtures of Experts

Arguably, the Gaussian distribution is the most popular distribution in machine learning. Thus it is natural that Gaussian mixture models (GMMs) are among the most popular mixture models. GMMs consist of a categorical mixture distribution

$$q(z) = \mathrm{Cat}(\pi)$$

and d multivariate Gaussian components

$$q(\mathbf{x}|z_i) = \mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right).$$

For the mixture of expert case various possibilities to formulate the gating and components exist. We want to focus on Gaussian components and softmax gating. For both, we consider models where the conditional parameters depend either linearly or nonlinearly on the conditioning variable. For the softmax, the linear and nonlinear cases are given by

$$q(z|\mathbf{y}) = \mathrm{softmax}(\mathbf{V}\mathbf{y} + \mathbf{v}) \quad \text{or} \quad q(z|\mathbf{y}) = \mathrm{softmax}(\psi_s(\mathbf{y}))$$

respectively. The Gaussian components can be formalized as

$$q(\mathbf{x}|z_i, \mathbf{y}) = \mathcal{N}(\mathbf{W}_i\mathbf{y} + \mathbf{w}_i, \boldsymbol{\Sigma}_i) \quad \text{or} \quad q(\mathbf{x}|z_i, \mathbf{y}) = \mathcal{N}\left(\psi_{\boldsymbol{\mu},i}(\mathbf{y}), \psi_{\boldsymbol{\Sigma},i}(\mathbf{y})\right),$$

for linear and nonlinear models respectively.

2.1.2 Learning Mixture Models

A common objective for fitting parametric model distributions to samples $x^{(j)}$ drawn from an unknown distribution $p(x)$ is maximizing the likelihood of the samples under the model $q(x)$

$$\max_{q(x)} \mathcal{L}(q(x)) = \prod_{j=1}^{N} q\left(x^{(j)}\right).$$

Due to the finite precision of computers, the product over the densities of all data points is problematic in practice. Hence, usually, the log likelihood

$$\max_{q(x)} \log \mathcal{L}(q(x)) = \sum_{j=1}^{N} \log q\left(x^{(j)}\right) \approx \max_{q(\mathbf{x})} \mathbb{E}_{p(x)}[\log q(x)]$$

is maximized. Yet, maximizing the likelihood for Gaussian mixture models is an ill-posed problem [Bishop, 2006]. If the model has more than one component, it is possible that one of the components focuses on a single sample which yields a variance of 0 and thus an infinitely large likelihood. Additionally, a less severe form of this problem, i.e., components focusing not on a single sample but on a small number of them, can lead to severe over-fitting.

Nevertheless, maximum likelihood is a popular approach to fit Gaussian mixture models, and a variety of algorithms for optimizing this objective exists. Besides the EM algorithm, which we will discuss in detail in the next section, gradient-based methods, such as Mixture Density Networks [Bishop, 1994], have been developed. Additionally, Bayesian approaches have been used to alleviate the aforementioned problems of maximum likelihood. By adding a suitable prior, the ill-posed problem formulation can be avoided, and over-fitting alleviated. This approach leads to the Variational Bayes EM algorithm [Attias, 1999; Bishop, 2006].

Expectation Maximization [Dempster et al., 1977] (EM) is a general algorithm for fitting latent variable models using maximum likelihood. The key assumption behind EM is that maximizing the log-likelihood of the observed variables x under the model, $\mathbb{E}_{p(x)}[\log q(x)]$ is hard while maximizing the joint log-likelihood of observed and latent variables $\mathbb{E}_{p(x,z)}[\log q(x,z)]$ is easier. In order to exploit this assumption and work with the joint likelihood, a lower bound to $\mathbb{E}_{p(x)}[\log q(x)]$ is derived. To this end an auxiliary distribution $\tilde{q}(z|x)$ is introduced. The following derivations are valid for all $\tilde{q}(z|x)$, and we will see later how to choose it. The lower bound is given by

$$
\begin{aligned}
\mathbb{E}_{p(x)}[\log q(x)] &= \mathbb{E}_{p(x)}\left[\int \tilde{q}(z|x)\log q(x)dz\right] \\
=&\mathbb{E}_{p(x)}\left[\int \tilde{q}(z|x)\left(\log q(x,z)-\log q(z|x)+\log\tilde{q}(z|x)-\log\tilde{q}(z|x)\right)dz\right] \\
=&\mathbb{E}_{p(x)}\left[\int \tilde{q}(z|x)\left(\log\frac{q(x,z)}{\tilde{q}(z|x)}+\log\frac{\tilde{q}(z|x)}{q(z|x)}\right)dz\right] \\
=&\mathbb{E}_{p(x)}\left[\int \tilde{q}(z|x)\log\frac{q(x,z)}{\tilde{q}(z|x)}dz\right]+\mathbb{E}_{p(x)}\left[\int \tilde{q}(z|x)\log\frac{\tilde{q}(z|x)}{q(z|x)}dz\right] \\
=&\ \underbrace{\mathcal{L}(q,\tilde{q})}_{\text{lower bound}}\ +\underbrace{\mathbb{E}_{p(x)}\left[\mathrm{KL}\left(\tilde{q}(z|x)\,\|\,q(z|x)\right)\right]}_{\geq 0}.
\end{aligned}
$$

Here KL denotes the Kullback-Leibler divergence which we will introduce in detail in section 2.2. Maximizing the lower bound is equivalent to maximizing the joint log-likelihood

$$
\begin{aligned}
\mathcal{L}(q,\tilde{q})) &= \int p(x)\int \tilde{q}(z|x)\log\frac{q(x,z)}{\tilde{q}(z|x)}dzdx = \iint \tilde{p}(x,z)\log\frac{q(x,z)}{\tilde{q}(z|x)}dzdx \\
&= \iint \tilde{p}(x,z)\log q(x,z)dzdx - \iint \tilde{p}(x,z)\log\tilde{q}(z|x)dzdx = \mathbb{E}_{\tilde{p}(x,z)}[\log q(x,z)] - \text{const},
\end{aligned}
$$

where $\tilde{p}(x,z)=p(x)\tilde{q}(z|x)$. Yet, as we only have samples from $p(x)$, $\tilde{q}(z|x)$ needs to be chosen in order to compute this expectation. EM works by iterating two steps, the E-step, and the M-step. During the E-step the model from the previous iteration $q_{\text{old}}(x)$ is used to obtain $\tilde{q}(z|x)$ by setting

$$
\tilde{q}(z|x)=\frac{q_{\text{old}}(x|z)q_{\text{old}}(z)}{q_{\text{old}}(x)}.
$$

With this choice of $\tilde{q}(z|x)$, the upper bound is tight after the E-step since the KL vanishes. The new $\tilde{q}(z|x)$ also allows us to estimate the latent $z^{(j)}$ for all samples $x^{(j)}$. Those are needed to maximize the joint log-likelihood in the next step, i.e., the M-step. Dempster et al. [1977] proved that this procedure monotonically increases the log-likelihood of the observed variables and thus eventually converges to a local maximum.

For mixture models the M-step can be decomposed into individual updates for each of the components and the mixture distribution. The derivations of EM for GMMs, mixtures of experts, and various other latent variable models can be found in [Murphy, 2012]. EM for GMMs is also displayed in algorithm 1.

Algorithm 1: Expectation Maximization for Gaussian Mixture Models [Murphy, 2012]

```
EM-for-GMMS($\{\mathbf{x}^{(j)}\}_{j=1\cdots N}, q(\mathbf{x})$);
Input: Data $\{\mathbf{x}^{(j)}\}_{j=1\cdots N}$, Initial Model $q(\mathbf{x}) = \sum_{i=1}^{k} q(z_i) q(\mathbf{x}|z_i) = \sum_{i=1}^{d} \pi_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
for $i$ in number of iterations do
    $q_{\text{old}}(z) = q(z)$, $q_{\text{old}}(\mathbf{x}|z_i) = q(\mathbf{x}|z_i)$ for all components $i$
    E-Step: compute responsibilities
    $r_{ij} = \tilde{q}\left(z_i|\mathbf{x}^{(j)}\right) = \frac{q_{\text{old}}(x|z) q_{\text{old}}(z)}{q_{\text{old}}(x)} = \frac{\pi_{\text{old},i} \mathcal{N}(\mathbf{x}^{(j)}|\boldsymbol{\mu}_{\text{old},i}, \boldsymbol{\Sigma}_{\text{old},i})}{\sum_{k=1}^{d} \pi_{\text{old},k} \mathcal{N}\left(\mathbf{x}^{(j)}|\boldsymbol{\mu}_{\text{old},k}, \boldsymbol{\Sigma}_{\text{old},k}\right)}$
    M-Step: maximum likelihood for joint $q(\mathbf{x}, z)$
    $r_i = \sum_{j=1}^{N} r_{ij}, \quad \pi_i = \frac{r_i}{N}, \quad \boldsymbol{\mu}_i = \frac{\sum_{j=1}^{N} r_{ij} \mathbf{x}^{(j)}}{r_i}, \quad \boldsymbol{\Sigma}_i = \frac{\sum_{j=1}^{N} r_{ij} \left(\mathbf{x}^{(j)} - \boldsymbol{\mu}_i\right)\left(\mathbf{x}^{(j)} - \boldsymbol{\mu}_i\right)^T}{r_i}$
```

2.2 Kullback-Leibler Divergence and Information-Projection

The Kullback-Leibler Divergence (KL) [Kullback and Leibler, 1951], sometimes also referred to as relative entropy, can be used to measure the divergence between two probability distributions $p_1(x)$ and $p_2(x)$ defined over the same sample space. It is defined as

$$\mathrm{KL}\left(p_1(x) \,\|\, p_2(x)\right) = \int p_1(x) \log \frac{p_1(x)}{p_2(x)} dx.$$

Note that the KL is not a metric in the strict mathematical sense since it is clearly not symmetric. However, using Jensen's inequality, it can be shown that $\mathrm{KL}\left((p_1(x) \,\|\, p_2(x)\right) \geq 0$ and $\mathrm{KL}\left(p_1(x) \,\|\, p_2(x)\right) = 0$ if and only if $p_1(x) = p_2(x)$ almost everywhere.

For conditional distributions $p_1(x|y)$, $p_2(x|y)$ and a distribution $p(y)$ over the conditioning variable y the expected KL is defined as $\mathbb{E}_{p(y)}\left[\mathrm{KL}\left(p_1(x|y) \,\|\, p_2(x|y)\right)\right]$.

2.2.1 Using the KL to fit Probability Distributions

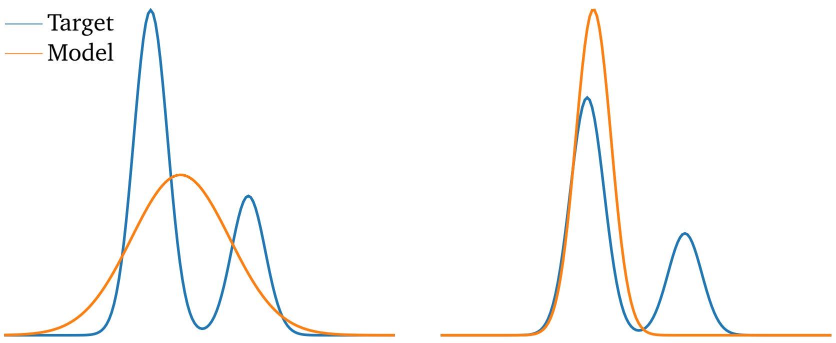
Due to its asymmetry the KL provides two different optimization problems to fit a model distribution $q(x)$ to a target distribution $p(x)$, i.e.,

$$\min_{q(x)} \mathrm{KL}\left(p(x) \,\|\, q(x)\right) \quad \text{and} \quad \min_{q(x)} \mathrm{KL}\left(q(x) \,\|\, p(x)\right).$$

The former is referred to as moment-projection (M-projection) and the latter as information-projection (I-projection). Some authors refer to the first simply as KL and to the latter as reverse KL. Since the KL is minimal, i.e., equal to 0, if and only if $q(x) = p(x)$ it can immediately be seen that both provide the same solution if the model is rich enough to perfectly match the target.

To see how the solutions differ if $q(x)$ is not rich enough to represent $p(x)$ first consider the M-projection

$$\mathrm{KL}\left(p(x) \,\|\, q(x)\right) = \int p(x) \log \frac{p(x)}{q(x)} dx = \int p(x) \log p(x) dx - \int p(x) \log q(x) dx.$$



(a) Moment-Projection **(b)** Information-Projection

Figure 2.1.: Moment- and information-projection of a univariate Gaussian mixture with two components onto the set of univariate Gaussians. The model is clearly not rich enough to represent the target distribution. The density of the M-projection needs to be greater 0 wherever the target distributions density is greater 0, and thus it needs to average over both modes. The I-projection, on the other hand, focuses on a single mode, in this case, the larger one. For the displayed example, focusing on the smaller mode is a spurious local minimum of the I-projection.

The first term is the negative entropy of the target $p(x)$ which is constant and thus irrelevant for the optimization. The second term becomes infinitely large if $q(x) = 0$ for some x with $p(x) > 0$, which forces the model to have a density greater than 0 wherever the target has a density greater than 0. Additionally, one can immediately see that computing the M-projection is equivalent to maximizing the likelihood by noting that

$$\min_{q(x)} \int p(x) \log p(x) dx - \int p(x) \log q(x) dx = \min_{q(x)} - \int p(x) \log q(x) dx = \max_{q(x)} \mathbb{E}_{p(x)} \left[\log q(x) \right].$$

Consider now the I-projection

$$\mathrm{KL}\left(q(x) \,\|\, p(x)\right) = \int q(x) \log \frac{q(x)}{p(x)} dx = \int q(x) \log q(x) dx - \int q(x) \log p(x) dx.$$

While the latter term would be optimal if all density mass of $q(x)$ would be at the point where the density of $p(x)$ is maximal, the former term prevents this by penalizing low entropy of $q(x)$. See Figure 2.1 for additional elaboration on the difference between M-projection and I-projection.

2.2.2 f-Divergences

A commonly used generalization of the KL-divergence is the concept of f-divergences [Ali and Silvey, 1966]. The f-Divergence between two distributions $p_1(x)$ and $p_2(x)$ is defined as

$$D_f(p_1(x) \,\|\, p_2(x)) = \int p_2(x) f\left(\frac{p_1(x)}{p_2(x)}\right) dx,$$

for convex functions $f(t)$ with $f(1) = 0$. For $f(t) = t \log(t)$ we obtain the M-projection of $p_1(x)$ onto $p_2(x)$ and for $f(t) = -\log(t)$ the I-projection of $p_1(x)$ onto $p_2(x)$.

2.3 Density Ratio Estimation

Suppose two probability distributions $p_1(x)$ and $p_2(x)$, of which only samples are available. As suggested by the name, density ratio estimation [Sugiyama et al., 2012a] aims at estimating the ratio between the density of the two distributions, i.e.,

$$r(x) = \frac{p_1(x)}{p_2(x)}.$$

The naive solution to this problem would be estimating the densities of both $p_1(x)$ and $p_2(x)$ individually and use them to compute the ratio. However, dividing by the estimated density of $p_2(x)$ can increase the unavoidable inaccuracy in the estimates. Additionally, especially in high dimensional spaces, density estimation in general is a hard problem and there exist simpler, more reliable approaches for estimating $p_1(x)/p_2(x)$ without the need of explicitly modeling either $p_1(x)$ or $p_2(x)$.

2.3.1 Density Ratio Estimation by Probabilistic Classification

Following [Sugiyama et al., 2012a] we artificially assign labels to the samples of $p_1(x)$ and $p_2(x)$ and introduce the conditional distribution

$$p(x|y=0) = p_1(x) \quad \text{and} \quad p(x|y=1) = p_2(x).$$

Using this distribution, Bayes rule can be applied to obtain

$$r(x) = \frac{p_1(x)}{p_2(x)} = \frac{p(x|y=0)}{p(x|y=1)} = \frac{p(y=0|x)p(x)}{p(y=0)} \frac{p(y=1)}{p(y=1|x)p(x)} = \frac{p(y=1)}{p(y=0)} \frac{p(y=0|x)}{p(y=1|x)}. \tag{2.1}$$

The first factor of Equation 2.1 can be estimated based on the amount of samples available from $p_1(x)$ and $p_2(x)$ and can be neglected if the same number of samples is available for both. The second factor can be estimated using a probabilistic classifier, which reduces the problem of density ratio estimation to probabilistic classification, a well-studied problem, solvable by a variety of established approaches. In this work, we choose a binary logistic regressor.

In general, binary logistic regression estimates the probability $p(y=1|x)$, which in our case corresponds to the probability that a given sample x was drawn from $p_2(x)$. To this end a parametric model $p(y=1|x) = \sigma(\phi(x))$ is used, where $\sigma(x)$ denotes the sigmoid function, i.e., $\sigma(x) = 1/(1+\exp(-x))$ and $\phi(x)$ is an arbitrary parametric function, e.g., a neural network. By using $p(y=0|x) = 1 - p(y=1|x)$ we get

$$\begin{aligned}\frac{p(y=0|x)}{p(y=1|x)} &= \left(1 - \frac{1}{1+\exp(-\phi(x))}\right) \frac{1+\exp(-\phi(x))}{1} \\ &= \left(\frac{1+\exp(-\phi(x))-1}{1+\exp(-\phi(x))}\right) \frac{1+\exp(-\phi(x))}{1} = \exp(-\phi(x)).\end{aligned}$$

In order to train the logistic regressor itself the following optimization problem is solved

$$\min_{C(x)} -\mathbb{E}_{p_2(x)}\left[\log C(x)\right] - \mathbb{E}_{p_1(x)}\left[\log\left(1 - C(x)\right)\right].$$

This objective is the well known binary cross entropy.

2.3.2 Density Ratio Estimation by Bregman Divergences

Bregman divergences [Bregman, 1967] are distance measures over elements of convex sets defined by

$$\mathrm{BR}_f(p_1||p_2) = f(p_1) - (f(p_2) + f'(p_2)(p1 - p2))$$

for strictly convex functions $f(t)$ and their derivatives $f'(t)$. Intuitively $\mathrm{BR}_f(p_1 \parallel p_2)$ is the difference between $f(p_1)$ and the first order Taylor expansion of f around p_2 at p_1. Not all Bregman divergences are metrics, however, many common metrics can be expressed as a special case of a Bregman divergence. For example, the squared euclidean distance can be obtained with $f(t) = \parallel t \parallel^2$.

Sugiyama et al. [2012b] derived a framework to obtain density ratio estimators $r(x)$ based on samples by minimizing the Bregman divergence to the true density ratio estimator $r^*(x) = p(x)/q(x)$. By using $p(x) = r^*(x)q(x)$ they obtain

$$\begin{aligned}\mathrm{BR}_f\left(r^*(x) \parallel r(x)\right) &= \int q(x)\big(f\left(r^*(x)\right) - \left(f(r(x)) + f'(r(x))\left(r^*(x) - r(x)\right)\right)\big)\,dx \\ &= \int q(x)f(r^*(x)) - q(x)f(r(x)) - q(x)f'(r(x))r^*(x) + q(x)f'(r(x))r(x)dx \\ &= \int q(x)\big(f'(r(x))r(x) - f(r(x))\big)\,dx - \int p(x)f'(r(x))dx + \text{const.}\end{aligned}$$

It becomes clear that a density ratio estimator can be obtained by solving

$$\min_{r(x)} \mathbb{E}_{q(x)}\left[f'(r(x))r(x) - f(r(x))\right] - \mathbb{E}_{p(x)}\left[f'(r(x))\right]. \tag{2.2}$$

Based on this optimization problem a whole family of density ratio estimation techniques is derived and by using

$$f(t) = t\log(t) - (1+t)\log(1+t)$$

the previously introduced case of density ratio estimation by logistic regression can be obtained.

2.4 Generative Adversarial Networks

First introduced by Goodfellow et al. [2014], Generative Adversarial Networks (GANs), are nowadays among the most popular deep models. The main idea is to train two competing networks, a generator and a discriminator. The generator's task is generating new data from random noise while the discriminators objective is to distinguish between the real data from the training set and the fake data generated by the generator. Generator and discriminator are trained in an alternating fashion and need to successively improve to keep up with their opponent. Given perfect conditions, i.e., rich enough models, infinite training data, and perfect optimization, this procedure results in an equilibrium where the generator produces samples indistinguishable from the real data, and the discriminator is maximally confused.

More formally, the generator, i.e., a latent variable model $q(x) = \int q(x|z)q(z)dz$, and the discriminator $D(x)$, play a two player min-max game

$$\min_{q(x)} \max_{D(x)} v(q(x), D(x)) \tag{2.3}$$

with objective $v(q(x), D(x))$. Usually, both generator and discriminator are realized by deep neural networks. In the case of the generator, this network takes a latent variable z, usually sampled from a fixed uniform distribution $q(z)$, as input and produces an output x. Thus, the generator network implements the conditional distribution $q(x|z)$. One of the main benefits of GANs is that they work solely based on samples of $q(x)$ and do not need to explicitly compute the density of $q(x|z)$ or $q(x)$, which allows the usage of powerful models, even if their density is intractable.
Solving min-max games as in Equation 2.4 analytically or numerically with a single optimization is intractable. Thus, one optimizes alternatingly w.r.t. $q(x|z)$ and $D(x)$ while keeping the other fixed.
In their original work Goodfellow et al. [2014] proposed realizing the discriminator as a neural network with sigmoid output and used

$$\min_{q(x)} \max_{D(x)} v(q(x), D(x)) = \mathbb{E}_{p(x)} [\log D(x)] + \mathbb{E}_{q(x)} [\log (1 - D(x)))] \tag{2.4}$$

as the min-max objective.

The popularity of GANs has led to a variety of works proposing different objectives v and a large number of other extensions aiming at more stable training, faster convergence and other aspects. In the following, we will focus on works and aspects relevant to this work.

2.4.1 GANs for f-Divergences

Nowozin et al. [2016] derived an objective that allows training GANs such that an arbitrary f-divergence between the true distribution $p(x)$ and the generator distribution $q(x)$ is minimized. To this end they use a lower bound of the f-divergence [Nguyen et al., 2010],

$$D_f(p(x) \parallel q(x)) \geq \sup_{T(x)} \mathbb{E}_{p(x)} [T(x)] - \mathbb{E}_{q(x)} [f^*(T(x))], \tag{2.5}$$

where $f^*(x) = \sup_u ux - f(u)$, which is known as the Fenchel conjugate [Hiriart-Urruty and Lemaréchal, 2012]. The variational function $T(x)$ can be seen as a discriminator and is parameterized by $g_f(\psi_D(x))$, where $\psi(x)$ is a neural network and $g_f(t)$ denotes the output function. Inserting $g_f(\psi_D(x))$ into Equation 2.5 yields

$$\min_{q(x)} \max_{\psi_D(x)} v(q(x), \psi_D) = \mathbb{E}_{p(x)} \left[g_f(\psi_D(x)) \right] + \mathbb{E}_{q(x)} \left[f^*(g_f(\psi_D(x))) \right].$$

The choice of $g_f(t)$ is arbitrary, yet it should exclusively output values within the domain of f^*, since its output values are subsequently fed into f^*. Nowozin et al. [2016] propose g_f for different f-divergences. Examples can be found in Table 2.1. Note that all of them are monotone increasing functions where large outputs correspond to samples from $p(x)$ and small outputs to samples from $q(x)$.

2.4.2 GANs and Density Ratio Estimation

Consider again the original GAN objective stated in Equation 2.4. When optimizing with respect to $D(x)$ and keeping $q(x|z)$ fixed the objective simplifies to

$$\min_{D(x)} -\mathbb{E}_{p(x)} [\log D(x)] - \mathbb{E}_{q(x)} [\log (1 - D(x))].$$

Divergence	$f(t)$	$f^*(t)$	$g(v)$
KL (M-Projection)	$t\log(t)$	$\exp(t-1)$	v
KL (I-Projection)	$-\log(t)$	$-1-\log(-t)$	$-\exp(-v)$
Jensen-Shannon	$-(t+1)\log\frac{1+t}{2}+t\log t$	$-\log(2-\exp(t))$	$\log(2)-\log(1+\exp(-v))$
Pearson χ^2	$(t-1)^2$	$\frac{1}{4}t^2+t$	v

Table 2.1.: Generating functions $f(t)$ for popular f-divergences, together with their Fenchel conjugate $f^*(t)$ and output activations $g(v)$ as proposed by Nowozin et al. [2016]. Note that the f-GAN corresponding to the Jensen-Shannon divergence is equivalent to the original GAN [Goodfellow et al., 2014], up to constants.

This objective is again the binary cross entropy. Thus, it becomes clear that the discriminator can be seen as a probabilistic classifier, aiming at classifying samples as true or fake. The same classifier was used for density ratio estimation in section 2.3.1, which relates GANs and density ratio estimation. While Goodfellow et al. [2014] did not elaborate on this connection in their original work, several other authors exploited it to derive alternative GAN formulations [Nowozin et al., 2016; Uehara et al., 2016; Poole et al., 2016].

It can be shown that the optimal $T(x)$ in Equation 2.5 is given by $T(x) = f'(r(x))$ with $r(x) = p(x)/q(x)$, i.e., the derivative of f at the density ratio between target and model distribution. While Nowozin et al. [2016] note this fact, they do not exploit it. Uehara et al. [2016], on the other hand, exploit the fact in their b-GAN approach. Given the true density ratio estimate the bound becomes tight and the f-divergence can be rewritten as

$$\begin{aligned} D_f(p(x)\,\|\,q(x)) &= \sup_{r(x)} \mathbb{E}_{p(x)}\left[f'(r(x))\right] - \mathbb{E}_{q(x)}\left[f^*(f'(r(x))\right] \\ &= \sup_{r(x)} \mathbb{E}_{p(x)}\left[f'(r(x))\right] - \mathbb{E}_{q(x)}\left[f'(r(x))r(x) - f(r(x))\right]. \end{aligned}$$

Interestingly, this optimization problem is equivalent to Equation 2.2. Thus, it becomes clear how a GAN can be formulated by alternating steps, minimizing the f-divergence between model and target distribution and minimizing the Bregman divergence between true and estimated density ratio. Formally, the corresponding adversarial objective is given by,

$$\min_{q(x)} \max_{r(x)} v(q(x), r(x)) = \mathbb{E}_{p(x)}\left[f'(r(x))\right] - \mathbb{E}_{q(x)}\left[f'(r(x))r(x) - f(r(x))\right].$$

Note that the same f needs to be used for both the Bregman and f-divergence. While the requirements regarding f for both divergences are very similar, and thus all commonly used f-divergences can be minimized, they can not be combined arbitrarily. Uehara et al. [2016] suggest using $f(t) = 0.5t^2 - 0.5$ for the density ratio estimation, which is equivalent to Least Squares Importance Fitting (LSIF) [Yamada et al., 2013]. Some authors claim that LSIF is a robust choice for density ratio estimation [Yamada et al., 2013; Dawid et al., 2016]. However, the corresponding f-divergence is the Pearson χ^2 divergence, which is a suboptimal choice for typical GAN tasks, such as image generation [Huszár, 2015].

Poole et al. [2016] also note this connection between the density ratio and f-divergence. Based on the fact that every f-divergence can be estimated using samples of one distribution and the density ratio estimate, they derive alternative generator updates for the f-GAN.

Yet, while providing interesting theoretical insights, the works of both, Poole et al. [2016] and Uehara et al. [2016], lack a reasonable empirical evaluation and in particular a comparison to the original f-GAN or other generative adversarial approaches.

2.4.3 GANs for Mixture Models

When working with generative adversarial approaches the model density is not required, or of interest. One only aims at efficiently generating high-quality samples. In this work, on the other hand, we aim at learning the density and use mixture models to get tractable densities. To the best of our knowledge, there exist no studies which use a generative adversarial approach to learn mixture models.
In order to learn a GMM with a generative adversarial approach, one needs to apply the reparameterization trick [Kingma and Welling, 2013] in order to allow back-propagation through the sampling process. For the Gaussian components, reparameterization is straight forward. However, for the categorical mixture distribution it is not and approximations, such as the recently proposed Gumble softmax [Jang et al., 2017], need to be employed.

Chen et al. [2016] provide an alternative approach. The main idea of their approach, InfoGAN, is to maximize the mutual information between a subset of the latent variables z and the observations x jointly with the original generator objective. This extended objective allows them to learn disentangled latent representations where single latent variables correspond to salient features of the data. For example, they demonstrate how their approach learns latent variables corresponding to writing style, shape, and digits when trained on the well known MNIST dataset.

The mutual information is defined as

$$\mathrm{I}(x,z) = \iint q(x,z) \log \frac{q(x,z)}{q(x)q(z)} dz dx$$

which is equivalent to $I(x,z) = \mathrm{H}(x) - \mathbb{E}_{q(z)}[\mathrm{H}(x|z)] = \mathrm{H}(z) - \mathbb{E}_{q(x)}[\mathrm{H}(z|x)]$. The only density available in the classical GAN setup is $q(z)$, and thus Chen et al. [2016] rely on a variational approximation of $q(z|x)$ to compute a lower bound to the mutual information. However, when working with GMMs both $q(x)$ and $q(x|z)$ are efficiently computable, and thus the mutual information can easily be obtained by a sample based approximation.
Chen et al. [2016] used the original GAN objective [Goodfellow et al., 2014]. Yet, their idea can easily be used to learn GMMs with arbitrary GAN objectives such as the f-GAN. However, the resulting approach may not converge to the corresponding f-divergence only the individual components do.

Both methods to learn GMMs with GANs will serve as baselines to our approach.

2.5 Reinforcement Learning and Policy Search

Reinforcement Learning [Sutton and Barto, 2018] aims at learning optimal behavior by interaction with an environment. The learned behavior is described using a policy $\pi(a|s)$, i.e., a conditional distribution over possible actions a given a state s. Which behavior is optimal is defined by a reward function $r(s,a)$, which is typically unknown and can only be sampled by interacting with the environment.

There are several characteristic issues for reinforcement learning problems which need to be addressed to successfully learn good policies. Usually, evaluating the reward for a given policy consist of running the

policy in the environment, which can be computationally very expansive for simulated environments. For real-world scenarios, it is usually even more time consuming. Thus, in order to be applicable to interesting problems reinforcement learning algorithms need to be very sample efficient.
Another inherent issue of reinforcement learning is known as the exploration-exploitation trade-off. It describes the dilemma of choosing between exploiting the currently best, known behavior and exploring the environment further. While further exploration might cause sub-optimal rewards, it is also the only way to find novel, better solutions. Thus, too little exploration might lead to premature convergence to spurious local optima, while uncontrolled exploration can cause oscillations and divergence in the optimization process and, in the case of real world systems, even hazardous behavior.

One of the main classes of reinforcement learning approaches is policy search [Deisenroth et al., 2013]. Those approaches directly optimize a parametric policy $\pi(a|s,\theta)$ to obtain maximal reward. One possible solution to find optimal parameters θ is stochastic search. To this end a search distribution $q(\theta)$ is defined over the space of all parameters and subsequently optimized

$$\max_{q(\theta)} \int q(\theta)R(\theta)d\theta.$$

It often makes sense to generalize a policy across multiple contexts to make use of similarities between tasks. Consider, for example, a robot tasked with reaching a specific goal point. Learning a completely different policy for each possible goal point is inefficient or even impossible if there are infinitely many possible goal positions, which is the case for continuous context spaces. A better solution is to model the policy as a conditional distribution dependent on the goal point. One way to solve these kinds of problems is contextual policy search

$$\max_{q(\theta|c)} \int p(c) \int q(\theta|c)R(\theta,c)d\theta dc,$$

where $p(c)$ denotes the distribution over contexts. Usually, this distribution is assumed to be unknown, and only samples of it are available.

To solve these problems standard black-box stochastic search methods such as CMA-ES [Hansen et al., 2003] can be employed. Yet, those are agnostic to the aforementioned issues with reinforcement learning and specialized approaches have been derived to solve policy search by stochastic search [Abdolmaleki et al., 2015]. Those approaches use information theoretic insides to make stochastic search more sample efficient and account for the exploration-exploitation trade-off.

2.5.1 Information Theoretic Policy Search

Information theoretic insights have been used to make policy search in general and stochastic search in particular more stable and sample efficient [Peters et al., 2010; Abdolmaleki et al., 2015].
Peters et al. [2010] introduced REPS, a policy search algorithm that constraints the change of the policy during each update by bounding the KL between old and new policy. This constraint bounds the loss of information during the update which helps the algorithm to converge faster and with fewer samples.
MORE [Abdolmaleki et al., 2015], a stochastic search algorithm, does not only constrain the change during the update but also the amount the entropy in the search distribution can be reduced. Adding a

constraint on the entropy loss prevents the search distribution from collapsing its variance which may result in premature convergence to spurious local optima. MORE works by iterativly solving

$$\max_{q(\theta)} \int q(\theta)R(\theta)d\theta \tag{2.6}$$
$$\text{s.t.} \quad \mathrm{KL}\left(q(\theta)\,\|\,q_{\text{old}}(\theta)\right) \leq \epsilon, \quad \mathrm{H}(q(\theta)) \geq \beta, \quad \int q(\theta)d\theta = 1$$

where $q_{\text{old}}(\theta)$ denotes the search distribution prior to the update.
The first constraint is the aforementioned KL constraint, bounding the change during the update. The second constraint bounds the loss in entropy, to this end β is set to be $\beta = \mathrm{H}(q_{\text{old}}(\theta)) - \beta_{\text{loss}}$. Both ϵ and β_{loss} are hyper-parameters of the algorithm. The third constraint is needed to ensure the new search distribution is properly normalized.

The first step in solving the optimization stated in Equation 2.6 is minimizing the dual problem

$$\begin{aligned} g(\eta,\omega) &= \eta\epsilon - \omega\beta + (\eta+\omega)\log \int q_{\text{old}}(\theta)^{\frac{\eta}{\eta+\omega}} \exp\left(\frac{R(\theta)}{\eta+\omega}\right) d\theta \\ &= \eta\epsilon - \omega\beta + (\eta+\omega)\log \int \exp\left(\frac{\eta \log q_{\text{old}}(\theta) + R(\theta)}{\eta+\omega}\right) d\theta. \end{aligned}$$

Here, η denotes the Lagrangian multiplier corresponding to the KL constraint and β the Lagrangian multiplier corresponding to the entropy constraint. Given the optimal Lagrangian multipliers, the new search distribution is given by

$$q(\theta) \propto q_{\text{old}}(\theta)^{\frac{\eta}{\eta+\omega}} \exp\left(\frac{R(\theta)}{\eta+\omega}\right) = \exp\left(\frac{\eta \log q_{\text{old}}(\theta) + R(\theta)}{\eta+\omega}\right). \tag{2.7}$$

Since the reward function $R(\theta)$ is typically not known and only samples are available, Abdolmaleki et al. [2015] propose approximating it with a local surrogate $R(\theta) \approx \hat{R}(\theta) = \hat{\mathbf{r}}^T \boldsymbol{\psi}_{\text{comp}}(\theta)$. The features $\boldsymbol{\psi}_{\text{comp}}(\theta)$ are chosen such that they are compatible to the search distribution [Kakade, 2002], i.e., of the same form as the distributions sufficient statistics. For exponential family distributions, the parameters of the surrogate $\hat{\mathbf{r}}$, together with the natural parameters of the old search distribution, $\mathbf{N}_{\text{old}}$, can be used to obtain the natural parameters of the new search distribution N by

$$\mathbf{N} = \frac{1}{\eta+\omega}\left(\eta \mathbf{N}_{\text{old}} + \hat{\mathbf{r}}\right).$$

If, for example, the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is considered, the natural parameters are the precision matrix $\mathbf{Q} = \boldsymbol{\Sigma}^{-1}$ and $\mathbf{q} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$. The corresponding compatible surrogate is a quadratic function of the form

$$\hat{R}(\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{\theta}^T \hat{\mathbf{R}} \boldsymbol{\theta} + \hat{\mathbf{r}}^T \boldsymbol{\theta} + \hat{r}_0.$$

Using the surrogate's parameters the natural parameters of the Gaussian distribution can be updated by

$$\mathbf{Q} = \frac{1}{\eta+\omega}\left(\eta \mathbf{Q}_{\text{old}} + \hat{\mathbf{R}}\right) \quad \text{and} \quad \mathbf{q} = \frac{1}{\eta+\omega}\left(\eta \mathbf{q}_{\text{old}} + \hat{\mathbf{r}}\right).$$

Alternatively, Equation 2.6 can be solved using a sample based approach, similar to [Peters et al., 2010; Daniel et al., 2016]. To this end, note that Equation 2.7 can be rewritten to

$$q(\theta) \propto q_{\text{old}}(\theta) \exp\left(\frac{-\omega \log q_{\text{old}}(\theta) + R(\theta)}{\eta + \omega}\right).$$

It becomes clear, that the new policy can be obtained by fitting it to weighted samples of the old policy using weighted maximum likelihood.

In the contextual case the expected KL and entropy are constrained instead, which yields

$$\max_{q(\theta|c)} \int p(c) \int q(\theta|c) R(\theta, c) d\theta dc \tag{2.8}$$

$$\text{s.t.} \quad \mathbb{E}_{p(c)}\left[\mathrm{KL}\left(q(\theta|c) \parallel q_{\text{old}}(\theta|c)\right)\right] \leq \epsilon, \quad \mathbb{E}_{p(c)}\left[\mathrm{H}(q(\theta|c))\right] \geq \beta, \quad \forall c: \int q(\theta|c) d\theta = 1.$$

The main problem with this formulation is the inner integral over the parameters. A sample based approximation is only reasonable if sufficiently many samples can be evaluated efficiently for each context, which is usually not the case in a reinforcement learning setup. There exist multiple ways to handle this problem.

For simple distributions, closed form solutions of the inner integral and updates based on reward surrogates are still possible. Akrour et al. [2018] derived them for linear Gaussians of the form $q(\boldsymbol{\theta}|\mathbf{c}) = \mathcal{N}(\mathbf{W}\boldsymbol{\theta} + \mathbf{w}, \boldsymbol{\Sigma})$.

An alternative is optimizing the joint $q(\theta, c)$ instead of the conditional, which allows working with a single expectation over the joint instead of individual expectations over context and search distribution. Approximating such an expectation requires fewer samples. Yet, when working with the joint one needs to ensure that the updated joint still reproduces the original context distribution, i.e., $\int q(\theta, c) d\theta = p(c)$ for all c. Adding these constraints results in an infinite number of constraints which are usually approximated by average feature matching [Peters et al., 2010; Deisenroth et al., 2013; Abdolmaleki et al., 2016].

Recently, Abdolmaleki et al. [2018] employed an actor-critic scheme to learn a global, efficiently evaluable approximation of the reward which they used to approximate the integral.

2.6 Variational Inference

For many interesting distributions $p(x)$, inference in closed form is intractable. A common approach, known as variational inference, is to approximate $p(x)$ with a parametric distribution form a tractable family $q(x)$ by means of optimization. Arguably, the most common objective for variational inference approaches is computing the I-projection of $q(x)$ onto $p(x)$. A well-known example is the mean field approach [Opper and Saad, 2001].

All variational inference methods have in common that they assume access to the unnormalized density of $p(x)$, i.e., some function $\tilde{p}(x) = cp(x)$ with a constant c. This assumption is fundamentally different from the assumption we make for our approach, i.e., only samples of $p(x)$ are available. Yet, a recent variational inference approach, Variational Inference by Policy Search (VIPS) [Arenz et al., 2018], is closely related to this work.

2.6.1 Variational Inference by Policy Search (VIPS)

VIPS [Arenz et al., 2018] aims at finding the I-projection of a model $q(x)$ onto an intractable, true distribution $p(x)$, $\mathrm{KL}(q(x) \parallel p(x))$ under the assumption of access to the unnormalized density of the target distribution $p^*(x) = cp(x)$ with $c > 0$. For the minimization of the KL, $p^*(x)$ can be used instead of $p(x)$ since

$$\mathrm{KL}\left(q(x) \,\|\, p(x)\right) = \int q(x) \log \frac{q(x)}{p(x)} dx = \int q(x) \log \frac{q(x)}{p^*(x)} dx + \log c.$$

For the model, Arenz et al. [2018] assume a latent variable model of the form $q(x) = \int q(x|z)q(z)dz$. By introducing an auxiliary distribution $\tilde{q}(z|x)$ the objective can be decomposed into an upper bound and an expected KL term

$$\int q(x) \log \frac{q(x)}{p^*(x)} dx = U(q, \tilde{q}, p^*) - \underbrace{\mathbb{E}_{q(x)}\left[\mathrm{KL}\left(q(z|x) \,\|\, \tilde{q}(z|x)\right)\right]}_{\geq 0}.$$

where the upper bound is given by

$$\begin{aligned} U_{\text{vips}}(q, \tilde{q}, p^*) &= \iint q(x|z)q(z)\left(\log \frac{q(x|z)q(z)}{p^*(x)} - \log \tilde{q}(z|x)\right) dx dz \\ &= \iint q(x|z)q(z)\left(-\log p^*(x) - \log \tilde{q}(z|x)\right) dx dz - \mathrm{H}(q(z)) - \mathbb{E}_{q(z)}\left[\mathrm{H}(q(x|z))\right]. \end{aligned} \quad (2.9)$$

Similar to Expectation Maximization (EM), this upper bound is minimized by iterating E- and M-steps. During the E-step, $\tilde{q}(z|x)$ is approximated with the model from the previous iteration by

$$\tilde{q}(z|x) = \frac{q_{\text{old}}(x|z)q_{\text{old}}(z)}{q_{\text{old}}(x)}.$$

The exact form of the M-step depends on the specific latent variable model used and Arenz et al. [2018] derive it only for GMMs. They rewrite the minimization of the upper bound as an information theoretic policy search problem

$$\max_{q(\mathbf{x}|z), q(z)} \underbrace{\sum q(z_i) \int q(\mathbf{x}|z_i)}_{\text{(hierachical) search distribution}} \overbrace{\left(\log p^*(\mathbf{x}) + \log \tilde{q}(z_i|\mathbf{x})\right)}^{\text{reward}} d\mathbf{x} + \underbrace{\mathrm{H}(q(z)) + \mathbb{E}_{q(z)}\left[\mathrm{H}(q(\mathbf{x}|z))\right]}_{\text{entropy terms}}.$$

This objective can be decomposed into individual update steps for the mixture distribution $q(z)$ and the individual components $q(\mathbf{x}|z_i)$. Opposed to the policy search methods discussed above, the entropy terms do not enter the objective through additional constraints but as part of the original objective. By adding KL constraints, each of the individual updates results in an instance of MORE, where ω, i.e., the Lagrangian multiplier corresponding to the entropy constraint, is not optimized but a set to $\omega = 1$.

A usual assumption for variational inference is that evaluating the normalized density $p^*(x)$ is computationally expensive and thus the algorithm has to work with a limited amount of samples. VIPS addresses this issue by employing importance sampling to reuse samples over multiple iterations.

3 Expected Information Maximization

We now introduce Expected Information Maximization (EIM), i.e., our approach for finding the I-projection between a model $q(x)$ and a true distribution $p(x)$

$$\min_{q(x)} \mathrm{KL}\left(q(x) \parallel p(x)\right)$$

based solely on samples of $p(x)$. To this end we first re-derive the upper bound used by VIPS for general latent variable models of the form $q(x) = \int q(x|z)q(z)dz$. We reformulate this upper bound such that the unknown part, i.e., $p(x)$, only appears in a log density ratio term together with the model $q(x)$, which can be approximated using the density ratio estimation techniques introduced above. Finally, we show how to interpret the obtained objective as an information theoretic policy search problem, which can be efficiently optimized using the methods introduced in section 2.5.
Afterwards, we will repeat the derivations for conditional models $q(x|y)$ and true distributions $p(x|y)$. This derivation consist of the same steps as the derivation for marginal distributions and will result in an algorithm capable of finding the expected I-projection

$$\min_{q(x|y)} \mathbb{E}_{p(y)}\left[\mathrm{KL}\left(q(x|y) \parallel p(x|y)\right)\right].$$

For brevity, we will only state the key results of the derivations here and refer to appendix A.1 for more details.

3.1 Upper Bound Objective

As already mentioned we assume a latent variable model of the form $q(x) = \int q(x|z)q(z)dz$. Additionally we introduce an auxiliary distribution $\tilde{q}(z|x)$. Note, that the derivations are valid for arbitrary $\tilde{q}(z|x)$. By using Bayes rule we get

$$\begin{aligned}
&\mathrm{KL}\left(q(x) \parallel p(x)\right) = \int q(x) \log \frac{q(x)}{p(x)} dx \\
&= \iint q(x|z)q(z)\left(\log \frac{q(x|z)q(z)}{p(x)} - \log q(z|x) + \log \tilde{q}(z|x) - \log \tilde{q}(z|x)\right) dz dx \\
&= \iint q(x|z)q(z)\left(\log \frac{q(x|z)q(z)}{p(x)} - \log \tilde{q}(z|x)\right) dx dz - \int q(x) \int q(z|x) \log \frac{q(z|x)}{\tilde{q}(z|x)} dx dz \\
&= \underbrace{U(q, \tilde{q}, p)}_{\text{upper bound}} - \underbrace{\mathbb{E}_{q(x)}\left[\mathrm{KL}\left(q(z|x) \parallel \tilde{q}(z|x)\right)\right]}_{\geq 0}. \qquad (3.1)
\end{aligned}$$

Since the expected KL is always non-negative, it can clearly be seen that $U(q, \tilde{q}, p)$ is an upper bound of the original objective $\mathrm{KL}\left(q(x) \parallel p(x)\right)$. This bound is equivalent to the bound used in VIPS [Arenz et al., 2018], except that it depends on the density of $p(x)$ and not the unnormalized density. The bound is tight if the expected KL term vanishes which happens if and only if $q(z|x) = \tilde{q}(z|x)$ for all x. This approach has strong similarities with EM as well as VIPS and similar to those approaches, EIM optimizes its objective iteratively using alternating E-steps and M-steps.

In each iteration we denote the old model, i.e., the output of the previous iteration, by $q_{\text{old}}(x) = \int q_{\text{old}}(x|z) q_{\text{old}}(z) dz$. During the E-step we tighten the bound by setting

$$\tilde{q}(z|x) = \frac{q_{\text{old}}(x|z) q_{\text{old}}(z)}{q_{\text{old}}(x)}. \tag{3.2}$$

During the M-step the upper bound is minimized. To this end, Equation 3.2 is plugged into the upper bound objective which simplifies to

$$\begin{aligned} U(q, \tilde{q}, p) &= \iint q(x|z) q(z) \left(\log \frac{q(x|z) q(z)}{p(x)} - \log \frac{q_{\text{old}}(x|z) q_{\text{old}}(z)}{q_{\text{old}}(x)} \right) dx dz \\ &= \iint q(x|z) q(z) \left(\log \frac{q_{\text{old}}(x)}{p(x)} + \log \frac{q(x|z)}{q_{\text{old}}(x|z)} + \log \frac{q(z)}{q_{\text{old}}(z)} \right) dx dz \\ &= \int q(z) \int q(x|z) \log \frac{q_{\text{old}}(x)}{p(x)} dx dz + \mathbb{E}_{q(z)} \left[\text{KL} \left(q(x|z) \,||\, q_{\text{old}}(x|z) \right) \right] + \text{KL} \left(q(z) \,||\, q_{\text{old}}(z) \right). \end{aligned} \tag{3.3}$$

We still cannot directly minimize this objective as it depends on $p(x)$. Yet, we can employ density ratio estimation to estimate $\log \left(q_{\text{old}}(x) / p(x) \right)$.

3.2 Using Density Ratio Estimator for Upper Bound

As described in section 2.3.1 we train a logistic regressor to classify between samples from $p(x)$ and $q_{\text{old}}(x)$ and get

$$\frac{q_{\text{old}}(x)}{p(x)} = \exp(-\phi(x)) \Leftrightarrow \log \frac{q_{\text{old}}(x)}{p(x)} = -\phi(x),$$

where $\phi(x)$ are the logits of the logistic regressor. Before each M-step the classifier needs to be retrained to account for the update during the last M-step, which can be viewed as an additional part of the E-step. By plugging the density ratio estimator into the upper bound we obtain a solvable optimization problem

$$\min_{q(x|z), q(z)} - \int q(z) \int q(x|z) \phi(x) dx dz + \mathbb{E}_{q(z)} \left[\text{KL} \left(q(x|z) \,||\, q_{\text{old}}(x|z) \right) \right] + \text{KL} \left(q(z) \,||\, q_{\text{old}}(z) \right). \tag{3.4}$$

As discussed in section 2.4.2 the density ratio estimation is closely related to the concept of a discriminator in a generative adversarial setup. The usage of density ratio estimation relates EIM to Generative Adversarial Networks (GANs). Yet, there are also fundamental differences. We elaborate on the connections to GANs in section 5.3.

3.3 Efficient Solutions for the M-Step

The exact form of the optimization problem 3.4 depends on the form of the latent variable model. While there exists a variety of black box approaches to solve the resulting optimization problem we want to point out similarities to the information theoretic policy search problems discussed in section 2.5. Those similarities can be exploited to obtain efficient solutions to the M-step update. First, we reformulate the problem as a maximization problem by inverting the sign and obtain

$$\max_{q(x|z),q(z)} \iint \underbrace{q(z)q(x|z)}_{\text{(hierarchical) search distribution}} \overbrace{\phi(x)}^{\text{reward}} dxdz - \underbrace{\mathbb{E}_{q(z)}\left[\mathrm{KL}\left(q(x|z) \parallel q_{\text{old}}(x|z)\right)\right] - \mathrm{KL}\left(q(z) \parallel q_{\text{old}}(z)\right)}_{\text{KL terms}}.$$

The first part is equal to the standard policy search problem, i.e., maximize the expected reward under the search distribution, where the log density ratio acts as a reward. Similar to the information theoretic policy search there are also KL terms between the distribution to optimize and the distribution prior to the optimization. Yet, those enter the optimization problem through the objective, not by additional constraints. Additionally, exploiting the hierarchical structure of the search distribution can further simplify the M-step. In chapter 4 we demonstrate how those similarities and the model's structure can be used to obtain efficient updates for Gaussian mixture models.

3.4 Conditional Distributions

We repeat the derivations for conditional distributions $p(x|y)$ and conditional latent variable models $q(x|y) = \int q(x|z,y)q(z|y)dz$. We start by deriving an upper bound for the expected KL by introducing an auxiliary distribution $\tilde{q}(z|x,y)$

$$\mathbb{E}_{p(y)}\mathrm{KL}\left(q(x|y) \parallel p(x|y)\right) = \underbrace{U_{\text{cond}}(q,\tilde{q},p)}_{\text{upper bound}} - \underbrace{\mathbb{E}_{p(y),q(x|y)}\left[\mathrm{KL}\left(q(z|x,y) \parallel \tilde{q}(z|x,y)\right)\right]}_{\geq 0}. \tag{3.5}$$

The derivations closely follow the derivations for the marginal case and can be found in appendix A.2. Again, we work with an EM-like procedure and tighten the bound during the E-step by setting $\tilde{q}(z|x,y) = q_{\text{old}}(x|z,y)q_{\text{old}}(z|y)/q_{\text{old}}(x|y)$. Replacing the auxiliary distribution in Equation 3.5 by the E-step yields

$$\begin{aligned} U_{\text{cond}}(q,\tilde{q},p) = &\iiint p(y)q(z|y)q(x|z,y)\log\frac{q_{\text{old}}(x|y)}{p(x|y)}dxdzdy \\ &+ \mathbb{E}_{p(y),q(z|y)}\left[\mathrm{KL}\left(q(x|z,y) \parallel q_{\text{old}}(x|z,y)\right)\right] + \mathbb{E}_{p(y)}\left[\mathrm{KL}\left(q(z|y) \parallel q_{\text{old}}(z|y)\right)\right]. \end{aligned} \tag{3.6}$$

In order to estimate the log-density ratio between $q_{\text{old}}(x|y)$ and $p(x|y)$ we train a classifier on the joint distributions $q_{\text{old}}(x,y)$ and $p(x,y)$. The negative logits of such a classifier can be used to approximate the log-density ratio since

$$-\phi(x,y) = \log\frac{q_{\text{old}}(x,y)}{p(x,y)} = \log\frac{q_{\text{old}}(x|y)p(y)}{p(x|y)p(y)} = \log\frac{q_{\text{old}}(x|y)}{p(x|y)}.$$

Inserting $\phi(x,y)$ into the conditional upper bound yields the M-step for conditional distributions

$$\begin{aligned} \min_{q(x|z,y),q(z|y)} -&\iiint p(y)q(z|y)q(x|z,y)\phi(x,y)dxdzdy \\ &+ \mathbb{E}_{p(y),q(z|y)}\left[\mathrm{KL}\left(q(x|z,y) \parallel q_{\text{old}}(x|z,y)\right)\right] + \mathbb{E}_{p(y)}\left[\mathrm{KL}\left(q(z|y) \parallel q_{\text{old}}(z|y)\right)\right]. \end{aligned} \tag{3.7}$$

As we shall see in chapter 4 methods from information theoretic contextual policy search can be exploited to efficiently solve this optimization problem.

4 EIM for Mixture Models

We demonstrate how to use EIM for two specific latent variable models. First, we use EIM for Gaussian mixture models (GMMs) and second, we use the conditional version of EIM to learn mixtures of experts. We again refer to the appendix for detailed derivations of the duals and closed form updates.

4.1 Gaussian Mixture Models

Recall that GMMs are given by d components, each a Gaussian distribution $q(\mathbf{x}|z_i) = \mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)$, and a categorical mixture distribution $q(z) = \mathrm{Cat}(\pi)$.
We exploit the structure of the model and split the M-step into individual parts for the mixture distribution and the individual components. For the individual updates, we add the same information theoretic constraints that were added in MORE [Abdolmaleki et al., 2015]. Due to the added constraints, the resulting optimization problems are similar to the optimization problem solved by MORE in each iteration. Those similarities allow us to derive efficient solutions and closed form distribution updates for the M-step. Yet, the MORE constraints are not only added for convenience but have other beneficial properties. We further justify these constrains in section 5.1.

We first present an overview in algorithm 2 before we state detailed explanations of the updates in the following. Those update derivations closely resemble the updates proposed by Arenz et al. [2018] in VIPS.

Algorithm 2: Expected Information Maximization for Gaussian Mixture Models.

```
EIM-for-GMMs($\{\mathbf{x}_\mathrm{p}^{(j)}\}_{j=1\cdots N}, q(\mathbf{x})$);
Input: Data $\{\mathbf{x}_\mathrm{p}^{(j)}\}_{j=1\cdots N}$, Initial Model $q(\mathbf{x}) = \sum_{i=1}^d q(\mathbf{x}|z_i)q(z_i) = \sum_{i=1}^d \pi_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
for i in number of iterations do
    E-Step:
    $q_\mathrm{old}(z) = q(z)$, $q_\mathrm{old}(\mathbf{x}|z_i) = q(\mathbf{x}|z_i)$ for all components $i$
    sample data from model $\{\mathbf{x}_\mathrm{q}^{(j)}\}_{j=1\cdots N} \sim q_\mathrm{old}(\mathbf{x})$
    retrain density ratio estimator $\phi(\mathbf{x})$ on $\{\mathbf{x}_\mathrm{p}^{(j)}\}_{j=1\cdots N}$ and $\{\mathbf{x}_\mathrm{q}^{(j)}\}_{j=1\cdots N}$
    M-Step Weights:
    for i in number of components do
        compute reward $r_i = \frac{1}{N}\sum_{j=1}^N \phi\left(\mathbf{x}_\mathrm{q}^{(j)}\right)$ with samples $\{\mathbf{x}_\mathrm{q}^{(j)}\}_{j=1\cdots N} \sim q_\mathrm{old}(\mathbf{x}|z_i)$
    update $q(z)$ using rewards $r_i$
    M-Step Components:
    for i in number of components do
        fit $\hat{\phi}(\mathbf{x})$ surrogate to pairs $\left(\mathbf{x}_\mathrm{q}^{(j)}, \phi\left(\mathbf{x}_\mathrm{q}^{(j)}\right)\right)$ with samples $\{\mathbf{x}_\mathrm{q}^{(j)}\}_{j=1\cdots N} \sim q_\mathrm{old}(\mathbf{x}|z_i)$
        update $q(\mathbf{x}|z_i)$ using surrogate $\hat{\phi}(\mathbf{x})$
```

4.1.1 M-Step Mixture Distribution

First, consider the M-step to update the mixture distribution. Since the components are fixed, i.e., $q(\mathbf{x}|z) = q_{\text{old}}(\mathbf{x}|z)$, the expected KL in Equation 3.4 vanishes. Furthermore, the outer expectation over $q(z)$ can be expressed as a sum. Adding the MORE constraints yields

$$\max_{q(z)} \sum_{i=1}^{d} p(z_i)\mathbb{E}_{q(\mathbf{x}|z_i)}\left[\phi(\mathbf{x})\right] - \text{KL}\left(q(z) \,\|\, q_{\text{old}}(z)\right)$$

$$\text{s.t.} \quad \text{KL}\left(q(z) \,\|\, q_{\text{old}}(z)\right) \leq \epsilon, \quad \text{H}(q(z)) \geq \beta, \quad \sum_{i=1}^{d} q(z_i) = 1.$$

We use a sample based approximation for the expectation over the log density ratio estimates and denote it by $\phi(z) = \mathbb{E}_{q(\mathbf{x}|z)}\left[\phi(\mathbf{x})\right]$. This optimization problem differs from the MORE optimization problem only by the additional KL term in the objective.

To solve the optimization we first compute the Lagrangian multipliers η for the KL constraint, and ω for the entropy constraint, by optimizing the dual function. As the derivations in the appendix A.3 show, combining the KL terms in the objective and constraint results in a dual that differs from the original MORE dual only in a 1 being added to η, i.e.,

$$g(\eta, \omega) = \eta\epsilon - \omega\beta + (\eta + 1 + \omega)\log\sum_{i=1}^{d}\exp\left(\frac{(\eta+1)\log q_{\text{old}}(z) + \phi(z)}{\eta + 1 + \omega}\right). \tag{4.1}$$

Not only the dual but also the update equations differ from the original MORE update only by the 1 added to η. Thus, given η and ω, we can compute the parameters $\boldsymbol{\pi}$ of $q(z)$ in closed form. Since the reward for each z_i can be expressed by a single value $\phi(z_i)$ we do not need to fit a surrogate but can directly work with $\hat{\mathbf{r}}$, a vector whose i-th entry is $r_i = \phi(z_i)$ and the new parameters are given by

$$\boldsymbol{\pi} = \text{softmax}\left(\frac{(\eta+1)\log\left(\boldsymbol{\pi}_{\text{old}}\right) + \hat{\mathbf{r}}}{\eta + 1 + \omega}\right),$$

where the log of the vector $\boldsymbol{\pi}_{\text{old}}$ is taken element-wise.

4.1.2 M-Step Components

Consider the M-step to update a single component $q(\mathbf{x}|z_i) = \mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)$. Due to the previous update of the mixture distribution $q(z) \neq q_{\text{old}}(z)$ and thus the $\text{KL}\left(q(z) \,\|\, q_{\text{old}}(z)\right)$ does not vanish. Yet, we can still neglect it since it is constant with respect to the optimization variable $q(\mathbf{x}|z_i)$. Together, with adding the MORE constraints we obtain

$$\max_{q(\mathbf{x}|z_i)} \int q(\mathbf{x}|z_i)\phi(\mathbf{x})d\mathbf{x} - \text{KL}\left(q(\mathbf{x}|z_i) \,\|\, q_{\text{old}}(\mathbf{x}|z_i)\right)$$

$$\text{s.t.} \quad \text{KL}\left(q(|\mathbf{x}|z_i) \,\|\, q_{\text{old}}(\mathbf{x}|z_i)\right) \leq \epsilon, \quad \text{H}(q(\mathbf{x}|z_i)) \geq \beta, \quad \int q(\mathbf{x})d\mathbf{x} = 1.$$

We again solve the optimization problem using a MORE-like update. In order for that, we need to fit a compatible surrogate to the log density ratios $\phi(x)$. The features compatible to a Gaussian yield a quadratic function of the form

$$\hat{\phi}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\hat{\mathbf{R}}\mathbf{x} + \hat{\mathbf{r}}^T\mathbf{x} + r_0,$$

where $\hat{\mathbf{R}}$ is a symmetric matrix. This function is fitted to samples from $q(\mathbf{x}|z_i)$ by minimizing

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{q(\mathbf{x}|z_i)}\left[\left(\phi(\mathbf{x}) - \hat{\phi}(\mathbf{x})\right)^2\right].$$

The dual problem is of the same analytical form as Equation 4.1, except that the sum over z becomes an integral over $\mathbf{x}$,

$$g(\eta,\omega) = \eta\epsilon - \omega\beta + (\eta + 1 + \omega)\log\int\exp\left(\frac{(\eta+1)\log q_{\text{old}}(\mathbf{x}|z_i) + \phi(\mathbf{x})}{\eta+1+\omega}\right)d\mathbf{x}. \qquad (4.2)$$

Yet, given the compatible surrogate, this integral can be solved in closed form for Gaussian distributions as demonstrated in the appendix A.3.2. This dual allows us to efficiently compute the optimal Lagrangian multipliers. Similar to the dual, we can reuse the MORE closed form update equations to obtain the new distribution. The natural parameters of the new distribution, i.e $\mathbf{q} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and $\mathbf{Q} = \boldsymbol{\Sigma}^{-1}$ are given by

$$\mathbf{q} = \frac{(\eta+1)\mathbf{q}_{\text{old}} + \hat{\mathbf{r}}}{\eta+1+\omega} \quad \text{and} \quad \mathbf{Q} = \frac{(\eta+1)\mathbf{Q}_{\text{old}} + \hat{\mathbf{R}}}{\eta+1+\omega}.$$

Note that since $\hat{\mathbf{R}}$ and $\mathbf{Q}_{\text{old}}$ are symmetric, $\mathbf{Q}$ is symmetric as well. Yet, it might not be positive definite and we need to check for this case, which is rare in practice. One possible solution to handle it is increasing η until $\mathbf{Q}$ becomes positive definite.

4.2 Mixtures of Experts

For conditional distributions, we consider mixtures of experts with d Gaussian components and a softmax gating. For both, the gating and the components, we derive the M-step for distributions that depend linearly and nonlinearly on the conditioning variable. In the case of the gating, the linear and nonlinear cases are very similar, as closed form updates are infeasible for both. Yet, deriving closed form updates for the components is possible in the linear case. Hence, we will consider the linear and nonlinear cases separately as closed form updates are no longer possible for the latter.

We again decompose the M-step into independent updates for the gating and the individual components. Following the same argumentation as for GMMs, we add the MORE constraints to the individual updates and derive efficient methods for them by exploiting similarities to information theoretic policy search.

4.2.1 M-Step Gating

First, consider the linear case $q(z|\mathbf{y}) = \text{softmax}(\mathbf{V}\mathbf{y} + \mathbf{v})$. Since we update the gating first, $q(z_i|\mathbf{y}) = q_{\text{old}}(z_i|\mathbf{y})$ for all i, thus the corresponding expected KL in Equation 3.7 vanishes. Furthermore, the integral over z becomes a sum as the gating distribution is discrete. Thus, adding the MORE constraints yields

$$\max_{q(z|\mathbf{y})} \int p(\mathbf{y}) \sum_{i=1}^{d} q(z_i|\mathbf{y}) \mathbb{E}_{q(\mathbf{x}|z_i,\mathbf{y})} [\phi(\mathbf{x},\mathbf{y})] \, d\mathbf{y} - \mathbb{E}_{p(\mathbf{y})} [\mathrm{KL}(q(z|\mathbf{y}) \,||\, q_{\mathrm{old}}(z|\mathbf{y}))]$$

$$\text{s.t.} \quad \mathbb{E}_{p(\mathbf{y})} [\mathrm{KL}(q(z|\mathbf{y}) \,||\, q_{\mathrm{old}}(z|\mathbf{y}))] \leq \epsilon, \quad \mathbb{E}_{p(\mathbf{y})} [\mathrm{H}(q(z|\mathbf{y}))] \geq \beta, \quad \forall \mathbf{y} : \sum_{i=1}^{d} q(z_i|\mathbf{y}) = 1.$$

In practice it is sufficient to approximate the expectation over the log density ratio with a single sample from $q(\mathbf{x}|z_i,\mathbf{y})$ and we denote

$$\phi(\mathbf{y},z) = \mathbb{E}_{q(\mathbf{x}|z_i,\mathbf{y})} [\phi(\mathbf{x},\mathbf{y})].$$

Combining the KL terms from the objective and the constraints yields the dual

$$g(\eta,\omega) = \eta\epsilon - \omega\beta + (\eta + 1 + \omega) \int p(\mathbf{y}) \log \sum_{i=1}^{d} \exp\left(\frac{(\eta+1)\log q_{\mathrm{old}}(z_i|\mathbf{y}) + \phi(\mathbf{y},z_i)}{\eta + 1 + \omega} \right) d\mathbf{y}. \tag{4.3}$$

Deriving closed form updates would be possible for the linear case. However, the features compatible with the softmax are linear and we need to fit a global surrogate for each component. It is not feasible to linearly approximate the potentially highly nonlinear log density ratio over the whole context space. Thus we use a sample based approach, similar to [Peters et al., 2010; Daniel et al., 2016]. The unnormalized probability values for a given $\mathbf{y}$ can be obtained by

$$q(z|\mathbf{y}) \propto \exp\left(\frac{(\eta+1)\log q_{\mathrm{old}}(z|\mathbf{y}) + \phi(\mathbf{y},z)}{\eta + 1 + \omega} \right)$$

and subsequently normalized to obtain the probabilities $q(z|\mathbf{y})$. Ultimately, the model of the gating distribution is re-fitted to the new values $q(z|\mathbf{y})$. Even for linear models, this regression is not realizable in closed form, and numerical optimization has to be employed.

In the nonlinear case $q(z|\mathbf{y}) = \mathrm{softmax}(\psi_s(\mathbf{y}))$ the same sample based procedure can be applied. Only the final step of fitting the model to the new targets needs to be adapted to the nonlinear model structure, which, for most classes of models will again result in a numerical optimization.

4.2.2 M-Step Linear Components

Lets consider the update of the i-th component in the linear case $q(\mathbf{x}|z_i,\mathbf{y}) = \mathcal{N}(\mathbf{W}_i\mathbf{y} + \mathbf{w}_i, \mathbf{\Sigma}_i)$. The expected KL, $\mathbb{E}_{p(\mathbf{y})} [\mathrm{KL}(q(z_i|\mathbf{y}) \,||\, q_{\mathrm{old}}(z_i|\mathbf{y}))]$, is constant with respect to the optimization variable $q(\mathbf{x}|z_i,\mathbf{y})$, thus the objective can be simplified to

$$\max_{q(\mathbf{x}|z_i,\mathbf{y})} \int p(\mathbf{y}) q(z_i|\mathbf{y}) \int q(\mathbf{x}|z_i,\mathbf{y}) \phi(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y} - \int p(\mathbf{y}) q(z_i|\mathbf{y}) \mathrm{KL}(q(\mathbf{x}|z_i,\mathbf{y}) \,||\, q_{\mathrm{old}}(\mathbf{x}|z_i,\mathbf{y})) \, d\mathbf{y}.$$

Opposed to the marginal case, the probability of the component $q(z_i|\mathbf{y})$ does not vanish in the conditional case. Intuitively, if a component is not responsible for a certain part of the context space, errors in that part should not be penalized. The same intuition applies to the MORE constraints. If the component is not responsible for a certain part of the context space, changes in that part should not be penalized.

In order to account for the responsibilities we take the expectation over $\tilde{p}(\mathbf{y}|z_i) = p(\mathbf{y})q(z_i|\mathbf{y})/q(z_i)$, not $p(\mathbf{y})$. To this end, we multiply the objective by the constant $1/q(z_i)$. Ultimately, we obtain

$$\max_{q(\mathbf{x}|z_i,\mathbf{y})} \int \tilde{p}(\mathbf{y}|z_i) \int q(\mathbf{x}|z_i,\mathbf{y})\phi(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} - \mathbb{E}_{\tilde{p}(\mathbf{y}|z_i)}\left[\mathrm{KL}\left(q(\mathbf{x}|z_i,\mathbf{y}) \,\|\, q_{\mathrm{old}}(\mathbf{x}|z_i,\mathbf{y})\right)\right]$$

$$\text{s.t.} \quad \mathbb{E}_{\tilde{p}(\mathbf{y}|z_i)}\left[\mathrm{KL}\left(q(\mathbf{x}|z_i,\mathbf{y}) \,\|\, q_{\mathrm{old}}(\mathbf{x}|z_i,\mathbf{y})\right)\right] \leq \epsilon, \quad \mathbb{E}_{\tilde{p}(\mathbf{y}|z_i)}\left[\mathrm{H}(q(\mathbf{x}|z_i,\mathbf{y}))\right] \geq \beta, \quad \forall \mathbf{y}: \int q(\mathbf{x}|z_i,\mathbf{y})d\mathbf{x} = 1.$$

The dual is similar to the dual for the gating update, i.e., Equation 4.3 and is given by

$$g(\eta,\omega) = \eta\epsilon - \omega\beta + (\eta+1+\omega)\int \tilde{p}(\mathbf{y}|z_i)\log\left(\int \exp\left(\frac{(\eta+1)\log q_{\mathrm{old}}(\mathbf{x}|z_i,\mathbf{y}) + \phi(\mathbf{x},\mathbf{y})}{\eta+1+\omega}\right)d\mathbf{x}\right)d\mathbf{y}. \tag{4.4}$$

To get closed form updates for the Gaussian components we need to fit a quadratic function locally. This is much more feasible than fitting a linear function globally, which would be required for the gating update. Thus, we can again work with MORE-like updates. The quadratic surrogate is of the form

$$\phi(\mathbf{x},\mathbf{y}) \approx \hat{\phi}(\mathbf{x},\mathbf{y}) = -\frac{1}{2}\mathbf{x}^T\hat{\mathbf{R}}_{xx}\mathbf{x} + \mathbf{x}^T\hat{\mathbf{R}}_{xy}\mathbf{y} - \frac{1}{2}\mathbf{y}^T\hat{\mathbf{R}}_{yy}\mathbf{y} + \hat{\mathbf{r}}_x^T\mathbf{x} + \hat{\mathbf{r}}_y^T\mathbf{y} + \hat{r}_0.$$

When fitting the surrogate, we once more need to account for the relevance of each context y, thus we minimize

$$\mathbb{E}_{q(\mathbf{x}|z_i,\mathbf{y})\tilde{p}(\mathbf{y}|z_i)}\left[\left(\phi(\mathbf{x},\mathbf{y}) - \hat{\phi}(\mathbf{x},\mathbf{y})\right)^2\right].$$

With such surrogates the update equations can be given in closed form

$$\mathbf{Q} = \left(\frac{(\eta+1)\mathbf{Q}_{\mathrm{old}} + \hat{\mathbf{R}}_{xx}}{\eta+1+\omega}\right), \quad \mathbf{L} = \left(\frac{(\eta+1)\mathbf{L}_{\mathrm{old}} + \hat{\mathbf{R}}_{xy}}{\eta+1+\omega}\right), \quad \mathbf{l} = \left(\frac{(\eta+1)\mathbf{l}_{\mathrm{old}} + \hat{\mathbf{r}}_{x}}{\eta+1+\omega}\right),$$

where $\mathbf{Q}$ denotes the precision matrix $\boldsymbol{\Sigma}^{-1}$ and $\mathbf{L}$ and $\mathbf{l}$ denote $\boldsymbol{\Sigma}^{-1}\mathbf{W}$ and $\boldsymbol{\Sigma}^{-1}\mathbf{w}$ respectively. The derivations for those updates are based on [Akrour et al., 2018] and can be found together with the dual in appendix A.4.2.

4.2.3 M-Step Nonlinear Components

In order to realize nonlinear components $q(\mathbf{x}|z_i,\mathbf{y}) = \mathcal{N}\left(\psi_{\mu,i}(\mathbf{y}), \psi_{\Sigma,i}(\mathbf{y})\right)$ we note that the dual in Equation 4.4 can be reformulated to

$$\begin{aligned} &g(\eta,\omega) \\ =&\eta\epsilon - \omega\beta + (\eta+1+\omega)\int \tilde{p}(\mathbf{y}|z_i)\log\left(\int q_{\mathrm{old}}(\mathbf{x}|z_i)\exp\left(\frac{-\omega\log q_{\mathrm{old}}(\mathbf{x}|z_i,\mathbf{y}) + \phi(\mathbf{x},\mathbf{y})}{\eta+1+\omega}\right)d\mathbf{x}\right)d\mathbf{y}, \end{aligned}$$

which allows sample based approximations of both integrals. Analogously, the update equation can be rewritten to

$$q(\mathbf{x}|z_i,\mathbf{y}) \propto q_{\mathrm{old}}(\mathbf{x}|z_i,\mathbf{y})\exp\left(\frac{-\omega\log q_{\mathrm{old}}(\mathbf{x}|z_i,\mathbf{y}) + \phi(\mathbf{x},\mathbf{y})}{\eta+1+\omega}\right).$$

Thus, it becomes clear how the model can be updated by a weighted fit to the samples of the old distribution. In order to account for the fixed context distribution $p(\mathbf{y})$ during the update, we need to normalize the weights such that the weights for all samples from one context sum to 1.

4.3 Practical Aspects

Here we want to elaborate on several practical aspects of the introduced method. Namely, the density ratio estimation which plays an important role in our approach, as well as, the weighted expectation during the conditional M-step and how to efficiently implement our approach.

4.3.1 Density Ratio Estimation

The two main design choices for the density ratio estimator are its parametric family and the objective used to train it. In all our experiments we realize the density ratio estimator as a neural network based logistic regressor.

Besides being easy to scale, neural networks have the appealing property that they are efficiently adaptable to the new model distribution after each M-step. Since the updates during the M-step are bounded the change in the density ratio estimate is small and the network can be adapted using a small number of epochs. Other reasonable approaches, such as Gaussian processes [Rasmussen, 2003] cannot be retrained this easily.

While the density ratio estimation under Bregman divergences [Sugiyama et al., 2012b] provides a variety of possible objectives, preliminary experiments showed that the binary cross entropy works best. In particular, Least Square Importance Fitting (LSIF) [Yamada et al., 2013], which is recommended by many studies [Uehara et al., 2016; Dawid et al., 2016], caused numerical instabilities and bad performance. The network output is the log density ratio $\phi(x)$ and computing the loss for LSIF includes computing the squared density ratio, i.e., $r(x)^2 = \exp(\phi(x))^2$, which can yield exploding gradients. Passing $\phi(x)$ through a sigmoid, whose output is always between 0 and 1 and whose gradient saturates if the absolute value of $\phi(x)$ is large, is numerically much more stable. Similar concerns are expressed by Poole et al. [2016]. Uehara et al. [2016], who recommend LSIF, need to clip the values heuristically in order for the training of their b-GAN to succeed.

Also note that the original density ratio estimate to be computed, $r(x) = q(x)/p(x)$ can be extended with an arbitrary conditional distribution $p_\theta(\theta|x)$

$$r(x) = \frac{q(x)}{p(x)} = \frac{q(x)p_\theta(\theta|x)}{p(x)p_\theta(\theta|x)} = \frac{q(x,\theta)}{p(x,\theta)}.$$

This insight can be used to provide additional information to the density ratio estimator that may allow for a better estimate and more efficient training.

4.3.2 Computing Expectations over $\tilde{p}(\mathbf{y}|z)$

For mixtures of experts we need to take expectations over $\tilde{p}\,(\mathbf{y}|z)$ during the component update. Yet, we have no samples of $\tilde{p}(\mathbf{y}|z)$ but only samples of $p(\mathbf{y})$. In order to compute the expectations importance sampling needs to be employed, the weights are given by

$$\frac{\tilde{p}(\mathbf{y}|z_i)}{p(\mathbf{y})} = \frac{q(z_i|\mathbf{y})}{q(z_i)},$$

where $q(z_i)$ is obtained by $q(z_i) = \mathbb{E}_{p(\mathbf{y})}\left[q(z_i|\mathbf{y})\right]$.

4.3.3 Efficient Implementation

Exploiting parallel computations allows an efficient implementation of the approach. In particular, the components updates are independent of one another and can be performed simultaneously. Additionally, when working with neural networks specialized frameworks such as Tensorflow [Abadi et al., 2016] allow for efficient computation using GPUs and automated gradient computation.

5 Qualitative Comparison

In the following, we elaborate on the similarities, differences, and connections of our approach to the previously introduced related work.

5.1 Reinforcement Learning

We exploit information theoretic policy search to realize efficient updates during the M-step for both Gaussian mixture models and mixtures of experts. Those methods have been designed to tackle problems typical to reinforcement learning and we elaborate on the extent to which those are relevant for our approach.

Sample efficiency is a relevant aspect of most machine learning methods. Yet, it is of particular importance in reinforcement learning and especially in stochastic search, as the evaluation of the reward for a single sample of the search distribution is computationally expensive. In our approach, obtaining the reward for a sample corresponds to evaluating the density ratio estimator which can be very fast. Realizing the density ratio estimator as a neural network allows evaluating it for thousands of samples efficiently and in parallel. Hence, there are effectively no problems regarding sample efficiency during the M-step. Thus, we can work directly with the sample based version of the dual in section 4.2.3, which is usually infeasible in the normal stochastic search setting. Note, however, that the amount of samples available from the true distribution is still an important issue for training the density ratio estimator during the E-step.

Another major issue reinforcement learning methods have to deal with is the exploration-exploitation trade-off. Our approach, on the other hand, assumes access to all data that will ever be available from the first iteration. Thus no exploration is needed or, in fact, possible. Yet, the mechanisms limiting the exploitation are crucial for EIM since the updates of the distribution depend on the density ratio estimator, whose estimates are usually far from perfect. The estimates are worse in regions where the model has little density since there are less training samples. Thus, staying close to the old distribution is important. Additionally, bounding the updates allows retraining the density ratio estimator after the update with little effort. When working with closed form updates and reward surrogates, the KL constraint is also necessary to define a trust region in which the surrogate is assumed to be reliable.

Constraining the loss in entropy is necessary for the same reason it is necessary in the normal stochastic search case, i.e., to prevent premature convergence to spurious local optima.

5.2 Expectation Maximization

Both EM and EIM do not optimize their objective directly but work by optimizing a bound to the objective instead.

It is interesting to compare our upper bound to the lower bound objective optimized by EM. As mentioned, maximizing the likelihood is equivalent to minimizing the M-projection, i.e., $\mathrm{KL}(p(x) \,\|\, q(x))$, where, in relation to our objective, the model and true distribution have switched places in the non-symmetric KL objective. Like our approach, EM introduces an auxiliary distribution $\tilde{q}(z|x)$ and bounds the objective from below by subtracting the KL between auxiliary distribution and model, i.e., $\mathrm{KL}(\tilde{q}(z|x) \,\|\, q(z|x))$. In contrast, we obtain our upper bound by adding $\mathrm{KL}(q(z|x) \,\|\, \tilde{q}(z|x))$ to the objective. Again, the distributions have exchanged places within the KL.

5.3 Generative Adversarial Networks

To the best of our knowledge, generative adversarial approaches, such as the f-GAN [Nowozin et al., 2016] are currently the only other approaches capable of computing the I-projection between a target and a model distribution solely based on samples. Besides the shared objective of modeling distributions over data, further similarities are rooted in the connections between density ratio estimation and discriminators discussed in section 2.4.2. Yet, there are also important differences.

A key benefit of GANs is that they do not require the density of the model to be tractable, which enables the use of powerful probabilistic models and allows modeling complex, high dimensional distributions, e.g., over images. However, using probabilistic models with intractable densities is clearly only reasonable if the density is not of interest. In the context of interest to us, i.e., autonomous systems and robotics, this case is rare. Here probabilistic models are often employed to exploit the additional information they provide, e.g., in the form of uncertainty. Because they do not rely on the density or structure of the model, GANs are agnostic to it during training. EIM, on the other hand, requires the density to be tractable, which allows us to exploit the model structure during training, resulting in specialized update rules that are arguably more effective.

Another major difference is that our approach is not adversarial in the sense that the objective is not a min-max game. Instead, the density ratio estimator can be viewed to support learning the model by providing good estimates about the density ratio estimate, yielding more effective updates and removing a major source of instability in GAN approaches. Additionally, in our approach, the density ratio between the distribution prior to the update, $q_{\mathrm{old}}(x)$ and the target distribution $p(x)$ is estimated. Thus, the log density ratio $\phi(x)$ does not depend on the optimization variable during the M-step, i.e., $q(x)$, which is accounted for by the additional KL terms in the objective. For GANs the discriminator depends on the current model $q(x)$ and needs to be fixed artificially during the generator update in order to obtain a tractable optimization problem.

5.4 Variational Inference

There exist approaches utilizing variational inference to fit probabilistic models to data, e.g., Variational Bayes EM [Bishop, 2006] and Variational Autoencoders [Kingma and Welling, 2013]. Based on a lower bound objective, those methods compute the I-projection between a variational distribution and the true, intractable, posterior over parameters θ given data x. Yet, $p(\theta|x)$ is proportional to the likelihood of the data given the parameters $p(x|\theta)$ and a prior over the parameters $p(\theta)$, i.e., $p(\theta|x) \propto p(x|\theta)p(\theta)$. Thus, parameters obtained from the variational distribution, e.g., by sampling or maximum a-posteriori, will yield a distribution over the data that is close to the maximum likelihood solution.

Our approach, on the other hand, directly computes the I-projection between the model and the true distribution over the data.

5.5 VIPS

Both VIPS [Arenz et al., 2018] and EIM work with the same upper bound to their original objective, i.e., finding the I-projection. To further emphasize the similarities consider our upper bound, stated in Equation 3.3. Since we assume only samples of $p(x)$ are available we can not directly work with this bound and need a density ratio estimator to make it computable. Under the assumption of VIPS, i.e., access to the unnormalized density $p^*(x) = cp(x)$ we can make the bound computable by noting

$$
\begin{aligned}
&\int q(z) \int q(x|z) \log \frac{q_{\text{old}}(x)}{p(x)} dx dz + \mathbb{E}_{q(z)} \left[\text{KL}\left(q(x|z) \,\|\, q_{\text{old}}(x|z)\right)\right] + \text{KL}\left(q(z) \,\|\, q_{\text{old}}(z)\right) \\
= &\int q(z) \int q(x|z) \log \frac{q_{\text{old}}(x)}{p^*(x)} dx dz - \log c + \mathbb{E}_{q(z)} \left[\text{KL}\left(q(x|z) \,\|\, q_{\text{old}}(x|z)\right)\right] + \text{KL}\left(q(z) \,\|\, q_{\text{old}}(z)\right).
\end{aligned}
$$

where the constant $\log c$ can be neglected during optimization. Plugging the E-step into the VIPS upper bound stated in Equation 2.9 yields the same result. Thus, it becomes clear that VIPS and EIM optimize the same upper bound objective under different assumptions, i.e., access to the unnormalized density for VIPS and only access to samples for EIM.

This equivalence has several interesting implications. First, it shows how the M-step in VIPS can be realized by adding a 1 to the Lagrangian multiplier corresponding to the KL, η, instead of setting the Lagrangian multiplier corresponding to the entropy constraint, ω, to 1. Second, if the M-step is derived for a particular latent variable model, it can be used for both VIPS and EIM. Third, it allows us to use our derivations of the conditional upper bound and mixtures of experts also for VIPS.

Yet, from a practical perspective, there are several differences between VIPS and EIM. As stated in section 5.1 sample efficiency is not an issue for EIM during the M-Step. For VIPS, on the other hand, it is an issue, since evaluating $p^*(x)$ is usually costly. Arenz et al. [2018] addressed this issue by reusing samples.

6 Quantitative Evaluation

To the best of our knowledge, Generative Adversarial Networks (GANs) are the only other existing method to find the I-projection solely based on samples. Naturally, we compare to them and analyze the influence of the differences between EIM and GANs. In a second experiment, we show EIM is more robust with regards to spurious local optima than EM. We also show several experiments on real world data, demonstrating the I-projections benefits and EIMs capability of finding good solutions. Finally, we provide a proof of concept for learning nonlinear components with context dependent covariances.

In all experiments, we realized the density ratio estimator as fully connected neural networks which we trained using ADAM [Kingma and Ba, 2014]. In most experiments we used dropout [Srivastava et al., 2014] and early stopping during the training of the density ratio estimator. A full overview of all used hyperparameters can be found in Appendix B.

6.1 Comparison to Generative Adversarial Approaches

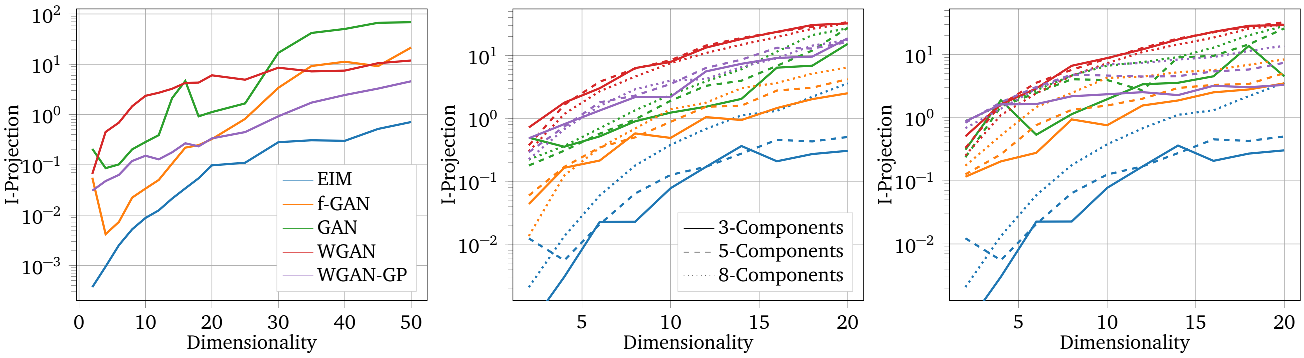
We compare EIM to several generative adversarial approaches, the original GAN [Goodfellow et al., 2014], the I-projection version of the f-GAN [Nowozin et al., 2016], the Wasserstein GAN (WGAN) [Arjovsky et al., 2017], and the Wasserstein GAN with gradient penalty (WGAN-GP) [Gulrajani et al., 2017]. We measure performance using the I-projection and evaluate the approaches on fitting models to randomly generated Gaussians and GMMs of varying dimensions and numbers of components. In all experiments, the models are rich enough to perfectly fit the true data distribution and thus all approaches should find the same minimum, despite optimizing different objectives.

In order to allow fitting GMMs with GANs we employed both methods introduced in section 2.4.3, i.e., reparamertization with the Gumbel softmax [Jang et al., 2017] and training the weights with an additional mutual information term. Results can be found in Figure 6.1.

In section 5.3 we identified three main differences between GANs and our approach. GANs have an adversarial objective, are agnostic to the model structure, and the discriminator is working with the current model while EIM optimizes a non-adversarial objective, exploits the model structure, and the density ratio estimator is working with the model from the previous iteration. To investigate the influences of those differences, we introduce two modified versions of our approach. First, unstructured EIM (EIM-US) does not exploit the model structure but realizes the M-step by propagating gradients back through the sampling process, similar to the GAN approaches. Second, EIM-KL does not have the KL term in the objective, which effectively removes the auxiliary distribution and the density ratio estimator is now working directly with the current model. We also investigate a combination of the two, EIM-US-KL. The results of the comparison can be found in Figure 6.2.

6.2 Comparing Robustness to EM

EM is known to be prone to spurious local optima. In order to show that EIM suffers less from this problem we performed a simple experiment, fitting a simple one dimensional conditional distribution with



(a) Gaussian

(b) GMM (Gumbel Softmax)

(c) GMM (Mutual Information)

Figure 6.1.: Average I-Projection achieved for EIM, GAN, f-GAN, WGAN and WGAN-GP. EIM outperforms all adversarial approaches, especially for high dimensions and numbers of components. Note that we displayed the same results for our approach in **(b)** and **(c)** in order to compare them to the generative adversarial approaches with different training methods for the weight distribution.

$p(x) = \text{Uniform}(-1, 1)$ and $p(y|x) = \sin(2.5\pi x) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mu = 0, \sigma = 0.1)$, starting from random initialization. Figure 6.3 shows that, while EM achieves higher log-likelihood if the global optimum is reached it often fails to do so. EIM, on the other hand, converges to the global optimum more often and thus ultimately achieves a higher average log-likelihood than EM, despite optimizing a different objective.

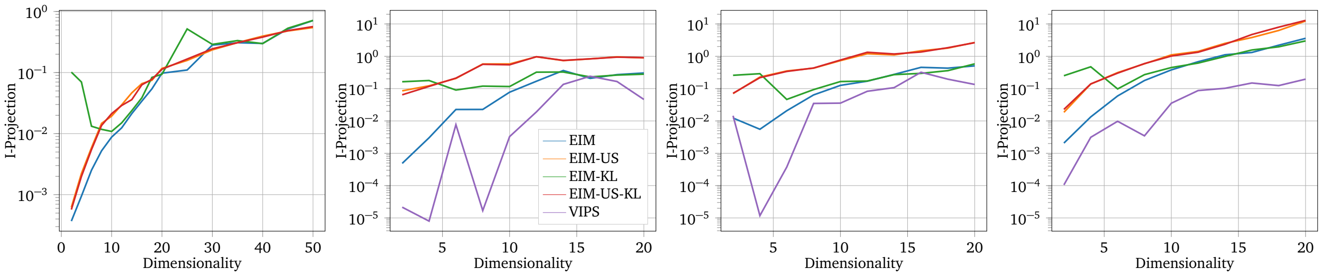
6.3 Line Reaching with Planar Robot

To further illustrate the benefits of the I-Projection we extended the introductory example of the planar reaching task and collected expert data from the robot tasked with reaching a point on a line.

We fitted GMMs with an increasing number of components using EIM, EIM with the x-coordinate of the robot end-effector provided as additional information to the density ratio estimator, and EM. Even for a large number of components, we see effects similar to the introductory example, i.e., the M-Projection solution, provided by EM, fails to reach the line while EIM manages to do so. However, for small numbers of components, parts of the line are ignored by EIM, while more and more parts of it get covered as we increase the number of components. For EIM with additional information, these effects are amplified. See Figure 6.4, for average distance between end-effector and the line as well as samples from both EM and EIM. Figure 6.5 shows histograms visualizing how well the samples cover the line. This example also illustrates that EIM is able to infer the underlying task, i.e., reaching the line, while EM does not manage this due to the averaging effects of the M-projection.

6.4 Inverse Kinematics of Real World Robot

We evaluated EIM for linear mixtures of experts on the task of learning the inverse kinematics (IK) of 7-link KUKA Iiwa robot. To this end, we collected training data by evaluating the forward kinematics for randomly chosen joint configurations. The input of the IK model is the end-effector position (3D cart. coordinates) while the output is given by the joint configuration. In order to get a good inverse kinematic model, each of the components needs to focus on a small part of the context space. Thus, the



(a) Gaussian **(b)** GMMs, 3 Components **(c)** GMMs, 5 Components **(d)** GMMs, 8 Components

Figure 6.2.: Average I-Projection achieved for EIM, the modified EIM versions, and VIPS. **(a)** Fitting Gaussian distributions. Especially for lower dimensions structured EIM with the additional KL term works better than the other versions. Yet, for a single Gaussian there is no decomposition of the model and in general not much structure to exploit by the structured EIM version. **(b)-(d)** Fitting GMMs. When fitting GMMs exploiting the model structure becomes more important and thus the normal EIM version outperforms the unstructured one. Interestingly, omitting the KL term does not seem to matter for the unstructured case. We also included results from VIPS. Clearly, this comparison is not a fair as VIPS has access to the log density of the target distribution and does not rely on density ratio estimation. Yet, the differences in performance indicate that there is still room for improvement regarding the density ratio estimation. In the Gaussian case, there are no local optima and VIPS always perfectly matches the target distribution, achieving average values of smaller than 10^{-20}, which we did not display for better visibility of the other results.

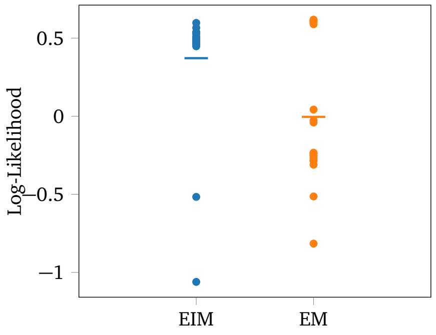


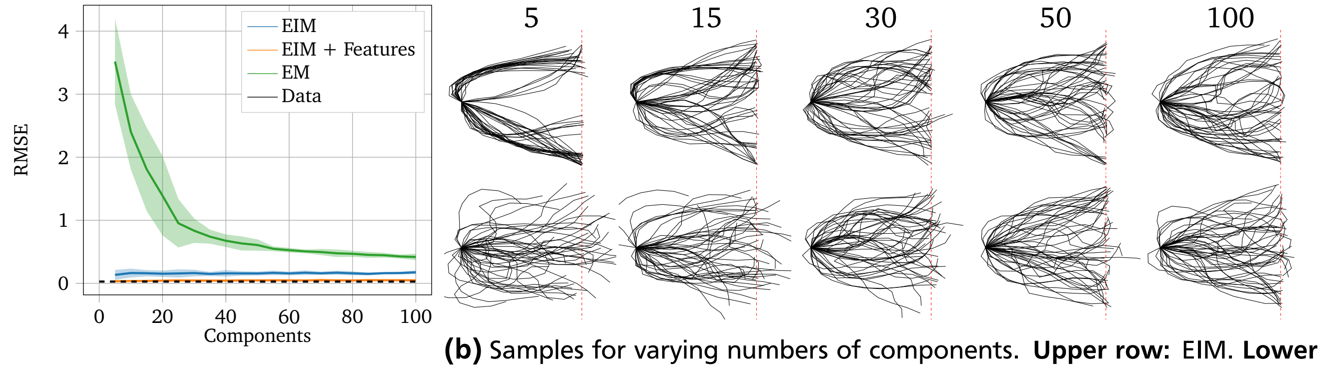
Figure 6.3.: Log-likelihood results for 20 runs of EM and EIM each. Bars indicate the mean. While EM achieves better results when it converges to the global optima it often fails to do so which is not the case for EIM. Thus ultimately our approach achieves the higher average likelihood.

models should evenly distribute the components over the context space and allow each component to focus on a small part of it. More formally, the models should have a high entropy in the marginal weight distribution, i.e $H(q(z)) = H\left(\mathbb{E}_{p(y)}\left[q(z|y)\right]\right)$ and a low expected entropy $q(z|y)$, i.e., $\mathbb{E}_{p(y)}\left[H(q(z|y))\right]$. Figure 6.6 shows average RMSE in end-effector position and the aforementioned entropy values for both, EIM and EM. It shows that EIM can significantly outperform the EM model that does not perform well due to averaging over multiple modes. Still, further improving the performance of EIM is needed in order to yield useful inverse kinematics models. Yet, EIM is the only method we are aware of that can deal with such multi-modal data.

6.5 Vehicle Trajectory Prediction

Additionally, we evaluated our approach on traffic data from the Next Generation Simulation program[1], in particular, the traffic data recorded on Lankershim Boulevard. Based on a single x-y coordinate of a vehicle EM and EIM had to predict the x-y coordinates at the next 5 time steps, with 2.5 seconds

[1] https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj



(a) Average distance to line

(b) Samples for varying numbers of components. **Upper row:** EIM. **Lower row:** EM

Figure 6.4.: Average distance to line and samples for robot line reaching. While EIM for small numbers of components ignores modes, not considering the whole line, it learns models that actually achieve the underlying task, i.e., reach the line. Providing additional information to the density ratio estimator further decreases the average distance to the line. EM, on the other hand, averages over the modes, and thus fails to reach the line even for large numbers of components.

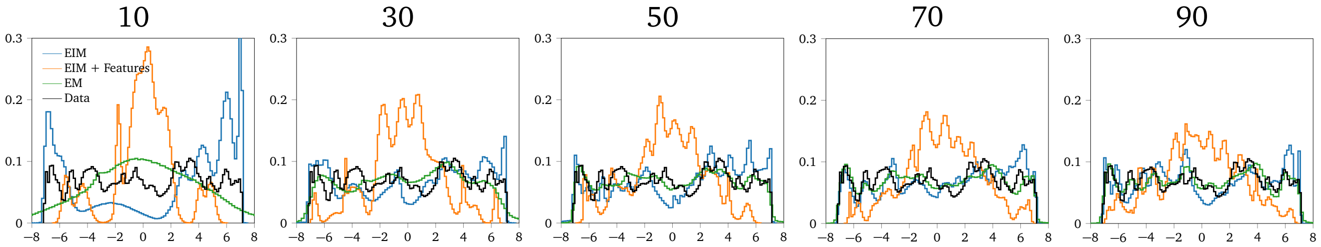
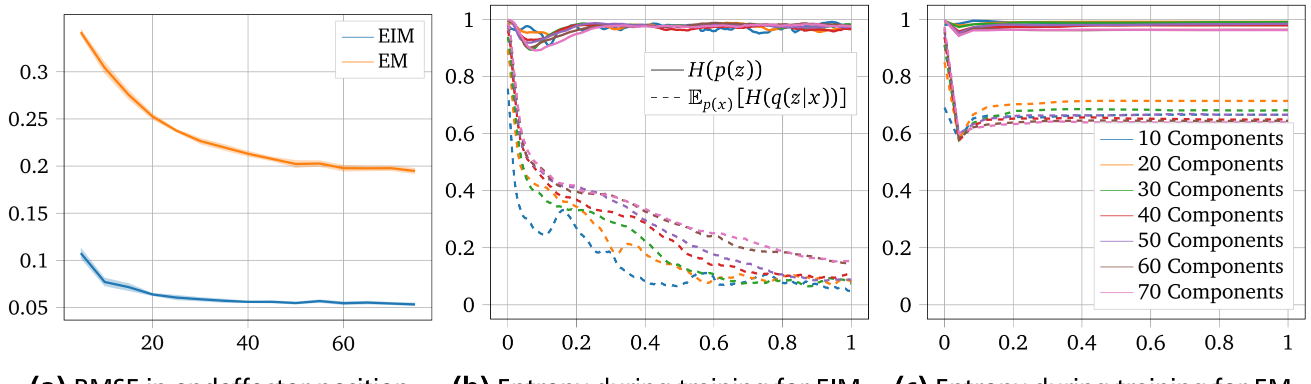


Figure 6.5.: Histograms over the y-position of the end-effector, i.e., which points on the line were reached, in samples generated with EIM, EIM with additional features, EM as well as the test data. This figure again illustrates how parts of the line are ignored by EIM for low numbers of components. Yet, the distribution of the test data is better recovered with increasing numbers of components while the solutions for EIM and EIM with additional features stay close to the line for all numbers of components, as shown in Figure 6.4.

between time steps. As the data is highly multi-modal, with vehicles driving at different speeds, stopping at red lights, changing lanes, and taking turns at intersections, accurate predictions of single trajectories are not possible based on the given features. We instead focus on the plausibility of the produced trajectories. Samples displayed together with the results in Figure 6.7, show that the data produced by EM is highly implausible with vehicles driving off the road, in between lanes and on the median strip. To formally compare the plausibility of the trajectories we used a kernel density estimator on all points of all trajectories in the test set and assessed whether a sample is plausible, i.e on the road, based on its density estimate, If no car has been seen in the position of a given sample, the position is likely to be off the road. We counted the number of off-road samples for EM and EIM with different numbers of components.

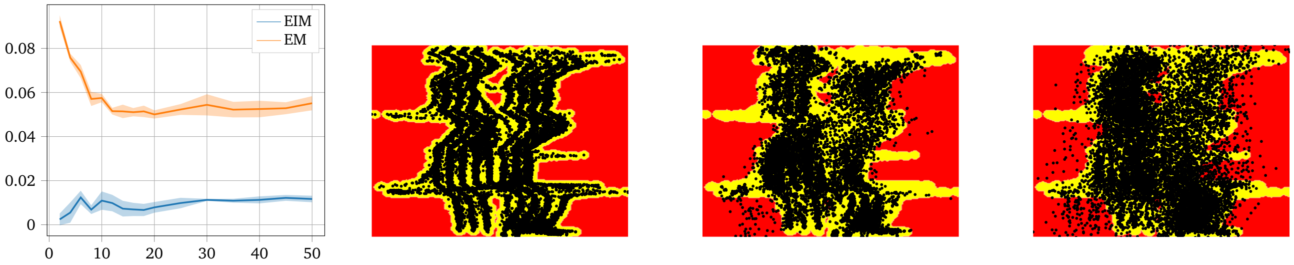
6.6 Nonlinear Components

In order to provide a proof of concept for learning nonlinear components we fitted a single univariate Gaussian $q(x|y) = \mathcal{N}\left(\mu = \psi_\mu(y), \sigma = \psi_\sigma(y)\right)$ to several simple target distributions. Figure 6.8 displayes results and shows that components can be fitted for highly nonlinear functions and learn vari-



(a) RMSE in endeffector position **(b)** Entropy during training for EIM **(c)** Entropy during training for EM

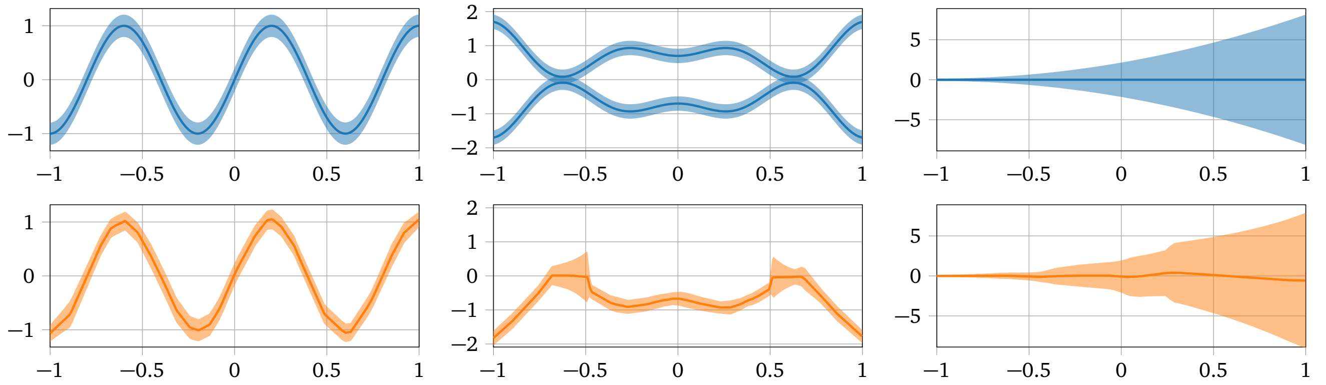
Figure 6.6.: **(a)** RMSE in end-effector position of the learned inverse kinematics model. **(b) and (c)** $\mathbb{E}_{p(x)}[H(p(z|x))]$ and $H(p(z)) = H\left(\mathbb{E}_{p(x)}[p(z|x)]\right)$ for our approach and EM. Note that we normalized the entropy and number of training iterations to get comparable plots for different numbers of components and both algorithms. While both approaches keep a high entropy in z, i.e., use all components, only our approach manages to lower the entropy of $p(z|x)$, i.e., allows the components to focus on individual parts of the state space.



(a) **(b)** Data **(c)** EIM **(d)** EM

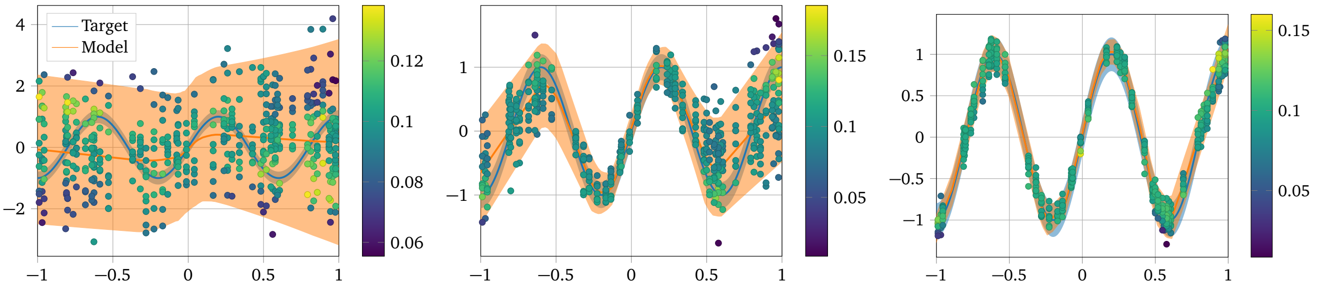
Figure 6.7.: **(a)** Fraction of off-road samples for EIM and EM for different numbers of components. It can clearly be seen that EIM produces much more plausible data. **(b)** Vehicle position in the true data. Yellow areas are defined as road, red areas as off-road. **(c)** and **(d)**. Samples from our approach and EM.

ances depending on the context. They also show that in the case of multi-modal distributions modes that can not be represented are ignored. Figure 6.9 demonstrates how the weighting of the samples successively draws the model distribution towards the target.



(a) **(b)** **(c)**

Figure 6.8.: For all experiment we sampled the contexts from $p(x) = \text{Uniform}(-1, 1)$. We evaluated for several target distributions $p(x|y)$. **(a)** $p(x|y) = \mathcal{N}(\mu = \sin(2.5\pi y), \sigma = 0.1)$, **(b)** $p(x|y) = 0.5\mathcal{N}(\mu = x\sin(2.5\pi y) + 0.7, \sigma = 0.1) \ + \ 0.5\mathcal{N}(\mu = -x\sin(2.5\pi y) - 0.7, \sigma = 0.1)$, **(c)** $p(x|y) = \mathcal{N}\left(\mu = 0, \sigma = (x+1)^2 + 0.05\right)$. The top row displays those distributions while the bottom row displays the learned models. We can see that the approach for nonlinear components successfully works with highly nonlinear functions, context dependent variance and, like expected, ignores modes it cannot represent.



(a) Iteration 10 **(b)** Iteration 50 **(c)** Iteration 200

Figure 6.9.: Current model, target distribution and samples for different iterations. The color of each sample indicates its weight for the weighted maximum likelihood fit performed to update the model during the M-step. Samples close to the target distribution get higher weights, drawing the model distribution towards the target.

7 Conclusion

We introduced Expected Information Maximization (EIM), a novel approach to compute the I-projection of a target distribution onto a family of parametric models based solely on samples of the target distribution. We derived a general, EM-like procedure for marginal as well as conditional latent variable models and presented efficient implementations of that procedure for Gaussian mixture models and mixtures of experts. We demonstrated the usefulness of the I-projection objective and our models capability of finding good solutions in our experiments. In our analysis, we showed that our approach is related to Generative Adversarial Networks (GANs), yet aims at problems where a tractable model is desired or necessary. In those cases exploiting the model structure yields significant benefits over GANs.

7.1 Future Work

The introduced work opens various pathways for future research and exploring them all lies beyond the scope of this work.

7.1.1 Improvements to the Approach

We evaluated the M-step for nonlinear components only for simple, low dimensional target distributions. For higher dimensional distributions, working with solutions based on the joint distribution, as introduced in section 2.5.1, might be necessary. In general, many ideas developed in the context of information theoretic policy search can be used in our approach. For example, constraining the KL also during the weighted maximum likelihood fit [Abdolmaleki et al., 2017, 2018].

As shown by the VIPS experiments in section 6.1, using the true density ratio yields significantly better performance than working with density ratio estimates, even for relatively simple target distributions. Clearly, the true density ratio will always work better than an estimate. Yet, this comparison also shows that there is a margin for improvement in the design of the density ratio estimator. Contrary to Uehara et al. [2016] we are free in the choice of the density ratio estimator loss function as our approach is not adversarial. Additionally, all introduced and considered density ratio estimation techniques assume both densities to be unknown and work solely based on samples. In our case the model's density would be available. Recently, Fu et al. [2018] introduced a method capable of exploiting the knowledge about one of the two distributions during density ratio estimation.

7.1.2 M-Step for Different Latent Variable Models

We exploited similarities to information theoretic policy search to realize the M-steps in all considered cases. Yet, the M-step can also be realized by directly minimizing the upper bound, however this might not yield optimal performance as shown in our experiments. Additionally, solutions based on information theoretic policy search are applicable in most scenarios.

While Gaussian mixture models and mixtures of experts are arguably among the most popular latent variable models, they are not the only ones of use. In the case of marginal mixtures, the dual derived for the component update, i.e., Equation 4.2, is valid not only for Gaussians but can be used, together with MORE or REPS like updates, to learn mixtures of arbitrary exponential family distributions.

For more complex and conditional models, modern deep reinforcement learning algorithms such as TRPO [Schulman et al., 2015] and PPO [Schulman et al., 2017] provide off the shelf solutions. Those allow for additional KL constraints and with slight modifications constraining the entropy is also possible. Yet, similar to our M-steps for nonlinear components and gating, exploiting the methods introduced in section 2.5.1 might yield additional performance and efficiency.

Ultimately, the upper bound objectives derived are not limited to discrete latent variables and ways to efficiently perform the M-step could be derived for models with continuous latent variables such as hidden Markov models.

7.1.3 EIM for High Dimensional Data

Nowadays, generative models, in particular, Generative Adversarial Networks, are often used to model distributions over extremely high dimensional data, such as images. Scaling to such high dimensional data is not possible with multivariate Gaussian distributions with full covariance as the number of entries in the covariance matrix grows quadratically with the number of dimensions. Thus in order to employ EIM for such tasks, either, simplified covariance models, such as diagonal or isotropic covariance matrices, or a completely different family of distributions needs to be used.

A recent study [Richardson and Weiss, 2018] showed how mixtures of factor analyzers [Ghahramani et al., 1996] can be used to obtain results, comparable to those of GANs, on modeling generative distributions over images using the EM algorithm. Yet, they reported the resulting images to be blurred. Arguably, this blurriness is an artifact of the averaging behavior of the maximum likelihood objective optimized by EM.

7.1.4 Reusing Results for VIPS

As discussed in section 5.5 VIPS and EIM optimize the same upper bound objective under different assumptions regarding the availability of information about the target distribution $p(x)$. Thus, the conditional upper bound and our derivations for mixtures of experts can easily be reused for VIPS.

On the other hand, Arenz et al. [2018] propose using a set of heuristics to adapt the number of components of the mixtures, automatically. Slightly modifying this set of heuristics would allow adapting the number of components for EIM.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Abdolmaleki, A., Lau, N., Reis, L. P., Peters, J., and Neumann, G. (2016). Contextual policy search for linear and nonlinear generalization of a humanoid walking controller. *Journal of Intelligent & Robotic Systems*, 83(3-4):393–408.

Abdolmaleki, A., Lioutikov, R., Peters, J. R., Lau, N., Reis, L. P., and Neumann, G. (2015). Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems*, pages 3537–3545.

Abdolmaleki, A., Price, B., Lau, N., Reis, L. P., and Neumann, G. (2017). Deriving and improving cma-es with information geometric trust regions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 657–664. ACM.

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*.

Akrour, R., Abdolmaleki, A., Abdulsamad, H., Peters, J., and Neumann, G. (2018). Model-free trajectory-based policy optimization with monotonic improvement. *The Journal of Machine Learning Research*, 19(1):565–589.

Ali, S. M. and Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 131–142.

Arenz, O., Zhong, M., and Neumann, G. (2018). Efficient gradient-free variational inference using policy search. In *Proceedings of the 35th International Conference on Machine Learning*, pages 234–243. pmlr.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 21–30, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Bishop, C. M. (1994). Mixture density networks. Technical report, Citeseer.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180.

Daniel, C., Neumann, G., Kroemer, O., and Peters, J. (2016). Hierarchical relative entropy policy search. *The Journal of Machine Learning Research*, 17(1):3190–3239.

Dawid, A. P., Musio, M., and Ventura, L. (2016). Minimum scoring rule inference. *Scandinavian Journal of Statistics*, 43(1):123–138.

Deisenroth, M. P., Neumann, G., Peters, J., et al. (2013). A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Fu, J., Luo, K., and Levine, S. (2018). Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*.

Ghahramani, Z., Hinton, G. E., et al. (1996). The em algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777.

Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (2012). *Fundamentals of convex analysis*. Springer Science & Business Media.

Huszár, F. (2015). How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations 2017*. OpenReviews. net.

Kakade, S. M. (2002). A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861.

Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279.

Opper, M. and Saad, D. (2001). *Advanced mean field methods: Theory and practice*. MIT press.

Peters, J., Mülling, K., and Altun, Y. (2010). Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta.

Poole, B., Alemi, A. A., Sohl-Dickstein, J., and Angelova, A. (2016). Improved generator objectives for gans. *arXiv preprint arXiv:1612.02780*.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer.

Richardson, E. and Weiss, Y. (2018). On gans and gmms. In *Advances in Neural Information Processing Systems*, pages 5852–5863.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. (2015). Trust region policy optimization. In *Icml*, volume 37, pages 1889–1897.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012a). *Density ratio estimation in machine learning*. Cambridge University Press.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012b). Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. (2016). Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*.

Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., and Sugiyama, M. (2013). Relative density-ratio estimation for robust distribution comparison. *Neural computation*, 25(5):1324–1370.

Yuksel, S. E., Wilson, J. N., and Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193.

A Derivations

In the following, we give full, detailed derivations for the equations stated above.

A.1 Upper Bound

Derivations of the upper bound stated in Equation 3.1. We assume latent variable model $q(x) = \int q(x|z)q(z)dz$ and use the identities $q(x|z)q(z) = q(z|x)q(x)$ and $\log q(x) = \log q(x|z)q(z) - \log q(z|x)$.

$$\begin{aligned}
&\mathrm{KL}\left(q(x) \,\|\, p(x)\right) = \int q(x) \log \frac{q(x)}{p(x)} dx = \iint q(x|z)q(z) \log \frac{q(x)}{p(x)} dz dx \\
=& \iint q(x|z)q(z) \left(\log \frac{q(x|z)q(z)}{p(x)} - \log q(z|x) \right) dz dx \\
=& \iint q(x|z)q(z) \left(\log \frac{q(x|z)q(z)}{p(x)} - \log q(z|x) + \log \tilde{q}(z|x) - \log \tilde{q}(z|x) \right) dz dx \\
=& \iint q(x|z)q(z) \left(\log \frac{q(x|z)q(z)}{p(x)} - \log \tilde{q}(z|x) \right) dz dx - \iint q(x|z)q(z) \left(\log q(z|x) - \log \tilde{q}(z|x) \right) dz dx \\
=& \iint q(x|z)q(z) \left(\log \frac{q(x|z)q(z)}{p(x)} - \log \tilde{q}(z|x) \right) dz dx - \int q(x) \int q(z|x) \log \frac{q(z|x)}{\tilde{q}(z|x)} dz dx \\
=& U(q, \tilde{q}, p) - \mathbb{E}_{q(x)} \left[\mathrm{KL}\left(q(z|x) \,\|\, \tilde{q}(z|x)\right) \right].
\end{aligned}$$

After plugging the E-Step, i.e., $\tilde{q}(z|x) = q_{\text{old}}(x|z)q_{\text{old}}(z)/q_{\text{old}}(x)$, into the objective it simplifies to

$$\begin{aligned}
&U(q, \tilde{q}, p) \\
=& \iint q(x|z)q(z) \left(\log \frac{q(x|z)q(z)}{p(x)} - \log \frac{q_{\text{old}}(x|z)q_{\text{old}}(z)}{q_{\text{old}}(x)} \right) dz dx \\
=& \iint q(x|z)q(z) \left(\log q(x|z) + \log q(z) - \log p(x) - \log q_{\text{old}}(x|z) - \log q_{\text{old}}(z) + \log q_{\text{old}}(x) \right) dz dx \\
=& \iint q(x|z)q(z) \left(\log \frac{q_{\text{old}}(x)}{p(x)} + \log \frac{q(x|z)}{q_{\text{old}}(x|z)} + \log \frac{q(z)}{q_{\text{old}}(z)} \right) dz dx \\
=& \int q(z) \left(\int q(x|z) \left(\log \frac{q_{\text{old}}(x)}{p(x)} + \log \frac{q(x|z)}{q_{\text{old}}(x|z)} \right) dx + \log \frac{q(z)}{q_{\text{old}}(z)} \right) dz \\
=& \int q(z) \int q(x|z) \log \frac{q_{\text{old}}(x)}{p(x)} dx dz + \int q(z) \int q(x|z) \log \frac{q(x|z)}{q_{\text{old}}(x|z)} dx dz + \int q(z) \log \frac{q(z)}{q_{\text{old}}(z)} dz \\
=& \iint q(x|z)q(z) \log \frac{q_{\text{old}}(x)}{p(x)} dz dx + \mathbb{E}_{q(z)} \left[\mathrm{KL}\left(q(x|z) \,\|\, q_{\text{old}}(x|z)\right) \right] + \mathrm{KL}\left(q(z) \,\|\, q_{\text{old}}(z)\right),
\end{aligned}$$

which concludes the derivation of Equation 3.3.

A.2 Conditional Upper Bound

By introducing an auxiliary distribution $\tilde{q}(z|x,y)$ the upper bound to the expected KL for conditional latent variable models $q(x|y)=\int q(x|z,y)q(z|y)dz$, stated in Equation 3.6 can be derived by

$$\begin{aligned}
&\mathbb{E}_{p(y)}\mathrm{KL}\left(q(x|y)\,\|\,p(x|y)\right)=\iint p(y)q(x|y)\log\frac{q(x|y)}{p(x|y)}dxdy\\
=&\int p(y)\iint q(x|z,y)q(z|y)\left(\log\frac{q(x|z,y)q(z|y)}{p(x|y)}-\log q(z|x,y)\right)dzdxdy\\
=&\int p(y)\iint q(x|z,y)q(z|y)\\
&\cdot\left(\log\frac{q(x|z,y)q(z|y)}{p(x|y)}-\log q(z|x,y)+\log\tilde{q}(z|x,y)-\log\tilde{q}(z|x,y)\right)dzdxdy\\
=&\int p(y)\iint q(x|z,y)q(z|y)\left(\log\frac{q(x|z,y)q(z|y)}{p(x|y)}-\log\tilde{q}(z|x,y)\right)dzdxdy\\
&-\int p(y)\iint q(x|z,y)q(z,y)\left(\log q(z|x,y)-\log\tilde{q}(z|x,y)\right)dzdxdy\\
=&\int p(y)\iint q(x|z,y)q(z|y)\left(\log\frac{q(x|z,y)q(z|y)}{p(x|y)}-\log\tilde{q}(z|x,y)\right)dzdxdy\\
&-\iint p(y)q(x|y)\int q(z|x,y)\log\frac{q(z|x,y)}{\tilde{q}(z|x,y)}dzdxdy\\
=&U(q,\tilde{q},p)-\mathbb{E}_{p(y),q(x|y)}\left[\mathrm{KL}\left(q(z|x,y)\,\|\,\tilde{q}(z|x,y)\right)\right].
\end{aligned}$$

During the E-step the bound is tightened by setting $\tilde{q}(z|x,y)=q_{\text{old}}(x|z,y)q_{\text{old}}(z|y)/q_{\text{old}}(x|y)$.

$$\begin{aligned}
&U(q,\tilde{q},p)\\
=&\int p(y)\iint q(x|z,y)q(z|y)\left(\log\frac{q(x|z,y)q(z|y)}{p(x|y)}-\log\frac{q_{\text{old}}(x|z,y)q_{\text{old}}(z|y)}{q_{\text{old}}(x|y)}\right)dzdxdy\\
=&\int p(y)\iint q(x|z,y)q(z|y)\\
&\cdot\left(\log q(x|z,y)+\log q(z|y)-\log p(x|y)-\log q_{\text{old}}(x|z,y)-\log q_{\text{old}}(z|y)+\log q_{\text{old}}(x|y)\right)dzdxdy\\
=&\int p(y)\iint q(x|z,y)q(z|y)\left(\log\frac{q_{\text{old}}(x|y)}{p(x|y)}+\log\frac{q(x|z,y)}{q_{\text{old}}(x|z,y)}+\log\frac{q(z|y)}{q_{\text{old}}(z|y)}\right)dzdxdy\\
=&\int p(y)\int q(z|y)\left(\int q(x|z,y)\left(\log\frac{q_{\text{old}}(x|y)}{p(x|y)}+\log\frac{q(x|z,y)}{q_{\text{old}}(x|z,y)}\right)dx+\log\frac{q(z|y)}{q_{\text{old}}(z|y)}\right)dzdy\\
=&\int p(y)\int q(z|y)\int q(x|z,y)\log\frac{q_{\text{old}}(x|y)}{p(x|y)}dxdzdy\\
&+\int p(y)\int q(z|y)\int q(x|z,y)\log\frac{q(x|z,y)}{q_{\text{old}}(x|z,y)}dxdzdy+\int p(y)\int q(z|y)\log\frac{q(z|y)}{q_{\text{old}}(z|y)}dzdy\\
=&\iiint p(y)q(z|y)q(x|z,y)\log\frac{q_{\text{old}}(x|y)}{p(x|y)}dxdzdy\\
&+\mathbb{E}_{p(y),q(z|y)}\left[\mathrm{KL}\left(q(x|z,y)\,\|\,q_{\text{old}}(x|z,y)\right)\right]+\mathbb{E}_{p(y)}\left[\mathrm{KL}\left(q(z|y)\,\|\,q_{\text{old}}(z|y)\right)\right],
\end{aligned}$$

which concludes the derivation of Equation 3.7.

A.3 M-Step for Gaussian Mixture Models

We derive the dual for optimization problems of the form

$$\max_{q(x)} \int q(x)\phi(x)dx - \mathrm{KL}\left(q(x) \,\|\, q_{\mathrm{old}}(x)\right)$$
$$\text{s.t.} \quad \mathrm{KL}\left(q(x) \,\|\, q_{\mathrm{old}}(x)\right) \leq \epsilon, \quad \mathrm{H}(q(x)) \geq \beta, \quad \int q(x)dx = 1.$$

The derivations can be applied to obtain both, the dual to update the components $q(\mathbf{x}|z_i)$ and the mixture distribution $q(z)$. We begin by considering the corresponding Lagrangian. We use the Lagrangian multipliers η, ω, and λ for the KL, entropy and normalization constraint respectively.

$$\begin{aligned}
&\mathcal{L}(q(x), \eta, \omega, \lambda) \\
=& \int q(x)\phi(x)dx - \mathrm{KL}\left(q(x) \,\|\, q_{\mathrm{old}}(x)\right) + \eta\left(\epsilon - \mathrm{KL}\left(q(x) \,\|\, q_{\mathrm{old}}(x)\right)\right) \\
&+ \omega\left(\mathrm{H}(q(x)) - \beta\right) + \lambda\left(1 - \int q(x)dx\right) \qquad \text{(A.1)} \\
=&\eta\epsilon - \omega\beta + \lambda + \int q(x)\phi(x)dx - (\eta+1)\mathrm{KL}\left(q(x) \,\|\, q_{\mathrm{old}}(x)\right) + \omega\mathrm{H}(q(x)) - \lambda \int q(x)dx \\
=&\eta\epsilon - \omega\beta + \lambda + \int q(x)\left(\phi(x) - (\eta+1)\left(\log q(x) - \log q_{\mathrm{old}}(x)\right) - \omega \log q(x) - \lambda\right)dx \\
=&\eta\epsilon - \omega\beta + \lambda + \int q(x)\left(\phi(x) - (\eta+1+\omega)\log q(x) + (\eta+1)\log q_{\mathrm{old}}(x)\right) - \lambda\right)dx.
\end{aligned}$$

We take the derivative of the Lagrangian w.r.t $q(x)$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial q(x)} &= \int \frac{\partial}{\partial q(x)} q(x)\left(\phi(x) - (\eta+1+\omega)\log q(x) + (\eta+1)\log q_{\mathrm{old}}(x)\right) - \lambda\right)dx \\
&= \int \left(q(x)\frac{-(\eta+1+\omega)}{q(x)} + \phi(x) - (\eta+1+\omega)\log q(x) + (\eta+1)\log q_{\mathrm{old}}(x)) - \lambda\right)dx \\
&= \int -(\eta+1+\omega+\lambda) + \phi(x) - (\eta+1+\omega)\log q(x) + (\eta+1)\log q_{\mathrm{old}}(x))dx.
\end{aligned}$$

For the optimal $q(x)$ this derivative needs to be zero which is the case if the integral is zero for all x. Thus,

$$\begin{aligned}
&-(\eta+1+\omega+\lambda) + \phi(x) - (\eta+1+\omega)\log q(x) + (\eta+1)\log q_{\mathrm{old}}(x) = 0 \\
\Leftrightarrow&(\eta+1+\omega)\log q(x) = -(\eta+1+\omega+\lambda) + \phi(x) + (\eta+1)\log q_{\mathrm{old}}(x). \qquad \text{(A.2)}
\end{aligned}$$

It follows that the new, optimal $q(x)$ is given by

$$\begin{aligned}
q(x) &= \exp\left(-\frac{\eta+1+\omega+\lambda}{\eta+1+\omega}\right)\exp\left(\frac{\phi(x)}{\eta+1+\omega}\right)q_{\mathrm{old}}(x)^{\frac{\eta+1}{\eta+1+\omega}} \\
&\propto \exp\left(\frac{\phi(x)}{\eta+1+\omega}\right)q_{\mathrm{old}}(x)^{\frac{\eta+1}{\eta+1+\omega}} = \exp\left(\frac{(\eta+1)\log q_{\mathrm{old}}(x) + \phi(x)}{\eta+1+\omega}\right). \qquad \text{(A.3)}
\end{aligned}$$

Not surprisingly this update rule for $q(x)$ is very similar to the MORE update rule and differs only by the 1 that is added to η. This update rule still depends on η and ω which can be obtained by minimizing the dual. In order to obtain the dual from the Lagrangian, we substitute the $(\eta+1+\omega)\log q(x)$ term by Equation A.2

$$\begin{aligned}&\mathcal{L}(q(x),\eta,\omega,\lambda) = \eta\epsilon - \omega\beta + \lambda \\ &+\int q(x)\left(\phi(x)+\eta+1+\omega+\lambda-\phi(x)-(\eta+1)\log q_{\text{old}}(x)+(\eta+1)\log q_{\text{old}}(x)\right)-\lambda)\,dx \\ =&\eta\epsilon-\omega\beta+\lambda+(\eta+1+\omega)\int q(x)dx = \eta\epsilon-\omega\beta+\lambda+\eta+1+\omega,\end{aligned} \tag{A.4}$$

since $\int q(x)dx = 1$ as the third constraints ensures that $q(x)$ is a properly normalized distribution. The proper normalization also implies $\log\int q(x)dx = 0$ and thus

$$\begin{aligned}0 &= \log\left(\int \exp\left(-\frac{\eta+1+\omega+\lambda}{\eta+1+\omega}\right)\exp\left(\frac{(\eta+1)\log q_{\text{old}}(x)+\phi(x)}{\eta+1+\omega}\right)dx\right) \\ &= \log\left(\exp\left(-\frac{\eta+1+\omega+\lambda}{\eta+1+\omega}\right)\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x)+\phi(x)}{\eta+1+\omega}\right)dx\right) \\ &= -\frac{\eta+1+\omega+\lambda}{\eta+1+\omega} + \log\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x)+\phi(x)}{\eta+1+\omega}\right)dx \\ \Leftrightarrow&\lambda+\eta+1+\omega = (\eta+1+\omega)\log\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x)+\phi(x)}{\eta+1+\omega}\right)dx.\end{aligned} \tag{A.5}$$

By plugging Equation A.5 into Equation A.4 we ultimately obtain the dual

$$g(\eta,\omega) = \eta\epsilon-\omega\beta+(\eta+1+\omega)\log\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x)+\phi(x)}{\eta+1+\omega}\right)dx. \tag{A.6}$$

Minimizing the dual is a convex problem and the gradients can directly be read of Equation A.1. They are given by

$$\frac{\partial g(\eta,\omega)}{\partial\eta} = \epsilon - \mathrm{KL}\left(q(x)\,\|\,q_{\text{old}}(x)\right) \quad \text{and} \quad \frac{\partial g(\eta,\omega)}{\partial\omega} = \mathrm{H}(q(x)) - \beta.$$

Those gradients allow us to efficiently optimize the dual using gradient based optimizers.

A.3.1 M-Step for Categorical Mixture Distributions

For the discrete categorical distribution, the integral in Equation A.6 simplifies to a sum. Thus, closed form computation is straight forward and can be numerically stabilized by the log-sum-exp trick.

A.3.2 M-Step for Gaussian Components

We derive the closed form update and dual for Gaussian components. In the following, we will work with the natural parametrization of the Gaussian and denote its dimensionality with k. The old distribution is given by

$$\begin{aligned}q_{\text{old}}(\mathbf{x}) &= \mathcal{N}\left(\mathbf{q}_{\text{old}} = \mathbf{\Sigma}_{\text{old}}^{-1}\boldsymbol{\mu}_{\text{old}}, \mathbf{Q}_{\text{old}} = \mathbf{\Sigma}_{\text{old}}^{-1}\right) \\ =&(2\pi)^{-\frac{k}{2}}\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}_{\text{old}}\mathbf{x}+\mathbf{q}_{\text{old}}^T\mathbf{x}-\frac{1}{2}\mathbf{q}_{\text{old}}^T\mathbf{Q}_{\text{old}}^{-1}\mathbf{q}_{\text{old}}+\frac{1}{2}\log\det(\mathbf{Q}_{\text{old}})\right).\end{aligned}$$

The compatible reward surrogate is a quadratic function given by

$$\phi(\mathbf{x}) \approx \hat{\phi}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\hat{\mathbf{R}}\mathbf{x} + \hat{\mathbf{r}}^T\mathbf{x} + \hat{r}_0.$$

We insert both, the reward and the density, into the general update given in Equation A.3 and obtain

$$\begin{aligned}
&q(\mathbf{x})\\
&\propto \exp\left(\frac{\eta+1}{\eta+1+\omega}\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}_{\text{old}}\mathbf{x} + \mathbf{q}_{\text{old}}^T\mathbf{x} - \frac{1}{2}\mathbf{q}_{\text{old}}^T\mathbf{Q}_{\text{old}}^{-1}\mathbf{q}_{\text{old}} + \frac{1}{2}\log\det\left(\mathbf{Q}_{\text{old}}\right) - \frac{k}{2}\log(2\pi)\right)\right.\\
&\quad\left. + \frac{1}{\eta+1+\omega}\left(-\frac{1}{2}\mathbf{x}^T\hat{\mathbf{R}}\mathbf{x} + \hat{\mathbf{r}}^T\mathbf{x} + \hat{r}_0\right)\right)\\
&= \exp\left(-\frac{1}{2}\mathbf{x}^T\left(\frac{(\eta+1)\mathbf{Q}_{\text{old}} + \hat{\mathbf{R}}}{\eta+1+\omega}\right)\mathbf{x} + \left(\frac{(\eta+1)\mathbf{q}_{\text{old}} + \hat{\mathbf{r}}}{\eta+1+\omega}\right)^T\mathbf{x}\right)\\
&\quad\cdot\exp\left(\frac{\eta+1}{\eta+1+\omega}\left(-\frac{1}{2}\mathbf{q}_{\text{old}}^T\mathbf{Q}_{\text{old}}^{-1}\mathbf{q}_{\text{old}} + \frac{1}{2}\log\det\left(\mathbf{Q}_{\text{old}}\right) - \frac{k}{2}\log(2\pi)\right) + \frac{\hat{r}_0}{\eta+1+\omega}\right) \qquad \text{(A.7)}\\
&= \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} + \text{const}\right).
\end{aligned}$$

Where

$$\mathbf{Q} = \frac{(\eta+1)\mathbf{Q}_{\text{old}} + \hat{\mathbf{R}}}{\eta+1+\omega} \quad \text{and} \quad \mathbf{q} = \frac{(\eta+1)\mathbf{q}_{\text{old}} + \hat{\mathbf{r}}}{\eta+1+\omega}$$

denote the natural parameters of the new distribution.

However, to find the optimal parameters we still need to optimize the dual. The required computations could be done based on samples, however, we can also solve the integral in the dual in closed form and obtain an analytical solution dependent only on the natural parameters of the old and new distribution. We start by plugging Equation A.7 into the dual Equation A.6 yields

$$\begin{aligned}
g(\eta,\omega) =& \eta\epsilon - \omega\beta + (\eta+1+\omega)\log\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x}\right) \qquad \text{(A.8)}\\
&\cdot\exp\left(\frac{\eta+1}{\eta+1+\omega}\left(-\frac{1}{2}\mathbf{q}_{\text{old}}^T\mathbf{Q}_{\text{old}}^{-1}\mathbf{q}_{\text{old}} + \frac{1}{2}\log\det\left(\mathbf{Q}_{\text{old}}\right) - \frac{k}{2}\log(2\pi)\right) + \frac{\hat{r}_0}{\eta+1+\omega}\right)d\mathbf{x}\\
=& \eta\epsilon - \omega\beta + (\eta+1)\left(-\frac{1}{2}\mathbf{q}_{\text{old}}^T\mathbf{Q}_{\text{old}}^{-1}\mathbf{q}_{\text{old}} + \frac{1}{2}\log\det\left(\mathbf{Q}_{\text{old}}\right) - \frac{k}{2}\log(2\pi)\right) + \hat{r}_0\\
&+ (\eta+1+\omega)\log\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x}\right)d\mathbf{x}. \qquad \text{(A.9)}
\end{aligned}$$

We consider only the integral part

$$\begin{aligned}
&\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x}\right)d\mathbf{x} = \int \exp\\
&\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} - \frac{1}{2}\mathbf{q}^T\mathbf{Q}^{-1}\mathbf{q} + \frac{1}{2}\mathbf{q}^T\mathbf{Q}^{-1}\mathbf{q} + \frac{1}{2}\log\det\left(\mathbf{Q}\right) - \frac{1}{2}\log\det\left(\mathbf{Q}\right) - \frac{k}{2}\log(2\pi) + \frac{k}{2}\log(2\pi)\right)d\mathbf{x}\\
&= \exp\left(\frac{1}{2}\mathbf{q}^T\mathbf{Q}^{-1}\mathbf{q} - \frac{1}{2}\log\det\left(\mathbf{Q}\right) + \frac{k}{2}\log(2\pi)\right)\\
&\quad\cdot\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} - \frac{1}{2}\mathbf{q}^T\mathbf{Q}^{-1}\mathbf{q} + \frac{1}{2}\log\det\left(\mathbf{Q}\right) - \frac{k}{2}\log(2\pi)\right)d\mathbf{x} \qquad \text{(A.10)}\\
&= \exp\left(\frac{1}{2}\mathbf{q}^T\mathbf{Q}^{-1}\mathbf{q} - \frac{1}{2}\log\det\left(\mathbf{Q}\right) + \frac{k}{2}\log(2\pi)\right).
\end{aligned}$$

The integral Equation A.10 is again the definition of the Gaussian density with natural parameters and hence equals to 1. Ultimately, inserting the resulting term back into Equation A.9 gives us a closed form solution that depends only on the natural parameters of the old and new distribution and it does not require computing any integrals

$$\begin{aligned} g(\eta,\omega) =& \eta\epsilon - \omega\beta + (\eta+1)\left(-\frac{1}{2}\mathbf{q}_{\text{old}}^T\mathbf{Q}_{\text{old}}^{-1}\mathbf{q}_{\text{old}} + \frac{1}{2}\log\det\left(\mathbf{Q}_{\text{old}}\right) - \frac{k}{2}\log(2\pi)\right) + \hat{r}_0 \\ &+ (\eta+1+\omega)\left(\frac{1}{2}\mathbf{q}^T\mathbf{Q}^{-1}\mathbf{q} - \frac{1}{2}\log\det\left(\mathbf{Q}\right) + \frac{k}{2}\log(2\pi)\right). \end{aligned}$$

Note that we can neglect $\hat{r}_0$ as it is constant.

A.4 M-Step for Mixtures of Experts

We derive the dual for optimization problems of the form

$$\begin{aligned} &\max_{q(x|y)} \int p(y) \int q(x|y)\phi(x,y)dx - \text{KL}\left(q(x|y) \,\|\, q_{\text{old}}(x|y)\right) \\ \text{s.t.} \quad & \mathbb{E}_{p(y)}\left[\text{KL}\left(q(x|y) \,\|\, q_{\text{old}}(x|y)\right)\right] \leq \epsilon, \quad \mathbb{E}_{p(y)}\left[\text{H}(q(x|y))\right] \geq \beta, \quad \forall y : \int q(x|y)dx = 1. \end{aligned}$$

The derivations can be applied to obtain both, the dual to update the components $q(\mathbf{x}|z_i,\mathbf{y})$ and the gating distribution $q(z|\mathbf{y})$. We begin by considering the corresponding Lagrangian. We use the Lagrangian multipliers η, ω and $\lambda(y)$ for the expected KL, expected entropy and normalization constraints respectively.

$$\begin{aligned} &\mathcal{L}(q(x,y),\eta,\omega,\lambda(y)) \\ &= \int p(y) \int q(x|y)\phi(x,y)dx - \text{KL}\left(q(x|y) \,\|\, q_{\text{old}}(x|y)\right) + \eta\left(\epsilon - \int p(y)\text{KL}\left(q(x|y) \,\|\, q_{\text{old}}(x|y)\right)dy\right) \\ &\quad + \omega\left(\left(\int p(y)\text{H}(q(x|y))dy - \beta\right) + \int \lambda(y)\left(1 - \int q(x|y)dx\right)dy\right. \qquad \text{(A.11)} \\ &= \eta\epsilon - \omega\beta + \int \lambda(y)dy + \int p(y)\int q(x|y)\phi(x,y)dxdy - (\eta+1)\int p(y)\text{KL}\left(q(x|y) \,\|\, q_{\text{old}}(x|y)\right)dy \\ &\quad + \omega \int p(y)\text{H}(q(x|y))dy - \int \lambda(y)\int q(x|y)dxdy \\ &= \eta\epsilon - \omega\beta + \int \lambda(y)dy - \iint q(x|y)\lambda(y)dxdy \\ &\quad + \int p(y)\int q(x|y)\left(\phi(x,y) - (\eta+1)\left(\log q(x|y) - \log q_{\text{old}}(x|y)\right) - \omega\log(q(x|y))\right)dxdy \\ &= \eta\epsilon - \omega\beta + \int \lambda(y)dy \\ &\quad + \iint q(x|y)\left(p(y)\left(\phi(x,y) - (\eta+1+\omega)\log q(x|y) + (\eta+1)\log q_{\text{old}}(x|y)\right) - \lambda(y)\right)dxdy. \end{aligned}$$

We take the derivative of the Lagrangian w.r.t $q(x|y)$

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial q(x|y)}\\
&=\iint \frac{\partial}{\partial q(x|y)} q(x|y)\left(p(y)\left(\phi(x,y)-(\eta+1+\omega)\log q(x|y)+(\eta+1)\log q_{\text{old}}(x|y)\right)-\lambda(y)\right)dxdy\\
&=\iint \frac{-p(y)(\eta+1+\omega)}{q(x|y)} q(x|y)\\
&\quad+p(y)\left(\phi(x,y)-(\eta+1+\omega)\log q(x|y)+(\eta+1)\log q_{\text{old}}(x|y)\right)-\lambda(y)dxdy\\
&=\iint p(y)\left(\phi(x,y)-(\eta+1+\omega)\log q(x|y)+(\eta+1)\log q_{\text{old}}(x|y)-(\eta+1+\omega)\right)-\lambda(y)dxdy.
\end{aligned}
$$

For the optimal $q(x|y)$ this derivative needs to be zero which is the case if the integral is zero for all x and y. Thus,

$$
\begin{aligned}
&p(y)\left(\phi(x,y)-(\eta+1+\omega)\log q(x|y)+(\eta+1)\log q_{\text{old}}(x|y)-(\eta+1+\omega)\right)-\lambda(y)=0\\
\Leftrightarrow &p(y)(\eta+1+\omega)\log q(x|y)=p(y)\left(\phi(x,y)+(\eta+1)\log q_{\text{old}}(x|y)-(\eta+1+\omega)\right)-\lambda(y)\\
&=p(y)\phi(x,y)+p(y)(\eta+1)\log q_{\text{old}}(x|y)-p(y)(\eta+1+\omega)-\lambda(y). \qquad \text{(A.12)}
\end{aligned}
$$

It follows that the new, optimal $q(x)$ is given by

$$
\begin{aligned}
q(x|y)&=\exp\left(-\frac{p(y)(\eta+1+\omega)+\lambda(y)}{\eta+1+\omega}\right)\exp\left(\frac{p(y)\phi(x,y)}{p(y)(\eta+1+\omega)}\right)q_{\text{old}}(x|y)^{\frac{p(y)(\eta+1)}{p(y)(\eta+1+\omega)}}\\
&\propto \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x|y)+\phi(x,y)}{\eta+1+\omega}\right). \qquad \text{(A.13)}
\end{aligned}
$$

This update rule still depends on η and ω which can be obtained by minimizing the dual. In order to obtain the dual from the Lagrangian, we substitute the $p(y)(\eta+1+\omega)\log q(x)$ term by Equation A.12

$$
\begin{aligned}
=&\eta\epsilon-\omega\beta+\int \lambda(y)dy\\
&+\iint q(x|y)\left(p(y)\left(\phi(x,y)-(\eta+1+\omega)\log q(x|y)+(\eta+1)\log q_{\text{old}}(x|y)\right)-\lambda(y)\right)dxdy\\
=&\eta\epsilon-\omega\beta+\int \lambda(y)dy\\
&+\iint q(x|y)\left(p(y)\phi(x,y)-p(y)(\eta+1+\omega)\log q(x|y)+p(y)(\eta+1)\log q_{\text{old}}(x|y)-\lambda(y)\right)dxdy\\
=&\eta\epsilon-\omega\beta+\int \lambda(y)dy+\iint q(x|y)(p(y)\phi(x,y)-p(y)\phi(x,y)-p(y)(\eta+1)\log q_{\text{old}}(x|y)\\
&+p(y)(\eta+1+\omega)+\lambda(y)+p(y)(\eta+1)\log q_{\text{old}}(x|y)-\lambda(y))dxdy\\
=&\eta\epsilon-\omega\beta+\int \lambda(y)dy+(\eta+1+\omega)\int p(y)\int q(x|y)dxdy\\
=&\eta\epsilon-\omega\beta+\int p(y)(\eta+1+\omega)+\lambda(y)dy, \qquad \text{(A.14)}
\end{aligned}
$$

since $\int q(x|y)dx = 1$ as the third constraint ensures that $q(x|y)$ is properly normalized for all y. The proper normalization also implies $\log \int q(x|y)dx = 0$ for all y and thus

$$\begin{aligned}
0 &= \log\left(\int \exp\left(-\frac{p(y)(\eta+1+\omega)+\lambda(y)}{\eta+1+\omega}\right)\exp\left(\frac{(\eta+1)\log q_{\text{old}}(x|y)+\phi(x,y)}{\eta+1+\omega}\right)dx\right) \\
&= \log\left(\exp\left(-\frac{p(y)(\eta+1+\omega)+\lambda(y)}{\eta+1+\omega}\right)\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x|y)+\phi(x,y)}{\eta+1+\omega}\right)dx\right) \\
&= -\frac{p(y)(\eta+1+\omega)+\lambda(y)}{\eta+1+\omega} + \log\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x|y)+\phi(x,y)}{\eta+1+\omega}\right)dx \\
\Leftrightarrow & p(y)(\eta+1+\omega)+\lambda(y) = (\eta+1+\omega)\log\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x|y)+\phi(x,y)}{\eta+1+\omega}\right)dx. \qquad \text{(A.15)}
\end{aligned}$$

By plugging Equation A.15 into Equation A.14 we ultimately obtain the dual

$$g(\eta,\omega) = \eta\epsilon - \omega\beta + (\eta+1+\omega)\int p(y)\log\int \exp\left(\frac{(\eta+1)\log q_{\text{old}}(x|y)+\phi(x,y)}{\eta+1+\omega}\right)dxdy. \qquad \text{(A.16)}$$

Minimizing the dual is a convex problem and the gradients can directly be read of Equation A.11. They are given by

$$\frac{\partial g(\eta,\omega)}{\partial \eta} = \epsilon - \int p(y)\mathrm{KL}(q(x|y)\,\|\,q_{\text{old}}(x|y))\,dy \quad \text{and} \quad \frac{\partial g(\eta,\omega)}{\partial \omega} = \int p(y)\mathrm{H}(q(x|y))dy - \beta, \qquad \text{(A.17)}$$

which again allows efficient optimization using gradient based optimizers.

A.4.1 M-Step for Softmax Gating distribution

The gradients of the dual stated in Equation A.17 depend on the updated model. When working with the MORE-like closed form updates this is efficient, however, when working with the sample based update it would mean fitting a model using softmax regression in each iteration. This is neither fast nor precise. Hence, we derive gradients of the reformulated dual

$$g(\eta,\omega) = \eta\epsilon - \omega\beta + (\eta+1+\omega)\int p(\mathbf{y})\log\sum_{i=1}^{d}\exp\left(\frac{(\eta+1)\log q_{\text{old}}(z_i|\mathbf{y})+\phi(\mathbf{y},z_i)}{\eta+1+\omega}\right)d\mathbf{y}.$$

To this end, we define auxiliary functions

$$\begin{aligned}
t_1(\eta,\omega,i,\mathbf{y}) &= \frac{(\eta+1)\log q_{\text{old}}(z_i|\mathbf{y})+\phi(\mathbf{y},z_i)}{\eta+1+\omega} \\
t_2(\eta,\omega) &= \int p(\mathbf{y})\log\sum_{i=1}^{d}\exp\left(\frac{(\eta+1)\log q_{\text{old}}(z_i|\mathbf{y})+\phi(\mathbf{y},z_i)}{\eta+1+\omega}\right)d\mathbf{y}.
\end{aligned}$$

Their gradients are given by

$$
\begin{aligned}
\frac{\partial t_1(\eta,\omega,i,\mathbf{y})}{\partial \eta} &= \frac{\log \tilde{p}(z_i|\mathbf{y})(\eta+1+\omega)-(\eta+1)\log \tilde{p}(z_i|\mathbf{y})-\phi(\mathbf{y},z_i)}{(\eta+1+\omega)^2} = \frac{\omega \log \tilde{p}(z_i|\mathbf{y})-\phi(\mathbf{y},z_i)}{(\eta+1+\omega)^2}\\
\frac{\partial t_1(\eta,\omega,i,\mathbf{y})}{\partial \omega} &= -\frac{(\eta+1)\log \tilde{p}(z_i|\mathbf{y})+\phi(\mathbf{y},z_i)}{(\eta+1+\omega)^2}\\
\frac{\partial t_2(\eta,\omega)}{\partial \eta} &= \int p(\mathbf{y}) \frac{1}{\sum_{i=1}^d \exp t_1(\eta,\omega,i,\mathbf{y})} \sum_{i=1}^d \frac{\partial t_1(\eta,\omega,i,\mathbf{y})}{\partial \eta} \exp(t_1(\eta,\omega,i,\mathbf{y}))d\mathbf{y}\\
\frac{\partial t_2(\eta,\omega)}{\partial \omega} &= \int p(\mathbf{y}) \frac{1}{\sum_{i=1}^d \exp t_1(\eta,\omega,i,\mathbf{y})} \sum_{i=1}^d \frac{\partial t_1(\eta,\omega,i,\mathbf{y})}{\partial \omega} \exp(t_1(\eta,\omega,i,\mathbf{y}))d\mathbf{y}.
\end{aligned}
$$

Using those, the gradient of the dual w.r.t. the Lagrangian multipliers is given by

$$
\begin{aligned}
\frac{\partial g(\eta,\omega)}{\partial \eta} &= \epsilon + t_2(\eta,\omega) + (\eta+\omega+1)\frac{\partial t_2(\eta,\omega)}{\partial \eta}\\
\frac{\partial g(\eta,\omega)}{\partial \omega} &= -\beta + t_2(\eta,\omega) + (\eta+\omega+1)\frac{\partial t_2(\eta,\omega)}{\partial \omega}.
\end{aligned}
$$

A.4.2 M-Step for Linear Components

We derive the closed form update and dual for Linear components $q(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{W}\mathbf{x}+\mathbf{w},\boldsymbol{\Sigma})$. We denote the dimensionality of $\mathbf{x}$ by k_x and the dimensionality of $\mathbf{y}$ by k_y. In the following we work with the natural parametrization of the Gaussian. We denote the precision matrix by $\mathbf{Q} = \boldsymbol{\Sigma}^{-1}$ and use the auxiliary parameters $\mathbf{L} = \boldsymbol{\Sigma}^{-1}\mathbf{W}$ and $\mathbf{l} = \boldsymbol{\Sigma}^{-1}\mathbf{w}$. The old distribution is given by

$$
\begin{aligned}
&q_{\text{old}}(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{L}_{\text{old}}\mathbf{y}+\mathbf{l}_{\text{old}},\mathbf{Q}_{\text{old}})\\
&= \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}_{\text{old}}\mathbf{x} + (\mathbf{L}_{\text{old}}\mathbf{y}+\mathbf{l}_{\text{old}})^T\mathbf{x} - \frac{1}{2}(\mathbf{L}_{\text{old}}\mathbf{y}+\mathbf{l}_{\text{old}})^T\mathbf{Q}_{\text{old}}^{-1}(\mathbf{L}_{\text{old}}\mathbf{y}+\mathbf{l}_{\text{old}}) + \frac{1}{2}\log\det(\mathbf{Q}_{\text{old}}) - \frac{k}{2}\log(2\pi)\right)\\
&= \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}_{\text{old}}\mathbf{x} + \mathbf{x}^T\mathbf{L}_{\text{old}}\mathbf{y} + \mathbf{l}_{\text{old}}^T\mathbf{x} - d_{\text{old}}(\mathbf{y})\right),
\end{aligned}
$$

where $d_{\text{old}}(\mathbf{y})$ summarizes all parts that either depend only on y or are constant. The compatible reward surrogate is a quadratic function of the form

$$
\begin{aligned}
\phi(\mathbf{x},\mathbf{y}) \approx \hat{\phi}(\mathbf{x},\mathbf{y}) &= -\frac{1}{2}\mathbf{x}^T\hat{\mathbf{R}}_{xx}\mathbf{x} + \mathbf{x}^T\hat{\mathbf{R}}_{xy}\mathbf{y} - \frac{1}{2}\mathbf{y}^T\hat{\mathbf{R}}_{yy}\mathbf{y} + \hat{\mathbf{r}}_x^T\mathbf{x} + \hat{\mathbf{r}}_y^T\mathbf{y} + \hat{r}_0\\
&= -\frac{1}{2}\mathbf{x}^T\hat{\mathbf{R}}_{xx}\mathbf{x} + \mathbf{x}^T\hat{\mathbf{R}}_{xy}\mathbf{y} + \hat{\mathbf{r}}_x^T\mathbf{x} + \hat{r}(\mathbf{y}),
\end{aligned}
$$

where $\hat{r}(\mathbf{y})$ summarizes all parts that either depend only on y or are constant. Plugging both the old distribution and reward surrogate into Equation A.13 yields

$$
\begin{aligned}
q(\mathbf{x}|\mathbf{y}) \propto \exp\Bigg(&\frac{\eta+1}{\eta+1+\omega}\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}_{\text{old}}\mathbf{x} + \mathbf{x}^T\mathbf{L}_{\text{old}}\mathbf{y} + \mathbf{l}_{\text{old}}^T\mathbf{x} - d_{\text{old}}(\mathbf{y})\right)\\
&+ \frac{1}{\eta+1+\omega}\left(-\frac{1}{2}\mathbf{x}^T\hat{\mathbf{R}}_{xx}\mathbf{x} + \mathbf{x}^T\hat{\mathbf{R}}_{xy}\mathbf{y} + \hat{\mathbf{r}}_x^T\mathbf{x} + \hat{r}(\mathbf{y})\right)\Bigg)\\
= \exp\Bigg(&-\frac{1}{2}\mathbf{x}^T\left(\frac{(\eta+1)\mathbf{Q}_{\text{old}}+\hat{\mathbf{R}}_{xx}}{\eta+1+\omega}\right)\mathbf{x} + \mathbf{x}^T\left(\frac{(\eta+1)\mathbf{L}_{\text{old}}+\hat{\mathbf{R}}_{xy}}{\eta+1+\omega}\right)\mathbf{y} + \left(\frac{(\eta+1)\mathbf{l}_{\text{old}}+\hat{\mathbf{r}}_x}{\eta+1+\omega}\right)^T\mathbf{x}\\
&+ \frac{(\eta+1)d_{\text{old}}(\mathbf{y})+\hat{r}(\mathbf{y})}{\eta+1+\omega}\Bigg)\\
= \exp\Bigg(&-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{L}\mathbf{y} + \mathbf{l}^T\mathbf{x} + \frac{(\eta+1)d_{\text{old}}(\mathbf{y})+\hat{r}(\mathbf{y})}{\eta+1+\omega}\Bigg). \qquad \text{(A.18)}
\end{aligned}
$$

Hence, the new precision matrix and auxiliary parameters are given by

$$\mathbf{Q} = \left(\frac{(\eta+1)\mathbf{Q}_{\text{old}} + \hat{\mathbf{R}}_{xx}}{\eta+1+\omega}\right), \quad \mathbf{L} = \left(\frac{(\eta+1)\mathbf{L}_{\text{old}} + \hat{\mathbf{R}}_{xy}}{\eta+1+\omega}\right), \quad \mathbf{l} = \left(\frac{(\eta+1)\mathbf{l}_{\text{old}} + \hat{\mathbf{r}}_{x}}{\eta+1+\omega}\right).$$

To get the optimal values for η and ω we need to minimize the dual. To simplify this optimization, we solve the inner integral of Equation A.16 analytically. We start by plugging in Equation A.18

$$\begin{aligned}
&g(\eta,\omega)\\
=&\eta\epsilon - \omega\beta + (\eta+1+\omega)\int p(y)\log\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{L}\mathbf{y} + \mathbf{l}^T\mathbf{x} + \frac{(\eta+1)d_{\text{old}}(\mathbf{y}) + \hat{r}(\mathbf{y})}{\eta+1+\omega}\right)d\mathbf{x}d\mathbf{y}\\
=&\eta\epsilon - \omega\beta + \int p(\mathbf{y})\left((\eta+1)d_{\text{old}}(\mathbf{y}) + \hat{r}(\mathbf{y}) + (\eta+1+\omega)\log\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{L}\mathbf{y} + \mathbf{l}^T\mathbf{x}\right)d\mathbf{x}\right)d\mathbf{y}.
\end{aligned}$$

We consider only the inner integral

$$\begin{aligned}
&\log\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{L}\mathbf{y} + \mathbf{l}^T\mathbf{x}\right)d\mathbf{x}\\
=&\log\int \exp\Bigg(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{L}\mathbf{y} + \mathbf{l}^T\mathbf{x} - \frac{1}{2}(\mathbf{L}\mathbf{y}+\mathbf{l})^T\mathbf{Q}^{-1}(\mathbf{L}\mathbf{y}+\mathbf{l}) + \frac{1}{2}\log\det(\mathbf{Q}) - \frac{k}{2}\log(2\pi)\\
&+\frac{1}{2}(\mathbf{L}\mathbf{y}+\mathbf{l})^T\mathbf{Q}^{-1}(\mathbf{L}\mathbf{y}+\mathbf{l}) - \frac{1}{2}\log\det(\mathbf{Q}) + \frac{k}{2}\log(2\pi)\Bigg)d\mathbf{x}\\
=&-d(\mathbf{y}) + \log\int \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{L}\mathbf{y} + \mathbf{l}^T\mathbf{x} - \frac{1}{2}(\mathbf{L}\mathbf{y}+\mathbf{l})^T\mathbf{Q}^{-1}(\mathbf{L}\mathbf{y}+\mathbf{l}) + \frac{1}{2}\log\det(\mathbf{Q}) - \frac{k}{2}\log(2\pi)\right)\\
=&-d(\mathbf{y}).
\end{aligned}$$

The log integral vanishes since it is equal to the density of the linear Gaussian model using the natural parameterization and hence integrates to 1. The dual is thus given by

$$\begin{aligned}
g(\eta,\omega) &= \eta\epsilon - \omega\beta + \int p(\mathbf{y})\left((\eta+1)d_{\text{old}}(\mathbf{y}) + \hat{r}(\mathbf{y}) - (\eta+1+\omega)d(\mathbf{y})\right)d\mathbf{y}\\
&\propto \eta\epsilon - \omega\beta + \int p(\mathbf{y})\left((\eta+1)d_{\text{old}}(\mathbf{y}) - (\eta+1+\omega)d(\mathbf{y})\right)d\mathbf{y}.
\end{aligned}$$

Similar to [Akrour et al., 2018], under the assumption that $p(\mathbf{y})$ is Gaussian distributed, we could solve the remaining integral in closed form. Yet, we refrain from making the Gaussian assumption and work with a sample based approximation instead.

A.4.3 M-Step for Nonlinear Components

We derive gradients to optimize the dual efficiently

$$g(\eta,\omega) = \eta\epsilon - \omega\beta + (\eta+1+\omega)\int p(\mathbf{y})q_{\text{old}}(\mathbf{x}|\mathbf{y})\exp\left(\frac{(-\omega\log q_{\text{old}}(\mathbf{x}|\mathbf{y}) + \phi(\mathbf{x},\mathbf{y}}{\eta+1+\omega}\right)d\mathbf{x}d\mathbf{y}.$$

Note that we omit the dependency on z_i for brevity. We define auxiliary functions

$$\begin{aligned} t_1(\eta, \omega, \mathbf{x}, \mathbf{y}) &= \frac{-\omega \log q_{\text{old}}(\mathbf{x}|\mathbf{y}) + \phi(\mathbf{x}, \mathbf{y})}{\eta + 1 + \omega} \\ t_2(\eta, \omega) &= \int p(\mathbf{y}) \log \int q_{\text{old}}(\mathbf{x}|\mathbf{y}) \exp\left(\frac{-\omega \log q_{\text{old}}(\mathbf{x}|\mathbf{y}) + \phi(\mathbf{x}, \mathbf{y})}{\eta + 1 + \omega} \right) d\mathbf{x} d\mathbf{y}. \end{aligned}$$

Their gradients are given by

$$\begin{aligned} \frac{\partial t_1(\eta, \omega), \mathbf{x}, \mathbf{y}}{\partial \eta} &= \frac{\omega \log q_{\text{old}}(\mathbf{x}|\mathbf{y}) - \phi(\mathbf{x}, \mathbf{y})}{(\eta + 1 + \omega)^2} \\ \frac{\partial t_1(\eta, \omega, , \mathbf{x}, \mathbf{y})}{\partial \omega} &= \frac{-\log q_{\text{old}}(\mathbf{x}|\mathbf{y})(\eta + 1 + \omega) + \omega \log q_{\text{old}}(\mathbf{x}|\mathbf{y}) - \phi(\mathbf{x}, \mathbf{y})}{(\eta + 1 + \omega)^2} \\ &= \frac{-\log q_{\text{old}}(\mathbf{x}|\mathbf{y})(\eta + 1) - \phi(\mathbf{x}, \mathbf{y})}{(\eta + 1 + \omega)^2} \\ \frac{\partial t_2(\eta, \omega)}{\partial \eta} &= \int p(\mathbf{y}) \frac{1}{\int \exp t_1(\eta, \omega, \mathbf{x}, \mathbf{y}) d\mathbf{x}} \int \frac{\partial t_1(\eta, \omega, \mathbf{x}, \mathbf{y})}{\partial \eta} \exp t_1(\eta, \omega, \mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\ \frac{\partial t_2(\eta, \omega)}{\partial \omega} &= \int p(\mathbf{y}) \frac{1}{\int \exp t_1(\eta, \omega, \mathbf{x}, \mathbf{y}) d\mathbf{x}} \int \frac{\partial t_1(\eta, \omega, \mathbf{x}, \mathbf{y})}{\partial \omega} \exp t_1(\eta, \omega, \mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}. \end{aligned}$$

Using those, the gradient of the dual w.r.t. the Lagrangian multipliers is given by

$$\begin{aligned} \frac{\partial g(\eta, \omega)}{\partial \eta} &= \epsilon + t_2(\eta, \omega) + (\eta + \omega + 1) \frac{\partial t_2(\eta, \omega)}{\partial \eta} \\ \frac{\partial g(\eta, \omega)}{\partial \omega} &= -\beta + t_2(\eta, \omega) + (\eta + \omega + 1) \frac{\partial t_2(\eta, \omega)}{\partial \omega}. \end{aligned}$$

B Hyperparameters

Parameter	6.1	6.2	6.3	6.4	6.5
			EIM		
ϵ components	0.005	0.005	0.005	0.005	0.005
ϵ weights	0.005	0.005	0.005	0.005	0.005
β_{loss} components	0.05	0.05	0.05	0.01	0.01
β_{loss} weights	0.05	0.05	0.05	0.1	0.01
			Density Ratio Estimator		
Hidden Layers	$[50, 50, 50]$	$[150, 150, 150]$	$[100, 100]$	$[400, 400, 400]$	$[150, 150, 150]$
Dropout Probability	0.1	0.1	0.0	0.1	0.1
Early Stopping	×	✓	✓	✓	✓
			Data		
Train Samples	10,000	5,000	10,000	10,000	8,000
Test Samples	5,000	5,000	5,000	5,000	3,430
Validation Samples	-	5,000	5,000	5,000	3,430

Table B.1.: Hyperparamters used for experiments described in sections 6.1 through 6.5

Parameter	Value	Parameter	Value
ϵ components	0.01	Density Ratio Estimator	
ϵ weights	0.01	Hidden Layers	$[150, 150, 150$
β_{loss} components	0.05	Dropout Probability	0.1
β_{loss} weights	0.05	Early Stopping	✓
Train Samples	5,000	Parameter Network	
Test Samples	2,000	Hidden Layers	$[50, 50]$
Validation Samples	5,000	$L2$-regularization	0.001

Table B.2.: Hyperparamters used for experiments in section 6.6