entropy

MDPI

Article

MEMe: An Accurate Maximum Entropy Method for Efficient Approximations in Large-Scale Machine Learning

Diego Granziol [1,2,*,†], Binxin Ru [1,2,*,†], Stefan Zohren [1,2], Xiaowen Dong [1,2], Michael Osborne [1,2] and Stephen Roberts [1,2]

[1] Machine Learning Research Group, University of Oxford, Walton Well Rd, Oxford OX2 6ED, UK; zohren@robots.ox.ac.uk (S.Z.); xdong@robots.ox.ac.uk (X.D.); mosb@robots.ox.ac.uk (M.O.); sjrob@robots.ox.ac.uk (S.R.)

[2] Oxford-Man Institute of Quantitative Finance, Walton Well Rd, Oxford OX2 6ED, UK

* Correspondence: diego@robots.ox.ac.uk (D.G.); robin@robots.ox.ac.uk (B.R.)

† These authors contributed equally to this work.

Received: 30 April 2019; Accepted: 29 May 2019; Published: 31 May 2019

Abstract: Efficient approximation lies at the heart of large-scale machine learning problems. In this paper, we propose a novel, robust maximum entropy algorithm, which is capable of dealing with hundreds of moments and allows for computationally efficient approximations. We showcase the usefulness of the proposed method, its equivalence to constrained Bayesian variational inference and demonstrate its superiority over existing approaches in two applications, namely, fast log determinant estimation and information-theoretic Bayesian optimisation.

Keywords: maximum entropy; log determinant estimation; Bayesian optimisation

1. Introduction

Algorithmic scalability is an important component of modern machine learning. Making high quality inference on large, feature rich datasets under a constrained computational budget is arguably the primary goal of the learning community. This, however, comes with significant challenges. On the one hand, the exact computation of linear algebraic quantities may be prohibitively expensive, such as that of the log determinant. On the other hand, an analytic expression for the quantity of interest may not exist at all, such as the case for the entropy of a Gaussian mixture model, and approximate methods are often both inefficient and inaccurate. These highlight the need for efficient approximations especially in solving large-scale machine learning problems.

In this paper, to address this challenge, we propose a novel, robust maximum entropy algorithm, stable for a large number of moments, surpassing the limit of previous maximum entropy algorithms [1–3]. We show that the ability to handle more moment information, which can be calculated cheaply either analytically or with the use of stochastic trace estimation, leads to significantly enhanced performance. We showcase the effectiveness of the proposed algorithm by applying it to log determinant estimation [4–6] and entropy term approximation in the information-theoretic Bayesian optimisation [7–9]. Specifically, we reformulate the log determinant estimation into an eigenvalue spectral estimation problem so that we can estimate the log determinant of a symmetric positive definite matrix via computing the maximum entropy spectral density of its eigenvalues. Similarly, we learn the maximum entropy spectral density for the Gaussian mixture and then approximate the entropy of the Gaussian mixture via the entropy of the maximum entropy spectral density, which provides an analytic upper bound. Furthermore, in developing our algorithm, we establish equivalence between maximum entropy methods and constrained Bayesian variational inference [10].

The main contributions of the paper are as follows:

- We propose a maximum entropy algorithm, which is stable and consistent for hundreds of moments, surpassing other off-the-shelf algorithms with a limit of a small number of moments. Based on this robust algorithm, we develop a new Maximum Entropy Method (MEMe) which improves upon the scalability of existing machine learning algorithms by efficiently approximating computational bottlenecks using maximum entropy and fast moment estimation techniques;
- We establish the link between maximum entropy methods and variational inference under moment constraints, hence connecting the former to well-known Bayesian approximation techniques;
- We apply MEMe to the problem of estimating the log determinant, crucial to inference in determinental point processes [11], and to that of estimating the entropy of a Gaussian mixture, important to state-of-the-art information-theoretic Bayesian optimisation algorithms.

2. Theoretical Framework

The method of maximum entropy, hereafter referred to as *MaxEnt* [12], is a procedure for generating the most conservative estimate of a probability distribution with the given information and the most non-committal one with regard to missing information [13]. Intuitively, in a bounded domain, the most conservative distribution, i.e., the distribution of maximum entropy, is the one that assigns equal probability to all the accessible states. Hence, the method of maximum entropy can be thought of as choosing the flattest, or most equiprobable distribution, satisfying the given moment constraints.

To determine the maximally entropic density $q(x)$, we maximise the entropic functional

$$S = -\int q(x)\log q(x)dx - \sum_{i=0}^{m}\alpha_i\left[\int q(x)x^i dx - \mu_i\right], \tag{1}$$

with respect to $q(x)$, where the second term with $\mu_i = \mathbb{E}_p[x^i]$ for some density $p(x)$ are the power moment constraints on the density $q(x)$, $\{\alpha_i\}$ are the corresponding Lagrange multipliers, and m is the number of moments. The first term in Equation (1) is referred to as the Boltzmann–Shannon–Gibbs (BSG) entropy, which has been applied in multiple fields, ranging from condensed matter physics [14] to finance [15]. For the case of $i \leq 2$, the Lagrange multipliers can be calculated analytically; for $i \geq 3$, they must be determined numerically.

In this section, we first establish links between the method of MaxEnt and Bayesian variational inference. We then describe fast moment estimation techniques. Finally, we present the proposed MaxEnt algorithm.

2.1. Maximum Entropy as Constrained Bayesian Variational Inference

The work of Bretthorst [16] makes the claim that the method of maximum entropy (MaxEnt) is fundamentally at odds with Bayesian inference. At the same time, variational inference [17] is a widely used approximation technique that falls under the category of Bayesian learning. In this section, we show that the method of maximum relative entropy [18] is equivalent to constrained variational inference, thus establishing the link between MaxEnt and Bayesian approximation.

2.1.1. Variational Inference

Variational methods [10,17] in machine learning pose the problem of intractable density estimation from the application of Bayes' rule as a functional optimisation problem:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \approx q(z), \tag{2}$$

where $p(z)$ and $p(z|x)$ represent the prior and posterior distributions of the random variable z, respectively. Variational inference therefore seeks to find $q(z)$ as an approximation of the posterior $p(z|x)$, which has the benefit of being a strict bound to the true posterior.

Typically, while the functional form of $p(x|z)$ is known, calculating $p(x) = \int p(x|z)p(z)dz$ is intractable. Using Jensen's inequality, we can show that:

$$\log p(x) \geq \mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)]. \tag{3}$$

Furthermore, the reverse Kullback-Leibler (KL) divergence between the posterior and the variational distribution, $\mathcal{D}_{\text{KL}}(q|p)$, can be written as:

$$\log p(x) = \mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)] + \mathcal{D}_{\text{KL}}(q|p). \tag{4}$$

Hence, maximising the evidence lower bound is equivalent to minimising the reverse KL divergence between p and q.

2.1.2. MaxEnt Is Equivalent to Constrained Variational Inference

Minimising the reverse KL divergence between our posterior $q(x)$ and our prior $q_0(x)$ as is done in variational inference, with respect to $q(x)$:

$$\mathcal{D}_{\text{KL}}(q|q_0) = -H(q) - \int_{x \in \mathcal{D}} q(x) \log q_0(x) dx, \tag{5}$$

where $H(q)$ denotes the differential entropy of the density $q(x)$, such that $\int q(x)dx = 1$ and $\int q(x)x^i dx = \mu_i$. By the theory of Lagrangian duality, the convexity of the KL divergence, and the affine nature of the moment constraints, we maximise the dual form [19] of Equation (5):

$$-H(q) - \int q(x) \log q_0(x) dx - \sum_{i=0}^{m} \alpha_i \left(\int q(x) x^i dx - \mu_i \right) \tag{6}$$

with respect to $q(x)$ or, alternatively, we minimise

$$H(q) + \int q(x) \log q_0(x) dx + \sum_{i=0}^{m} \alpha_i \left(\int q(x) x^i dx - \mu_i \right). \tag{7}$$

In the field of information physics, the minimisation of Equation (7) is known as the method of relative entropy [18]. It can be derived as the unique functional satisfying the axioms of *locality*, *coordinate invariance*, *sub-system invariance* and *objectivity*.

The restriction to a functional is derived from considering the set of all distributions $\mathcal{Q} = \{q_j(x)\}$ compatible with the constraints and devising a transitive ranking scheme (Transitive ranking means if $A > B$ and $B > C$, then $A > C$). Furthermore, it can be shown that Newton's laws, non-relativistic quantum mechanics, and Bayes' rule can all be derived under this formalism [18]. In the case of a flat prior, we reduce to the method of maximum entropy with moment constraints [13] as $q_0(x)$ is a constant. Hence, MaxEnt and constrained variational inference are equivalent.

2.2. *Fast Moment Estimation Techniques*

As shown in Section 2.1, constrained variational inference requires the knowledge of the moments of the density being approximated to make inference. In this section, we discuss fast moment estimation techniques for both the general case and the case for which analytic expressions exist.

2.2.1. Moments of a Gaussian Mixture

In some problems, there exist analytic expressions which can be used to compute the moments, One popular example is the Gaussian distribution (and the mixture of Gaussians). Specifically, for the one-dimensional Gaussian, we can find an analytic expression for the moments:

$$\int_{-\infty}^{\infty} x^m \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \sum_{i=0}^{m} (\sqrt{2}\sigma)^{i+1} \binom{m}{i} \mu^{m-i} \zeta_i, \tag{8}$$

where μ and σ are the mean and standard deviation of the Gaussian, respectively, and:

$$\zeta_i = \int_{-\infty}^{\infty} y^i \mathrm{e}^{-y^2} dy = \begin{cases} \frac{(i-1)!!\sqrt{\pi}}{2^{i/2}}, & i \text{ even}, \\ [1/2(i-1)]!, & i \text{ odd}. \end{cases} \tag{9}$$

where $(i-1)!!$ denotes double factorial. Hence, for a mixture of k Gaussians with mean μ_k and standard deviation σ_k, the n-th moment is analytic:

$$\sum_k w_k \sum_{i=0}^{m} (\sqrt{2}\sigma_k)^{i+1} \binom{m}{i} \mu_k^{m-i} \zeta_i. \tag{10}$$

where w_k is the weight of k-th Gaussian in the mixture.

2.2.2. Moments of the Eigenvalue Spectrum of a Matrix

Stochastic trace estimation is an effective way of cheaply estimating the moments of the eigenvalue spectrum of a given matrix $K \in \mathbb{R}^{n \times n}$ [20]. The essential idea is that we can estimate the non-central moments of the spectrum by using matrix-vector multiplications.

Recall that for any multivariate random variable $\mathbf{z}$ with mean $\mathbf{m}$ and variance Σ, we have:

$$\mathbb{E}(\mathbf{z}^T\mathbf{z}) = \mathbf{m}^T\mathbf{m} + \Sigma \overset{\mathbf{m}=0,\Sigma=I}{=\!=\!=\!=} I, \tag{11}$$

where for the second equality we have assumed (without loss of generality) that $\mathbf{z}$ possesses zero mean and unit variance. By the linearity of trace and expectation, for any number of moments $m \geq 0$ we have:

$$\sum_{s=1}^{n} \lambda_s^m = \mathrm{Tr}(IK^m) = \mathbb{E}(\mathbf{z}K^m\mathbf{z}^T), \tag{12}$$

where $\{\lambda_s\}$ are the eigenvalues of K. In practice, we approximate the expectation in Equation (12) with a set of probing vectors and a simple Monte Carlo average, i.e., for d random unit vectors $\{\mathbf{z}_j\}_{j=1}^d$:

$$\mathbb{E}(\mathbf{z}K^m\mathbf{z}^T) \approx \frac{1}{d}\left(\sum_{j=1}^{d} \mathbf{z}_j K^m \mathbf{z}_j^T\right), \tag{13}$$

where we take the product of the matrix K with the vector $\mathbf{z}_j^T$ a total of m times and update $\mathbf{z}_j^T$ each time, so as to avoid the costly matrix multiplication with $\mathcal{O}(n^3)$ complexity. This allows us to calculate the non-central moment expectations in $\mathcal{O}(dmn^2)$ complexity for dense matrices, or $\mathcal{O}(dm \times n_{nz})$ complexity for sparse matrices, where $d \times m \ll n$ and n_{nz} is the number of non-zero entries in the matrix. The random vector $\mathbf{z}_j$ can be drawn from any distribution, such as a Gaussian. The resulting stochastic trace estimation algorithm is outlined in Algorithm 1.

Algorithm 1 Stochastic Trace Estimation [20]

```
Input: Matrix $K \in \mathbb{R}^{n \times n}$, Number of Moments $m$, Number of Probe Vectors $d$
Output: Moment Expectations $\hat{\mu}_i, \forall i \leq m$
Initialise $\hat{\mu}_i = 0 \; \forall i$
for $j = 1, \ldots, d$ do
    Initialise $\mathbf{z}_j = \text{rand}(1, n)$
    for $i = 1, \ldots, m$ do
        $\mathbf{z}_j^T = K\mathbf{z}_j^T$
        $\hat{\mu}_i = \hat{\mu}_i + \frac{1}{d}\mathbf{z}_j\mathbf{z}_j^T$
    end for
end for
```

2.3. The Proposed MaxEnt Algorithm

In this section, we develop an algorithm for determining the maximum relative entropy density given moment constraints. Under the assumption of a flat prior in a bounded domain, this reduces to a generic MaxEnt density, which we use in various examples.

As we discussed previously, in order to obtain the MaxEnt distribution, we maximise the generic objective of Equation (1). In practice, we instead minimise the dual form of this objective [19], which can be written as follows:

$$\mathcal{S}(q, q_0) = \int_{x \in \mathcal{D}} q_0(x) \exp(-[1 + \sum_{i=0}^{m} \alpha_i x^i])dx + \sum_{i=0}^{m} \alpha_i \mu_i. \tag{14}$$

Notice that this dual objective admits analytic expressions for both the gradient and the Hessian:

$$\frac{\partial \mathcal{S}(q, q_0)}{\partial \alpha_j} = \mu_j - \int_{x \in \mathcal{D}} q_0(x) x^j \exp(-[1 + \sum_{i=0}^{m} \alpha_i x^i])dx, \tag{15}$$

$$\frac{\partial^2 \mathcal{S}(q, q_0)}{\partial \alpha_j \partial \alpha_k} = \int_{x \in \mathcal{D}} q_0(x) x^{j+k} \exp(-[1 + \sum_{i=0}^{m} \alpha_i x^i])dx. \tag{16}$$

For a flat prior in a bounded domain, which we use as a prior for the spectral densities of large matrices and for Gaussian mixture models, $q_0(x)$ is a constant that can be dropped out. With the notation

$$q_\alpha(x) = \exp(-[1 + \sum_{i=0}^{m} \alpha_i x^i]), \tag{17}$$

we obtain the MaxEnt algorithm in Algorithm 2.

The input $\{\mu_i\}$ of Algorithm 2 are estimated using fast moment estimation techniques, as explained in Section 2.2.2. In our implementation, we use Python's SciPy Newton-conjugate gradient (CG) algorithm to solve the minimisation in step 4 (line 6), having firstly computed the gradient within a tolerance level of ϵ as well as the Hessian. To make the Hessian better conditioned so as to achieve convergence, we symmetrise it and add jitter of intensity $\eta = 10^{-8}$ along the diagonal. We estimate the given integrals with quadrature using a grid size of 10^{-4}. Empirically, we observe that the algorithm is not overly sensitive to these choices. In particular, we find that, for well conditioned problems, the need for jitter is obviated and a smaller grid size works fine; in the case of worse conditioning, jitter helps improve convergence and the grid size becomes more important, where a smaller grid size leads to a computationally more intensive implementation.

Algorithm 2 The Proposed MaxEnt Algorithm

```
Input: Moments $\{\mu_i\}$, Tolerance Level $\epsilon$, Jitter variance in Hessian $\eta = 10^{-8}$
Output: Coefficients $\{\alpha_i\}$
Initialise $\alpha_i = 0$
Compute gradient: $g_j = \mu_j - \int_0^1 q_\alpha(x)x^j dx$
Compute Hessian: $H_{jk} = \int_0^1 q_\alpha(x)x^{j+k}dx,\ H = \frac{1}{2}(H + H^T) + \eta I$
Minimize $\int_0^1 q_\alpha(x)dx + \sum_i \alpha_i \mu_i$ using Conjugate Gradients until $\forall j$: $g_j < \epsilon$
```

Given that any polynomial sum can be written as $\sum_i \alpha_i x^i = \sum_i \beta_i f_i(x)$, where $f_i(x)$ denotes another polynomial basis, whether we choose to use power moments in our constraints or another polynomial basis, such as the Chebyshev or Legendre basis, does not change the entropy or solution of our objective. However, the performance of optimisers working on these different formulations may vary [21]. For simplicity, we have kept all the formulas in terms of power moments. However, we find vastly improved performance and conditioning when we switch to orthogonal polynomial bases (so that the errors between moment estimations are uncorrelated), as shown in Section 3.2.2. We implement both Chebyshev and Legendre moments in our Lagrangian and find similar performance for both.

3. Applications

We apply the proposed algorithm to two problems of interest, namely, log determinant estimation and Bayesian optimisation. In both cases we demonstrate substantial speed up and improvement in performance.

3.1. Log Determinant Estimation

Calculation of the log determinant is a common hindrance in machine learning, appearing in Gaussian graphical models [22], Gaussian processes [23], variational inference [24], metric and kernel learning [25], Markov random fields [26] and determinantal point processes [11]. For a large positive definite matrix $K \in \mathbb{R}^{n \times n}$, the canonical solution involves the Cholesky decomposition of $K = LL^T$. The log determinant is then trivial to calculate as $\log\det(K) = 2\sum_{i=1}^n \log L_{ii}$, where L_{ii} is the ii-th entry of L. This computation invokes a computational complexity of $\mathcal{O}(n^3)$ and storage complexity of $\mathcal{O}(n^2)$, which becomes prohibitive for large n, i.e., $n > 10^4$.

3.1.1. Log Determinant Estimation as a Spectral Estimation Problem Using MaxEnt

Any symmetric positive definite (PD) matrix K is diagonalisable by a unitary matrix U, i.e., $K = U^T D U$, where D is the diagonal matrix of eigenvalues of K. Hence we can write the log determinant as:

$$\log\det(K) = \log\prod_i \lambda_i = \sum_{i=1}^n \log\lambda_i = n\mathbb{E}_p(\log\lambda), \tag{18}$$

where we have used the cyclicity of the determinant and $\mathbb{E}_p(\log\lambda)$ denotes the expectation under the spectral measure. The latter can be written as:

$$\mathbb{E}_p(\log\lambda) = \int_{\lambda_{\min}}^{\lambda_{\max}} p(\lambda)\log\lambda d\lambda = \int_{\lambda_{\min}}^{\lambda_{\max}} \sum_{i=1}^n \frac{1}{n}\delta(\lambda - \lambda_i)\log\lambda d\lambda, \tag{19}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ correspond to the largest and smallest eigenvalues, respectively.

Given that the matrix is PD, we know that $\lambda_{\min} > 0$ and we can divide the matrix by an upper bound of the eigenvalue, $\lambda_{\max} \leq \lambda_u$, via the Gershgorin circle theorem [27] such that:

$$\log\det(K) = n\mathbb{E}_p(\log\lambda') + n\log\lambda_u, \tag{20}$$

where $\lambda' = \lambda/\lambda_u$ and $\lambda_u = \max_i(\sum_{j=1}^{n}|K_{ij}|)$, i.e., the maximum sum of the rows of the absolute of the matrix K. Hence we can comfortably work with the transformed measure:

$$\int_{\lambda_{\min}/\lambda_u}^{\lambda_{\max}/\lambda_u} p(\lambda')\log\lambda' d\lambda' = \int_0^1 p(\lambda')\log\lambda' d\lambda', \tag{21}$$

where the spectral density $p(\lambda')$ is 0 outside of its bounds of $[0,1]$. We therefore arrive at the following maximum entropy method (MEMe) for log determinant estimation detailed in Algorithm 3.

Algorithm 3 MEMe for Log Determinant Estimation

```
Input: PD Symmetric Matrix $K \in \mathbb{R}^{n\times n}$, Number of Moments $m$, Number of Probe Vectors $d$, Tolerance Level $\epsilon$
Output: Log Determinant Approximation $\log\det(K)$
$\lambda_u = \max_i(\sum_{j=1}^{n}|K_{ij}|)$
$B = K/\lambda_u$
Compute moments, $\{\mu_i\}$, via Stochastic Trace Estimation (Algorithm 1) with inputs $(B, m, d)$
Compute coefficients, $\{\alpha_i\}$, via MaxEnt (Algorithm 2) with inputs $(\{\mu_i\}, \epsilon)$
Compute $q(\lambda) = \exp\left[-(1+\sum_i \alpha_i\lambda^i)\right]$
Estimate $\log\det(K) \approx n\int\log(\lambda)q(\lambda)d\lambda + n\log(\lambda_u)$
```

3.1.2. Experiments

Log Determinant Estimation for Synthetic Kernel Matrix

To specifically compare against commonly used Chebyshev [4] and Lanczos approximations [28] to the log determinant, and see how their accuracy degrades with worsening condition number, we generate a typical squared exponential kernel matrix [23], $K \in \mathbb{R}^{n\times n}$, using the Python GPy package with 6 dimensional Gaussian inputs with a variety of realistic uniform length-scales. We then add noise of variance $\eta = 10^{-8}$ along the diagonal of K.

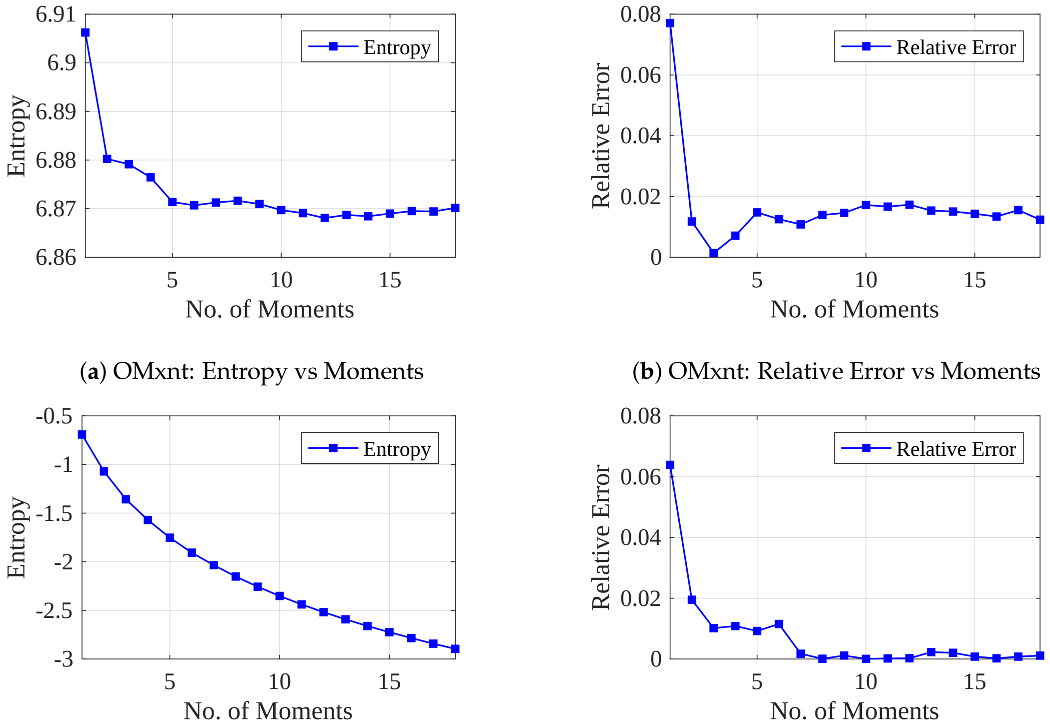
We use $m = 30$ moments and $d = 50$ Hutchinson probe vectors [20] in MEMe for the log determinant estimation, and display the absolute relative estimation error for different approaches in Table 1. We see that, for low condition numbers, the benefit of framing the log determinant as an optimisation problem is marginal, whereas for large condition numbers, the benefit becomes substantial, with orders of magnitude better results than competing methods.

Table 1. Relative estimation error for MEMe, Chebyshev, and Lanczos approaches, with various length-scale l and condition number κ on the squared exponential kernel matrix $K \in \mathbb{R}^{1000\times1000}$.

κ	l	**MEMe**	**Chebyshev**	**Lanczos**
3×10^{1}	0.05	**0.0014**	0.0037	0.0024
1.1×10^{3}	0.15	0.0522	**0.0104**	0.0024
1.0×10^{5}	0.25	0.0387	0.0795	**0.0072**
2.4×10^{6}	0.35	0.0263	0.2302	**0.0196**
8.3×10^{7}	0.45	**0.0284**	0.3439	0.0502
4.2×10^{8}	0.55	**0.0256**	0.4089	0.0646
4.3×10^{9}	0.65	**0.00048**	0.5049	0.0838
1.4×10^{10}	0.75	**0.0086**	0.5049	0.1050
4.2×10^{10}	0.85	**0.0177**	0.5358	0.1199

Log Determinant Estimation on Real Data

The work in Granziol and Roberts [29] has shown that the addition of an extra moment constraint cannot increase the entropy of the MaxEnt solution. For the problem of log determinant, this signifies that the entropy of the spectral approximation should decrease with the addition of every moment constraint. We implement the MaxEnt algorithm proposed in Bandyopadhyay et al. [2], which we refer to as OMxnt, in the same manner as applied for log determinant estimation in Fitzsimons et al. [30], and compare it against the proposed MEMe approach. Specifically, we show results on the Ecology dataset [31], with $n = 999,999$, for which the true log determinant can be calculated. For OMxnt, we see that after the initial decrease, the error (Figure 1b) begins to increase for $m > 3$ moments and the entropy (Figure 1a) increases at $m = 6$ and $m = 12$ moments. For the proposed MEMe algorithm, the performance is vastly improved in terms of estimation error (Figure 1d); furthermore, the error continually decreases with increasing number of moments, and the entropy (Figure 1c) decreases smoothly. This demonstrates the superiority both in terms of consistency and performance of our novel algorithm over established existing alternatives.



(**a**) OMxnt: Entropy vs Moments

(**b**) OMxnt: Relative Error vs Moments

(**c**) MEMe: Entropy vs Moments

(**d**) MEMe: Relative Error vs Moments

Figure 1. Comparison of the Classical (OMxnt) and our novel (MEMe) MaxEnt algorithms in log determinant estimation on real data. The entropy value (**a**) and estimation error (**b**) of OMxnt are shown in the top row. Those of the MEMe are shown in (**c**,**d**) in the bottom row.

3.2. *Bayesian Optimisation*

Bayesian Optimisation (BO) is a powerful tool to tackle global optimisation challenges. It is particularly useful when the objective function is unknown, non-convex and very expensive to evaluate [32], and has been successfully applied in a wide range of applications including automatic machine learning [33–36], robotics [37,38] and experimental design [39]. When applying BO, we need to choose a statistical prior to model the objective function and define an acquisition function which trades off exploration and exploitation to recommend the next query location [40]. The generic BO algorithm is presented in Algorithm A1 in Appendix B. In the literature, one of the most popular prior models is the Gaussian processes (GPs), and the most recent class of acquisition functions that demonstrate state-of-the-art empirical performance is the information-theoretic ones [7–9,41].

3.2.1. MaxEnt for Information-Theoretic Bayesian Optimisation

Information-theoretic BO methods select the next query point that maximises information gain about the unknown global optimiser/optimum. Despite their impressive performance, the computation of their acquisition functions involves an intractable term of Gaussian mixture entropy, as shown in Equation (22), if we perform a fully Bayesian treatment for the GP hyperparameters. Accurate approximation of this Gaussian mixture entropy term is crucial for the performance of information-theoretic BO, but can be difficult and/or expensive. In this paper, we propose an efficient approximation of the Gaussian mixture entropy by using MEMe, which allows for efficient computation of the information-theoretic acquisition functions.

As an concrete example, we consider the application of MEMe for Fast Information-Theoretic Bayesian Optimisation (FITBO) [9] as described in Algorithm 4, which is a highly practical information-based BO method proposed recently. The FITBO acquisition function has the following form:

$$\alpha(\mathbf{x}|\mathcal{D}_t) = H\Big[\frac{1}{M}\sum_{j}^{M} p(y|\mathcal{I}^{(j)})\Big] - \frac{1}{M}\sum_{j}^{M} H\big[p(y|\mathcal{I}^{(j)})\big], \tag{22}$$

where $p(y|\mathcal{I}^{(j)}) = p(y|\mathcal{D}_t, \mathbf{x}, \boldsymbol{\theta}^{(j)})$ is the predictive posterior distribution for y conditioned on the observed data $\mathcal{D}_t$, the test location $\mathbf{x}$, and a hyperparameter sample $\boldsymbol{\theta}^{(j)}$. The first entropy term is the entropy of a Gaussian mixture, where M is the number of Gaussian components.

The entropy of a Gaussian mixture does not have a closed-form solution and needs to be approximated. In FITBO, the authors approximate the quantity using numerical quadrature and moment matching (This corresponds to the maximum entropy solution for two moment constraints as well as the normalization constraint.). In this paper, we develop an effective analytic upper bound of the Gaussian mixture entropy using MEMe, which is derived from the non-negative relative entropy between the true density $p(x)$ and the MaxEnt distribution $q(x)$ [29]:

$$\mathcal{D}_{\text{KL}}(p||q) = -H(p) + H(q) \geq 0, \tag{23}$$

hence

$$H(p) \leq H(q). \tag{24}$$

Notice that $q(x)$ shares the same moment constraints as $p(x)$; furthermore, the entropy of the MaxEnt distribution $q(x)$ can be derived analytically:

$$H(q) = 1 + \sum_{i=0}^{m} \alpha_i \mu_i, \tag{25}$$

where $\{\mu_i\}$ are the moments of a Gaussian mixture, which can be computed analytically using Equation (10), and $\{\alpha_i\}$ are the Lagrange multipliers that can be obtained by applying Algorithm 2. The overall algorithm for approximating the Gaussian mixture entropy is then summarised in Algorithm 5. In Appendix B.1, we also prove that the moments of the Gaussian mixture uniquely determine its density, and the bound becomes tighter with every extra moment constraint: in the $m \to \infty$ limit, the entropy of the MaxEnt distribution converges to the true Gaussian mixture entropy. Hence, the use of a moment-based MaxEnt approximation can be justified [3].

Algorithm 4 MEMe for FITBO

```
Input: Observed data $D_t$
Output: Acquisition function $\alpha_n(\mathbf{x}|D_t)$
Sample $M$ hyperparameters $\left\{\boldsymbol{\theta}^{(j)}\right\}$
for $j = 1, \ldots, M$ do
    Approximately compute $p(y|\mathcal{D}_t, \mathbf{x}, \boldsymbol{\theta}^{(j)}) = \mathcal{N}(y; m_j, K_j)$ and its entropy $H\left[p(y|\mathcal{D}_t, \mathbf{x}, \boldsymbol{\theta}^{(j)})\right]$
end for
Approximate $H[\frac{1}{M}\sum_j^M \mathcal{N}(y; m_j, K_j)]$ with MEMe following Algorithm 5
Compute $\alpha(\mathbf{x}|D_t)$ as in Equation (22)
```

Algorithm 5 MEMe for Approximating Gaussian Mixture Entropy

```
Input: A univariate Gaussian mixture GM $= \frac{1}{M}\sum_j^M \mathcal{N}(y; m_j, \sigma_j^2)$ with mean $m_j$ and variance $\sigma_j^2$
Output: $H_{\mathrm{GM}} \approx H[\frac{1}{M}\sum_j^M \mathcal{N}(y; m_j, \sigma_j^2)]$
Compute the moments of GM, $\{\mu_i\}$, analytically using Equation (10)
Compute the Lagrange multipliers, $\{\alpha_i\}$, using Algorithm 2
$H_{\mathrm{GM}} \approx -(1 + \sum_i \alpha_i \mathbb{E}[y^i]) = -(1 + \sum_i \alpha_i \mu_i)$
```

3.2.2. Experiments

Entropy of the Gaussian Mixture in Bayesian Optimisation

We first test a diverse set of methods to approximate the entropy of two sets of Gaussian mixtures, which are generated using FITBO with 200 hyperparameter samples and 400 hyperparameter samples on a 2D problem. Specifically, we compare the following approaches for entropy approximation: MaxEnt methods using 10 and 30 power moments (MEMe-10 and MEMe-30), MaxEnt methods using 10 and 30 Legendre moments (MEMeL-10 and MEMeL-30), variational upper bound (VUB) [42], method proposed in [43] with 2 Taylor expansion terms (Huber-2), Monte Carlo with 100 samples (MC-100), and simple moment matching (MM).

We evaluate the performance in terms of the approximation error, i.e., the relative error between the approximated entropy value and the true entropy value, the latter of which is estimated via expensive numerical integration. The results of the mean approximation errors by all methods over 80 different Gaussian mixtures are shown in Table 2 (The version with the standard deviation of errors is presented as Table A1 in Appendix B.2). We can see clearly that all MEMe approaches outperform other methods in terms of the approximation error. Among the MEMe approaches, the use of Legendre moments, which apply orthogonal Legendre polynomial bases, outperforms the use of simple power moments.

Table 2. Mean fractional error in approximating the entropy of the mixture of M Gaussians using various methods.

Methods	$M = 200$	$M = 400$
MEMe-10	1.24×10^{-2}	1.38×10^{-2}
MEMe-30	1.13×10^{-2}	1.06×10^{-2}
MEMeL-10	1.01×10^{-2}	0.85×10^{-2}
MEMeL-30	$\mathbf{0.50 \times 10^{-2}}$	$\mathbf{0.36 \times 10^{-2}}$
VUB	22.0×10^{-2}	29.1×10^{-2}
Huber-2	24.7×10^{-2}	35.5×10^{-2}
MC-100	1.60×10^{-2}	2.72×10^{-2}
MM	2.78×10^{-2}	3.22×10^{-2}

The mean runtime taken by all approximation methods over 80 different Gaussian mixtures are shown in Table 3 (The version with the standard deviation is presented as Table A2 in Appendix B.2). To ensure a fair comparison, all the methods are implemented in MATLAB and all the timing tests are performed on a 2.3GHz Intel Core i5 processor. As we expect, the moment matching technique, which enables us to obtain an analytic approximation for the Gaussian mixture entropy, is the fastest method. MEMe approaches are significantly faster than the rest of approximation methods. This demonstrates that MEMe approaches are highly efficient in terms of both approximation accuracy and computational speed. Among all the MEMe approaches, we choose to apply MaxEnt with 10 Legendre moments in the BO for the next set of experiments, as it is able to achieve lower approximation error than MaxEnt with higher power moments while preserving the computational benefit of FITBO.

Table 3. Mean runtime of approximating the entropy of the mixture of M Gaussians using various methods.

Methods	$M = 200$	$M = 400$
MEMe-10	1.38×10^{-2}	1.48×10^{-2}
MEMe-30	2.59×10^{-2}	3.21×10^{-2}
MEMeL-10	1.70×10^{-2}	1.75×10^{-2}
MEMeL-30	4.18×10^{-2}	4.66×10^{-2}
VUB	12.9×10^{-2}	50.7×10^{-2}
Huber-2	20.9×10^{-2}	82.2×10^{-2}
MC-100	10.6×10^{-2}	40.1×10^{-2}
MM	$\mathbf{2.71 \times 10^{-5}}$	$\mathbf{2.87 \times 10^{-5}}$

Information-Theoretic Bayesian Optimisation

We now test the effectiveness of MEMe for information-theoretic BO. We first illustrate the entropy approximation performance of MEMe using a 1D example. In Figure 2, the top plot shows the objective function we want to optimise (red dash line) and the posterior distribution of our surrogate model (blue solid line and shaded area). The bottom plot shows the acquisition functions computed based on Equation (22) using the same surrogate model but three different methods for Gaussian mixture entropy approximation, i.e., expensive numerical quadrature or Quad (red solid line), MM (black dash line), and MEMe using 10 Legendre moments (green dash line). In BO, the next query location is obtained by maximising the acquisition function, therefore the location instead of the magnitude of the modes of the acquisition function matters most. We can see from the bottom plot that MEMeL-10 results in an approximation that well matches the true acquisition function obtained by Quad. As a result, MEMeL-10 manages to recommend the same next query location as Quad. In comparison, the loose upper bound of the MM method, though successfully capturing the locations of the peak values, fails to correctly predict the global maximiser of the true acquisition function. MM therefore recommends a query location that is different from the optimal choice. As previously mentioned, the acquisition function in information-theoretic BO represents the information gain about the global optimum by querying at a new location. The sub-optimal choice of the next query location thus imposes a penalty on the optimisation performance as seen in Figure 3.

In the next set of experiments, we evaluate the optimisation performance of three versions of FITBO that use different approximation methods. Specifically, FITBO-Quad denotes the version that uses expensive numerical quadrature to approximate the entropy of the Gaussian mixture, FITBO-MM denotes the one using simple moment matching, and FITBO-MEMeL denotes the one using MEMe with 10 Legendre moments. We test these BO algorithms on two challenging optimisation problems, i.e., the Michalewicz-5D function [44] and the Hartmann-6D function [45], and measure the optimisation performance in terms of the immediate regret (IR): IR $= |f^* - \hat{f}|$, which measures the absolute difference between the true global minimum value f^* and the best guess of the global minimum value $\hat{f}$ by the BO algorithm. The average (median) result over 10 random initialisations for each experiment is shown in Figure 3. It is evident that the MEMe approach (FITBO-MEMeL-10), which better approximates the Gaussian mixture entropy, leads to a superior performance of the BO algorithm compared to the BO algorithm using simple moment matching technique (FITBO-MM).

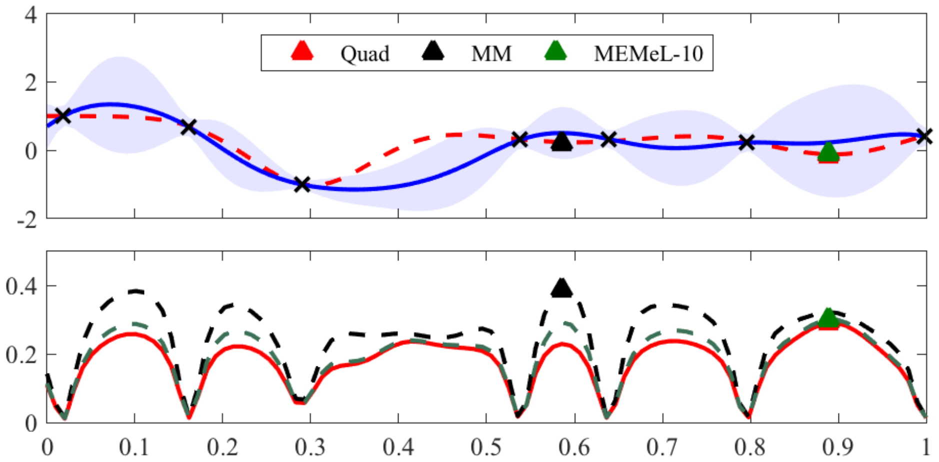
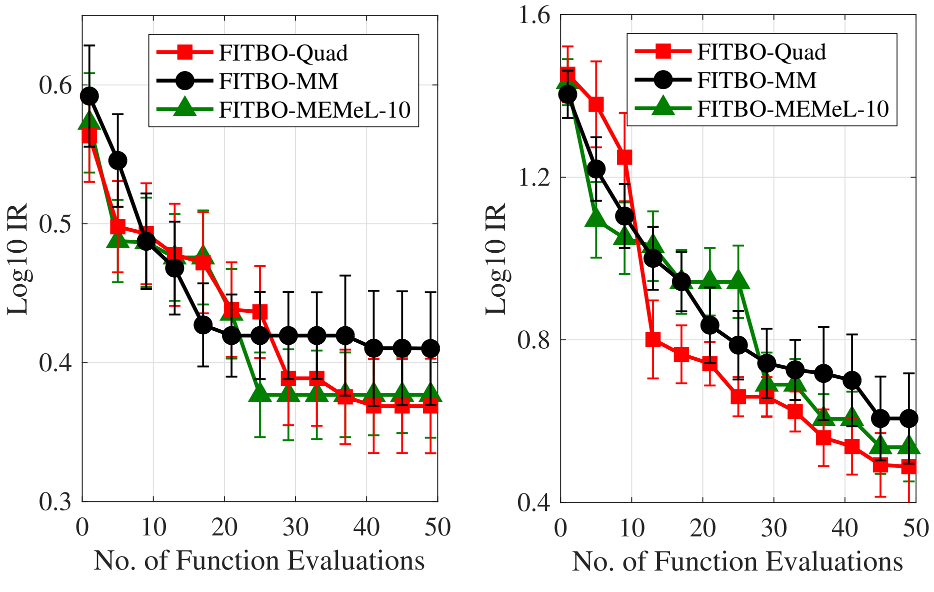


Figure 2. Bayesian Optimisation (BO) on a 1D toy example with acquisition functions computed by different approximation methods. In the top subplot, the red dash line is the unknown objective function, the black crosses are the observed data points, and the blue solid line and shaded area are the posterior mean and variance, respectively, of the GP surrogate that we use to model the latent objective function. The coloured triangles are the next query point recommended by the BO algorithms, which correspond to the maximiser of the acquisition functions in the bottom subplot. In the bottom plot, the red solid line, black dash line, and green dotted line are the acquisition functions computed by Quad, MM, and MEMe using 10 Legendre moments, respectively.



(**a**) Michalewicz-5D (**b**) Hartmann-6D

Figure 3. Performance of various versions of FITBO on 2 benchmark test problems: (**a**) Michalewicz-5D function and (**b**) Hartmann-6D function. The immediate regret (IR) on the y-axis is expressed in the logarithm to the base 10.

4. Conclusions

In this paper, we established the equivalence between the method of maximum entropy and Bayesian variational inference under moment constraints, and proposed a novel maximum entropy algorithm (MEMe) that is stable and consistent for a large number of moments. We apply MEMe in two applications, i.e., log determinant estimation and Bayesian optimisation, to demonstrate its effectiveness and superiority over state-of-the-art approaches. The proposed algorithm can further benefit a wide range of large-scale machine learning applications where efficient approximation is of crucial importance.

Author Contributions: Conceptualization, D.G.; formal analysis, investigation, methodology, software and writing—original draft, D.G. and B.R.; supervision and writing—review and editing, S.Z., X.D., M.O. and S.R.

Funding: B.R. would like to thank the Oxford Clarendon Fund and the Oxford-Man Institute of Quantitative Finance; D.G., S.Z. and X.D. would like to thank the Oxford-Man Institute of Quantitative Finance for financial support; S.R. would like to thank the UK Royal Academy of Engineering and the Oxford-Man Institute of Quantitative Finance.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MEMe	Maximum entropy method
MaxEnt	Maximum entropy
PD	Positive definite
CG	Conjugate gradient
OMxnt	Old MaxEnt algorithm proposed by Bandyopadhyay et al. [2]
BO	Bayesian optimisation
GP	Gaussian process
FITBO	Fast information-theoretic Bayesian optimisation
GM	Gaussian mixture
VUB	Variational upper bound
Huber	Method proposed by Huber et al. [43] for estimating the Gaussian mixture entropy
MC	Monte Carlo sampling
MM	Moment matching
Quad	Numerical quadrature
IR	Immediate regret

Appendix A. Polynomial Approximations to the Log Determinant

Recent work [4–6] has considered incorporating knowledge of the non-central moments (Also using stochastic trace estimation.) of a normalised eigenspectrum by replacing the logarithm with a finite polynomial expansion,

$$\int_0^1 p(\lambda)\log(\lambda)d\lambda = \int_0^1 p(\lambda)\log(1-(1-\lambda))d\lambda. \tag{A1}$$

Given that $\log(\lambda)$ is not analytic at $\lambda = 0$, it can be seen that, for any density with a large mass near the origin, a very large number of polynomial expansions, and thus moment estimates, will be required to achieve a good approximation, irrespective of the choice of basis.

Appendix A.1. Taylor Approximations Are Unsound

In the case of a Taylor expansion, Equation (A1) can be written as,

$$-\int_0^1 p(\lambda)\sum_{i=1}^{\infty}\frac{(1-\lambda)^i}{i} \approx -\int_0^1 p(\lambda)\sum_{i=1}^{m}\frac{(1-\lambda)^i}{i}. \tag{A2}$$

The error in this approximation can be written as the difference of the two sums,

$$-\sum_{i=m+1}^{\infty}\frac{\mathbb{E}_p(1-\lambda)^i}{i}, \tag{A3}$$

where we have used the Taylor expansion of $\log(1-x)$ and $\mathbb{E}_p$ denotes the expectation under the spectral measure. We begin with complete ignorance about the spectral density $p(\lambda)$ (other than its domain $[0,1]$) and by some scheme after seeing the first m non-central moment estimates we propose a surrogate density $q(\lambda)$. The error in our approximation can be written as,

$$\begin{aligned}&\int_0^1 [p(\lambda)-q(\lambda)]\log(\lambda)d\lambda \\ =&\int_0^1 -[p(\lambda)-q(\lambda)]\sum_{i=1}^{\infty}\frac{(1-\lambda)^i}{i}d\lambda.\end{aligned} \tag{A4}$$

For this error to be equal to that of our Taylor expansion, Equation (A3), our implicit surrogate density must have the first m non-central moments of $(1-\lambda)$ identical to the true spectral density $p(\lambda)$ and all others 0.

For any PD matrix K, for which $E_p(1-\lambda)^i > 0$, $\forall i \leq m$, (We ignore the trivial case of a Dirac distribution at $\lambda = 1$, which is of no practical interest.) for Equation (A4) to be equal to Equation (A3), we must have,

$$\int_0^1 q(\lambda)\sum_{i=m+1}^{\infty}\frac{(1-\lambda)^i}{i}d\lambda = 0. \tag{A5}$$

Given that $0 \leq \lambda \leq 1$ and that we have removed the trivial case of the spectral density (and by implication its surrogate) being a delta function at $\lambda = 1$, the sum is manifestly positive and hence $q(\lambda) < 0$ for some λ, which violates the definition of a density.

Appendix B. Bayesian Optimisation

The generic algorithm for Bayesian optimisation can be summarised in Algorithm A1.

Algorithm A1 Bayesian Optimisation

```
Input: A black-box function $y$, Initial observed data $D_0$
Output: The best guess about the global optimiser $\hat{\mathbf{x}}_N$
for $n = 0, \dots, N$ do
    Select $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha_n(\mathbf{x}|D_t)$
    Query $y$ at $\mathbf{x}_{n+1}$ to obtain $y_{n+1}$
    $D_{n+1} \leftarrow D_t \cup (\mathbf{x}_{n+1}, f_{n+1})$
    Update model $p(y|\mathbf{x}, D_{n+1}) = \mathcal{GP}(y; m(\cdot), K(\cdot,\cdot))$
end for
$\hat{\mathbf{x}}_N = \arg\max_{\mathbf{x}\in\mathcal{X}} m_N(\mathbf{x})$
```

Appendix B.1. Are Moments Sufficient to Fully Describe the Problem?

It was shown in Billingsley [46] that, for a probability measure μ having finite moments of all orders $\alpha_k = \int_{-\infty}^{\infty} x^k \mu(dx)$, if the power series $\sum_k \alpha_k/k!$ has a positive radius of convergence, then μ is the only probability measure with the moments $\{\alpha_i\}$. Informally, a Gaussian has finite moments of all orders; therefore, any finite combination of Gaussians must necessarily possess finite moments of all orders hence the above condition is satisfied. We explicitly show this for the case of a one-dimensional Gaussian as follows.

We take the location parameter as m_i and standard deviation as σ_i. It can be seen that the $2k$-th moment of the Gaussian is given as:

$$G_{2k} = \sum_{2p=0}^{2k} \binom{2k}{2p} m_i^{2(k-p)} \beta_i^{-p}, \tag{A6}$$

where we make use of the fact that all odd central power moments of the Gaussian are 0. Hence for the Gaussian mixture model we have:

$$GM_{2k} = \sum_{i=1}^{N} w_i \sum_{2p=0}^{2k} \binom{2k}{2p} m_i^{2(k-p)} \beta_i^{-p}, \tag{A7}$$

where $\{w_i\}$ are the weights for the components that satisfy $\sum_i w_i = 1$ and $0 \leq w_i \leq 1$. Notice that μ_i is upper bounded by a quantity greater than 1 and β_i is lower bounded by a quantity smaller than 1.

Furthermore, the following relationship holds: $\sum_{2p=0}^{2k} \binom{2k}{2p} < (k+1)\frac{(2k)!}{(k!)^2}$. Therefore, the expression in Equation (A7) can be upper bounded as:

$$GM_{2k} < N(k+1)\frac{(2k)!}{(k!)^2(\mu_{max}^{2k} \beta_i^{-k})}, \tag{A8}$$

which is smaller than $(2k)!$ in the $k \to \infty$ limit by taking the logarithm:

$$\frac{\log N}{2k} + \frac{\log(k+1)}{2k} + \log \mu_{max} + \frac{|\log \beta_i|}{2} \leq \log k. \tag{A9}$$

Appendix B.2. Experimental Results on Approximating the Gaussian Mixture Entropy

The mean and standard deviation of the approximation error and the runtime taken by all approximation methods over 80 different Gaussian mixtures are shown in Table A1 and Table A2 respectively.

Table A1. Fractional error in approximating the entropy of Gaussian mixtures using various methods.

Methods	***M* = 200**	***M* = 400**
MEMe-10	1.24×10^{-2} $(\pm 4.12 \times 10^{-2})$	1.38×10^{-2} $(\pm 5.46 \times 10^{-2})$
MEMe-30	$\mathbf{1.13 \times 10^{-2}}$ $(\pm 3.68 \times 10^{-2})$	$\mathbf{1.06 \times 10^{-2}}$ $(\pm 4.47 \times 10^{-2})$
MEMeL-10	$\mathbf{1.01 \times 10^{-2}}$ $(\pm 3.68 \times 10^{-2})$	$\mathbf{0.85 \times 10^{-2}}$ $(\pm 3.81 \times 10^{-2})$
MEMeL-30	$\mathbf{0.50 \times 10^{-2}}$ $(\pm 2.05 \times 10^{-2})$	$\mathbf{0.36 \times 10^{-2}}$ $(\pm 1.62 \times 10^{-2})$
Variational Upper Bound	22.0×10^{-2} $(\pm 28.0 \times 10^{-2})$	29.1×10^{-2} $(\pm 78.6 \times 10^{-2})$
Huber-2	24.7×10^{-2} $(\pm 46.1 \times 10^{-2})$	35.5×10^{-2} $(\pm 130.4 \times 10^{-2})$
MC-100	1.60×10^{-2} $(\pm 3.80 \times 10^{-2})$	2.72×10^{-2} $(\pm 11.9 \times 10^{-2})$
Moment Matching	2.78×10^{-2} $(\pm 4.85 \times 10^{-2})$	3.22×10^{-2} $(\pm 7.94 \times 10^{-2})$

Table A2. Runtime of approximating the entropy of Gaussian mixtures using various methods.

Methods	M = 200	M = 400
MEMe-10	$\mathbf{1.38 \times 10^{-2}}$	$\mathbf{1.48 \times 10^{-2}}$
	$(\pm 2.59 \times 10^{-3})$	$(\pm 2.22 \times 10^{-3})$
MEMe-30	2.59×10^{-2}	3.21×10^{-2}
	$(\pm 4.68 \times 10^{-3})$	$(\pm 5.17 \times 10^{-3})$
MEMeL-10	$\mathbf{1.70 \times 10^{-2}}$	$\mathbf{1.75 \times 10^{-2}}$
	$(\pm 3.00 \times 10^{-3})$	$(\pm 3.17 \times 10^{-3})$
MEMeL-30	4.18×10^{-2}	4.66×10^{-2}
	$(\pm 5.63 \times 10^{-3})$	$(\pm 5.00 \times 10^{-3})$
Variational	12.9×10^{-2}	50.7×10^{-2}
Upper Bound	$(\pm 8.22 \times 10^{-3})$	$(\pm 10.4 \times 10^{-3})$
Huber-2	20.9×10^{-2}	82.2×10^{-2}
	$(\pm 7.54 \times 10^{-3})$	$(\pm 14.8 \times 10^{-3})$
MC-100	10.6×10^{-2}	40.1×10^{-2}
	$(\pm 6.19 \times 10^{-3})$	$(\pm 17.8 \times 10^{-3})$
Moment	$\mathbf{2.71 \times 10^{-5}}$	$\mathbf{2.87 \times 10^{-5}}$
Matching	$(\pm 1.16 \times 10^{-4})$	$(\pm 1.19 \times 10^{-4})$

References

1. Granziol, D.; Roberts, S.J. Entropic determinants of massive matrices. In Proceedings of the 2017 IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017; pp. 88–93.
2. Bandyopadhyay, K.; Bhattacharya, A.K.; Biswas, P.; Drabold, D. Maximum entropy and the problem of moments: A stable algorithm. *Phys. Rev. E* **2005**, *71*, 057701. [CrossRef] [PubMed]
3. Mead, L.R.; Papanicolaou, N. Maximum entropy in the problem of moments. *J. Math. Phys.* **1984**, *25*, 2404–2417. [CrossRef]
4. Han, I.; Malioutov, D.; Shin, J. Large-scale log-determinant computation through stochastic Chebyshev expansions. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 908–917.
5. Dong, K.; Eriksson, D.; Nickisch, H.; Bindel, D.; Wilson, A.G. Scalable Log Determinants for Gaussian Process Kernel Learning. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6330–6340.
6. Zhang, Y.; Leithead, W.E. Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *J. Stat. Comput. Simul.* **2007**, *77*, 329–348. [CrossRef]
7. Hernández-Lobato, J.M.; Hoffman, M.W.; Ghahramani, Z. Predictive entropy search for efficient global optimization of black-box functions. In Proceedings of the 27st Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 918–926.
8. Wang, Z.; Jegelka, S. Max-value Entropy Search for Efficient Bayesian Optimization. *arXiv* **2017**, arXiv:1703.01968.
9. Ru, B.; McLeod, M.; Granziol, D.; Osborne, M.A. Fast Information-theoretic Bayesian Optimisation. In Proceedings of the 2018 International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4381–4389.
10. Fox, C.W.; Roberts, S.J. A tutorial on variational Bayesian inference. *Artif. Intell. Rev.* **2012**, *38*, 85–95. [CrossRef]
11. Kulesza, A. Determinantal Point Processes for Machine Learning. *Found. Trends Mach. Learn.* **2012**, *5*, 123–286. [CrossRef]
12. Pressé, S.; Ghosh, K.; Lee, J.; Dill, K.A. Principles of Maximum Entropy and Maximum Caliber in Statistical Physics. *Rev. Mod. Phys.* **2013**, *85*, 1115–1141. [CrossRef]
13. Jaynes, E.T. Information Theory and Statistical Mechanics. *Phys. Rev.* **1957**, *106*, 620–630. [CrossRef]
14. Giffin, A.; Cafaro, C.; Ali, S.A. Application of the Maximum Relative Entropy method to the Physics of Ferromagnetic Materials. *Phys. A Stat. Mech. Appl.* **2016**, *455*, 11 – 26. [CrossRef]
15. Neri, C.; Schneider, L. Maximum Entropy Distributions inferred from Option Portfolios on an Asset. *Financ. Stoch.* **2012**, *16*, 293–318. [CrossRef]

16. Bretthorst, G.L. The maximum entropy method of moments and Bayesian probability theory. *AIP Conf. Proc.* **2013**, *1553*, 3–15.
17. Beal, M.J. Variational Algorithms for Approximate Bayesian Inference. Master's Thesis, University of London, London, UK, 2003.
18. Caticha, A. Entropic Inference and the Foundations of Physics (Monograph Commissioned by the 11th Brazilian Meeting on Bayesian Statistics-EBEB-2012. 2012. Available online: https://www.albany.edu/physics/ACaticha-EIFP-book.pdf (accessed on 31 May 2019).
19. Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2009.
20. Hutchinson, M.F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Commun. Stat. Simul. Comput.* **1990**, *19*, 433–450. [CrossRef]
21. Skilling, J. The eigenvalues of mega-dimensional matrices. In *Maximum Entropy and Bayesian Methods*; Springer: Berlin, Germany, 1989; pp. 455–466.
22. Friedman, J.; Hastie, T.; Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **2008**, *9*, 432–441. [CrossRef] [PubMed]
23. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, USA, 2006; pp. 715–719.
24. MacKay, D.J. *Information Theory, Inference and Learning Algorithms*; Cambridge University Press: Cambridge, UK, 2003.
25. Van Aelst, S.; Rousseeuw, P. Minimum volume ellipsoid. *Wiley Interdiscip. Rev. Comput. Stat.* **2009**, *1*, 71–82. [CrossRef]
26. Wainwright, M.J.; Jordan, M.I. Log-determinant relaxation for approximate inference in discrete Markov random fields. *IEEE Trans. Signal Process.* **2006**, *54*, 2099–2109. [CrossRef]
27. Gershgorin, S.A. Über die Abgrenzung der Eigenwerte einer Matrix. *Izvestija Akademii Nauk SSSR Serija Matematika* **1931**, *6*, 749–754.
28. Ubaru, S.; Chen, J.; Saad, Y. Fast Estimation of $tr(f(A))$ via Stochastic Lanczos Quadrature. *SIAM J. Matrix Anal. Appl.* **2016**, *38*, 1075–1099. [CrossRef]
29. Granziol, D.; Roberts, S. An Information and Field Theoretic approach to the Grand Canonical Ensemble. *arXiv* **2017**, arXiv:1703.10099.
30. Fitzsimons, J.; Granziol, D.; Cutajar, K.; Osborne, M.; Filippone, M.; Roberts, S. Entropic Trace Estimates for Log Determinants. *arXiv* **2017**, arXiv:1704.07223.
31. Davis, T.A.; Hu, Y. The University of Florida sparse matrix collection. *ACM Trans. Math. Softw.* **2011**, *38*, 1. [CrossRef]
32. Hennig, P.; Osborne, M.A.; Girolami, M. Probabilistic numerics and uncertainty in computations. *Proc. R. Soc. A* **2015**, *471*, 20150142. [CrossRef]
33. Bergstra, J.S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. In Proceedings of the 24th International Conference on Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; pp. 2546–2554.
34. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 2951–2959.
35. Thornton, C.; Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 847–855.
36. Hoffman, M.; Shahriari, B.; Freitas, N. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In Proceedings of the 17th International Conference on Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–24 April 2014; pp. 365–374.
37. Lizotte, D.J.; Wang, T.; Bowling, M.H.; Schuurmans, D. Automatic Gait Optimization with Gaussian Process Regression. *IJCAI* **2007**, *7*, 944–949.
38. Martinez-Cantin, R.; de Freitas, N.; Doucet, A.; Castellanos, J.A. Active policy learning for robot planning and exploration under uncertainty. *Robotics Sci. Syst.* **2007**, *3*, 321–328.
39. Azimi, J.; Jalali, A.; Fern, X. Hybrid batch Bayesian optimization. *arXiv* **2012**, arXiv:1202.5597.

40. Brochu, E.; Cora, V.M.; de Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv* **2010**, arXiv:1012.2599.
41. Hennig, P.; Schuler, C.J. Entropy search for information-efficient global optimization. *J. Mach. Learn. Res.* **2012**, *13*, 1809–1837.
42. Hershey, J.R.; Olsen, P.A. Approximating the Kullback Leibler divergence between Gaussian mixture models. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, HI, USA, 15–20 April 2007.
43. Huber, M.F.; Bailey, T.; Durrant-Whyte, H.; Hanebeck, U.D. On entropy approximation for Gaussian mixture random vectors. In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Seoul, South Korea, 20–22 August 2008; pp. 181–188.
44. Molga, M.; Smutnicki, C. Test Functions for Optimization Needs. Available online: http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf (accessed on 30 May 2005).
45. Dixon, L.C.W. The global optimization problem. An introduction. *Toward Glob. Optim.* **1978**, *2*, 1–15.
46. Billingsley, P. *Probability and Measure*; Wiley: Hoboken, NJ, USA, 2012.