entropy

MDPI

Article

Numerical Evaluation of Gaussian Mixture Entropy

Basheer Joudeh * and Boris Škorić *

Department of Computer Science and Mathematics, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands

* Correspondence: b.joudeh@tue.nl (B.J.); b.skoric@tue.nl (B.Š.)

Abstract

We develop an approximation method for the differential entropy $h(\mathbf{X})$ of a q-component Gaussian mixture in $\mathbb{R}^n$. We provide two examples of approximations using our method denoted by $\bar{h}_{C,m}^{\text{Taylor}}(\mathbf{X})$ and $\bar{h}_{C}^{\text{Polyfit}}(\mathbf{X})$. We show that $\bar{h}_{C,m}^{\text{Taylor}}(\mathbf{X})$ provides an easy-to-compute lower bound to $h(\mathbf{X})$, while $\bar{h}_{C}^{\text{Polyfit}}(\mathbf{X})$ provides an accurate and efficient approximation to $h(\mathbf{X})$. $\bar{h}_{C}^{\text{Polyfit}}(\mathbf{X})$ is more accurate than known bounds and is conjectured to be much more resilient than other approximations in high dimensions.

Keywords: Gaussian mixture; entropy; mixture distribution; differential entropy

1. Introduction

1.1. Differential Entropy of Gaussian Mixtures

A Gaussian mixture is a probability density function on $\mathbb{R}^n$ of the form $f(\mathbf{x}) = \sum_{j=1}^{q} p_j \mathcal{N}_{\mathbf{w}_j,K_j}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^n$, the $p_j > 0$ are weights satisfying $\sum_{j=1}^{q} p_j = 1$, and $\mathcal{N}_{\mathbf{w},K}$ stands for a Gaussian distribution with mean $\mathbf{w}$ and covariance matrix K. In other words, f is a mixture of q Gaussian pdfs, with arbitrary weights, allowing each Gaussian to be of different shape and displacement. A mixture like this occurs, e.g., when a stochastic process, with probability mass function $p_1, \ldots, p_q$, determines which distribution holds for $\mathbf{x}$. Such situations occur in many scientific areas. Furthermore, Gaussian mixtures are often employed as function approximators, by virtue of being smooth and localized. They have been used in a wide variety of studies, e.g., on diffusion models in physics and machine learning [1–4], non-adiabatic thermodynamics [5], wireless communication [6], wireless authentication [7], Byzantine attacks [8], gene expression [9,10], dark matter kinematics [11], and quasar spectra [12]. The differential entropy $h(X)$ of a continuous random variable $X \in \mathcal{X}$, with probability density function f_X, is defined as $h(X) = -\int_{\mathcal{X}} f_X(x) \ln f_X(x)\, \mathrm{d}x$. It represents the continuum limit of the (discrete) Shannon entropy for the probability mass function $f_X(x_i)\triangle x$, where the volume $\triangle x$ is sent to zero and the infinite contribution $\ln \frac{1}{\triangle x}$ is subtracted. The differential entropy of a random vector admits the same functional form as in the univariate case, and our results hold more generally in n dimensions.

In thermodynamics, the dissipation of heat associated with the irreversible erasure of information is at least $k_b T \ln 2$ as shown by Landauer [13]. In particular, Gaussian mixtures have been used to model the state of a bit before erasure, and their differential entropy is important to further bound the energy, hence improving upon the Landauer bound [14]. In [15], it is shown how neuronal populations encode information in the presence of an external source. The mutual information between the input signal and the stationary distribution of the neuronal populations is used to quantify the information encoded by the neurons, and positivity hinges on the time scale of the input. Although when the input

Academic Editor: Andrea Murari

Received: 26 February 2026
Revised: 23 March 2026
Accepted: 25 March 2026
Published: 30 March 2026

evolves rapidly the mutual information vanishes, it is shown that when the input signal is of sufficiently low frequency, the stationary distribution of the neural populations is a Gaussian mixture, and evaluating its entropy accurately is crucial in that setting. In machine learning, the Active Diffusion Subsampling method [16] uses so-called particles whose distribution is approximated by a Gaussian mixture. The entropy of the mixture plays a crucial role in choosing new samples.

Analytically computing or estimating the differential entropy of a Gaussian mixture is a notoriously difficult problem. Even numerical evaluation can be problematic in high dimensions. The special case of a single Gaussian is simple and yields $h(X) = \frac{1}{2} \ln \det(2\pi e K)$.

1.2. Related Work

Various methods have been proposed to approximate the differential entropy of a Gaussian mixture. A loose upper bound can be obtained from the fact that a Gaussian distribution maximizes entropy, given the first and second moments. This yields [17] $h(X) \leq \frac{1}{2} \ln \det(2\pi e \Sigma)$, with $\Sigma = \sum_{i=1}^{q} p_i (w_i w_i^{\mathrm{T}} + K_i) - \left(\sum_{i=1}^{q} p_i w_i\right)\left(\sum_{j=1}^{q} p_j w_j\right)^{\mathrm{T}}$. In [18], a numerical approximation was given for the case $q = 2, n = 1, w_2 = -w_1$. A sequence of upper bounds for arbitrary q was obtained in [17], but only for $n = 1$. Furthermore, the results are not in closed form, and the bounds in the sequence do not get progressively tighter. Another method is to replace the density f inside the logarithm by a single Gaussian $\bar{f}$ which has mean and covariance equal to the mixture. This leads to an exact expression containing the relative entropy, $h(X) = \frac{1}{2} \ln \det(2\pi e \Sigma) - D(f||\bar{f})$; one can then find approximations or bounds for the relative entropy, as in [19,20]. This is done with Monte Carlo integration, which has the drawback of being computationally demanding and not giving an analytic expression.

In [21], an approximation was obtained by performing a Taylor expansion of $\ln f(x)$ in the variables $x - w_j$. To avoid the need for expansion powers above 2, they introduced the trick of representing wide Gaussians approximately as the sum of several narrow Gaussians, in the f outside the logarithm. The Taylor expansion, as well as the splitting trick, introduces inaccuracies. As shown in [21], one can obtain a basic concavity deficit upper bound $h(X) \leq \sum_{j=1}^{q} p_j \ln \frac{1}{p_j} + \sum_{j=1}^{q} p_j \frac{1}{2} \ln \det(2\pi e K_j)$. This is significantly tighter than the above-mentioned bound $\frac{1}{2} \ln[(2\pi e)^n \det \Sigma]$; it is exact in the case of a single Gaussian, and it gets arbitrarily close to the true value of $h(X)$ when the overlap between the components of the mixture becomes negligible. A refinement was also introduced by means of merging parts of the mixture that are clustered together. However, in configurations where the mixture components have significant overlap whilst keeping the mixture non-trivial, this bound becomes inaccurate.

In [22], tighter bounds are introduced for the differential entropy of general mixture distributions. In particular, the concavity deficit bound is refined as $h(X) \leq (\sup_i ||g_i - \bar{g}_i||_{\mathrm{TV}}) \sum_{j=1}^{q} p_j \ln \frac{1}{p_j} + \sum_{j=1}^{q} p_j \frac{1}{2} \ln \det(2\pi e K_j)$, where $|| \cdot ||_{\mathrm{TV}}$ denotes the total variation distance, $\{g_i\}$ are the components of the mixture, and $\bar{g}_j = \sum_{i \neq j} \frac{p_i}{1-p_j} g_i$ is the mixture complement of g_j. Calculating the total variation distances between each component and its complement becomes computationally demanding when both the number of components and the dimension are large.

1.3. Contributions

We introduce a new method for estimating the differential entropy of a Gaussian mixture. We approximate $f \ln \frac{1}{f}$ by a polynomial $\mathrm{poly}(f)$. For any positive integer k, the power $f^k = (p_1 \mathcal{N}_1 + \ldots + p_q \mathcal{N}_q)^k$ can be written as a multinomial sum containing a product of powers of Gaussians. Hence, the integral $\int \mathrm{poly}(f(x))\, \mathrm{d}x$ can be computed

analytically. This yields a systematic way to approximate and/or lower bound $h(X)$ by analytic expressions.

- We consider polynomials that minimize the square error $\int \text{weight}(s)[s \ln \frac{1}{s} - \text{poly}(s)]^2 \, \mathrm{d}s$ on the range of f, with tunable function $\text{weight}(s) = s^r$. We observe that negative r gives better results than positive r, which is to be expected since most of the volume in $\mathbb{R}^n$ has $f(x)$ close to zero. In particular, $r \approx -2$ performs best in the mixture configurations that we have studied, yielding relative errors in $h(X)$ of less than 1% already for the degree-3 polynomial fit.
- We consider the truncated Taylor series $-\ln \frac{f}{m} \approx \sum_{k=1}^{\cdots} \frac{1}{k}(1 - \frac{f}{m})^k$ for some constant m. This too gives rise to an approximation for $f \ln \frac{1}{f}$ that is polynomial in f. While not as accurate as the above-mentioned fit, it guarantees that each k-term has the same sign if $m \geq \max_{x \in \mathbb{R}^n} f(x)$, and hence it gives rise to a sequence of increasingly accurate analytic lower bounds on $h(X)$. We observe that the relative error in $h(X)$ is still several percent even at polynomial degree 10.

In Section 2.1, we obtain a general recipe to approximate the differential entropy via a polynomial approximation as shown in Corollary 1. In Section 2.2, we apply Corollary 1 to the Taylor series of the logarithm to obtain a specific approximation for the differential entropy. Although it is not a very accurate approximation, it serves as an easy-to-compute lower bound. In Section 2.3, we apply Corollary 1 again using a polynomial approximation of $f(s) = -s \ln s$, which gives an accurate approximation to the differential entropy. In Section 3, we show numerical results of our approximations for various configurations, and in Section 4, we assess our method and give pointers towards future work.

2. Polynomial Approximation

2.1. General Polynomial

Consider a random variable $\mathbf{X} \in \mathbb{R}^n$ whose probability density function (pdf) is a Gaussian mixture with weights $\{p_i\}_{i=1}^q$. The Gaussian pdfs have covariance matrices $\{K_i\}_{i=1}^q$, and they are centered on points $\mathbf{w}_1, \ldots, \mathbf{w}_q \in \mathbb{R}^n$.

$$f_{\mathbf{X}}(\mathbf{x}) = \sum_{j=1}^{q} p_j \mathcal{N}_{\mathbf{w}_j, K_j}(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} \sum_{j=1}^{q} p_j (\det K_j)^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{w}_j)^{\mathrm{T}} K_j^{-1} (\mathbf{x}-\mathbf{w}_j)}. \tag{1}$$

We let $g_i(\mathbf{x}) \equiv \mathcal{N}_{\mathbf{w}_i, K_i}(\mathbf{x})$ denote the Gaussian pdf in the following. The differential entropy is $h(\mathbf{X}) = -\mathbb{E}_{\mathbf{X}}[\ln f_{\mathbf{X}}]$, which can be written as follows:

$$h(\mathbf{X}) = -\mathbb{E}_{\mathbf{X}}[\ln f_{\mathbf{X}}] = -\mathbb{E}_{\mathbf{X}}[\ln m m^{-1} f_{\mathbf{X}}] = -\ln m - \mathbb{E}_{\mathbf{X}}\left[\ln \frac{f_{\mathbf{X}}}{m}\right], \tag{2}$$

where m can be chosen to enforce that the argument of the logarithm lies in a certain interval. We would like to approximate the entropy by means of a polynomial approximation in powers of $f_{\mathbf{X}}$:

$$-f_{\mathbf{X}} \ln \frac{f_{\mathbf{X}}}{m} \approx \sum_{a=1}^{C} c_a f_{\mathbf{X}}^a, \tag{3}$$

In other words, we consider an order-C polynomial approximation. We set $c_0 = 0$ to avoid diverging integrals. Equation (3) is more relaxed than truncating a series of the form $\sum_{a=1}^{\infty} c_a f_{\mathbf{X}}^a$; i.e., we allow $\{c_a\}$ to depend on C. Using Equation (3), the differential entropy $h(\mathbf{X})$ reads as follows:

$$h(\mathbf{X}) \approx -\ln m + \sum_{a=1}^{C} c_a \int f_{\mathbf{X}}^a(\mathbf{x}) \mathrm{d}\mathbf{x}, \tag{4}$$

and by choosing $\{c_a\}$ appropriately and analytically evaluating the integral, we obtain an efficient approximation for $h(\mathbf{X})$. We start by rewriting the integral in Equation (4), and the result is Corollary 1.

Lemma 1. *Let $\{g_i(\mathbf{x})\}_{i=1}^{q}$ be given in accordance with Equation (1), $\hat{t} \equiv (t_1, \ldots, t_q)$ s.t. $t_i \geq 0$ and $\sum_{i=1}^{q} t_i = a$. Furthermore, let M^{-1} and $\boldsymbol{\mu}$ be given by the following:*

$$M^{-1} = \sum_{j=1}^{q} t_j K_j^{-1}, \tag{5}$$

$$\boldsymbol{\mu} = \sum_{j=1}^{q} t_j M K_j^{-1} \mathbf{w}_j, \tag{6}$$

and then we have the following:

$$\int \prod_{j=1}^{q} g_j^{t_j}(\mathbf{x}) \mathrm{d}\mathbf{x} = D(\hat{t}), \tag{7}$$

where $D(\hat{t})$ is given by the following:

$$D(\hat{t}) = (2\pi)^{-n(a-1)/2} \left(\prod_{i=1}^{q} (\det K_i)^{-t_i/2} \right) (\det M)^{1/2} e^{-\frac{1}{2}\left(\sum_{l=1}^{q} t_l \mathbf{w}_l^{\mathrm{T}} K_l^{-1} \mathbf{w}_l - \boldsymbol{\mu}^{\mathrm{T}} M^{-1} \boldsymbol{\mu}\right)}. \tag{8}$$

Proof. See Appendix A. □

It follows that the integral in Equation (4) can be rewritten as shown in the following corollary.

Corollary 1. *The differential entropy $h(\mathbf{X})$ of the Gaussian mixture is approximated by the following:*

$$\bar{h}_{\vec{c},m}(\mathbf{X}) = -\ln m + \sum_{a=1}^{C} c_a \sum_{\hat{t} \in \mathcal{T}_a} \binom{a}{\hat{t}} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}), \tag{9}$$

where $\mathcal{T}_a = \{\hat{t} \in \mathbb{Z}_+^q : |\hat{t}| = a\}$ and $\binom{a}{\hat{t}} = \binom{a}{t_1 \ldots t_q}$.

Proof. See Appendix B. □

2.2. Taylor Series Approximation

We perform a Taylor series for the logarithm only, in order to obtain the coefficients $\{c_a\}$ in Equation (3). We define $Z \equiv 1 - m^{-1} f_{\mathbf{X}}(\mathbf{X})$ and we write:

$$-f_{\mathbf{X}} \ln \frac{f_{\mathbf{X}}}{m} = -f_{\mathbf{X}} \ln(1 - Z) = f_{\mathbf{X}} \sum_{k=1}^{\infty} \frac{1}{k} Z^k. \tag{10}$$

The range of allowed m values is the one that keeps Z in the radius of convergence of the Taylor series: $|Z| < 1$, i.e., $m \geq \max_{\mathbf{x} \in \mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})/2$. For $m = \max_{\mathbf{x} \in \mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})/2$ then $Z \in [-1, 1)$, while for $m = \max_{\mathbf{x} \in \mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})$ then $Z \in [0, 1)$. Furthermore, values above $\max_{\mathbf{x} \in \mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})$ such as $m = \sum_i p_i g_i(\mathbf{w}_i)$ shrink the range of Z from below towards 1. By performing the Taylor expansion for the logarithm, we obtain a series of the form given by Equation (3) as shown in Theorem 1.

https://doi.org/10.3390/e28040381

Theorem 1. *The procedure of performing the Taylor expansion of the logarithm as detailed in Equation (10) yields the following coefficients* $\{c_a^{\text{Taylor}}\}_{a=1}^{C}$ *of the corresponding polynomial approximation in Equation (3):*

$$c_a^{\text{Taylor}} = \begin{cases} H_{C-1}, & a = 1, \\ \frac{(-1)^{a+1}}{m^{a-1}} \frac{1}{a-1} \binom{C-1}{a-1}, & \text{otherwise.} \end{cases}, \tag{11}$$

and the corresponding differential entropy approximation as given by Equation (9) can be written as follows:

$$\bar{h}_{C,m}^{\text{Taylor}}(\mathbf{X}) = -\ln m + H_{C-1} - m \sum_{a=1}^{C-1} \frac{B_{a+1}}{a} \binom{C-1}{a}, \tag{12}$$

where H_k *is the k-th Harmonic number, and* $\{B_a\}$ *are given by the following:*

$$B_a = \frac{(-1)^a}{m^a} \sum_{\hat{t} \in \mathcal{T}_a} \binom{a}{\hat{t}} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}). \tag{13}$$

Proof. See Appendix C. □

Note that one can show that $h(\mathbf{X})$ is given by the infinite series:

$$h(\mathbf{X}) = -\ln m - m \sum_{k=1}^{\infty} \sum_{a=0}^{k} \frac{B_{a+1}}{k} \binom{k}{a}, \tag{14}$$

which is not the result of an infinite power series in $f_{\mathbf{X}}^a$ as evident by the a summation which is inside the k summation. Taking the first $C-1$ terms and changing the order of summations makes the coefficients $\{c_a^{\text{Taylor}}\}$ dependent on C. Note that Theorem 1 with $m = \max_{\mathbf{x} \in \mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})$ provides a lower bound for $h(\mathbf{X})$. For some applications, having a bound is more important than having a good approximation.

2.3. Polyfit

2.3.1. General Polyfit

Suppose we have a function $f(s)$ defined on $\mathcal{I} = [a, b]$ that we wish to approximate via a polynomial expansion. We write

$$f(s) \approx \tilde{f}(s) = \sum_{i=1}^{C} d_{\mathcal{I},i} s^i, \tag{15}$$

and we define the error resulting from our estimation as follows:

$$E = \frac{1}{b-a} \int_a^b w(s)(f(s) - \tilde{f}(s))^2 \mathrm{d}s, \tag{16}$$

where we consider the possibility of favoring portions of $\mathcal{I}$ more than others by choosing the weight function $w(s)$ appropriately, and $w(s) = 1$ reduces E to the mean squared error. Our choice of w and $\mathcal{I}$ must be such that $M_{\mathcal{I}}$ in Lemma 2 is invertible.

Lemma 2. *Let $M_{\mathcal{I}}$, $\vec{d}_{\mathcal{I}}$, and $\vec{z}_{\mathcal{I}}$ be given by the following:*

$$(M_{\mathcal{I}})_{ij} \equiv \int_a^b w(s) s^{i+j} \mathrm{d}s, \tag{17}$$

$$(\vec{d}_{\mathcal{I}})_i \equiv d_{\mathcal{I},i}, \tag{18}$$

$$(\vec{z}_{\mathcal{I}})_i \equiv \int_a^b w(s) f(s) s^i \mathrm{d}s, \tag{19}$$

and then for invertible $M_{\mathcal{I}}$, the coefficients $d_{\mathcal{I},i}$ in Equation (15) that minimize the error function E in Equation (16) are given by the following:

$$\vec{d}_{\mathcal{I}} = M_{\mathcal{I}}^{-1} \vec{z}_{\mathcal{I}}. \tag{20}$$

Proof. See Appendix D. □

2.3.2. Entropy Estimate Obtained from Polyfit

We now wish to use Lemma 2 to obtain an approximation for $h(\mathbf{X})$. We first apply our estimator to the function $f(s) = -s \ln s$ in the following lemma.

Lemma 3. *Let $f(s) = -s \ln s$ on $\mathcal{I} = (a, b]$, $r \in \mathbb{R}$ if $a > 0$ and $r > -3$ if $a = 0$. Let $w(s) = s^r$, and then the estimator $\tilde{f}(s)$ in Equation (15) is given by the following:*

$$-s \ln s \approx \sum_{i=1}^{C} d_{\mathcal{I},i} s^i, \tag{21}$$

where $d_{\mathcal{I},i}$ are obtained by solving Equation (20) with $M_{\mathcal{I}}$ and $\vec{z}_{\mathcal{I}}$ given by the following:

$$(M_{\mathcal{I}})_{ij} = \frac{b^{i+j+r+1} - a^{i+j+r+1}}{i+j+r+1}, \tag{22}$$

$$(\vec{z}_{\mathcal{I}})_i = \frac{\left[b^{i+r+2}(1 - (i+r+2)\ln b) - (b \to a)\right]}{(i+r+2)^2}.$$

Proof. It follows from Equation (15), Lemma 2, and direct calculation. □

Corollary 2. *Let $f(s) = -s \ln s$ on $\mathcal{I} = (0, b]$, and $w(s) = s^r$ with $r > -3$; then, $\{d_{\mathcal{I},i}\}$ are given by the following:*

$$d_{\mathcal{I},i} = \begin{cases} \tilde{d}_1 - \ln b, & \text{if } i = 1 \\ b^{1-i} \tilde{d}_i, & \text{otherwise} \end{cases}, \tag{23}$$

where $\vec{\tilde{d}}$ is obtained by solving

$$\vec{\tilde{d}} = \tilde{M}^{-1} \vec{\tilde{z}}, \tag{24}$$

with $\tilde{M}$ and $\vec{\tilde{z}}$ given by the following:

$$\tilde{M}_{ij} = \frac{1}{i+j+r+1}, \quad \tilde{z}_i = \frac{1}{(i+r+2)^2}. \tag{25}$$

Proof. See Appendix E. □

Corollary 2 shows that the problem of finding a polynomial fit for $f(s) = -s \ln s$ on $(0, b]$ is equivalent to finding the inverse of $\tilde{M}$, which is independent of b. However, $\tilde{M}$ is an ill-conditioned matrix as the elements $\tilde{M}_{ij}$ become increasingly smaller for higher $i + j$. For example, if we take $r = -2$, then $\tilde{M}$ is the Hilbert matrix with inverse elements,

$(H^{-1})_{ij} = (-1)^{i+j}(i+j-1)\binom{C+i-1}{C-j}\binom{C+j-1}{C-i}\left[\binom{i+j-2}{i-1}\right]^2$, which become numerically unstable at $C \sim 10$. If we write $f_{\max} \equiv \max_{\mathbf{x}\in\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})$, then for our Gaussian mixture $f_{\mathbf{X}}$, the possible values are in $(0, f_{\max}]$. In a very large part of $\mathbb{R}^n$, we find ourselves in the tails of the Gaussian mixture, which corresponds to $f_{\mathbf{X}}(\mathbf{x})$ being close to zero. Hence, we want our weight function $w(s)$ to reflect this, and emphasize the region close to zero in the application of Corollary 2. Note that we can write

$$\begin{aligned} h(\mathbf{X}) &= -\int f_{\mathbf{X}}(\mathbf{x}) \ln f_{\mathbf{X}}(\mathbf{x}) \mathrm{d}\mathbf{x} = -\int \left(\int_0^{f_{\max}} \delta(s - f_{\mathbf{X}}(\mathbf{x})) s \ln s \, \mathrm{d}s \right) \mathrm{d}\mathbf{x} \\ &= -\int_0^{f_{\max}} s \ln s \left(\int \delta(s - f_{\mathbf{X}}(\mathbf{x})) \mathrm{d}\mathbf{x} \right) \mathrm{d}s = \int_0^{f_{\max}} f(s) V(s) \, \mathrm{d}s, \end{aligned} \tag{26}$$

where $V(s) = \int \delta(s - f_{\mathbf{X}}(\mathbf{x}))\mathrm{d}\mathbf{x}$ can be thought of as the $(n-1)$-dimensional volume where the function $f_{\mathbf{X}}$ equals s. We see from Equation (26) that our entropy approximation is not only determined by how well we can estimate $f(s)$ using Corollary 2, but also by $V(s)$, which for Gaussian mixtures will put more weight around $s = 0$. That is, we expect $r < 0$ in Corollary 2 to produce better results for our entropy approximation than $r > 0$, although they will produce worse estimates of $f(s)$ over any interval. Ideally, we set $w(s) = V(s)$ in Lemma 3, but $V(s)$ has no closed-form solution. This leads us to the following entropy approximation given by Theorem 2.

Theorem 2. *Let $r > -3$, $\mathcal{I} = (0, f_{\max}]$; then, the entropy approximation $\bar{h}_C^{\text{Polyfit}}(\mathbf{X})$ based on Lemma 3, Corollary 2, and Corollary 1 is given by the following:*

$$\bar{h}_C^{\text{Polyfit}}(\mathbf{X}) = \sum_{a=1}^{C} c_a^{\text{Polyfit}} \sum_{\hat{t}\in\mathcal{T}_a} \binom{a}{\hat{t}} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}), \tag{27}$$

where $\vec{c}^{\,\text{Polyfit}}$ is given by the following:

$$c_a^{\text{Polyfit}} = \begin{cases} \tilde{d}_1 - \ln f_{\max}, & \text{if } a = 1 \\ f_{\max}^{1-a} \tilde{d}_a, & \text{otherwise} \end{cases}, \quad \vec{\tilde{d}} = \tilde{M}^{-1} \vec{\tilde{z}}, \tag{28}$$

with $\tilde{M}_{ij}$ and $\tilde{z}_i$ given by the following:

$$\tilde{M}_{ij} = \frac{1}{i+j+r+1}, \quad \tilde{z}_i = \frac{1}{(i+r+2)^2}. \tag{29}$$

Proof. It follows from the direct application of Lemma 3, Corollary 2, and Corollary 1 to $-f_{\mathbf{X}} \ln f_{\mathbf{X}}$. □

In Appendix F, we derive the volume function $V(s)$ for the trivial case $q = 1$. Around $s \approx 0$, it blows up as $1/s$, with some additional logarithmic divergence; this gives some intuition as to why we get good results at $r \leq -1$.

3. Numerical Results

3.1. Polyfit Results

We now present results of Theorem 2 applied to different Gaussian mixtures in order to estimate $h(\mathbf{X})$. The performance metric we use is the percentage error compared to the exact value for the entropy. The mixtures we used to show the validity of our approximation are shown in Table 1. The examples cover various values of n and q. This is not meant to be

all-encompassing, but it still covers a sufficient part of the parameter space to showcase the accuracy of our approximation.

Table 1. The different Gaussian mixtures tested using Theorem 2. The resulting entropy errors appear in Figure 1.

q	n	Mixture Parameters
3	2	$\mathbf{w}_1 = (1,0)^{\mathrm{T}}$, $\mathbf{w}_2 = (-1,0)^{\mathrm{T}}$, $\mathbf{w}_3 = (0,1.5)^{\mathrm{T}}$, $K_i = \mathrm{I}_2$, $\hat{p} = (0.2,0.3,0.5)^{\mathrm{T}}$
	2	$\mathbf{w}_1 = (0,0)^{\mathrm{T}}$, $\mathbf{w}_2 = (-1.5,1.5)^{\mathrm{T}}$, $\mathbf{w}_3 = (1.5,1.5)^{\mathrm{T}}$, $K_1 = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$, $K_2 = \begin{pmatrix} 1 & 0.2 \\ 0.2 & 1 \end{pmatrix}$, $K_3 = \begin{pmatrix} 1 & -0.2 \\ -0.2 & 1 \end{pmatrix}$, $\hat{p} = (0.2,0.3,0.5)^{\mathrm{T}}$
4	3	$\mathbf{w}_1 = (0,0,0)^{\mathrm{T}}$, $\mathbf{w}_2 = (-1.5,1.5,-1.5)^{\mathrm{T}}$, $\mathbf{w}_3 = (1.5,1.5,1.5)^{\mathrm{T}}$, $\mathbf{w}_4 = (1,1,1)^{\mathrm{T}}$, $K_i = \mathrm{I}_3$, $\hat{p} = (0.2,0.3,0.3,0.2)^{\mathrm{T}}$
	8	$\mathbf{w}_1 = (0,\ldots,0)^{\mathrm{T}}$, $\mathbf{w}_2 = (-1.5,1.5,\ldots,1.5)^{\mathrm{T}}$, $\mathbf{w}_3 = (1.5,\ldots,1.5)^{\mathrm{T}}$, $\mathbf{w}_4 = (1,\ldots,1)^{\mathrm{T}}$, $K_i = \mathrm{I}_8$, $\hat{p} = (0.2,0.3,0.3,0.2)^{\mathrm{T}}$
5	4	$\mathbf{w}_1 = (0,0,0,0)^{\mathrm{T}}$, $\mathbf{w}_2 = (-1.5,1.5,-1.5,1.5)^{\mathrm{T}}$, $\mathbf{w}_3 = (1.5,1.5,1.5,1.5)^{\mathrm{T}}$, $\mathbf{w}_4 = (1,1,1,1)^{\mathrm{T}}$ $\mathbf{w}_5 = (-3,3,-3,3)^{\mathrm{T}}$, $K_i = \mathrm{I}_4$, $\hat{p} = (0.2,0.3,0.3,0.1,0.1)^{\mathrm{T}}$

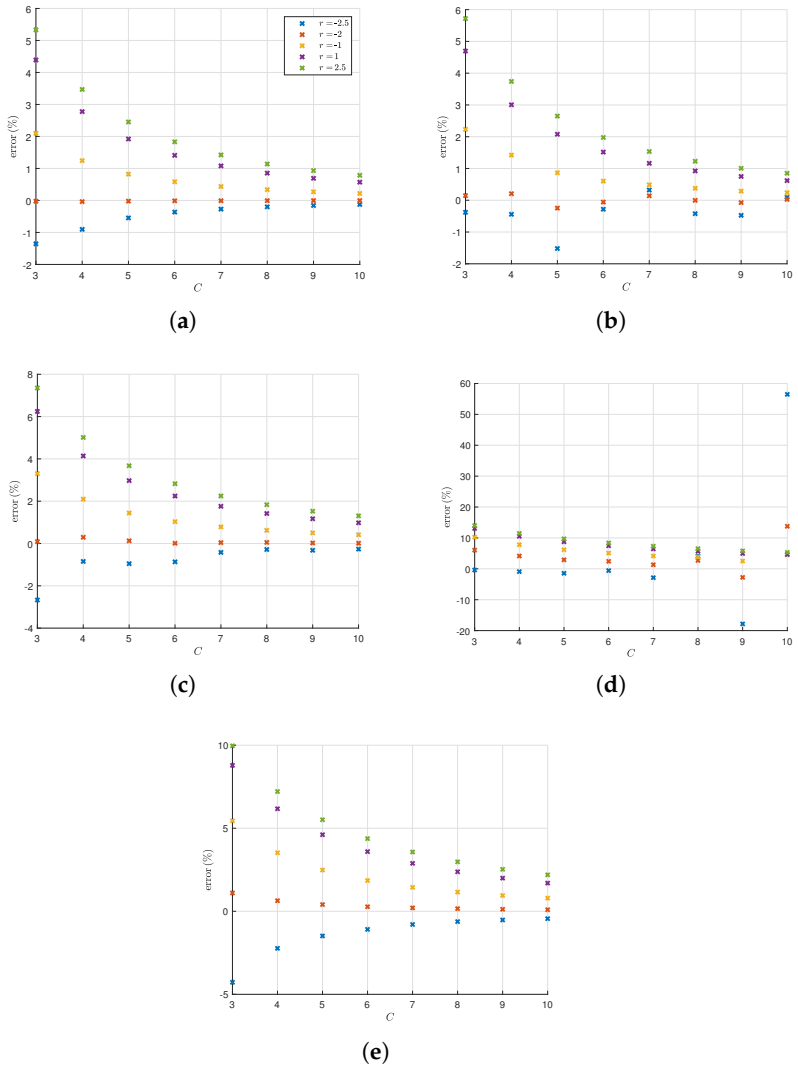


Figure 1. The results of testing Theorem 2 on the Gaussian mixtures in Table 1. The graphs show the percentage error $\left(\frac{h(\mathbf{X})-\bar{h}_C^{\mathrm{Polyfit}}(\mathbf{X})}{h(\mathbf{X})}\,100\right)$ of the approximation $\bar{h}_C^{\mathrm{Polyfit}}(\mathbf{X})$ at different values of C. **(a)** $q = 3, n = 2, K_i = I_2$; **(b)** $q = 3, n = 2$; **(c)** $q = 4, n = 3$; **(d)** $q = 4, n = 8$; **(e)** $q = 5, n = 4$.

From Figure 1, we see that for low-dimensional cases ($q = 3, n = 2$), our approximation converges quickly to the exact value of the entropy (within less than 1% error), and we can fairly trust numerical accuracy for those cases. As for the optimal r value for the weight function, $r = -2$ is consistently the best choice for those parameters, with $r > 0$ being less favorable, as predicted in Section 2.3.2.

At $q = 4$, we start to see divergent behavior as n becomes larger. For the lower-dimensional case of $q = 4, n = 3$, the behavior is similar to previous cases with very high accuracy and $r = -2$ being the most favorable weight function. However, in the case of $q = 4, n = 8$, the result diverges as C grows larger, although up to $C = 7$, the approximation is fairly accurate. Furthermore, the best-performing r in this case is $r = -2.5$ with an error percentage of around -0.5% at $C = 6$. We conjecture that the divergent behavior in this case is caused by numerical inaccuracies in the evaluation of multinomial coefficients.

For $q = 5, n = 4$, we again have convergent behavior yielding an accurate approximation of the entropy with $r = -2$ being the best weight function. This prompts us to conclude that our method produces a very highly accurate approximation when $r \in [-2.5, -2]$ and C is kept between 3 and 8 in order to avoid computational inaccuracies. Note that the upper bound mentioned in the introduction $h(\mathbf{X}) \leq \sum_{j=1}^{q} p_j \ln \frac{1}{p_j} + \sum_{j=1}^{q} p_j \frac{1}{2} \ln \det(2\pi e K_j)$ produces values that are within 3–19% above the true entropy for the configurations in Table 1.

In Figure 2, we show Polyfit approximations of $f(s) = -s \ln s$ as described in Lemma 3 for different values of r. We see clearly in Figure 2 that for $r = -2$, the approximation is overall inaccurate compared to $r = -1$ and $r = 1$. However, $r = -2$ yields the best approximation for the entropy $h(\mathbf{X})$. As we discussed in Section 2.3.2, the reason for this is that the value of $r = -2$ makes the approximation better around $s \approx 0$, which is where $V(s)$ is highest.

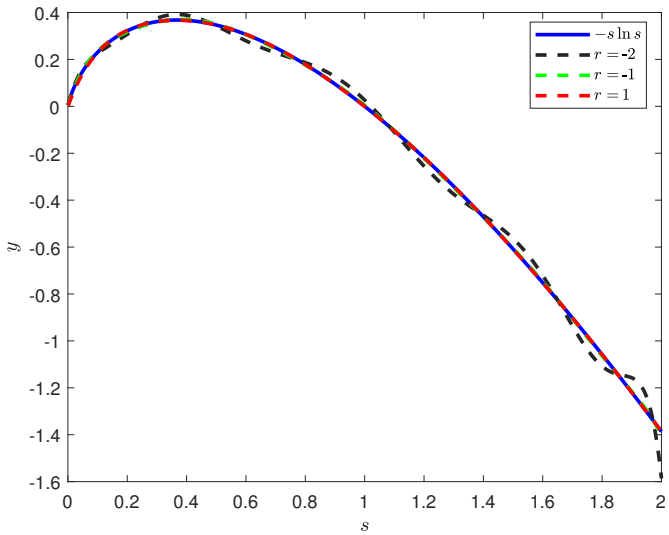
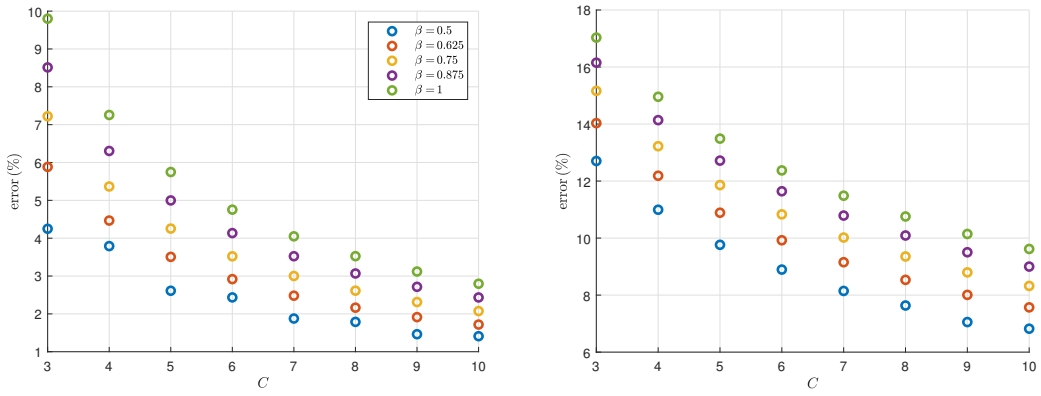


Figure 2. Polyfit approximations of $f(s) = -s \ln s$ for different values of r on the interval (0, 2] in accordance with Lemma 3.

3.2. Taylor Series Results

We now show results for our approximation $\bar{h}_{C,m}^{\text{Taylor}}(\mathbf{X})$ based on the Taylor expansion of the logarithm as outlined in Theorem 1. We do not expect it to be as accurate as $\bar{h}_{C,m}^{\text{Polyfit}}(\mathbf{X})$; however, it provides us with an easy-to-compute lower bound for the entropy, as mentioned earlier in Section 2.2. We show the percentage error compared to the exact value of the entropy for different values of β for two mixtures that we take from Table 1. We use the case of $q = 3, n = 2$ with non-spherical covariance matrices, as well as the case of $q = 4, n = 8$. The results are shown in Figure 3. As we can see from Figure 3, $\bar{h}_{C,m}^{\text{Taylor}}(\mathbf{X})$ is not as accurate

as $\bar{h}_{C,m}^{\text{Polyfit}}(\mathbf{X})$ as expected, especially in the higher-dimensional case. However, it does provide monotonous and convergent behavior. Furthermore, the case $\beta = 1/2$ is the fastest-converging case, which is also the case where computational inaccuracies start to appear the fastest as we increase C.



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

Figure 3. The results of testing Theorem 1 on two of the Gaussian mixtures in Table 1. The graphs show the percentage error $\left(\frac{h(\mathbf{X})-\bar{h}_{C,m}^{\text{Taylor}}(\mathbf{X})}{h(\mathbf{X})}\,100\right)$ of the approximation $\bar{h}_{C,m}^{\text{Taylor}}(\mathbf{X})$ at different values of C and for different values of β. (**a**) $q = 3, n = 2$; (**b**) $q = 4, n = 8$.

4. Discussion

Gaussian mixtures are one of the most applicable distributions in the literature, from wireless communications to physics and diffusion models. They have found success in either describing or modeling real life phenomena to a high degree of accuracy. One of the most important quantities in characterizing a distribution is the Shannon (differential) entropy, which, for Gaussian mixtures, does not have a known closed-form expression.

In this paper, we have derived a method by which the differential entropy of Gaussian mixtures can be approximated both accurately and efficiently. Our method enjoys a high degree of simplicity as it relies on the fact that for Gaussian mixtures "*almost all the support has an image that is almost zero*". For various configurations, this trick allows our approximation to get very close to the true value for the differential entropy by summing only a small number of terms. Although some configurations are more difficult to approximate with a small number of terms, our method is accurate for a large number of randomly generated Gaussian mixtures (not reported in Section 3). This makes inaccuracy an exception that can be circumvented by avoiding numerical instabilities and taking into account higher-order terms.

It is not straightforward to benchmark against the existing literature. Many methods are at least partly numerical, whereas our approximation technique yields fully analytic expressions. The work [21] is closest to our technique in that it approximates $h(\mathbf{X})$ with closed-form analytic expressions. However, there are significant qualitative differences. Since [21] tries to polynomially approximate $\ln f(\mathbf{x})$, the order C of their Taylor series has to increase with q as $C = 2q$ (in the worst case) in order to correctly fit all the 'bumps' in the function. The error in $\ln f(\mathbf{x})$ then explodes as $|\mathbf{x}|^{2q}$ when $|\mathbf{x}|$ increases beyond a certain radius. This large error should be damped by the fact that it gets multiplied by the Gaussian tail of $f(\mathbf{x})$. However, at large q, this can require drastic application of the splitting trick, which introduces errors. In contrast, our choice of polynomial order C is independent of q, and increasing q does not lead to errors. In our technique, inaccuracies are caused by the mismatch between the function $s \ln \frac{1}{s}$ and the polynomial fit around $s = 0$. The performance of our technique is constrained by the fact that the function $s \ln \frac{1}{s}$ has infinite derivative at $s = 0$ and hence cannot be fitted with a polynomial of finite degree at that point.

Our Polyfit method relies on choosing an appropriate weight function $w(s)$ as discussed in Section 2.3. Ideally for Gaussian mixtures, $w(s)$ should be optimally chosen as $V(s)$, which is difficult to estimate. However, from our experiments, we notice that a simple power $w(s) \propto s^r$ gives good results, especially for negative r around $r = -2$. At negative r, a lot of weight is given to small values of s. This makes sense because most of the n-dimensional volume in the integral $\int \mathrm{d}\mathbf{x}$ covers the tails of the distribution $f(\mathbf{X})$. For a single Gaussian with spherical symmetry, it is easy to verify (see Appendix F) that the volume function $V(s) = \int \mathrm{d}\mathbf{x}\, \delta(s - f(\mathbf{x}))$ behaves as $V(s) \propto \frac{1}{s}(\ln \frac{f_{\max}}{s})^{\frac{n}{2}-1}$. At small s, this expression blows up faster than $\frac{1}{s}$ but slower than $\frac{1}{s^2}$. This suggests that r should lie in the vicinity of $[-2, -1]$. More extensive testing, with larger mixtures in higher dimensions, will show how generally applicable our method is. We have not exhaustively studied all possible polynomial fits. It is quite likely that some improvement can be gained by choosing a better weight function, e.g., one that more resembles $V(s)$.

Projection pursuit was introduced in 1974 as a method for analyzing multivariate data through its lower-dimensional projections [23]. However, exploring the full space of projections becomes increasingly impractical as the dimension grows [24]. Subsequent work addressed this issue by identifying "interesting" projections, namely those that reveal structure in the original data, through indices that measure departures from Gaussianity. One such index is differential entropy [25]. When the data arise from a finite mixture, skewness has also been used as a measure of non-normality for classifying projections [26,27]. Other approaches include the differential entropy estimator proposed in [28]. It would be of interest to compare these approaches with accurate estimation of differential entropy, using our method and taking our estimate as the projection index. In particular, an important question is whether maximizing skewness also leads to entropy optimization. A similar investigation could be carried out in relation to the work of Peña and Prieto [29–31]. These directions are left for future work.

Author Contributions: Conceptualization, B.J. and B.Š.; Methodology, B.J.; Software, B.J.; Validation, B.J. and B.Š.; Formal analysis, B.J.; Investigation, B.J. and B.Š.; Resources, B.J.; Writing—original draft, B.J.; Writing—review & editing, B.J. and B.Š.; Visualization, B.J. and B.Š.; Supervision, B.Š.; Funding acquisition, B.Š. All authors have read and agreed to the published version of the manuscript.

Funding: Joudeh was supported by NWO grant CS.001 (Forwardt) and by the Dutch Groeifonds project Quantum Delta NL KAT-2.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Appendix A. Proof of Lemma 1

$$
\begin{aligned}
\int \prod_{j=1}^{q} g_j^{t_j}(\mathbf{x})\mathrm{d}\mathbf{x} &= \int \prod_{j=1}^{q} \Big((2\pi)^{-\frac{n}{2}} (\det K_j)^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{w}_j)^{\mathrm{T}} K_j^{-1}(\mathbf{x}-\mathbf{w}_j)} \Big)^{t_j} \mathrm{d}\mathbf{x} \\
&= C(\hat{t}) \int e^{-\frac{1}{2}\sum_{j=1}^{q} t_j (\mathbf{x}-\mathbf{w}_j)^{\mathrm{T}} K_j^{-1}(\mathbf{x}-\mathbf{w}_j)} \mathrm{d}\mathbf{x} \\
&= C(\hat{t}) e^{-\frac{1}{2}\sum_{j=1}^{q} t_j \mathbf{w}_j^{\mathrm{T}} K_j^{-1} \mathbf{w}_j} \int e^{-\frac{1}{2}\sum_{j=1}^{q} t_j \left(\mathbf{x}^{\mathrm{T}} K_j^{-1} \mathbf{x} - \mathbf{x}^{\mathrm{T}} K_j^{-1} \mathbf{w}_j - \mathbf{w}_j^{\mathrm{T}} K_j^{-1} \mathbf{x} \right)} \mathrm{d}\mathbf{x} \\
&= C(\hat{t}) e^{-\frac{1}{2}\sum_{j=1}^{q} t_j \mathbf{w}_j^{\mathrm{T}} K_j^{-1} \mathbf{w}_j} \int e^{-\frac{1}{2}\left(\mathbf{x}^{\mathrm{T}} \left(\sum_{j=1}^{q} t_j K_j^{-1} \right) \mathbf{x} - \mathbf{x}^{\mathrm{T}} \left(\sum_{j=1}^{q} t_j K_j^{-1} \mathbf{w}_j \right) - \left(\mathbf{w}_j^{\mathrm{T}} \sum_{j=1}^{q} t_j K_j^{-1} \right) \mathbf{x} \right)} \mathrm{d}\mathbf{x} \\
&= C(\hat{t}) e^{-\frac{1}{2}\sum_{j=1}^{q} t_j \mathbf{w}_j^{\mathrm{T}} K_j^{-1} \mathbf{w}_j} \int e^{-\frac{1}{2}\left(\mathbf{x}^{\mathrm{T}} M^{-1} \mathbf{x} - \mathbf{x}^{\mathrm{T}} M^{-1} \boldsymbol{\mu} - \boldsymbol{\mu}^{\mathrm{T}} M^{-1} \mathbf{x} \right)} \mathrm{d}\mathbf{x} \\
&= C(\hat{t}) e^{-\frac{1}{2}\sum_{j=1}^{q} t_j \mathbf{w}_j^{\mathrm{T}} K_j^{-1} \mathbf{w}_j} e^{\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}} M^{-1} \boldsymbol{\mu}} \int e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} M^{-1} (\mathbf{x}-\boldsymbol{\mu})} \mathrm{d}\mathbf{x} \\
&= C(\hat{t}) e^{-\frac{1}{2}\sum_{j=1}^{q} t_j \mathbf{w}_j^{\mathrm{T}} K_j^{-1} \mathbf{w}_j} e^{\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}} M^{-1} \boldsymbol{\mu}} (2\pi)^{n/2} (\det M)^{1/2},
\end{aligned}
\tag{A1}
$$

where we have the following:

$$C(\hat{t}) = \prod_{j=1}^{q} (2\pi)^{-nt_j/2} (\det K_j)^{-t_j/2} = (2\pi)^{-na/2} \prod_{j=1}^{q} (\det K_j)^{-t_j/2}. \tag{A2}$$

Note that we can alternatively write:

$$\boldsymbol{\mu}^{\mathrm{T}} M^{-1} \boldsymbol{\mu} = \left(\sum_{j=1}^{q} t_j K_j^{-1} \mathbf{w}_j \right)^{\mathrm{T}} M \left(\sum_{l=1}^{q} t_l K_l^{-1} \mathbf{w}_l \right). \tag{A3}$$

Appendix B. Proof of Corollary 1

$$\begin{aligned} \int f_{\mathbf{X}}^{a}(\mathbf{x}) \mathrm{d}\mathbf{x} &= \int \left(\sum_{i=1}^{q} p_i g_i(\mathbf{x}) \right)^{a} \mathrm{d}\mathbf{x} = \int \sum_{\substack{t_1+t_2+\cdots+t_q=a \\ t_1,t_2,\ldots,t_q \geq 0}} \binom{a}{t_1, t_2, \ldots, t_q} \left(\prod_{i=1}^{q} p_i^{t_i} \right) \left[\prod_{j=1}^{q} g_j^{t_j}(\mathbf{x}) \right] \mathrm{d}\mathbf{x} \\ &= \sum_{\substack{t_1+t_2+\cdots+t_q=a \\ t_1,t_2,\ldots,t_q \geq 0}} \binom{a}{t_1, t_2, \ldots, t_q} \left(\prod_{i=1}^{q} p_i^{t_i} \right) \int \prod_{j=1}^{q} g_j^{t_j}(\mathbf{x}) \mathrm{d}\mathbf{x} = \sum_{\substack{t_1+t_2+\cdots+t_q=a \\ t_1,t_2,\ldots,t_q \geq 0}} \binom{a}{t_1, t_2, \ldots, t_q} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}), \end{aligned} \tag{A4}$$

where the last equality follows from Lemma 1. Plugging into Equation (4) gives the desired expression.

Appendix C. Proof of Theorem 1

$$\begin{aligned} f_{\mathbf{X}} Z^k &= f_{\mathbf{X}} \sum_{b=0}^{k} \binom{k}{b} \left(-m^{-1} f_{\mathbf{X}} \right)^b = \sum_{b=0}^{k} \binom{k}{b} (-m)^{-b} f_{\mathbf{X}}^{b+1} = -m \sum_{b=0}^{k} \binom{k}{b} (-1)^{b+1} \frac{f_{\mathbf{X}}^{b+1}}{m^{b+1}} \\ &= -m \sum_{a=1}^{k+1} \binom{k}{a-1} (-1)^a \frac{f_{\mathbf{X}}^a}{m^a}, \end{aligned} \tag{A5}$$

and plugging into Equation (10), we have the following:

$$-f_{\mathbf{X}} \ln \frac{f_{\mathbf{X}}}{m} = \sum_{k=1}^{\infty} \frac{-m}{k} \sum_{a=1}^{k+1} \binom{k}{a-1} (-1)^a \frac{f_{\mathbf{X}}^a}{m^a} = \sum_{k=1}^{\infty} \sum_{a=1}^{k+1} \tilde{c}_{k,a} f_{\mathbf{X}}^a, \tag{A6}$$

where $\tilde{c}_{k,a}$ is given by the following:

$$\tilde{c}_{k,a} = \binom{k}{a-1} \frac{(-1)^{a+1}}{k m^{a-1}}. \tag{A7}$$

If we now take the first $C-1$ terms, we have the following:

$$\begin{aligned} -f_{\mathbf{X}} \ln \frac{f_{\mathbf{X}}}{m} &= \sum_{k=1}^{C-1} \sum_{a=1}^{k+1} \tilde{c}_{k,a} f_{\mathbf{X}}^a + \cdots = \sum_{k=1}^{C-1} \sum_{a=1}^{C} \tilde{c}_{k,a} f_{\mathbf{X}}^a + \cdots = \sum_{a=1}^{C} \sum_{k=1}^{C-1} \tilde{c}_{k,a} f_{\mathbf{X}}^a + \cdots \\ &= \sum_{a=1}^{C} c_a^{\text{Taylor}} f_{\mathbf{X}}^a + \cdots \end{aligned} \tag{A8}$$

where we have the following:

$$c_1^{\text{Taylor}} = \sum_{k=1}^{C-1} \tilde{c}_{k,1} = H_{C-1}, \tag{A9}$$

https://doi.org/10.3390/e28040381

where H_k is the k-th Harmonic number. Furthermore, we have ($a \neq 1$):

$$\begin{aligned} c_a^{\text{Taylor}} &= \sum_{k=1}^{C-1} \tilde{c}_{k,a} = \sum_{k=1}^{C-1} \binom{k}{a-1} \frac{(-1)^{a+1}}{k m^{a-1}} = \sum_{k=a-1}^{C-1} \binom{k}{a-1} \frac{(-1)^{a+1}}{k m^{a-1}} \\ &= \frac{(-1)^{a+1}}{m^{a-1}} \frac{1}{a-1} \binom{C-1}{a-1}, \end{aligned} \tag{A10}$$

and the last equality follows from the following:

$$\sum_{k=a}^{C} \frac{1}{k} \binom{k}{a} = \sum_{\tilde{k}=0}^{C-a} \frac{1}{\tilde{k}+a} \binom{\tilde{k}+a}{a} = \sum_{\tilde{k}=0}^{C-a} \frac{1}{a} \binom{\tilde{k}+a-1}{a-1} = \frac{1}{a} \binom{C}{a}, \tag{A11}$$

where we used the binomial identities:

$$\frac{1}{a} \binom{a}{b} = \frac{1}{b} \binom{a-1}{b-1}, \quad a, b \neq 0, \tag{A12}$$

$$\sum_{j=0}^{m} \binom{n+j}{n} = \binom{n+m+1}{n+1} = \binom{n+m+1}{m}. \tag{A13}$$

Plugging the coefficients $\{c_a^{\text{Taylor}}\}$ into Corollary 1, we have the following:

$$\begin{aligned} h(\mathbf{X}) &\approx -\ln m + H_{C-1} \sum_{\substack{t_1+t_2+\cdots+t_q=1 \\ t_1,t_2,\ldots,t_q \geq 0}} \binom{1}{t_1, t_2, \ldots, t_q} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}) \\ &+ \sum_{a=2}^{C} \frac{(-1)^{a+1}}{m^{a-1}} \frac{1}{a-1} \binom{C-1}{a-1} \sum_{\substack{t_1+t_2+\cdots+t_q=a \\ t_1,t_2,\ldots,t_q \geq 0}} \binom{a}{t_1, t_2, \ldots, t_q} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}) \\ &= -\ln m + H_{C-1} + \sum_{a=1}^{C-1} \frac{(-1)^a}{m^a} \frac{1}{a} \binom{C-1}{a} \sum_{\substack{t_1+t_2+\cdots+t_q=a+1 \\ t_1,t_2,\ldots,t_q \geq 0}} \binom{a+1}{t_1, t_2, \ldots, t_q} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}) \\ &= -\ln m + H_{C-1} - m \sum_{a=1}^{C-1} \frac{B_{a+1}}{a} \binom{C-1}{a}, \end{aligned} \tag{A14}$$

where we used:

$$\sum_{\substack{t_1+t_2+\cdots+t_q=1 \\ t_1,t_2,\ldots,t_q \geq 0}} \binom{1}{t_1, t_2, \ldots, t_q} \left(\prod_{i=1}^{q} p_i^{t_i} \right) D(\hat{t}) = 1. \tag{A15}$$

Appendix D. Proof of Lemma 2

If we take the partial derivative of E w.r.t $d_{\mathcal{I},i}$, we get the following:

$$\begin{aligned} \frac{\partial E}{\partial d_{\mathcal{I},i}} &= \frac{-2}{b-a} \int_a^b w(s)(f(s) - \hat{f}(s)) s^i \mathrm{d}s \\ &= \frac{-2}{b-a} \left(\int_a^b w(s) f(s) s^i \mathrm{d}s - \sum_{j=1}^{C} d_{\mathcal{I},j} \int_a^b w(s) s^{i+j} \mathrm{d}s \right), \end{aligned} \tag{A16}$$

and setting it to zero, we get the following:

$$\sum_{j=1}^{C} d_{\mathcal{I},j} \int_a^b w(s) s^{i+j} \mathrm{d}s = \int_a^b w(s) f(s) s^i \mathrm{d}s, \tag{A17}$$

or written in matrix form as follows:

$$M_{\mathcal{I}}\vec{d}_{\mathcal{I}} = \vec{z}_{\mathcal{I}}. \tag{A18}$$

In order to show that our solution is a global minimum, we show that the Hessian matrix $H_{ij} = \dfrac{\partial E}{\partial d_{\mathcal{I},i}\partial d_{\mathcal{I},j}}$ is positive definite. Taking the partial derivatives, we have the following:

$$H_{ij} = \frac{2}{b-a}\int_a^b w(s)s^i s^j \mathrm{d}s, \tag{A19}$$

which is a Gram matrix (note that $w(s) > 0$ and a Gram matrix is positive semi-definite). It is straightforward to see that H is positive definite since $\{\sqrt{w(s)}s^i\}_{i=1}^{C}$ are linearly independent.

Appendix E. Proof of Corollary 2

From Lemma 3 and Equation (20), we have the following:

$$\begin{aligned}
&\sum_j (M_{\mathcal{I}})_{ij}(\vec{d}_{\mathcal{I}})_j = (\vec{z}_{\mathcal{I}})_i \Rightarrow \sum_j \frac{b^{i+j+r+1}}{i+j+r+1}(\vec{d}_{\mathcal{I}})_j = \frac{b^{i+r+2}}{(i+r+2)^2} - \frac{b^{i+r+2}\ln b}{i+r+2}\\
&\Rightarrow \sum_j \frac{b^{i+j+r+1}}{i+j+r+1}(\vec{d}_{\mathcal{I}})_j + \frac{b^{i+r+2}\ln b}{i+r+2} = \frac{b^{i+r+2}}{(i+r+2)^2}\\
&\Rightarrow \sum_{j\neq 1} \frac{b^{i+j+r+1}}{i+j+r+1}(\vec{d}_{\mathcal{I}})_j + \frac{b^{i+r+2}}{i+r+2}(\vec{d}_{\mathcal{I}})_1 + \frac{b^{i+r+2}\ln b}{i+r+2} = \frac{b^{i+r+2}}{(i+r+2)^2}\\
&\Rightarrow \sum_{j\neq 1} \frac{b^{j-1}}{i+j+r+1}(\vec{d}_{\mathcal{I}})_j + \frac{(\vec{d}_{\mathcal{I}})_1 + \ln b}{i+r+2} = \frac{1}{(i+r+2)^2}\\
&\Rightarrow \sum_{j\neq 1} \frac{b^{j-1}}{i+j+r+1}(\vec{d}_{\mathcal{I}})_j + \left(\frac{b^{j-1}(\vec{d}_{\mathcal{I}})_j + \ln b}{i+j+r+1}\right)\Bigg|_{j=1} = \frac{1}{(i+r+2)^2}\\
&\Rightarrow \sum_j \tilde{M}_{ij}\tilde{d}_j = \tilde{z}_i.
\end{aligned} \tag{A20}$$

Appendix F. Volume $V(s)$ for a Single Gaussian

We consider the trivial 'mixture' that consists of a single Gaussian centered on the origin, with spherical covariance matrix $\sigma^2\mathbf{1}$. The volume $V(s)$ as defined in (26) is then given by

$$V(s) = \int \mathrm{d}\mathbf{x}\, \delta\big(s - \mathcal{N}_{0,\sigma^2\mathbf{1}}(\mathbf{x})\big) \propto \int_0^\infty \mathrm{d}r\, r^{n-1}\delta\left(s - \frac{\exp -\frac{r^2}{2\sigma^2}}{(\sigma\sqrt{2\pi})^n}\right). \tag{A21}$$

We apply the rule $\delta(s - y(r)) = \frac{\delta(r - y^{\mathrm{inv}}(s))}{|y'(r)|}$ and obtain

$$\begin{aligned}
V(s) &\propto \int_0^\infty \mathrm{d}r\, r^{n-1}\frac{\delta\left(r - \sqrt{-2\sigma^2\ln[s(\sigma\sqrt{2\pi})^n]}\right)}{\frac{\exp -\frac{r^2}{2\sigma^2}}{(\sigma\sqrt{2\pi})^n}\cdot\frac{r}{\sigma^2}} \propto \frac{1}{s}\left(\sqrt{\ln\frac{1}{s(\sigma\sqrt{2\pi})^n}}\right)^{n-2}\\
&= \frac{1}{s}\left(\ln\frac{f_{\max}}{s}\right)^{\frac{n}{2}-1}.
\end{aligned} \tag{A22}$$

Here, we have used that $\frac{\exp -\frac{r^2}{2\sigma^2}}{(\sigma\sqrt{2\pi})^n} = s$ and that $f_{\max} = \frac{1}{(\sigma\sqrt{2\pi})^n}$.

https://doi.org/10.3390/e28040381

References

1. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
2. Wu, Y.; Chen, M.; Li, Z.; Wang, M.; Wei, Y. Theoretical insights for diffusion guidance: A case study for Gaussian mixture models. *arXiv* **2024**, arXiv:2403.01639. [CrossRef]
3. Guo, H.; Lu, C.; Bao, F.; Pang, T.; Yan, S.; Du, C.; Li, C. Gaussian mixture solvers for diffusion models. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 25598–25626.
4. Sulam, J.; Romano, Y.; Elad, M. Gaussian mixture diffusion. In *Proceedings of the 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*; IEEE: New York, NY, USA, 2016; pp. 1–5.
5. Raymond, N.; Iouchtchenko, D.; Roy, P.N.; Nooijen, M. A path integral methodology for obtaining thermodynamic properties of nonadiabatic systems using Gaussian mixture distributions. *J. Chem. Phys.* **2018**, *148*, 194110. [CrossRef]
6. Turan, N.; Böck, B.; Chan, K.J.; Fesl, B.; Burmeister, F.; Joham, M.; Fettweis, G.; Utschick, W. Wireless channel prediction via Gaussian mixture models. *arXiv* **2024**, arXiv:2402.08351. [CrossRef]
7. Qiu, X.; Jiang, T.; Wu, S.; Hayes, M. Physical layer authentication enhancement using a Gaussian mixture model. *IEEE Access* **2018**, *6*, 53583–53592. [CrossRef]
8. Parmar, A.; Shah, K.; Captain, K.; López-Benítez, M.; Patel, J. Gaussian mixture model based anomaly detection for defense against Byzantine attack in cooperative spectrum sensing. *IEEE Trans. Cogn. Commun. Netw.* **2023**, *10*, 499–509. [CrossRef]
9. McNicholas, P.D.; Murphy, T.B. Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics* **2010**, *26*, 2705–2712. [CrossRef]
10. Toh, H.; Horimoto, K. Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics* **2002**, *18*, 287–297. [CrossRef]
11. Zhu, H.; Guo, R.; Shen, J.; Liu, J.; Liu, C.; Xue, X.X.; Zhang, L.; Mao, S. The local dark matter kinematic substructure based on LAMOST K giants. *arXiv* **2024**, arXiv:2404.19655. [CrossRef]
12. Turner, W.; Martini, P.; Karaçaylı, N.G.; Aguilar, J.; Ahlen, S.; Brooks, D.; Claybaugh, T.; de la Macorra, A.; Dey, A.; Doel, P.; et al. New measurements of the Lyman-α forest continuum and effective optical depth with LyCAN and DESI Y1 data. *arXiv* **2024**, arXiv:2405.06743.
13. Landauer, R. Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.* **1961**, *5*, 183–191. [CrossRef]
14. Talukdar, S.; Bhaban, S.; Salapaka, M. Beating Landauer's bound by memory erasure using time-multiplexed potentials. *IFAC—PapersOnLine* **2017**, *50*, 7645–7650. [CrossRef]
15. Barzon, G.; Busiello, D.M.; Nicoletti, G. Excitation-inhibition balance controls information encoding in neural populations. *Phys. Rev. Lett.* **2025**, *134*, 068403. [CrossRef] [PubMed]
16. Nolan, O.; Stevens, T.S.W.; van Nierop, W.L.; van Sloun, R. Active diffusion subsampling. *arXiv* **2025**, arXiv:2406.14388.
17. Nielsen, F.; Nock, R. A series of maximum entropy upper bounds of the differential entropy. *arXiv* **2016**, arXiv:1612.02954. [CrossRef]
18. Michalowicz, J.V.; Nichols, J.M.; Bucholtz, F. Calculation of differential entropy for a mixed Gaussian distribution. *Entropy* **2008**, *10*, 200. [CrossRef]
19. Hershey, J.R.; Olsen, P.A. Approximating the Kullback–Leibler divergence between Gaussian mixture models. In *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2007; Volume 4, pp. IV-317–IV-320.
20. Goldberger, J.; Gordon, S.; Greenspan, H. An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*; IEEE: New York, NY, USA, 2003; pp. 487–493.
21. Huber, M.F.; Bailey, T.; Durrant-Whyte, H.; Hanebeck, U.D. On entropy approximation for Gaussian mixture random vectors. In *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*; IEEE: New York, NY, USA, 2008; pp. 181–188.
22. Melbourne, J.; Talukdar, S.; Bhaban, S.; Madiman, M.; Salapaka, M.V. The differential entropy of mixtures: New bounds and applications. *IEEE Trans. Inf. Theory* **2022**, *68*, 2123–2146. [CrossRef]
23. Friedman, J.H.; Tukey, J.W. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.* **1974**, *C-23*, 881–890. [CrossRef]
24. Tukey, P.A.; Tukey, J.W. Data-driven view selection, agglomeration, and sharpening. In *Interpreting Multivariate Data*; Wiley: Hoboken, NJ, USA, 1981; pp. 215–243.
25. Jones, M.C.; Sibson, R. What is projection pursuit? *J. R. Stat. Soc. Ser. A (Gen.)* **1987**, *150*, 1–18. [CrossRef]
26. Loperfido, N. Vector-valued skewness for model-based clustering. *Stat. Probab. Lett.* **2015**, *99*, 230–237. [CrossRef]
27. Loperfido, N. Finite mixtures, projection pursuit and tensor rank: A triangulation. *Adv. Data Anal. Classif.* **2019**, *13*, 145–173. [CrossRef]

28. Van Hulle, M.M. Edgeworth approximation of multivariate differential entropy. *Neural Comput.* **2005**, *17*, 1903–1910. [CrossRef] [PubMed]
29. Peña, D.; Prieto, F.J. The kurtosis coefficient and the linear discriminant function. *Stat. Probab. Lett.* **2000**, *49*, 257–261. [CrossRef]
30. Peña, D.; Prieto, F.J. Multivariate outlier detection and robust covariance matrix estimation. *Technometrics* **2001**, *43*, 286–310. [CrossRef]
31. Peña, D.; Prieto, F.J. Cluster identification using projections. *J. Am. Stat. Assoc.* **2001**, *96*, 1433–1445. [CrossRef]