Sampling from mixtures with negative weights: application to density approximation by Gaussian processes

Luca Martino*

* Università di Catania, Italy.

email: luca.martino@unict.it

Abstract

Mixtures of probability densities are widely used in statistics and machine learning. While classical mixtures restrict weights to be non-negative, allowing negative weights enables more flexible density approximation. However, negative weights introduce challenges in handling and sampling such distributions. For this purpose, we propose efficient Monte Carlo (MC) methods (including MC quadratures, rejection sampling and importance sampling schemes) for computing integrals and generating samples from these mixtures. A tailored proposal density ensures accurate and efficient generation of (unweighted) samples. Furthermore, we introduce an IS scheme which employs a mixture with negative coefficients as a proposal density, yielding samples with both positive and negative importance weights. Applications in Gaussian process-based density estimation demonstrate the practical relevance and efficiency of proposed schemes. Related Matlab is provided.

Keywords: Non-convex mixtures, mixtures with negative weights, Gaussian processes, rejection sampling, importance sampling

1 Introduction

Mixtures of probability densities are fundamental tools in statistics, signal processing, and machine learning [3]. A mixture model represents a probability distribution as a convex combination of simpler component distributions, such as Gaussians, exponentials, or Gamma probability density function (pdf), to name a few. Mixture models provide a powerful and flexible framework for modeling complex data [5, 17, 18].

While classical mixture models restrict weights to be non-negative, allowing negative weights opens new theoretical and practical possibilities. When weights can be negative, the resulting function may no longer be a proper probability density. In this work, we focus on the case where the mixture remains positive and proper. Mixtures with negative components are also referred to as non-convex or pseudo-convex mixtures [2, 7, 6]. In statistics and machine learning, mixtures with negative weights can be particularly useful for density approximation [11, 13, 29]. For example, Gaussian process (GP) regressors, often used for density estimation, can lead to expansions with

both positive and negative coefficients [13, 19, 20, 21]. In this context, negative weights can enable better approximation of sharp features, heavy tails, and periodic behaviors/patterns that may be difficult to capture with strictly non-negative mixtures. However, negative weights also introduce significant challenges. The resulting function may not always be a proper density, and classical sampling methods cannot be directly applied [8, 15, 23].

In this work, we describe several Monte Carlo quadrature and sampling methods involving mixtures with (possibly) negative coefficients (Mix-NCs). First, in Section 3, we focus on the efficient computation of integrals involving non-convex mixtures. Second, in Section 4, we propose an efficient proposal density to be used within rejection sampling (RS) and/or importance sampling with resampling (IS+R) schemes. In both cases, we obtain (unweighted) samples (exactly in RS, or asymptotically in IS+R) that are distributed according to the target mixture with negative weights. The proposal density introduced here ensures good performance in both RS and importance sampling (IS) schemes, as it is itself a "piece" of the target density. Last but not least, in Section 5 we describe how to use a mixture with negative coefficients as a proposal density within an IS scheme. In this setting, some of the generated samples carry negative importance weights. As a result, the proposed procedure admits a physical analogy in which samples with positive importance weights correspond to "matter", while samples with negative importance weights correspond to "anti-matter" [24, 25]. We also describe in Section 6 the application of these methods to GP-based density approximation. Theoretical discussions are also provided. The numerical simulations given in Section 7 demonstrate the efficiency and accuracy of the proposed techniques. Related code is also provided.[1]

2 Framework and main notation

Let consider a finite mixture of densities, $\phi_n(\mathbf{x})$, with potentially negative associated weights, i.e.,

$$\bar{p}(\mathbf{x}) \propto p(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x}), \tag{1}$$

$$= p_+(\mathbf{x}) + p_-(\mathbf{x}), \tag{2}$$

$$= \sum_{i=1}^{M} \alpha_i^+ \phi_i(\mathbf{x}) + \sum_{k=1}^{N-M} \alpha_k^- \phi_k(\mathbf{x}) \geq 0, \qquad \forall \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_X}, \tag{3}$$

where $\alpha_i^+ > 0$ and $\alpha_i^- < 0$. We have also set $p_+(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i^+ \phi_i(\mathbf{x}) \geq 0$ and $p_-(\mathbf{x}) = \sum_{k=1}^{N-M} \alpha_k^- \phi_k(\mathbf{x}) \leq 0$. Clearly, we have

$$\alpha_1 = \alpha_1^+ > 0, ..., \alpha_M = \alpha_M^+ > 0, \quad \alpha_{M+1} = \alpha_1^- < 0, ..., \alpha_N = \alpha_{N-M}^- < 0.$$

Without loss of generality, we are assuming that the components are ordered: the first M components are associated to positive weights, α_i^+, and the rest of $N - M$ components have assigned to the negative weights, α_i^-. Additional assumptions are:

[1] Related Matlab code is given at `http://www.lucamartino.altervista.org/public_code_NegMix2025.zip`.

(a) $\phi_n(\mathbf{x}) \geq 0$ and $\int_{\mathcal{X}} \phi_n(\mathbf{x}) d\mathbf{x} = 1$, for all n.

(b) We can evaluate and we can draw samples from each component $\phi_n(\mathbf{x})$.

Given the assumptions above, and since we consider a proper/normalized mixture density, i.e., $\bar{p}(\mathbf{x}) \geq 0$ and $\int_{\mathcal{X}} \bar{p}(\mathbf{x}) d\mathbf{x} = 1$, we can write

$$\bar{p}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} = \frac{\sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} = \sum_{n=1}^{N} \bar{\alpha}_n \phi_n(\mathbf{x}), \tag{4}$$

where we have defined

$$\bar{\alpha}_n = \frac{\alpha_n}{\sum_{j=1}^{N} \alpha_j}, \qquad n = 1, ..., N. \tag{5}$$

Note that:

- $\sum_{n=1}^{N} \bar{\alpha}_n = 1$ (since $\int_{\mathcal{X}} \bar{p}(\mathbf{x}) d\mathbf{x} = 1$ and $\int_{\mathcal{X}} \phi_n(\mathbf{x}) d\mathbf{x} = 1$ for all n),
- even if we have $\bar{\alpha}_n > 0$ for $n = 1, ..., M$,
- and $\bar{\alpha}_n < 0$ for $n = M+1, ..., N$.

Hence, Eq. (4) is *not* a convex combination of $\phi_n(\mathbf{x})$. Thus, we also have the condition $\sum_{i=1}^{M} \alpha_i^+ > \sum_{k=1}^{N-M} \alpha_k^-$, since we need $\sum_{n=1}^{N} \alpha_n > 0$ to have $p(\mathbf{x}) \geq 0$. We can also define the two *partial-mixtures*,

$$\bar{p}_+(\mathbf{x}) = \frac{p_+(\mathbf{x})}{\sum_{i=1}^{M} \alpha_i^+} = \sum_{m=1}^{M} \bar{\alpha}_m^+ \phi_m(\mathbf{x}), \quad \bar{p}_-(\mathbf{x}) = \frac{p_-(\mathbf{x})}{\sum_{i=1}^{N-M} \alpha_i^-} = \sum_{k=1}^{M} \bar{\alpha}_k^- \phi_k(\mathbf{x}), \tag{6}$$

where

$$\bar{\alpha}_m^+ = \frac{\alpha_m^+}{\sum_{i=1}^{M} \alpha_i^+}, \qquad \bar{\alpha}_k^- = \frac{\alpha_k^-}{\sum_{j=1}^{N-M} \alpha_j^-}, \tag{7}$$

Note that:

- $\bar{p}_+(\mathbf{x}) \geq 0$ and $\bar{p}_-(\mathbf{x}) \geq 0$ (despite $\alpha_j^- < 0$ and $p_-(\mathbf{x}) \leq 0$),
- and they are both proper classical mixtures with non-negative weights, i.e., $\bar{\alpha}_m^+ \geq 0$, $\bar{\alpha}_k^- \geq 0$ and $\sum_{m=1}^{M} \bar{\alpha}_m^+ = 1$, $\sum_{k=1}^{N-M} \bar{\alpha}_k^- = 1$ (despite $\alpha_j^- < 0$).

Finally, we can also write the complete mixture $\bar{p}(\mathbf{x})$, in Eq. (4), as function of the partial mixtures $\bar{p}_+(\mathbf{x})$ and $\bar{p}_-(\mathbf{x})$ in Eq. (6), i.e.,

$$\begin{aligned}
\bar{p}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sum_{j=1}^{N}\alpha_j} &= \frac{p_+(\mathbf{x})}{\sum_{j=1}^{N}\alpha_j} + \frac{p_-(\mathbf{x})}{\sum_{j=1}^{N}\alpha_j}, \\
&= \frac{\sum_{i=1}^{M}\alpha_i^+}{\sum_{i=1}^{M}\alpha_i^+}\frac{p_+(\mathbf{x})}{\sum_{j=1}^{N}\alpha_j} + \frac{\sum_{i=1}^{N-M}\alpha_i^-}{\sum_{i=1}^{N-M}\alpha_i^-}\frac{p_-(\mathbf{x})}{\sum_{j=1}^{N}\alpha_j}, \\
&= \frac{\sum_{i=1}^{M}\alpha_i^+}{\sum_{j=1}^{N}\alpha_j}\bar{p}_+(\mathbf{x}) + \frac{\sum_{i=1}^{N-M}\alpha_i^-}{\sum_{j=1}^{N}\alpha_j}\bar{p}_-(\mathbf{x}) \\
&= \underbrace{\beta^+}_{>1}\bar{p}_+(\mathbf{x}) + \underbrace{(1-\beta^+)}_{<0}\bar{p}_-(\mathbf{x}),
\end{aligned} \tag{8}$$

where

$$\beta^+ = \frac{\sum_{i=1}^{M}\alpha_i^+}{\sum_{j=1}^{N}\alpha_j} > 1. \tag{9}$$

Therefore, even if the sum of two weights β^+ and $1-\beta^+$ is one, the linear combination $\beta^+\bar{p}_+(\mathbf{x}) + (1-\beta^+)\bar{p}_-(\mathbf{x})$ is not a convex combination. Table 1 summarizes the main notation of the work.

Remark. Although we have assumed that $p(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$, the schemes proposed in this work remain valid even when $p(\mathbf{x}) < 0$ for some $\mathbf{x}$. In such cases, the methods can be applied to a modified function defined as $\tilde{p}(\mathbf{x}) = \max[p(\mathbf{x}), 0]$. This transformation ensures non-negativity while preserving the essential structure of the original function in regions where $p(\mathbf{x})$ is positive. Moreover, we will highlight later that, within adaptive procedures, as the number of components N increases and the hyper-parameters are progressively optimized, the approximation tends to enforce the condition $p(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$. In this asymptotic setting, the need for truncation vanishes, and the original formulation becomes directly applicable without modification.

3 Quadratures for integral approximations involving Mix-NCs

The best procedure for approximating integral involving to a mixture $\bar{p}(\mathbf{x})$ of densities with possibly negative weights (Mix-NCs) is related to a quadrature trick [12, 20]. Indeed, if we are

Table 1: Main notation of the work.

Notation	Description
$p(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n \phi_n(\mathbf{x})$	Unnormalized Mix-NCs
$\bar{p}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sum_{j=1}^{N} \alpha_j} = \sum_{n=1}^{N} \bar{\alpha}_n \phi_n(\mathbf{x})$	Normalized Mix-NCs
$\alpha_n = \alpha_n^+ > 0, \quad n = 1, ..., M$ $\alpha_n = \alpha_n^- < 0, \quad n = N+M, ..., N$	Unnormalized coefficients of the complete Mix-NCs
$\bar{\alpha}_n = \frac{\alpha_n}{\sum_{j=1}^{N} \alpha_j}$	Normalized coefficients of the complete Mix-NCs
$p_+(\mathbf{x}) = \sum_{n=1}^{M} \alpha_n^+ \phi_n(\mathbf{x})$	Unnormalized positive partial mixture
$p_-(\mathbf{x}) = \sum_{n=1}^{M} \alpha_n^- \phi_n(\mathbf{x})$	Unnormalized negative partial mixture
$\bar{p}_+(\mathbf{x}) = \frac{p_+(\mathbf{x})}{\sum_{i=1}^{M} \alpha_i^+} = \sum_{m=1}^{M} \bar{\alpha}_m^+ \phi_m(\mathbf{x})$	Normalized negative partial mixture
$\bar{p}_-(\mathbf{x}) = \frac{p_-(\mathbf{x})}{\sum_{i=1}^{N-M} \alpha_i^-} = \sum_{m=1}^{M} \bar{\alpha}_k^- \phi_k(\mathbf{x})$	Normalized negative partial mixture
$\bar{p}(\mathbf{x}) = \beta^+ \bar{p}_+(\mathbf{x}) + (1-\beta^+)\bar{p}_-(\mathbf{x})$	Normalized Mix-NCs as function of the partial mixtures
$\beta^+ = \frac{\sum_{i=1}^{M} \alpha_i^+}{\sum_{j=1}^{N} \alpha_j} > 1$	Coefficient of the partial mixtures combination

For any generic variables $\bar{\psi}_k$ or functions $\bar{\psi}(\mathbf{x})$ denoted with an *upper bar*, the following conditions hold:

$$\sum_k^K \bar{\psi}_n = 1, \qquad \int_{\mathcal{X}} \bar{\psi}(\mathbf{x}) d\mathbf{x} = 1.$$

interested to approximate a generic moment or any integral involving to the distribution $\bar{p}(\mathbf{x})$, i.e.,

$$I_{\bar{p}} = \mathbb{E}_{\bar{p}}[f(\mathbf{x})] = \int_{\mathcal{X}} f(\mathbf{x})\bar{p}(\mathbf{x})d\mathbf{x}, \tag{10}$$

$$= \frac{1}{\sum_{j=1}^{N}\alpha_j}\sum_{n=1}^{N}\alpha_n \int_{\mathcal{X}} f(\mathbf{x})\phi_n(\mathbf{x})d\mathbf{x}, \tag{11}$$

$$= \sum_{n=1}^{N}\bar{\alpha}_n J_n. \tag{12}$$

where $f(\mathbf{x})$ is a generic integrable function. Note that above we have set $J_n = \int_{\mathcal{X}} f(\mathbf{x})\phi_n(\mathbf{x})d\mathbf{x}$ and $\bar{\alpha}_n = \frac{\alpha_n}{\sum_{j=1}^{N}\alpha_j}$. Since we are able to draw from $\phi_n(\mathbf{x})$, we can approximate each J_n by a simple Monte Carlo procedure (i.e., a stochastic quadrature rule) [15],

$$\widehat{J}_n = \frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x}_n^{(s)}), \qquad \mathbf{x}_n^{(s)} \sim \phi_n(\mathbf{x}). \tag{13}$$

Therefore, replacing in the expressions above into $I = \sum_{n=1}^{N}\bar{\alpha}_n J_n$, we obtain the final estimator:

$$\widehat{I}_{\bar{p}} = \sum_{n=1}^{N}\bar{\alpha}_n \widehat{J}_n = \frac{1}{S}\sum_{n=1}^{N}\sum_{s=1}^{S}\bar{\alpha}_n f(\mathbf{x}_n^{(s)}). \tag{14}$$

As $S \to \infty$, we have $\widehat{I}_{\bar{p}} \to I_{\bar{p}}$ [8, 15, 23]. Recall that $\bar{\alpha}_n > 0$ for $n = 1, ..., M$ and $\bar{\alpha}_n < 0$ for $n = M+1, ..., N$. The consistency of $\widehat{I}_{\bar{p}}$ is ensured as $S \to \infty$ by standard MC arguments, since $\widehat{J}_n \to J_n$ and, as a consequence, $\widehat{I}_{\bar{p}} \to I$ [1]. For further details about other convergence properties, involving also N, see [12, Section 7].

4 Sample generation from Mix-NCs

In this section, we discuss two sampling schemes: a rejection sampling method and an importance sampling technique. Note that we are able to generate samples from

$$\bar{p}_+(\mathbf{x}) = \frac{p_+(\mathbf{x})}{\sum_{i=1}^{M}\alpha_i^+} = \sum_{m=1}^{M}\bar{\alpha}_m^+\phi_m(\mathbf{x}), \qquad \bar{\alpha}_m^+ = \frac{\alpha_m^+}{\sum_{i=1}^{M}\alpha_i^+}, \tag{15}$$

in a classical way, since $0 \leq \bar{\alpha}_m^+ \leq 1$ and $\sum_{m=1}^{M}\bar{\alpha}_m^+ = 1$. Moreover, by construction,

$$p_+(\mathbf{x}) \geq p(\mathbf{x}), \tag{16}$$

that is the inequality required for applying the RS technique [15, Chapter 3]. Furthermore, $\bar{p}_+(\mathbf{x})$ represents a part (a "piece") of the target density $p(\mathbf{x})$. Namely, a generic proposal density $q(\mathbf{x})$ must satisfy the inequality $q(\mathbf{x}) \geq p(\mathbf{x})$ in order to apply rejection sampling correctly. If this

condition is violated, the samples generated by the algorithm are no longer distributed according to the target density $p(\mathbf{x})$, but instead follow $\tilde{p}(\mathbf{x}) \propto \min\{p(\mathbf{x}), q(\mathbf{x})\}$ [15, Chapter 3]. This observation highlights the relevance and ability of constructing proposals that satisfy the bound (16). In particular, the proposed choice $q(\mathbf{x}) = p_+(\mathbf{x})$ automatically fulfills this requirement. Moreover, $p_+(\mathbf{x})$ is typically an efficient proposal within an RS scheme, since it is assembled directly from components of the target density $p(\mathbf{x})$, thereby preserving its main structural features and generally remaining close to it in shape. Hence, $\bar{p}_+(\mathbf{x})$ constitutes an appropriate choice of proposal density in Monte Carlo methods, specially in a RS technique. Below we outline the proposed RS scheme.

Rejection sampling (RS) for Mix-NCs:

1. Set $s = 1$.
2. Draw a candidate $\mathbf{z}' \sim \bar{p}_+(\mathbf{x})$,
3. With probability
$$p_A(\mathbf{z}') = \frac{p(\mathbf{z}')}{p_+(\mathbf{z}')}, \tag{17}$$
set $\mathbf{x}^{(s)} = \mathbf{z}'$ and increase $s \leftarrow s + 1$. Otherwise, with prob. $1 - p_A(\mathbf{z}')$ discard $\mathbf{z}'$.
4. if $s \leq S$, repeat from step 2.

Note that $p_A \in [0, 1]$. The algorithm provides exact samples from $\bar{p}(\mathbf{x})$ and its validity is ensured by the inequality (16) [15, Chapter 3]. The acceptance rate A_r is:

$$A_r = \int_{\mathcal{X}} \frac{p(\mathbf{x})}{p_+(\mathbf{x})} \bar{p}_+(\mathbf{x}) d\mathbf{x} = \frac{1}{\sum_{i=1}^{M} \alpha_i^+} \int_{\mathcal{X}} p(\mathbf{x}) d\mathbf{x}, \tag{18}$$

$$= \frac{\sum_{n=1}^{N} \alpha_n}{\sum_{i=1}^{M} \alpha_i^+}, \tag{19}$$

$$= 1 - \frac{\sum_{k=1}^{N-M} \alpha_k^-}{\sum_{i=1}^{M} \alpha_i^+} = 1 - \rho, \tag{20}$$

where we set $\rho = \frac{\sum_{k=1}^{N-M} \alpha_k^-}{\sum_{i=1}^{M} \alpha_i^+}$. If $\rho \to 0$ then $A_r \to 1$ and we have a perfect sampler [15, 23]. Hence, the acceptance rate A_r is close to 1 if the sum of negative weights is close to zero, or the sum of positive weights is much larger than the sum of negative weights. This is particularly interesting for the Gaussian process application: in a GP approximation of a density, the number of negative weights should tend to disappear as the number of points in the regression grows, and the hyper-parameters are updated and optimized. The corresponding importance sampling (IS) with resampling (IS+R) scheme is described below.

Importance sampling with resampling (IS+R) for Mix-NCs:

1. Draw $\mathbf{z}_1, ..., \mathbf{z}_S \sim \bar{p}_+(\mathbf{x})$,

2. Assign the weight

$$w_s = \frac{p(\mathbf{z}_s)}{\bar{p}_+(\mathbf{z}_s)} = \left[\sum_{i=1}^{M} \alpha_i^+\right] p_A(\mathbf{z}_s), \quad s = 1, ..., S, \tag{21}$$

   where we have used $p_A(\mathbf{z})$ given in Eq. (17).

3. Define the normalized weights

$$\bar{w}_s = \frac{w_s}{\sum_{i=1}^{S} w_i}, \qquad s = 1, ..., S. \tag{22}$$

4. Resample S times within the set $\{\mathbf{z}_1, ..., \mathbf{z}_S\}$ according to the probability mass function defined by the normalized weights $\bar{w}_s$, with $s = 1, ..., S$, obtaining a new set of unweighted samples $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(S)}\}$.

Unlike RS, the IS+R does not return exact samples, but provides samples $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(S)}\}$ that are approximately and asymptotically distributed as $\bar{p}(\mathbf{x})$. However, the quality of this approximation improves as S grows [8, 9, 23]. Furthermore, no samples are rejected/discarded as in the RS scheme. It is also important to remark that the weights w_s present good theoretically properties due to the choice of the proposal density. For instance, since $w_s \propto p_A(\mathbf{z}_s)$, the IS weights are bounded

$$w_s \in \left[0, \sum_{i=1}^{M} \alpha_i^+\right], \tag{23}$$

hence their distribution has not heavy tails, and the variance of the weights is always bounded [4, 11, 26, 28]. Standard IS can produce highly variable estimates, especially when weights have heavy right tails [27]. Namely, extreme values of the weights lead to unstable estimates [26, 27] or yield estimators with infinite variance (see numerical experiments in [10]). However, these undesirable scenarios cannot occur in the proposed IS scheme, due to the property in Eq. (23). Furthermore, note that

$$\bar{w}_s = \frac{w_s}{\sum_{i=1}^{S} w_i} = \frac{p_A(\mathbf{z}_s)}{\sum_{i=1}^{S} p_A(\mathbf{z}_i)}.$$

Recall that the proposal density, described in this work, provides good performance within RS and IS schemes since it is itself a piece of the target density [9].

Histograms. Histograms can be constructed in the usual way using the unweighted samples, that are generated by the RS or by IS+R schemes as well. Alternatively, one can directly use the

weighted samples $\{\mathbf{z}_s, w_s\}_{s=1}^S$ obtained by the IS procedure. Indeed, in each bin of the histogram, instead of considering the number of samples inside the bin, we can sum the corresponding weights. Namely, let consider the bin $\mathcal{B} \subset \mathcal{X}$. The value of the histogram in the bin must be $\sum_{\mathbf{z}_j \in \mathcal{B}} w_j$, where we sum each w_j with j such that $\mathbf{z}_j \in \mathcal{B}$ (for the underlying theory see [14]). The histogram can be normalized dividing all the values by the complete sum of the weights, i.e., $\sum_{s=1}^S w_s$.

5 Mix-NCs as proposal density within IS methods

In many applications, a good approximation of a distribution is needed. For instance, let us consider a target-posterior distribution $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ where $Z = \int_{\mathcal{X}} \pi(\mathbf{x})d\mathbf{x}$. In order to extract information about the posterior $\bar{\pi}(\mathbf{x})$, often we are interested in computing integrals which generally involve the product of a generic function f and the posterior $\bar{\pi}$,

$$I_{\bar{\pi}} = \mathbb{E}_{\bar{\pi}}[f(\mathbf{x})] = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} = \frac{1}{Z}\int_{\mathcal{X}} f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}. \tag{24}$$

Note that the expectation above $\mathbb{E}_{\bar{\pi}}[f(\mathbf{x})]$ is different from the expectation $\mathbb{E}_{\bar{p}}[f(\mathbf{x})]$ given in Eq. (10), since in Eq. (24) the density involved is the posterior $\bar{\pi}$ instead of the mixture $\bar{p}$, i.e., $I_{\bar{\pi}} = \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}$. Note that we can write:

$$\begin{aligned}
I_{\bar{\pi}} = \mathbb{E}_{\bar{\pi}}[f(\mathbf{x})] &= \int_{\mathcal{X}} f(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}, && (25)\\
&= \int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\bar{p}(\mathbf{x})d\mathbf{x},\\
&= \int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\left(\beta^+\bar{p}_+(\mathbf{x}) + (1-\beta^+)\bar{p}_-(\mathbf{x})\right)d\mathbf{x},\\
&= \beta^+\int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\bar{p}_+(\mathbf{x})d\mathbf{x} + (1-\beta^+)\int_{\mathcal{X}} \frac{f(\mathbf{x})\bar{\pi}(\mathbf{x})}{\bar{p}(\mathbf{x})}\bar{p}_-(\mathbf{x})d\mathbf{x},\\
&= \beta^+\mathbb{E}_{\bar{p}_+}[f(\mathbf{x})\bar{\pi}(\mathbf{x})] + (1-\beta^+)\mathbb{E}_{\bar{p}_-}[f(\mathbf{x})\bar{\pi}(\mathbf{x})], && (26)\\
&= \frac{\beta^+}{Z}\mathbb{E}_{\bar{p}_+}[f(\mathbf{x})\pi(\mathbf{x})] + \frac{1-\beta^+}{Z}\mathbb{E}_{\bar{p}_-}[f(\mathbf{x})\pi(\mathbf{x})], && (27)
\end{aligned}$$

The expression (26) induces the design of an importance sampling scheme with positive and negative IS weights. The idea is to apply the MC approach to approximate the expectations $\mathbb{E}_{\bar{p}_+}[f(\mathbf{x})\pi(\mathbf{x})]$ and $\mathbb{E}_{\bar{p}_-}[f(\mathbf{x})\pi(\mathbf{x})]$, as shown below.

Importance sampling with an Mix-NCs as proposal density:

1. Draw S samples from $\bar{p}_+(\mathbf{x})$ and S samples from $\bar{p}_-(\mathbf{x})$, i.e.,

$$\mathbf{x}_s^+ \sim \bar{p}_+(\mathbf{x}), \qquad \mathbf{x}_s^- \sim \bar{p}_-(\mathbf{x}), \qquad s = 1, ..., S. \tag{28}$$

2. To each sample, assign the weights

$$w_s^+ = \beta^+ \frac{\pi(\mathbf{x}_s^+)}{\bar{p}(\mathbf{x}_s^+)} \geq 0, \qquad w_s^- = (1-\beta^+) \frac{\pi(\mathbf{x}_s^-)}{\bar{p}(\mathbf{x}_s^-)} \leq 0, \tag{29}$$

   with $s = 1, ..., S$. Note that, both denominators of weights in Eq. (29) contain the complete mixture $\bar{p}(\mathbf{x})$.

3. **If Z is known**, the resulting estimator is given by the formula:

$$\widehat{I}_{\bar{\pi}} = \frac{1}{SZ} \left(\sum_{j=1}^{S} w_j^+ f(\mathbf{x}_j^+) + \sum_{j=1}^{S} w_j^- f(\mathbf{x}_j^-) \right). \tag{30}$$

4. **If Z is unknown**, estimate and replace Z above with:

$$\widehat{Z} = \frac{1}{S} \left(\sum_{j=1}^{S} w_j^+ + \sum_{j=1}^{S} w_j^- \right). \tag{31}$$

The estimator in Eq. (31) is based on the following equality:

$$\begin{aligned} Z &= \beta^+ \int_{\mathcal{X}} \frac{\pi(\mathbf{x})}{\bar{p}(\mathbf{x})} \bar{p}_+(\mathbf{x}) d\mathbf{x} + (1-\beta^+) \int_{\mathcal{X}} \frac{\pi(\mathbf{x})}{\bar{p}(\mathbf{x})} \bar{p}_-(\mathbf{x}) d\mathbf{x}, \\ &= \beta^+ \mathbb{E}_{\bar{p}_+}[\pi(\mathbf{x})] + (1-\beta^+) \mathbb{E}_{\bar{p}_-}[\pi(\mathbf{x})]. \end{aligned} \tag{32}$$

Note that the samples $\mathbf{x}_s^-$ generated from $\bar{p}_-(\mathbf{x})$ have associated negative weights w_s^-. Thus, the above procedure admits a physical analogy, wherein the samples $\mathbf{x}_s^+$ correspond to "matter" samples, and the samples $\mathbf{x}_s^-$ correspond to "anti-matter" samples, or any other similar physical analogy involving negative-signed particles [24, 25].

Remark. In this designed IS scheme, we draw samples from both partial mixtures, $\bar{p}_+(\mathbf{x})$ and $\bar{p}_-(\mathbf{x})$. The samples drawn from $\bar{p}_-(\mathbf{x})$ (denoted $\mathbf{x}_s^-$) carry negative importance weights.

6 Applications to GP approximation of a density

6.1 Emulation of a density

In many applications, e.g., in some adaptive Monte Carlo schemes, there is the need of constructing (via regression) a non-parametric density (a.k.a., emulator/surrogate model), which mimics a

posterior distribution [11, 13, 20]. More precisely, let us consider again a target-posterior distribution $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$. Related to this target density, we have a (possibly noisy) set of points $\{\mathbf{x}_n, t_n\}$ with $n = 1, ..., N$, where $t_n = \pi(\mathbf{x}_n)$ or $t_n = t(\mathbf{x}_n) = \pi(\mathbf{x}_n) + \epsilon_n$ (where ϵ_n is a noise perturbation) but always we have $t_n \geq 0$ [11, 13]. More specifically, in many practical cases, we have access only to a noisy realization $t(\mathbf{x})$ related to $\pi(\mathbf{x})$. Namely, fixing $\mathbf{x}$, $t(\mathbf{x})$ is a random variable with mean $\mathbb{E}[t(\mathbf{x})] = m(\mathbf{x}) = \pi(\mathbf{x}) - \mu(\mathbf{x})$, and variance $\text{var}[t(\mathbf{x})] = s^2(\mathbf{x})$, where $\mu(\mathbf{x})$ is a bias. For more details, see [13].

Emulation versus estimation of a density. GP-based emulation is a *supervised* approach: an unnormalized version of a density is evaluated at selected input locations, and a GP regressor is trained to approximate this function over the domain. This framework is particularly well suited to scenarios in which evaluations of the target density are noisy or costly (i.e., expensive) [13, 20]. In contrast, kernel density estimation (KDE) is an *unsupervised* technique that constructs a density estimate directly from samples drawn from the unknown distribution, without requiring point-wise evaluations of the density itself. In this work, we are focusing in the first setting, i.e., density emulation.

6.2 Gaussian processes (GPs) for regression

Let us assume that we apply a Gaussian process (GP) regression model [22]. We consider a kernel function, $k(\mathbf{x}, \mathbf{z}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, then we can define a $N \times N$ kernel matrix $\mathbf{K}$ where each entry is $[\mathbf{K}]_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$, and a $N \times 1$ kernel vector $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_N)]^\top$. For the purpose of density approximation, it is more appropriate to consider kernel functions $k(\mathbf{x}, \mathbf{z})$ that are themselves normalized probability densities [12]. For simplicity and as an example, we assume Gaussian kernels,

$$k(\mathbf{x}, \mathbf{z}) = \left(\frac{1}{2\pi\lambda^2}\right)^{\frac{d_X}{2}} \exp\left(-\frac{||\mathbf{x} - \mathbf{z}||^2}{2\lambda^2}\right), \quad \lambda > 0. \tag{33}$$

Defining also the vector $\mathbf{t} = [t_1, ..., t_J]^\top$, the density approximation is given by the formulas:

$$\bar{p}(\mathbf{x}) \propto p(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \eta \mathbf{I}_J)^{-1} \mathbf{t}, \qquad \eta \geq 0, \tag{34}$$

$$= \mathbf{k}(\mathbf{x})^\top \boldsymbol{\alpha}, \tag{35}$$

$$= \sum_{n=1}^{N} \alpha_n \underbrace{k(\mathbf{x}, \mathbf{x}_n)}_{\phi_n(\mathbf{x})}, \tag{36}$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^\top$ is defined as

$$\boldsymbol{\alpha} = (\mathbf{K} + \eta \mathbf{I}_J)^{-1} \mathbf{t}, \tag{37}$$

and the values α_n play the same role of the coefficients in Eq. (2), and the kernel functions play the role of densities in Eq. (2), i.e., $\phi_n(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}_n)$.[2]

[2]Clearly, the positive and negative values of α_n provided in Eq. (37) are not ordered, as assumed in Section 2. Nevertheless, this does not affect the validity or applicability of the proposed methodologies.

6.3 Choice of the hyper-parameters

We can assume that a vector of hyper-parameters $[\lambda, \eta]$ is chosen such that $p(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$; that is, it is always possible to find such hyper-parameters, although they may be far from optimal. For example, decreasing the value of λ tends to ensure that $p(\mathbf{x}) \geq 0$ for all $\mathbf{x}$, but an excessively small λ may result in a poor approximation of $\pi(\mathbf{x})$. The negativity of $p(\mathbf{x})$ can be also avoided just defining $\widetilde{p}(\mathbf{x}) = \max[p(\mathbf{x}), 0]$ [11]. However, this issue tends to disappear as N grows and the hyper-parameters are optimized.
An optimal approach is to update the hyper-parameters $[\lambda, \eta]$ by adopting a suitable optimization criterion, such as classical marginal likelihood maximization [10, 16]. For the bandwidth parameter, a faster alternative is to rely on heuristic "rules of thumb", for example setting $\lambda \approx 1/N^{1/d_X}$, or choosing λ as the median pairwise distance among the input points. Specifically, given $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ for all i, j, one may set $\lambda \approx \text{median}\,(D_{ij})$. In any case, the number of negative values of α_n's should decrease as the number N of acquired data, $\{\mathbf{x}_n, t_n\}_{n=1}^N$, grows.
However, for a finite N, we can have negative values of α_n. Hence, the methods proposed in Sections 3, 4 and 5 are required for a suitable application of the emulator.

6.4 Guidelines for practical applications of the proposed schemes

The proposed approaches have been designed to be applied in two different regimes, depending on the quality of the emulator. If the emulator is sufficiently close to $\pi(\mathbf{x})$, that is, $p(\mathbf{x}) \approx \pi(\mathbf{x})$, the proposed method can be used to approximate integrals (e.g., moments) involving $\bar{p}(\mathbf{x})$. In this setting, samples are drawn from $\bar{p}(\mathbf{x})$ using the schemes described in Sections 3-4, where the positive partial mixture $\bar{p}_+(\mathbf{x})$ is employed as the proposal and $\bar{p}(\mathbf{x})$ acts as the target pdf. Since $p(\mathbf{x}) \approx \pi(\mathbf{x})$, this approach also enables the extraction of information from $\bar{\pi}(\mathbf{x})$ without requiring additional evaluations of $\bar{\pi}(\mathbf{x})$. In other words, when the emulator $p(\mathbf{x})$ closely approximates the (unnormalized) target density $\pi(\mathbf{x})$, we can directly extract information from the complete mixture $\bar{p}(\mathbf{x})$ that is effectively equivalent to extracting information about the true target $\bar{\pi}(\mathbf{x})$. This can be done "for free," in the sense that no additional evaluations of the costly target density are required.

Conversely, when the emulator is less accurate, a more robust strategy should be used instead. Namely, when the emulator $p(\mathbf{x})$ is not sufficiently close to $\pi(\mathbf{x})$, we can instead adopt the IS scheme of Section 4. Here, the full emulator $\bar{p}(\mathbf{x})$ acts as the proposal and the true target $\bar{\pi}(\mathbf{x})$ is evaluated explicitly. In this case, additional evaluations of $\pi(\mathbf{x})$ are required, resulting in an increased computational cost. However, this extra cost can be leveraged to adaptively refine the emulator, leading to progressively more efficient sampling in subsequent iterations [11, 12, 13]. Note that the IS scheme of Section 4 may also be employed in the first scenario described above, although this would entail additional evaluations of π.

6.5 Scalability and application to high-dimensional spaces

All the proposed methodologies do not present any particular scalability issues, nor do they suffer from intrinsic limitations in high-dimensional spaces, if the second assumption regarding the basis

functions $\phi_n(\mathbf{x})$ holds, namely, that each component $\phi_n(\mathbf{x})$ can be efficiently evaluated and drawn from.
In contrast, Gaussian Process (GP) emulation suffers from well-known scalability limitations as the number of training points N increases, since the inversion of the kernel matrix $\mathbf{K}$ entails a computational cost that grows cubically with N. As a consequence, GP-based emulators often rely on parsimonious or adaptive strategies (such as sparse approximations, inducing points, or sequential design) to limit the number of regression points while maintaining predictive accuracy [11, 12]. Although these techniques can substantially alleviate the computational burden, they introduce additional modeling and algorithmic complexity. A detailed discussion of such strategies is beyond the scope of this work. The interested reader is referred to [13, 22] for representative solutions and further developments.

7 Numerical simulations

In this section, we first test the proposed RS technique described in Section 4 and then, in the second section, we test the IS scheme introduced in Section 5.[3]

7.1 RS for drawing samples from the emulator

Let us consider that we desire to emulate (i.e., approximate by regression) the following multi-modal target density

$$\bar{\pi}(x) \propto \pi(x) = \sin(x)^2 \exp\left(-\frac{x^2}{30}\right),$$

that is shown in solid line in Figure 1(a). We consider three scenarios.

Scenario 1: We consider $N = 9$ points in the regression, more precisely,

$$x_i \in \{-5, -4, -1, 0, 0.5, 1, 2, 5, 10\}, \tag{38}$$

and $t_i = \pi(x_i)$. We apply the GP regressor with Gaussian kernel and hyper-parameters $\lambda = 1$, $\eta = 0$. In this case, we obtain 6 positive coefficients and 3 negative coefficients in the vector $\boldsymbol{\alpha}$. The 3 negative coefficients are associated to the kernels localized at $x_i = 0, 0.5$ and 2. Applying the proposed RS scheme, the histogram of the accepted samples is given in Figure 1(d). The acceptance rate $A_r = 1-\rho = 0.417$. The unnormalized densities $p(x)$, $p_+(x)$ and the corresponding versions are also shown in Figure 1.

Scenario 2: Now, we consider $N = 11$ points in the regression,

$$x_i \in \{-8, -5, -4, -1.2, -0.8, 0.5, 1, 2, 5, 8, 10\}, \tag{39}$$

and again $t_i = \pi(x_i)$, as shown in Figure 2. We apply the GP regressor with Gaussian kernel and hyper-parameters $\lambda = 0.4$, $\eta = 0$. In this case, we have a unique negative coefficient, α_6,

[3] Related Matlab code is given at `http://www.lucamartino.altervista.org/public_code_NegMix2025.zip`.

corresponding to $x_6 = 0.5$. The acceptance rate $A_r = 1 - \rho = 0.974$, that is sensibly greater than in scenario 1. However, in both scenarios, we obtain more than reasonable acceptance rates, due to the suitable choice of the proposal density.

Scenario 3: We consider again the same $N = 9$ input points in Eq. (38) of Scenario 1. However, now we apply the GP regressor with different hyper-parameters of the Gaussian kernel, more precisely, $\lambda = 2$ and $\eta = 0.5$. Since $\eta > 0$ we have a regression instead of interpolation. In this scenario, we have two negative coefficients, α_4 and α_5. The acceptance rate $A_r = 0.503$ (i.e., 50%), that is again a good acceptance rate.

7.2 Emulator with negative coefficients as proposal density within IS

Let us consider again the following target density

$$\bar{\pi}(x) \propto \pi(x) = \sin(x)^2 \exp\left(-\frac{x^2}{30}\right).$$

For clarity of exposition and to help the understanding of the interested reader, we consider a simple emulator formulation in which negative coefficients appear and are explicitly shown. We consider an emulator with $N = 2$ that can be expressed as

$$\bar{p}(x) \propto p(x) = \alpha_1 k(x, x_1) + \alpha_2 k(x, x_2) = \alpha_1 \phi_1(x) + \alpha_2 \phi_2(x),$$

following Eq. (36). We consider Gaussian kernels, with $\mathbf{x}_1 = 0$, $\mathbf{x}_2 = 1$, $\lambda = 4$ and $\eta > 0$ is chosen such that $\alpha_1 = 3$ and $\alpha_2 = -1$. Then, we can write the (unnormalized) emulator as:

$$p(x) = 3\frac{1}{\sqrt{32\pi}} \exp\left(-\frac{x^2}{32}\right) - \frac{1}{\sqrt{32\pi}} \exp\left(-\frac{(x-1)^2}{32}\right).$$

Since $\bar{\alpha}_1 = \frac{\alpha_1}{\alpha_1+\alpha_2} = 1.5$ and $\bar{\alpha}_2 = \frac{\alpha_2}{\alpha_1+\alpha_2} = -0.5$, the normalized mixture/emulator is:

$$\begin{aligned} \bar{p}(x) &= \frac{p(x)}{\alpha_1 + \alpha_2} = \bar{\alpha}_1 \phi_1(x) + \bar{\alpha}_2 \phi_2(x), \\ &= 1.5\frac{1}{\sqrt{32\pi}} \exp\left(-\frac{x^2}{32}\right) - 0.5\frac{1}{\sqrt{32\pi}} \exp\left(-\frac{(x-1)^2}{32}\right). \end{aligned} \tag{40}$$

Note that, in this case, we have

$$\beta^+ = \alpha_1 = 1.5, \qquad 1 - \beta^+ = \alpha_2 = -0.5,$$

and $\bar{p}_+(x) = \frac{1}{\sqrt{32\pi}} \exp\left(-\frac{x^2}{32}\right)$ and $\bar{p}_-(x) = \frac{1}{\sqrt{32\pi}} \exp\left(-\frac{(x-1)^2}{32}\right)$, i.e., the partial mixtures are formed by just one component. Hence, with the previous definitions, we have

$$\bar{p}(x) = \beta^+ \bar{p}_+(x) + (1 - \beta^+)\bar{p}_-(x). \tag{41}$$

Given Eqs. (40)-(41), we can apply the IS scheme in Section 5 using $\bar{p}(x)$ are proposal density and approximate the following 3 integrals:

$$Z = \int_{-\infty}^{+\infty} \pi(x)dx, \quad I_1 = \int_{-\infty}^{+\infty} x^2\bar{\pi}(x)dx, \quad I_2 = \int_{-\infty}^{+\infty} x^4\bar{\pi}(x)dx. \tag{42}$$

We compute the ground-truth values using tight grids and evaluate the relative absolute errors (rel-AEs) of the estimates. The results are averaged over 10^5 independent runs, and a global relative absolute error is obtained as the arithmetic mean of the three rel-AEs. This procedure is repeated for different numbers of samples, $S \in \{10^2, 10^3, 10^4, 10^5, 10^6\}$, drawn from the $\bar{p}_+(x)$ and $\bar{p}_-(x)$. The results are given in Table 2. We can observe that the relative error decreases quickly as S grows.

Table 2: Global relative absolute error as function of S, averaged over 10^5 runs.

$S = 10^2$	$S = 10^3$	$S = 10^4$	$S = 10^5$	$S = 10^6$
0.40	0.19	0.11	0.05	0.04

8 Conclusions

In this work, we focused on mixtures with negative coefficients. These generalized mixture models enable more flexible and accurate density approximations, though they introduce challenges for handling and sampling such distributions. To address these challenges, we proposed efficient Monte Carlo methods (including quadrature techniques, rejection sampling, and importance sampling schemes) capable of accurately approximating integrals and generating (unweighted) samples from these non-convex mixtures. The use of a tailored proposal density ensures both accuracy and efficiency. Furthermore, we have designed how to utilize a mixture with negative coefficients as a proposal density in an importance sampling scheme. In this approach, some generated samples have negative importance weights. Consequently, this method can be likened to a physical analogy where samples with positive importance weights represent "matter", while those with negative importance weights represent "anti-matter". Applications to Gaussian process-based density estimation illustrate the practical relevance and effectiveness of the proposed methods, highlighting their potential for broader use in complex density modeling tasks.

Acknowledgements

This work has been partially supported by the PIACERI Starting Grant BA-GRAPH (UPB 28722052144) and the project PIACERI LikeFree-BA-GRAPH (UPB 28722052159) of the University of Catania.

Author Contribution declaration

Luca Martino is the single author of this work.

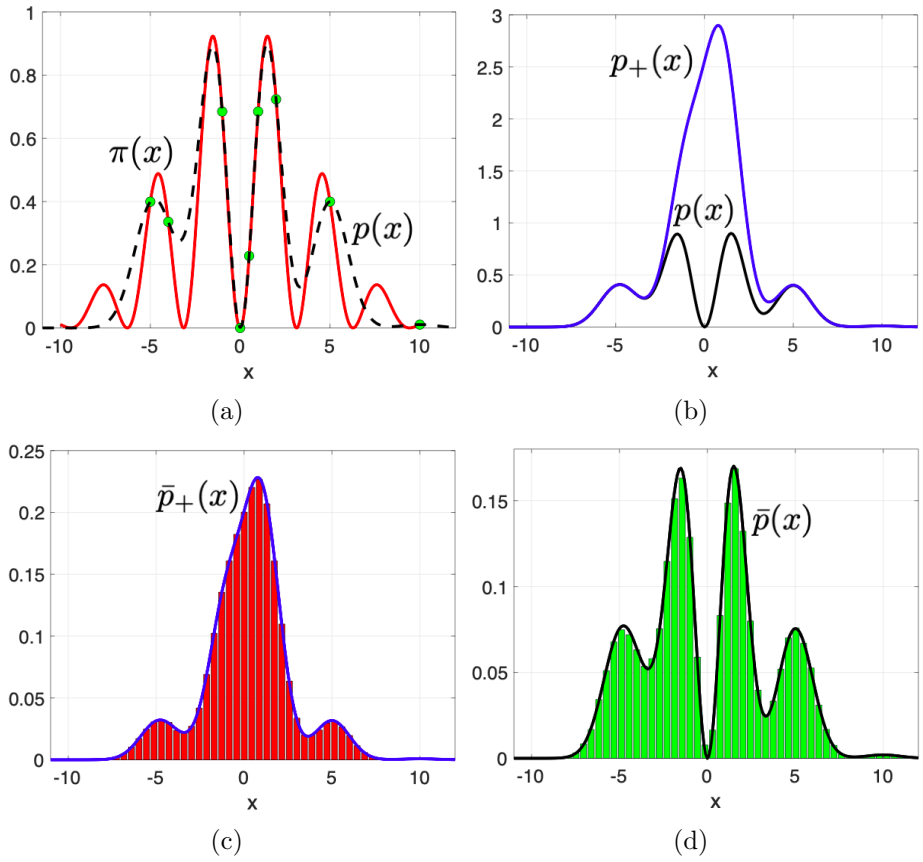
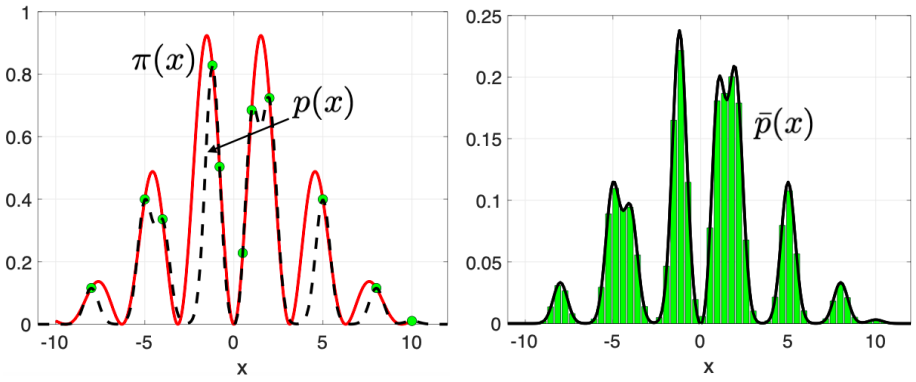


Figure 1: **Scenario 1: (a)** The red solid line shows $\pi(\mathbf{x})$, whereas the black dashed line represents the unnormalized Mix-NCs $p(\mathbf{x})$. The $N = 9$ input points $x_i \in \{-5, -4, -1, 0, 0.5, 1, 2, 5, 10\}$ in regression are depicted in green dots; in this scenario, with $\lambda = 1$, $\eta = 0$, we have 6 positive coefficients and 3 negative coefficients in $\boldsymbol{\alpha}$ (and, hence, in $p(\mathbf{x})$). The 3 negative coefficients are associated to the kernels localized at $x_i = 0,\ 0.5$ and 2. **(b)** The unnormalized positive partial mixture $p_+(\mathbf{x})$ (blue line) and the unnormalized Mix-NCs $p(\mathbf{x})$ in black line. **(c)** The normalized positive partial mixture $\bar{p}_+(\mathbf{x})$, used as proposal pdf in a RS scheme, and the corresponding histogram. **(d)** Histograms of the accepted samples in the RS scheme distributed according to $\bar{p}(\mathbf{x})$. The theoretical acceptance rate is $A_r = 1 - \rho = 0.417$, and the empirical acceptance rate is ≈ 0.416, in line with the theoretical expression. This means that drawing 20000 samples from $\bar{p}_+(x)$, in one run we obtain $S = 8326$ samples from $\bar{p}(x)$.

Data availability and related code

The datasets generated and analyzed during the current study are available (jointly with the related Matlab code) at `http://www.lucamartino.altervista.org/public_code_NegMix2025.zip`.



(a) (b)

Figure 2: **Scenario 2: (a)** The red solid line shows $\pi(\mathbf{x})$, whereas the black dashed line represents the unnormalized Mix-NCs $p(\mathbf{x})$. The $N = 11$ input points $x_i \in \{-8, -5, -4, -1.2, -0.8, 0.5, 1, 2, 5, 8, 10\}$ in regression are depicted in green dots; ; in this scenario, with $\lambda = 0.6$, $\eta = 0$, we have only one negative coefficient in $\boldsymbol{\alpha}$ (and, hence, in $p(\mathbf{x})$). **(b)** Histograms of the accepted samples in the RS scheme distributed according to $\bar{p}(\mathbf{x})$. The theoretical acceptance rate $A_r = 1 - \rho = 0.974$ which is virtually identical with the empirical acceptance rate obtained, i.e., ≈ 0.974. This means that drawing 20000 samples from $\bar{p}_+(x)$, in one run we obtain $S = 19481$ samples from $\bar{p}(x)$. The corresponding histogram is depicted in figure (b).

References

[1] O. D. Akyildiz. Global convergence of optimized adaptive importance samplers. *Foundations of Data Science*, 7(4):944–962, 2025.

[2] D. J. Bartholomew. Sufficient conditions for a mixture of exponentials to be a probability density function. *The Annals of Mathematical Statistics*, 40(6):2183–2188, 1969.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] G. Deligiannidis, P. E. Jacob, E. M. Khribch, and G. Wang. On importance sampling and independent Metropolis-Hastings with an unbounded weight function. *arXiv:2411.09514*, pages 1–43, 2025.

[5] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[6] M. Felgueiras. Mixtures with negative weights. Technical report, Center for Statistics and Applications, University of Lisbon, 2018.

[7] M. Felgueiras, J. Martins, and R. Santos. Pseudo-convex mixtures. *AIP Conference Proceedings*, 1479(1):1125–1128, 2012.

[8] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[9] F. Llorente and L. Martino. Optimality in importance sampling: a gentle survey. *arXiv:2502.07396*, 2025.

[10] F. Llorente, L. Martino, D. Delgado, and J. López-Santiago. Marginal likelihood computation for model selection and hypothesis testing: An extensive review. *SIAM Review*, 65(1):3–58, 2023.

[11] F. Llorente, L. Martino, D. Delgado-Gomez, and G. Camps-Valls. Deep importance sampling based on regression for model inversion and emulation. *Digital Signal Processing*, 116:103104, 2021.

[12] F. Llorente, L. Martino, V. Elvira, D. Delgado, and J. López-Santiago. Adaptive quadrature schemes for Bayesian inference via active learning. *IEEE Access*, 8:208462–208483, 2020.

[13] F. Llorente, L. Martino, J. Read, and D. Delgado-Gómez. A survey of Monte Carlo methods for noisy and costly densities with application to reinforcement learning and ABC. *International Statistical Review*, 93(1):18–61, 2025.

[14] L. Martino, V. Elvira, and G. Camps-Valls. Group importance sampling for particle filtering and MCMC. *Digital Signal Processing*, 82:133–151, 2018.

[15] L. Martino, D. Luengo, and J. Míguez. *Independent Random Sampling Methods*. Springer Publishing Company, Incorporated, 1st edition, 2018.

[16] L. Martino and J. Read. A joint introduction to Gaussian Processes and relevance vector machines with connections to Kalman filtering and other kernel smoothers. *Information Fusion*, 74:17–38, 2021.

[17] A. Mazza and A. Punzo. Mixtures of multivariate contaminated normal regression models. *Statistical Papers*, 61(2):577–608, 2020.

[18] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

[19] I. Murray, D. MacKay, and R. P. Adams. The Gaussian process density sampler. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

[20] C. J. Oates, J. Cockayne, F. X. Briol, and M. Girolami. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, 2017.

[21] G. Rabusseau and F. Denis. Learning negative mixture models by tensor decompositions. *ArXiv:1403.4224*, 2014.

[22] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[23] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[24] J.-M. Sellier. A signed particle formulation of non-relativistic quantum mechanics. *Journal of Computational Physics*, 297:48–74, 2015.

[25] M. Troyer and W.-J. Wiese. Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. *Physical Review Letters*, 94(17):170201, 2005.

[26] M. A. Vazquez and J. Míguez. Importance sampling with transformed weights. *Electronics Letters*, 53(12):783–785, 2017.

[27] A. Vehtari, D. Simpson, A. Gelman, Y. Yao, and J. Gabry. Pareto smoothed importance sampling. *J. Mach. Learn. Res.*, 25(1), 2024.

[28] T. Yu, L. Lu, and J. Li. A weight-bounded importance sampling method for variance reduction. *arXiv:1811.09436*, pages 1–14, 2019.

[29] B. Zhang and C. Zhang. Finite mixture models with negative components. In *Machine Learning and Data Mining in Pattern Recognition*, pages 31–41. Springer Berlin Heidelberg, 2005.