Estimating Committor Function with Mondrian Forest

Qiming Du, Tony Lelièvre

Abstract

In Molecular Dynamics, the rare event of interest can be modeled by the transition of some underlying Markov process between metastable states. In Transition Path Theory, a very important model reduction technique is to design a *reaction coordinate*, which is a function that measures the advance of a reactive trajectory towards a metastable state. Let A and B be two metastable states. A *committor function* is the perfect choice of reaction coordinate in the sense that it measures exactly the probability of reaching B before A. The committor function is also, in some sense, the perfect reaction coordinate for generalized Adaptive Multilevel Splitting (gAMS, see [BGG+16]) and crucial to the performances of many other rare-event estimation algorithms. We investigate the performance of Mondrian Forests (MF) [LRT14] to estimate the committor function, and we provide strategies to couple gAMS iteratively: first, since gAMS can also provide information on the committor function, we can use this algorithm to generate training data for MF, and, conversely, to update gAMS by using the trained MF model as its reaction coordinate. As iterations go, the updated-gAMS to generate better quality data, and with better quality data, MF should yield a good approximation of the committor function.

Contents

1 Introduction

In Transition Path Theory (TPT, see, e.g., [EVE10]), a typical problem is to sample the transition paths between a metastable state A and another metastable state B (see Figure 1). More precisely, let E denote a state space in which the underlying dynamics is modeled by a Markov process $\mathbf{X} := (X_t; t \geq 0)$. A metastable state of $\mathbf{X}$ is an open subset of E such that when $\mathbf{X}$ is trapped in such set, it takes an extremely long time for $\mathbf{X}$ to escape (see Figure 1). Let A, B be two metastable sets in E, then the committor function at point x is the probability that $\mathbf{X}$ starting from x, reaches B before A.

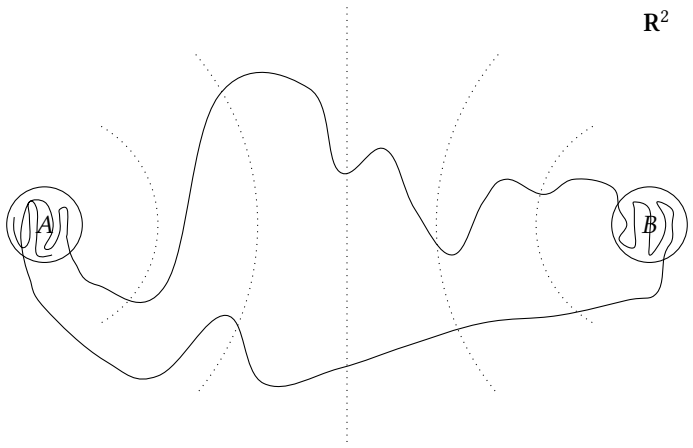

Figure 1: Schematic picture of Transition Path Theory.

By the metastability of A and B, when evaluating the values of the committor function close to A, crude Monte Carlo typically fails because the process is very much likely to go back to A rather than to go to B: the probability to be evaluated is thus very small. This metastability problem is related to a time scale problem: although the molecular transitions are not so rare at the macroscopic timescale, the dynamics encoded in the Markov process can only be simulated w.r.t. femtosecond timescale. Therefore, in order to ensure a certain level of accuracy, the wall-clock time of simulation is typically intractable.

One popular numerical approach to counter this is the gAMS framework [BGG+16]. The basic idea is to generate an Interacting Particle System (IPS) based on an adaptive level updating strategy, where the trajectories that advance more survive. A level is calculated w.r.t. a reaction coordinate, which is of crucial importance to the performance of the algorithm. It is well-known (see, e.g., [BLR15, CGR19]) that the committor function is the optimal reaction coordinate for gAMS. In addition, since gAMS is able to evaluate efficiently the values of the committor function close to A, the idea of designing a regressor to estimate the committor function is therefore natural.

An elementary approach is to split the state space E or some compact subset based on a regular grid cells, and a natural approximation of the committor function can then be derived in each cell thanks to ensemble of paths generated by the gAMS algorithm. The reader is referred to [LL19] for a rich list of numerical experiments and a concrete application on alanine dipeptide. However, this natural construction will only work in low dimensional settings, since it is not possible to create a uniform mesh when the dimension of the state space is large. Intuitively, a possible generalization of such a method is to find an intelligent way to create an adaptive mesh, such that it may also work in a high-dimensional setting.

Mondrian Forests (MF) [LRT14], a variant of Random Forests (RF), i.e., an ensemble of randomized decision trees, are proposed as the regressor of committor function. The construction of MF are based on a stochastic process called Mondrian process (MP, see, e.g., [RT09]), taking values in guillotine partitions of an axis-aligned box. Roughly speaking, a Mondrian process is a high-dimensional generalization of the partition on an interval that is split by a Poisson process. Said differently, MF provide an intelligent way to create "randomized uniform mesh" that is tractable in a high-dimensional setting. From a different angle, MF can be regarded as an RF-based regressor, such that *online learning* is available. More concretely, when new training data come, there is no need to retrain the model in order to have a better quality regressor. This property is crucial for our iterative updating strategy (cf. Figure 2).

Unfortunately, the theoretical understanding of MF are still in its infancy. We refer to [MGS17, MGS18] for recent theoretical developments on MF, where a Purely Random Forests version of MF are proposed, along with a min-max convergence rate analysis. Therefore, we mainly provide numerical illustrations and ideas on how to use MF to design efficient strategy to estimate the committor function.

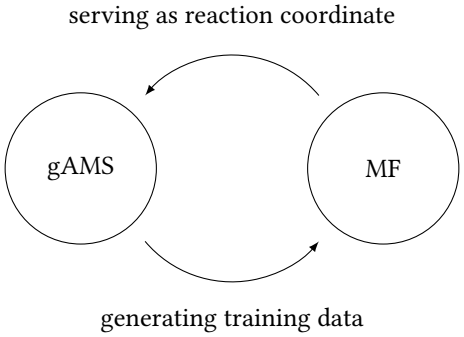

Figure 2: The illustration of iterative updating strategy.

2 Setting

2.1 Overdampled Langevin dynamics

We consider an overdamped Langevin process $\mathbf{X} = (X_t; t \geq 0)$ taking values in the state space $E = \mathbf{R}^d$ defined by

$$dX_t = -\nabla V(X_t)dt + \sqrt{2\beta^{-1}}dW_t, \tag{1}$$

where $(W_t; t \geq 0)$ denotes a d-dimensional Wiener process, V denotes the associated energy and the inverse temperature $(\kappa_B T)^{-1}$ is denoted by β.

A metastable state is an open subset of E such that when $\mathbf{X}$ is trapped in such set, it takes an extremely long time for $\mathbf{X}$ to escape. Let us denote A and B two metastable states in a state space E. For the Markov process $\mathbf{X} := (X_t, t \geq 0)$, let τ_A and τ_B denote respectively the stopping times

$$\tau_A := \inf\{t \geq 0 \mid X_t \in A\},$$

and

$$\tau_B := \inf\{t \geq 0 \mid X_t \in B\}.$$

The *committor function* $\xi^* : E\backslash(A \cup B) \mapsto [0, 1]$ is defined by

$$\xi^*(x) := \mathbf{P}(\tau_B < \tau_A \mid X_0 = x).$$

Our goal is to design an efficient strategy to estimate this function on some compact subset of the state space E. In particular, we are interested in exploiting this estimation to improve the performance of gAMS and vice versa.

2.2 Ground truth

In this section, we show how one can obtain the reference values of a committor function in overdamped Langevin dynamics for some low-dimensional toy examples, which will be referred to as *ground truth*, namely the "real" values of the committor function. This method only works in low-dimensional setting. It is well known (see, e.g. [BLR15]) that ξ^* is the solution of the following elliptic Partial Differential Equation (PDE):

$$-\nabla V \cdot \nabla u + \beta^{-1}\Delta u = 0 \quad \text{on} \quad E\backslash(A \cup B), \tag{2}$$

with the boundary conditions

$$\begin{cases} u = 0 & \text{on } \partial A; \\ u = 1 & \text{on } \partial B. \end{cases} \tag{2'}$$

For sufficiently fine grids, the numerical solution of (2) using Finite Difference method is precise enough so that it can be regarded as the real committor function.

Three-hole potential The main example investigated in this article is the following 2-dimensional potential function (see Figure 3):

$$\begin{aligned} V(x, y) :=& 3\exp\left(-x^2 - (y - 1/3)^2\right) - 3\exp\left(-x^2 - (y - 5/3)^2\right) \\ &- 5\exp\left(-(x - 1)^2 - y^2\right) - 5\exp\left(-(x + 1)^2 - y^2\right) + x^4/5 + (y - 1/3)^2/5. \end{aligned}$$

We consider the following two metastable states respectively defined by

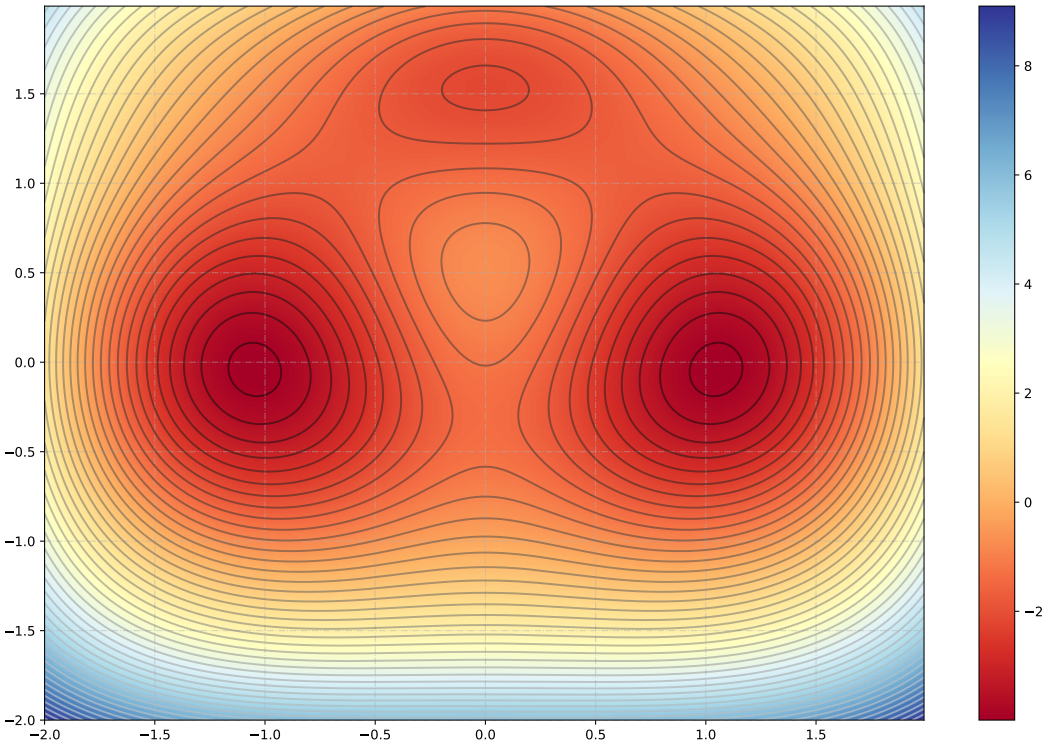

Figure 3: Representation of the three-hole energy landscape.

$$A := \{(x, y) \in \mathbf{R}^d : (x + 1)^2 + y^2 < 0.1\},$$

and

$$B := \{(x, y) \in \mathbf{R}^d : (x - 1)^2 + y^2 < 0.1\}.$$

The reader is referred to [MSVE06] for a wider list of toy examples. In order to solve the elliptic PDE (2) by Finite Difference method, we consider a rectangular domain $\Omega = [-2, 2] \times [-2, 2]$ and a uniform mesh with stepsize 0.01. We also add a Neumann boundary condition on the boundary $\partial\Omega$ of the rectangular domain, i.e., $\partial_{\vec{n}} u = 0$ where $\vec{n}$ denotes the unit normal vector on the boundary. These fictitious boundary conditions do not affect too much the quality of the result since when starting far from A and B, the difference of the values of the committor function for two close points is negligible. This ensures that the Finite Difference result yields an accurate approximation of the solution to (2)-(2'). The numerical solution of (2)-(2') for the inverse temperature $\beta = 1.67$ is illustrated in Figure 4.

2.3 On the choice of regressors

The committor function is a smooth function, taking values in $[0, 1]$. In this sense, any *state-of-the-art* machine learning model may serve as a regressor the committor function. However, when interacting with gAMS methods, certain practical aspects have to be taken into account when choosing the learning method:

- First, the regressor has to be fast in terms of prediction speed. In fact, gAMS algorithm requires that at each state of each trajectory, the reaction coordinate is evaluated. This is a *huge* amount of computation costs if the prediction of the regressor is complicated, and most of them would not be useful. In this sense, a sophisticated fine-tuned Deep Learning model may not be a relevant candidate.
- Second, the regressor must be able to execute online learning or incremental learning. When new data come, the regressor should be able to update in order to have better accuracy. In this sense, classic Random Forests may not be a good choice.
- Finally, we expect it to be as adaptive and robust as possible in high dimension, since we do not have much insights on how to tune a specific model to learn a high-dimensional function.

As such, we may consider two families of regressors in order to estimate the committor function efficiently. The advantages and disadvantages are listed as follows.

(i) Gradient-based learning algorithms, such as logistic regression, XGBoosting, or some shallow neural network:

  (a) Online learning is available by a Stochastic Gradient Descent-based optimization algorithm;

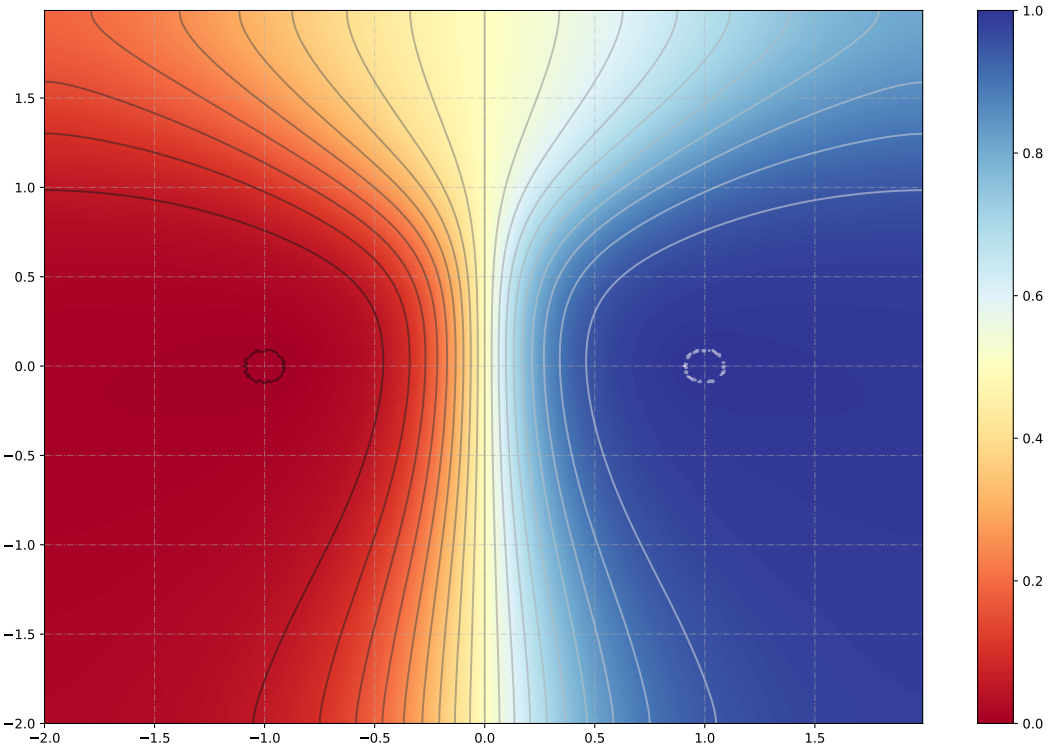

Figure 4: The numerical solution of (2) in a high temperature scheme with $\beta = 1.67$.

(b) It is easy to add regularization modules, such as the "smoothness" of the prediction, etc;

(c) The parameters of the algorithm are hard to tune in general;

(d) Sometimes, the results are hard to interpret.

(ii) Ensemble-based learning algorithms, mainly variants of Random Forests:

(a) The ensemble methods are super robust in high-dimensional settings by the design of Monte Carlo-typed structure;

(b) Nearly no tuning is needed, which also means that the regularization modules are not easy to implement in general;

(c) Online learning is generally difficult to design;

(d) It is straightforward to understand and control possible dangerous situations.

3 A brief introduction to Mondrian Forests

Mondrian Forests (MF) were introduced in [LRT14], named after the famous Dutch painter Piet Mondrian, as the partitions created by each Mondrian tree (MT) and Mondrian's paintings have similar style. The crucial idea of the construction is based on a guillotine-partition-valued stochastic process called the Mondrian Process (MP). For details, the readers are referred to [RT09] and [BW15]. In general, MF are a variant of RF such that online learning is available, and designed in a smart way. In this section, we present the basic mechanism of a Mondrian tree and we explain why MF are a competitive candidate as a regressor for estimating the committor function.

3.1 The mechanism of a Mondrian tree

In this section, we provide an intuitive interpretation on the construction of a Mondrian tree on a 2-dimensional toy example. For the generic algorithm, the reader is referred to [LRT14]. Let us fix a parameter $\lambda \in \mathbf{R} \cup \{+\infty\}$, which defines the *lifetime* of a Mondrian tree. Let us denote by $(X_i, Y_i; 1 \leq i \leq n)$ the training data. Before starting, we remark that the partition defined by a MT only depends on the inputs $(X_i; 1 \leq i \leq n)$ of the training data.

Construction of MT in dimension 2 At step 0, we denote $ABCD$ the minimum rectangle that covers $X_1, X_2, \ldots, X_n$ (see Figure 5). Then, we sample an exponential random variable E_0 with rate $(|AB| + |BC|)$. If $E_0 < \lambda$, a split on the side AB or BC will be executed. More precisely, we sample a uniform random variable U on the interval $[0, |AB| + |BC|]$. If $U \leq |AB|$, a splitting point will be uniformly sampled on the side AB; otherwise,

the split will be done on the side BC. After the splitting point is determined, a split will be executed orthogonally to the chosen side. For short, this amounts to say that an orthogonal split is performed uniformly on ABC. In Figure 5, the splitting point is on the side AB and its abscissa is denoted by x_0. Therefore, the training data is vertically divided into two subgroups.

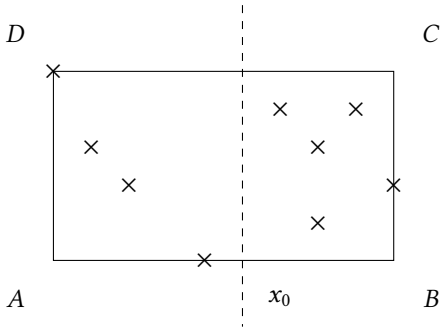

Figure 5: First split in the construction of MT.

At the same time, a node in a decision tree according to the split is therefore constructed (see Figure 6). By definition, the condition node $x > x_0$ illustrated in Figure 6 can be determined by the pair of sets $(\{x > x_0\}, \{x \leq x_0\})$. We say that the lifetime of the node $(\{x > x_0\}, \{x \leq x_0\})$ is E_0. Note that a decision tree uniquely determines a partition on the whole state space $\mathbf{R}^2$.

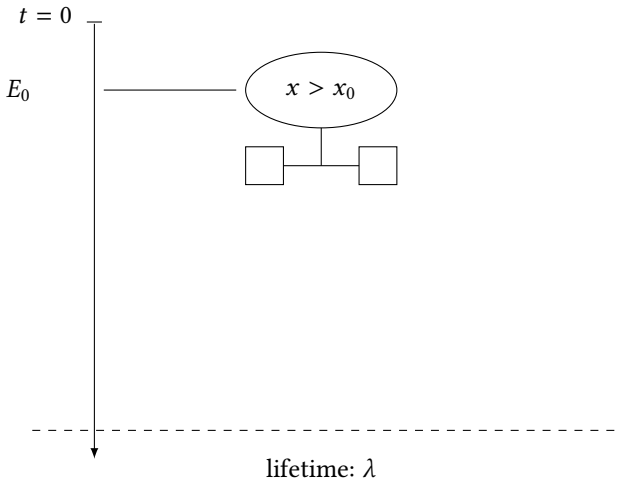

Figure 6: Decision tree corresponding to the first split.

Next, we perform the splitting for these two subgroups of data recursively. Let us start by the subgroup of data on the left-hand side (see Figure 7). Denote again by $ABCD$ the minimum rectangle that covers all the data in this subgroup of data. We sample E_1^1 w.r.t. an exponential distribution with rate $(|AB| + |BC|)$. Now, if $E_0 + E_1^1 < \lambda$, we perform a uniform split on BCD, in the same way as presented at step 0. The same mechanism is applied mutatis mutandis to the right-hand side (see Figure 8). The associated decision tree is also updated accordingly, and one tracks the lifetime of each node (see Figure 9).

The splitting procedure stops when the lifetime of the proposed condition node surpasses the prefixed lifetime λ. We also remark that, when a subset contains only 1 data, the splitting stops automatically since the minimum rectangle that contains one point degenerates to a point. In the final stage, a randomized decision tree is therefore constructed, along with a random partition of the state space $\mathbf{R}^2$. In a high-dimensional setting, the orthogonal lines used to execute the splitting procedures are replaced by hyperplanes, and the rectangles are replaced accordingly by hypercubes. The basic mechanism remains the same. The prediction of MT is then provided following the decision tree. More precisely, the prediction on the point x^* is defined as the average of the outputs of the training data that are in the same hypercube as x^*. The online learning of MT exploits the memoryless property of exponential distribution. Indeed, when new data come, one regenerates part of the decision tree such that the existing structure is not changed. It turns out that the online training of a Mondrian tree does not change its posterior distribution given the same data (cf. (4)), and the reader is referred to Section 5 of [LRT14] for detailed algorithms.

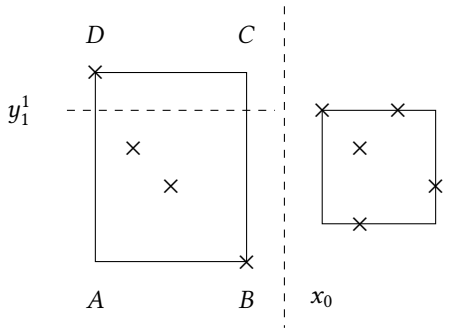

Figure 7: Second split in the construction of MT.

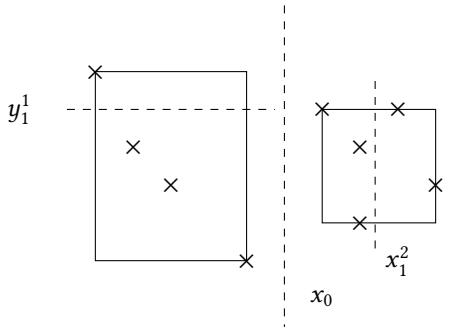

Figure 8: Second and third splits in the construction of MT.

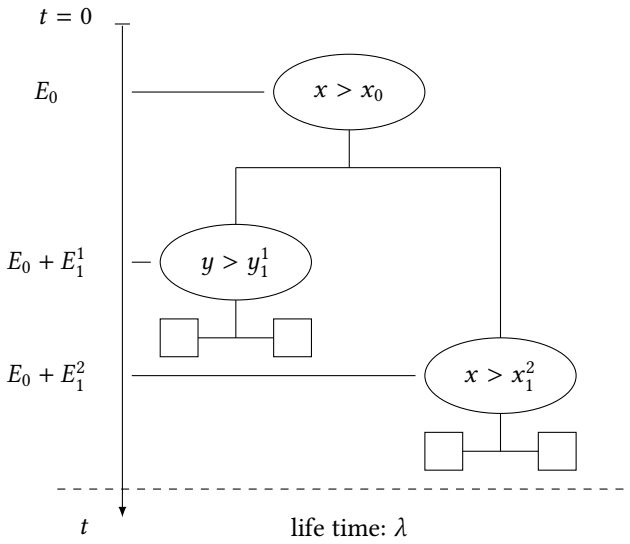

Figure 9: The decision tree with three splits.

3.2 Theoretical aspects of MF

In this section, we provide some theoretical properties of MF, highlighting the missing part in the theory.

Properties of Ensemble methods MF are ensemble methods, meaning that many weak regressors are constructed independently and we use the average of their predictions as the final estimation. Here, these weak regressors are Mondrian trees: they are variants of classic decision trees used in Random Forests. Denote by $D_{[n]} = (D_i; 1 \le i \le n) = (X_i, Y_i; 1 \le i \le n)$ a sequence of training data. We also denote $\mathbb{T}_1(D_{[n]}), \cdots, \mathbb{T}_M(D_{[n]})$ a collection of conditional i.i.d. Mondrian trees. We denote $\mathbf{Pred}_{\mathbb{T}_m}\left(x \mid D_{[n]}\right)$ the prediction at point x made by $\mathbb{T}_m(D_{[n]})$. Then, the final prediction is

$$\mathbf{Pred}_{\mathbb{T}_{[M]}}\left(x \mid D_{[n]}\right) := \frac{1}{M}\sum_{m=1}^{M} \mathbf{Pred}_{\mathbb{T}_m}\left(x \mid D_{[n]}\right)$$

Hoeffding inequality and Borel-Cantelli lemma give the following law of large numbers as the prediction is uniformly bounded by 1 for the committor function estimation problem:

$$\mathbf{Pred}_{\mathbb{T}_{[M]}}\left(x \mid D_{[n]}\right) \xrightarrow[M\to\infty]{a.s.} \mathbf{E}\left[\mathbf{Pred}_{\mathbb{T}_1}(x) \mid D_{[n]}\right].$$

At the same time, we have

$$\mathrm{Var}\left[\mathbf{Pred}_{\mathbb{T}_{[M]}}\left(x \mid D_{[n]}\right)\right] = \frac{1}{M}\mathrm{Var}\left[\mathbf{Pred}_{\mathbb{T}_m}\left(x \mid D_{[n]}\right)\right].$$

The final prediction is then more consistent than the prediction made by each regressor. Hence, M can be set as large as possible in practice such that we only have to deal with the randomness introduced by a single MT.

Missing parts in the convergence analysis As is shown above, the variance brought by ensemble methods can be reduced simply by adding more MT in MF. However, for the expectation of the prediction provided by MT given data, the consistency is not guaranteed in general. Let us give a more detailed setting. Assume that the distribution of D_i writes,

$$\begin{cases} X_i \sim \mathrm{Unif}(\Omega); \\ \epsilon_i \sim \mathcal{N}(0, \sigma_i^2); \\ Y_i = \xi^*(X_i) + \epsilon_i, \end{cases}$$

there is no theoretical guarantee such that

$$\mathbf{E}\left[\mathbf{Pred}_{\mathbb{T}_1}(x) \mid D_{[n]}\right] \xrightarrow[n\to\infty]{a.s. \text{ or } \mathbf{P} \text{ or } L^p} \xi^*(x). \tag{3}$$

The assumption that ϵ_i is Gaussian is quite natural since both crude Monte Carlo and gAMS provide estimation with normal limit distribution. Recent results [MGS17] show that when $\lambda < +\infty$, there is no consistency in general. A more refined analysis and a variant of Purely Random Forests version of MF are proposed in [MGS18], with a min-max rate for α-Hölder functions. However, we did not use this variant since the construction of the purely random Mondrian tree allows empty cell, and 0 is set to be the estimation when the evaluation is needed to be conducted in such cells. This induces problems in the implementation of gAMS algorithm since $\xi = 0$ means a sudden death of a transition path, which makes the implementation of gAMS numerically unstable.

Since providing a huge amount of training data is not possible due to the computational cost of gAMS methods, we are more interested by the performance when relatively few training data is provided. Hence, asymptotic properties such as consistency are not the priority for the applications in order to improve the performance of gAMS, since the final estimation of rare-event simulation is eventually estimated by gAMS, and even with non-converged estimation of committor function, we still have a theoretical-guaranteed unbiased estimator. In addition, since the construction of decision tree-type regressor ensures that the estimator can only take finite values, gAMS enters into Asymmetric SMC framework introduced in Chapter 3 and can also provide theoretical guaranteed consistent estimations.

Self-consistency of MT The online training of each MT does not depend on the arrival orders of $D_i = (X_i, Y_i)$. More precisely, we have

$$\mathbb{T}_m(D_{[n+1]}) \sim \mathcal{T}_\lambda(D_{[n+1]}) \Leftarrow \begin{cases} \mathbb{T}_m(D_{[n]}) \sim \mathcal{T}_\lambda(D_{[n]}); \\ \mathbb{T}_m(D_{[n+1]}) \,\Big|\, D_{n+1} \sim \mathcal{M}_{D_{n+1}}\left(\mathbb{T}_m(D_{[n]}), \cdot\right), \end{cases} \tag{4}$$

where $\mathcal{M}_{D_{n+1}}$ represents the online updating strategy of Mondrian tree given the data D_{n+1} and $\mathcal{T}_\lambda(D_{[n]})$ denotes the distribution of Mondrian tree with lifetime λ given the data $D_{[n]}$. According to the author of [LRT14],

this is the only construction of decision tree available such that the property above is verified. However, this beautiful property comes with a cost: the splits of the MF do not use the value of the Y_i. Therefore, they are not the "optimal" splits given the training data used in classical decision tree's construction. Hence, the robustness over the "extremely bad quality data" will be affected. This will be discussed in the numerical illustration in the next section.

4 Numerical illustrations

In this section, we provide some numerical illustrations and interpretations on the estimation of committor function. The energy is set to be the three-hole energy introduced in Section 2.2. In the following sections, the sample points are uniformly sampled in the rectangle $\Omega = [-2, 2] \times [-2, 2]$, and by "perfect training data" we mean the numerical solution of (2) given by Finite Difference methods.

4.1 Learning with perfect data

Unlike the typical estimation problems in statistics and machine learning context, the quality of the training data is indeed controllable in the committor function estimation problem. Although it is of no practical interest to do this kind of trade-off, we investigate the performance of training MF with perfect data (ground truth) to test its adequacy and to have an idea on how many data is needed to provide reasonable approximations. The results are given in Figure 10.

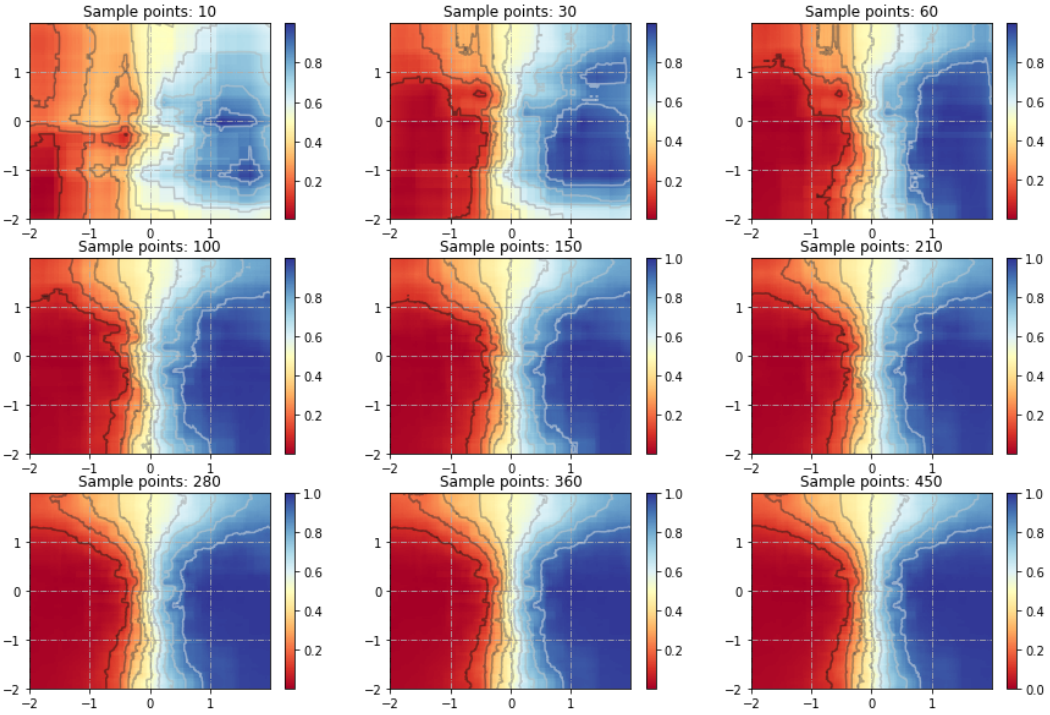

Figure 10: Learning with the ground truth, where the training data is uniformly sampled over $\Omega = [-2, 2] \times [-2, 2]$. For the parameters of MF, we set $\lambda = \infty$ and $M = 50$. The inverse temperature is $\beta = 1.67$.

4.2 Learning with noisy data

Now, we consider adding some artificial Gaussian noise to the training data. For a gAMS algorithm with $N = 100$ and $K^* = 20$, the typical variance over the rectangle Ω is between 10^{-6} to 10^{-4}. Therefore, we consider adding a slightly larger centered normal noise with variance 4×10^{-4}. The results are shown in Figure 11. Although it seems that MF cannot handle the situation perfectly, the approximation is still quite impressive. Since the added noises are i.i.d. normal random variables, the relative variance on the left ($x \leq 0$) is noticeably larger than on the right ($x \geq 0$). This is why the estimation quality on the left is worse than on the right in general. The situation where the noise is relatively large is presented in Figure 12. We use the possibly largest variance, 1, that can be made by crude Monte Carlo or gAMS. This time, unsurprisingly, MF failed to provide reasonable predictions.

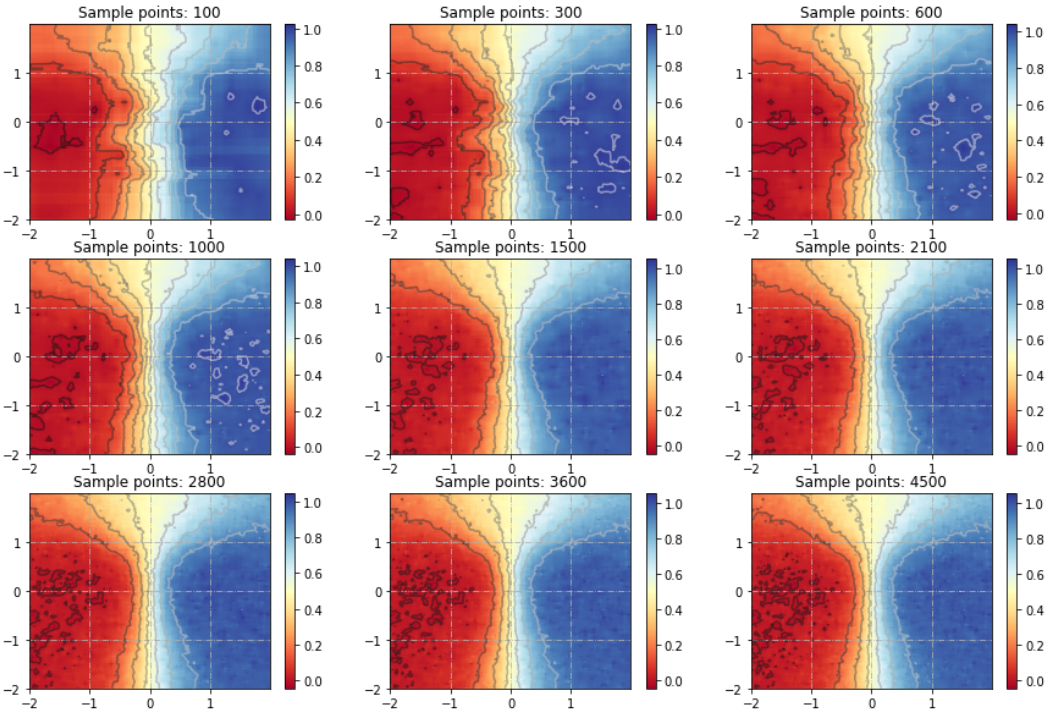

Figure 11: Learning with slightly perturbed data, where (X_i) are uniformly sampled over $\Omega = [-2, 2] \times [-2, 2]$. For the parameters of MF, we set $\lambda = \infty$ and $M = 50$. The inverse temperature is $\beta = 1.67$.

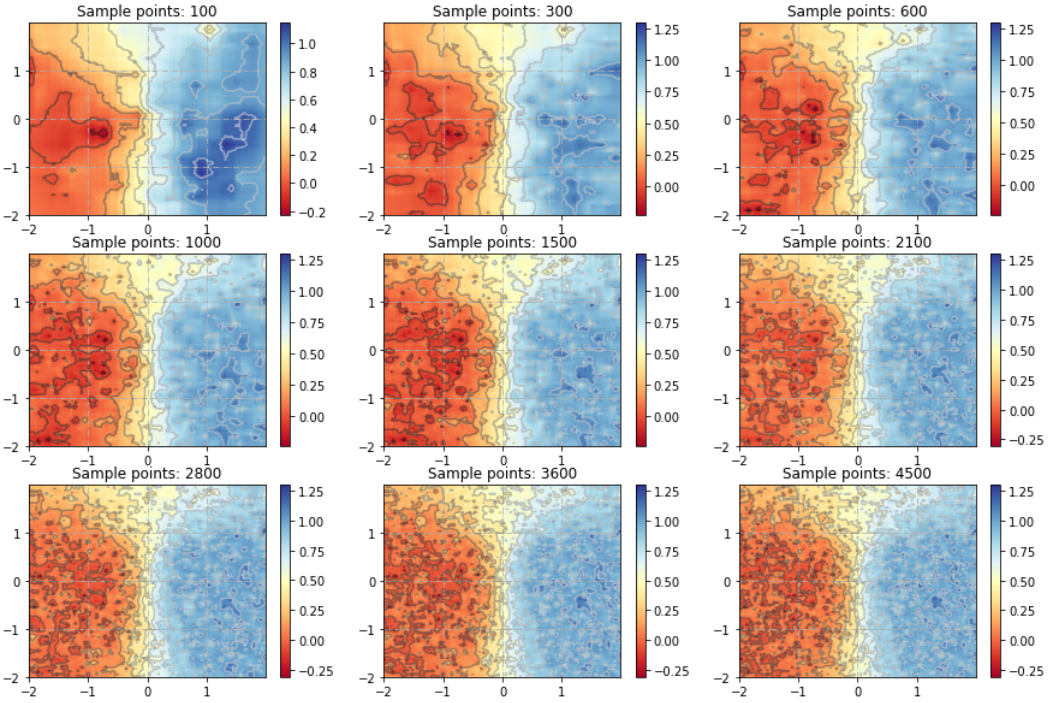

Figure 12: Learning with largely perturbed data, where (X_i) are uniformly sampled over $\Omega = [-2, 2] \times [-2, 2]$. For the parameters of MF, we set $\lambda = \infty$ and $M = 50$. The inverse temperature is $\beta = 1.67$.

4.3 Capability of recovering from a ruined model

By a "ruined model" we mean the model generated using extremely low quality training data. We start by training a MF model with largely perturbed data, and then, we continue by providing perfect data to the MF model, to see if it could recover from the ruined model. As presented in Figure 13, it may be difficult for an MF model to recover from a ruined one, meaning that a huge amount of high quality training data is required. According to Figure 10, a high quality training data of size 1000 is enough to provide an accurate approximation.

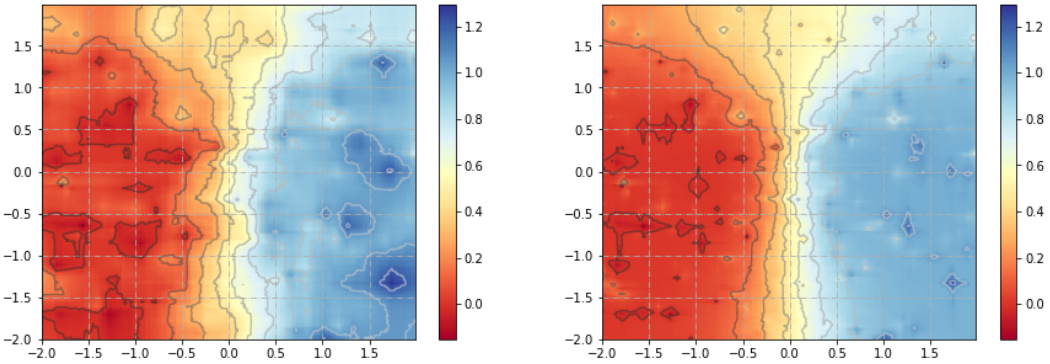

Figure 13: The figure on the left represents the prediction made by a MF model trained by 100 largely perturbed data. The one on the right represents the model trained with 1000 more perfect data. (X_i) are uniformly sampled over $\Omega = [-2, 2] \times [-2, 2]$. For the parameters of MF, we set $\lambda = \infty$ and $M = 50$. The inverse temperature is $\beta = 1.67$.

4.4 Conclusion of the numerical tests

In this section, we summarize the empirical knowledge on MF that we have collected in the numerical tests:

(i) MF do not need a huge sample size to give a reasonable approximation of committor functions;

(ii) MF prefer having less high quality data rather than more low quality data;

(iii) MF are not robust to largely perturbed data;

(iv) It is difficult for a ruined MF model to recover by updating through online learning with high quality training data.

As a consequence, in the design of the iterative updating strategy mentioned in the previous sections, it is crucial to ensure the quality of the training data for MF. This intuition is contrary to the one in typical Machine Learning applications, where the size of the data is usually of top priority.

5 Iterative updating strategy with gAMS and MF

In this section, we discuss some possible combinations of MF and gAMS algorithm, and explain how they can help each other in order to improve accuracy. Some numerical illustrations are also provided.

5.1 Crude interaction between gAMS and MF

Let us start by using gAMS to estimate committor function in a crude way: we use the estimation provided by gAMS as the training data of MF, with some prefixed reaction coordinate such as the Euclidean distance to the state A. The number of replicas is $N = 100$, and the minimum number of replicas to be killed at each iteration is $K^* = 20$. The reaction coordinate ξ_1 is the Euclidean distance to the point $(-1, 0)$ with the threshold $L^* = 1.8$. The starting point X_0 is sampled uniformly over the rectangle Ω. In particular, when the sampled point turns out to be in A (resp. B), we provide directly 0 (resp. 1) as the output of gAMS. At each point, we simulate independently $n_{sim} = 50$ runs of gAMS, and the final estimation is the average. The prediction given by the MF model is presented in Figure 14. Then, we use the trained MF model as the new reaction coordinate and the updated gAMS is therefore implemented to generate new training data for MF. The final performance of the MF model is provided in Figure 15.

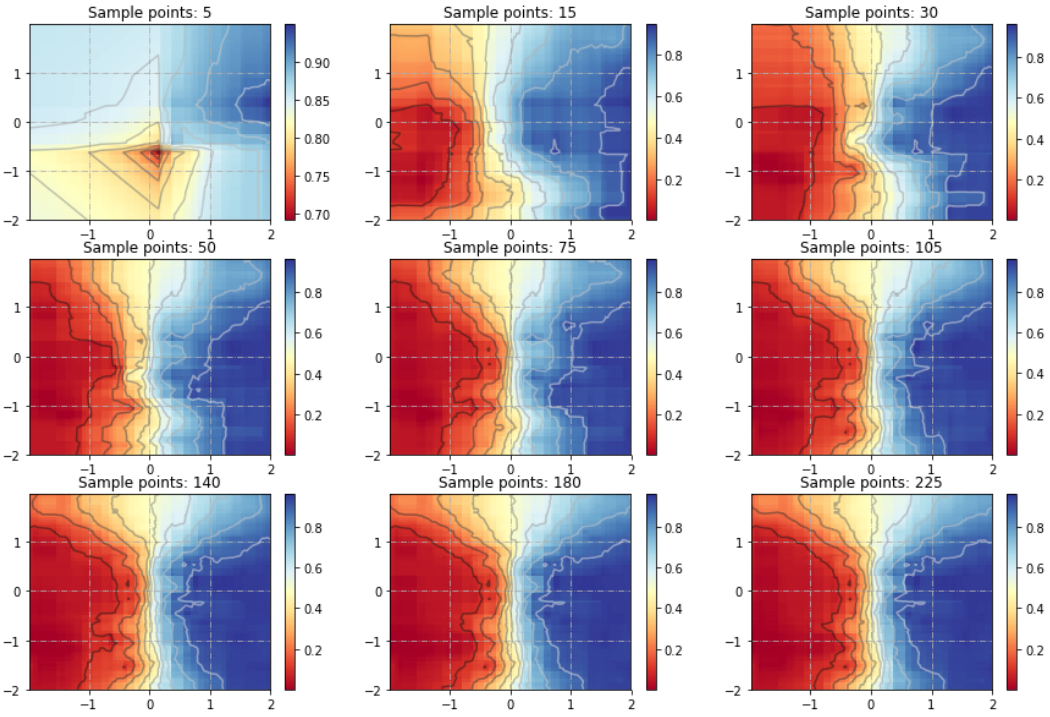

Figure 14: Learning with estimation by gAMS algorithm, with reaction coordinate ξ_1. The inverse temperature is $\beta = 1.67$.

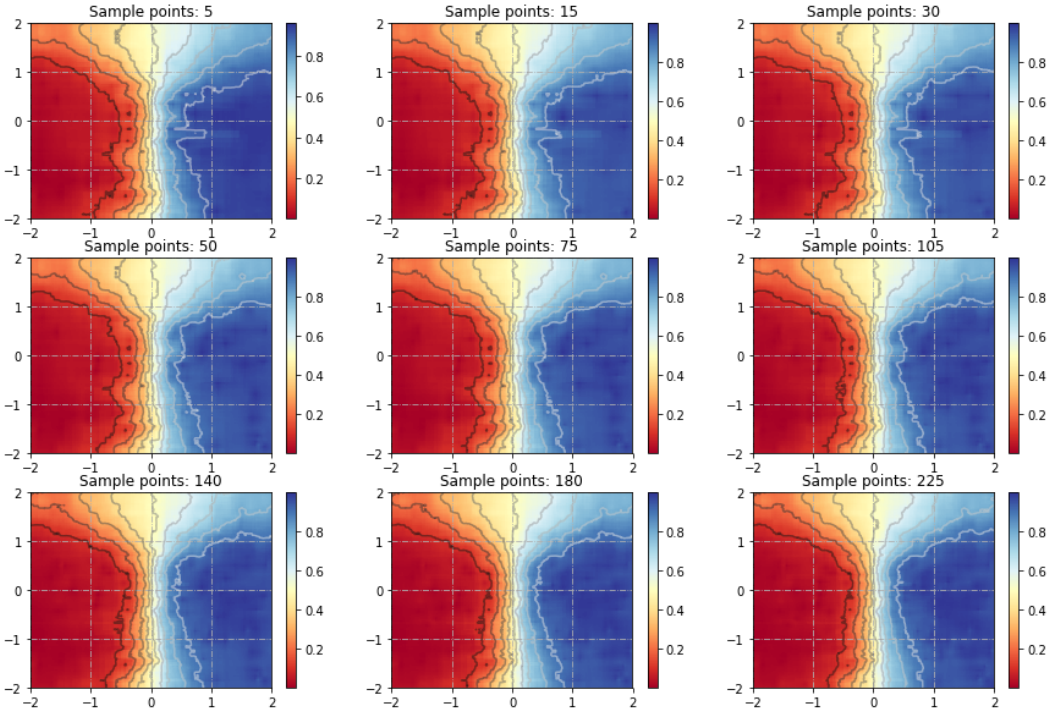

Figure 15: Learning with a trained MF model as reaction coordinate. The inverse temperature is $\beta = 1.67$.

Clearly, the quality of approximation is improved. This procedure can be done repeatedly and we expect the approximation of the committor function to be more and more accurate.

5.2 Tempering

In the previous section, we need to have a reasonable reaction coordinate in order to run gAMS in the first place. This is not always tractable, especially in high dimensional situations. Here, we provide a possible strategy that allows to "start out of nowhere". The idea is simple: when the temperature is high, the rare events associated to the energy barriers become less rare. We start with a small inverse temperature β_0, such that crude Monte Carlo provides reasonable estimations. Then, we estimate the associated committor function ξ_{β_0} by crude Monte Carlo. Next, we consider a tempering sequence (see Figure 16) $\beta_0 < \beta_1, < \cdots < \beta_{q^*-1} < \beta_{q^*} = \beta^*$, where β^* denotes the target inverse temperature. For each $q \in [n]$, we estimate ξ_{β_q} by using $\xi_{\beta_{q-1}}$ as reaction coordinate, until we get the target inverse temperature. Although this strategy is computationally intensive, it allows to design gAMS algorithm in a much more adaptive and optimal way.

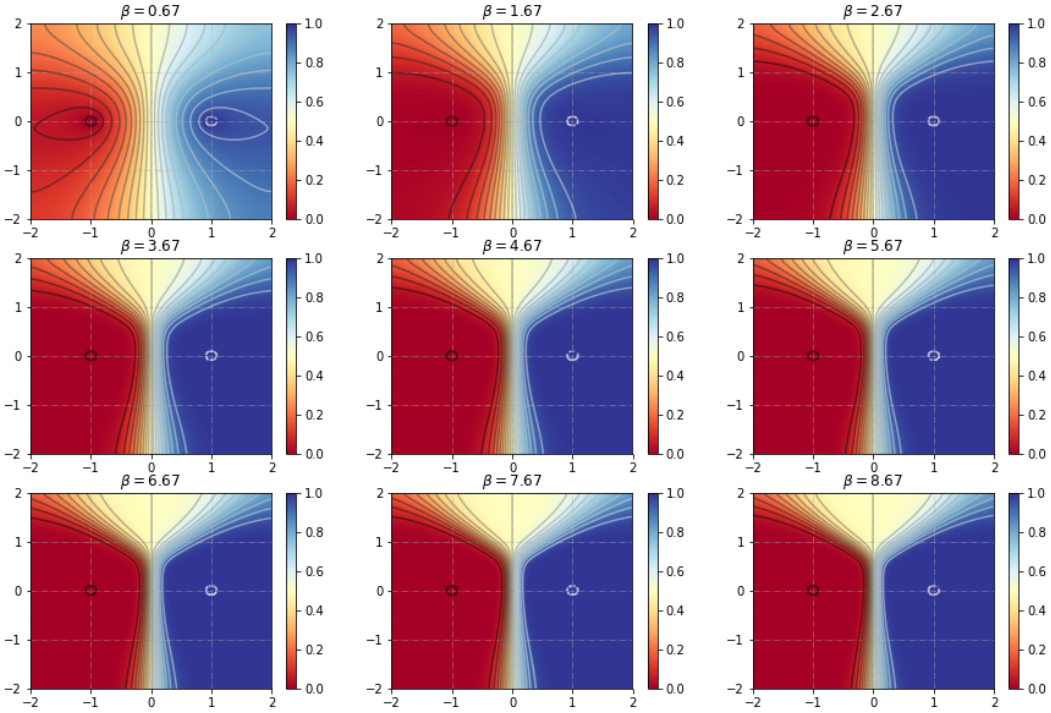

Figure 16: Illustrations of committor functions for different temperatures.

6 Discussions

In this chapter, we mainly discussed the possibility of using MF to estimate committor functions. This paves the way to a wide range of possible interacting strategies of gAMS and MF. However, a more refined study still needs to be conducted.

The first remark is on the sample points of the training data. In high-dimensional settings, the uniform sampling strategy is typically intractable. In fact, one does not need to estimate the whole picture of the committor function if one only seeks to improve the performance of gAMS. A possible approach is to sample along the reactive trajectories generated by gAMS. In this way, it is expected that the prediction of MF to be more accurate along these transition paths, which yields a more accurate gAMS estimator. However, to develop a proper sampling strategy for the sample points is not trivial. For example, for the overdamped Langevin dynamics, the reactive trajectories stay longer in the well of the energy landscape, which means that by uniformly sampling the points along the reactive trajectories is not efficient in general. Hence, the details still need to be investigated.

The second remark is on hybrid approaches that combines gAMS and crude Monte Carlo. In fact, the implementation of gAMS is mainly due to the behaviors of committor function close to the metastable states A and B. We remark that when close to B the rare event is of form $1 - p_*$, namely the values of the committor function is very close to 1. This means that in a lot of places where the values of the committor function is around 0.5, crude Monte Carlo is already able to provide reasonable estimations. Therefore, the design of an adaptive hybrid approach may greatly increase the efficiency of the algorithm.

The final remark is on Mondrian Forests. Since it is proved that finite lifetime parameter λ yields non-consistent estimator, it would be interesting to explore another alternative stopping criterion for the growth of MT, in order to improve its robustness against largely perturbed data. One possible choice is standard in a Random Forests context, that is to fix a threshold to control the minimum number of data in each subset of the partition of the decision tree. Said differently, we fix a number $N_{\min}$ in $\mathbf{N}$ such that when a subset in the partition contains less data than $N_{\min}$, the split is rejected. The ideal case is that we can somehow design a strategy such that this number $N_{\min}$ can be evaluated by the variance of the output of the data. Since variance estimation is available for both crude Monte Carlo and gAMS with MF as reaction coordinate, it would then be possible to develop more automated and advanced algorithms.

References

[BGG+16] Charles-Edouard Bréhier, Maxime Gazeau, Ludovic Goudenège, Tony Lelièvre, and Mathias Rousset. Unbiasedness of some generalized adaptive multilevel splitting algorithms. *Ann. Appl. Probab.*,

26(6):3559–3601, 2016.

[BLR15] Bréhier, Charles-Edouard, Lelièvre, Tony, and Rousset, Mathias. Analysis of adaptive multilevel splitting algorithms in an idealized case. *ESAIM: PS*, 19:361–394, 2015.

[BW15] Matej Balog and Yee Whye Teh. The Mondrian Process for Machine Learning. *arXiv e-prints*, page arXiv:1507.05181, Jul 2015.

[CGR19] Frédéric Cérou, Arnaud Guyader, and Mathias Rousset. Adaptive multilevel splitting: Historical perspective and recent results. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(4):043108, 2019.

[EVE10] Weinan E and Eric Vanden-Eijnden. Transition-path theory and path-finding algorithms for the study of rare events. *Annual Review of Physical Chemistry*, 61(1):391–420, 2010. PMID: 18999998.

[LL19] Laura J. S. Lopes and Tony Lelièvre. Analysis of the adaptive multilevel splitting method on the isomerization of alanine dipeptide. *Journal of Computational Chemistry*, 40(11):1198–1208, 2019.

[LRT14] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. Mondrian forests: Efficient online random forests. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3140–3148. Curran Associates, Inc., 2014.

[MGS17] Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. Universal consistency and minimax rates for online mondrian forests. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3758–3767. Curran Associates, Inc., 2017.

[MGS18] Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. Minimax optimal rates for mondrian trees and forests. *arXiv preprint arXiv:1803.05784*, 2018.

[MSVE06] Philipp Metzner, Christof Schütte, and Eric Vanden-Eijnden. Illustration of transition path theory on a collection of simple examples. *The Journal of chemical physics*, 125:084110, 09 2006.

[RT09] Daniel M Roy and Yee W. Teh. The mondrian process. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1377–1384. Curran Associates, Inc., 2009.