JUMP DIFFUSION OVER FEATURE SPACE FOR OBJECT RECOGNITION

Sébastien Gadat

SIAM J. CONTROL OPTIM.
Vol. 47, No. 2, pp. 904–935

JUMP DIFFUSION OVER FEATURE SPACE FOR OBJECT RECOGNITION*

SÉBASTIEN GADAT†

Abstract. We present a dynamical model for a population of tests in pattern recognition. Taking a preprocessed initialization of a feature set, we apply a stochastic algorithm based on an efficiency criterion and a Gaussian noise to recursively build and improve the feature space. This algorithm simulates a Markov chain which estimates a probability distribution $\mathbb{P}$ on the set of features. The features are structured as binary trees and we show that such random forests are a good way to represent the evolution of the feature set. We then obtain properties on the dynamic of the features space before applying this algorithm to practical examples such as face detection and microarray analysis. Lastly, we identify the weak limit of our process as a jump-diffusion process defined using the Skorokhod map over simplices.

Key words. Markov processes, jump-diffusion algorithms, stochastic approximation, Skorokhod map, feature selection, pattern recognition

AMS subject classifications. 60J75, 60H10, 62L20, 93E35, 62H30

DOI. 10.1137/060656759

1. Introduction.

In this paper, we study a learning algorithm designed for the construction of features in pattern recognition tasks. This algorithm is constructed as the stochastic approximation of a constrained jump-diffusion process, for which we provide an asymptotic analysis.

The algorithm originates from the following issue. A pattern recognition problem corresponds to the classification of *input data* into two or more classes. To solve this, an algorithm, called *a classifier*, is used to design a function which associates a class prediction to an observation of the input variables. There exists several types of competing approaches for building classifiers. Our goal is not to build a new one, but to optimize and improve the prediction by feeding the algorithm with the "best" input variables. Poorly informative variables indeed act like noise in a dataset and reduce the quality of learning algorithms, and fewer variables generally is a guarantee for robustness and reduced generalization ability. Also, a good understanding of the features which have more impact in the classification is critical in some subjects such as biology or text categorization: In microarray analysis, for example, it is important to identify the genes which express a pathology, and in spam detection, one can expect that the presence of some special chain of words enables better detection of nondesirable spam for some classical algorithms such as support vector machines (SVMs), classification trees (CART), or random forests, for instance.

Denote by $\mathcal{F}_0$ the initial set of variables; in the machine learning community, these are also called *features* and this is the word we will use in this paper. In several recent interesting applications, $\mathcal{F}_0$ is a large set, which contains hundreds, maybe thousands, of elements. Given that what we want to consider are not only a few useful elements of $\mathcal{F}_0$, but also useful combinations of them, we face an overwhelming space of possible explanatory variables that we need to explore in the selection process. Our goal will

*Received by the editors April 8, 2006; accepted for publication (in revised form) November 7, 2007; published electronically February 27, 2008.
http://www.siam.org/journals/sicon/47-2/65675.html
†Institut Mathématiques de Toulouse, Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse, France (sebastien.gadat@math.ups-tlse.fr).

be to provide a suboptimal stochastic approach to recursively explore and build new composed features space.

For simplicity, the only combinations we consider in this paper are products of variables. If we denote $\mathcal{P}(\mathcal{F}_0)$ parts of $\mathcal{F}_0$, we estimate, from a training set of samples, a subset $\mathcal{F}$ of $\mathcal{P}(\mathcal{F}_0)$ of "useful" variables, those which are the most important for the classification task. This set will be estimated as a jump process which will be denoted $(\mathcal{F}_t)_{t\geq 0}$.[1]

This jump process will in fact be driven by an auxiliary process, denoted $\mathbb{P}_t$, such that, at all times t, $\mathbb{P}_t$ is a probability measure supported by $\mathcal{F}_t$. We will define the pair $(\mathbb{P}_t, \mathcal{F}_t)$ as a jump-diffusion process, designed to maximize the efficiency of the variables belonging to $\mathcal{F}_t$. The practical implementation will be a stochastic approximation of this process. The primary goal of this paper is to provide a convergence study of both algorithms, the diffusion, and its approximation.

Since there are important motivations and applications from feature extraction, finding a universal alphabet of features has intrigued researchers in computer vision, and the construction of feature sets has become an active research domain. Direct methods, based on principal (or discriminant) components analysis (PCA) or independent components analysis (ICA) [24], can be used for reduction of dimension, but are not able to create new variables by composition and do not help us to easily understand the selection. Methods based on hierarchically structured variables have also been developed: Amit and Geman [2] and Fleuret and Geman [16] build recursive sets of binary decision trees using coarse to fine procedures. These recursive algorithms combine statistical and geometric properties to assemble discriminative sequential testing and reach very low rates of error in many image classification problems. But in most cases for these algorithms, the amount of features constructed is not limited and can regularly increase if the learning procedure is not stopped [25] and conclusions about optimization results are not inferred. Our approach to the feature space structure will be largely inspired from this sequential testing method, based on statistical correlation [16], entropy [20], or mutual information [15].

Finally, methods based on the optimization of margin of support vector machines have been recently proposed to make recursive feature elimination (RFE [31], [10]). These methods use exact expressions of margin separation of SVM and optimize weights on features to keep only those with high influence on the margin formula. This yielded interesting results on several classification tasks, such as pedestrian detection and cancer morphology classification, with a quantity of features. However, all these methods perform only backward selections from an initial fixed set of features, while adding new features obtained from composition of initial ones could improve efficiency of classification.

Building a set of features derived from an initial set, which contains a reduced number of variables, and complex combinations of variables, is at this point a largely open issue. Our objective will be to handle this problem using simultaneously upward and backward stochastic strategies. Such evolutionary algorithms are commonly used in the framework of regression, adding and removing variables with respect to any information criterion (AIC (Akaike information criterion), MSE (mean square error), etc.). We show here how one can think about similar ideas for the framework of pattern classification without using logistic regression, which may be considered somewhat artificial. Moreover, contrary to most variable selection procedures for linear analysis, we provide a theoretical background for our stochastic exploration of

[1] The construction will in fact be slightly more complex, involving trees instead of subsets.

features subsets dedicated to an optimization criterion. Lastly, our method can be used with any classification algorithm. This is an important point since for the commonly investigated classification problems, there does not exist a best classifier among all methods developed by statisticians.

Our paper will be organized as follows. In the next section, we give a precise description of our framework and introduce notation. The third section is devoted to the theoretical model of our jump-diffusion Markov process. Then, section 4 gives exact rules to enable features space to evolve over time. These rules use a Metropolis–Hastings evolution based on an energy $\mathcal{E}$ to be minimized over time. Section 5 gives dynamic properties of the model previously defined, whereas section 6 provides a statistical implementation and approximation method to simulate the jump-diffusion process of section 3. Finally, we conclude our work with experiments on synthetic data and real classification problems (face detection and leukemia classification) before giving future developments and applications to other situations in pattern recognition. Lastly, note that we choose to formalize our work in a continuous setting (Markov processes) rather than in a discrete form (Markov chains). One motivation will be to provide an understanding of the limit behavior of our exploration/extraction algorithm. Continuous setup will make it easier to precisely describe the dynamic of our constrained optimization method (section 5.1), while the formalism of the martingale problem and generator for Markov processes will be very powerful in identifying the asymptotic measure of our algorithm (Theorems 6.3 and 7.2). In fact, one can also describe our algorithm in a discrete setting (it is, moreover, the way it is numerically implemented) but the identification of the asymptotic behavior requires a time continuous approach with the use of the Skorokhod map. We thus choose to directly present the algorithm in the continuous framework to avoid some additional notation and repetitions.

2. Notation and settings.

2.1. Classes and features. We address the following pattern recognition problem. Given a large integer d which will denote the initial number of features, an input signal $I \in \mathbb{R}^d$ must be classified into a fixed number of classes denoted $\mathcal{C} = \{C_1, \ldots, C_N\}$. Each input I is described by its coordinates $(X^1(I), \ldots, X^d(I))$. $\mathcal{F}_0$ is the set of initial coordinates maps:

$$\mathcal{F}_0 = \left\{X^1, \ldots, X^d\right\}.$$

In our experiments, the X^j will be the projection to the jth component, it can be binary (values in $\{0,1\}$) or ternary (values in $\{-1,0,1\}$) for the image processing problem of section 8, or more generally, real-valued coordinates can also be considered (see the microarray analysis experiments of section 8).

A classification algorithm is a function which assigns a class C_i of the finite set $\mathcal{C}$ to an observed signal I. This function is estimated on the basis of a training set, which is a finite family of correctly labeled signals. However, for obvious dimensional complexity, the algorithm assumes a specific parametric form for the classification function: it could be, for instance, CART, SVMs, linear discriminant analysis, nearest neighbor, etc. In the two-class problem, the simplest classification rule is based on linear separation: Compute the sum $\beta_0 + \sum_{j=1}^{d} \beta_j X^j$, and decide for the first class if it is negative and for the second otherwise. The parameters $(\beta_0, \beta_j, j \in \{1 \ldots p\})$ are estimated so that this rule is as consistent as possible with the training data. Various definitions of the consistency criterion, variants on the functional form of the decision

rule and of the optimization algorithms, yield a very large family of classifiers, as provided by the literature. We will use in our applications an SVM with a linear kernel because of the generalization ability of this algorithm. Note that the previous linear separation rule assumes that *all* the features are used *as monomials*. Our goal in this context is twofold:

- Selection: Use less than the total family of features, which can be very large ($d > 1000$, for instance).
- Composition: Use more complex expressions than monomials by combining the features, and thus define one way to combine them.

This last point implies heuristic or stochastic exploration of the several compositions we can produce starting from $\mathcal{F}_0$: simplest ones are $X^j X^k, (j,k) \in \{1 \dots d)$, and $X^j X^k X^l, (j,k) \in \{1 \dots d\}$. One can see the exponential growth of the size of possible composition space, and our algorithm proposes a stochastic approach of this exploration step.

Example 2.1. Consider the following synthetic example that will be used first in the experiments section. We deal with 3 classes of signals described by 100 ternary features. We thus have $\mathcal{F}_0 = \{X^1, \dots, X^{100}\}$. One can imagine that these 3 classes behave differently on several subset of features $\mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_3$ (which may overlap or not) and follow exactly the same distribution on variables in $\mathcal{F}_0 \backslash \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$. This is the case for most signal processing situations, where some variables act as independent noise whatever the class of the signal is, although different statistic distributions are located on some other special set of variables for each corresponding class ($\mathcal{G}_1$ for C_1, $\mathcal{G}_2$ for C_2, and $\mathcal{G}_3$ for C_3).

We are interested in the problem of detecting interactions of features encoded in all $\mathcal{G}_i$, filtering out noisy features in $\mathcal{F}_0 \backslash \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$, and forming new compositional variables corresponding to each subset $\mathcal{G}_i$. We will provide more details on practical examples in section 8.

2.2. Composition of features. We introduce notation regarding the composition of features. Individual features from the original set will be denoted $\mathcal{F}_0$, while the set of features obtained at time t will be naturally named $\mathcal{F}_t$. In the definition of the jump diffusion, there will be many advantages in ensuring that the jumps are *reversible*. To obtain such a property, it will be necessary (see section 4.2) for each element of $\mathcal{F}_t$ to remember how it has been constructed. For this reason, we introduce trees on the set of features as follows.

To an elementary feature X^j in $\mathcal{F}_0$ we associate the elementary tree (and keep the same notation "X^j"):

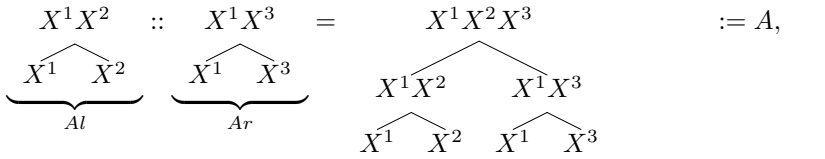
$$X^j := \begin{array}{c} X^j \\ \emptyset \quad \emptyset \end{array} . \tag{1}$$

A tree feature $\mathcal{A}$ is a binary tree such that each node contains a composition of elementary features, and terminal nodes (leaves) are elementary features of $\mathcal{F}_0$. Moreover, each nonterminal node in $\mathcal{A}$ must be the concatenation (union) of its descendants so that one can easily infer how any tree has been formed. The root of $\mathcal{A}$, denoted $r(\mathcal{A})$, is the main node associated to the tree. Tree features $\mathcal{A}, \mathcal{B}$ are aggregated with the construction rule "::"

$$\mathcal{A} :: \mathcal{B} = \begin{array}{c} r(\mathcal{A}) \cup r(\mathcal{B}) \\ \mathcal{A} \qquad \mathcal{B} \end{array} . \tag{2}$$

Note that we do not take into account any order or repetition of elementary features taken in $r(\mathcal{A}) \cup r(\mathcal{B})$.

Example 2.2. For instance, in the operation



we can reform left and right sons (Ar and Al) from A by cutting A's main node. It is manifest here that without this tree structure of features, the same composition will be

$$\underbrace{X^1X^2}_{Bl} :: \underbrace{X^1X^3}_{Bl} = X^1X^2X^3 := B$$

but we cannot directly obtain from B the way it has been formed since some sons could be $\{(X^1X^2);(X^1X^3)\}$ or $\{(X^2X^3);(X^2X^1)\}$.

To restrict the number of notations, we will keep again the notation $\mathcal{F}_0$ for the set of elementary trees over the initial set of variables created by operation (1). Similarly, $\mathcal{F}_t$ will be the set of features handled at time t by our algorithm. We will denote by $\mathbf{F}^\sharp$ the set of all trees over $\mathcal{F}_0$ defined by (1) using (2). Technically, $\mathcal{F}_t$ will be a jump process on $\mathbf{F}^\sharp$ (we will call them forests), and $\mathbb{P}_t$ will be a jump diffusion process with values in the set of probability distributions on $\mathbf{F}^\sharp$, which will be supported by a subset of the process $\mathcal{F}_t$.

We use the map $\mathcal{A} \to r(\mathcal{A})$ only for the computation of trees over input signals since each value of any tree $\mathcal{A}$ will naturally be defined on any signal I by

$$r(\mathcal{A})(I) = X^{i_1}(I) \times \cdots \times X^{i_p}(I)$$

if r is written as $r(\mathcal{A}) = X^{i_1} \dots X^{i_p}$.

2.3. Base classification algorithm $\mathbb{A}$. In this paper, we consider a classification algorithm, denoted $\mathbb{A}$, as a "black box" with the following functionalities. We assume that $\mathbb{A}$ can be conditioned by any subset $\omega \subset \mathbf{F}^\sharp$ of *active variables*. In training mode, $\mathbb{A}$ uses a database to build an optimal classifier $\mathbb{A}_\omega : \mathcal{I} \to \mathcal{C}$, such that $\mathbb{A}_\omega(I)$ depends only on variables $\omega(I)$. The test mode simply consists in the instantiation of $\mathbb{A}_\omega$ on a given signal of the test set.

We work with a randomized version of $\mathbb{A}$, in which the randomization is on the set of variables. This randomization of features spaces has been introduced by Amit and Geman [1] and Breiman [8] who build accurate random classifiers with very low dependence to outliers and noise. In the training phase, this works as follows: First, extract a collection $\{\omega^{(1)}, \dots, \omega^{(N)}\}$ of subsets of $\mathbf{F}^\sharp$, and build the classifiers $\mathbb{A}_{\omega^{(1)}}, \dots, \mathbb{A}_{\omega^{(N)}}$. Then, derive the classification in the test phase using a majority rule within these N classifiers. This final algorithm will be denoted $\bar{\mathbb{A}} = \bar{\mathbb{A}}(\omega^{(1)}, \dots, \omega^{(N)})$. In test mode, it is run with fixed $\omega^{(i)}$'s, which have been obtained in the learning phase.

In addition to being an auxiliary process that we use for variable selection, the probability $\mathbb{P}_t$ will also be used for sampling the $\omega^{(k)}$ in the construction of randomized algorithms. Note that the present paper focuses on the way to construct an automatic process creating the random subsets of $\mathbf{F}^\sharp$ and not designing the classification algorithm $\mathbb{A}$, for which we use standard procedures.

We will construct a process $(\mathcal{F}_t, \mathbb{P}_t)$, where $\mathcal{F}_t$ is a jump process over forests (subsets of $\mathbf{F}^\sharp$) and between jumps, and $\mathbb{P}_t$ is a diffusion process, constrained to the set of probabilities on $\mathcal{F}_t$, designed to optimize the performance of the classification algorithm. We start by describing the diffusion process. We will then consider the transition probabilities at jump times, both for $\mathcal{F}_t$ and $\mathbb{P}_t$.

3. Constrained diffusion.

Important notation. From now on, we will denote with capital letters the Markov process $(\mathcal{F}_t)$ among the forests; $(\mathbb{P}_t)$ will denote the Markov process among the probabilities although F (and F_1, $F_2, \ldots$) or P (P_1, $P_2, \ldots$) will be some possible realizations of these two processes. This distinction will be important to the understanding of settings of the next sections.

Description of the dynamic. Between jumping times, the probability will essentially evolve according to the diffusion

$$\mathrm{d}\mathbb{P}_t = -\nabla \mathcal{E}_{err}(\mathbb{P}_t)\mathrm{d}t + \sigma \mathrm{d}W_t, \tag{3}$$

where $\mathcal{E}_{err}(P)$ is a cost function measuring the quality of the classifier using variables sampled from P; this will be precisely defined in the following paragraph. Such a process classically stabilizes around probabilities P with low cost $\mathcal{E}_{err}$.

This process must, however, be modified in order to ensure that $\mathbb{P}_t$ is a probability supported by $\mathcal{F}_t$. If $\mathcal{F}$ is a subset of $\mathbf{F}^\sharp$, we denote by $\mathcal{H}_\mathcal{F}$ the hyperplane in $\mathbb{R}^\mathcal{F}$ of equation $\sum_{\delta \in \mathcal{F}} P(\delta) = 1$. Let $\pi_\mathcal{F}$ be the affine orthogonal projection onto $\mathcal{H}_\mathcal{F}$ (which is $\pi_\mathcal{F}(U) = U - \sum_\delta U(\delta)/|\mathcal{F}|$). We denote $\nabla^\mathcal{F} \mathcal{E}_{err}(P) = \pi_\mathcal{F} \nabla \mathcal{E}_{err}(P)$. We can restrict (3) to $\mathcal{H}_\mathcal{F}$ by replacing ∇ by $\nabla^\mathcal{F}$ and using a Brownian motion on $\mathcal{H}_\mathcal{F}$, or equivalently, using

$$\mathrm{d}\mathbb{P}_t = -\nabla \mathcal{E}_{err}(\mathbb{P}_t)\mathrm{d}t + \Sigma^{\mathcal{F}_t} \mathrm{d}W_t, \tag{4}$$

where W is a Brownian motion on $\mathbb{R}^{\mathbf{F}^\sharp}$ and $\Sigma^\mathcal{F} = \sigma \pi_\mathcal{F}$.

Denoting $\mathcal{S}_\mathcal{F}$ for the set of all such probability distributions on $\mathcal{F}$, we need to modify (4) to ensure that $\mathbb{P}_t$ belongs to $\mathcal{S}_{\mathcal{F}_t}$ at all times. This is done using a constrained diffusion process, which is here a reflected diffusion process:

$$\mathrm{d}\mathbb{P}_t = -\nabla \mathcal{E}_{err}(\mathbb{P}_t)\mathrm{d}t + \Sigma^{\mathcal{F}_t} \mathrm{d}W_t + \mathrm{d}Z_t,$$

where Z_t acts as a correction to ensure that the positivity constraints are satisfied at all times. This means that $d|Z_t|$ is positive only when $\mathbb{P}_t$ hits $\partial \mathcal{S}_\mathcal{F}$.

3.1. Cost function. We now define two costs functions for our system forest F +probability P. The first function $\mathcal{E}_{err}(P)$ measures the average performance of the classifier based on random feature selection according to P. The second measures a structural cost of the set of features F and does not depend on P. These two functions enable us to form the global cost for the pairwise process $(\mathcal{F}_t, \mathbb{P}_t)$.

3.1.1. Measuring the mean performance of $\mathbb{A}$: The energy $\mathcal{E}_{err}$. Consider a set of trees $F \subset \mathbf{F}^\sharp$ and a probability distribution P on $\mathbf{F}^\sharp$ supported by F. The algorithm $\mathbb{A}$ provides a different classifier $\mathbb{A}_\omega$ for each choice of a subset ω of k features $\omega = (\omega_1, \ldots, \omega_k) \subset F$. We let η be the classification error, $\eta(\omega) = \mathbf{P}(\mathbb{A}_\omega(I) \neq \mathcal{C}(I))$, which will be estimated by

$$g(\omega) = \hat{\mathbf{P}}(\mathbb{A}_\omega(I) \neq \mathcal{C}(I)),$$

where $\hat{\mathbf{P}}$ is the empirical probability on the training set. As we want to use a small number of features, we fix an integer k; the distribution $P^{\otimes k}$ corresponds to k independent trials with replacement with respect to the distribution P. We define the cost function $\mathcal{E}_{err}$ by

$$\mathcal{E}_{err}(P) = \mathbb{E}_{P^{\otimes k}} g(\omega) = \sum_{\omega \in F^k} g(\omega) P^{\otimes k}(\omega) = \sum_{\omega \in F^k} g(\omega) P(\omega_1) \dots P(\omega_k).$$

One can thus remark that minimizing $\mathcal{E}_{err}$ according to the control parameter P will drive us to a distribution with important weights on useful features for the classification (low error rate induced by $\mathbb{A}$).

3.1.2. Global cost function on (F, P): The energy $\mathcal{E}$. We now describe the global cost function, denoted by $\mathcal{E}$. It will take the form

$$\mathcal{E}(F, P) = \mathcal{E}_{err}(P) + \mathcal{E}_{struct}(F),$$

where $\mathcal{E}_{struct}$ is a structural energy on the forest. More precisely,

$$\mathcal{E}_{struct}(F) = \underbrace{\sum_{\mathcal{A} \in F} |\mathcal{A}|}_{\mathcal{E}_s^1(F)} - \underbrace{\sum_{\mathcal{A} \in F} \hat{I}(\mathcal{A}.g, \mathcal{A}.d)}_{\mathcal{E}_s^2(F)} \tag{5}$$

and $\hat{I}(\mathcal{A}.g, \mathcal{A}.d)$ is the empirical mutual information function between the left and right subtrees of $\mathcal{A}$. The first term $\mathcal{E}_s^1$ limits the size of the forest and comes from the minimum description length principle of information theory [28]. The last term $\mathcal{E}_s^2$ is of a compositional nature and favors the concatenation of correlated trees (or trees with high mutual information) [16]. Our goal is now to minimize $\mathcal{E}$ over the space $\mathbf{F}^\sharp \times \mathcal{S}_{\mathbf{F}^\sharp}$, which has a discrete component and a continuous one.

4. Jumps. We first introduce the notion of *weak reversibility* of a jump process since this property will have critical importance in the stochastic dynamic search $(\mathcal{F}_t, \mathbb{P}_t)$.

4.1. General rule. The time differences between jumps are assumed to be mutually independent, and independent from the rest of the process. Jumps occur as a Poisson process (interjump times are independent and identically distributed (i.i.d.) exponential). Coupled with the constrained diffusion process, this allows the inference algorithm to visit $\mathbf{F}^\sharp \times \mathcal{S}_{\mathbf{F}^\sharp}$. This accomodates the discrete nature of the problem. At jump times, the transitions $\mathcal{F}_t \to \mathcal{F}_{t+dt}$ will correspond to deletion, addition, or combination of elements of $\mathcal{F}_t$. Each of these rules will be designed using an accept/reject scheme (Hastings) as follows. We handle here the complete cost function, $\mathcal{E}(\mathcal{F}_t, \mathbb{P}_t)$ defined by (5). Below, we review some general notions on the Metropolis–Hastings method (this section may be skipped).

4.1.1. Generality on the Metropolis–Hastings algorithm. The situation is as follows: Let Ω be a measurable set with a measure m and let μ be a measure on Ω with density (also denoted μ) w.r.t. m. The Metropolis–Hastings transitions follow a two-step rule:

- From state $x \in \Omega$, first propose a state y with probability $Q_0(x, dy)$;
- then, accept the transition with a probability which is adjusted so that μ is invariant.

We assume the following property: For all $x \in \Omega$, there exists a measure ρ_x such that

(A1) $Q_0(x,.)$ has a density $q(x,.)$ w.r.t. ρ_x.
(A2) $q(x,y) > 0 \Leftrightarrow q(y,x) > 0$.
(A3) the measure $\rho_x(dy) \otimes m(dx)$ is symmetrical: For any function f on Ω^2,

$$\int_\Omega f(x,y)\rho_x(dy)m(dx) = \int_\Omega f(y,x)\rho_x(dy)m(dx).$$

The transition Q is then defined by

$$\begin{aligned} Q(x,dy) &= \min\left(\frac{\mu(y)q(y,x)}{\mu(x)q(x,y)},1\right) Q_0(x,dy) \\ &\quad + \left(1 - \int_\Omega \min\left(\frac{\mu(z)q(z,x)}{\mu(x)q(x,z)},1\right) Q_0(x,dz)\right) \mathbb{1}_x(dy). \end{aligned} \tag{6}$$

The distribution of two consecutive states is then $Q(x,dy)\otimes m(dx)$, and to ensure the reversibility we need to verify that it is symmetrical. But

$$\begin{aligned} Q(x,dy) \otimes m(dx) &= \min\left(\mu(y)q(y,x), \mu(x)q(x,y)\right) \rho_x(dy) \otimes m(dx) \\ &\quad + \left(1 - \int_\Omega \min\left(\mu(z)q(z,x), \mu(x)q(x,z)\right) \rho_x(dz)\right) \mathbb{1}_x(dy) \otimes m(dx). \end{aligned}$$

The second line takes the form $g(x)\mathbb{1}_x(dy)m(dx)$ and is obviously symmetric, although the first one is symmetric thanks to our assumption on ρ_x. Consequently, we need to give a transitions rule satisfying the previous assumptions (A1), (A2), and (A3) for our framework on weighted forests.

Remark 4.1 (necessity of weak reversibility). It is important here to underline why the building process of $(\mathcal{F}_t)$ must be weakly reversible (assumptions (A1), (A2), and (A3)). We can present at least two reasons for this imperative condition:

- First, note that our exploration process of $\mathbf{F}^\sharp$ has a stochastic nature and may be mistaken for some iteration because of the Metropolis–Hastings acceptance strategy. We thus need to cancel the decision taken at this step (assumption (A2)), and weak reversibility guarantees this possibility in only one reverse jump.
- Furthermore, the Metropolis–Hastings acceptance rate computation (6) involves the ratio $q(x,y)/q(y,x)$ because of assumptions (A1), (A3) applied to $q(x,.)$ and $q(y,.)$. Obviously, if the features are not structured as a tree, one cannot compute this ratio since we do not have from any set of variables x the unique pair of its antecedents.

4.1.2. Metropolis–Hastings transitions on weighted forests. We denote by m_F the Lebesgue measure on $\mathcal{S}_F$ and consider m as the global measure on $\mathcal{P}(\mathbf{F}^\sharp) \times \mathcal{S}_{\mathbf{F}^\sharp}$ defined by

$$m = \sum_{F \subset \mathbf{F}^\sharp} \mathbb{1}_F \otimes m_F,$$

which means that

$$\int f(F,P)dm(F,P) = \sum_{F \subset \mathbf{F}^\sharp} \int_{\mathcal{S}_F} f(F,P)dm_F(P).$$

Here, $\mathbb{1}_F$ is the Dirac measure at a forest F. Consider any forest F_1 and an element P_1 of $\mathcal{S}_{F_1}$. The transitions are defined as follows: Choose a new forest $F_2 \in V_{F_1}$, where V_{F_1} is the set of forests which are reachable in one jump, and then choose an element of $\mathcal{S}_{F_2}$ according to a probability which depends on F_1, F_2, and P_1. Assume that this probability has a positive density w.r.t. some measure denoted $\psi_{F_1,F_2}(P_1,.)$ on $\mathcal{S}_{F_2}$. This implies that the measures, w.r.t. which the densities of the transitions are computed, are

$$\rho_{F_1,P_1}(F_2,.) = \mathbb{1}_{V_{F_1}}(F_2)\psi_{F_1,F_2}(P_1,.),$$

where $\rho_{F_1,P_1} = \rho_x$ is the measure defined in the former paragraph. Therefore, we need to construct ρ, ψ, and a neighborhood $V_.$ in order to satisfy assumptions (A1)–(A3). We design in the next section transitions satisfying (A1) and (A2). Next, we will show that the symmetry requirement is true. Since in the framework of a weighted forest we have

$$m_{F_1}(dP_1)\rho_{F_1,F_2}(P_1,dP_2) = \mathbb{1}_{V_{F_1}}(F_2)\psi_{F_1,F_2}(P_1,.)m_{F_1}(dP_1),$$

it will be sufficient to establish

$$m_{F_1}(dP_1)\psi_{F_1,F_2}(P_1,dP_2) = m_{F_2}(dP_2)\psi_{F_2,F_1}(P_2,dP_1).$$

4.2. Transitions between forests. We first construct a set $\mathcal{T}$ of compositional rules before showing the *weak reversibility* (assumptions (A1), (A2), and (A3)) of our system. This set of transitions does not seem standard and is different from what is done in genetic algorithms. However, to satisfy the *weak reversibility* needed by the Metropolis sampling scheme, this set of transitions $\mathcal{T}$ will be necessary.

DEFINITION 4.2 (transition rules $\mathcal{T}$). *$\mathcal{T}$ is the set of applications from $\mathcal{P}(\mathbf{F}^\sharp)\times\mathcal{S}_{\mathbf{F}^\sharp}$ to $\mathcal{P}(\mathbf{F}^\sharp)\times\mathcal{S}_{\mathbf{F}^\sharp}$ formed by buddings, cuttings, suppressions, or rebirths. By $(F,P)\in\mathcal{P}(\mathbf{F}^\sharp)\times\mathcal{S}_{\mathcal{F}^\sharp}$ we enumerate the states which are reachable in one jump from (F,P). For convenience of notation, U will denote the set of active variables in F with their associated weights in P. The quantities p_b, p_c, p_s, and p_r will represent the nonnegative probability at each jump time of choosing budding, cutting, suppression, or rebirth. We first enumerate the* ***budding transitions:***

Transition	*Symbol*	*Antecedents*	*Changes in U*	*Probability*
Budding without suppression	$\mathcal{B}$	$(\mathcal{A}_1,p_1);(\mathcal{A}_2,p_2)$	*Add* $(\mathcal{A}_1::\mathcal{A}_2,p)$ *Change the weights:* $(\mathcal{A}_1,p_1-p+x)$ $(\mathcal{A}_2,p_2-x)$ *where* $p\sim\mathcal{U}_{[0;p_1+p_2]}$ $x\sim\mathcal{U}_{[p-p_1;p_2]}$	$p_b/4$
Budding with left suppression	$\mathcal{B}_l$	$(\mathcal{A}_1,p_1);(\mathcal{A}_2,p_2)$	*Add* $(\mathcal{A}_1::\mathcal{A}_2,p_1)$ *Leave* $(\mathcal{A}_2,p_2)$ *Remove* $(\mathcal{A}_1,p_1)$	$p_b/4$
Budding with right suppression	$\mathcal{B}_r$	$(\mathcal{A}_1,p_1);(\mathcal{A}_2,p_2)$	*Add* $(\mathcal{A}_1::\mathcal{A}_2,p_2)$ *Leave* $(\mathcal{A}_1,p_1)$ *Remove* $(\mathcal{A}_2,p_2)$	$p_b/4$
Budding with both suppressions	$\mathcal{B}_{lr}$	$(\mathcal{A}_1,p_1);(\mathcal{A}_2,p_2)$	*Add* $(\mathcal{A}_1::\mathcal{A}_2,p_1+p_2)$ *Remove* $(\mathcal{A}_1,p_1)$ *Remove* $(\mathcal{A}_2,p_2)$	$p_b/4$

We present next the ***cut transitions:***

Transition	*Notation*	*Antecedents*	*Changes in U*	*Probability*
Cut without creation	$\mathcal{C}$	$(\mathcal{A}_1 :: \mathcal{A}_2, p)$ $(\mathcal{A}_1, p_1); (\mathcal{A}_2, p_2)$	*Remove* $(\mathcal{A}_1 :: \mathcal{A}_2, p)$ *Change the weights:* $(\mathcal{A}_1, p_1 + p - x)$ $(\mathcal{A}_2, p_2 + x)$ *where* $x \sim \mathcal{U}_{[-p_2; p_1+p]}$	$p_c/4$
Cut with left creation	$\mathcal{C}_l$	$(\mathcal{A}_1 :: \mathcal{A}_2, p)$ $(\mathcal{A}_2, p_2)$	*Remove* $(\mathcal{A}_1 :: \mathcal{A}_2, p)$ *Add* $(\mathcal{A}_1, p)$ *Leave* $(\mathcal{A}_2, p_2)$	$p_c/4$
Cut with right creation	$\mathcal{C}_r$	$(\mathcal{A}_1 :: \mathcal{A}_2, p)$ $(\mathcal{A}_1, p_1)$	*Remove* $(\mathcal{A}_1 :: \mathcal{A}_2, p)$ *Add* $(\mathcal{A}_2, p)$ *Leave* $(\mathcal{A}_1, p_1)$	$p_c/4$
Cut with both creation	$\mathcal{C}_{lr}$	$(\mathcal{A}_1 :: \mathcal{A}_2, p)$	*Remove* $(\mathcal{A}_1 :: \mathcal{A}_2, p)$ *Add* $(\mathcal{A}_1, x)$ *Add* $(\mathcal{A}_2, p - x)$ *where* $x \sim \mathcal{U}_{[0;p]}$	$p_c/4$

Lastly, we have the ***suppression and rebirth transitions:***

Transition	*Notation*	*Antecedents*	*Changes in U*	*Probability*
Suppression	$\mathcal{S}$	$(\mathcal{A}, p)$	*Remove* $(\mathcal{A}, p)$ *Change the weights* $\forall (\mathcal{B}, q) \in U \Rightarrow (\mathcal{B}, q/(1-p))$	p_s
Rebirth	$\mathcal{S}$	$\mathcal{A} \in \mathcal{F}_0 \setminus F$	*Add* $(\mathcal{A}, x)$ *Change the weights* $\forall (\mathcal{B}, q) \Rightarrow (\mathcal{B}, q(1-x))$	p_r

With these former transition rules, it is now possible to establish the *weak reversibility conditions*.

PROPOSITION 4.3 (weak reversibility of $\mathcal{T}$). *Assumptions* (A1), (A2), *and* (A3) *are true under the dynamic of* $(\mathcal{F}_t, \mathbb{P}_t)$ *induced by* $\mathcal{T}$.

Proof. Take a forest F in $\mathcal{P}(\mathcal{F}^\sharp)$ and define V_F as the set of reachable forests using one (and only one) transition of $\mathcal{T}$. We first remark that if we enumerate all transitions between two forests, we have for any couples of forests (F_1, F_2):

$$F_2 \in V_{F_1} \iff F_1 \in V_{F_2}.$$

Roughly speaking, if one tree is created, cut, or deleted using $\mathcal{T}$, it is instantaneously possible to flashback and cancel this transition using another rule in $\mathcal{T}$. This point is also true if we study weights over forests. For instance, the inverse of budding without suppression is a cut without creation, and if

$$\{(\mathcal{A}_1, p_1); (\mathcal{A}_2, p_2)\} \longmapsto \{(\mathcal{A}_1, q_1 = p_1 - p + x); (\mathcal{A}_2, q_2 = p_2 - x); (\mathcal{A}_1 :: \mathcal{A}_2, q_3 = p)\},$$

one can see easily that q_1 takes all values in $[0; p_1 + p_2]$ and q_2 in $[0; p_1 + p_2]$ with, in addition, $q_1 + q_2 + q_3 = p_1 + p_2$. Consequently, cut without creation from such $\{(\mathcal{A}_1, q_1); (\mathcal{A}_2, q_2); (\mathcal{A}_1 :: \mathcal{A}_2, q_3)\}$ can reach the initial state. We can verify that this

point is true for all transitions given in the three former arrays while enumerating all possible transitions.

If we then denote by $\rho_{F,P}$ the uniform measure among reachable sets from (F,P) using $\mathcal{T}$, and by $q((F,P),.)$ the density of proposition law Q_0 defined in section 4.1.1, we naturally obtain that (A1) and (A2) are true.

We now study assumption (A3). Denote first (F_1,P_1) as a weighted forest and (F_2,P_2) as reachable from (F_1,P_1) using $\mathcal{T}$. We must compare $m_{F_1}(dP_1)\psi_{F_1,F_2}(P_1, dP_2)$ with $m_{F_2}(dP_2)\psi_{F_2,F_1}(P_2,dP_1)$. We must then number all transitions of $\mathcal{T}$ and verify the symmetrical relation. This point is more or less complicated according to the relation considered. For instance, take again the case of budding without creation (remember that m_{F_1} is the Lebesgue measure defined on the simplex S_{F_1}), and denote by N the length of vector $P_1=(p_1,p_2,\dots,p_N)$. Without loss of generality, we can suppose that we choose to bud trees 1 and 2 so that other weighted trees of F_1 remain unchanged. The length of P_2 is consequently $N+1$, and we have thus $P_2=(q_1,q_2,\dots q_N,q_{N+1})$.

Hence, to one side we have

$$m_{F_1}(dP_1)\psi_{F_1,F_2}(P_1,dP_2)=\prod_{i=1}^{N} m_{F_1}(dp_i)\otimes\prod_{i=3}^{N}\mathbb{1}_{p_i}(q_i)\otimes\mathcal{U}_{X^{p_1,p_2}}(q_1,q_2,q_{N+1}),$$

and $\psi_{F_1,F_2}(P_1,.)$ is a Dirac for all coordinates in P_1 which are not modified by the bud and

$$X^{p_1,p_2}=\left\{(a,b,c)\in\mathbb{R}_+^3 \quad | \quad a+b+c=p_1+p_2\right\}.$$

On the other hand, we can equally write the transition measure

$$m_{F_2}(dP_2)\psi_{F_2,F_1}(P_2,dP_1)=\prod_{i=1}^{N+1} m_{F_2}(dq_i)\otimes\prod_{i=3}^{N}\mathbb{1}_{q_i}(p_i)\otimes\mathcal{U}_{X^{q_1,q_2,q_{N+1}}}(p_1,p_2).$$

The symmetrical claim is satisfied since, for all measurable functions f on $\mathcal{S}_{F_1}\times\mathcal{S}_{F_2}$,

$$\begin{aligned}
&\left\langle m_{F_1}(dP_1)\psi_{F_1,F_2}(P_1,dP_2);f\right\rangle\\
&\quad=\iint_{\mathcal{S}_{F_1}\times\mathcal{S}_{F_2}} f(p_1,\dots,p_N,q_1,\dots,q_{N+1})m_{F_1}(dP_1)\psi_{F_1,F_2}(P_1,dP_2)\\
&\quad=\int\prod_{i=1}^{N}dp_i\int_0^{p_1+p_2}dp\int_{p-p_1}^{p_2}dx f(P_1,p_1-p+x,p_2-x,p_3,\dots,p_N,p)\\
&\quad=\int\prod_{i=1}^{N+1}dq_i\int_{-q_2}^{q_1+q_{N+1}}dx\int_{q_1+q_{N+1}-x}^{q_2+x} f(P_2,q_1+p-x,q_2+x,q_3,\dots,q_N)\\
&\quad=\left\langle m_{F_2}(dP_2)\psi_{F_2,F_1}(P_2,dP_1);f\right\rangle.
\end{aligned}$$

A similar change of variables can be done for all other types of transitions of $\mathcal{T}$, and we can conclude that the symmetrical assumption (A3) is also true. ∎

4.3. Decision steps of the Markovian dynamic of jumps. Taking a jump time t_j and any state of our process $(\mathcal{F}_{t_j},\mathbb{P}_{t_j})$, we use rules taken from $\mathcal{T}$ to modify $\mathcal{F}_{t_j}$ and $\mathbb{P}_{t_j}$ to $\mathcal{F}_{t_j+\mathrm{d}t}$ and $\mathbb{P}_{t_j+\mathrm{d}t}$. There are exactly three steps for the choice of which transition of $\mathcal{T}$ is applied.

Step 1. We first choose which kind of transition is proposed in $\mathcal{T}$ (bud, cut, suppression, or rebirth) according to the probability distribution specified in the last columns of arrays of section 4.2.

Step 2. When the rule is chosen, select the trees to which the rule is applied. One can make this decision regardless of whether it is dependent on $\mathbb{P}_{t_j}$. The simpler method is to choose uniformly among all trees in $\mathcal{F}_{t_j}$ or in $\mathcal{F}_0 \backslash \mathcal{F}_{t_j}$.

Step 3. Accept (or not) the transition according to a differential energy criterion,

$$Q((\mathcal{F}_{t_j}, \mathbb{P}_{t_j}); (F, P)) = \min\left(1, e^{\mathcal{E}(\mathcal{F}_{t_j}, \mathbb{P}_{t_j}) - \mathcal{E}(F,P)} \times R\right), \tag{7}$$

where

$$R = \frac{Q_0((F,P); (\mathcal{F}_{t_j}, \mathbb{P}_{t_j}))q((F,P); (\mathcal{F}_{t_j}, \mathbb{P}_{t_j}))}{Q_0((\mathcal{F}_{t_j}, \mathbb{P}_{t_j}); (F,P))q((F_{t_j}, \mathbb{P}_{t_j}); (F,P))}.$$

The computation of the first step is easy with a discrete probability distribution on the rules constituting $\mathcal{T}$. At Step 2, the choice of which trees to apply the rule can depend on the distribution $\mathbb{P}_{t_j}$. The main idea is to favor trees with high probability for *budding* and low probability for *cuts*. Trees selected for *rebirth* are chosen uniformly in the feature space $\mathcal{F}_0 \setminus \mathcal{F}_{t_j}$. More details can be found in [17].

5. Existence of the jump-diffusion process. From the beginning of this section, special attention will be dedicated to the indexing of our random processes. They will be described first (up to and including section 6) using a continuous setting $(\mathcal{F}_t, \mathbb{P}_t)$, which looks somewhat artificial since in section 7 the algorithm works in a discrete framework with $(\mathcal{F}_n, \mathbb{P}_n)$. Morevover, the description of the continuous setup will be much more complicated than the discretized one mainly owing to the projection term in the set of probability distributions.

Actually, the asymptotic behavior of the Markov chain $(\mathcal{F}_n, \mathbb{P}_n)$ will be presented following a classical scheme of compactness/identification. The compactness is studied in section 7, although the identification of the stationary measure in section 7 will critically use uniqueness of the stationary measure for the continuous process. Thus, the heavy use of the Skorokhod map is highly motivated by this asymptotic study since the identification of the stationary measure is easily deduced from the Markov generator of the process. Since we will need this continuous approach for this last identification, we directly describe the learning process in a continuous setting. Lastly, the weak limit of our discrete Markov chain will be the continuous reflected jump diffusion, and the description of this last process will need the Skorokhod map.

But the Skorokhod map can be skipped to intuitively make the understanding easier in this section, and one can replace the continuous processes by the discretized ones using a simple convex projection to keep $\mathbb{P}_n$ in a set of probability measures.

5.1. Existence of the reflected diffusion between jump times. We work in this section with fixed $\mathcal{F}_t = F$ of size S and discuss the existence of a Markovian reflected diffusion process which drives the evolution in the absence of jumps:

$$\mathrm{d}\mathbb{P}_t = -\underbrace{\nabla}_{(=\nabla^F)} \mathcal{E}_{err}(\mathbb{P}_t)\mathrm{d}t + \sigma \mathrm{d}W_t + \mathrm{d}Z_t. \tag{8}$$

The construction of solutions of (8) relies on the Skorokhod map Γ associated to $\mathcal{S}_F$ and a set of unit vectors $d_c(x)$ for all x on the boundary $\partial\mathcal{S}_F$. This map associates

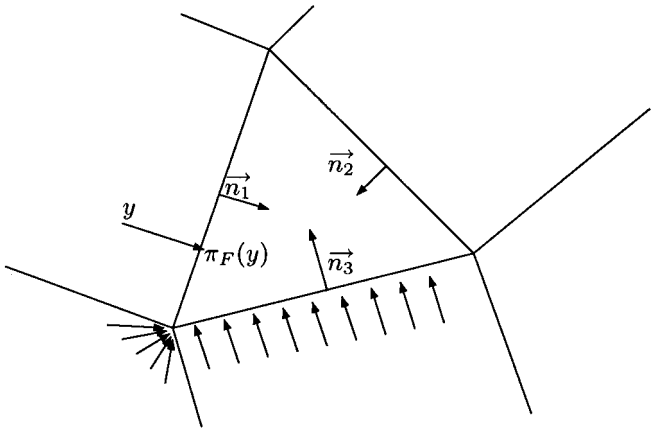


FIG. 1. *Directions of reflection vectors for x in $\partial\mathcal{S}_F$.*

to any càdlàg trajectory a constrained càdlàg trajectory that satisfies some boundary conditions based on $d_c(.)$. We refer to [12] and [13] for further precise technical definitions on this construction. For the sake of completeness, we provide only the constraint vectors we used. The important fact is thus that Γ will exist and define a Lipschitz function on càdlàg trajectories.

DEFINITION 5.1 (directions of constraints $d_c(.)$). *We call $\overrightarrow{n_i}$ the unit vectors belonging to the hyperplane supporting $\mathcal{S}_F$ that normally enter the ith face of the simplex. The directions of constraints are given by*

$$\forall x \in \partial\mathcal{S}_{\mathcal{F}}, \qquad d_c(x) = \left\{ \gamma = \sum_{i \,|\, x_i = 0} \alpha_i \overrightarrow{n_i} \quad | \quad \alpha_i \geq 0, \|\gamma\| = 1 \right\}.$$

These directions of reflection on $\partial\mathcal{S}_F$ can be expressed easily in a different way as follows.

PROPOSITION 5.2 (directions of constraints $d_c(.)$). *For any point x in $\partial\mathcal{S}_F$, vectors $d_c(x)$ coincide exactly with the sets of unit vectors:*

$$d_c(x) = \left\{ \overrightarrow{\gamma} \textit{ with } \|\overrightarrow{\gamma}\|_2 = 1 \quad | \quad \exists\, y \in \mathcal{H}_F \quad y - \pi_F(y) = \alpha\gamma, \quad \alpha \leq 0, x = \pi_F(y) \right\},$$

where π_F is the natural convex projection on the simplex $\mathcal{S}_F$.

Figure 1 summarizes this natural property. One can remark that directions $d_c(x)$ are strongly connected to convex projections on $\mathcal{S}_F$: they correspond exactly to the unitary vectors that can be used to project any exterior point to $\mathcal{S}_F$. In stochastic approximation algorithms, it is the usual way of introducing convex constraints. This yields a set of possible callback vectors shown in Figure 1.

The Skorokhod map allows us to formalize the reflected diffusion (8) as a system of integral equations:

$$\begin{cases} X_t = \mathbb{P}_0 - \displaystyle\int_0^t \nabla\mathcal{E}_{err}(\mathbb{P}_s)\mathrm{d}s + \sigma \mathrm{d}W(s), \\ \mathbb{P}_t = \Gamma(X)_t. \end{cases}$$

This system is equivalent to the stochastic differential equation (8) [3, 12]. Strong (and obviously weak) existence and uniqueness of such an integral system is standard using a fixed point method [30], Lipschitz regularity of Γ, and Lipschitz continuity of the drift $\nabla\mathcal{E}_{err}$. Indeed, for $\omega \in F^k$ and $\delta \in F$, denote by $C(\omega, \delta)$ the number of occurences of δ in ω:

$$C(\omega, \delta) = |\{i \in \{1, \dots, k\} \quad | \quad \omega_i = \delta\}|.$$

Since the drift term is polynomial in variables $P(\delta)$, it is obviously Lipschitz continuous. Its exact expression is for any $P \in \mathcal{S}_F$; then

$$\forall \delta \in F, \qquad \nabla_P \mathcal{E}_{err}(\delta) = \sum_{\omega \in F^k} \frac{C(\omega, \delta) P^{\otimes k}(\omega)}{P(\delta)} g(\omega). \tag{9}$$

We can thus infer the following result.

THEOREM 5.3 (existence and uniqueness of (8)). *Let $(\Omega, \mathcal{T}, Q)$ be a probability space with an increasing filtration $\mathcal{T}_t$, let W_t be standard Brownian motion on $\mathbb{R}^{|\mathcal{F}|}$, and let $\mathbb{P}$ be a random variable $\mathcal{T}_0$-measurable. Then there exists a unique pair $(\mathbb{P}_t, Z_t)$ $\mathcal{T}_t$-measurable satisfying (8) with*

1.
$$\forall T > 0, \qquad |Z_T| < +\infty \quad \mathcal{T}_T\text{-a.s.}$$

2.
$$\forall t \geq 0, \qquad |Z|_t = \int_0^t \mathbb{1}_{\mathbb{P}_s \in \partial \mathcal{S}_{\mathcal{F}}} d|Z|_s.$$

3.
$$\forall t \geq 0, \qquad dZ_t \in d_c(\mathbb{P}_t).$$

Proof. See [30, Chapter 5]. ∎

5.2. Existence of the complete process. Since the jump time is a Poisson process independent of the rest, the previous result, combined with the Markov transitions at jump times, trivially implies the existence and uniqueness of the complete jump-diffusion process. An example of the evolution of such a stochastic process is summarized in Figure 2 using a sequence of four different simplices and jumping times. Each simplex corresponds to a features space while the a.s. continuous trajectory points to the evolution of our extraction method $\mathbb{P}_s$. We represent here one reflection on $\mathcal{S}_{\mathcal{F}_{t_{s_2}}}$ and several jumps between several (i.e., 4) simplices. Note that if it is possible to jump from $\mathcal{S}_{\mathcal{F}_{t_{s_2}}}$ to $\mathcal{S}_{\mathcal{F}_{t_{s_3}}}$, it is equally possible to jump from $\mathcal{S}_{\mathcal{F}_{t_{s_3}}}$ to $\mathcal{S}_{\mathcal{F}_{t_{s_2}}}$ (weak reversibility).

We will denote by Φ the stationary solution of the stochastic differential equation of the reflected jump diffusion based on reflected diffusion on each simplex and jumps between subspaces of features. This solution is defined as follows.

DEFINITION 5.4 (stationary solution Φ). *Let $(\Omega, \mathcal{T}, Q)$ be a probability space with an increasing filtration $\mathcal{T}_t$. Let $(W_t)_{t\geq 0}$ be a standard Brownian motion on $\mathbb{R}^{|\mathbf{F}^\sharp|}$ and $(N_t)_{t\geq 0}$ be a Poisson jump process, both adapted to filtration $\mathcal{T}$. Suppose likewise that W and N are independent. We call $\Phi = (\mathbb{P}, \mathcal{F})$ the stationary solution of the*

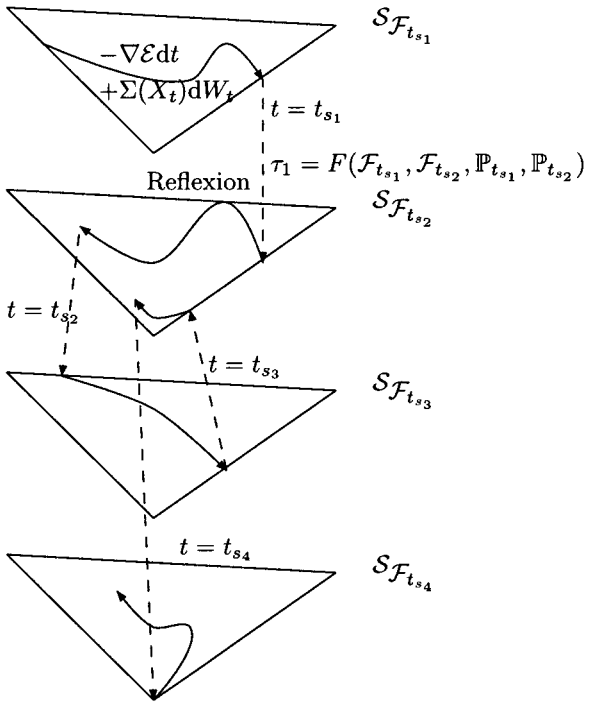


FIG. 2. *General form of the stochastic jump-diffusion process.*

stochastic differential equation with jumps:

$$d\begin{pmatrix}\mathbb{P}_t\\ \mathcal{F}_t\end{pmatrix} = -\begin{pmatrix}\nabla^{\mathcal{F}_t}\mathcal{E}(\mathbb{P}_t)dt + \Sigma^{\mathcal{F}_t}dW_t + dZ_t\\ 0\end{pmatrix} + \int_{\mathcal{F}\subset\mathbf{F}^\sharp, \mathbb{P}\in\mathcal{S}_{\mathcal{F}}} Q\left[\begin{pmatrix}\mathcal{F}_t\\ \mathbb{P}_t\end{pmatrix};\begin{pmatrix}\mathcal{F}\\ \mathbb{P}\end{pmatrix}\right] N\left(d\begin{pmatrix}\mathbb{P}\\ \mathcal{F}\end{pmatrix};dt\right).$$

6. Dynamical properties of the algorithm. In this section, we briefly summarize the dynamical properties of the unique solution of the reflected jump-diffusion process. Our goal is to prove that the process is positive recurrent with a unique stationary measure, given by the the density

$$\mu(F,P) = \frac{e^{-\mathcal{E}(F,P)}}{Z}, \tag{10}$$

with respect to the measure on $\mathcal{P}_{\mathbf{F}^\sharp} \times \mathcal{S}_{\mathbf{F}^\sharp}$,

$$m = \sum_{F\subset\mathbf{F}^\sharp} \mathbb{1}_F \otimes m_F.$$

We first give the expression of the infinitesimal generator of the process, then establish that $(\mathbb{P}_s, X_s)_{s\geq 0}$ is positive recurrent and prove that its stationary measure is the Gibbs field μ associated to $\mathcal{E}$.

6.1. Infinitesimal generator of $(\mathcal{F}_s, \mathbb{P}_s)_{s\geq 0}$. Our Markov process is a combination of a reflected diffusion process and a jump process. A generic function on $\mathcal{P}(\mathbf{F}^\sharp) \times \mathcal{S}_{\mathbf{F}^\sharp}$ can be decomposed as

$$f(F,P) = \sum_{F'\subset \mathbf{F}^\sharp} \mathbb{1}_{F'}(F) f_{F'}(P).$$

The generator A of this process can be decomposed into a diffusion part and a jump part, yielding $Af = A^d f + A^j f$, with

$$A^d f(P,F) = -\langle \nabla_P^F \mathcal{E}_{err} | \nabla_P^F f_F \rangle + \frac{1}{2}\Delta^F f_F(x)$$

and

$$A^j f(P,F) = \int_{\mathcal{P}(\mathbf{F}^\sharp)\times \mathcal{S}_{F^\sharp}} [f_{F'}(P') - f_F(P)] Q\left[(F,P),(F',P')\right] \mathrm{d}m(F',P'),$$

where Q is the transition probability at jump times.

6.2. Positive recurrence. The main result of this section uses the positive definite nature of Σ_F on $\mathcal{H}_F$ and a result of [23, Theorem 1, section 7] ensuring a positive recurrence of the reflected process (without any jump). For any reachable simplex $\mathcal{S}_F$, the unique process solution of

$$\mathrm{d}\mathbb{P}_t = -\nabla \mathcal{E}_\epsilon(\mathbb{P}_t)\mathrm{d}t + \sigma \mathrm{d}W_t + \mathrm{d}Z_t$$

satisfies the following for all compact sets $S \subset \mathcal{S}_\mathcal{F}$ of nonnegative Lebesgue measure $\lambda(S) > 0$ (if P_p is the probability of one event for which initialization of our process is taken at point p):

$$\inf_{p\in\mathcal{S}_\mathcal{F}} P_p\left[\tau_S \leq 1\right] > 0, \tag{11}$$

where

$$\tau_S = \inf\left\{t/\mathbb{P}_t \in S\right\}.$$

Equation (11) means that starting at any point p of simplex $\mathcal{S}_F$, one can reach S in less time than with a probability strictly positive. This implies in particular (see [4, Theorem 2.8]) the positive recurrence of (8) without a jump, and the existence of a unique invariant measure. The extension of the results to the jump-diffusion process now requires only the following fact. Denote

$$p_{_{F,T}}(S) = \inf_{p\in\mathcal{S}_F} P_p\left[\tau_S \leq T\right]$$

for $S \subset \mathcal{S}_F$, where τ_S is the hitting time of S. We have the following result.

COROLLARY 6.1.

$$p_{F,T}(S) > 0,$$

and the general reflected process with jumps is positive recurrent.

Proof. The jumps have been designed so that there exists an integer N such that for any F and F', and for any $P \in \mathcal{S}_F$, the transition $(F,P) \to F'$ in N steps has a probability strictly larger than some positive constant, η. Since the probability of making N jumps before T, and no other jump after, is strictly positive, the result is a direct consequence of the positive recurrence of the process without a jump. ∎

6.3. Invariant measure of the process. Properties of the invariant measure can be inferred from the positive recurrence of $(\mathbb{P}_t, \mathcal{F}_t)_{t\geq 0}$. First, note that for any initialization $(\mathbb{P}_0, \mathcal{F}_0)$ of the process, the family of occupation measures $(\mu_t)_{t\geq 0}$, defined by

$$\mu_t(A) = \frac{1}{t}\int_0^t P_{(\mathbb{P}_0,\mathcal{F}_0)}\left[(\mathbb{P}_s, \mathcal{F}_s) \in A\right] \mathrm{d}s,$$

is tight and any weak limit is an invariant measure of $(\mathbb{P}_t, X_t)_{t\geq 0}$ since the process is Feller–Markov. Uniqueness is derived from the nondegeneracy of the diffusion of the process into each simplex and the weak reversibility between each simplex of $(\mathbb{P}_t, \mathcal{F}_t)$. Identification of this measure from the characterization of [14] is used, for example, in [29].

We use here the well-posedness of the associated martingale problem. Define first the core of this generator as

$$\mathbb{D} = \left\{ f = \sum_{F\subset\mathcal{F}^\sharp} \mathbb{1}_{\mathcal{S}_F}(P) f_F(P) \quad | \quad \forall F \subset \mathcal{F}^\sharp \quad \forall P \in \partial S_F \qquad \nabla f_F(P) = 0 \right\}.$$

We start noticing that for any function in $\mathbb{D}$, the mean effect of generator A with distribution μ given by (10) is null.

PROPOSITION 6.2. *Assume f is an element of $\mathbb{D}$; then we have*

$$\int Af \mathrm{d}\mu = 0.$$

Proof. This result is proved by integration by parts, using the Neumann conditions on each simplex $\mathcal{S}_F$, where $F \subset \mathbf{F}^\sharp$, the Ostrogradski formula, and the stability equation on the transition acceptance threshold (7). Similar arguments can be found in [29]. ∎

We are now able to prove the next theorem.

THEOREM 6.3. *The Gibbs field μ given by* (10) *is the unique invariant measure of the global reflected jump-diffusion process, and the martingale problem associated to A on $\mathbb{D}$ is well-posed.*

Proof. We first apply Echeverria's theorem (see [14, Theorem 9.17, Chapter 9]) to show that μ is stationary. Denote by E the compact set $\{(F, P) \quad F \subset \mathbf{F}^\sharp, P \in \mathcal{S}_F\}$; note first that $\mathbb{D}$ is dense in $\mathcal{C}(E)$ by the Uryshon lemma applied in each simplex $\mathcal{S}_F$. Now, A satisfies the positive maximum principle on $\mathbb{D}$ (A is a classical jump-diffusion generator) and the measure μ satisfies

$$\forall f \in \mathbb{D}, \qquad \int_E Af d\mu = 0.$$

Consequently μ is stationary for A. Since A satisfies the maximum principle, A is dissipative on $\mathcal{C}(E)$ and E is separable. Denote then by ν a measure on E; we can apply the result of [14, Theorem 4.1, Chapter 4] to conclude that uniqueness holds for the martingale problem (A, ν) and every solution of the martingale problem is Markov. The martingale problem is well-posed on $\mathcal{C}(E)$, every solution of the martingale problem is a weak solution of the stochastic differential equation of jump diffusion, and μ is the unique stationary distribution of $(\mathcal{F}_t, \mathbb{P}_t)$. ∎

7. Stochastic approximations. We now address the computational part of the algorithm, which is not trivial because the drift term involves a sum over an untractable number of terms. Fortunately, this sum can be interpreted as an expectation, which allows us to replace it by a stochastic approximation of Robbins–Monro type. Before passing to the drift term, we first address the time discretization issues.

7.1. Time discretization. To solve (8), we use a time discretization scheme with a discretization step α,

$$\forall n \in \mathbb{N}, \qquad \mathbb{P}_{n+1} = \mathbb{P}_n - \alpha \nabla^{\mathcal{F}} \mathcal{E}_{err}(\mathbb{P}_n) + \sqrt{\alpha}\sqrt{\sigma} \mathrm{d}\xi_n + \mathrm{d}z_n,$$

where $\mathrm{d}\xi_n$ is a centered normal $|\mathcal{F}|$ dimensional vector and $\mathrm{d}z_n$ is the smaller vector that is added to make $\mathbb{P}_{n+1} \in \mathcal{S}_{\mathcal{F}}$. In other words,

$$\forall n \in \mathbb{N}, \qquad \mathbb{P}_{n+1} = \pi_{\mathcal{F}} \left(\mathbb{P}_n - \alpha \nabla^{\mathcal{F}} \mathcal{E}_{err}(\mathbb{P}_n) + \sqrt{\alpha}\sqrt{\sigma} \mathrm{d}\xi_n \right).$$

However, the computational issue comes from the gradient of $\mathcal{E}_{err}$, given in (9), which requires a sum over all ω in $\mathcal{F}^p$. This is an untractable sum, since $|\mathcal{F}|$ is typically thousands and p hundreds. However, it can be replaced by the stochastic approximation defined in the next section.

7.2. Stochastic differential equation method for approximation. Stochastic approximation can be seen as noisy discretizations of stochastic differential equations ([26]). They are generally expressed under the form

$$X_{n+1} = X_n + \alpha_n F(X_n, \zeta_{n+1}) + \sqrt{\alpha_n}\sqrt{\sigma}\xi_n + \alpha_n z_n + \alpha_n^2 T_n, \tag{12}$$

where X_n is the current state of the process, ζ_{n+1} a random perturbation, ξ_n a random perturbation of known distribution, z_n a random variable designed to ensure the constraints, and T_n a secondary error term. If the distribution of ζ_{n+1} depends only on the current value of X_n, then one defines an average drift $X \mapsto G(X)$ by

$$G(X) = \mathbb{E}[F(X, \zeta)|X],$$

and (12) can be shown to evolve similarly to the stochastic differential equation: $\mathrm{d}X_t = G(X)\mathrm{d}t + \sqrt{\sigma}\mathrm{d}w_t + \mathrm{d}z_t$, in the sense that the trajectories coincide when $(\epsilon_n)_{n\in\mathbb{N}}$ goes to 0 (a more precise statement is given below).

To implement our reflected diffusion equations (8) in this framework, we need to design a random variable d_n (identified as $F(X_n, \zeta_n)$ in (12)) such that

$$\mathbb{E}\left[d_n\right] = -\nabla^{\mathcal{F}} \mathcal{E}_{err}(\mathbb{P}_n) = -\Pi_{\overrightarrow{\mathcal{H}_{\mathcal{F}}}} \left[\nabla \mathcal{E}_{err}(\mathbb{P}_n)\right], \tag{13}$$

where $\Pi_{\overrightarrow{\mathcal{H}_{\mathcal{F}}}}$ is the vectorial projection on the hyperplane supporting $\mathcal{S}_{\mathcal{F}}$. We will then define

$$\mathbb{P}_{n+1} = \mathbb{P}_n - \alpha_n d_n + \sqrt{\alpha_n}\sqrt{\sigma}\xi_n + \mathrm{d}z_n = \pi_{\mathcal{F}} \left(\mathbb{P}_n - \alpha_n d_n + \sqrt{\alpha_n}\sqrt{\sigma}\xi_n \right).$$

From (9), we obtain

$$\nabla \mathcal{E}_{err}(\mathbb{P})(\delta) = \mathbb{E}_{\mathbb{P}^{\otimes k}} \left[\frac{C(\omega, \delta) g(\omega)}{\mathbb{P}(\delta)} \right].$$

Using the linearity of the projection $\Pi_{\overrightarrow{\mathcal{H}_{\mathcal{F}}}}$, we get

$$\Pi_{\overrightarrow{\mathcal{H}_{\mathcal{F}}}} \left(\nabla \mathcal{E}(\mathbb{P})\right)(\delta) = \mathbb{E}_{\mathbb{P}^{\otimes k}} \left[\Pi_{\overrightarrow{\mathcal{H}_{\mathcal{F}}}} \left(\frac{C(\omega, .) g(\omega)}{\mathbb{P}(.)} \right)(\delta) \right].$$

Consequently, following (13), it is now natural to define the approximation term of the reflected diffusion (8) by

$$d_n(\delta) = \Pi_{\overrightarrow{\mathcal{H}_{\mathcal{F}}}}\left(\frac{C(\omega_n,.)}{\mathbb{P}_n(.)}\right)(\delta),$$

where the set of k features ω_n is a random variable extracted from $\mathcal{F}$ with law $\mathbb{P}_n^{\otimes k}$.

This results in the following numerical simulation scheme:

1. Step 0: Initialization: Set $\mathbb{P}_0 = \mathcal{U}_{\mathcal{F}}$.
2. Step n: Draw a sample ω_n in $\mathcal{F}^k$ with respect to $\mathbb{P}_n^{\otimes k}$.
3. Step n: Compute $g(\omega_n)$.
4. Step n: Update $\mathbb{P}_{n+1}$ with

$$\begin{aligned}\mathbb{P}_{n+1} &= \pi_{\mathcal{F}}\left(\mathbb{P}_n - \alpha_n\left[\frac{C(\omega_n,.)}{\mathbb{P}_n}\right] + \sqrt{\alpha_n}\sqrt{\sigma}d\xi_n\right) \\ &= \mathbb{P}_n - \alpha_n\left[\frac{C(\omega_n,.)}{\mathbb{P}_n}\right] + \sqrt{\alpha_n}\sqrt{\sigma}d\xi_n + dz_n,\end{aligned} \tag{14}$$

where $-\alpha_n C(\omega_n,.)/\mathbb{P}_n$ is the approximated value of $-\nabla\mathcal{E}_{err}(\mathbb{P}_n)$ and $d\xi_n$ is a centered normal $|\mathcal{F}|$ dimensional vector.

To simulate the stochastic approximation of the jump-diffusion algorithm, (14) must be combined with transitions of $(\mathcal{F},\mathcal{P})$ at jump times. This results in the following new complete scheme:

1. Step 0: Initialization: Set $\mathbb{P}_0 = \mathcal{U}_{\mathcal{F}_0}$. Sample the first jumping time t^1 with an exponential distribution, set $t = 0$, and set $n = 0$.
2. Step j ($j \geq 1$): While $t < t^j$, run the previous discretization scheme (for the reflected diffusion), t being iteratively computed by $t = \alpha_0 + \cdots + \alpha_n$.
3. When $t > t_j$: Update $\mathcal{F}_t$ and $\mathbb{P}_t$ according to the Markov transition rules.
4. Compute the next jump time with t^{j+1} by adding an exponential variable to t^j and return to 2.

7.3. Weak convergence of the numerical scheme. In the following paragraphs, we will define $(\mathbb{P}^n(t)_{t\geq 0})_{n\in\mathbb{N}}$ as a sequence of continuous processes that interpolates the behavior of the discrete sequence of $(\mathbb{P}_n)_{n\in\mathbb{N}}$.

7.3.1. Interpolated approximations. Following classic notation of [26], we set up the time parameter τ_n as

$$\tau_n = \sum_{i\leq n}\alpha_i,$$

and set up the map m permitting the association of continuous time and discrete iteration as

$$m(t) = \sup_{\tau_n\leq t}\{n\in\mathbb{N}\}.$$

Given that the jth jump occurs at time ν_j, we construct its values according to the distribution $Q((\mathcal{F}_{\nu_j-},\mathbb{P}_{\nu_j-}),.)$ to obtain $(\mathcal{F}_{\nu_j},\mathbb{P}_{\nu_j})$. It is thus possible to define the discrete jump term in the discrete case as

$$q_j = \mathbb{P}_{m(\nu_j)+1} - \mathbb{P}_{m(\nu_j)},$$

which corresponds to the term we add to compute the jump from ν_j- to ν_j.

We now define the sequence of right continuous interpolation processes $(\mathbb{P}^n(t))_{t\geq 0}$ initialized at $\mathbb{P}_n$.

DEFINITION 7.1 (processes $(\mathbb{P}^n(.), Y^n(.), W^n(.), Z^n(.))_{n\in\mathbb{N}}$). *We define the processes* $(\mathbb{P}^n, Y^n, W^n, Z^n)$ *valued in* $\mathbb{R}^{\mathbf{F}^\sharp}$ *by*

$$\forall n \in \mathbb{N}, \quad \forall t \in \mathbb{R}_+, \qquad Y^n(t) = \sum_{i=n}^{m(\tau_n+t)} \alpha_i y_i,$$

where the term y_i *satisfies*

$$\forall \delta \in \mathcal{F}, \qquad y_i(\delta) = -\frac{C(\omega_i, \delta) g(\omega_i)}{\mathbb{P}_i(\delta)} \quad \text{if} \quad \mathbb{P}_i(\delta) \neq 0 \quad \text{and} \quad y_i(\delta) = 0 \quad \text{if} \quad \mathbb{P}_i(\delta) = 0.$$

Likewise, we define

$$W^n(t) = \sum_{i=n}^{m(\tau_n+t)} \sqrt{\alpha_i} d\xi_i,$$

where $d\xi_i$ *is considered as an element of* $\mathbb{R}^{\mathbf{F}^\sharp}$,

$$Z^n(t) = \sum_{i=n}^{m(\tau_n+t)} dz_i.$$

Finally,

$$\mathbb{P}^n(t) = \mathbb{P}_n + Y^n(t) + W^n(t) + Z^n(t) + \sum_{\tau_n \leq \nu_j \leq \tau_n + t} q_j.$$

With these definitions, it is obvious that $\mathbb{P}^n$ is a process on $\mathcal{S}_{\mathbf{F}^\sharp}$, which is right continuous with left limits (in the space $\mathcal{D}$ of càdlàg trajectories). To get theoretical convergence results on these sequence of processes, we will now classically choose (α_n) such that $\sum \alpha_i = \infty$ and $\sum \alpha_i^2 < \infty$ (see [5], [26], for instance).

7.3.2. Convergence of $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$. We will show that the family of processes $(\mathbb{P}^n(.), Y^n(.), W^n(.), Z^n(.))_{n\in\mathbb{N}}$ is weakly compact in the space $\mathcal{D}$. The associated topology on this space is derived from the Skorokhod distance [6], [26] and we consider weak convergence of trajectories of $\mathcal{D}([0;\infty[)$.

THEOREM 7.2. *The processes* $(\mathbb{P}^n, Z^n)_{n\in\mathbb{N}}$, *which are stepwise constant, weakly converge toward the unique invariant solution of the stochastic differential equation without jumps and* $(\mathbb{P}_n, \mathcal{F}_n)_{n\in\mathbb{N}}$ *converges toward the stationary measure* μ.

The proof of Theorem 7.2 includes three steps: First, prove the tightness of the family $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$, then identify the unique possible weak limit, and finally show the convergence toward the stationary measure μ.

7.3.3. Tightness. To show tightness, we use the following criterion.

THEOREM 7.3 (see [26], [6]). *Let* X^n *be a sequence in* $\mathcal{D}$; $(X^n)_{n\in\mathbb{N}}$ *is tight iff*

1. *for any time* T *and* $\epsilon > 0$, *there exist an integer* n_0 *and a real* K *satisfying*

$$\forall n \geq n_0, \qquad P\left[\sup_{t\leq T} |X^n(t)| \geq K\right] \leq \epsilon. \tag{15}$$

2.

$$\forall \epsilon > 0, \qquad \lim_{\delta \mapsto 0} \limsup_{n} P\left[w'_{X^n}(\delta) \geq \epsilon\right] = 0. \tag{16}$$

We establish successively (15) and (16) for our family of processes $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$. The next proposition shows that (15) is true for $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$ and consequently guarantees the tightness of the family $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$.

PROPOSITION 7.4. *The sequence of processes* $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$ *satisfies* (15).

Proof. The result is obvious for $\mathbb{P}^n$, since it is compactly supported. To get the result for $(Y^n)_{n\in\mathbb{N}}$, we define the sequence $\tilde{y}_n$ as

$$\tilde{y}_n = y_n - \underbrace{\mathbb{E}_{\mathbb{P}_n}[y_n]}_{=h_n}.$$

Fix any real time T and a real number $\epsilon > 0$. We define the sequence of processes

$$\tilde{Y}^n(t) = \sum_{i=n}^{m(\tau_n+t)} \alpha_i \tilde{y}_i.$$

Since $\mathbb{E}_{\mathbb{P}_n}[y_n]$ is bounded by M, the first tightness criterion is true for processes H^n:

$$H^n(t) = \sum_{i=n}^{m(\tau_n+t)} \alpha_i h_i,$$

and we study the sequence of $(\tilde{Y}^n)_{n\in\mathbb{N}}$. Now the sum M_p^n given by

$$M_p^n = \sum_{i=n}^{n+p} \alpha_i \tilde{y}_i$$

is a martingale for the filtration generated by $\mathbb{F}_p^n = \sigma(\mathbb{P}_i, \xi_i, w_{i-1}, i \leq n+p)$. We can use Doob's inequality to show that

$$P\left(\sup_{q\leq p} |M_q^n| > K\right) \leq \frac{1}{K}\mathbb{E}\left(|M_p^n|\right).$$

Now,

$$\mathbb{E}\left(|M_p^n|\right) \leq \sum_{i=n}^{p} \alpha_i \mathbb{E}[|\tilde{y}_i|] \leq \sup_i \mathbb{E}[|\tilde{y}_i|] \sum_{i=n}^{p} \alpha_i.$$

Finally, $\mathbb{E}[|\tilde{y}_i|] = \mathbb{E}\left(\mathbb{E}[|\tilde{y}_i||\mathbb{F}_i^n]\right)$, and $\mathbb{E}[|\tilde{y}_i||\mathbb{F}_i^n]$ is bounded by $2M$. We have $\sum_{i=n}^{p} \alpha_i \leq T$ and we can deduce from these upper-bounds that

$$\lim_{K\mapsto\infty} P\left(\sup_{q\leq p} |M_q^n| > K\right) = 0.$$

The fact that

$$\lim_{K\mapsto\infty} \sup_{n\in\mathbb{N}} P\left[\sup_{t\leq T} |W^n(t)| \geq K\right] = 0$$

is standard and can be found in [26], [17]. Finally, since $Z^n = \mathbb{P}^n - Y^n - W^n$, $(Z^n)_{n\in\mathbb{N}}$ obviously satisfies (15). ∎

We must now establish condition (16) to achieve tightness of $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$.

PROPOSITION 7.5 (condition (16)). *Each of the processes* $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n\in\mathbb{N}}$ *satisfies* (16).

Proof. We first establish (16) for $(Y^n)_{n\in\mathbb{N}}$. Note that

$$\mathbb{E}\left[Y^n(t+s) - Y^n(t)\right] = \sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} \alpha_i \mathbb{E}\left[\mathbb{E}\left[y_i | \mathbb{F}_0^i\right]\right].$$

Then, use the fact that the expectations of y_k are bounded by M to obtain

$$\mathbb{E}\left[|Y^n(t+s) - Y^n(t)|\right] \leq Ms.$$

We thus conclude that (16) is true for $(Y^n)_{n\in\mathbb{N}}$ using the Markov inequality. The argument is standard to get a similar result for $(W^n)_{n\in\mathbb{N}}$ by Doob's inequality (see [26]). The jump component involved by terms q_j defines also a sequence of processes:

$$J^n(t) = \sum_{\tau_n \leq \nu_j \leq \tau_n + t} q_j.$$

Inequality (16) for $(J^n)_{n\in\mathbb{N}}$ is here clearly satisfied since jumps occur exponentially as each term q_j is bounded. Consequently, we have

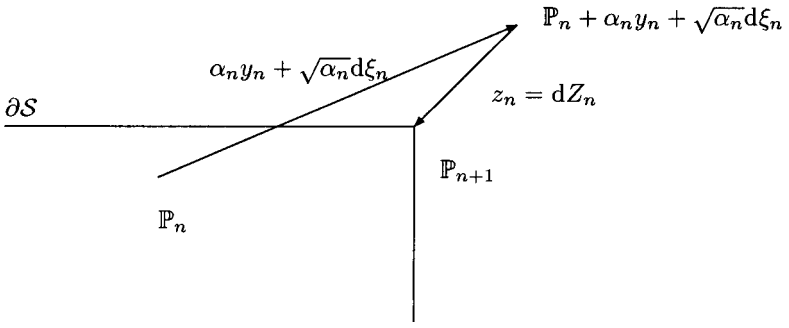
$$\limsup_n P[w'_{J^n}(\delta) \geq \epsilon] = o(\delta).$$

To deal with the processes $(Z^n)_{n\in\mathbb{N}}$, it is important to note that

$$|Z^n(t+s) - Z^n(t)| \leq C \sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} |\alpha_i y_i + \sqrt{\alpha_i}\mathrm{d}\xi_i|,$$

since

$$|z_i| \leq C|\alpha_i y_i + \sqrt{\alpha_i}\mathrm{d}\xi_i|. \tag{17}$$



Using inequality (16) for $(Y^n)_{n\in\mathbb{N}}$ and $(W^n)_{n\in\mathbb{N}}$, and (17), we obtain the second tightness inequality needed on the processes $(Z^n)_{n\in\mathbb{N}}$. The conclusion is immediate for $(\mathbb{P}^n)_{n\in\mathbb{N}}$. ∎

7.3.4. Proof of Theorem 7.2. We end the proof of Theorem 7.2 using compactness of the trajectories. Note first that if $(\mathbb{P}^n, Y^n, W^n, Z^n)$ is weakly convergent toward $(\mathbb{P}, Y, W, Z)$, then $(\mathbb{P}, Y, W, Z)$ is a solution of the reflected jump diffusion Φ initialized to the weak limit of $(\mathbb{P}^n(0), Y^n(0), W^n(0), Z^n(0))$ using the same argument of [26, Theorem 2.3].

While replacing $(\mathbb{P}^n, \mathcal{F}^n)$ by $\mathbb{P}^n$ and taking any sequence extracted from $(\mathbb{P}^n)_{n\in\mathbb{N}}$, we note $(N_k)_{k\in\mathbb{N}}$ this extraction procedure and show that $(\mathbb{P}^{N_k})_{k\in\mathbb{N}}$ is weakly convergent to the unique invariant measure μ. Denote by ν_∞ the weak limit of $(\mathbb{P}^{N_k}(0))_{k\in\mathbb{N}}$; it is then sufficient to show that for any measurable function ϕ,

$$\mathbb{E}_{\nu_\infty}\phi = \mathbb{E}_\mu \phi.$$

Denote by P_ν^t the law of our process at time t initialized by measure ν, since μ is the unique stationary measure we have for any compact set of measures K:

$$\forall \nu \in K \quad \forall \epsilon > 0, \quad \exists T > 0 \quad \forall t \geq T, \qquad \Big| \int \phi(y) \mathrm{d}P_\nu^t - \int \phi(y) \mathrm{d}\mu(y) \Big| \leq \epsilon. \tag{18}$$

Taking ϵ strictly positive and applying (18) to the family of measures K formed by the law of $(\mathbb{P}^{N_k})_{k\in\mathbb{N}}$, which is tight and thus compact, we find T such that

$$\forall t \geq T, \qquad \Big| \int \phi(y) \mathrm{d}P_\nu^t - \int \phi(y) \mathrm{d}\mu(y) \Big| \leq \epsilon.$$

Now, if ν'_∞ is the weak limit of the sequence of processes $(\mathbb{P}^{N_k}(. - T))_{k\in\mathbb{N}}$, which is also the weak limit of $(\mathbb{P}(\tau_{N_k} - T))_{k\in\mathbb{N}}$, we have

$$\begin{aligned}
\Big| \int \phi(y) \mathrm{d}\nu_\infty(y) - \int \phi(y) \mathrm{d}\mu(y) \Big| \leq & \Big| \int \phi(y) \mathrm{d}\nu_\infty(y) - \mathbb{E}\left[\phi\left(\mathbb{P}(\tau_{N_k})\right)\right] \Big| \\
& + \Big| \mathbb{E}\left[\phi\left(\mathbb{P}(\tau_{N_k})\right)\right] - \int \phi(y) \mathrm{d}P_{\nu'_\infty}^T(y) \Big| \\
& + \Big| \int \phi(y) \mathrm{d}P_{\nu'_\infty}^T(y) - \int \phi(y) \mathrm{d}\mu(y) \Big|.
\end{aligned}$$

Making $N_k \longmapsto \infty$, then $\tau_{N_k} \longmapsto \infty$, and under our hypotheses on T, ν_∞, and ν'_∞, we obtain

$$\Big| \int \phi(y) \mathrm{d}\nu_\infty(y) - \int \phi(y) \mathrm{d}\mu(y) \Big| \leq \epsilon.$$

Finally, we conclude that $\nu_\infty = \mu$ and this fact ensures that $(\mathbb{P}^n)_{n\in\mathbb{N}}$ and $(\mathbb{P}^n(0))_{n\in\mathbb{N}}$ weakly converge toward μ.

8. Experiments. We present here three experiments. The first one is a synthetic mixture model, and we compare our result with standard algorithms. The other databases are real problems on image processing and microarray data. In these last two cases, we use Fisher rule selection, random forest selection (see [8]), foward/backward selection, and OFW (optimal feature weighting) (see [18]) to draw comparisons with our method. In each of these cases, the number of selected features is computed using an internal cross-validation step.

8.1. Synthetic data.

8.1.1. Description of the database. We first test our algorithm on a simple synthetic example. We consider $f = 100$ ternary variables ($|\mathcal{F}| = 100$) and three classes (similar results can be obtained with more classes and variables). We let $I \in \{-1;0;1\}^{100}$ and let $\mathcal{G}$ be a subset of $\mathcal{F}$. We define the probability distribution $\mu(\,;\mathcal{G})$ on $\mathcal{I}$ to be the one for which all X^j in $\mathcal{G}$ are independent, $X^j(I)$ follows a uniform distribution on $\{-1;0;1\}$ if $X^j \notin \mathcal{G}$, and $X^j(I) = 1$ if $X^j \in \mathcal{G}$. We model each class by a mixture of such a distribution, including a small proportion of noise. More precisely, for a class C_i, $i = 1, 2, 3$, we define

$$\mu_i(I) = \frac{q}{3}\left(\mu(I;\mathcal{G}_i^1) + \mu(I;\mathcal{G}_i^2) + \mu(I;\mathcal{G}_i^3)\right) + (1-q)\mu(I;\emptyset),$$

with $q = 0.9$ and

$$\begin{aligned}
&\mathcal{G}_1^1 = \{X^1;X^3;X^5;X^7\}, \qquad \mathcal{G}_1^2 = \{X^1;X^5\}, \qquad \mathcal{G}_1^3 = \{X^3;X^7\},\\
&\mathcal{G}_2^1 = \{X^2;X^4;X^6;X^8\}, \qquad \mathcal{G}_2^2 = \{X^2;X^4\}, \qquad \mathcal{G}_2^3 = \{X^6;X^8\},\\
&\mathcal{G}_3^1 = \{X^1;X^4;X^8;X^9\}, \qquad \mathcal{G}_3^2 = \{X^1;X^8\}, \qquad \mathcal{G}_3^3 = \{X^4;X^9\}.
\end{aligned}$$

We sample with this mixture model enough data to obtain well-conditioned statistical problems. We expect our learning algorithm to put large weights on features that compose the sets $\mathcal{G}_i^j$ and to filter out the other noisy ones. The algorithm $\mathbb{A}$ we use in this case is a p nearest neighbor classification algorithm, with distance given by

$$d(I_1, I_2) = \sum_j \mathbb{1}_{X^j(I_1) \neq X^j(I_2)}.$$

This synthetic example is interesting because it makes it possible to compute the exact gradient of $\mathcal{E}$ for small values of M and $k = |\omega|$. See [17] and [18] for more details on this experiment when the set of features is fixed.

8.1.2. Results and comparisons with existing methods.

OFW and jump algorithm. We compare first the reflected diffusion (OFW of [18]) with our jump algorithm. Performances obtained with the jump algorithm are better than the ones without any jump as shown in Figure 3. The trees constructed by our algorithm are deeper than elementary ones since the mean depth achieved by our algorithm is 3. We compute the mean occupation measure of each tree in the process $\mathcal{F}_t$ as

$$\mu_t(\mathcal{A}) = \frac{1}{t}\int_0^t \mathbb{1}_{\mathcal{A} \in \mathcal{F}_{t_s}}\,\mathrm{d}s.$$

We can then infer from this measure the importance of a tree while looking at the real numbers $\mu_t(\mathcal{A})$. We rank the nodes of these trees by decreasing importance of $\mu_t(\mathcal{A})$ and we give the main roots detected by our algorithm below:

$$\{X^2;X^4\},\ \{X^1;X^5\},\ \{X^4;X^9\},\ \{X^1;X^8\},\ \{X^6;X^8\},\ \{X^3;X^7\},\ X^1,\ X^4,\ X^8.$$

It is important to remark that the nodes selected by our jump-diffusion algorithm are very similar to the sources $\mathcal{G}_i^j$, while the favored nodes are those which are *reusable features*.

One can, however, consider using standard feature selection techniques such as anova coupled with the logistic regression method or the more recent random forests feature selection.

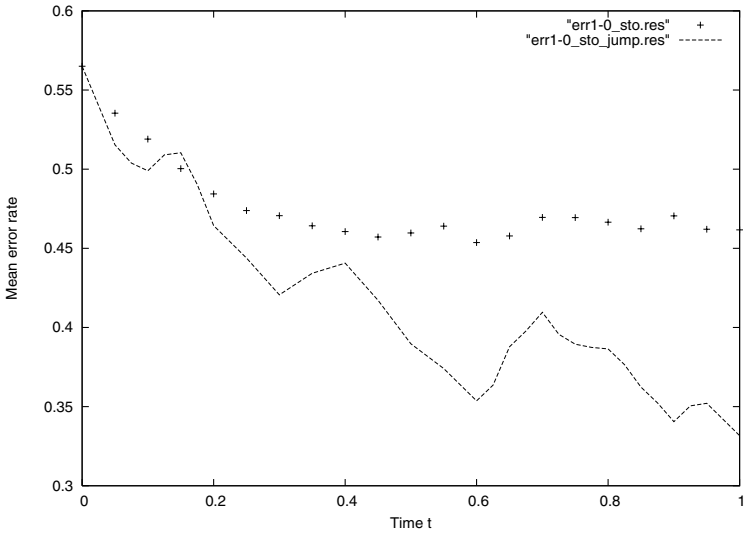


FIG. 3. *Evolution of the mean error rate for the reflected diffusion (crosses) and the reflected jump diffusion (dashed line).*

Backward selection, random forest, and Fisher selection. We run first a ternary (three classes) logistic regression coupled with a backward selection based on the anova criterion. In this particular case, the selected features cannot be composed and interactions with features are completely missed. We logically obtain the subset of selected variables $\left\{X^1; X^2; X^3; X^4; X^5; X^6; X^7; X^8; X^9\right\}$ with small p-value. This result is not surprising and is coherent with the selected features of the OFW [18] (diffusion algorithm without the jump process). Note also that the selected features here are the same while running the random forest selection method and we are convinced that in this simple example, several other classical criteria, such as PLS (partial least square method), AIC, etc., achieve the same result. Lastly, we remark that we do not run a forward selection method with the logistic regression because of the high number of features, which make this greedy method numerically costly.

Learning composition: The forward/backward selection. Next, we use the classical forward/backward selection method combined with logistic regression [21] since other feature selection methods, such as random forest, do not provide any composed features. In this very simple example (there are "only" 100 variables although typical real applications will use thousands of variables), the computational time to run this forward/backward selection is much more important (it takes several hours to stabilize the model). In addition to each singleton X^i, we obtain all subsets $\mathcal{G}_i^j$ given in the description of the way we construct our synthetic example.

Comparison. To conclude this section on the synthetic data, we observe that many other feature selection algorithms achieve the identification of useful variables.

Only one of them (the forward/backward method coupled with detection of interactions) can also compose features. This method has an important numerical cost. If this method succeeds in locating the interactions between features, it does not provide a selection as small as our method does.

Moreover, the main drawback in this framework is that the forward/backward criterion can be performed only with a sufficiently large database (we need to have

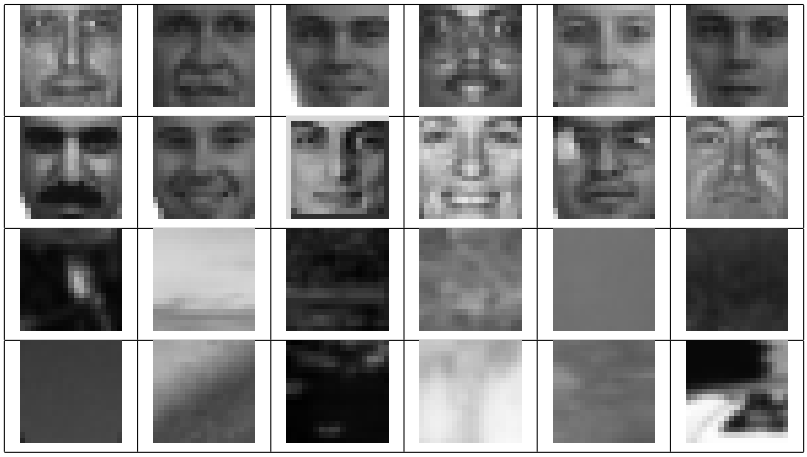


FIG. 4. *Sample of images taken from* [27] *database.*

more observations than initial number of variables). This point is very annoying in some real applications such as microarray analysis, where standard situations are described with thousands of variables for less than 100 observations.

Lastly, the forward/backward algorithm is dedicated only to very special classification algorithms such as logistic regression, although we can apply our approach to every classification algorithm $\mathbb{A}$. It is an important point too since there does not exist a universal classifier that beats all other algorithms. It can thus be helpful to run our meta-jump algorithm to the more appropriate $\mathbb{A}$ regarding the database which is studied.

8.2. Face recognition.

8.2.1. Description of the database. We use in this section the face database from [27], which contains 19×19 grayscale images. The elementary features in $\mathcal{F}_0$ are simply edge detectors constructed by Amit and Geman in [2]. The initial number of elementary features in $\mathcal{F}_0$ is nearly 2000. The number of observations in this database is 7000 in the training set and 23,000 in the test set. Figure 4 presents some examples of images taken in this database.

8.2.2. Results and comparisons.

OFW and jump algorithm. Efficiency of the reflected diffusion (OFW algorithm) is already described in [18]. In this paper, our approach permits largely improved error rates on the same datasets and we can easily give an interpretation of features constructed by our jump-diffusion process. To illustrate these advantages, we can plot first in Figures 5 and 6 the evolution of the number of trees selected by our algorithm with time t.

The decreasing of the number of trees is consequently important since starting with almost 2000 features, we reduce the amount of variables to below 800. Even if this number seems to be strictly decreasing in Figure 5, this is not the case if we "zoom" the evolution of the cardinal $t \longmapsto |\mathcal{F}_t|$, as shown in Figure 6.

Moreover, by using a linear SVM and a voting procedure with the subsets $\omega^{(i)}$ extracted with the process $\mathbb{P}_t$, we obtain a null false positive rate (images taken from the font class are perfectly classified) and the global misclassification rate is improved

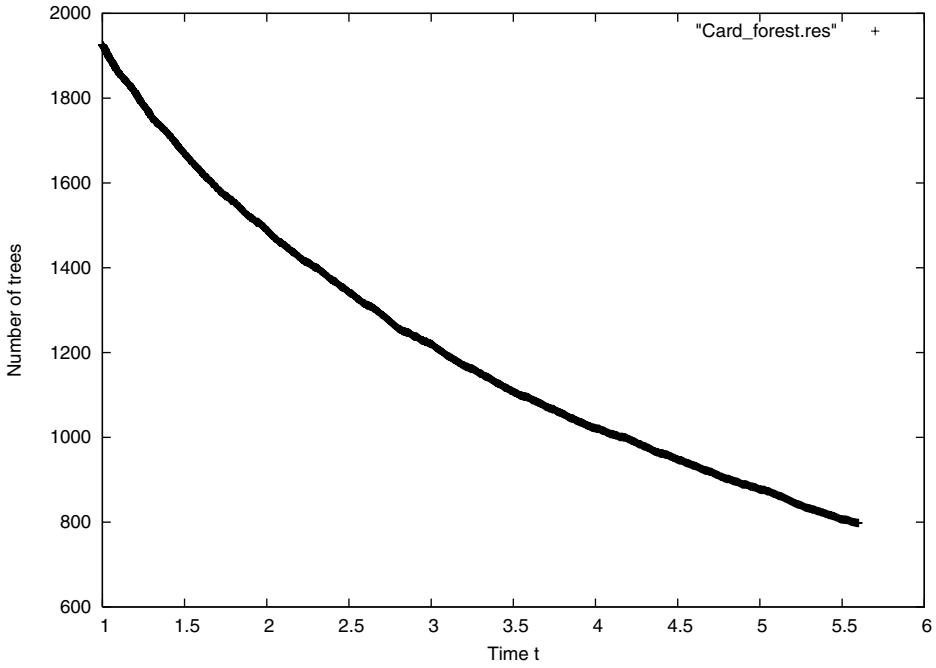


FIG. 5. *Evolution of the number of trees in the forest $\mathcal{F}_t$ with time for the face recognition problem.*

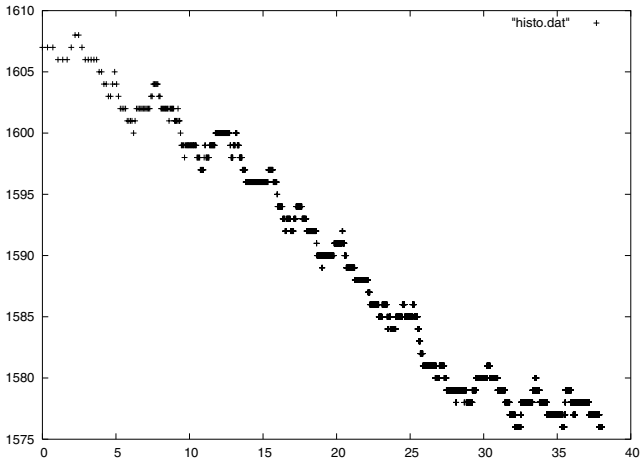


FIG. 6. *Microscopic evolution of the number of trees in the forest $\mathcal{F}_t$ for the faces experiment.*

since we get 1.2% misclassified test samples using only 650 features (3.5% error rate with the OFW approach without jump and the same amount of features).

Comparison with random forest, Fisher selection, and forward/backward selection. Without features composition, the random forest classifier provides more than 1000 useful features and gives a general misclassification rate of 1%; note also that this rate has been achieved using 1000 trees in the random forest. The Fisher selection method combined with a linear SVM algorithm yields an error rate of 4% with a selection

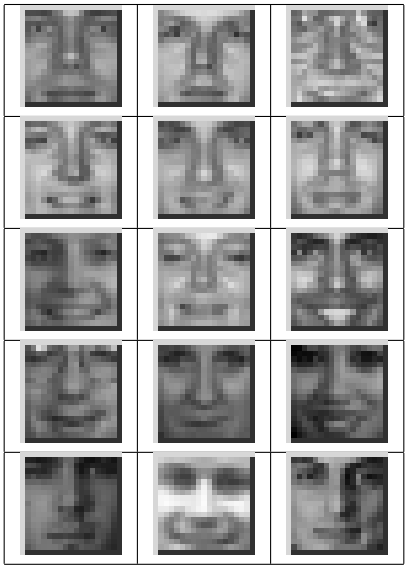


FIG. 7. *Representation of the main aggregation of edge detectors selected by our process.*

of more than 1000 features. Lastly, the logistic regression algorithm combined with forward/backward selection and composition performs poorly (more than 1000 useful features and 12% of misclassified signals). It seems here that the classifier $\mathbb{A}$ used by the forward/backward (logistic regression) is not adequate for this database. This illustrates the fact that it is important to have a meta-algorithm to select features in order to apply any classifier which seems adapted to the problem. It is not the case with the forward/backward selection method.

In this case, the better global misclassification rate is obtained using the random forest selection method. Our method obtains good results too since only 1.2% of signals are misclassified. Lastly, the selection obtained using our jump algorithm is much more compact than the one obtained by random forest.

Selected features. We show in Figure 7 the main composition of edges selected by our process of jump diffusion. The important fact is that complex features (as well as elementary ones) are constructed and used by our algorithm. It is this point that permits us to obtain the perfect false positive rate since these complex compositions of features filter out the background images.

8.3. Leukemia microarray classification.

Description of the database. Finally, we benchmark our selection of features on the standard leukemia cancer dataset available online from the NCI.[2] Data are preprocessed and transformed into a collection of 3859 genes of 72 leukemia samples. They are divided into 47 samples of Acute Lymphoblastic Leukemia (ALL) and 25 samples of Acute Myeloblastic Leukemia (AML). As we cannot provide a simple meaning of concatenation of real variables, we only permit suppression ($\mathcal{S}$) and rebirth ($\mathcal{R}$) of some genes in $\mathcal{F}_t$. As this database does not contain any train or test sets, we estimate the misclassification rate using a tenfold cross-validation method. The cross-validation method is a good way to estimate performances of our algorithm [7].

[2]National Cancer Institute, http://www.cancer.gov.

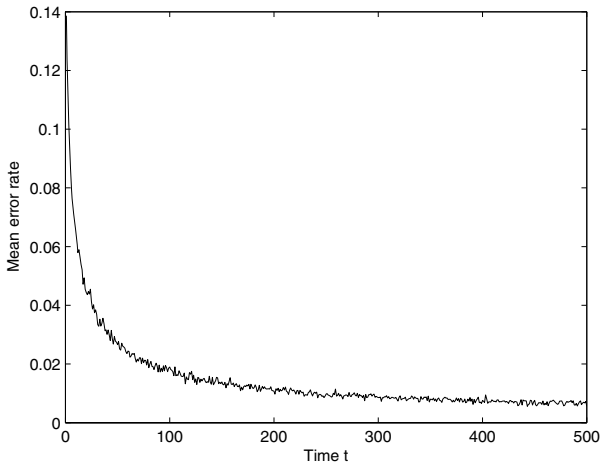


FIG. 8. *Mean error rate on the training set of the ALL–AML database with time t.*

TABLE 1
Rate of misclassified samples using several number of genes selected by our algorithm.

Number of genes	OFW	Reflected jump diffusion	Random forest	F-test
4-9	6.9%	4.8%	4%	5.5%
10-19	5.5%	4.3%	3.8%	4.2%
20-24	4.1%	2.1%	4.5%	4.1%
25-45	3.5%	0.9%	4.5%	4.8%

Comparison with random forest, Fisher selection, and forward/backward selection. After a learning procedure, we then ranked genes by a decreasing importance criterion based on the probability distribution $\mathbb{P}_t$. For the jump-diffusion method, we do not run a tenfold cross validation because of the time of computation needed by this method, and we then employ a more simple three-fold cross validation.

We use for $\mathbb{A}$ a linear SVM classifier, and $k = 100$ genes are extracted at each step. The evolution of the error rate all along our learning algorithm is shown in Figure 8. We obtain in [9] interesting results on classification rates on this database applying other algorithms such as CART to the OFW meta-algorithm.

We present in Table 1 results obtained using our jump process. We cannot run here a logistic regression coupled with a forward/backward algorithm because of the small number of signals in the database. The several selection methods used highlight the good performance of our jump algorithm, comparing it to standard methods such as Fisher tests.

Our results improve those referred to in [22], and the genes selected by our algorithm are consistent with some of the genes selected in other works (such as Zyxin in [11]). However, our selected features are nearly similar to those reported in [19]. One can again note the improvement using the jump process (second and third columns of Table 1).

Figure 9 represents the evolution of the number of genes selected at time t. In this case, we note again the good dimensionality reduction that permits our algorithm.

Finally, we can extract from the set of variables the names of genes most selected by our algorithm. We do not obtain exactly the same results for the 10 most important

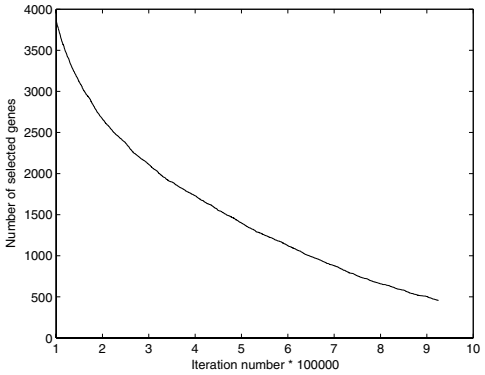


FIG. 9. *Evolution of the number of trees in the forest $\mathcal{F}_t$ with iteration number n.*

OFW Algorithm	Reflected jump diffusion
CTSD Cathepsin D	CTSD Cathepsin D
MPO Myeloperoxidase	MPO Myeloperoxidase
MB-1 gene	MB-1 gene
Catalase (EC 1.11.1.6)	Catalase (EC 1.11.1.6)
PROTEASOME IOTA CHAIN	Kazal-type serine proteinase
Zyxin	PROTEASOME IOTA CHAIN
Terminal transferase mRNA	VIL2 Villin 2
Kazal-type serine proteinase	PRG1 Proteoglycan 1
CCND3 Cyclin D3	CD37 CD37 antigen
CD37 CD37 antigen	HLA CLASS I HISTO. ANTIGEN

FIG. 10. *Genes most selected by our algorithm.*

genes listed in Figure 10 whether we use the OFW or the jump-diffusion approach.

9. Conclusion. From a theoretical point of view, we provide in this paper a mathematical algorithm to select variables in a large amount of features dealing with the general untractable problems using full data. This is not the case of filter methods (forward/backward, for instance) that use a heuristic strategy to compose features, and these methods are not useful in some situations. Our approach is based on a jump-diffusion stochastic differential equation, where jumps are transitions between spaces of features. We have seen that the structure of trees is convenient to deal with Markov processes since this enables us to identify the dynamical structure of our method. This method is highly motivated by real problems and we have shown (Theorem 7.2) the "optimality" of our algorithm since it converges toward the unique Gibbs field measure inferred from an energy $\mathcal{E}$.

From a practical point of view, we have reached interesting results in real data such as face recognition and microarray analysis, even if we do not perform any composition rule with this last database. We have obtained similar results as other standard methods on the synthetic example and have clearly overcome the forward/backward algorithm in the face recognition problem, which is the only other method known to permit features composition. On this last point, one can consider two hypotheses. Either the selected features are not so good with the forward/backward strategy

(it would be surprising) or (and it is the more likely) the classifier used after this selection is powerless compared to the SVM used with our method. This stresses the fact that our approach is usable with any classification algorithm: One can use for $\mathbb{A}$ SVMs, linear discriminant analysis, random forests, etc., and it is well known that at the moment there does not exist one algorithm which performs best on all pattern recognition problems.

In a forthcoming paper, we will present several computational results on this algorithm applied to several databases described by thousands of variables. Numerically, it would be interesting to use our composition strategy with real variables (instead of binary or ternary ones) since we have not used it on the leukemia database, for instance.

Similarly, it would be useful to interpret the composition of real variables as a process to learn a kernel for the SVM. We believe that using a Rademacher penalty term in energy $\mathcal{E}$ will improve the generalization ability of the algorithm and could permit us to obtain Oracle's inequality. Another improvement can be made using a simulated annealing strategy to fix the selected features to a deterministic version in the end of the algorithm.

Acknowledgments. This paper contains research performed during my thesis and I am glad to thank my Ph.D. advisor Laurent Younes for the numerous and helpful discussions we had on this occasion.

REFERENCES

[1] Y. Amit and D. Geman, *Shape quantization and recognition with randomized trees*, Neural Computation, 9 (1997), pp. 1545–1588.
[2] Y. Amit and D. Geman, *A computational model for visual selection*, Neural Computation, 11 (1999), pp. 1691–1715.
[3] R. F. Anderson and S. Orey, *Small random perturbation of dynamical systems with reflecting boundary*, Nagoya Math. J., 60 (1976), pp. 189–216.
[4] R. Atar, A. Budhiraja, and P. Dupuis, *Correction note: On positive recurrence of constrained diffusion processes*, Ann. Probab., 29 (2001), p. 1404.
[5] A. Benveniste, M. Métivier, and P. Priouret, *Algorithmes adaptatifs et approximations stochastiques*, Théorie et applications à l'identification, au traitement du signal et à la reconnaissance des formes, Masson, Paris, 1987. English translation available as *Adaptive Algorithms and Stochastic Approximations*, Appl. Math. 22, Springer-Verlag, Berlin, 1990.
[6] P. Billingsley, *Convergence of Probability Measures*, 2nd ed., Wiley Ser. Probab. Statist. Probab. Statist., John Wiley, New York, 1999.
[7] U. M. Braga-Neto and E. R. Dougherty, *Is cross-validation valid for small-sample microarray classification?*, Bioinformatics, (2003), pp. 1061–1069.
[8] L. Breiman, *Arcing classifiers*, Ann. Statist., 26 (1998), pp. 801–849.
[9] K.-A. Lê Cao, O. Gonçalves, P. Besse, and S. Gadat, *Selection of biologically relevant genes with a wrapper stochastic algorithm*, Statistical Applications in Genetics and Molecular Biology, 6 (2007), article 29.
[10] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, *Choosing multiple parameters for support vector machines*, Machine Learning, 46 (2002), pp. 131–159.
[11] K. Deb and R. Reddy, *Classification of Two-Class Cancer Data Reliably Using Evolutionary Algorithms*, KanGAL report 2003001, Kanpur Genetics Algorithms Laboratory, Kanpur, India, 2003. http://www.iitk.ac.in/kangal/papers/k2003001.pdf
[12] P. Dupuis and H. Ishii, *On Lipschitz continuity of the solution mapping to the Skorokhod problem, with applications*, Stochastics and Stochastics Rep., 35 (1991), pp. 31–62.
[13] P. Dupuis and K. Ramanan, *Convex duality and the Skorokhod problem.* I, II, Probab. Theory Related Fields, 115 (1999), pp. 153–195; 197–236.
[14] S. Ethier and T. Kurtz, *Markov Processes*, John Wiley, New York, 1986.
[15] F. Fleuret, *Fast binary feature selection with conditional mutual information*, J. Mach. Learn. Res., (2004), pp. 1531–1555.

[16] F. Fleuret and D. Geman, *Coarse-to-fine face detection*, International Journal of Computer Vision, 41 (2001), pp. 85–107.
[17] S. Gadat, *Apprentissage d'un vocabulaire symbolique pour la détection d'objets dans une image*, Thèse de l'École Normale Supérieure de Cachan, 2004.
[18] S. Gadat and L. Younes, *A stochastic algorithm for feature selection in pattern recognition*, J. Mach. Learn. Res., 8 (2007), pp. 509–547.
[19] D. Geman, C. d'Avignon, D. Naiman, and R. Winslow, *Classifying gene expression profiles from pairwise MRNA comparisons*, Statistical Applications in Genetics and Molecular Biology, 3 (2004), article 19.
[20] D. Geman and B. Jedynak, *Model-based classification trees*, IEEE Trans. Inform. Theory, 47 (2001), pp. 1075–1082.
[21] A. S. Goldberger and D. B. Jochems, *Note on stepwise least squares*, J. Amer. Statist. Assoc., 56 (1961), pp. 105–110.
[22] T. R. Golub, D. K. Slonim, P. Tamazyo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, *Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring*, Science, 286 (1999), pp. 531– 537.
[23] J. M. Harrison and R. J. Williams, *Brownian models of feedforward queueing networks: Quasireversibility and product form solutions*, Ann. Appl. Probab., 2 (1992), pp. 263–293.
[24] C. Jutten and J. Hérault, *Blind separation of sources, part 1: An adaptative algorithm based on neuromimetic architecture*, Signal Process., 24 (1991), pp. 1–10.
[25] S. Krempp, D. Geman, and Y. Amit, *Sequential Learning of Reusable Parts for Object Detection*, Technical report, Center for Imaging Science, Johns Hopkins University, Baltimore, MD, 2002. http://cis.jhu.edu/publications/papers_in_database/GEMAN/seqlearning.pdf
[26] H. J. Kushner, and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed., Appl. Math. (New York), 35, Stochastic Modelling and Applied Probability, Springer-Verlag, New York, 2003.
[27] MIT, *CBCL Face Database*, Center for Biological and Computation Learning, MIT, Cambridge, MA, 2000. http://cbcl.mit.edu/software-datasets/FaceData2.html
[28] J. Rissanen, *A universal prior for integers and estimation by minimum description length*, Ann. Statist., 11 (1983), pp. 416–431.
[29] A. Srivastava, M. I. Miller, and U. Grenander, *Ergodic algorithms on special Euclidean groups for ATR*, in Systems and Control in the Twenty-First Century (St. Louis, MO, 1996), Progr. Systems Control Theory 22, Birkhäuser Boston, Boston, MA, 1997, pp. 327–350.
[30] D. W. Stroock and S. R. S. Varadhan, *Multidimensional Diffusion Processes*, Grundlehren der Math. Wiss. [Fundamental Principles of Math. Sci.] 233, Springer-Verlag, Berlin, 1979.
[31] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, *Feature selection for SVMs*, in Proceedings of the Neural Information Processing Systems Conference (NIPS 2000), MIT Press, Cambridge, MA, pp. 668–674.