JUMP DIFFUSION OVER FEATURE SPACE FOR OBJECT RECOGNITION. *

Sébastien Gadat

sebastien.gadat@cict.fr

CMLA, ENS Cachan, 61 avenue du président Wilson, 94235 CACHAN Cedex, FRANCE
LSP, Université Paul Sabatier, 118 Route de Narbonne, 31062 TOULOUSE Cedex, FRANCE

Abstract

We present a dynamical model for a population of tests in pattern recognition. Taking a pre-processed initialization of a feature set, we apply a stochastic algorithm based on an efficiency criterion and a brownian noise to recursively build and improve the feature space. This algorithm simulates a jump-diffusion Markov process and estimates a probability distribution $P$ 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 like face detection and microarray analysis.

Keywords: Markov Processes, jump diffusion algorithms, stochastic approximation, Skorokhod map, feature selection, pattern recognition

1 Introduction

In this paper, we study a learning algorithm designed for the construction of features in pattern recognition tasks. This algorithm is designed 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 errors. Also, a good understanding of the features which have the more impact for the classification is critical in some subjects like biology: in microarray analysis, for example, it is important to identify the genes which express a pathology.

*Acknowledgment: this paper is a research performed during my thesis and I am glad to thank my Phd advisor Laurent Younes for the numerous and helpful discussions we had on this occasion.
Denote by \( \mathcal{F}_0 \) the initial set of features (input variables). In the interesting applications, this is a large set, which contains hundreds, maybe thousands, of elements. Given that, what we want to consider are not only the elements of \( \mathcal{F}_0 \), but also combinations of them, we face an overwhelming space of possible explanatory variables that we need to explore in the selection process. Our goal will be to provide a suboptimal stochastic approach to recursively explore and build new features space.

For simplicity, the only combinations we consider in this paper are products. 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 \( (\mathcal{F}_t)_{t \geq 0} \).

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, 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) [JH91], can be used for reduction of dimension, but are not able to create new variables by composition and do not drive to understand easily the selection. Methods based on hierarchically structured variables have also been developed: Amit, Geman and Fleuret in [AG99] and [FG01] 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 of cases for these algorithms, the amount of features constructed is not limited and can regularly increase if the learning procedure is not stopped [KGA02] and conclusions about optimization results are not inferred. Our approach of the feature space structure will be largely inspired from this sequential testing method, based on statistical correlation [FG01], entropy [GJ01] or mutual information [Fle04].

Finally, methods based on the optimisation of margin of support vector machines have been recently proposed to make recursive feature elimination (RFE [WMC+00], [CVBM02]). 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 to interesting results on several classification tasks like pedestrian detection or cancer morphology classification with a few 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 opened issue. Our objective will be to handle this problem using simultaneously upward and backward stochastic strategies.

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 in the course of time. These rules use a Metropolis Hastings evolution
based on an energy \( E \) 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. At last, note that we choose to formalize our work in a continuous settings (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 easier to precisely describe the dynamic of our constrained optimization method (section 5.1) while the formalism of Martingale Problem and generator for Markov processes will be very powerful to identify the asymptotic measure of our algorithm (theorems 6.3 and 7.2).

2 Notation and settings

2.1 Classes and features

We address the following pattern recognition problem. An input signal \( I \) must be classified into a fixed number of classes denoted \( \mathcal{C} = \{C_1, \ldots, C_N\} \). We assume that, from each input \( I \), a family of features, denoted \( \mathcal{F}_0 \), can be computed. Therefore, each \( \delta \) in \( \mathcal{F}_0 \) is a function of the input space \( I \), which, in our experiments, will be binary (values in \( \{0, 1\} \)) or ternary (values in \( \{-1, 0, 1\} \)), but more general real-valued functions can also be considered. The set \( \{\delta(I) \mid \delta \in \mathcal{F}_0, I \in \mathcal{I}\} \) will be the only information that will be used from \( I \) in the classification, which means that, for our purpose, it can harmlessly be identified to \( I \).

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 training data, which is a finite family of correctly labelled signals. However, for obvious dimensional complexity, the algorithm assume a specific parametric form for the classification function: it could be for instance classification trees (CART), support vector machines (SVM), linear discriminant analysis, nearest neighbor . . . In the two class problem, the simplest classification rule is based on linear separation: compute the sum \( \beta_0 + \sum_{\delta \in \mathcal{F}_0} \beta_\delta \delta \), and decide for the first class if this is negative and for the second otherwise. The parameters \( (\beta_0, \beta_\delta, \delta \in \mathcal{F}_0) \) are estimated so that this rule is as consistent as possible with the training data. Various definition 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 a SVM with a linear kernel because of the generalization ability of this algorithm. Note that the previous separation rule assumes that \emph{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.
- Composition: Use more complex expressions than monomials by combining the features and thus define one way to combine them.

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 \( \mathcal{I} = \{-1; 0; 1\}^{100} \) and \( \mathcal{F}_0 = \{\delta_1, \ldots, \delta_{100}\} \). 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 be unconnected or not) and follow exactly the same distribution on \( \mathcal{F}_0 \setminus (\mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3) \). We are interested in this paper on
the problem to detect interactions of features encoded in all \( G_i \), filter out noisy features in \( F_0 \backslash (G_1 \cup G_2 \cup G_3) \) and how one can form new compositional variables corresponding to each subset \( G_i \). We will more details on practical examples in section 8.

### 2.2 Composition of features

We here introduce notation regarding the composition of features. Individual features, from the original set will be denoted \( F_0 \) while the set of features obtained at time \( t \) will be naturally named \( F_t \). 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) that each composition of \( F_t \) keeps a memory of how it has been constructed. For this reason, we introduce trees on the set of features as follows.

To an elementary feature \( \delta \) in \( F_0 \), we associate the elementary tree (and keep the same notation “\( \delta \)“):

\[
\delta := \delta
\]

A tree feature \( A \) is a binary tree such that each node contains a composition of elementary features, and terminal nodes (leaves) are elementary features of \( F_0 \). Moreover, each non-terminal node in \( A \) must be the concatenation (union) of its descendants so that one can easily infer how any tree has been formed. The root of \( A \), denoted \( r(A) \), is the main node associated to the tree. Tree features \( A, B \) are aggregated with the construction rule “::”

\[
A :: B = r(A) \cup r(B)
\]

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

**Example 2.2** For instance, in the following operation

\[
\delta_1 \delta_2 :: \delta_1 \delta_3 = \delta_1 \delta_2 \delta_3 := A
\]

we can reform left and right sons (\( A_l \) and \( A_r \)) from \( A \) cutting his main node. It is manifest here that without this tree structure of features, the same composition will be

\[
\delta_1 \delta_2 :: \delta_1 \delta_3 = \delta_1 \delta_2 \delta_3 := B
\]

but we cannot directly obtain from \( B \) the way it has been formed since some sons could be \( \{(\delta_1 \delta_2); (\delta_1 \delta_3)\} \) or \( \{(\delta_2 \delta_3); (\delta_2 \delta_1)\} \).

To restrict the number of notations, we will keep again the notation \( F_0 \) for the set of elementary trees over the initial set of variables created by operation (1). Similarly, \( F_t \) will be the set of features handled at time \( t \) by our algorithm. We will denote by \( F^t \) the set of all trees over \( F_0 \) reachable from \( F_0 \) defined by (1) reachable using (2). Technically, the sets \( F_t \) in our jump process are subsets of \( F^t \) (we will call them forests), and \( P_t \) is a probability on \( F^t \), still supported by \( F_t \). However, this trivially reduces to parts of elementary features and probabilities on this set via the map \( A \rightarrow r(A) \). Each value of any tree \( A \) will naturally defined on any signal \( I \in I \) by \( r(A)(I) = \delta_{i_1}(I) \times \ldots \delta_{i_p}(I) \) if \( r \) is written as \( r(A) = \delta_{i_1} \ldots \delta_{i_p} \).
2.3 Base classification algorithm \( \mathcal{A} \)

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

We work with a randomized version of \( \mathcal{A} \), in which the randomization is on the set of variables. This randomization of features spaces has been introduced by Amit, Geman and Breiman (see [AG97],[Bre98]) where authors build accurate random classifiers with very low dependence to outliers and noise. In the training phase, this works as follows: first extract a base classification \( \mathcal{A} \), then randomize the features spaces, and build the classifiers \( \mathcal{A}_{\omega (1)}, \ldots, \mathcal{A}_{\omega (N)} \). Then, derive the classification in test phase using a majority rule within these \( N \) classifiers. This final algorithm will be denoted \( \mathcal{A} = \mathcal{A} (\omega (1), \ldots, \omega (N)) \). 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 \( P_t \) 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 \( \mathbb{F}^d \) and not designing the classification algorithm, \( \mathcal{A} \), for which we use standard procedures.

We will construct a process \((\mathcal{F}_t, P_t)\) where \( \mathcal{F}_t \) is a jump process over forests (subsets of \( \mathbb{F}^d \)) and between jumps, \( P_t \) 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 \( P_t \).

3 Constrained Diffusion

Between jump times, the probability will essentially evolve according to the diffusion

\[
dP_t = -\nabla \mathcal{E}_{err}(P_t) dt + \sigma dW_t.
\]

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 \( P_t \) with low cost \( \mathcal{E}_{err} \).

This process must however be modified in order to ensure that \( P_t \) is a probability supported by \( \mathcal{F}_t \). If \( \mathcal{F} \) is a subset of \( \mathbb{F}^d \), we denote \( \mathcal{H}_\mathcal{F} \) 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 equation (3) to \( \mathcal{H}_\mathcal{F} \) by replacing \( \nabla \) by \( \nabla^\mathcal{F} \) and using a Brownian motion on \( \mathcal{H}_\mathcal{F} \), or equivalently, using

\[
dP_t = -\nabla \mathcal{E}_{err}(P_t) dt + \Sigma^\mathcal{F} dW_t,
\]

where \( W \) is a Brownian motion on \( \mathbb{R}^{\mathcal{F}} \) 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 \( P_t \) belongs to \( \mathcal{S}_\mathcal{F} \) at all times. This is done using a constrained diffusion process which is here a reflected diffusion process:

\[
dP_t = -\nabla \mathcal{E}_{err}(P_t) dt + \Sigma^\mathcal{F} dW_t + dZ_t.
\]
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 $\partial S_F$.

### 3.1 Cost function

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

#### 3.1.1 Measuring the mean performance of $A$: the energy $E_{\text{err}}$

The algorithm $A$ provides a different classifier $A_\omega$ for each choice of a subset $\omega$ of $k$ features $\omega = (\omega_1, \ldots, \omega_k) \subset F$. We let $\eta$ be the classification error: $\eta(\omega) = \mathbb{P}(A_\omega(I) \neq C(I))$ which will be estimated by

$$g(\omega) = \hat{\mathbb{P}}(A_\omega(I) \neq C(I)) \quad \text{(6)}$$

where $\hat{\mathbb{P}}$ is the empirical probability on the training set.

Consider a probability distribution $\mathbb{P}$ on $F^k$. As we want to use a small number of feature, we fix an integer $k$, the distribution $\mathbb{P} \otimes^k$ corresponds to $k$ independent trials with replacement with respect to the distribution $\mathbb{P}$. We define the cost function $E_{\text{err}}$ by

$$E_{\text{err}}(\mathbb{P}) = \mathbb{E}_{\mathbb{P} \otimes^k} g(\omega) = \sum_{\omega \in F^k} g(\omega) \mathbb{P} \otimes^k(\omega) = \sum_{\omega \in F^k} g(\omega) \mathbb{P}(\omega_1) \ldots \mathbb{P}(\omega_k) \quad \text{(7)}$$

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

#### 3.1.2 Global cost function on $(F, \mathbb{P})$: the energy $\mathcal{E}$

We now describe the global cost function denoted $\mathcal{E}$. It will take the form

$$\mathcal{E}(F, \mathbb{P}) = E_{\text{err}}(\mathbb{P}) + E_{\text{struct}}(F)$$

where $E_{\text{struct}}$ is a structural energy on the forest. It takes the form

$$E_{\text{struct}}(F) = \sum_{A \in F} |A| - \sum_{A \in F} \hat{I}(A, g, A, d) \quad \text{(8)}$$

where $\hat{I}(A, g, A, d)$ is the empirical mutual information function between the left and right sub-trees of $A$. The first term $\mathcal{E}_1^A$ limits the size of the forest and comes from the minimum description length principle of information theory [Ris83]. The last term $\mathcal{E}_2^A$ is of compositional nature, and favors the concatenation of correlated trees (or trees with high mutual information) [FG01]. Our goal is now to minimize $\mathcal{E}$ over the space $F^d \times S_{\mathbb{F}_t}$ 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 get a critical importance for the stochastic dynamic search of good $(F_t, \mathbb{P}_t)$. 

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 exponentially distributed times, allowing the inference algorithm to visit $F_t^* \times S_{F_t}$. This accommodates the discrete nature of the problem.

At jump times, the transitions $F_t \rightarrow F_{t+dt}$ will correspond to deletion, addition or combination of elements of $F_t$. Each of these rules will be designed using an accept/reject scheme (Hastings) as follows. We handle here the complete cost function, $E(F_t, P_t)$ defined by (8). We bring back first some general notions on Metropolis-Hastings method (this paragraph may be jumped).

4.1.1 Generality on Metropolis-Hasting’s algorithm

The situation is as follows: let $\Omega$ a measurable set with a measure $m$ and let $\mu$ be a measure on $\Omega$ with density also denoted $\mu$ w.r.t $m$. Metropolis-Hastings transitions follow 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 $\mu$ is invariant.

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

(A1) $Q_0(x, \cdot)$ has a density $q(x, \cdot)$ w.r.t. $\rho_x$

(A2) $q(x, y) > 0 \iff q(y, x) > 0$

(A3) The measure $\rho_x(dy) \otimes m(dx)$ is symmetrical: for any function $f$ on $\Omega^2$

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

The transition $Q$ is then defined by

$$Q(x, dy) = \min \left( \frac{\mu(y)q(y, x)}{\mu(x)q(x, y)}, 1 \right) Q_0(x, dy) + \left( 1 - \int_{\Omega} \min \left( \frac{\mu(z)q(z, x)}{\mu(x)q(x, z)}, 1 \right) Q_0(x, dz) \right) \mathbb{1}_x(dy)$$

(9)

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

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

The second lines takes the form $g(x) \mathbb{1}_x(dy) m(dx)$ ans is obviously symmetric although the first one is symmetric thanks to our assumption on $\rho_x$. Consequently, we need to give 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 $(F_t)$ must be weakly reversible (assumptions (A1), (A2) and (A3)). We can advance at least two reasons for this imperative condition:

- **First**, note that our exploration process of $F^*$ has a stochastic nature and may be mistaken for some iteration because of the Metropolis-Hasting’s acceptation 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-Hasting’s acceptation rate computation (9) involves the ratio $q(x,y)/q(y,x)$ because of the assumptions (A1,A3) applied to $q(x,.)$ and $q(y,.)$. Obviously, if the features are not structured as 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-Hasting’s transitions on weighted forests

Denote by $m_F$ the Lebesgue measure on $S_F$ and consider $m$ the global base measure defined by

$$m = \sum_{\mathcal{F} \subseteq F^*} \mathbb{1}_\mathcal{F} \otimes m_F,$$

which means that

$$\int f(\mathcal{F}, \mathbb{P}) dm = \sum_{\mathcal{F} \subseteq F^*} \int_{S_F} f(\mathcal{F}, \mathbb{P}) dm_F(\mathbb{P})$$

Here, $\mathbb{1}_\mathcal{F}$ is Dirac’s measure at a forest $\mathcal{F}$. Consider a forest $\mathcal{F}_1$ and a weight $\mathbb{P}_1$ element of $S_{\mathcal{F}_1}$. The transitions are defined as follows: choose a new forest $\mathcal{F}_2 \in V_{\mathcal{F}_1}$ where $V_{\mathcal{F}_1}$ is the set of forests which are reachable in one jump and then choose an element of $S_{\mathcal{F}_2}$ according to a probability which depends on $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathbb{P}_1$. Assume that this probability has a positive density w.r.t some measure denoted $\psi_{\mathcal{F}_1, \mathcal{F}_2}(\mathbb{P}_1,.)$ on $S_{\mathcal{F}_2}$. This implies that the measures with respect to which the densities of the transitions are computed are

$$\rho_{\mathcal{F}_1, \mathbb{P}_1}(\mathcal{F}_2, .) = \mathbb{1}_{V_{\mathcal{F}_1}}(\mathcal{F}_2) \psi_{\mathcal{F}_1, \mathcal{F}_2}(\mathbb{P}_1, .)$$

where $\rho_{\mathcal{F}_1, \mathbb{P}_1} = \mathbb{1}_x$ is the measure defined in the former paragraph. Therefore, we need to construct $\rho$, $\psi$ and neighborhood $V$ in order to satisfy assumptions (A1),(A2) and (A3). We design in the next paragraph transitions satisfying (A1) and (A2). Next we will show that the symmetry requirement is true. Since in the framework of weighted forest

$$m_{\mathcal{F}_1}(d\mathbb{P}_1)\rho_{\mathcal{F}_1, \mathcal{F}_2}(\mathbb{P}_1, d\mathbb{P}_2) = \mathbb{1}_{V_{\mathcal{F}_1}}(\mathcal{F}_2)\psi_{\mathcal{F}_1, \mathcal{F}_2}(\mathbb{P}_1, .)m_{\mathcal{F}_1}(d\mathbb{P}_1)$$

it will be sufficient to establish

$$m_{\mathcal{F}_1}(d\mathbb{P}_1)\psi_{\mathcal{F}_1, \mathcal{F}_2}(\mathbb{P}_1, d\mathbb{P}_2) = m_{\mathcal{F}_2}(d\mathbb{P}_2)\psi_{\mathcal{F}_2, \mathcal{F}_1}(\mathbb{P}_2, d\mathbb{P}_1)$$

(10)

### 4.2 Transitions between forests

We first construct a set $T$ of compositional rules, before showing the weak reversibility (assumptions (A1) (A2) and (A3)) of our system. This set of transition 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, these set of transitions $T$ will be necessary.
**Definition 4.2 (Transition rules T)** \( T \) is the set of applications from \( \mathcal{P}(\mathbb{F}^2) \times S_{\mathbb{F}^2} \) to \( \mathcal{P}(\mathbb{F}^2) \times S_{\mathbb{F}^2} \) formed by Buddings, Cuttings, Suppressions or Rebirths. Let \( (\mathcal{F}, \mathcal{P}) \in \mathcal{P}(\mathbb{F}^2) \times S_{\mathbb{F}^2} \) we enumerate the states which are reachable in one jump from \( (\mathcal{F}, \mathcal{P}) \). For convenience of notations, \( U \) will denote the set of active variables in \( \mathcal{F} \) with their associated weights in \( \mathcal{P} \). The following quantities \( p_b, p_c, p_s \) and \( p_r \) will represent the non-negative probability at each jump time to choose budding, cut, suppression or rebirth. We first enumerate the **Budding Transitions:**

<table>
<thead>
<tr>
<th>Transition</th>
<th>Notation</th>
<th>Antecedents</th>
<th>Changes in ( (\mathcal{F}, \mathcal{P}) = U )</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budding without suppression</td>
<td>( B )</td>
<td>((A_1, p_1); (A_2, p_2) \in U)</td>
<td>Add ((A_1 :: A_2, p)) in (U) &lt;br&gt;Change the weights: ((A_1, p_1 - p + x), (A_2, p_2 - x)) &lt;br&gt;( p \sim U[0; p_1 + p_2], x \sim U[0; p_1 p_2] )</td>
<td>( p_b/4 )</td>
</tr>
<tr>
<td>Budding with left suppression</td>
<td>( B_l )</td>
<td>((A_1, p_1); (A_2, p_2) \in U)</td>
<td>Add ((A_1 :: A_2, p_1)) in (U) &lt;br&gt;Leave ((A_2, p_2)) in (U) &lt;br&gt;Remove ((A_1, p_1)) from (U)</td>
<td>( p_b/4 )</td>
</tr>
<tr>
<td>Budding with right suppression</td>
<td>( B_r )</td>
<td>((A_1, p_1); (A_2, p_2) \in U)</td>
<td>Add ((A_1 :: A_2, p_2)) in (U) &lt;br&gt;Leave ((A_1, p_1)) in (U) &lt;br&gt;Remove ((A_2, p_2)) from (U)</td>
<td>( p_b/4 )</td>
</tr>
<tr>
<td>Budding with both suppression</td>
<td>( B_{lr} )</td>
<td>((A_1, p_1); (A_2, p_2) \in U)</td>
<td>Add ((A_1 :: A_2, p_1 + p_2)) in (U) &lt;br&gt;Remove ((A_1, p_1)) from (U) &lt;br&gt;Remove ((A_2, p_2)) from (U)</td>
<td>( p_b/4 )</td>
</tr>
</tbody>
</table>

We present next the **Cut Transitions:**

<table>
<thead>
<tr>
<th>Transition</th>
<th>Notation</th>
<th>Antecedents</th>
<th>Changes in ( (\mathcal{F}, \mathcal{P}) = U )</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut without creation</td>
<td>( C )</td>
<td>((A_1 :: A_2, p) \in U) &lt;br&gt;((A_1, p_1); (A_2, p_2) \in U)</td>
<td>Remove ((A_1 :: A_2, p)) from (U) &lt;br&gt;Change the weights: ((A_1, p_1 + p - x), (A_2, p_2 + x)) &lt;br&gt;where ( x \sim U[-p_2 p_1 + p])</td>
<td>( p_c/4 )</td>
</tr>
<tr>
<td>Cut with left creation</td>
<td>( C_l )</td>
<td>((A_1 :: A_2, p) \in U) &lt;br&gt;((A_2, p_2) \in U)</td>
<td>Remove ((A_1 :: A_2, p)) from (U) &lt;br&gt;Add ((A_1, p)) in (U) &lt;br&gt;Leave ((A_2, p_2)) in (U)</td>
<td>( p_c/4 )</td>
</tr>
<tr>
<td>Cut with right creation</td>
<td>( C_r )</td>
<td>((A_1 :: A_2, p) \in U) &lt;br&gt;((A_1, p_1) \in U)</td>
<td>Remove ((A_1 :: A_2, p)) from (U) &lt;br&gt;Add ((A_2, p)) in (U) &lt;br&gt;Leave ((A_1, p_1)) in (U)</td>
<td>( p_c/4 )</td>
</tr>
<tr>
<td>Cut with both creation</td>
<td>( C_{lr} )</td>
<td>((A_1 :: A_2, p) \in U)</td>
<td>Remove ((A_1 :: A_2, p)) from (U) &lt;br&gt;Add ((A_1, x)) in (U) &lt;br&gt;Add ((A_2, p - x)) in (U) &lt;br&gt;where ( x \sim U[0; p])</td>
<td>( p_c/4 )</td>
</tr>
</tbody>
</table>

At last, we have the **Suppression and Rebirth transitions:**
\begin{tabular}{|c|c|c|c|}
\hline
Transition & Notation & Antecedents & Changes in \((F, \mathbb{P}) = U\) & Probability \\
\hline
Suppression & \(S\) & \((A, p) \in U\) & Remove \((A, p)\) from \(U\) \newline Change the weights for each \(\) other trees \((B, q) \in U\) in \((B, q/(1 - p))\) & \(p_s\) \\
\hline
Rebirth & \(S\) & \(A \in F_0 \setminus F\) & Add \((A, x)\) in \(U\) \newline Change the weights for each \(\) other trees \((B, q) \in U\) in \((B, q \ast (1 - x))\) & \(p_r\) \\
\hline
\end{tabular}

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

**Proposition 4.3 (Weak reversibility of \(T\))** Assumptions \((A1), (A2)\) and \((A3)\) are true under the dynamic of \((F_t, \mathbb{P}_t)\) induced by \(T\).

**Proof:** Take a forest \(F\) in \(\mathcal{P}(\mathcal{F}^d)\) and define \(V_F\) as the set of reachable forests using one (and only one) transition of \(T\). We first remark that if we enumerate all transitions between 2 forest, we have
\[
F_2 \in V_{F_1} \iff F_1 \in V_{F_2}
\]
Roughly speaking, if one tree is created, cut, or deleted using \(T\), it is instantaneously possible to flashback and cancel this transition using another rule in \(T\). 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
\[
\{(A_1, p_1); (A_2, p_2)\} \longmapsto \{(A_1, q_1 = p_1 - p + x); (A_2, q_2 = p_2 - x); (A_1 :: A_2, q_3 = p)\}
\]
One can see easily that \(q_1\) take 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 \(\{(A_1, q_1); (A_2, q_2); (A_1 :: A_2, q_3)\}\) can reach the initial state.

We can verify this point is true for all transitions given in the 3 former arrays while enumerating all possible transitions.

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

We now study the assumption \((A3)\). We have to compare the measure \(m_{F_1} (d\mathbb{P}_1) \psi_{F_1, F_2}(\mathbb{P}_1, d\mathbb{P}_2)\) with \(m_{F_2}(d\mathbb{P}_2) \psi_{F_2, F_1}(\mathbb{P}_2, d\mathbb{P}_1)\) where \((F_1, \mathbb{P}_1)\) is a weighted forest and \((F_2, \mathbb{P}_2)\) is reachable from \((F_1, \mathbb{P}_1)\) using \(T\).

We can enumerate then all transitions of \(T\) 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 \(m_{F_2}\) is the Lebesgue measure defined on the simplex \(S_{\mathcal{F}_1}\), denote \(N\) the length of vector \(\mathbb{P}_1\) and \(\mathbb{P}_1 = (p_1, p_2, \ldots, p_N)\) are coordinates. Without loss of generality, we can suppose that we choose to bud trees 1 and 2 so that we unchange other weighted trees of \(F_1\). The length of \(\mathbb{P}_2\) is consequently \(N + 1\), we have thus \(\mathbb{P}_2 = (q_1, q_2, \ldots, q_N, q_{N+1})\).

Hence, to one side we have
\[
m_{F_1}(d\mathbb{P}_1) \psi_{F_1, F_2}(\mathbb{P}_1, d\mathbb{P}_2) = \prod_{i=1}^N m_{F_1}(dp_i) \otimes \prod_{i=3}^N \mathbb{P}_1(q_i) \otimes \mathcal{U}_{\Delta_{\mathbb{P}_1, p_2}}(q_1, q_2, q_{N+1})
\]
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
\[
\Delta_{p_1,p_2} = \{(a,b,c) \in \mathbb{R}^3_+ \mid a + b + c = p_1 + p_2\}
\]
On the other side, we can equally write the transition measure
\[
m_{F_2}(dP_2)\psi_{F_2,F_1}(P_2,dP_1) = \prod_{i=1}^{N+1} m_{F_2}(dq_i) \otimes \prod_{i=3}^{N} \mathbb{I}_{q_i}(p_i) \otimes \mathcal{U}_{\Delta_{q_1,q_2,q_{N+1}}}(p_1,p_2)
\]
The symmetrical claim is then satisfied since for all measurable function $f$ on $S_{F_1} \times S_{F_2}$:
\[
\left< m_{F_1}(dP_1)\psi_{F_1,F_2}(P_1,dP_2);f \right> = \int_{S_{F_1} \times S_{F_2}} f(p_1,\ldots,p_N,q_1,\ldots,q_{N+1})m_{F_1}(dP_1)\psi_{F_1,F_2}(P_1,dP_2)
\]
\[
= \int \prod_{i=1}^{N} dp_i \int_0^{p_1+p_2} dp \int_{p-p_1}^{p_2} dx f(P_1,p_1+p+x,p_2-x,p_3,\ldots,p_N,p)
\]
\[
= \int \prod_{i=1}^{N+1} dq_i \int_{q_2}^{q_1+q_{N+1}+x} dx \int_{q_1+q_{N+1}-x}^{q_2+x} f(q_1+p-x,q_2+x,q_3,\ldots,q_N,P_2)
\]
\[
= \left< m_{F_2}(dP_2)\psi_{F_2,F_1}(P_2,dP_1);f \right>
\]
Same change of variables can be done for all other type of transitions of $T$ and we can conclude that the symmetrical assumption (A3) is also true.

\[\square\]

4.3 Decision steps of Markovian dynamic of jumps

Taking a jump time $t_j$ and any state of our process $(F_{t_j}, P_{t_j})$, we use rules taken from $T$ to modify $F_{t_j}$ and $P_{t_j}$ to $F_{t_j}+dt$ and $P_{t_j}+dt$. There are exactly 3 steps for the choice of which transition of $T$ is applied.

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

Step 2 When the rule is chosen, select the trees to which the rule is applied. One can make this decision dependent of the distribution $P_{t_j}$ or not. The simpler method is to choose uniformly among all trees in $F_{t_j}$ or in $F_0 \setminus F_{t_j}$.

Step 3 Accept or not the transition according to a differential energy criterion:
\[
Q((F_{t_j},P_{t_j});(F,P)) = \min \left( 1, e^{\epsilon(F_{t_j},P_{t_j})-\epsilon(F,P)} \frac{Q_0((F,P);(F_{t_j},P_{t_j}))q((F_{t_j},P_{t_j});(F,P))}{Q_0((F_{t_j},P_{t_j});(F,P))q((F_{t_j},P_{t_j});(F,P))} \right)
\]

Computation of the first step is easy with a discrete probability distribution on the rules constituting $T$. Choice at step 2 of trees to apply the rule selected can depend on the distribution $P_{t_j}$. 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 $F_0 \setminus F_{t_j}$. More details can be found in [Gad04].
5 Existence of the jump diffusion process

5.1 Existence of the reflected diffusion between jump times

We work in this section with fixed $\mathcal{F}_t = \mathcal{F}$ which contains $n$ trees (i.e. $|\mathcal{F}| = n$) and discuss the existence of a markovian reflected diffusion process which drives the evolution in the absence of jumps.

$$d\mathbb{P}_t = - \nabla \mathcal{E}_{err}(\mathbb{P}_t) dt + \sigma dW_t + dZ_t \tag{12}$$

Existence of solutions to (12) can be deduced from the construction of the Skorokhod map ([DI91],[DR99]) associated to $\mathcal{S}_\mathcal{F}$. Given a closed set $G$ of $\mathbb{R}^n$, a set of unit vectors $d(x)$ for all $x$ on the boundary $\partial G$, and a trajectory $\psi$ taking values in $\mathbb{R}^n$, the Skorohod Problem (SP) defines a constrained version $\phi$ of $\psi$ using the least effort to keep $\psi$ in $G$. We will use the formalization introduced in [DI91].

**Definition 5.1 (Skorokhod Problem)** Let $\psi \in \mathbb{D}([0; \infty[, \mathbb{R}^n)$, space of càdlàg trajectories in $\mathbb{R}^n$, such that $\psi(0) \in G$. Then $(\phi, \zeta)$ solves the SP for $\psi$ with respect to $G$ and $d(.)$ (directions of constraints), if for all $t \geq 0$

1. $\phi(t) = \psi(t) + \zeta(t)$
2. $\phi(t) \in G$
3. $|\zeta|(t) < \infty$
4. $|\zeta|(t) = \int_0^t \mathbb{1}_{\phi(s) \in \partial G} d|\zeta|(s)$
5. $\exists \gamma : [0; \infty[ \rightarrow \mathbb{R}^n$ such that $\gamma(.) \in d(\phi(.))$ (d$|\zeta|$ a.s.) and $\zeta(t) = \int_0^t \gamma(s) d|\zeta|(s)$

In our situation, we have $G = \mathcal{S}_\mathcal{F}$.

We call $\vec{n}_i$ the unit vectors $\vec{n}_i(j) = \delta_{i,j}$ and if $\vec{N}$ is the unit normal vector to the hyperplane of $\mathbb{R}^\mathcal{F}$ which carry $\mathcal{S}_\mathcal{F}$, we then have

$$\vec{N} = \frac{1}{\sqrt{J}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

and the unit orthogonal projections of $\vec{n}_i$ on $\mathcal{S}_\mathcal{F}$ are given by

$$\vec{n}_i = \frac{\vec{n}_i - \langle \vec{n}_i; \vec{N} \rangle \vec{N}}{||\vec{n}_i - \langle \vec{n}_i; \vec{N} \rangle \vec{N}||_2}$$

where $\langle :, \rangle$ is the euclidean scalar product on $\mathbb{R}^\mathcal{F}$. Thus, we can choose the directions of constraints associated to the simplex $\mathcal{S}_\mathcal{F}$ and to our (SP) by:
Definition 5.2 (Directions of constraints $d(.)$)

$$\forall x \in \partial S_F \quad d(x) = \left\{ \gamma = \sum_{i \mid x_i = 0} \alpha_i n_i \mid \alpha_i \geq 0, \|\gamma\| = 1 \right\}$$

These directions of reflexion on $\partial S_F$ can be expressed easily in a different form:

Proposition 5.3 (Directions of constraints $d(.)$) For any point $x$ in $\partial S_F$, vectors $d(x)$ coincide exactly with the sets of unit vectors:

$$d(x) = \{ -\gamma \mid \|\gamma\| = 1 \mid \exists y \in H_F \; y - \pi_F(y) = \alpha \gamma \; \text{and} \; \alpha \leq 0 \; x = \pi_F(y) \}$$

where $\pi_F$ is the natural convex projection on the simplex $S_F$.

Figure 1 summarize this natural property. One can remark that directions $d(x)$ are strongly connected to convex projections on $S_F$ since they correspond exactly to the unitary vectors that can be used to project any exterior point to $S_F$ exactly on the boundary point $x$. These directions $d$ reach to the definition of the continuous process $Z$ and we will describe in section 7 how stochastic algorithm approximates this term. In stochastic approximation algorithms, the usual way of introducing convex constraints is to use convex projection. Here, from the point of view of the Markov process, one need to define the possible constraint vectors to add to our process to keep it in $S_F$. This yield to a set of possible callback vectors at each boundary point of $S_F$ (see last definition).

Figure 1: Directions of reflexion vectors for $x$ in $\partial S_F$.

Theorem 5.2 and the construction of the “dual set” of paragraphe 4.3.2 in [DR99] imply wellposedness of the (SP) on $D([0; \infty[, S_F])$. This enables us to define the Skorokhod Map $\Gamma : \psi \mapsto \phi$ where $\phi$ is the unique solution of (SP) associated to $\psi$ with respect to $S_F$ and $d(.)$. Since assumption 2.1 and assumption 5.1 of [DR99] are also satisfied, the Skorokhod Map $\Gamma$ is Lipschitz continuous using $\|\cdot\|_{\infty}$ for trajectories of $D([0; \infty[, S_F])$. These results are summarized in the next theorem:

Theorem 5.4 (Skorokhod Map on $D([0; \infty[, S_F])$) For all càdlàg map $\psi$, there exists a unique map $\phi$ such that $(\phi, \psi - \phi)$ solves the SP. Moreover, using the notation $\Gamma$, we have:

$$\forall (\psi_1, \psi_2) \in D([0; \infty[, \mathbb{R}^n)^2 \quad \sup_{t \geq 0} \left| \Gamma(\psi_1)(t) - \Gamma(\psi_2)(t) \right| \leq K_{S_F} \sup_{t \geq 0} \left| \psi_1(t) - \psi_2(t) \right|$$
The Skorohod map allows to formalize the reflected diffusion as a system of integral equations:

\[
\begin{align*}
X_t &= P_0 - \int_0^t \nabla E_{err}(P_s) ds + \sigma dW(s) \\
\mathbb{P}_t &= \Gamma(X)_t
\end{align*}
\]

This system is equivalent to the stochastic differential equation (12) [AO76, DI91]. Strong (and obviously weak) existence and uniqueness of such integral system is standard using a fixed point method [Gad04, SV79] and Lipschitz continuity of the drift $\nabla E_{err}$.

For $\omega \in \mathcal{F}_k$ and $\delta \in \mathcal{F}$, denote by $C(\omega, \delta)$ the number of occurrences of $\delta$ in $\omega$:

\[C(\omega, \delta) = |\{i \in \{1, \ldots, k\} \mid \omega_i = \delta\}|.\] (13)

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

\[
\nabla_P E_{err}(\delta) = \sum_{\omega \in \mathcal{F}_k} C(\omega, \delta) \frac{P^{\otimes k}(\omega)}{P(\delta)} g(\omega).
\] (14)

Taking a constant symmetric definite positive matrix on $\mathcal{S}_\mathcal{F}$ covariance matrix $\sigma$:

\[\sigma = \text{Id} - \overrightarrow{N} \overleftarrow{N}^t\]

we can thus infer the following result.

**Theorem 5.5 (Existence and uniqueness of (12))** Let $(\Omega, \mathcal{T}, Q)$ a probability space with an increasing filtration $\mathcal{T}_t$, let $W_t$ standard brownian motion on $\mathbb{R}^{\mathcal{F}}$ and $P$ a random variable $\mathcal{T}_0$-measurable, there exists a unique pair $(\mathbb{P}_t, Z_t)$ $\mathcal{T}_t$-measurable satisfying (12) with

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

2. $\forall t \geq 0 \quad |Z_t| = \int_0^t 1_{\mathbb{P}_s \in \mathcal{S}_\mathcal{F}} d|Z_s|

3. $\forall t \geq 0 \quad dZ_t \in d(\mathbb{P}_t)$

**Proof :** See [Gad04] chapter 4 or [SV79] chapter 5. \qed

### 5.2 Existence of the complete process

Since the jump times 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 4 different simplices and jumping times. Each simplex corresponds to a features space while the a.s. continuous trajectory points the evolution of our extraction method $P_s$. We represent here one reflexion on $\mathcal{S}_{\mathcal{F}_{i_2}}$ and several jumps between several (4) simplices. Note that if it is possible to jump from $\mathcal{S}_{\mathcal{F}_{i_2}}$ to $\mathcal{S}_{\mathcal{F}_{i_3}}$ we remark that it is equally possible to jump from $\mathcal{S}_{\mathcal{F}_{i_3}}$ to $\mathcal{S}_{\mathcal{F}_{i_2}}$ illustrating here the weak reversibility defined in section 4.2.

We will denote $\Phi$ the stationary solution of the s.d.e. of the reflected jump diffusion based on reflected diffusion on each simplex and jumps between subspaces of features. This solution is defined as follows
Figure 2: General form of the stochastic jump diffusion process

Definition 5.6 (Stationary solution $\Phi$) Let $(\Omega, \mathcal{T}, Q)$ probability space with an increasing filtration $\mathcal{T}_t$, let $(W_t)_{t \geq 0}$ a standard brownian motion on $\mathbb{R}^{\mathcal{F}[t]}$ and $(N_t)_{t \geq 0}$ 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 stochastic differential equation with jumps:

$$d \left( \frac{\mathbb{P}_t}{\mathcal{F}_t} \right) = -\left( \nabla \mathbb{E}(\mathbb{P}_t) dt + \Sigma \mathbb{F}_t dW_t + dZ_t \right) + \int_{\mathcal{F} \subset \mathcal{F}[t], \mathbb{P} \in \mathcal{S} \mathcal{F} \mathcal{P}} Q \left[ \left( \frac{\mathcal{F}_t}{\mathbb{P}_t} ; \frac{\mathcal{F}}{\mathbb{P}} \right) \right] N \left( d \left( \frac{\mathbb{P}}{\mathcal{F}} ; dt \right) \right)$$

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

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

with respect to the measure

$$m = \sum_{\mathcal{F} \subset \mathcal{F}[t]} 1_{\mathcal{F}} \otimes m_{\mathcal{F}}.$$

on $\mathcal{P}_{\mathcal{F}[t]} \times \mathcal{S}_{\mathcal{F}[t]}$, $m_{\mathcal{F}}$ being the uniform measure on the simplex $\mathcal{S}_{\mathcal{F}}$.

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 $\mu$ associated to $\mathcal{E}$. 
6.1 Infinitesimal generator of $\left(\mathcal{F}_s, \mathbb{P}_s\right)_{s \geq 0}$

Our Markov process is a combination of a reflected diffusion process and a jump process. A generic function on $\mathcal{P}(\mathbb{P}^\mathcal{F}) \times S_\mathcal{F}$ can be decomposed as

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

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

$$A^d f(\mathbb{P}, \mathcal{F}) = -\langle \nabla_{\mathbb{P}} \mathcal{E}_{\text{err}} | \nabla_{\mathbb{P}} f(\mathcal{F}) \rangle + \frac{1}{2} \Delta_{\mathbb{P}} f(\mathcal{F}(x))$$

and

$$A^j f(\mathbb{P}, \mathcal{F}) = \int_{\mathcal{P}(\mathbb{P}^\mathcal{F}) \times S_\mathcal{F}} [f_{\mathcal{F}'}(\mathbb{P}') - f_{\mathcal{F}}(\mathbb{P})]Q \left[ (\mathcal{F}, \mathbb{P}, (\mathcal{F}', \mathbb{P}')) \right] d\mu(\mathcal{F}', \mathbb{P}')$$

where $Q$ is the transition probability at jump times.

6.2 Positive recurrence

The main result of this paragraph uses the positive definite nature of $\Sigma_{\mathcal{F}}$ on $\mathcal{H}_{\mathcal{F}}$ and a result of [HW92] (theorem 1, section 7) ensuring positive recurrence of reflected process (without any jump). For any reachable simplex $S_\mathcal{F}$, the unique process solution of

$$d\mathbb{P}_t = -\nabla \mathcal{E}_t(\mathbb{P}_t) dt + \sigma dW_t + dZ_t$$

satisfy for all compact set $S \subset S_\mathcal{F}$ 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 $p$):

$$\inf_{p \in S_\mathcal{F}} P_p[\tau_S \leq 1] > 0 \quad (16)$$

where

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

Equation (16) means that starting for any point $p$ of simplex $S_\mathcal{F}$, one can reach $S$ with time shorter than one with a probability strictly positive. This implies in particular [ABD01] (theorem 2.8) the positive recurrence of (12) without jump, and the existence of a unique invariant measure.

The extension of the results to the jump-diffusion process now only requires the following fact. Denote

$$P_{\mathcal{F},T}(S) = \inf_{p \in S_\mathcal{F}} P_p[\tau_S \leq T]$$

for $S \subset S_\mathcal{F}$ and $\tau_S$ is the hitting time of $S$. We have

**Corollary 6.1**

$$P_{\mathcal{F},T}(S) > 0$$

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 $\mathcal{F}$ and $\mathcal{F}'$, for any $\mathbb{P} \in S_\mathcal{F}$, the transition $(\mathcal{F}, \mathbb{P}) \rightarrow \mathcal{F}'$ in $N$ steps has a probability strictly larger than some positive constant, $\eta$. 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 jump. □
6.3 Invariant measure of the process

Properties of the invariant measure can be inferred from the positive recurrence of \((P_t, \mathcal{F}_t)_{t \geq 0}\). First, remark that for any initialisation \((P_0, \mathcal{F}_0)\) of the process, the family of occupation measures \((\mu_t)_{t \geq 0}\), defined by
\[
\mu_t(A) = \frac{1}{t} \int_0^t P_{(P_0, \mathcal{F}_0)}[(P_s, \mathcal{F}_s) \in A] \, ds
\]
is tight and any weak limit is an invariant measure of \((P_t, X_t)_{t \geq 0}\) 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 \((P_t, \mathcal{F}_t)\). Identification of this measure from the characterization of [EK86] is used for example in [SMG97].

We use here the well-posedness of the associated Martingale Problem. Define first the core of this generator as
\[
\mathcal{D} = \left\{ f = \sum_{\mathcal{F} \subset \mathcal{F}^\sharp} \mathbb{1}_{S_{\mathcal{F}}}(P) f_{\mathcal{F}}(P) \mid \forall \mathcal{F} \subset \mathcal{F}^\sharp \quad \forall P \in \partial S_{\mathcal{F}} \quad \nabla f_{\mathcal{F}}(P) = 0 \right\}
\]
We start noticing that for any function in \(\mathcal{D}\), the mean effect of generator \(A\) with distribution \(\mu\) given by (15) is null.

**Proposition 6.2** Assume \(f\) is an element of \(\mathcal{D}\), then we have
\[
\int Af \, d\mu = 0
\]

**Proof :** This result is proved by integration by parts, using the Neumann conditions on each simplex \(S_{\mathcal{F}}\) where \(\mathcal{F} \subset \mathcal{F}^\sharp\), Ostrogradski formula and the stability equation on the transition acceptance threshold (11). Similar arguments can be found in [SMG97]. \(\square\)

We are now able to prove the next theorem

**Theorem 6.3** The Gibbs field \(\mu\) given by equation (15) is the unique invariant measure of the global reflected jump diffusion process and the Martingale Problem associated to \(A\) on \(\mathcal{D}\) is well posed.

**Proof :** We first apply Echeverria’s theorem [EK86] (see theorem 9.17 chapter 9) to show that \(\mu\) is stationary. Denote \(E\) the compact set \(\{(\mathcal{F}, P) \mid \mathcal{F} \subset \mathcal{F}^\sharp, P \in S_{\mathcal{F}}\}\), remark first that \(\mathcal{D}\) is dense in \(C(E)\) by Uryshon lemma applied in each simplex \(S_{\mathcal{F}}\).

Now, \(A\) satisfy the positive maximum principle on \(\mathcal{D}\) (\(A\) is a classical jump-diffusion generator) and the measure \(\mu\) satisfies
\[
\forall f \in \mathcal{D} \quad \int_E Af \, d\mu = 0
\]
Hence, \(\mu\) is stationary for \(A\). Since \(A\) satisfies the maximum principle, \(A\) is dissipative on \(C(E)\) and \(E\) is separable. Denote then \(\nu\) a measure on \(E\), we can apply result of theorem 4.1,chapter 4 in [EK86] to conclude that uniqueness holds for the martingale problem \((A, \nu)\) and every solution of the martingale problem is Markov.

The martingale problem is thus well-posed on \(C(E)\), every solution of the Martingale Problem is a weak solution of the stochastic differential equation of jump diffusion. Moreover, \(\mu\) is the unique stationary distribution of the process \((\mathcal{F}_t, P_t)\). \(\square\)
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 to replace it by a stochastic approximation of the Robbins-Monro type.

Before passing to the drift term, we first address the time discretization issues.

7.1 Time discretization

To solve (12), we use a time-discretization scheme with a discretization step $\alpha$:

$$\forall n \in \mathbb{N} \quad P_{n+1} = P_n - \alpha \nabla^F \mathcal{E}_{err}(P_n) + \sqrt{\alpha \sqrt{\sigma}} d\xi_n + dz_n$$

where $d\xi_n$ is a centered normal $|\mathcal{F}|$ dimensional vector and $dz_n$ is the smaller vector to add to make $P_{n+1} \in \mathcal{S}_\mathcal{F}$. In other words

$$\forall n \in \mathbb{N} \quad P_{n+1} = \pi_\mathcal{F} (P_n - \alpha \nabla^F \mathcal{E}_{err}(P_n) + \sqrt{\alpha \sqrt{\sigma}} d\xi_n)$$

However, the computational issue comes from the gradient of $\mathcal{E}_{err}$, given in (14), which requires a sum over all $\omega$ in $\mathcal{F}^p$. This is an untractable sum, since $|\mathcal{F}|$ is typically thousands and $p$ hundreds. It however can be replaced by the stochastic approximation defined in the next section.

7.2 SDE method for approximation

Stochastic approximation can be seen as noisy discretizations of SDE’s ([KY03]). They are generally expressed under the form

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

where $X_n$ is the current state of the process, and $\zeta_{n+1}$ a random perturbation, $\xi_n$ 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 $\zeta_{n+1}$ only depends on the current value of $X_n$, then, one defines an average drift $X \mapsto G(X)$ by

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

and the equation (20) can be shown to evolve similarly to the SDE: $dX_t = G(X)dt + \sqrt{\sigma}dw_t + dz_t$, 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 (12) in this framework, we need to design a random variable $d_n$ (identified to $F(X_n, \zeta_n)$ in (20)) such that

$$E[d_n] = -\nabla^F \mathcal{E}_{err}(P_n) = -\Pi_{\mathcal{H}_\mathcal{F}} [\nabla \mathcal{E}_{err}(P_n)]$$

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

$$P_{n+1} = P_n - \alpha_n d_n + \sqrt{\alpha_n} \sqrt{\sigma} \xi_n + dz_n = \pi_\mathcal{F} (P_n - \alpha_n d_n + \sqrt{\alpha_n} \sqrt{\sigma} \xi_n)$$

From (14), we have:

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

Using the linearity of the projection $\Pi_{\mathcal{H}_\mathcal{F}}$, we get
$$\Pi_{\mathcal{H}_X} (\nabla \mathcal{E}(\mathbb{P})) (\delta) = \mathbb{E}_{\mathbb{P}^{\otimes k}} \left[ \Pi_{\mathcal{H}_X} \left( \frac{C(\omega, \cdot)}{\mathbb{P}(\cdot)} \right) (\delta) \right]$$

Consequently, following (22), it is now natural to define the approximation term of the reflected diffusion (12) by

$$d_n(\delta) = \Pi_{\mathcal{H}_X} \left( \frac{C(\omega_n, \cdot)}{\mathbb{P}(\cdot)} \right) (\delta)$$

(24)

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

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 $\omega_n$ in $\mathcal{F}^k$ with respect to $\mathbb{P}_{\otimes k}$.
3. Step $n$: compute $g(\omega_n)$.
4. Step $n$: update $\mathbb{P}_{n+1}$ with

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

(25)

where $-\alpha_n C(\omega_n, \cdot)/\mathbb{P}_n$ 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, the previous must be combined with transitions of $(\mathcal{F}, \mathbb{P})$ at jump times. This results in the new following 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 $n = 0$.
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 $(\mathbb{P}^n(t)_{t \geq 0})_{n \in \mathbb{N}}$ 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 notations of [KY03], we set up the time parameter $\tau_n$ by

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

and the map $m$ permitting to associate continuous time and discrete iteration:

$$m(t) = \sup_{\tau_n \leq t} \{ n \in \mathbb{N} \}$$
Given that the $j$-th jump occurs at time $\nu_j$, we construct its values according to the distribution $Q((F_{\nu_j}, P_{\nu_j}), \ldots)$ to obtain $(F_{\nu_j}, P_{\nu_j})$. It is thus possible to define the discrete jump term in the discrete case as

$$q_j = P_{m(\nu_j)+1} - P_{m(\nu_j)}$$

which corresponds to the term we add to compute the jump from $\nu_j$ to $\nu_j$.

We now define the sequence of right continuous interpolation processes $(P_n(t))_{t \geq 0}$ initialized at $P_n$.

**Definition 7.1 (Processes $(P_n, Y_n, W_n, Z_n)_{n \in \mathbb{N}}$)** We define $(P_n, Y_n, W_n, Z_n)$ valued in $\mathbb{R}^{\mathbb{F}^1}$ by

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

where the term $y_i$ satisfies

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

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 $\mathbb{R}^{\mathbb{F}^1}$,

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

Finally,

$$P_n(t) = P_n + Y_n(t) + W_n(t) + Z_n(t) + \sum_{\tau_n \leq \nu_j \leq \tau_n + t} q_j$$

With these definitions, it is obvious to verify that $P_n$ is a process on $\mathbb{S}_{\mathbb{F}^1}$ 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 $(\alpha_n)$ such that $\sum \alpha_i = \infty$ and $\sum \alpha_i^2 < \infty$ (see [BMP87, KY03] for instance).

**7.3.2 Convergence of $(P_n, Y_n, W_n, Z_n)_{n \in \mathbb{N}}$**

We will show that the family of processes $(P_n(\cdot), Y_n(\cdot), W_n(\cdot), Z_n(\cdot))_{n \in \mathbb{N}}$ is weakly compact in the space $\mathcal{D}$. The associated topology on this space is derived from the Skorokhod distance ([Bil99, KY03]) and we consider weak convergence of trajectories of $\mathcal{D}([0; \infty])$.

**Theorem 7.2** The processes $(P_n, Z_n)_{n \in \mathbb{N}}$, which are stepwise constant, weakly converge towards the unique stationary solution of sde without jumps and $(P_n, F_n)_{n \in \mathbb{N}}$ converges towards the stationary measure $\mu$.

Proof of theorem 7.2 includes three steps: first prove the tightness of the family $(P_n, Y_n, W_n, Z_n)_{n \in \mathbb{N}}$, then identify the unique possible weak limit and finally show the convergence towards the stationary measure $\mu$. 
7.3.3 Tightness

To show tightness, we use the following criterion:

**Theorem 7.3** ([KY03, Bil99]) Let $X^n$ a sequence in $D; (X^n)_{n \in \mathbb{N}}$ is tight iff

1. For any time $T$ and $\epsilon > 0$, there exists an integer $n_0$ and a real $K$ satisfying
   
   $\forall n \geq n_0 \quad P \left( \sup_{t \leq T} |X^n(t)| \geq K \right) \leq \epsilon \quad (26)$

2. $\forall \epsilon > 0 \quad \lim_{\delta \to 0} \limsup_n P \left[ w'_{X^n}(\delta) \geq \epsilon \right] = 0 \quad (27)$

We establish successively (26) and (27) for our family of processes $(P^n, Y^n, W^n, Z^n)_{n \in \mathbb{N}}$. The next proposition shows that (26) is true for $(P^n, Y^n, W^n, Z^n)_{n \in \mathbb{N}}$ and consequently guarantees the tightness of the family $(P^n, Y^n, W^n, Z^n)_{n \in \mathbb{N}}$.

**Proposition 7.4** The sequence of processes $(P^n, Y^n, W^n, Z^n)_{n \in \mathbb{N}}$ satisfy (26).

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

$\tilde{y}_n = y_n - \mathbb{E}_n [y_n]$

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 $\mathbb{E}_n [y_n]$ is bounded by $M$, the first tightness criterion is true for the 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

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

is a martingale for the filtration generated by $\mathbb{F}_p = \sigma(\mathbb{P}_i, \xi_i, w_{i-1}, i \leq n + p)$ and 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, $E[|\tilde{y}_i|] = E(E[|\tilde{y}_i||F^n_i])$ and $E[|\tilde{y}_i||F^n]$ is bounded by $2M$. We have $\sum_{i=n}^{p} \alpha_i \leq T$ and we can deduce from these upper-bounds that

$$\lim_{K \to \infty} P\left(\sup_{q \leq p} |M^n_q| > K\right) = 0$$

The fact that

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

is standard and can be found in [KY03, Gad04]. Finally, since $Z^n = \mathbb{P}^n - Y^n - W^n$, $(Z^n)_{n \in \mathbb{N}}$ obviously satisfies (26).

We must now establish condition (27) to achieve tightness of $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n \in \mathbb{N}}$.

**Proposition 7.5 (Condition (27))** Each one of the processes $(\mathbb{P}^n, Y^n, W^n, Z^n)_{n \in \mathbb{N}}$ satisfy (27).

**Proof:** We first establish (27) for $(Y^n)_{n \in \mathbb{N}}$. Remark that

$$E[Y^n(t+s) - Y^n(t)] = \sum_{i=m(\tau_n + t)}^{m(\tau_n + t+s)} \alpha_i E\left[E\left[|y_i| |F^i_0\right]\right]$$

Then, use the fact that the expectation of $y_k$ are bounded by $M$ to obtain:

$$E[|Y^n(t+s) - Y^n(t)|] \leq Ms$$

We thus conclude that (27) is true for $(Y^n)_{n \in \mathbb{N}}$ using the Markov inequality. The argument is standard to get similar result for $(W^n)_{n \in \mathbb{N}}$ by Doob’s inequality (see [KY03]). The jump component involved by terms $q_j$ defines also a sequence of processes:

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

Inequality (27) for $(J^n)_{n \in \mathbb{N}}$ is here clearly satisfied since jump occurs exponentially as each term $q_j$ is bounded. Consequently, we have:

$$\limsup_{n} P[w^n, (\delta) \geq \epsilon] = o(\delta)$$

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

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

since

$$|z_i| \leq C|\alpha_i y_i + \sqrt{\alpha_i} d\xi_i|$$

(28)
Using inequality (27) for \((Y^n)_{n \in \mathbb{N}}\) and \((W^n)_{n \in \mathbb{N}}\), and (28), we obtain the second tightness inequality needed on the processes \((Z^n)_{n \in \mathbb{N}}\). 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. Remark first that if \((\mathbb{P}^n, Y^n, W^n, Z^n)\) is weakly convergent towards \((\mathbb{P}, Y, W, Z)\), then \((\mathbb{P}, Y, W, Z)\) is solution of the reflected jump diffusion \(\Phi\) initialized to the weak limit of \((\mathbb{P}^n(0), Y^n(0), W^n(0), Z^n(0))\) using the same argument of [KY03] 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 stationary measure \(\mu\). Denote \(\nu_\infty\) the weak limit of \((\mathbb{P}^{N_k}(0))_{k \in \mathbb{N}}\), it is then sufficient to show that for any measurable function \(\phi\):

\[
\mathbb{E}_{\nu_\infty} \phi = \mathbb{E}_\mu \phi
\]

Denote \(P^t_\nu\) the law of our process at time \(t\) initialized by measure \(\nu\), since \(\mu\) is the unique stationary measure we have for any compact set of measures \(K\)

\[
\forall \nu \in K \quad \forall \epsilon > 0 \quad \exists T > 0 \quad \forall t \geq T \quad \left| \int \phi(y)dP^t_\nu - \int \phi(y)d\mu(y) \right| \leq \epsilon
\]  

(29)

Taking \(\epsilon\) strictly positive and applying (29) 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

\[
\forall t \geq T \quad \left| \int \phi(y)dP^t_\nu - \int \phi(y)d\mu(y) \right| \leq \epsilon
\]

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

\[
\left| \int \phi(y)d\nu'_\infty(y) - \int \phi(y)d\mu(y) \right| \leq \left| \int \phi(y)d\nu_\infty(y) - \mathbb{E} [\phi(\mathbb{P}(\tau_{N_k}))] \right|
\]

\[
+ \left| \mathbb{E} [\phi(\mathbb{P}(\tau_{N_k}))] - \int \phi(y)dP^T_{\nu'_\infty}(y) \right|
\]

\[
+ \left| \int \phi(y)dP^T_{\nu'_\infty}(y) - \int \phi(y)d\mu(y) \right|
\]

Making \(N_k \to \infty\), then \(\tau_{N_k} \to \infty\) and under our hypotheses on \(T, \nu_\infty\) and \(\nu'_\infty\), we obtain

\[
\left| \int \phi(y)d\nu_\infty(y) - \int \phi(y)d\mu(y) \right| \leq \epsilon
\]

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 towards \(\mu\). □

8 Experiments

8.1 Synthetic data

We first test our algorithm on a simple synthetic example. We consider \(f = 100\) ternary variables \((|\mathcal{F}| = 100)\) and 3 classes (similar results can be obtained with more classes and variables).
Figure 3: Evolution of the mean error rate for the reflected diffusion (dashed line) and the reflected jump diffusion (full line).

We let \( \mathcal{I} = \{-1; 0; 1\}^f \), and feature \( \delta_i(I) \) simply be the \( i \)th coordinate of \( I \in \mathcal{I} \).

Let \( \mathcal{G} \) be a subset of \( \mathcal{F} \). We define the probability distribution \( \mu(\cdot; \mathcal{G}) \) in \( \mathcal{I} \) to be the one for which all \( \delta \) in \( \mathcal{F} \) are independent, \( \delta(I) \) follows a uniform distribution on \( \{-1; 0; 1\} \) if \( \delta \not\in \mathcal{G} \) and \( \delta(I) = 1 \) if \( \delta \in \mathcal{G} \). We model each class by a mixture of such distribution, including a small proportion of noise. More precisely, for a class \( C_i, i = 1, 2, 3 \), we define

\[
\mu_i(I) = \frac{q}{3} (\mu(I; \mathcal{F}_1^i) + \mu(I; \mathcal{F}_2^i) + \mu(I; \mathcal{F}_3^i)) + (1-q)\mu(I; \emptyset)
\]

with \( q = 0.9 \) and

\[
\mathcal{F}_1^1 = \{\delta_1; \delta_3; \delta_5; \delta_7\} \quad \mathcal{F}_1^2 = \{\delta_1; \delta_5\} \quad \mathcal{F}_1^3 = \{\delta_3; \delta_7\}
\]
\[
\mathcal{F}_2^1 = \{\delta_2; \delta_4; \delta_6; \delta_8\} \quad \mathcal{F}_2^2 = \{\delta_2; \delta_4\} \quad \mathcal{F}_2^3 = \{\delta_6; \delta_8\}
\]
\[
\mathcal{F}_3^1 = \{\delta_1; \delta_4; \delta_6; \delta_9\} \quad \mathcal{F}_3^2 = \{\delta_1; \delta_8\} \quad \mathcal{F}_3^3 = \{\delta_4; \delta_9\}
\]

We expect our learning algorithm to put large weights on features in \( \mathcal{F}_i^j \) and ignore the other ones.

The algorithm \( A \) we use in the case is a \( p \) nearest neighbour classification algorithm, with distance given by

\[
d(I_1, I_2) = \sum_{\delta \in \omega} \mathbb{1}_{\delta(I_1) \neq \delta(I_2)}
\]

This toy example is interesting because it is possible to compute the exact gradient of \( \mathcal{E} \) for small values of \( M \) and \( k = |\omega| \). See [Gad04] and [GY04] for more details on this experiment when the set of features is fixed. We can compare the reflected diffusion with the one with jumps and modifications of the spaces of features. Performances obtained with the algorithm of jump diffusion are better than the ones obtained without any jump. The trees constructed by our algorithm are deeper than elementary ones since the mean depth achieved by our
algorithm is 3. Moreover, we compute the mean occupation measure of each tree in the process $F_t$

$$\mu_t(A) = \frac{1}{t} \int_0^t \mathbb{1}_{A \in F_t} \, ds$$

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

$$\{\delta_2; \delta_4\} \quad \{\delta_1; \delta_5\} \quad \{\delta_4; \delta_9\} \quad \{\delta_1; \delta_8\} \quad \{\delta_6; \delta_8\} \quad \{\delta_3; \delta_7\} \quad \delta_1 \quad \delta_4 \quad \delta_8$$

It is important to remark that the nodes selected by our jump diffusion algorithm are very similar to the sources $F_i^j$ while the favored nodes are those which are reusable features.

### 8.2 Face recognition

We use in this section the face database from [MIT], which contains $19 \times 19$ gray level images. The elementary features in $F_0$ are simply edges detectors constructed by Amit and Geman in [AG99].

![Figure 4: Sample of images taken from [MIT] database](image)

Efficiency of the reflected diffusion is already described in [GY04]. In this paper, our approach permit besides to largely improve error rates on the same datasets and we can easily give an interpretation of features constructed by our jump diffusion process. To instance these advantages, we can show 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 underneath 800. Even this number seems to be strictly decreasing in figure 5, this is not the case if we zoom the evolution of the cardinal $t \mapsto |F_t|$ as shown in figure 6.

Moreover, by using a linear SVM and a voting procedure with the subsets $\omega^{(i)}$ extracted with the process $F_t$, we obtain a false positive rate null (images taken from font class are perfectly classified) and the global misclassification rate is improved since we get 1.2\% misclassified test samples (3.5\% error rate with the fixed dictionary approach).

At last, 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
Figure 5: Evolution of the number of trees in the forest $F_t$ with time for the face recognition problem.

constructed and used by our algorithm. It is this point permit then to obtain the perfect false positive rate since these complex compositions of features filter out the background images.

8.3 Leukemia microarray classification

We benchmark our selection of features on the standard Leukemia cancer dataset downloaded from http://www.iitk.ac.in/kangal/bioinformatics. Data are preprocessed and transformed to a collection of 3859 genes of 72 leukemia samples. They are divided in 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 ($S$) and rebirth ($R$) of some genes in $F_t$.

We decide to stop our simulation when $(F_t)_{t \geq 0}$ do not change during a sufficiently long time. We then rank genes by decreasing importance criterion based on the probability distribution $P_t$. To estimate the misclassification rate for the reflected diffusion algorithm, we use a 10-fold Cross Validation method while we measure the efficiency of our jump algorithm by a classical train and test set method. We divide samples in a training set (60% of datas) and a test set (40%) which is independent from the training set. The Cross Validation method is less unbiased and consequently is a better way to estimate performances of our algorithm [BND03]. For the jump diffusion method, we do not run CV because of the time of computation needed by this method and we then employ a classical training set and test set.

We use for $\mathcal{A}$ a linear SVM classifier and $k = 100$ genes are extracted at each step. We obtain in [Gad05] interesting results on classification rates on this database.

Without any jump process, we need for instance to use only 4 genes to achieve a very low misclassification rate on the test set. We present in figure 9 results obtained using our jump process.
Figure 6: Microscopic evolution of the number of trees in the forest $F$.  

Figure 7: Representation of the main aggregation of edges detectors selected by our process  

Our results improve those refered in [GST+99] and the genes selected by our algorithm are consistent with some of the genes selected in other works (Zyxin). However, our results are nearly similar to those reported in [GDANW04]. Figure 10 represents the evolution of the number of genes selected at time $t$.  

Finally, we can extract from the set of variables the names of genes mostly selected by our algorithm. We obtain not exactly same results for the 10 most important genes as refered in the tabular of figure 11 with the two methods.  

9 Conclusion  

In this paper, we have given a precise rigorous method to select variables in a large amount of features dealing with the general untractable problems using full data. 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 tree is convenient to deal with Markov processes.  

This method is highly motivated by real problems and we have shown (theorem 7.2) the optimality of our algorithm since it converges towards the unique Gibbs field measure inferred
Figure 8: Mean error rate on the training set of Leukemia ALL-AML database with time $t$

<table>
<thead>
<tr>
<th>Number of genes</th>
<th>Reflected diffusion, fixed dictionary, CV estimation</th>
<th>Reflected Jump Diffusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-10</td>
<td>$\approx 6.9%$</td>
<td>$\approx 5%$</td>
</tr>
<tr>
<td>11-20</td>
<td>$\approx 5.5%$</td>
<td>$\approx 5%$</td>
</tr>
<tr>
<td>20-23</td>
<td>$\approx 5.5%$</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>$\approx 4.1%$</td>
<td>0</td>
</tr>
<tr>
<td>25-45</td>
<td>$\approx 2.8%$</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 9: Rate of misclassified samples using several number of genes selected by our algorithm from an energy $E$.

In a forthcoming paper, we will present several computational results on this algorithm applied to several datasets described by thousands of variables. We believe that using a Rademacher penalty term in energy $E$ will improve the generalization ability of the algorithm and can permit to obtain Oracle’s inequality.

References


Figure 10: Evolution of the number of trees in the forest $\mathcal{F}_t$ with iteration number $n$.

<table>
<thead>
<tr>
<th>Fixed dictionary algorithm</th>
<th>Reflected jump diffusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTSD Cathepsin D</td>
<td>CTSD Cathepsin D</td>
</tr>
<tr>
<td>MPO Myeloperoxidase</td>
<td>MPO Myeloperoxidase</td>
</tr>
<tr>
<td>MB-1 gene</td>
<td>MB-1 gene</td>
</tr>
<tr>
<td>Catalase (EC 1.11.1.6)</td>
<td>Catalase (EC 1.11.1.6)</td>
</tr>
<tr>
<td>PROTEASOME IOTA CHAIN</td>
<td>Kazal-type serine proteinase</td>
</tr>
<tr>
<td>Zyxin</td>
<td>PROTEASOME IOTA CHAIN</td>
</tr>
<tr>
<td>Terminal transferase mRNA</td>
<td>VIL2 Villin 2</td>
</tr>
<tr>
<td>Kazal-type serine proteinase</td>
<td>PRG1 Proteoglycan 1</td>
</tr>
<tr>
<td>CCND3 Cyclin D3</td>
<td>CD37 CD37 antigen</td>
</tr>
<tr>
<td>CD37 CD37 antigen</td>
<td>HLA CLASS I HISTOCOMPATIBILITY ANTIGEN</td>
</tr>
</tbody>
</table>

Figure 11: Genes mostly selected by our algorithm


Sébastien Gadat


