UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

Modèle de diffusion : génération de données synthétiques

Auteur : De Cock Nathan
Promoteur : Hainaut Donatien
Lecteur : Van Oirbeek Robin
Année académique 2024-2025

Remerciements

Je souhaite remercier toutes les personnes qui, de près ou de loin,m'ont aidé et soutenu tout au long de mon parcours académique.

Je remercie aussi Donatien Hainaut d'avoir non seulement accepté d'être mon promoteur dans le cadre de ce mémoire mais aussi pour le temps et l'énergie qu'il y a consacré. Ses observations et remarques étaient toutes aussi pertinentes les unes que les autres. Je tiens également à saluer Charlotte Jamotton pour les remarques fournies, ses conseils avisés et son partage de connaissances sur le sujet.

Enfin, pour terminer, je ne peux m'empêcher d'avoir une pensée pour ma famille qui m'a sans cesse soutenu tout au long de mon parcours scolaire.

Résumé

Le mémoire étudie les modèles de diffusion dans le cadre de la génération de données tabulaires. Une motivation théorique est présentée dans un premier temps. Le modèle est d'abord analysé dans le cadre de la génération d'images, puis adapté à la génération de données continues. Les modèles de diffusion multinomiaux sont introduits afin de gérer la génération de variables catégorielles. Le modèle global est ensuite appliqué à une base de données d'assureur. Enfin, une analyse des résultats est réalisée, avec un point d'attention particulier sur la qualité des données générées, grâce à l'introduction de nouvelles notions telles que le β-*recall* et l'α-*precision*.

Summary

This thesis studies diffusion models in the context of tabular data generation. A theoretical motivation is first presented. The model is initially analyzed in the context of image generation and then adapted for continuous data generation. Multinomial diffusion models are introduced to handle the generation of categorical variables. The overall model is subsequently applied to an insurance database. Finally, the results are analyzed, with particular attention paid to the quality of the generated data through the introduction of new concepts such as β-*recall* and α-*precision*.

Table des matières

Introduction

Au cours des dernières années, les technologies numériques sont devenues à la fois de plus en plus performantes et de plus en plus complexes, donnant des résultats très impressionnants. Cette croissance du numérique touche énormément de secteurs, tous différents les uns des autres : la finance, la santé, la science, les commerces, etc. Les intelligences artificielles devenant sans cesse davantage présentes autour de nous peuvent être vues comme un produit dont la matière première est constituée de données. En effet, sans données, il est impossible de développer un modèle prédictif, d'automatiser des rapports, etc. Les systèmes et machines devenant très performants permettent une capacité de calcul croissante. Si l'on conjugue cela avec les avancées théoriques dans le domaine de l'intelligence artificielle, les modèles conçus deviennent de plus en plus sophistiqués, permettant ainsi de traiter et de produire une grande quantité d'informations.

Cette croissance des technologies numériques et des intelligences artificielles engendre énormément de débats au sein de la société par rapport à la protection des données personnelles. Ce sont des débats qui questionnent la confidentialité et l'éthique de ces intelligences artificielles. Ces dernières années, il y a eu un certain nombre de scandales liés à la protection des données qui ont fait les gros titres des journaux. Cela a eu pour conséquence que le législateur renforce les lois liées à la protection des données personnelles. En Europe, on entend souvent parler du Règlement Général sur la Protection des Données, plus souvent appelé RGPD, qui impose aux entreprises des règles strictes par rapport à l'usage de données personnelles. Ces nouvelles règles engendrent des conséquences importantes au sein de certains secteurs qui, à l'avenir, vont devoir peut-être revoir leurs pratiques. Prenons le cas du secteur de l'assurance : une contrainte importante pourrait voir le jour : lorsqu'une police d'assurance quitte le portefeuille de l'assureur, les données liées à la police ne peuvent plus être utilisées par l'assureur. Cette contrainte pourrait mettre au défi tout le secteur de l'assurance, puisque les assureurs construisent leurs modèles de risque ou de provision, par exemple, sur base de leur historique sinistre.

Afin de contrer cette problématique, l'assureur pourrait produire des données artificielles ou synthétiques de telle manière à ce que la base de données générée contienne les mêmes propriétés statistiques que la base de données initiale. C'est une solution qui utilisera la puissance de l'apprentissage statistique tout en respectant les lois de plus en plus contraignantes en termes de confidentialité des données personnelles. Une idée est d'étudier les modèles de diffusion, qui sont des modèles souvent utilisés dans le cadre de la génération d'images synthétiques. C'est un modèle qui se base sur une approche probabiliste et s'inspire du processus physique de diffusion

et de renversement du bruit. Il est redoutablement efficace et permet, de manière progressive, de générer de nouvelles images à partir d'une image uniquement composée de bruit.

Le but de ce mémoire est dans un premier temps d'étudier le modèle de diffusion dans le cadre de la génération de données synthétiques pour des bases de données uniquement composées de variables continues. Par la suite, le mémoire se concentrera sur les variables discrètes. En effet, à l'inverse des images (qui sont en réalité juste un vecteur de variables continues où chaque pixel représente un chiffre entre 0 et 230 correspondant à la couleur du pixel), les variables discrètes n'ont pas les mêmes propriétés que les variables continues. Les modèles de diffusion ne fonctionnent pas pour les données discrètes puisque le modèle de diffusion repose sur une hypothèse de normalité. Il est donc nécessaire de pouvoir trouver une alternative aux modèles de diffusion pour la génération de bases de données synthétiques.

Ce travail s'articulera en trois parties. Dans la première partie, nous introduisons le cadre général des modèles de diffusion [1] [3]. Elle introduira et présentera le cadre des modèles de diffusion. Un cadre théorique sera posé pour ces modèles de diffusion, avec notamment un accent sur les preuves mathématiques menant au fait que l'entraînement du modèle repose simplement sur une minimisation d'une fonction d'erreur quadratique. L'objectif sera de comprendre les idées générales des modèles de diffusion et notamment d'illustrer le concept de bruitage et débruitage progressif sur lequel se basent ces modèles. Ensuite, une application du modèle sera réalisée pour générer des images synthétiques [5]. Enfin, le modèle sera appliqué sur un jeu de données assurantielles uniquement composé de variables continues.

Le deuxième chapitre se concentrera uniquement sur l'étude de la génération de données synthétiques dans le cadre de données discrètes (uniquement). Une motivation sera faite pour expliquer et comprendre pourquoi les modèles de diffusion ne s'appliquent pas aux variables catégorielles. Les modèles étudiés pour les données catégorielles seront des modèles multinomiaux basés sur l'article [4]. Ce sont des modèles où l'idée est fortement similaire aux modèles de diffusion, à savoir une distribution des données bruitées qui progresse avec un pas de temps en partant d'une distribution où chaque catégorie d'une variable est équi-distribuée. L'approche utilisée finalement sera plutôt similaire à celle de l'article [8], qui propose une méthode avec un pas de temps continu plutôt que discret. Cette méthode montrera des résultats plus encourageants. Cette méthode se nomme TabDiff. Enfin, le modèle sera appliqué sur un jeu de données assurantielles uniquement composé de variables discrètes.

Pour terminer, le dernier chapitre sera consacré à la combinaison des deux chapitres qui précèdent, à savoir la génération d'une base de données contenant des variables continues et catégorielles. Une combinaison des éléments des deux premiers chapitres sera réalisée pour produire un modèle capable de générer des données synthétiques contenant les propriétés de la base de données initiale. Une analyse des résultats obtenus en termes de qualité statistique et de pertinence pratique (c'est-à-dire ne pas uniquement faire un tirage aléatoire des lignes de la base de données initiale) sera réalisée. Des métriques telles que α-precision, β-recall et *Distance to Closest Record* seront donc introduites pour mesurer la qualité des données générées. Une analyse complète de la fréquence sinistre des données initiales et des données synthétiques sera faite afin de pouvoir juger la capacité du modèle génératif à capturer des tendances complexes du dataset initial. Cette

dernière section est également l'occasion de discuter les limites de notre approche, les perspectives d'amélioration et les prolongements possibles de ce travail.

Ce mémoire s'inscrit dans le cadre d'études en sciences des données. Il est préférable que le lecteur soit familier avec les concepts de base en machine learning, comme par exemple la modélisation prédictive, le concept de réseaux de neurones. Les différents modèles ont été implémentés en Python, en utilisant Google Colab, dans le but de pouvoir utiliser un GPU puisque le temps de calcul est conséquent au vu du grand nombre de paramètres que les modèles impliquent dans le cadre de ce travail. Les analyses descriptives et graphiques ont été réalisées en utilisant RStudio.

Chapitre 1

Modèle de diffusion probabilistique

Le premier chapitre sera consacré au modèle de diffusion probabiliste. Ce modèle est principalement utilisé pour la génération d'images. C'est une méthode relativement récente puisqu'elle a vu le jour en 2015 [1]. L'objectif de ce chapitre est donc de se familiariser avec cette technique générative tout en restant dans le contexte d'images synthétiques. Ce chapitre sera illustré d'exemples construits à partir d'une base de données comportant des images de fleurs.

FIGURE 1.0.1 – Exemples de fleurs issues de la base données utilisées pour l'illustration de ce chapitre.

Le point fort des modèles de diffusion est que le concept est facile à comprendre. Il faut entraîner un modèle de *deep learning* dans le but d'enlever le bruit que contient une image. Les modèles de diffusion se décomposent en deux grandes étapes :

— Le processus *forward q* qui va bruiter une image. Ce processus est important pour construire le modèle de *deep learning*.

— Le processus *reverse p* qui va retirer une partie du bruit que possède une image, avec comme objectif de la rendre plus nette.

Ces deux processus sont illustrés sur la figure 1.0.2. Sur cette figure, on part d'une image nette x_0, à laquelle du bruit est ajouté au fur et à mesure à l'aide du processus *forward*, afin d'obtenir une image x_T uniquement composée de bruit. Le chemin inverse, lui, est défini par le processus *reverse*, de telle manière à ce qu'une image x_T uniquement composée de bruit devienne petit à petit de plus en plus nette, afin de représenter une image "fictive" qui laisserait croire que cette image a été tirée aléatoirement dans le training set.

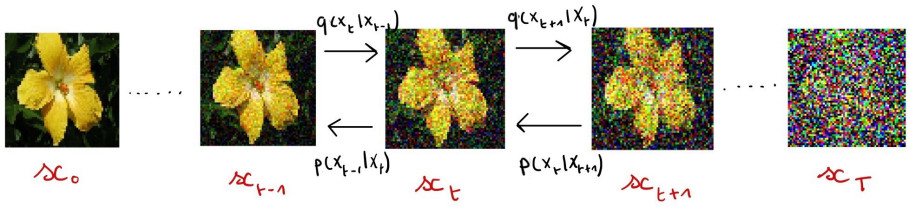


FIGURE 1.0.2 – Illustration

1.1 Processus *Forward*

Soit une image notée x_0. L'objectif est de modifier de manière progressive cette image afin d'obtenir une image x_T qui est un bruit gaussien.

Définition 1.1.1. Un bruit gaussien est une variable aléatoire dont la fonction de densité est celle d'une distribution Normale avec une moyenne nulle et une variance égale à 1.

Soit une fonction q qui va permettre d'ajouter graduellement du bruit à l'image x_0. En appliquant cette fonction T fois, un échantillon $(x_0, x_1, \cdots, x_T)$ d'images sera disponible avec des images qui seront de plus en plus bruitées. x_t avec $t \in \{1, \cdots, T\}$ est défini comme

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}, \tag{1.1.1}$$

où ϵ_{t-1} est une normale de moyenne 0 et de variance 1 et $\beta_t \in [0,1]$. β_t est appelé le taux de diffusion.

Cette équation est intéressante puisque si l'image x_0 est standardisée, alors x_t pour tout $t \in \{1, \cdots, T\}$ sera aussi gaussienne de moyenne 0 et de variance 1. En effet, si x_{t-1} est gaussien avec moyenne nulle et variance unitaire alors

$$\begin{aligned}\mathbb{E}(x_t) &= \sqrt{1-\beta_t}\mathbb{E}(x_{t-1}) + \sqrt{\beta_t}\mathbb{E}(\epsilon_{t-1}) \\ &= 0 + 0 = 0\end{aligned}$$

et puisque ϵ_{t-1} et x_{t-1} sont indépendants,

$$\begin{aligned}\mathbb{V}\text{ar}(x_t) &= (1-\beta_t)\mathbb{V}\text{ar}(x_{t-1}) + \beta_t\mathbb{V}\text{ar}(\epsilon_{t-1}) \\ &= 1 - \beta_t + \beta_t = 1.\end{aligned}$$

C'est une propriété importante puisque cela garantit que x_T est un bruit gaussien. Il est ainsi aisé de simuler une image x_T et appliquer le processus de diffusion *reverse*.

Il est donc possible définir la fonction $q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$. Ainsi, la fonction

de densité jointe $q(x_{1:T}|x_0)$ est définie comme

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{1.1.2}$$

Donc, x_t conditionellement à x_{t-1} est une variable aléatoire avec une distribution normale de moyenne $\sqrt{1-\beta_t}\, x_{t-1}$ et une variance $\beta_t\, \mathbf{I}$.

Proposition 1.1.2. *Si* $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ *et* $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ *alors* $X_1 + X_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$.

Il est également possible d'exprimer x_t pour tout $t \in \{1, \cdots, T\}$ en fonction de x_0. Soient $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, alors par l'équation (1.1.1)

$$\begin{aligned} x_t &= \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1}. \end{aligned}$$

Par la proposition 1.1.2,

$$\begin{aligned} x_t &= \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon \\ &= \cdots \\ &= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \end{aligned}$$

avec $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Donc,

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, 1 - \bar{\alpha}_t \mathbf{I}).$$

Définition 1.1.3. Un processus de Markov est un processus stochastique possédant la propriété de Markov : l'information utile pour la prédiction du futur est entièrement contenue dans l'état présent du processus et n'est pas dépendante des états antérieurs.

Le processus de *forward* est un processus de Markov c'est à dire que

$$q(x_t|x_{t-1}, x_0) = q(x_t|x_{t-1}). \tag{1.1.3}$$

C'est une propriété qui sera utile dans la suite de ce travail, notamment pour comprendre l'implémentation du processus *reverse*.

1.1.1 Le taux de diffusion

Le taux de diffusion β_t dépend du temps. En effet, dans les différents articles, β_t est fonction du temps. L'article [3] utilise $\beta_t = 0.0001 + 0.02t$ avec $t \in [0, 1]$. Cet article utilise un taux de diffusion linéaire.

Plus tard, un taux de diffusion cosinusoïdal a été préféré au taux de diffusion linéaire comme c'est

le cas dans le livre [5]. Le taux de diffusion cosinusoïdal est alors défini comme

$$\bar{\alpha}_t = \cos^2\left(\frac{t\pi}{2}\right).$$

Dans ce cas du taux de diffusion cosinusoïdal, il n'existe pas une forme fermée pour définir β_t. Dans la suite de ce travail, le taux de diffusion cosinusoïdal sera utilisé car il ajoute du bruit de manière plus parcimonieuse comparativement au taux de diffusion linéaire au début du processus *forward*. Ceci est illustré sur les figures 1.1.1 et 1.1.2.

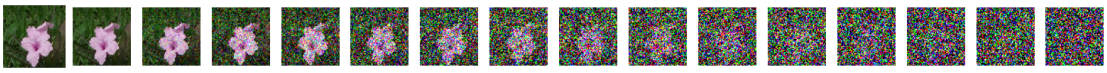


FIGURE 1.1.1 – Une image d'une fleur qui est de plus en plus bruitée avec un taux de diffusion cosinusoïdal.

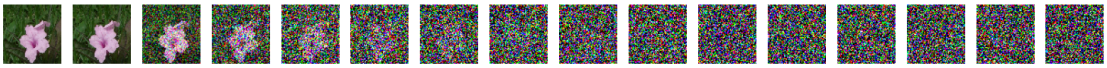


FIGURE 1.1.2 – Une image d'une fleur qui est de plus en plus bruitée avec un taux de diffusion linéaire.

La figure 1.1.3 compare l'évolution de $\bar{\alpha}_t$ et $1 - \bar{\alpha}_t$ pour les taux de diffusion linéaire et cosinusoïdal.

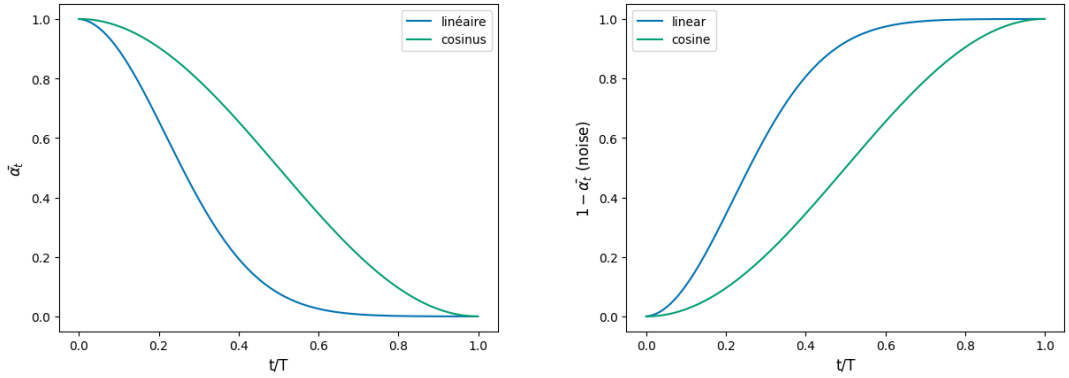


FIGURE 1.1.3 – $\bar{\alpha}_t$ et $1 - \bar{\alpha}_t$ en fonction de t.

1.2 Processus *Reverse*

Cette section sera consacrée au processus *reverse*. Le but de ce processus est inverse à celui du processus *forward*. Ce dernier avait pour objectif d'ajouter petit à petit du bruit afin d'obtenir à la

fin une image composée uniquement de bruit. Le processus *reverse* quant à lui a pour objectif de retirer du bruit au fur et à mesure. L'idée est donc de partir d'une image uniquement composée de bruit et d'ensuite, appliquer le processus *reverse* T fois. Chaque fois que le processus *reverse* est appliqué à une image, il va lui retirer une petite partie du bruit. En exècutant cela un grand nombre de fois, une image de plus en plus nette se dégagera et le bruit sera de moins en moins visible au fil des étapes.

Dans la section précédente, une image x_t pouvait s'exprimer comme

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon.$$

Appliquer le processus *forward* sur une image est trivial. Cela ne demande aucun apprentissage. Le processus *reverse* a pour objectif d'enlever le bruit d'une image x_t et ce processus repose sur une fonction $\epsilon_\theta(x_t, t)$ (un réseau de neurones) dépendante de t et de x_t. Ainsi, pour pouvoir calibrer le processus *reverse*, il faut entrainer un réseau de neurones à partir d'une image x_t et du taux de bruitage $\bar{\alpha}_t$ qui aura comme but de prédire ϵ. Les paramètres du réseau de neurones seront mis à jour via une méthode de descente du gradient.

L'idée est de trouver la fonction de densité $p_\theta(x_0) := \int p_\theta(x_{0:T})\mathrm{d}x_{1:T}$ où $x_1, \cdots, x_T$ sont des variables latentes. En pratique, elles correspondront aux images bruitées par le processus *forward*. La distribution jointe $p_\theta(x_{0:T})$ est appelée le processus *reverse* et est défini comme

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{1.2.1}$$

avec

$$p_\theta(x_{i-1}|x_i) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

et

$$p(x_T) = \mathcal{N}(x_T; 0, I).$$

La principale difficulté est donc de pouvoir trouver des estimateurs des fonctions $\mu_\theta(x_t, t)$ et $\Sigma_\theta(x_t, t)$.

Par définition de $p_\theta(x_{0:T})$, l'équation (1.2.1) et l'équation (1.1.2),

$$\begin{aligned} p_\theta(x_0) &= \int p_\theta(x_{0:T}) \frac{q(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} \mathrm{d}x_{1:T} \\ &= \int q(x_{1:T}|x_0) p_\theta(x_T) \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \mathrm{d}x_{1:T}. \end{aligned}$$

La fonction de log-vraissemblance est donnée par

$$\begin{aligned}\log L &= \int q(x_0)\ \log(p_\theta(x_0))\,\mathrm{d}x_0 \\ &= \int q(x_0)\ \log\Big(\int q(x_{1:T}|x_0)\ p_\theta(x_T)\ \prod_{t=1}^{T}\ \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t_1})}\,\mathrm{d}x_{1:T}\Big)\mathrm{d}x_0.\end{aligned}$$

Proposition 1.2.1. *Inégalité de Jensen : Soit f une fonction concave sur un intervalle I et X une variable aléatoire à valeurs dans I, dont l'espérance $\mathbb{E}(f(X))$ existe. Alors,*

$$f(\mathbb{E}(X)) \geqslant \mathbb{E}(f(X)).$$

Démonstration. Puisque f est une fonction concave, alors pour tout $x_* \in \mathbb{R}$ et $x_* \in \mathbb{R}$,

$$f(x) \leqslant f(x_*) + f'(x_*)(x - x_*).$$

Posons $x_* = \mathbb{E}(X)$. Alors

$$f(X) \leqslant f(\mathbb{E}(X)) + f'(\mathbb{E}(X))(X - \mathbb{E}(X)).$$

En appliquant l'espèrance de chaque coté,

$$\mathbb{E}(f(X)) \leqslant f(\mathbb{E}(X)).$$

□

Proposition 1.2.2. *Theorème de Bayes : Soient deux évènement A et B avec $P(B) > 0$ alors*

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}.$$

Par la proposition 1.2.1, le théorème de Bayes, l'équation (1.1.3) et les propriétés de la fonction

logarithmique,

$$\begin{aligned}
\log L &\geqslant \int q(x_{0:T}) \log \Big(p_\theta(x_T) \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \Big) \mathrm{d}x_{0:T} \\
&= \mathbb{E}_q \Big[\log \Big(p_\theta(x_T) \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \Big) \Big] \\
&= \mathbb{E}_q \Big[\log p_\theta(x_T) + \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \Big] \\
&= \mathbb{E}_q \Big[\log p_\theta(x_T) + \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} + \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)} \Big] \\
&= \mathbb{E}_q \Big[\log p_\theta(x_T) + \sum_{t=2}^{T} \log \Big(\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \Big) + \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)} \Big] \\
&= \mathbb{E}_q \Big[\log \frac{p_\theta(x_T)}{q(x_T|x_0)} + \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} + \log p_\theta(x_0|x_1) \Big].
\end{aligned}$$

Cette borne est appelé dans la littérature : *evidence lower bound (ELBO)*.

Proposition 1.2.3. *Dans le contexte de ce travail,*

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}|\tilde{\mu}(x_t, x_0), \tilde{\beta}_t \mathbf{I})$$

avec

$$\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t$$

et

$$\tilde{\beta}_t = \frac{(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \beta_t.$$

Démonstration. La preuve est faite dans le cas unidimensionnel, mais elle peut se généraliser dans $\mathbb{R}^n$.

Pour rappel,

- $q(x_{t-1}|x_0) = \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0, \bar{\alpha}_{t-1}\mathbf{I})$,
- $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I})$,
- $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, \bar{\alpha}_t \mathbf{I})$,
- La fonction de densité d'une variable aléatoire normale $\mathcal{N}(\mu, \sigma^2)$ est donnée par :

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2} \right).$$

Par le théorème de Bayes, les propriétés des probabilités conditionnelles et le fait que $q(x_t|x_{t-1}, x_0)$ est un processus de Markov :

$$\begin{aligned} q(x_{t-1}|x_t, x_0) &= \frac{q(x_{t-1}, x_t|x_0)}{q(x_t|x_0)} \\ &= \frac{q(x_t|x_{t-1}, x_0) \cdot q(x_{t-1}|x_0)}{q(x_t|x_0)} \\ &= \frac{q(x_t|x_{t-1}) \cdot q(x_{t-1}|x_0)}{q(x_t|x_0)}. \end{aligned}$$

Maintenant, en utilisant la définition d'une densité normale :

$$\begin{aligned} q(x_{t-1}|x_t, x_0) &= \frac{\frac{1}{\sqrt{2\pi\beta_t}} \exp\left(-\frac{(x_t-\sqrt{1-\beta_t}x_{t-1})^2}{2\beta_t}\right) \cdot \frac{1}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})}} \exp\left(-\frac{(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0)^2}{2(1-\bar{\alpha}_{t-1})}\right)}{\frac{1}{\sqrt{2\pi(1-\bar{\alpha}_t)}} \exp\left(-\frac{(x_t-\sqrt{\bar{\alpha}_t}x_0)^2}{2(1-\bar{\alpha}_t)}\right)} \\ &= \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})\beta_t}} \cdot \exp\left(\frac{-x_t^2 + 2\sqrt{1-\beta_t}x_t x_{t-1} - (1-\beta_t)x_{t-1}^2}{2\beta_t}\right. \\ &\quad + \frac{-x_{t-1}^2 + 2\sqrt{\bar{\alpha}_{t-1}}x_{t-1}x_0 - \bar{\alpha}_{t-1}x_0^2}{2(1-\bar{\alpha}_{t-1})} \\ &\quad \left. + \frac{x_t^2 - 2\sqrt{\bar{\alpha}_t}x_t x_0 + \bar{\alpha}_t x_0^2}{2(1-\bar{\alpha}_t)}\right) \\ &= \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})\beta_t}} \cdot \exp\left(\frac{1}{2(1-\bar{\alpha}_{t-1})\beta_t}\right. \\ &\quad \left(\frac{-(1-\bar{\alpha}_{t-1})x_t^2 + 2(1-\bar{\alpha}_{t-1})\sqrt{1-\beta_t}x_t x_{t-1} - (1-\bar{\alpha}_{t-1})(1-\beta_t)x_{t-1}^2}{1}\right. \\ &\quad + \frac{-\beta_t x_{t-1}^2 + 2\beta_t\sqrt{\bar{\alpha}_{t-1}}x_{t-1}x_0 - \beta_t\bar{\alpha}_{t-1}x_0^2}{1} \\ &\quad \left.\left. + \frac{(1-\bar{\alpha}_{t-1})\beta_t x_t^2 - 2\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})\beta_t x_t x_0 + (1-\bar{\alpha}_{t-1})\beta_t\bar{\alpha}_t x_0^2}{1-\bar{\alpha}_t}\right)\right). \end{aligned}$$

En utilisant le fait que $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, on regroupe les termes dans l'exponentielle :

— **Terme en** x_{t-1}^2 :

$$(-\alpha_t + \bar{\alpha}_t - 1 + \alpha_t)x_{t-1}^2 = (1-\bar{\alpha}_t)x_{t-1}^2$$

.

— **Termes en $x_{t-1}x_t$ et $x_{t-1}x_0$:**

$$\begin{aligned}
&\frac{2(1-\bar{\alpha}_t)(1-\bar{\alpha}_{t-1})\sqrt{1-\beta_t}x_t x_{t-1} + 2(1-\bar{\alpha}_t)\beta_t\sqrt{\bar{\alpha}_{t-1}}x_{t-1}x_0}{1-\bar{\alpha}_t} \\
&= 2(1-\bar{\alpha}_t)x_{t-1}\left(\frac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}x_0 + \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}x_t\right) \\
&= 2(1-\bar{\alpha}_t)x_{t-1}\cdot\tilde{\mu}(x_t, x_0).
\end{aligned}$$

— **Termes en x_0^2, $x_t x_0$ et x_t^2 :**

$$\begin{aligned}
&\frac{-(1-\bar{\alpha}_t)(1-\bar{\alpha}_{t-1})x_t^2 - (1-\bar{\alpha}_t)\beta_t\bar{\alpha}_{t-1}x_0^2 + (1-\bar{\alpha}_{t-1})\beta_t x_t^2}{1-\bar{\alpha}_t} \\
&\quad - \frac{2\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})\beta_t x_t x_0 - (1-\bar{\alpha}_{t-1})\beta_t\bar{\alpha}_t x_0^2}{1-\bar{\alpha}_t} \\
&= -(1-\bar{\alpha}_t)\left(\frac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}x_0 + \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}x_t\right)^2 \\
&= -(1-\bar{\alpha}_t)\cdot\tilde{\mu}(x_t, x_0)^2.
\end{aligned}$$

Finalement, on obtient :

$$q(x_{t-1}|x_t, x_0) = \frac{1}{\sqrt{2\pi\tilde{\beta}_t}}\exp\left(-\frac{(x_{t-1}-\tilde{\mu}(x_t, x_0))^2}{2\tilde{\beta}_t}\right)$$

Cela correspond bien à une densité normale de moyenne $\tilde{\mu}(x_t, x_0)$ et de variance $\tilde{\beta}_t$. □

Avant de continuer le développement de la fonction de log-vraissemblance, une nouvelle notion doit être comprise : la divergence de Kullback-Leibler.

Définition 1.2.4. Pour des distributions $\mathcal{P}$ et Q continues, la divergence de Kullback-Leibler (aussi appelée l'entropie relative) est définie comme

$$D_{KL}(P||Q) = \int p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x$$

où p et q sont les fonctions de densité de $\mathcal{P}$ et Q, respectivement.

La divergence de Kullback-Leibler est une distance statistique. Cela mesure la différence entre la distribution de $\mathcal{P}$ et la distribution de Q. Ce n'est cependant pas une métrique puisque cette mesure n'est pas symmétrique. La divergence de Kullback-Leibler est non négative et est nulle si $\mathcal{P}$ et Q sont identiques.

Dès lors par la définition 1.2.4,

$$\log L \geqslant \mathbb{E}\Big[- D_{KL}[q(x_T|x_0)||p_\theta(x_T)] \ - \sum_{t=2}^{T} D_{KL}[q(x_{t-1}|x_0,x_t)||p_\theta(x_{t-1}|x_t)] + \log p_\theta(x_0|x_1)\Big].$$

Afin de continuer le dévellopement de la borne inférieure de la fonction de vraissemlance, introduisons la définition de trace.

Définition 1.2.5. Pour tout matrice $A \in \mathbb{R}^{n\times n}$, la trace est

$$tr(A) = \sum_{i=1}^{n} A_{ii}.$$

La divergence de Kullback-Leibler admet plusieurs propriétés dont celles-ci dessous qui va permettre de continuer le développement de la fonction de vraissemblance.

Proposition 1.2.6. *Soient les variables aléatoires* $\mathcal{N}_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$ *et* $\mathcal{N}_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$ *avec* $\mu_1 \in \mathbb{R}^n$ *et* $\mu_2 \in \mathbb{R}^n$*, alors*

$$D_{KL}(\mathcal{N}_1||\mathcal{N}_2) = \frac{1}{2}\Big(\ln \frac{|\Sigma_2|}{|\Sigma_1|} - n + tr\Big\{\Sigma_2^{-1}\Sigma_1\Big\} + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)\Big).$$

Cette proposition va nous permettre de continuer à développer la fonction de vraisemblance. En effet, puisque

— par hypothèse, $p_\theta(x_{t-1}|x_t) \sim \mathcal{N}(x_{t-1}|\mu_\theta, \Sigma_\theta)$
— par la proposition 1.2.3, $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}|\tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I})$,

cette proposition va permettre de développer la partie $D_{KL}[q(x_{t-1}|x_0,x_t)||p_\theta(x_{t-1}|x_t)]$ de la borne.

La démonstration de la propriéte 1.2.6 repose sur deux résultats intermédiaires que nous énonçons ci-dessous. Elles seront ensuite utilisées pour démontrer la propriété principale.

Proposition 1.2.7. *Soient* $\mu \in \mathbb{R}^n$ *et* Σ *et* $A \in \mathbb{R}^{n\times n}$ *deux matrices symétriques. Si la variable aléatoire* $X \sim \mathcal{N}(\mu, \Sigma)$ *alors*

$$\mathbb{E}[X^TAX] = tr(A\Sigma) + \mu^TA\mu.$$

Démonstration. Puisque

$$X^TAX = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}X_iX_j,$$

alors

$$\mathbb{E}[X^T AX] = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\mathbb{E}[X_i X_j].$$

En utilisant le fait que

$$\begin{aligned}\mathbb{E}[X_i X_j] &= \mathbb{E}[X_i]\mathbb{E}[X_j] + Cov(X_i, X_j)\\ &= \mu_i\mu_j + \Sigma_{ij},\end{aligned}$$

on trouve que

$$\begin{aligned}\mathbb{E}[X^T AX] &= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}(\mu_i\mu_j + \Sigma_{ij})\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\mu_i\mu_j + \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\Sigma_{ij}\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\mu_i\mu_j + \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ji}\Sigma_{ij}\\ &= \mu^T A\mu + tr(A\Sigma).\end{aligned}$$

□

Proposition 1.2.8. *Pour toutes matrices $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times p}$ et $C \in \mathbb{R}^{p\times n}$,*

$$tr(ABC) = tr(BCA) = tr(CAB).$$

Démonstration. Par définition de la trace,

$$tr(ABC) = \sum_{i=1}^{n}(ABC)_{ii}.$$

On trouve facilement que pour tout $i \in \{1, \cdots, n\}$,

$$(ABC)_{ii} = \sum_{j=1}^{m}\sum_{k=1}^{p} A_{ij}B_{jk}C_{ki}.$$

Il en découle que

$$\begin{aligned} tr(ABC) &= \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p} A_{ij}B_{jk}C_{ki} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p} B_{jk}C_{ki}A_{ij} = tr(BCA) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p} C_{ki}A_{ij}B_{jk} = tr(CAB). \end{aligned}$$

□

Démonstration proposition 1.2.6. Afin de faciliter les notations, définissons

$$P \sim \mathcal{N}(\mu_1, \Sigma_1)$$

et

$$Q \sim \mathcal{N}(\mu_2, \Sigma_2).$$

Par définition de la divergence de Kullback-Leibler,

$$D_{KL}(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x.$$

On obtient donc que

$$\begin{aligned} D_{KL}(P||Q) &= \int_{\mathbb{R}^n} \mathcal{N}(x; \mu_1, \Sigma_1) \ln \frac{\mathcal{N}(x; \mu_1, \Sigma_1)}{\mathcal{N}(x; \mu_2, \Sigma_2)} \mathrm{d}x \\ &= \mathbb{E}_{p(x)}\left[\ln \frac{\mathcal{N}(x; \mu_1, \Sigma_1)}{\mathcal{N}(x; \mu_2, \Sigma_2)} \right] \end{aligned}$$

Puisque par définition définition,

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right],$$

alors

$$\begin{aligned}
D_{KL}(P||Q) &= \mathbb{E}_{p(x)}\Big[\ln \frac{\frac{1}{\sqrt{(2\pi)^n|\Sigma_1|}}\exp\Big[-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\Big]}{\frac{1}{\sqrt{(2\pi)^n|\Sigma_2|}}\exp\Big[-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\Big]}\Big] \\
&= \mathbb{E}_{p(x)}\Big[\frac{1}{2}\ln\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\Big] \\
&= \frac{1}{2}\mathbb{E}_{p(x)}\Big[\ln\frac{|\Sigma_2|}{|\Sigma_1|} - (x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + (x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\Big]
\end{aligned}$$

Puisque pour tout $a \in \mathbb{R}, tr(a) = a$ et que $tr(ABC) = tr(BCA)$, alors

$$\begin{aligned}
D_{KL}(P||Q) &= \frac{1}{2}\mathbb{E}_{p(x)}\Big[\ln\frac{|\Sigma_2|}{|\Sigma_1|} - tr\Big(\Sigma_1^{-1}(x-\mu_1)^T(x-\mu_1)\Big) + tr\Big(\Sigma_2^{-1}(x-\mu_2)^T(x-\mu_2)\Big)\Big] \\
&= \frac{1}{2}\mathbb{E}_{p(x)}\Big[\ln\frac{|\Sigma_2|}{|\Sigma_1|} - tr\Big(\Sigma_1^{-1}(x-\mu_1)^T(x-\mu_1)\Big) + tr\Big(\Sigma_2^{-1}\Big\{xx^T - 2\mu_2x^T + \mu_2\mu_2^T\Big\}\Big)\Big].
\end{aligned}$$

En utilisant, la linéarité de l'espérance et de la trace, alors

$$\begin{aligned}
D_{KL}(P||Q) &= \frac{1}{2}\Big(\ln\frac{|\Sigma_2|}{|\Sigma_1|} - tr\Big\{\Sigma_1^{-1}\mathbb{E}_{p(x)}\Big[(x-\mu_1)^T(x-\mu_1)\Big]\Big\} + tr\Big\{\Sigma_2^{-1}\mathbb{E}_{p(x)}\Big[xx^T - 2\mu_2x^T + \mu_2\mu_2^T\Big]\Big\}\Big) \\
&= \frac{1}{2}\Big(\ln\frac{|\Sigma_2|}{|\Sigma_1|} - tr\Big\{\Sigma_1^{-1}\mathbb{E}_{p(x)}\Big[(x-\mu_1)^T(x-\mu_1)\Big]\Big\} \\
&+ tr\Big\{\Sigma_2^{-1}\Big(\mathbb{E}_{p(x)}\Big[xx^T\Big] - \mathbb{E}_{p(x)}\Big[2\mu_2x^T\Big] + \mathbb{E}_{p(x)}\Big[\mu_2\mu_2^T\Big]\Big)\Big\}\Big).
\end{aligned}$$

On en déduit que

$$\begin{aligned}
D_{KL}(P||Q) &= \frac{1}{2}\Big(\ln\frac{|\Sigma_2|}{|\Sigma_1|} - tr\Big\{\Sigma_1^{-1}\Sigma_1\Big\} + tr\Big\{\Sigma_2^{-1}\Big(\Sigma_1 + \mu_1\mu_1^T - 2\mu_2\mu_1^T + \mu_2\mu_2^T\Big)\Big\}\Big) \\
&= \frac{1}{2}\Big(\ln\frac{|\Sigma_2|}{|\Sigma_1|} - tr\Big\{I_n\Big\} + tr\Big\{\Sigma_2^{-1}\Sigma_1\Big\} + tr\Big\{\Sigma_2^{-1}\Big(\mu_1\mu_1^T - 2\mu_2\mu_1^T + \mu_2\mu_2^T\Big)\Big\}\Big) \\
&= \frac{1}{2}\Big(\ln\frac{|\Sigma_2|}{|\Sigma_1|} - n + tr\Big\{\Sigma_2^{-1}\Sigma_1\Big\} + tr\Big\{\mu_1\Sigma_2^{-1}\mu_1^T - 2\mu_2\Sigma_2^{-1}\mu_1^T + \mu_2\Sigma_2^{-1}\mu_2^T\Big\}\Big) \\
&= \frac{1}{2}\Big(\ln\frac{|\Sigma_2|}{|\Sigma_1|} - n + tr\Big\{\Sigma_2^{-1}\Sigma_1\Big\} + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)\Big).
\end{aligned}$$

□

Cette proposition a été démontrée par [9].

Exemple 1.2.9. Dans le cas univarié, soient les variables aléatoires, $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ et $X_2 \sim$

$\mathcal{N}(\mu_2, \sigma_2^2)$, alors

$$D_{KL}(X_1||X_2) = \frac{1}{2}\Big(\ln \frac{\sigma_2^2}{\sigma_1^2} - 1 + \frac{\sigma_1^2}{\sigma_2^2} + \frac{(\mu_2 - \mu_1)^2}{\sigma_2^2}\Big).$$

Cela semble assez intuitif avec la notion de divergence. Pour rappel la divergence KL mesure une distance entre deux distributions statistiques. Pour deux variables aléatoires normales $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ et $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$, on observe que

— Si $\mu_1 = \mu_2$ et $\sigma_1 = \sigma_2$, alors = $D_{KL}(X_1||X_2) = 0$.
— Si $\sigma_1 = \sigma_2$, alors plus $|\mu_1 - \mu_2|$ est grande plus $D_{KL}(X_1||X_2)$ sera grande.
— $D_{KL}(X_1||X_2) \geqslant 0$.

Les différents articles [5], [3] proposent pour des raisons numériques de poser $\Sigma_\theta(x_t, t) = \sigma_t^2 \mathbf{I}$ avec $\sigma_t^2 = \beta_t$. Notons également que dans cette section β_t sont des constantes. En réalité, ce sont des paramètres mais ils sont considérés comme connus pour le moment.

Par les propositions 1.2.3 et 1.2.6,

$$L_{t-1} := \mathbb{E}\Big[D_{KL}[q(x_{t-1}|x_0, x_t)||p_\theta(x_{t-1}|x_t)]\Big] = \mathbb{E}\Big[\frac{1}{2\sigma^2}||\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2\Big] + C,$$

où C est une constante qui ne dépend pas de θ.

Dès lors, puisque $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$,

$$\begin{aligned}
L_{t-1} - C &= \mathbb{E}\Big[\frac{1}{2\sigma^2}||\tilde{\mu}_t(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon)) - \mu_\theta(x_t, t)||^2\Big] \\
&= \mathbb{E}\Big[\frac{1}{2\sigma^2}||\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon) + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t - \mu_\theta(x_t, t)||^2\Big] \\
&= \mathbb{E}\Big[\frac{1}{2\sigma^2}||\frac{-\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{1 - \bar{\alpha}_t\bar{\alpha}_t}}\epsilon + \frac{\beta_t + \alpha_t(1 - \bar{\alpha}_{t-1})}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}}x_t - \mu_\theta(x_t, t)||^2\Big] \\
&= \mathbb{E}\Big[\frac{1}{2\sigma^2}||\frac{-\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{1 - \bar{\alpha}_t\bar{\alpha}_t}}\epsilon + \frac{1 - \alpha_t + \alpha_t - \bar{\alpha}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}}x_t - \mu_\theta(x_t, t)||^2\Big] \\
&= \mathbb{E}\Big[\frac{1}{2\sigma^2}||\frac{-\beta_t}{\sqrt{1 - \bar{\alpha}_t\alpha_t}}\epsilon + \frac{1 - \bar{\alpha}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}}x_t - \mu_\theta(x_t, t)||^2\Big] \\
&= \mathbb{E}\Big[\frac{1}{2\sigma^2}||\frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon\Big) - \mu_\theta(x_t, t)||^2\Big].
\end{aligned}$$

Cette dernière équation signifie que $\mu_\theta(x_t, t)$ doit prédire $\frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\Big)$ pour x_t donné. Puisque x_t est connu grâce au processus *forward*, alors on définit

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon_\theta(x_t, t)\Big)$$

où ϵ_θ est une fonction qui est sensée prédire ϵ pour x_t donné. Donc,

$$\begin{aligned} L_{t-1} - C &= \mathbb{E}\Big[\frac{\beta^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2\Big] \\ &= \mathbb{E}\Big[\frac{\beta^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}||\epsilon - \epsilon_\theta(x_t, t)||^2\Big]. \end{aligned}$$

L'idée est donc de maximiser la fonction de log-vraissemblance. Il n'est pas possible de trouver une forme fermée pour cette fonction. Cependant, il a été possible de trouver une borne inférieure à cette fonction. L'idée est donc de maximiser cette borne pour calibrer le modèle.

$$\max_\theta \log L \geqslant \max_\theta \mathbb{E}\Big[-D_{KL}[q(x_T|x_0)||p_\theta(x_T)] - \sum_{t=2}^{T} D_{KL}[q(x_{t-1}|x_0,x_t)||p_\theta(x_{t-1}|x_t)] + \log p_\theta(x_0|x_1)\Big].$$

Étant donné que $p_\theta(x_T) = \mathcal{N}(x_T; 0, \mathbf{I})$ et que $q(x_T|x_0)$ ne dépend pas de θ,

$$\max_\theta \log L \geqslant \max_\theta \mathbb{E}\Big[- \sum_{t=2}^{T} D_{KL}[q(x_{t-1}|x_0,x_t)||p_\theta(x_{t-1}|x_t)] + \log p_\theta(x_0|x_1)\Big] + C.$$

$$\max_\theta \log L \geqslant \mathbb{E}\Big[- \sum_{t=2}^{T} \min_\theta D_{KL}[q(x_{t-1}|x_0,x_t)||p_\theta(x_{t-1}|x_t)]\Big] + C.$$

Il faut dès lors minimiser $\mathbb{E}\Big[\sum_{t=2}^{T} D_{KL}[q(x_{t-1}|x_0,x_t)||p_\theta(x_{t-1}|x_t)]\Big]$. Puisque la divergence de Kullback-Leibler mesure une distance entre deux distributions statistiques, il faut donc trouver la fonction p_θ tel que pour tout $t \in \{1, \cdots, T\}$, $D_{KL}[q(x_{t-1}|x_0,x_t)||p_\theta(x_{t-1}|x_t)]$ soit la plus petite. Au final, cela reste assez intuitif : de manière informelle, on veut que la distribution de x_{t-1} soit semblable sous q que la distribution de x_{t-1} sous p_θ.

Ce qui est intéressant dans ce résultat c'est que la maximisation de la borne inférieure repose tout simplement sur la minimisation d'une simple fonction d'erreur quadratique. En effet, maximiser la borne inférieure revient à minimiser pour tout $t \in \{1, \cdots, T\}$,

$$||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2.$$

Cela veut dire que le processus repose sur la minimisation de l'erreur moyenne quadratique entre le bruit ajouté et la prédiction du bruit ϵ_θ prédite par le modèle. C'est un fait assez intéressant puisque la génération d'images peut sembler être un processus bien complexe alors que la fonction objectif est tout simplement une fonction classique et simple.

Remarque 1.2.10. Jusqu'ici, nous avons considéré que t prenait des valeurs discrètes dans l'ensemble $\{1, 2, 3, \ldots, T\}$. Cependant, il est possible de définir t comme un nombre continu entre 0 et 1. Ainsi, au lieu d'utiliser $T = 1, 2, \ldots, T$, nous pourrions dire que t prend des valeurs

dans l'ensemble $\left\{\frac{1}{T}, \frac{2}{T}, \frac{3}{T}, \ldots, 1\right\}$. Cette approche permet de généraliser le processus, rendant la notation valide pour toutes les étapes t et peu importe la valeur de T. Nous noterons que $x_t = x_{\frac{t}{T}}$.

1.3 Génération nouvelle image

Pour générer une nouvelle image x_0, il faut partir d'un bruit x_T. Sur cette image x_T, l'idée est toujours la même il faut appliquer de manière graduelle le processus *reverse* afin d'obtenir une image x_t de plus en plus nette.

Pour générer aléatoirement x_{t-1} connaissant x_t, il faut générer la variable aléatoire $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$. Dans la section précédente, $p_\theta(x_{t-1}|x_t)$ était défini comme

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

avec $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$ et $\Sigma_\theta(x_t, t) = \beta_t \mathbf{I}$. Dès lors,

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \beta_t z$$

avec $z \sim \mathcal{N}(0, \mathbf{I})$.

1.4 Génération d'images synthétiques

Cette section détaillera les trois grandes étapes à appliquer pour générer de nouvelles images. Comme expliqué depuis le début de ce chapitre, les modèles de diffusion sont populaires grâce aux résultats générées pour la génération d'images synthétiques. Avant de se concentrer sur les données tabulaires, quelques illustrations d'images synthétiques seront illustrées dans cette section.

Les données utilisées pour l'application d'un modèle de diffusion sur des images synthhétiques est le dataset 102 Flower qui est disponible sur Kaggle (https://www.kaggle.com/datasets/nunenuh/pytorch-challange-flower-dataset). C'est un dataset qui contient 8000 images de fleurs. Les images sont composées de pixels 64×64 avec chaque pixel qui est une valeur entre 0 et 1. La valeur du pixel correspond à la couleur de ce dernier. De manière informelle, une image est un vecteur numérique.

1.4.1 Entrainement du modèle

Le modèle de diffusion repose sur deux processus : le processus *forward* et le processus *reverse*. Le premier sert à bruiter petit à petit une image afin d'obtenir une image uniquement composée de bruit. Le second à pour objectif de faire l'inverse. Le processus *reverse* retire le bruit d'une image au fur et à mesure afin d'au final obtenir une une image de plus en plus nette.

Maintenant, l'objectif principal est de pouvoir générer de nouvelles images. La génération d'images comprend deux étapes :

1. Comme expliqué en détail auparavant, le processus *reverse* dépend d'un réseau de neurones ϵ_θ. Ce réseau de neurones ϵ_θ prend comme input une image bruitée ainsi que le moment t de la diffusion. Puisque le réseau de neurones cherche à prédire le bruit contenu dans l'image initial au moment $t \in [0, 1]$, il n'est pas nécessaire de simuler t de manière discrète. Cela va permettre d'utiliser le même réseau de neurones pour des générations avec un nombre d'étapes T différent. De plus, cela va permettre de mieux généraliser le processus. L'output de ce réseau de neurones est le bruit ϵ ajouté à l'image initiale. Ci-dessous se trouve l'algorithme pour l'entrainement du réseau de neurones ϵ_θ en pseudo-code.

Algorithm 1 L'algorithme pour l'entraînement du modèle pour le processus *reverse*.

Répéter
$x_0 \sim q(x_0)$ ▷ Image au hasard dans le dataset
$t \sim \text{Uniform}(0, 1)$ ▷ Simuler t
$\epsilon \sim \mathcal{N}(0, \mathbf{I})$ ▷ Simuler ϵ quit une Normale
Ajuster les paramètres avec la méthode du gradient
$\Delta_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon)||^2$ ▷ Minimiser la distance entre l'output et ϵ
Convergence

2. Une fois le réseau de neurones calibré, il suffit de simuler un bruit gaussien et d'appliquer le processus *reverse* de manière récursive pour générer une nouvelle image. Voici ci-dessous le pseudo code pour générer une nouvelle image.

Algorithm 2 L'algorithme de génération d'une nouvelle image.

Choisir T
$t = 1$
$x_T \sim N(0, \mathbf{I})$
while t > 0 **do**
$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$
$t = t - \frac{1}{T}$
end whilereturn x_0

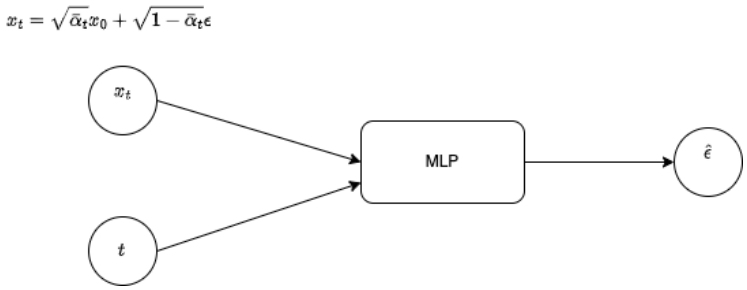


FIGURE 1.4.1 – Schéma du réseau de neurones ϵ_θ.

1.4.2 Exemple d'images synthétiques

Pour la génération d'images, il faut appliquer le processus de diffusion *reverse*. Il faut partir d'une image uniquement composée de bruit et utiliser le modèle pour ensuite, enlever le bruit de manière progressive. Normalement, après plusieurs étapes, la silhouette d'une fleur apparaîtra.

Il faut bien garder en tête que le réseau de neurone est construit de manière à prédire la totalité du bruit ϵ qui compose l'image bruitée. Cependant, la méthode ne va pas juste déduire directement l'image finale à partir de l'image initiale (uniquement composée de bruit). Elle va retirer le bruit de manière graduelle en plusieurs petites étapes.

Pour passer d'une image x_t à x_{t-1}, le modèle prédit le bruit $\epsilon_\theta(x_t, t)$ pour ensuite déduire $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \beta_t z$ avec $z \sim \mathcal{N}(0, \mathbf{I})$.

Lors de cette section, la forme du réseau de neurones n'est pas recherchée. La forme du réseau de neurones utilisée est celle fournie dans le livre [5]. Les auteurs utilisent une architecture de réseau de neurones qui s'appelle U-net. Ce sont des réseaux de neurones qui sont fortement appréciés lorsque l'input du réseau de neurones a une forme identique à l'output du réseau de neurones. C'est exactement ce qui se passe lors du modèle de diffusion puisque le bruit prédit a une forme identique à l'image bruitée en input. L'architecture d'un réseau U-net se décompose en deux. Dans la première partie, les images sont compressées (on réduit la taille) mais le nombre de canaux augmente. Dans la deuxième moitié du réseau, le nombre de canaux se réduit mais la taille des images augmente pour obtenir une taille identique à celle des images initiales. La figure 1.4.2 schématise de manière high-level la structure du réseau de neurones. En réalité, la structure est beaucoup plus compliquée.

Pour rappel, le réseau de neurones dispose du pas de temps t comme input. Une fonction appelée *Sinusoidal Embedding* est utilisée pour convertir le scalaire t en un vecteur de plus grande dimension. Cette technique est utilisée car un scalaire contient peu d'information exploitable pour un réseau de neurones complexe et de grande taille. Cette fonction va donc transformer ce scalaire t en vecteur de taille $2N$ en utilisant la formule suivante

$$\gamma_{\sin}(t) = \left(\sin(2\pi e^{0p}t), \cdots, \sin(2\pi e^{(N-1)p}t), \cos(2\pi e^{0p}t), \cdots, \cos(2\pi e^{(N-1)p}t)\right)$$

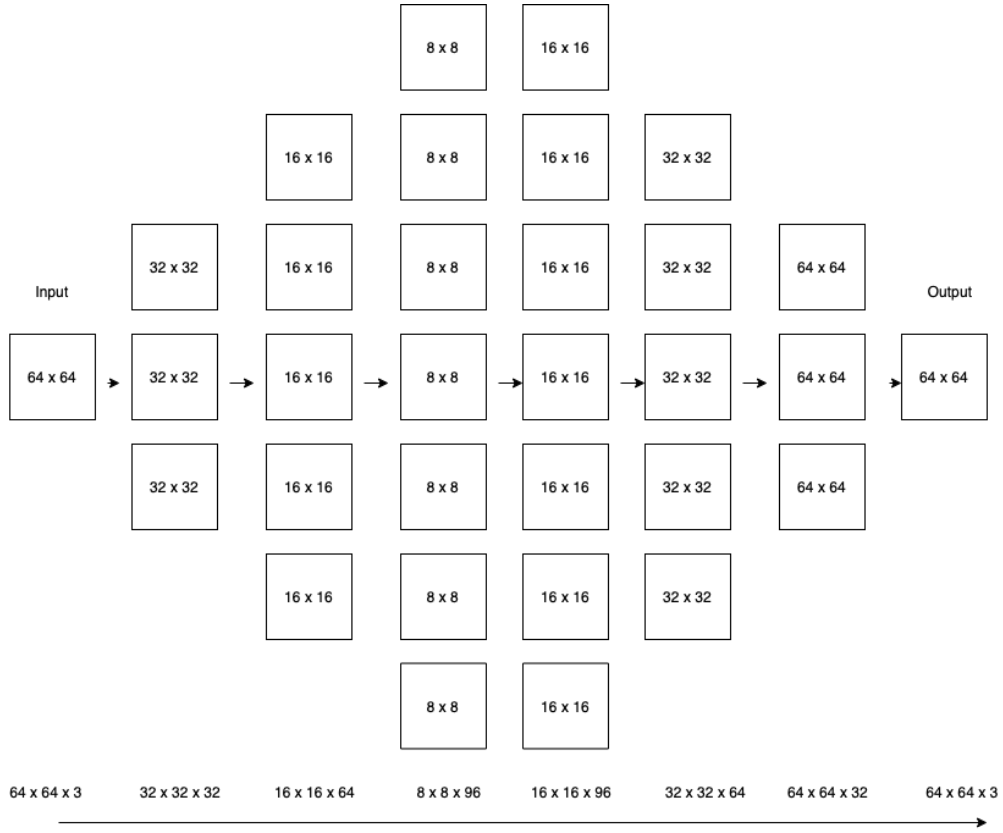


FIGURE 1.4.2 – Représention high-level de l'architecture d'un réseau U-net. Le réseau prend comme input une image de taille $64 \times 64 \times 3$ (1 canal par couleur (bleu, rouge et vert) avec la valeur du pixel qui détermine l'intensité) pour ensuite avoir plus de canaux mais de forme plus petite. Ensuite, le nombre de canaux diminue mais la taille des canaux augmente.

avec $p = \frac{\ln 1000}{N-1}$.

La figure 1.4.3 contient 10 images générées à l'aide du modèle de diffusion. Ces images montrent également l'importance d'avoir un nombre de pas suffisamment grand. Les images générées sont produites en utilisant $\{1, 2, 3, 4, 5, 20, 100, 200, 500, 1000\}$ étapes entre l'image initiale composée uniquement de bruit et l'image générée. La première ligne est un modèle de diffusion avec 1 seule étape. Les images produites avec une seule étape ne sont pas fortement lisibles et paraissent troubles. Les images générées ne sont pas nettes. Il est difficile de distinguer des fleurs. Cela montre bien ce qui était expliqué plus haut avec le fait que prédire une image directement depuis une image composée uniquement de bruit gaussien ne donne pas des résultats satisfaisants. On constate que plus le nombre d'étape augmente, plus les images deviennent nettes. Les fleurs sont bien visibles et les images produites donnent des résultats très satisfaisants. Un autre constat est qu'à vue d'oeil, la différence entre les images synthétiques avec 100 étapes donnent des résultats assez similaires aux images synthétiques produites avec 1000 étapes. L'image montre 10 images

synthétiques de fleurs mais on constate également que ces images sont bien différentes les unes des autres. Le modèle semble être bien diversifié dans le sens où les images produites sont des images de nature différentes. Les fleurs sont de forme et de couleurs différentes. Idéalement, il aurait fallu vérifier avec le dataset original, que les fleurs produites ne soient pas juste la réplication de fleurs contenues dans la base de données utilisées pour entraîner le modèle. Cela demanderait beaucoup de temps au vu du nombre d'images que le dataset comprend.

Les images de la figure 1.4.3 ont été produites avec un réseau de neurones contenant 1.953.507 paramètres. C'est donc relativement grand comparé aux réseaux de neurones utilisées pour faire des modèles prédictifs par exemple. Vu le grand nombre de paramètres, le temps pour calibrer le réseau de neurones est assez élevé.

FIGURE 1.4.3 – Les images générées sont produites en utilisant $\{1, 2, 3, 4, 5, 20, 100, 200, 500, 1000\}$ étapes entre l'image initiale composée uniquement de bruit et l'image générée.

1.5 Application du modèle de diffusion aux données continues

Comme expliqué dans l'introduction, l'objectif est de pouvoir générer des données synthétiques à partir d'une base de données initiales. Les bases de données contiennent des variables continues et discrètes. Générer les variables continues est facile via le modèle de diffusion. Une image x_0 étant un vecteur (en 3 dimensions) numérique où chaque élément du vecteur représente une valeur numérique correspondant à la couleur du pixel. Appliquer le modèle de diffusion aux variables continues est une simple application du modèle de diffusion construit dans le cadre de la génération d'images synthétiques. Le processus est donc identique à ce qui a été présenté jusqu'ici.

1.5.1 Présentation des données continues

Dans le cadre de ce mémoire, une base de données relative à l'assurance auto est utilisée. C'est une base de données visant à modéliser la fréquence sinistre d'une police d'assurance en fonction des caractéristiques de cette dernière. Cette base de données contient 5 variables continues :

1. **L'exposition** : Une valeur strictement positive qui représente le nombre d'année où la police est restée dans le portefeuille.
2. **L'age du véhicule** : Une valeur positive représentant l'âge du véhicule.
3. **L'age de l'assuré** : Une valeur positive représentant l'âge de l'assuré.
4. **La densité** : Le nombre de personnes au 10 000 mètres carrés dans la ville où réside l'assuré.
5. **Bonus malus** : Un score entre 0 et 350 représentant la sinistralité passée de l'assuré. Plus le score est bas moins l'assuré à un profil à "risque" pour l'assureur.

L'exposition, l'âge du véhicule, l'âge de l'assuré et la densité sont les quatres variables conservées pour la suite. L'échelle bonus-malus n'est pas utilisée puisque celle-ci étant une variable "posterior" n'est pas souvent utilisée par un assureur pour modéliser la fréquence sinistre d'un assuré. L'objectif final étant de construire un modèle sur les données générées et un autre modèle sur les données initiales, les variables uniquement utilisées en pratique sont gardées.

La table 1.5.1 reprend le minimum, le quantile 25%, la moyenne, la médiane, le quantile 75% et le maximum pour chacune des quatre variables continues.

Statistique	**VehAge**	**DrivAge**	**Density**	**Exposure**
Minimum	0.00	18.00	1	0.26
Quantile 25%	3.00	35.00	84	0.49
Médiane	6.00	46.00	338	0.77
Moyenne	7.27	46.63	1685	0.73
Quantile 75%	11.00	56.00	1445	1.00
Max.	24.00	100.00	27000	2.01

TABLE 1.5.1 – Statistiques descriptives des variables `VehAge`, `DrivAge`, `Density` et `Exposure`

Il est également possible de visualiser la densité pour chacune des 4 variables continues sur la figure 1.5.1. Il est notable que les quatre variables ont des distributions bien différentes les unes des autres. En effet, elles prennent des formes différentes. Certaines ont des formes assez lisses comme l'âge du conducteur et l'âge du véhicule. A l'inverse, l'exposition et la densité ont des formes un peu plus irrégulières. Cela va permettre de tester la performance des modèles de diffusion sur différentes variables continues et, qui plus est, ont des distributions bien distinctes.

— `VehAge` : La densité est concentrée sur les premières années avec un pic autour de 1 ou 2 ans, puis décroît progressivement, indiquant que la majorité des véhicules sont relativement récents. On observe une légère augmentation autour de 10 ans.
— `DrivAge` : La densité est assez plate entre 30 et 60 ans, avec un léger maximum autour de 50 ans, ce qui suggère une forte concentration de conducteurs d'âge moyen.
— `Density` : La distribution est extrêmement asymétrique à droite, avec une concentration très élevée sur les faibles valeurs de densité. Cela indique que la majorité des observations proviennent de zones peu denses.
— `Exposure` : On remarque un gros poids en 1, ce qui montre que la plupart des polices d'assurance couvrent une année complète. Quelques valeurs inférieures traduisent des périodes de couverture plus courtes.

1.5.2 Preprocessing sur les données continues

Il a été montré auparavant que dans le contexte des modèles de diffusion que pour tout $t \in \{1, \cdots, T\}$ alors

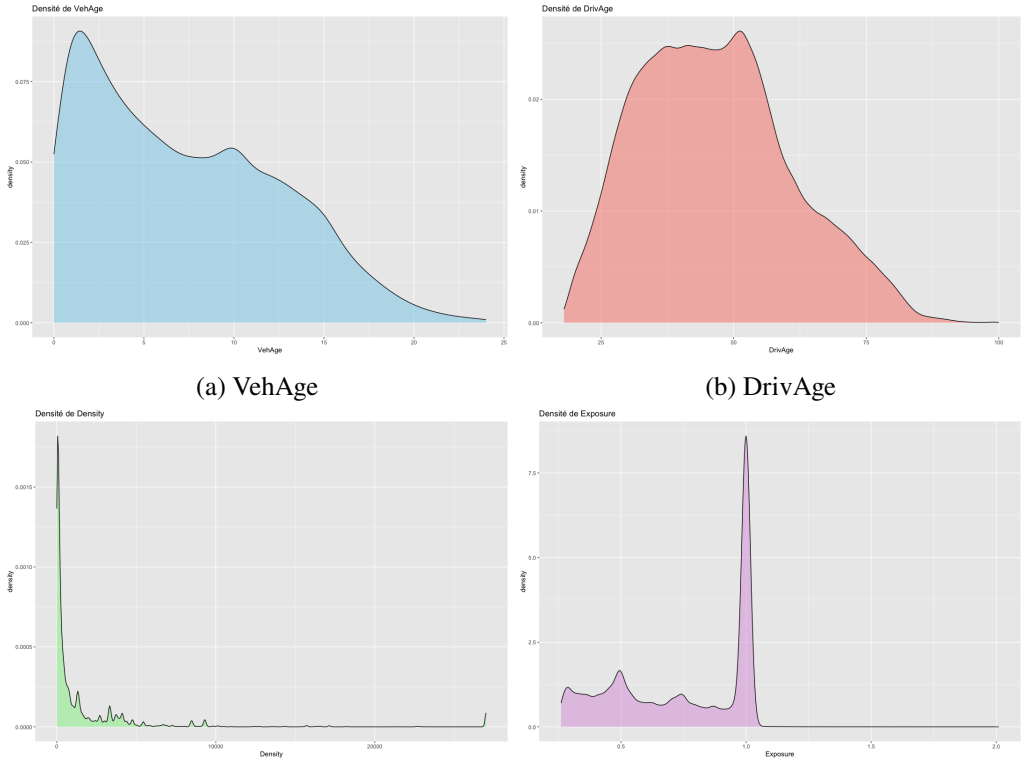
1. $\mathbb{E}(x_t) = 0$,
2. $\mathbb{V}ar(x_t) = 1$,
3. $p(x_t) \sim Normale$.

Il est évidemment peu probable que les données initiales respectent ces trois hypothèses. Il est donc nécessaire de manipuler les données initiales afin que ces hypothèses soient respectées. Une transformation quantile sera appliquée et une standardisation des données sera effectuée sur les données initiales.

Transformation quantile

La première étape est d'appliquer une transformation quantile aux données afin de faire en sorte que les données suivent une distribution normale. C'est une méthode qui est souvent utilisée en sciences de données pour réduire l'influence des valeurs extrêmes d'une variable, ou comme dans le cas des modèles de diffusion, pour rendre les données compatibles aux hypothèses d'un modèle.

Définition 1.5.1. Supposons $\{X_t\}_{t=1,\cdots,N}$, un échantillon d'une variable aléatoire inconnue X et Y la distribution d'une variable alléatoire (connue). La transformation quantile se déroule en deux étapes :



(a) VehAge (b) DrivAge

(c) Density (d) Exposure

FIGURE 1.5.1 – Densité des variables continues

1. Pour tout $t \in \{t = 1, \cdots, N\}$, calculer $F_X(X_t)$ où F_X est la fonction de répartion empirique de l'échantillon $\{X_t\}_{t=1,\cdots,N}$.
2. Pour tout $t \in \{t = 1, \cdots, N\}$, définir Y_t comme le quantile $F_X(X_t)$ de la distribution Y.

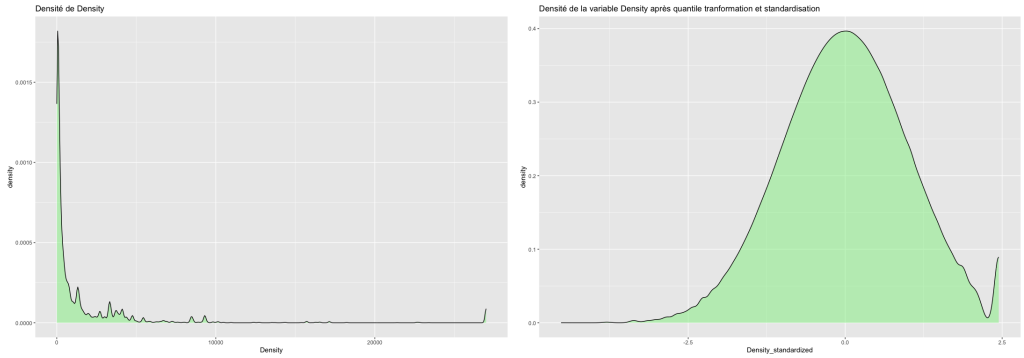
Standardisation des données

La standardisation des données consiste à centrer et réduire les données initiales afin que celles-ci aient une moyennne égale à 0 et une variance égale à 1.

Définition 1.5.2. Soit $\{x_t\}_{t=1,\cdots,N}$, un échantillon d'une variable aléatoire X. L'échantillon standardisée $\{z_t\}_{t=1,\cdots,N}$ est défini comme

$$z_t = \frac{x_t - \bar{x}}{\sigma_X}$$

où $\bar{x}$ et σ_X sont la moyenne et l'écart-type de l'échantillon $\{x_t\}_{t=1,\cdots,N}$, respectivement.



(a) La densité de la variable densité avant le preprocessing.

(b) La densité de la variable densité après le preprocessing.

FIGURE 1.5.2 – Comparaison de la densité de la variable Densité avant et après avoir effectué une transformation quantile et une standardisation.

1.5.3 Entraînement sur modèle

Précedemment, le modèle de diffusion a été appliqué dans le cadre de la génération d'images synthétiques. Le but du mémoire étant de générer des données tabulaires, il sera à présent appliqué sur les données présentées ci-dessus dans le but de produire un nouveau dataset contenant les mêmes propriétés statistiques que le dataset initial. Pour produire le dataset synthétique, 428.000 données seront utilisées. 80% des données formeront les données pour l'entraînement du modèle et les 20% restantes seront les données qui formeront le testing set. La taille de l'embedding donnant au réseau la connaissance de l'instant t du processus de diffusion est de 16. La forme du réseau de neurones utilisée est un réseau de neurones avec 4 couches entièrement connectées avec comme fonction d'activation la fonction ReLU.

Définition 1.5.3. La fonction d'activation $ReLU$ est définie par

$$ReLU(x) = \max(0, x).$$

Un nombre de neurones décroissant par couches sera utilisé (512, 256,128,64). La dernière couche avec les 4 neurones finaux ne contient aucune fonction d'activation (ou fonction d'activation linéaire) puisque l'output peut prendre des valeurs dans $\mathbb{R}$. Le taux de diffusion sera un taux de diffusion cosinusoïdal. La taille du lot (batch size) est fixée à 4092, tandis que le taux d'apprentissage (learning rate) utilisé est de1.5×10^{-5}. L'entraînement sera effectué sur un total de 1000 itérations (epochs).

Dans un premier temps, puisque le temps pour calibrer le modèle est conséquent, une seule forme de réseau de neurones sera utilisée. En effet, ici, le nombre de paramètre est conséquent puisque le réseau de neurones contient exactement 183.492 neurones.$((4+16) \times 512 + 512 + 512 \times 256 + 256 + 256 \times 128 + 128 + 128 \times 64 + 64 + 64 \times 4 + 4$ neurones) En plus de cela, le dataset contient

un grand nombre d'observations.

TABLE 1.5.2 – Paramètres utilisée pour le modèle de diffusion pour la génération des données synthétiques avec uniquement des variables continues

Paramètre	**Valeur**
Taille du dataset utilisé	428 000 observations (80% entraînement, 20% test)
Taille de l'embedding	16
Nombre de couches cachées	4
Taille des couches cachées	$512 \to 256 \to 128 \to 64$
Couche de sortie	4 neurones sans activation (linéaire)
Taille du batch	4092
Taux d'apprentissage (learning rate)	1.5×10^{-5}
Nombre d'itérations (epochs)	1000
Taux de diffusion	Cosinusoïdal

Remarque 1.5.4. Il n'est pas nécessaire de donner au réseau de neurones la taille T représentant le nombre d'étapes pour effectuer la diffusion complète. En effet, une fois que le réseau de neurones est calibré, on peut utiliser ce même réseau de neurones pour générer des données synthétiques avec un nombre d'étapes de diffusion différent. Puisque le réseau $\epsilon_\theta(x_t, t)$ est calibré pour tout $t \in [0, 1]$, il peut par exemple être utilisé pour la génération avec 5 étapes de diffusion mais aussi pour la génération avec 100 étapes. Etant donné que dans le modèle de diffusion, l'entrée du réseau est donnée par la relation :

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$$

où x_0 désigne la donnée initiale, ϵ un bruit gaussien, et $\bar{\alpha}_t$ le taux de diffusion à l'instant t. Le réseau de neurones a pour objectif d'estimer le terme de bruit ϵ à partir de l'observation bruitée x_t et de l'instant t. On remarque que cette formulation est indépendante du nombre total d'itérations de diffusion T, puisque seul l'instant $t \in [0, 1]$ intervient dans le calcul. Cela permet de traiter t comme une variable continue, sans faire de lien avec le nombre total de pas T, ce qui rend le modèle flexible.

1.5.4 Génération de nouvelles données

Cette section sera consacrée à l'analyse de données synthétiques générées par le réseau de neurones décrit dans la section précédente. Le nombre d'étape T pour effectuer la diffusion sera de 1000. La taille du dataset généré fera 400.000 lignes. L'idée est donc de comparer les deux jeux de données.

La première chose à faire est de regarder si les données générées ne sont pas uniquement des données qui sont présentes dans le dataset original. Si c'est le cas, cela signifie que le dataset synthétique réplique juste les données initiales. Ce n'est évidemment pas l'objectif. Le but étant

de produire un dataset différent mais gardant les propriétés initiales. Si les données générées sont juste identiques aux données de départ, le modèle serait équivalent à tirer au sort 400 000 lignes du dataset original.

Afin d'être cohérent, puisque dans le jeu de données original les variables `DrivAge`, `VehAge` et `Density` sont arrondies à l'unité, ces variables dans le dataset synthétique sont arrondies à l'unité. La variable `Exposure` est arrondie à 2 chiffres après la virgule.

Parmi les 400 000 lignes générées, 5 879 lignes sont présentes dans le dataset original. Cela resprésente environ 2% des données synthétiques. Le modèle ne fait donc pas juste un simple copier-coller des données originales.

Le tableau 1.5.3 reprend les statistiques de base des deux jeux de données. On remarque que la moyenne et la médiane sont très bien respectées pour chacune des 4 variables. Par exemple, la moyenne de l'Exposition au risque est de 0.734 pour le jeu de données original, et de 0.74 pour le synthétique, ce qui signifie que le modèle capture bien la moyenne. De même, pour la médiane qui est quasiment identique. On aurait tendance à dire que le modèle capture assez bien la moyenne et la médiane des différentes variables. Les chiffres sont proches mais ne sont pas forcément exactement identiques. De plus, on ne remarque pas une tendance à la hausse ou à la baisse qui vont dans le même sens pour toutes les variables. En effet, pour la variable de l'exposition la moyenne synthétique est plus haute que la moyenne originale alors que pour les autres variables la moyenne synthétique est plus basse que la moyenne initiale.

Les quantiles 25% et 75% sont quasiment identiques pour toutes les variables. On observe une légère différence pour le quantile 75% de la variable liée à l'âge du conducteur et pour le quantile 25% liée à la variable de l'ancienneté de la voiture. Cela laisse sous-entendre que le modèle capture relativement bien les distributions des deux jeux de données. Le maximum et le minimum sont également bien représentés.

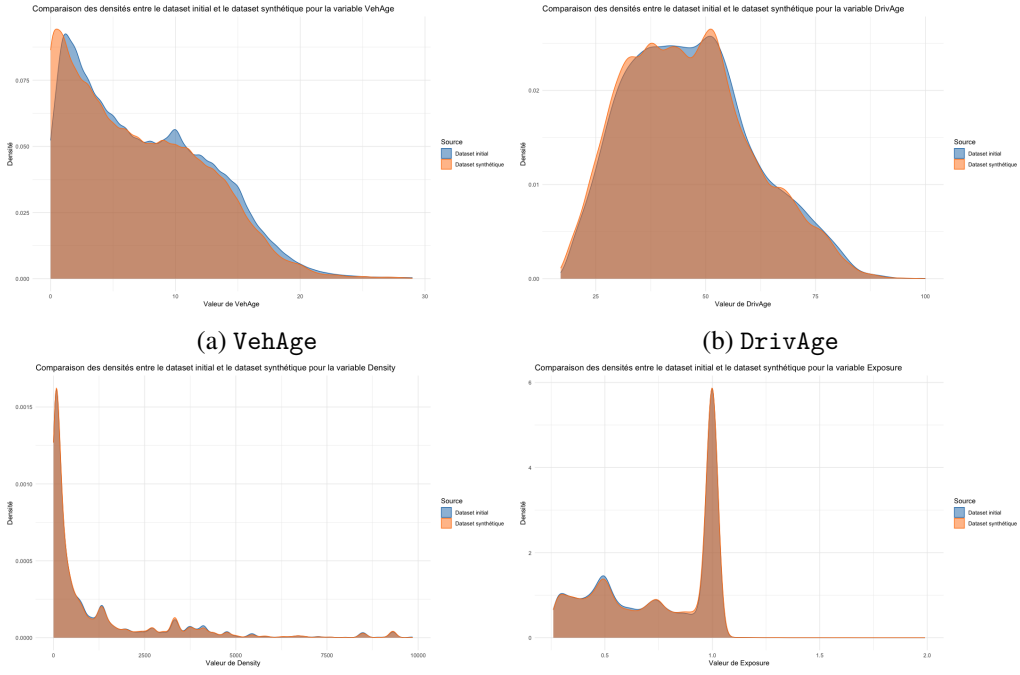
Statistique	**Dataset**	**Exposure**	**VehAge**	**DrivAge**	**Density**
Minimum	Initial	0.26	0	18	1
	Synthétique	0.26	0	17	3
Q25	Initial	0.49	3	35	81
	Synthétique	0.49	2	35	81
Médiane	Initial	0.78	7	46	302
	Synthétique	0.80	6	45	298
Moyenne	Initial	0.734	7.39	46.6	1159
	Synthétique	0.740	6.91	46.1	1143
Q75	Initial	1.00	11	56	1326
	Synthétique	1.00	11	55	1326
Maximum	Initial	1.99	29	100	9850
	Synthétique	1.63	29	98	9817

TABLE 1.5.3 – Résumé statistique des variables continues pour les datasets initial et synthétique

La figure 1.5.3 permet de comparer les didtributions des quatre variables pour le jeu de données synthétique et le jeu de données initial. A première vue, les distributions sont très ressemblantes. Pour les variables `Exposure` et `Density`, les courbes de densité sont très proches entre les deux dataset. Les pics de chacune des deux distributions sont quasiment répliqués à l'identique. Les graphiques indiquent que les distributions sont assez bien répliquées.

Pour les variables `DrivAge` et `VehAge`, les densités sont assez bien respectées dans leur forme générale. Cependant, si l'on compare avec les densités des deux autres variables, les résultats ne sont pas aussi bons sans pour autant être mauvais. Pour la variables `VehAge`, les données synthétiques suggèrent un pic en 1 alors que ce pic arrive en 2 pour les données originales. Les données synthétiques ne capturent pas très bien non plus le léger pic en 10. Pour la variable `DrivAge`, la structure générale est assez bonne. Cependant, la distribution des données synthétiques entre 30 et 60 ans ne capture pas très bien les légères variations observées dans le dataset original.

De manière générale, les différents résultats montrent que le modèle est en mesure de produire des données synthétiques qui respectent les caractéristiques du dataset initial.



(a) `VehAge` (b) `DrivAge`

(c) `Density` (d) `Exposure`

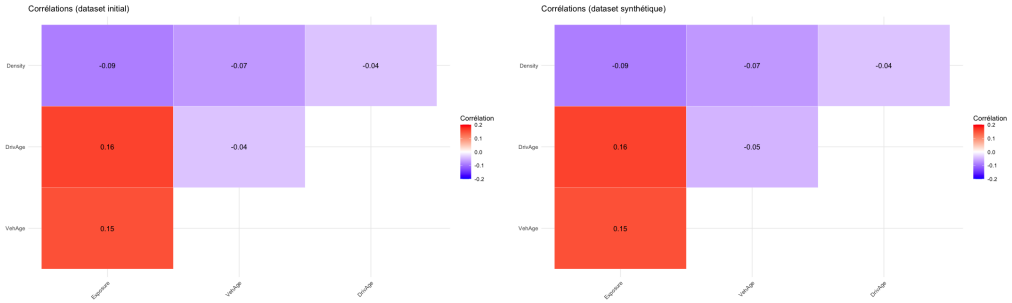
FIGURE 1.5.3 – Comparaison des densités pour chaque variable numérique entre les données initiales et les données synthétiques.

Jusqu'à présent, les analyses étaient concentrées sur les distributions univariées de chacune des quatre variables. Un point important est de vérifier que le modèle respecte aussi dépendances entre celles-ci. L'idée est donc de comparer les matrices de corrélation des deux jeux de données.

Les matrices de corrélation permettent de mesurer une partie de la dépendance entre les variables numériques des datasets.

La figure 1.5.4 contient les matrices de corrélation entre les quatre variables numériques pour les deux jeux de données. De manière générale, les coéfficients sont relativement petits ce qui indique que les variables ne montrent pas de fortes tendances linéaires entre elles. On constate que les deux matrices de corrélation sont identiques. L'unique différence est issue entre la corrélation des variables `DrivAge` et `VehAge` puisque dans le dataset original elle est de -0.04 alors que dans le dataset synthétique, elle vaut -0.05. Cette différence est cependant minime.

Globalement, la structure des dépendances entre les variables est très bien capturée par le modèle de diffusion.



(a) Matrice de corrélation pour les variales numériques du dataset initial.

(b) Matrice de corrélation pour les variales numériques du dataset synthétique.

FIGURE 1.5.4 – Comparaison des matrices de corrélation pour les deux jeux de données.

Chapitre 2

Génération de variables discrètes

La première partie du mémoire était consacrée au modèle de diffusion. Ce modèle permet, comme expliqué plus tôt, de bien générer des données synthétiques continues. L'objectif est de pouvoir également générer des données discrètes. Ces données sont présentes en multitude dans de nombreuses bases de données. L'objectif est de pouvoir générer ces variables discrètes synthétiques et que ces dernières gardent la même structure que celle présente dans la base de données initiale. Il n'est pas suffisant que la distribution des catégories soit identique aux données initiales. Il est aussi important, par exemple, que les distributions jointes entre les différentes variables soient aussi respectées. Sans ces conditions, des simulations de Monte Carlo, variable par variable, auraient juste fait l'affaire. Supposons que nous ayons une base de données avec une variable "marque du véhicule" et une variable représentant "la puissance du véhicule". Il est intuitif d'imaginer que certaines marques ont plus de modèles avec des moteurs puissants que d'autres marques. Ainsi, il faudra que les données synthétiques capturent cette structure entre les deux variables. L'objectif de ce chapitre est de trouver une méthode qui permet de générer des variables discrètes. La solution viendra de l'article [7], qui est l'article fondateur d'une méthode appelée TabDDPM. C'est une méthode qui permet de générer des bases de données contenant à la fois des variables discrètes et des variables continues. Ce chapitre se concentrera uniquement sur les variables discrètes. C'est une méthode basée sur un modèle de diffusion pour des variables avec une distribution multinomiale. Cependant, l'article [8] propose une méthode similaire à la méthode TabDDPM mais produisant des résultats plus qualitatifs. C'est une méthode qui s'appelle TabDIFF. Dans les modèles TabDDPM [7], le pas de temps est discret et est fixe puisque t est sous la forme de $t = k/T$ avec T qui est donc fixé à l'avance. Les modèles TabDIFF [8] généralisent mieux le processus puisque les modèles sont construits pour $t \in [0, 1]$. Il n'y a donc pas de discrétisation prédéfinie, il ne faut donc pas choisir à l'avance le nombre d'étapes T pour effectuer la diffusion. Cela engendre que la méthode TabDIFF produit des données de meilleure qualité. Une présentation théorique du modèle sera faite et une explication détaillée des algorithmes d'entraînement et de simulation sera réalisée. La fin du chapitre sera dédiée à l'application du modèle sur une base de données assurantielles.

2.1 Pourquoi le modèle de diffusion ne fonctionne pas avec les données discrètes ?

Dans un premier temps, il a été essayé d'appliquer le modèle de diffusion avec des données discrètes. Les variables discrètes sont *one-hot encodées*, c'est-à-dire que chaque variable discrète x^i_{cat} est transformée en n_i variables continues, où $i = 1, \ldots, n$, n_i est le nombre de catégories de la variable discrète i, et n est le nombre de variables discrètes présentes dans la base de données. Chaque nouvelle variable représente une catégorie unique et prend la valeur 1 si l'observation appartient à cette catégorie, et 0 sinon.

Exemple 2.1.1. Soit une variable catégorielle avec x avec 3 catégories : A, B et C. L'encodeur one-hot transforme cette variable en 3 variables numériques

$$x_A = \begin{cases} 1 & \text{si } x = A \\ 0 & \text{sinon} \end{cases}, \quad x_B = \begin{cases} 1 & \text{si } x = B \\ 0 & \text{sinon} \end{cases}, \quad x_C = \begin{cases} 1 & \text{si } x = C \\ 0 & \text{sinon} \end{cases}.$$

Ensuite, le modèle de diffusion est appliqué à l'ensemble de ces données. Puisque toutes les variables sont considérées comme continues, les données synthétiques générées peuvent être différentes de 0 et 1. Pour retrouver la catégorie, celle dont la colonne correspondante est la plus proche de 1 est attribuée à la donnée générée.

D'un point de vue résultat, il y a deux problèmes majeurs avec ces données synthétiques :

1. Les variables où certaines catégories sont sur-représentées ont tendance à l'être encore plus dans le cas des données générées. De plus, celles dont la proportion est très faible sont également mal gérées par le modèle, puisque cette proportion est souvent encore plus faible dans les données synthétiques.
2. Les variables discrètes avec un nombre important de catégories sont difficilement comprises par le modèle. De manière générale, la structure n'est pas souvent respectée et la structure générée semble "aléatoire". Parfois, la calibration de deux modèles (réseaux de neurones) identiques ne donne pas du tout les mêmes résultats.

Plusieurs structures de réseaux de neurones ont été essayées en jouant à la fois sur le nombre de couches du réseau de neurones et sur le nombre de neurones. Plusieurs fonctions d'activation ont été testées également. Les résultats contenaient toujours les mêmes défauts, peu importe la forme du réseau de neurones. Ensuite, plusieurs encodeurs ont été essayés pour les variables discrètes afin de voir si c'était un problème lié à l'encodeur initial utilisé. À nouveau, les résultats comportaient les mêmes soucis.

Pour rappel, le modèle de diffusion fait comme hypothèse que x_0 est distribué selon une normale $\mathcal{N}(0, 1)$ afin de pouvoir définir $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$. De cette manière, $q(x_t|x_0)$ suit également une normale. C'est sous cette hypothèse que repose la suite de la preuve pour le modèle de diffusion. Cependant, dans le cas des variables discrètes, peu importe l'encodeur utilisé et le nombre de dimensions de l'espace projeté, $q(x_0)$ sera très rarement une distribution normale.

C'est une explication qui peut être une des raisons pour lesquelles le modèle de diffusion ne fonctionne pas pour les données discrètes.

2.2 Modèle de diffusion multinomial

Cette section explicitera la méthode utilisée pour construire une méthode de génération de données synthétiques discrètes. Cette méthode se base sur l'article [4], qui introduit le modèle multinomial. C'est un modèle qui ressemble fortement, dans l'idée, au modèle de diffusion dans le cadre de variables continues.

Notation Soit x^i une variable catégorielle prenant des valeurs dans un ensemble fini de n_i catégories. On note $x^i_{\text{cat},0} \in \{0,1\}^{n_i}$ sa représentation vectorielle *one-hot encodée*, c'est-à-dire un vecteur de dimension n_i contenant un unique 1 à la position correspondant à la catégorie observée, et des zéros ailleurs.

Exemple 2.2.1. Si la variable x^i représente une couleur avec $n_i = 3$ catégories possibles : `rouge`, `vert`, `bleu`, alors :

— $x^i = \texttt{rouge} \quad \Rightarrow \quad x^i_{\text{cat},0} = (1,0,0)$
— $x^i = \texttt{vert} \quad \Rightarrow \quad x^i_{\text{cat},0} = (0,1,0)$
— $x^i = \texttt{bleu} \quad \Rightarrow \quad x^i_{\text{cat},0} = (0,0,1)$

Chaque variable discrète $x^i := x^i_{cat,0}$ est transformée via un encodeur one-hot. Le modèle de diffusion multinomial est défini comme

$$q_{cat}(x^i_{cat,t}|x^i_{cat,t-1}) = C\Big(x^i_{cat,t}|(1-\beta_t)x^i_{cat,t-1} + \frac{\beta_t}{n_i}\Big).$$

avec C qui représente la distribution d'une variable discrète avec les paramètres de la distribution après |.

Exemple 2.2.2. Si la variable x^i représente une couleur avec $n_i = 3$ catégories possibles : `rouge`, `vert`, `bleu` avec comme distribution :

$$C(x^i_{cat,0}|[0.5, 0.3, 0.2])$$

alors :

— $x^i = \texttt{rouge}$ avec probabilité 50% ;
— $x^i = \texttt{vert}$ avec probabilité 30% ;
— $x^i = \texttt{bleu}$ avec probabilité 20%.

Intuitivement, à chaque étape, un petit rééquilibrage β_t est appliqué à travers chaque catégorie pour chacune des variables discrètes. Comme c'était le cas pour le modèle de diffusion, cette distribution représente une chaîne de Markov. On peut également définir $q_{cat}(x^i_{cat,t}|x^i_{cat,0})$ par

$$q_{cat}(x^i_{cat,t}|x^i_{cat,0}) = C\left(x^i_{cat,t}\middle|\bar{\alpha}_t\, x^i_{cat,0} + \frac{(1-\bar{\alpha}_t)}{n_i}\right).$$

avec $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

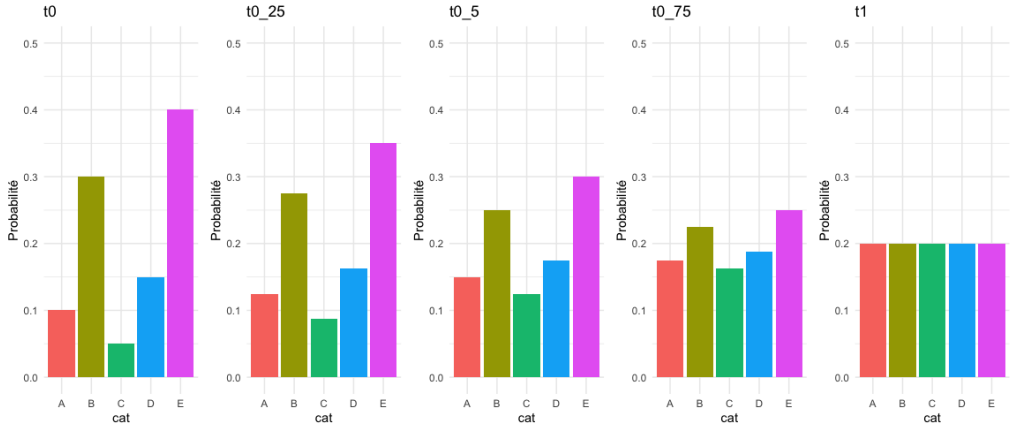


FIGURE 2.2.1 – Exemple de processus multinomial. $x_{cat,0}$ représente la distribution initiale des catégories de la variable x_{cat}. Le graphique de gauche représente la distribution de $x_{cat,0}$. Les graphiques de droite représentent un rééquilibrage entre les classes.

La formule $q_{cat}(x^i_{cat,t} \mid x^i_{cat,t-1})$ représente l'équivalent du processus *forward* dans le modèle de diffusion avec des variables continues. Il faut maintenant trouver l'équivalent du processus *backward* dans le cadre des variables discrètes. C'est-à-dire trouver un estimateur de $x_{cat,t-1}$ sachant $x_{cat,t}$, de manière à pouvoir simuler $x_{cat,T}$ facilement, puis appliquer récursivement ce processus *backward* pour obtenir $x_{cat,0}$.

Proposition 2.2.3. *Soit la variable discrète x^i_{cat} une variable discrète avec $n_i \in \mathbb{N}$ modalités. Le processus backward dans le cas des modèles de diffusion pour des variables discrètes est estimé par :*

$$q_{cat}(x^i_{cat,t-1} \mid x^i_{cat,t}, x^i_{cat,0}) = C\left(x^i_{cat,t}; \frac{\pi}{\sum_{k=1}^{K} \pi_k}\right),$$

où

$$\pi = \left[\alpha_t\, x_t + \frac{1-\alpha_t}{n_i}\right] \odot \left[\bar{\alpha}_{t-1}\, x_0 + \frac{1-\bar{\alpha}_{t-1}}{n_i}\right].$$

Démonstration. Par le théorème de Bayes, on a :

$$q_{\text{cat}}(x^i_{\text{cat},t-1} \mid x^i_{\text{cat},t}, x^i_{\text{cat},0}) = \frac{q_{\text{cat}}(x^i_{\text{cat},t} \mid x^i_{\text{cat},t-1}, x^i_{\text{cat},0}) \cdot q_{\text{cat}}(x^i_{\text{cat},t-1} \mid x^i_{\text{cat},0})}{q_{\text{cat}}(x^i_{\text{cat},t} \mid x^i_{\text{cat},0})}.$$

Puisque nous sommes dans un processus de Markov,

$$q_{\text{cat}}(x^i_{\text{cat},t} \mid x^i_{\text{cat},t-1}, x^i_{\text{cat},0}) = q_{\text{cat}}(x^i_{\text{cat},t} \mid x^i_{\text{cat},t-1}).$$

On en déduit que

$$q_{\text{cat}}(x^i_{\text{cat},t-1} \mid x^i_{\text{cat},t}, x^i_{\text{cat},0}) \propto q_{\text{cat}}(x^i_{\text{cat},t} \mid x^i_{\text{cat},t-1}) \cdot q_{\text{cat}}(x^i_{\text{cat},t-1} \mid x^i_{\text{cat},0}).$$

Puisque nous sommes dans le cadre de variables discrètes encodées en one-hot, chaque vecteur

$$x^i_{\text{cat},t-1} \in \{0,1\}^{n_i}$$

est un vecteur à une seule composante égale à 1, les autres étant nulles. Autrement dit, il existe un unique $k \in \{1,\dots,n_i\}$ tel que $x^{i,(k)}_{\text{cat},t-1} = 1$, ce qui correspond à la modalité k de la variable catégorielle i.

De plus, pour chaque composante $k \in \{1,\dots,n_i\}$, la probabilité que la composante k soit activée dans $x^i_{\text{cat},t-1}$ (i.e. $x^{i,(k)}_{\text{cat},t-1} = 1$) conditionnellement à $x^i_{\text{cat},t}$ et $x^i_{\text{cat},0}$ est proportionnelle à :

$$q_{\text{cat}}(x^{i,(k)}_{\text{cat},t-1} = 1 \mid x^i_{\text{cat},t}, x^i_{\text{cat},0}) \propto q_{\text{cat}}(x^i_{\text{cat},t} \mid x^{i,(k)}_{\text{cat},t-1} = 1) \cdot q_{\text{cat}}(x^{i,(k)}_{\text{cat},t-1} = 1 \mid x^i_{\text{cat},0}).$$

Les deux termes du produit s'écrivent alors comme suit :

— **Deuxième terme** :
Le vecteur $x^i_{\text{cat},0} \in \{0,1\}^{n_i}$ encode la modalité initiale de la variable i en one-hot. Ainsi, pour chaque $k \in \{1,\dots,n_i\}$, la composante $x^{i,(k)}_{\text{cat},0} \in \{0,1\}$ indique si la modalité initiale est k. On a donc :
$$q_{\text{cat}}(x^{i,(k)}_{\text{cat},t-1} = 1 \mid x^i_{\text{cat},0}) = \bar{\alpha}_{t-1} \cdot x^{i,(k)}_{\text{cat},0} + \frac{1-\bar{\alpha}_{t-1}}{n_i}.$$

— **Premier terme** :
De même, si $x^i_{\text{cat},t}$ est observé et vaut un vecteur one-hot de modalité l, alors $x^{i,(l)}_{\text{cat},t} = 1$ et les autres composantes sont nulles. Pour chaque $k \in \{1,\dots,n_i\}$, on a :

$$q_{\text{cat}}(x^{i,(l)}_{\text{cat},t} = 1 \mid x^{i,(k)}_{\text{cat},t-1} = 1) = \begin{cases} 1 - \beta_t + \dfrac{\beta_t}{n_i}, & \text{si } k = l, \\ \dfrac{\beta_t}{n_i}, & \text{sinon.} \end{cases}$$

Ainsi, pour chaque composante $k \in \{1,\dots,n_i\}$, la probabilité associée à $x^{i,(k)}_{\text{cat},t-1} = 1$ conditionnellement à $x^i_{\text{cat},t}$ et $x^i_{\text{cat},0}$ est proportionnelle à :

$$\gamma_k \propto \left[(1-\beta_t)\delta_{k,l} + \frac{\beta_t}{n_i}\right] \cdot \left[\bar{\alpha}_{t-1} \cdot x_{\text{cat},0}^{i,(k)} + \frac{1-\bar{\alpha}_{t-1}}{n_i}\right]$$

avec $\delta_{k,l} = 1$ si $k = l$, 0 sinon.

Ce qui est équivalent à dire que

$$\gamma \propto \left[(1-\beta_t)x_{cat,t}^{i} + \frac{\beta_t}{n_i}\right] \odot \left[\bar{\alpha}_{t-1} \cdot x_{\text{cat},0}^{i} + \frac{1-\bar{\alpha}_{t-1}}{n_i}\right].$$

En introduisant un facteur de normalisation Z, on obtient :

$$q(x_{\text{cat},t-1}^{i} \mid x_{\text{cat},t}^{i}, x_{\text{cat},0}^{i}) = \text{Cat}(x_{\text{cat},t-1}^{i}; \gamma),$$

avec :

$$\gamma_k = \frac{1}{Z}\left[(1-\beta_t)\delta_{k,l} + \frac{\beta_t}{n_i}\right] \cdot \left[\bar{\alpha}_{t-1} \cdot x_{\text{cat},0}^{i,(k)} + \frac{1-\bar{\alpha}_{t-1}}{n_i}\right],$$

et :

$$Z = \sum_{k=1}^{n_i}\left[(1-\beta_t)\delta_{k,l} + \frac{\beta_t}{n_i}\right] \cdot \left[\bar{\alpha}_{t-1} \cdot x_{\text{cat},0}^{i,(k)} + \frac{1-\bar{\alpha}_{t-1}}{n_i}\right].$$

□

Pour rappel, l'objectif est de minimiser :

$$\log L \geqslant \mathbb{E}\Big[-D_{KL}[q(x_T \mid x_0) \parallel p_\theta(x_T)] - \sum_{t=2}^{T} D_{KL}[q(x_{t-1} \mid x_0, x_t) \parallel p_\theta(x_{t-1} \mid x_t)] + \log p_\theta(x_0 \mid x_1)\Big].$$

Cette inégalité est identique à celle du cas des variables continues, puisque la démonstration ne dépend pas des distributions a priori.

Le processus *reverse* $p_\theta(x_{t-1}^{i} \mid x_t^{i})$ est paramétré comme $q_{cat}(x_{cat,t-1}^{i} \mid x_{cat,t}^{i}, \widehat{x}_{cat,0}^{i}(\mathbf{x_{cat,t}}, t))$, avec $\widehat{x}_{cat,0}^{i}(\mathbf{x_{cat,t}}, t)$ prédit par un réseau de neurones et où $\mathbf{x_{cat,t}} = [x_{cat,t}^{1}, \cdots, x_{cat,t}^{k}]$.

Définition 2.2.4. Pour deux distributions de probabilité discrètes $\mathcal{P}$ et Q définies sur un ensemble X, la divergence de Kullback-Leibler de $\mathcal{P}$ par rapport à Q est définie par :

$$D_{KL}(P \parallel Q) = \sum_{x \in X} P(x) \ln \frac{P(x)}{Q(x)}.$$

Exemple 2.2.5. Soit X une variable discrète pouvant prendre 3 catégories (A, B et C), et soient $\mathcal{P}$ et Q deux distributions de probabilité discrètes sur X :

Distribution	**A**	**B**	**C**
$\mathcal{P}(X)$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
$\mathcal{Q}(X)$	$\frac{1}{5}$	$\frac{3}{10}$	$\frac{1}{2}$

TABLE 2.2.1 – Exemple de distributions de probabilité discrètes

Pour cet exemple :

$$\begin{aligned} D_{KL}(P \parallel Q) &= P(A) \ln \frac{P(A)}{Q(A)} + P(B) \ln \frac{P(B)}{Q(B)} + P(C) \ln \frac{P(C)}{Q(C)} \\ &= \frac{1}{3} \ln \frac{1/3}{1/5} + \frac{1}{3} \ln \frac{1/3}{3/10} + \frac{1}{3} \ln \frac{1/3}{1/2} \\ &= 0.0702. \end{aligned}$$

Dans le cas des variables catégorielles, le calcul de la fonction de perte est direct. À partir des probabilités de chacune des catégories, calculer la divergence de Kullback-Leibler est aisé et le calcul est immédiat.

2.2.1 Processus d'entraînement

Cette section est consacrée à la description du processus d'entraînement du modèle pour la génération de données catégorielles.

Pour calibrer le réseau de neurones, le processus est relativement similaire au cas des variables continues.

1. Chaque ligne x_0 du dataset à partir duquel le modèle est construit correspond à la distribution empirique des données, c'est-à-dire à la distribution non bruitée.
2. Pour chaque ligne x_0, on simule t selon une distribution uniforme $Unif(0, 1)$. Cela va représenter l'instant dans le processus de bruitage pour l'itération. Le paramètre t est également encodé via un *embeddings*.
3. À partir de x_0 et t et pour chacune des k variables discrètes, on simule x_t selon la distribution $q_{cat}(x^i_{cat,t}|x^i_{cat,0})$, qui représente la version bruitée de la variable ligne x^i_0 à l'instant t.
4. Le modèle a pour but de pouvoir approximer la distribution de $q(x^i_{t-1} \mid x^i_t, x^i_0)$. Pour cet objectif, il cherchera à prédire $\widehat{x}^i_{\text{cat},0}(\mathbf{x_{cat,t}}, t)$, qui est une estimation de $x^i_{\text{cat},0}$. La dernière couche du réseau de neurones utilisera une fonction d'activation SoftMax.

   Définition 2.2.6. Soit $x \in \mathbb{R}^k$, la fonction softmax f est définie comme :

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}.$$

5. Les paramètres du modèle sont ajustés par descente de gradient pour minimiser la divergence de Kullback-Leibler suivante :

$$D_{\mathrm{KL}}\left[q(x_{t-1} \mid x_0, x_t) \,\|\, q_{\mathrm{cat}}\left(x^i_{\mathrm{cat},t-1} \mid x^i_{\mathrm{cat},t}, \widehat{x}^i_{\mathrm{cat},0}(\mathbf{x_{cat,t}}, t)\right)\right]$$

6. Ce processus est répété un grand nombre de fois pour pouvoir parcourir un grand nombre de fois les étapes 1 et 2 et ainsi obtenir la convergence.

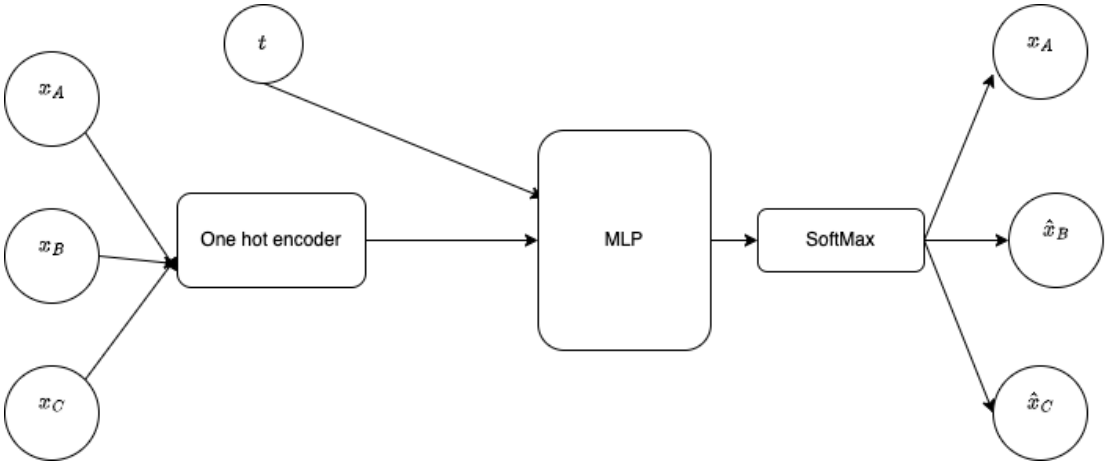


FIGURE 2.2.2 – Forme du réseau de neurones pour la génération de données catégorielles.

Algorithm 3 L'algorithme pour l'entraînement du modèle pour le processus *reverse*.

Répéter
$x_0 \sim q(x_0)$ ▷ Tirer au hasard une ligne du dataset
$t \sim \mathrm{Uniform}(0, 1)$ ▷ Simuler un instant t
$\mathbf{x_{cat,t}} \sim q_{cat}(\mathbf{x_{cat,t}} \mid \mathbf{x_{cat,0}})$ ▷ Simuler $\mathbf{x_{cat,t}}$
for $i = 1$ **to** K **do**
Ajuster les paramètres avec la descente de gradient pour minimiser

$$D_{KL}\left[q(x^i_{cat,t-1} \mid x^i_{cat,0}, x^i_{cat,t}) \,\|\, q_{cat}(x^i_{cat,t-1} \mid x^i_{cat,t}, \widehat{x}^i_{cat,0}(\mathbf{x_{cat,t}}, t))\right]$$

end for
Convergence

2.2.2 Processus de génération

Le processus de génération est décrit ci dessous.

1. Pour chaque variable catégorielle x^i avec n_i catégories, une simulation à partir d'une

distribution uniforme est effectuée

$$x^i_{\text{cat},T} \sim C\left(\frac{1}{n_i}, \ldots, \frac{1}{n_i}\right)$$

où T est le nombre d'étapes dans le processus de diffusion. A l'étape T, l'incertitude est maximale c'est-à-dire que chaque catégorie a la même probabilité d'être tirée au sort

2. Pour chaque t de T à 1,
   — Prédire l'état initial $\widehat{x}^i_{\text{cat},0}$ à l'aide du réseau de neurones qui aura pris en entrée le temps t et la catégorie x_t issue de la distribution bruitée au temps t.
   — Simuler l'état précédent $x^i_{\text{cat},t-1}$ à partir de la distribution

$$x^i_{\text{cat},t-1} \sim q_{\text{cat}}\left(x^i_{\text{cat},t-1} \mid x^i_{\text{cat},t}, \widehat{x}^i_{\text{cat},0}\right)$$

3. Obtenir pour chaque variable x_i, l'état final $x^i_{\text{cat},0}$ qui aura été généré à partir de la simulation initiale.

Algorithm 4 Algorithme de génération de variables discrètes par la méthode de diffusion multinomial

Initialisation :
for chaque variable catégorielle x^i à n_i catégories **do**
  $x^i_{\text{cat},T} \sim C\left(\frac{1}{n_i}, \ldots, \frac{1}{n_i}\right)$ ▷ Distribution uniforme
end for
for $t = \frac{T}{T}, \frac{T-1}{T}, \ldots, \frac{1}{T}$ **do**
  $\widehat{x}^i_{\text{cat},0} \leftarrow \widehat{x}^i_{\text{cat},0}(\mathbf{x}_{\mathbf{cat},\mathbf{t}}, t)$ ▷ Prédiction de la valeur originale
  $x^i_{\text{cat},t-1} \sim q_{\text{cat}}\left(x^i_{\text{cat},t-1} \mid x^i_{\text{cat},t}, \widehat{x}^i_{\text{cat},0}\right)$ ▷ Échantillonnage via la transition inverse
end for
return $x^i_{\text{cat},0}$

2.3 Application de la méthode et analyse des données générées

Au cours de cette section, une application de la méthode sera réalisée sur des données discrètes. L'objectif de cette section est de montrer que cette méthode fonctionne. C'est-à-dire démontrer qu'elle génère un nouveau dataset en tenant compte des caractéristiques statistiques de la base de données initiale. Il faudra d'abord vérifier que le dataset synthétique n'est pas une simple réplication du dataset original. On vérifiera par la suite si les distributions univariées sont bien modélisées par la méthode. Ensuite, on analysera quelques exemples de distributions bivariées et tri-variées. Cette analyse a pour but de vérifier si le modèle capture bien les dépendances présentes entre les différentes variables.

2.3.1 Analyse descriptive

La base de données utilisée pour l'analyse de cette partie du mémoire est la même que celle utilisée pour le modèle de diffusion avec des données continues. Cette fois-ci, uniquement les variables discrètes seront utilisées. La base de données contient six variables catégorielles.

- **Le nombre de sinistres :** Le nombre de sinistres subis par l'assuré durant sa période dans le portefeuille. C'est un nombre entier variant entre 0 et 3.
- **La zone géographique :** C'est une variable catégorielle avec 6 catégories (A, B, C, D, E et F) indiquant la zone géographique de l'assuré. La zone A étant la moins dense et la zone F la plus dense.
- **La puissance du véhicule :** Cette variable représente la puissance du véhicule. La puissance est représentée par un chiffre entier variant entre 6 et 15. La catégorie 6 correspond aux véhicules les moins puissants, et à l'inverse, les véhicules les plus puissants ont une puissance de 15.
- **La marque du véhicule :** La marque du véhicule est représentée par un code à 2 ou 3 caractères. La base de données contient des voitures de 12 marques différentes.
- **Le carburant :** Indique si le véhicule est un véhicule essence ou diesel.
- **La région de l'assuré :** La région dans laquelle l'assuré est domicilié. Il y a 22 régions différentes, désignées par 3 caractères.

La figure 2.3.2 permet de visualiser la proportion des catégories pour chacune des variables discrètes.

- **Le nombre de sinistres :** La grande majorité des assurés (94%) n'ont déclaré aucun sinistre. Les sinistres multiples sont très rares, représentant moins de 1% des cas.
- **La zone géographique :** La majorité des assurés vivent dans les zones C et B, représentant à elles seules plus de 50% des individus. La zone F est très peu représentée.
- **La puissance du véhicule :** Les véhicules à faible puissance sont majoritaires. La proportion décroît pour les véhicules plus puissants.
- **La marque du véhicule :** Trois marques (B1, B2 et B12) dominent le portefeuille, chacune représentant environ 25% du total, tandis que les autres marques sont très peu représentées.
- **Le carburant :** La répartition est relativement équilibrée entre Diesel (48%) et Essence (52%), avec une légère majorité pour les véhicules essence.
- **La région de l'assuré :** La distribution des assurés est la plus concentrée dans trois grandes régions (R24 : 20%, R82 : 13%, R93 : 10%), tandis que de nombreuses autres régions ont des parts très faibles (< 10%).

Il y a donc six variables discrètes différentes. C'est intéressant, puisque ces variables présentent des formes très différentes les unes des autres. Certaines variables (carburant et zone géographique) ont des catégories relativement bien équidistribuées, c'est-à-dire qu'aucune catégorie n'est fortement sous-représentée ou sur-représentée. La variable *Nombre de sinistres*, quant à

(a) Le nombre de sinistres

(b) Le carburant du véhicule

(c) La marque du véhicule

(d) La puissance du véhicule

(e) La zone géographique

(f) La région géographique

FIGURE 2.3.1 – Répartition des catégories pour chacune des variables discrètes

elle, est distribuée de manière très déséquilibrée puisque 94% des assurés ne déclarent aucun sinistre. La base de données possède également des variables avec un grand nombre de modalités, comme c'est le cas pour la variable *région géographique*.

En résumé, la base de données contient des variables catégorielles, chacune présentant des propriétés statistiques distinctes. Il sera intéressant d'analyser le résultat du modèle pour ces

différentes variables. Est-ce que la distribution des variables avec beaucoup de catégories sera bien générée ? Les catégories relativement peu représentées seront-elles bien générées par le modèle ?

2.3.2 Entraînement du modèle

Cette section sera consacrée à la description du réseau de neurones utilisé pour la génération de données catégorielles. Pour rappel, l'input du réseau de neurones dans le cadre des variables catégorielles est le temps t définissant le moment de la diffusion du processus, ainsi que la simulation $x_{cat,t}$ issue de la distribution bruitée au moment t.

Puisque la variable *ClaimNb* est considérée comme étant une variable catégorielle, la calibration du modèle se limitera aux lignes où l'exposition est égale à 1. En effet, puisque c'est une variable importante pour un assureur et que nous voudrions tester si le modèle capture bien l'interaction entre le nombre de sinistres et les autres variables, il est important de comparer les choses de manière équitable. Intuitivement, les assurés qui restent un seul mois dans le portefeuille ont moins de chances de subir un sinistre qu'un assuré qui y reste deux ans, et ce, peu importe les caractéristiques de ce dernier. En se restreignant aux assurés avec une exposition d'exactement un an, nous travaillerons avec un jeu de données contenant 165 066 lignes. 80% des données formeront les données pour l'entraînement du modèle, et les 20% restantes seront les données qui formeront le *testing set*.

La taille de l'*embedding* donnant au réseau la connaissance de l'instant t du processus de diffusion est de 16. La forme du réseau de neurones utilisée est un réseau de neurones avec 4 couches entièrement connectées, avec comme fonction d'activation la fonction ReLU. Le réseau est constitué de six parties parallèles, possédant exactement le même input. Chacune des six parties sera focalisée sur une variable catégorielle. Chaque partie est identique et contient trois couches (avec un nombre de neurones décroissant), entièrement connectées avec comme fonction d'activation la fonction ReLU (256, 128, 64 neurones). La dernière couche utilisera une fonction d'activation Softmax afin de prédire la probabilité des catégories pour la variable liée à la partie du réseau en question. Évidemment, les dimensions de sortie vont varier en fonction du nombre de catégories différentes pour la variable liée à la branche du réseau de neurones concernée.

Le taux de diffusion sera cosinusoïdal. La taille du lot (*batch size*) est fixée à 516, tandis que le taux d'apprentissage (*learning rate*) utilisé est de 1.5×10^{-5}. L'entraînement sera effectué sur un total de 1 000 itérations (*epochs*). Le temps de calcul est relativement important puisque le réseau de neurones contient 361 079 paramètres.

2.3.3 Génération des données synthétiques

Comme ce fut le cas pour les variables continues, cette section sera consacrée à l'analyse des résultats produits. Il faudra faire en sorte que les données produites conservent les mêmes propriétés statistiques que le dataset initial. Par exemple, dans le cadre du dataset, il est fort probable que la variable `VehPower` soit fortement dépendante de la variable `VehBrand`. Intuitivement, certaines

Paramètre	**Valeur**
Taille du dataset utilisé	165 066 lignes (*Exposure* = 1)
Partage entraînement / test	80% entraînement, 20% test
Taille de l'*embedding*	16
Structure du réseau	6 branches parallèles identiques
Couches par branche	$256 \rightarrow 128 \rightarrow 64$
Fonction d'activation des couches intermédiaires	ReLU
Taille du batch	516
Taux d'apprentissage	1.5×10^{-5}
Nombre d'epochs	1 000
Forme du taux de diffusion	Cosinusoïdal

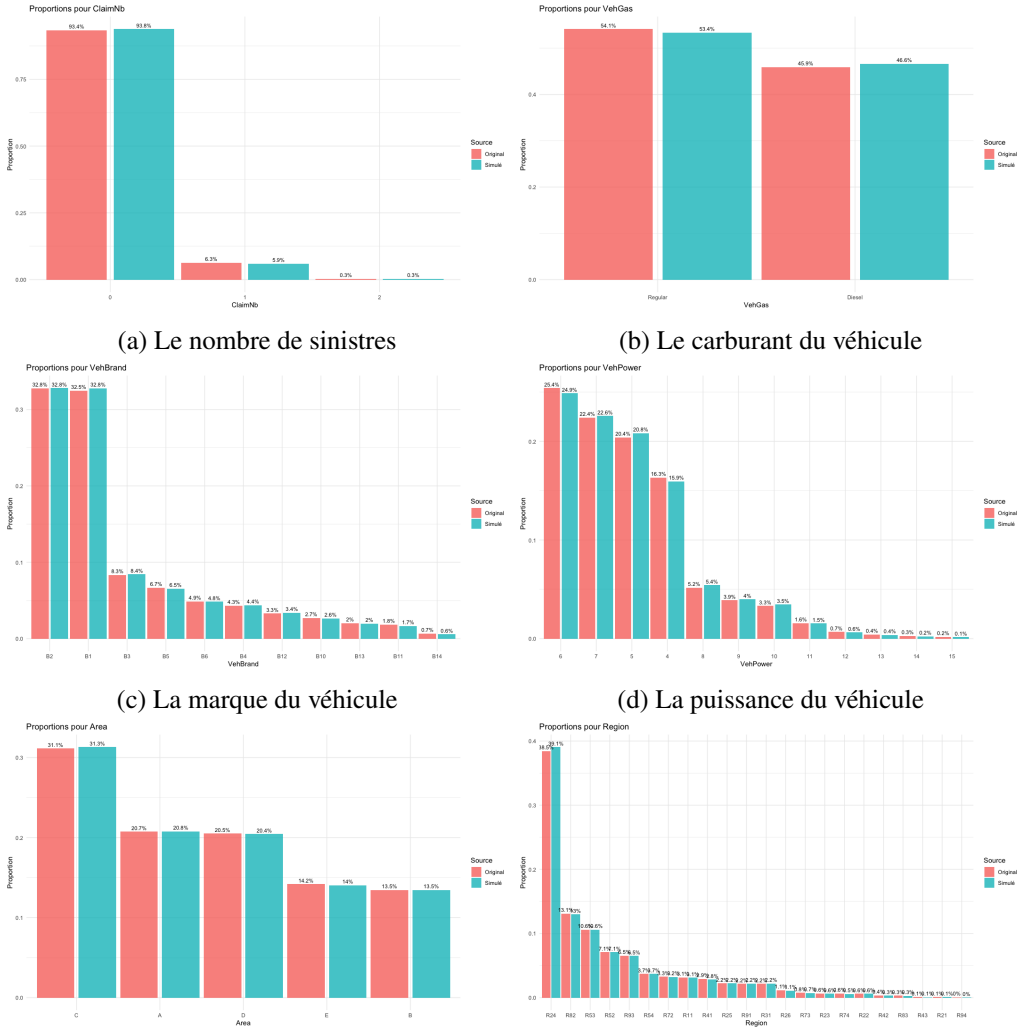
TABLE 2.3.1 – Résumé des paramètres du réseau de neurones pour la génération des données catégorielles

marques de véhicules produisent plus de véhicules sportifs que d'autres, et inversement. La zone géographique et la région sont aussi des variables où l'on s'attend à de l'interaction. Il faudra donc vérifier que le comportement de ces distributions bivariées soit identique dans les deux jeux de données.

On génère 100 000 données en utilisant un pas de temps égal à 1/1000. Parmi les 100 000 données générées, on constate que 95 222 données sont présentes dans le dataset initial. Cela représente quand même 95% du jeu de données synthétique. Cette proportion non négligeable est liée au faible nombre de variables discrètes (6) et au grand nombre d'observations que contient le dataset initial. Puisque le dataset contient uniquement 6 variables catégorielles avec 3, 5, 2, 11, 12 et 23 catégories, le nombre maximum de lignes distinctes possibles est de 91 080. Si l'on supprime la variable `ClaimNb`, étant donné que près de 95% des assurés ont 0 sinistre, ce nombre passe à 30 360. Dans le dataset initial, on observe près de 13 000 lignes distinctes. Il est donc difficile ici de juger l'hétérogénéité des données synthétiques par rapport au dataset initial.

Comparaison univariée

La table 2.3.2 compare la répartition des catégories pour chacune des 6 variables discrètes. Le nombre de sinistres est très bien représenté, les distributions sont quasiment identiques dans les deux jeux de données. Le dataset synthétique génère légèrement moins de sinistres que le dataset initial (une différence de 0,4%). Le carburant du véhicule est également bien distribué dans le dataset synthétique. On remarque une légère sous-estimation de la proportion des véhicules avec un moteur essence et donc logiquement, une légère sur-estimation des moteurs diesel. La distribution de la marque du véhicule est quasiment parfaite, c'est relativement impressionnant au vu du grand nombre de catégories présentes au sein de cette variable. La puissance du véhicule est bien distribuée dans le dataset final. Cependant, on observe de légères différences dans la répartition des catégories, même si celles-ci sont minimes. On observe une réplication quasiment parfaite de



(a) Le nombre de sinistres

(b) Le carburant du véhicule

(c) La marque du véhicule

(d) La puissance du véhicule

(e) La zone géographique

(f) La région géographique

FIGURE 2.3.2 – Répartition des catégories pour chacunes des variables discrètes

la variable représentant la zone géographique. Le grand nombre de catégories définissant la région géographique est distribué de manière presque identique entre le jeu de données synthétiques et le jeu de données initial. Ces résultats montrent que le modèle arrive à bien capturer les effets univariés des différentes variables discrètes.

Distribution jointe `VehBrand` - `VehPower`

Maintenant, nous allons regarder les distributions bivariées des variables discrètes. Pour commencer, focalisons-nous sur les variables `VehBrand` et `VehPower`. La figure 2.3.2 fournit une

comparaison de la distribution de la puissance du véhicule par la marque de l'auto pour les données synthétiques et les données originales. On constate sur le premier tableau de la figure 2.3.2 que les variables `VehPower` et `VehBrand` sont des variables dépendantes l'une de l'autre. On observe que certaines marques de véhicules ont tendance à produire des véhicules avec un moteur moins puissant, comme par exemple les marques B14, B4 ou B6. Des marques comme B10 ou B12 sont des marques avec des voitures plus sportives, puisque les véhicules de ces marques ont des moteurs plus puissants.

(a) Distribution dans le dataset initial

	4	5	6	7	8	9	10	11	12	13	14	15
B1	16%	17%	32%	24%	4%	3%	1%	2%	0%	0%	0%	0%
B10	0%	2%	4%	24%	16%	16%	15%	6%	10%	4%	2%	1%
B11	2%	1%	1%	45%	10%	8%	7%	9%	6%	8%	2%	1%
B12	6%	5%	21%	19%	10%	10%	18%	3%	5%	0%	1%	2%
B13	6%	8%	24%	17%	14%	12%	11%	4%	1%	1%	1%	0%
B14	28%	9%	11%	8%	14%	16%	7%	4%	0%	1%	2%	0%
B2	19%	23%	25%	23%	4%	3%	2%	1%	0%	0%	0%	0%
B3	9%	30%	30%	16%	5%	4%	3%	1%	1%	0%	1%	0%
B4	25%	30%	15%	20%	4%	2%	2%	1%	0%	0%	0%	0%
B5	12%	37%	19%	21%	3%	2%	5%	1%	0%	0%	0%	0%
B6	32%	16%	20%	17%	9%	3%	1%	1%	0%	0%	0%	0%

(b) Différences de proportions (synthétique - initial)

	4	5	6	7	8	9	10	11	12	13	14	15
B1	+1%	+1%	-1%	0%	+1%	0%	+1%	0%	0%	0%	0%	0%
B10	+1%	+2%	+1%	+1%	0%	-1%	0%	-2%	-1%	0%	0%	0%
B11	+1%	0%	0%	+1%	-1%	+1%	0%	-1%	-1%	-1%	0%	0%
B12	+1%	+1%	+1%	+1%	0%	-1%	0%	0%	-1%	0%	-1%	-1%
B13	+1%	0%	+2%	0%	-1%	0%	-1%	0%	0%	0%	-1%	0%
B14	+1%	+2%	+1%	+1%	+1%	-4%	0%	-1%	0%	0%	0%	0%
B2	-1%	0%	-1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
B3	0%	0%	-1%	+1%	0%	0%	0%	0%	0%	0%	-1%	0%
B4	-1%	-1%	0%	0%	+1%	0%	+1%	0%	0%	0%	0%	0%
B5	0%	-2%	+1%	+1%	0%	0%	+1%	0%	0%	0%	0%	0%
B6	-1%	+1%	0%	0%	0%	0%	+1%	0%	0%	0%	0%	0%

TABLE 2.3.2 – Distribution des puissances par marque et comparaison avec le dataset synthétique

On constate à l'aide du deuxième tableau sur la figure 2.3.2 que la distribution de la puissance du véhicule par la marque est assez bien respectée. Les écarts sont vraiment petits entre les deux jeux

de données ($\pm 1\%$). Les différences les plus importantes surviennent pour les marques B10 et B14, où le modèle semble avoir légèrement plus de mal à capturer la distribution de la puissance pour ces véhicules. Ces erreurs peuvent être expliquées en partie par le faible nombre de véhicules B10 et B14 dans le dataset original, ce qui rend plus difficile l'apprentissage de ces interactions par le modèle. Pour certaines marques comme B2, B4 ou B6, les écarts sont presque nuls ou très faibles, ce qui indique que la distribution de la puissance par la marque du véhicule est très bien capturée par le réseau de neurones.

Distribution jointe `Area` - `Region`

La table 2.3.3 contient la comparaison de la répartition des zones géographiques au sein de chaque région entre le dataset original et le dataset synthétique. Le premier tableau montre la distribution observée dans les données réelles, tandis que le second indique les écarts en pourcentage entre les proportions dans le dataset synthétique et celles observées initialement.

De manière générale, on observe que les écarts restent minimes, souvent compris entre -4% et $+4\%$. Ce sont de petits écarts, mais ils sont plus importants que lors de la comparaison de la distribution de la puissance du véhicule par la marque. Cela indique que le modèle capture les grandes dépendances entre les zones géographiques et les régions. Par exemple, les répartitions pour les régions R24, R53 ou R52 montrent des différences inférieures à 2% pour toutes les zones géographiques.

Cependant, certaines régions présentent des divergences plus prononcées. On peut citer R43 où la zone D est sous-estimée de 12%, le modèle ayant tendance à attribuer davantage de zones B, C et E. De même, R21 montre une surestimation de la zone A (+9%) et une sous-estimation des zones C (-6%) et D (-5%). Ce sont cependant deux régions qui représentent moins de 2% du dataset original.

En conclusion, les grandes tendances sont globalement bien respectées, ce qui montre que le modèle est capable de capturer des interactions.

Distribution jointe `VehBrand` - `VehPower` - `VehGas`

La table .1.1 (en annexe) représente, pour chaque niveau de puissance du véhicule, la répartition de la marque et du type de carburant. Ainsi, la somme de chaque colonne est égale à 100%. Cette figure permet donc d'analyser comment chaque puissance de voiture se répartit entre les différentes marques et types de carburants. Ici, on ajoute une dimension supplémentaire par rapport aux deux tableaux précédents. En effet, on va analyser le comportement de trois variables entre elles. L'objectif est de pouvoir montrer que le modèle arrive à capturer ces effets entre trois variables.

On remarque par exemple que les véhicules avec une puissance de 6 ou 7 sont majoritairement des véhicules des marques B1, B2 et B5. De plus, ces puissances de véhicules sont souvent associées à des moteurs diesel. À l'inverse, les puissances plus élevées montrent une plus grande

(a) Données originales

Région	A	B	C	D	E
R11	3%	4%	16%	29%	47%
R21	41%	9%	26%	18%	6%
R22	13%	21%	16%	21%	29%
R23	16%	19%	38%	15%	11%
R24	38%	17%	28%	13%	4%
R25	16%	17%	34%	19%	14%
R26	34%	12%	44%	4%	7%
R31	3%	9%	23%	47%	17%
R41	2%	5%	30%	61%	3%
R42	2%	4%	27%	21%	46%
R43	32%	16%	26%	22%	3%
R52	10%	18%	30%	25%	16%
R53	7%	17%	50%	19%	7%
R54	23%	18%	30%	27%	2%
R72	19%	12%	32%	15%	21%
R73	28%	14%	30%	11%	16%
R74	54%	4%	36%	6%	0%
R82	6%	7%	30%	27%	31%
R83	46%	12%	22%	8%	11%
R91	8%	8%	47%	25%	12%
R93	4%	4%	23%	25%	44%
R94	41%	13%	9%	37%	0%

(b) Différences simulé - original

Région	A	B	C	D	E
R11	+1%	+1%	+2%	+1%	-4%
R21	+9%	+1%	-6%	-5%	+1%
R22	+4%	-2%	+4%	-2%	-3%
R23	+3%	-3%	-1%	+3%	0%
R24	-1%	-1%	+1%	0%	0%
R25	0%	-2%	+1%	0%	+1%
R26	-1%	-1%	-1%	+3%	0%
R31	+3%	0%	+1%	-5%	+1%
R41	+2%	+1%	0%	-5%	+1%
R42	+2%	+3%	+3%	-1%	-7%
R43	-4%	+4%	+4%	-12%	+9%
R52	0%	0%	+1%	0%	-1%
R53	+1%	0%	-1%	0%	+1%
R54	-1%	0%	0%	-1%	+2%
R72	0%	+1%	-1%	+1%	0%
R73	0%	-1%	-2%	+3%	+1%
R74	-5%	+2%	+2%	+1%	+1%
R82	0%	0%	0%	-1%	-1%
R83	-4%	+1%	+4%	+1%	0%
R91	+1%	+2%	-2%	0%	0%
R93	+1%	+1%	0%	0%	-3%
R94	-4%	+6%	-2%	0%	0%

TABLE 2.3.3 – Comparaison des distributions par région et zone (original vs. synthétique)

proportion de véhicules essence, en particulier pour les marques B10, B11 et B12, ce qui peut laisser sous-entendre que ce sont des marques produisant des voitures plus sportives.

La table .1.2 analyse les différences entre les données initiales et les données synthétiques produites par le modèle. Elle montre que le modèle synthétique reproduit globalement bien la structure entre les trois variables, avec des écarts souvent inférieurs à ±1%, bien que certains cas plus rares présentent des écarts plus marqués. Cela montre que le modèle est capable d'apprendre non seulement les distributions marginales, mais aussi des structures conditionnelles plus complexes, comme les interactions entre puissance, marque et carburant.

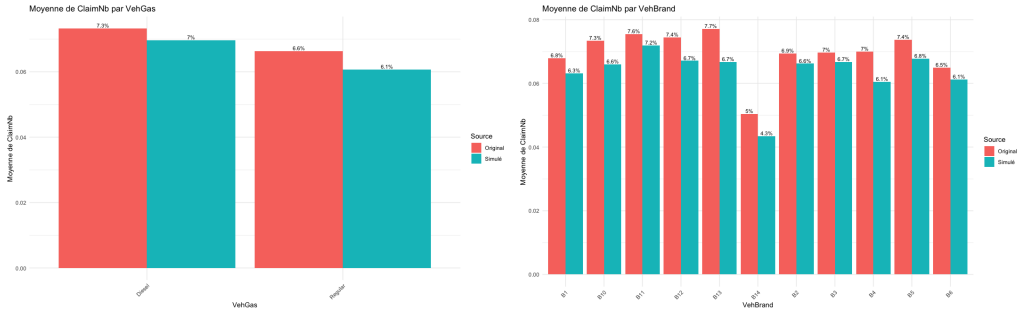
Fréquence sinistre

La figure 2.3.3 analyse la fréquence de sinistre, c'est-à-dire le nombre de sinistres par contrat. La fréquence de sinistre pour un assuré correspond à son nombre de sinistres pondéré par le temps durant lequel il est resté dans le portefeuille (l'Exposition). Puisque le modèle s'est basé sur le

portefeuille limité aux lignes où l'exposition est égale à 1, cela revient simplement à prendre la moyenne du nombre de sinistres par individu. Ainsi, il est possible d'obtenir, pour chaque modalité de chaque variable, la fréquence de sinistre comme illustré dans la figure 2.3.3.

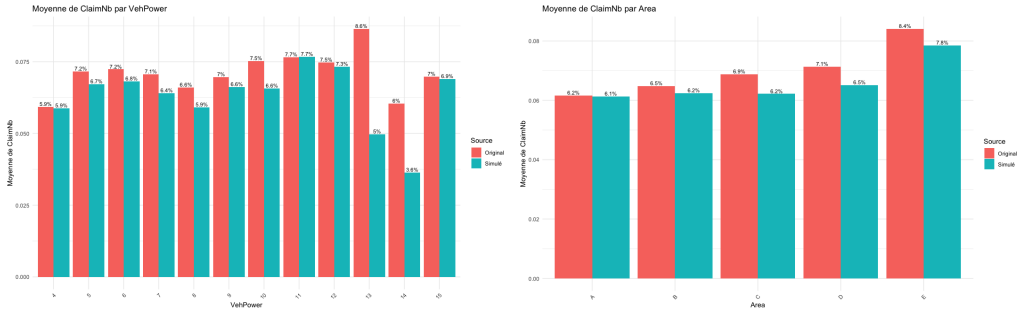
Contrairement aux analyses précédentes, les résultats sont moins nets. De manière générale, on constate, pour chaque variable et chaque modalité, un écart entre la fréquence de sinistre des données simulées et celle des données initiales. Les histogrammes bleus sont souvent inférieurs aux histogrammes rouges, ce qui indique que le modèle a tendance à sous-estimer la fréquence de sinistre de manière générale. Cela est relativement logique, puisque la figure 2.3.2 montre que les données synthétiques contiennent légèrement plus d'assurés (en proportion) ne subissant aucun sinistre.

Cependant, la figure 2.3.3 montre que le modèle est capable de rétablir un ordre et de capturer les tendances. Par exemple, pour le type de moteur, le modèle parvient à capter que les véhicules diesel ont une fréquence de sinistre légèrement supérieure à celle des véhicules essence. Pour la puissance du véhicule, la forme générale du graphique en barres est assez bien respectée. Les niveaux de sinistres plus élevés pour certaines tranches de puissance sont bien reproduits, même si les valeurs exactes varient. Le modèle semble capter la structure « hiérarchique » des risques, même s'il ne retrouve pas avec précision la fréquence de sinistre du risque pour chaque modalité.



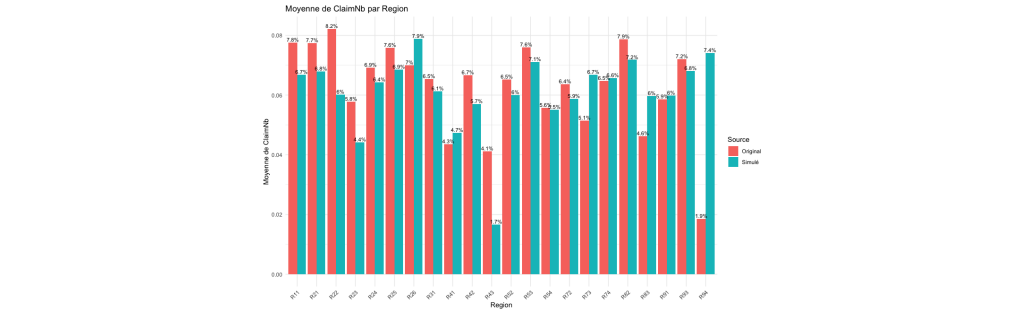
(a) Le carburant du véhicule

(b) La marque du véhicule



(c) La puissance du véhicule

(d) La zone géographique



(e) La région géographique

FIGURE 2.3.3 – Fréquence sinistre pour chaque modalité pour chaque variable

Chapitre 3

Génération d'une base de données avec des données discrètes et des données continues.

La première section de ce mémoire était consacrée à la génération de données synthétiques dans le cadre de variables continues. Ce modèle était inspiré des modèles de diffusion, qui sont fortement utilisés dans le processus de génération d'images synthétiques. La deuxième section de ce mémoire se focalisait sur la génération de données, mais uniquement dans le cadre de données discrètes. Cette dernière section va regrouper les deux processus ensemble. Elle sera consacrée à la génération de données tabulaires contenant à la fois des variables continues et des variables discrètes.

Les modèles qui englobent les modèles de diffusion et les modèles multinomiaux sont appelés des modèles TabDDPM ou TabDIFF. Dans les modèles TabDDPM [7], le pas de temps est discret et est fourni au réseau puisque t est sous la forme $t = k/T$, avec T fixé à l'avance. Les modèles TabDIFF [8] généralisent mieux le processus puisque les modèles sont construits pour $t \in [0, 1]$. Il n'y a donc pas de discrétisation prédéfinie, et il n'est pas nécessaire de choisir à l'avance le nombre d'étapes T pour effectuer la diffusion. Cela implique que le second modèle produit des données de meilleure qualité.

Techniquement, cette section n'apportera pas de grandes avancées par rapport aux deux sections précédentes. Cependant, elle ira plus en profondeur dans l'analyse des résultats et leur interprétation. Le but étant de comprendre au mieux comment le modèle parvient à capter toutes les propriétés statistiques contenues dans la base de données initiale. La section analysera de manière descriptive les résultats dans un premier temps. Ensuite, des modèles plus complexes, comme des modèles linéaires généralisés (GLM) et des modèles de gradient boosting (GBM), seront utilisés pour analyser les résultats. De manière informelle : est-ce que le modèle construit sur les données synthétiques est identique au modèle construit sur les données initiales ?

Prenons l'exemple d'un assureur : la législation devenant de plus en plus stricte en matière de protection des données, un assureur ne peut plus conserver au sein de son entreprise les données d'un assuré qui quitte son portefeuille. Il doit donc subir une perte de données, ce qui pose

problème pour la tarification de ses produits. Il souhaiterait donc pouvoir générer une fausse base de données à partir de ses données initiales, pour ensuite faire une série d'analyses sur cette "fausse" base de données et l'utiliser pour tarifer ses produits futurs. Voici un exemple concret de motivation à la génération de bases de données synthétiques.

Hormis le fait de capter les structures "statistiques", il faudra aussi analyser la "qualité" des données générées. En effet, il faudra éviter que le modèle se contente simplement de copier les données initiales et donc de répliquer celles-ci. Il existe différentes méthodes pour calculer une "distance" entre la base de données initiale et la base de données générée.

3.1 TabDiff

Cette section aura pour objectif de rappeler les concepts fondamentaux des deux chapitres précédents. La génération de données synthétiques pour une base de données contenant uniquement des variables numériques repose sur les modèles de diffusion, qui sont utilisés dans la génération d'images synthétiques. Les modèles de diffusion reposent sur deux processus : *forward* et *backward*. Ces modèles ont pour objectif d'inverser le processus *forward*, qui a pour but d'ajouter progressivement du bruit à des données. Cela est fait dans le but de pouvoir appliquer ce modèle à un bruit gaussien afin de générer de nouvelles données (*reverse*).

Pour les données discrètes, le concept est le même. Le processus *forward* applique une perturbation sur la distribution des modalités dans le but d'équilibrer la distribution de chaque modalité. C'est-à-dire d'obtenir, à terme, une distribution où chaque modalité a la même probabilité de survenance. Le processus *reverse* inverse le processus *forward*, et donc, à partir de la distribution où chaque modalité a la même probabilité de survenance, le processus *reverse* tente, pas à pas, de retrouver la distribution initiale.

3.1.1 Données numériques

Soit $x_{\text{num},0} \in \mathbb{R}^d$ un vecteur de variables numériques de dimension d. Le processus *forward*, qui est un processus de Markov, a pour objectif d'obtenir une variable $x_{\text{num},T}$ qui suit une loi normale $\mathcal{N}(0, I)$ en ajoutant petit à petit (en T étapes) un bruit gaussien à $x_{\text{num},0}$. Pour chaque t, on a :

$$x_{\text{num},t} = \sqrt{1-\beta_t}\, x_{\text{num},t-1} + \sqrt{\beta_t}\, \epsilon_{t-1},$$

où $\epsilon_{t-1} \sim \mathcal{N}(0, I)$ et $\beta_t \in [0, 1]$. Le paramètre β_t est appelé le *taux de diffusion*.

On obtient alors :

$$q_{\text{num}}(x_{\text{num},t} \mid x_{\text{num},0}) = \mathcal{N}(x_{\text{num},t}; \sqrt{\bar{\alpha}_t}\, x_{\text{num},0}, (1-\bar{\alpha}_t)\, \mathbf{I})$$

avec $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ et $\alpha_i = 1 - \beta_i$.

Le processus *reverse* est appris à l'aide d'un modèle ϵ_θ (typiquement, un réseau de neurones)

qui prédit le bruit ajouté lors de chaque étape. Le modèle cherche à prédire le bruit ϵ en prenant comme input le temps t et les données bruitées $x_{\text{num},t}$ au moment t.

Par hypothèse de normalité du processus *reverse*, celui-ci est défini par :

$$p_\theta(x_{\text{num},t-1} \mid x_{\text{num},t}) = \mathcal{N}(x_{\text{num},t-1}; \mu_\theta(x_{\text{num},t}, t), \Sigma_\theta(x_{\text{num},t}, t))$$

avec :

$$\mu_\theta(x_{\text{num},t}, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_{\text{num},t} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_{\text{num},t}, t) \right),$$
$$\Sigma_\theta(x_{\text{num},t}, t) = \beta_t \, \mathbf{I}.$$

Le modèle est entraîné de manière à minimiser, pour tout $t \in [0, 1]$:

$$L_t^{\text{num}} = \mathbb{E}\left[\left\| \epsilon - \epsilon_\theta(x_{\text{num},t}, t) \right\|^2 \right].$$

3.1.2 Données discrètes

Dans le cadre de données catégorielles, le modèle de génération est aussi un modèle de diffusion. C'est un modèle qui s'appuie sur un processus de bruitage multinomial.

Soit $x_{cat,0}$ une variable catégorielle avec n catégories, qui est encodée de manière one-hot. Le processus *forward*, également un processus de Markov, se définit comme :

$$q_{cat}(x_{cat,t} \mid x_{cat,t-1}) = C\left(x_{cat,t} \mid (1-\beta_t) x_{cat,t-1} + \frac{\beta_t}{n} \right).$$

avec $\beta_t \in (0, 1)$, le taux de bruit à l'instant t. On en déduit que :

$$q_{cat}(x_{cat,t} \mid x_{cat,0}) = C\left(x_{cat,t} \mid \bar{\alpha}_t \, x_{cat,0} + \frac{1-\bar{\alpha}_t}{n} \right).$$

Le processus *reverse* p_θ est paramétré par un modèle paramétrique $\hat{x}_{cat,0}(x_{cat,t}, t)$ qui cherche à prédire $x_{cat,0}$ à partir de $x_{cat,t}$ et t. Il est défini comme :

$$p_\theta(x_{cat,t-1} \mid x_{cat,t}) = q_{cat}(x_{cat,t-1} \mid x_{cat,t}, \widehat{x}_{cat,0}(x_{cat,t}, t))$$

où :

$$q_{cat}(x_{cat,t-1} \mid x_{cat,t}, x_{cat,0}) = C\left(x_{cat,t}; \frac{\pi}{\sum_{k=1}^{n} \pi_k} \right),$$

avec :

$$\pi = \left[\alpha_t \, x_{cat,t} + \frac{1-\alpha_t}{n} \right] \odot \left[\bar{\alpha}_{t-1} \, x_{cat,0} + \frac{1-\bar{\alpha}_{t-1}}{n} \right],$$

et π_k est l'élément k du vecteur π.

Le chapitre 1 montre que le modèle de diffusion maximise la borne inférieure de la fonction de vraisemblance :

$$\log L \geqslant \mathbb{E}\left[-D_{KL}(q(x_T \mid x_0)\|p_\theta(x_T)) \; - \sum_{t=2}^{T} D_{KL}(q(x_{t-1} \mid x_0, x_t)\|p_\theta(x_{t-1} \mid x_t)) + \log p_\theta(x_0 \mid x_1)\right].$$

Pour les variables catégorielles, la borne inférieure est identique puisque la preuve est générale pour tout type de distribution. Supposons que la base de données contient C variables. Alors, définissons pour tout $t \in [0, 1]$ et $i \in \{1, \cdots, C\}$:

$$L_t^{cat,i} = D_{KL}\left[q_{cat}(x^i_{cat,t-1} \mid x^i_{cat,t}, x^i_{cat,0}) \,\|\, q_{cat}(x^i_{cat,t-1} \mid x^i_{cat,t}, \widehat{x}^i_{cat,0}(x_{cat,t}, t))\right]$$

où :

$$x_{cat,t} = [x^1_{cat,t}, \cdots, x^C_{cat,t}].$$

L'objectif est donc de minimiser, pour tout $t \in [0, 1]$:

$$L_t^{cat} = \frac{1}{C}\sum_{i=1}^{C} L_t^{cat,i}.$$

3.1.3 Données mixtes (continues + discrètes)

Les données mixtes sont des données qui contiennent à la fois des variables numériques et des variables discrètes. Supposons qu'une base de données contienne C variables catégorielles et N variables numériques alors $x_{cat,0}$ représente le vecteur des variables catégorielles one-hot encodé (avec $x^i_{cat,0}$ la variable $i \in \{1, \cdots, C\}$ one-hot encodée et donc $x_{cat,0} = [x^1_{cat,0}, \cdots, x^C_{cat,0}]$) et $x_{num,0}$ le vecteur contenant les variables numériques. Notons $x_0 = [x_{num,0}, x_{cat,0}]$.

L'objectif est de minimiser pour tout $t \in [0, 1]$

$$L_t = L_t^{num} + L_t^{cat}.$$

Cependant, dans le cadre des données mixtes

$$L_t^{num} = \mathbb{E}\Big[||\epsilon - \epsilon_\theta(x_t, t)||^2\Big].$$

et

$$L_t^{cat,i} = D_{KL}[q_{cat}(x^i_{cat,t-1}|x^i_{cat,t}, x^i_{cat,0})||q_{cat}(x^i_{cat,t-1}|x^i_{cat,t}, \widehat{x}^i_{cat,0}(x_t, t))].$$

Le réseau de neurones ϵ_θ, qui dans le cadre d'un modèle de diffusion avec uniquement des données numériques prédit le bruit ajouté aux variables numériques à partir de $x_{num,t}$ et de t, maintenant, puisque nous sommes dans un contexte de données mixtes, le réseau de neurones ϵ_θ dépend

également des variables catégorielles $x_{cat,t}$. Cela veut dire que la prédiction du bruit pour les variables numériques s'appuie à la fois sur les variables numériques bruitées mais également sur les variables discrètes issues de la distribution bruitée au moment t. De façon similaire, le réseau de neurones $\widehat{x}_{cat,0}$, chargé de prédire les variables catégorielles, ne se base plus uniquement sur les variables discrètes issues de la distribution bruitée au moment t mais ajoute aussi comme input les variables numériques bruitées. Cela semble tout à fait logique, puisque les deux sortes de variables contiennent de l'information utile l'une pour l'autre.

Notons que les articles [7] et [8] utilisent un seul réseau de neurones dans leur étude. En effet, ils utilisent un réseau de neurones qui prédit à la fois le bruit ajouté pour les variables numériques et les prédictions des états initiaux pour les variables catégorielles avec plusieurs fonctions d'activation softmax dépendant du nombre de variables discrètes. Dans le cadre de ce mémoire, on décide d'utiliser deux réseaux de neurones comme indiqué juste avant. Cela permet d'avoir plus de flexibilité et de mieux pouvoir gérer la fonction de perte puisque l'ordre de grandeur n'est pas le même. Il se pourrait également que la structure idéale du réseau de neurones ne soit pas la même pour les variables discrètes que pour les variables numériques. La figure 3.1.1 schématise la structure générale du modèle pour la génération de données mixtes.

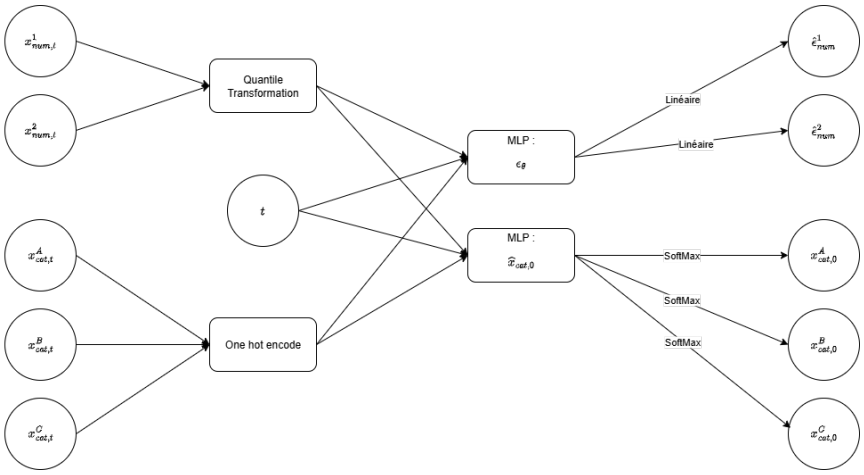


FIGURE 3.1.1 – Structure générale du modèle pour les données mixtes

3.2 Simulation de données mixtes

Cette section est consacrée à la simultaion de données mixtes. L'algorithme 5 décrit le processus de génération de données mixtes à partir d'un modèle de diffusion, en combinant à la fois des variables numériques et catégorielles en utilisant $T \in \mathbb{N}$ étapes de diffusion.

Le processus commence à l'étape $t = 1$, le bruit y est maximal. Les N variables numériques $x_{num,1}$ sont initialisées avec un bruit gaussien $\mathcal{N}(0, \mathbf{I}_N)$. Chacune des variables catégorielles est échantillonée selon une distribution où chaque modalité est équi-probable. Une boucle décroissante sur

$k \in \{T, \ldots, 1\}$ est effectuée.

À chaque itération, on cherche à obtenir des données de moins en moins bruitées à l'instant $t = \frac{k-1}{T}$ à partir des données bruitées x_{t^+} à l'instant $t^+ = \frac{k}{T}$ obtenues à l'étape précédente. Pour les variables numériques, $x_{cat,t}$ suit l'équation de débruitage construite dans le cadre des modèles de diffusion pour les variables continues. Cette mise à jour utilise la prédiction du bruit $\epsilon_\theta(x_{t^+}, t^+)$ produite par le réseau de neurones ϵ_θ. A chaque étape, les variables catégorielles sont échantillonées à partir de la distribution conditionnelle $q_{\text{cat}}(x^i_{\text{cat},t} \mid x^i_{\text{cat},t^+}, \widehat{x}^i_{cat,0}(x_{t^+}, t^+))$, où $\widehat{x}^i_{cat,0}(x_{t^+}, t^+)$ est l'estimation de la variable catégorielle initiale, obtenue par le réseau de neurones $\widehat{x}^i_{cat,0}$ à partir des données bruitées x_{t^+}.

L'objectif est donc de pouvoir utiliser cet algorithme pour générer de nouvelles données mais cette fois, des données contenant à la fois des variables discrètes et des variables numériques.

Algorithm 5 Algorithme de génération de données mixtes avec T étapes de diffusion

$x_{num,1} \sim \mathcal{N}(0, \mathbf{I}_N)$
$x^i_{cat,1} \sim q_{cat}(x^i_{cat,1}) \quad \forall i \in \{1, \cdots, C\}$
for $k = T \longrightarrow 1$: **do**
  $t^+ = \frac{k}{T}$
  $t = \frac{k-1}{T}$
  $\epsilon \sim \mathcal{N}(0, \mathbf{I}_N)$
  $x_{num,t} = \frac{1}{\sqrt{\alpha_t}}\left(x_{num,t^+} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_{t^+}, t^+)\right)$
  $x^i_{cat,t} = q_{cat}(x^i_{cat,t} | x^i_{cat,t^+}, \widehat{x}^i_{cat,0}(x_{t^+}, t^+)) \quad \forall i \in \{1, \cdots, C\}$
  $x_{cat,t} = [x^1_{cat,t}, \cdots, x^C_{cat,t}]$
  $x_t = [x_{num,t}, x_{cat,t}]$
end for
return x_0

3.3 Entraînement du modèle

Lors des deux chapitres précédents, aucune recherche sur la structure des réseaux de neurones n'a été faite. Pour la génération de données mixtes, nous avons essayé de chercher la structure optimale pour chaque réseau de neurones. On essaiera une dizaine de structures différentes par réseau de neurones. Les résultats sont repris dans l'annexe sur les tables .2.1 et .2.2. Nous avons cherché à optimiser exclusivement la structure du réseau de neurones. Le taux de diffusion utilisé correspond à un schéma cosinusoïdal. Le temps t est introduit dans le modèle au moyen d'un *sinusoidal embedding* de dimension 32.

Afin de déterminer les paramètres optimaux, le jeu de données a été divisé en trois ensembles : training set (70% des données), validation set (15%) et test set (15%). L'apprentissage est réalisé sur l'ensemble d'entraînement à l'aide de la descente de gradient, et les paramètres du modèle sont

mis à jour tant que l'erreur de validation continue de s'améliorer. L'entraînement est interrompu dès que l'erreur sur l'ensemble de validation ne diminue plus pendant 20 époques consécutives (*early stopping*). Le modèle final est sélectionné comme celui ayant obtenu la plus faible erreur sur l'ensemble de test.

Les structures optimales qui seront utilisées pour la suite de l'analyse sont :

— Pour le réseau $\epsilon_t(x_t, t)$, qui a pour but de prédire le bruit contenu dans les variables continues, la structure du réseau de neurones est une structure avec 5 couches intermédiaires (et des fonctions d'activation ReLU) et des neurones qui vont décroître : $1024 \times 512 \times 256 \times 128 \times 64$.

— Pour le réseau $\widehat{x_0}(x_t, t)$, qui a pour objectif de prédire la catégorie x_0, la structure de ce dernier est la suivante : $1024 \times 512 \times 256$.

3.4 Analyse de données synthétiques mixtes

Dans cette section, on génère des données synthétiques à partir de la base de données utilisée dans les deux chapitres précédents. Cette fois-ci, on utilise les variables discrètes et continues. On se concentrera sur les données produites, comme ce fut le cas dans les deux autres chapitres. On regardera notamment les distributions des variables, mais également comment les variables réagissent entre elles. Est-ce que les structures de dépendance sont bien respectées ?

Une partie importante de cette section sera aussi de pouvoir juger la qualité des données produites. De nouvelles notions seront introduites pour pouvoir justement critiquer la qualité des données générées.

On générera 100 000 données synthétiques qui comprendront donc à la fois des variables discrètes et des variables continues.

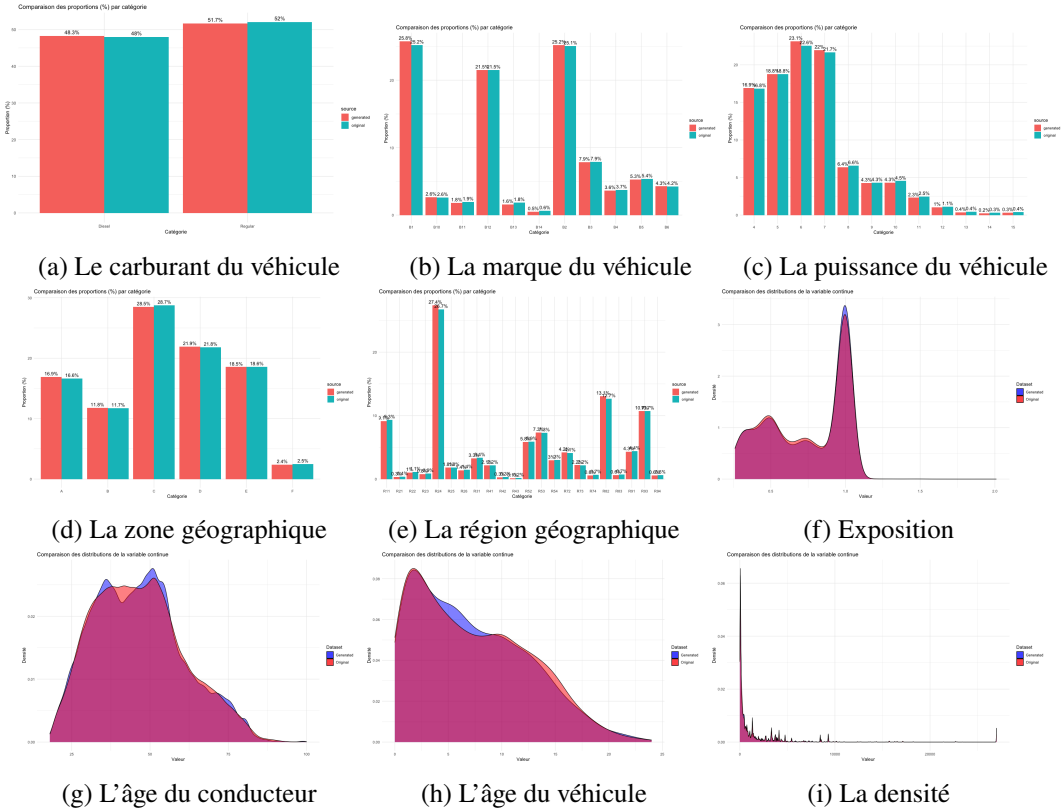
3.4.1 Distribution univariée des données

Cette sous-section aura pour objectif de montrer les distributions univariées de chaque variable afin de pouvoir comparer les distributions dans les deux jeux de données. On constate que les distributions sont assez bien respectées. La figure 3.4.1 contient les distributions des différentes variables du jeu de données. Cette figure permet de comparer, pour chacune des variables, la distribution dans chacun des deux datasets.

Pour les variables catégorielles, les proportions des différentes modalités du jeu de données synthétiques sont proches de celles des données réelles. On observe de petites variations, mais ces variations sont très minimes. Elles sont toujours inférieures à 1 %. Évidemment, on souhaiterait obtenir aucune différence dans les proportions, mais ces écarts sont également un bon signe puisque cela laisse sous-entendre que le modèle ne fait pas d'overfitting. Les variables avec un grand nombre de catégories, comme la marque du véhicule ou la région, sont bien distribuées dans les données synthétiques puisque les proportions sont fortement similaires à celles des données originales.

Pour les variables continues, on constate de nouveau que le modèle capte bien la structure des données. Les distributions univariées sont relativement bien reproduites dans les données synthétiques. La densité et l'exposition au risque ont une distribution quasiment identique dans les deux jeux de données. Cela montre que le modèle arrive à bien capter la tendance de ces variables. La forme générale des courbes de densité pour l'âge du conducteur et l'âge du véhicule est bonne. Elles sont similaires dans les deux jeux de données. On observe cependant de petites différences. Les données synthétiques contiennent plus de véhicules âgés entre 5 et 10 ans que la base de données initiale. Les données synthétiques ne capturent pas exactement les petites variations dans la densité de la variable liée à l'âge du conducteur entre 30 et 50 ans.

De manière générale, le modèle donne des résultats très satisfaisants pour les distributions univariées. Il arrive à capturer les distributions statistiques de chacune des variables sans pour autant faire de l'overfitting.



(a) Le carburant du véhicule (b) La marque du véhicule (c) La puissance du véhicule

(d) La zone géographique (e) La région géographique (f) Exposition

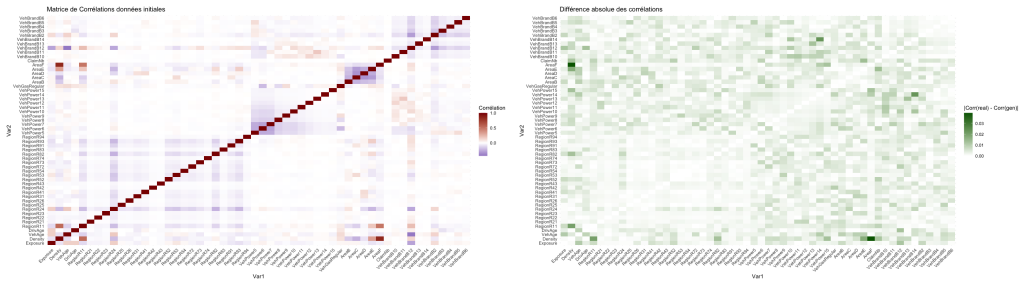
(g) L'âge du conducteur (h) L'âge du véhicule (i) La densité

FIGURE 3.4.1 – Comparaison des densités pour chacune des variables entre les données synthétiques et les données initiales

3.4.2 Corrélation

Cette fois-ci, pour vérifier que le modèle capture bien la structure entre les différentes variables, une analyse des matrices de corrélation est faite. Pour réaliser cela, on one-hot encode les variables discrètes et on calcule la matrice de corrélation pour le jeu de données initial. On réalise la même chose pour les données synthétiques et ensuite on peut facilement obtenir une matrice contenant la différence des deux matrices de corrélation (en valeur absolue). Le résultat est disponible sur la figure 3.4.2.

La figure 3.4.2 montre sans surprise que les différentes variables interagissent ensemble. Il est évident que la variable puissance du véhicule n'est pas indépendante de la variable marque du véhicule. Certaines marques ont tendance à produire des voitures plus puissantes que d'autres. Par exemple, il est clair que la variable continue `Density` interagit avec la variable de la zone géographique. On voit bien qu'il y a des corrélations non nulles dans la base de données initiale. Sur la figure de droite, la matrice des différences en valeur absolue montre que le modèle capture bien la corrélation entre les variables. En effet, l'écart le plus grand en valeur absolue est inférieur à 4 %. On observe que la majorité des différences est inférieure à 1 %, ce qui indique que le modèle capture bien les interactions entre les différentes variables.



(a) La matrice de corrélation des données initiales avec les variables discrètes encodée avec un one-hot encodeur

(b) La différence entre les matrice de corrélation des données initiales et des données synthétiques en valeur absolue

FIGURE 3.4.2 – Comparaison des matrices de corrélation

3.4.3 α-precision

La mesure α-precision est une métrique qui sert à mesurer si les données synthétiques ressemblent aux données initiales. En d'autres mots, c'est une mesure de cohérence des données synthétiques. Cela cherche à mesurer que, par exemple, pour l'âge du conducteur, on n'obtienne pas des âges négatifs.

Soient $\mathcal{D}_{\text{réel}}$ la base de données réelle et $\mathcal{D}_{\text{syn}}$ la base de données synthétique. La mesure α-precision est définie comme la proportion des données synthétiques $\mathcal{D}_{\text{syn}}$ pour lesquelles il existe

au moins une donnée issue des données réelles $\mathcal{D}_{\text{réel}}$ située à une distance inférieure à un seuil r :

$$\alpha\text{-precision} = \frac{1}{|\mathcal{D}_{\text{syn}}|} \sum_{x_{\text{syn}} \in \mathcal{D}_{\text{syn}}} \mathbf{1}\left\{ \min_{x \in \mathcal{D}_{\text{réel}}} d(x_{\text{syn}}, x) \leqslant r \right\},$$

où $d(x_{\text{syn}}, x)$ est la distance euclidienne entre x et x_{syn}, $\mathbf{1}$ est la fonction indicatrice, et $|\mathcal{D}_{\text{syn}}|$ est le nombre de données contenu dans le dataset $\mathcal{D}_{\text{syn}}$.

Cette mesure est donc une mesure de fiabilité des données générées afin de pouvoir mesurer si les données produites sont fiables ou non. Si cette mesure est petite, cela voudrait laisser sous-entendre que les données synthétiques sont absurdes.

Pour choisir le seuil r, on calcule, pour chaque point $x_i \in \mathcal{D}_{\text{réel}}$, la boule de rayon r_i centrée en x_i contenant les 10 plus proches voisins de x_i. Ce rayon est donc égal à la distance entre x_i et son dixième plus proche voisin. Ceci est évidemment effectué sur un espace réduit et standardisé (à l'aide d'une PCA) pour plus de fiabilité et pour des raisons numériques. Formellement, le rayon r_i de la boule centrée en x_i contenant ses 10 plus proches voisins est défini comme :

$$r_i = ||x_i - \text{NN}_{10}(x_i)||^2$$

avec $\text{NN}_{10}(x_i)$ qui est le 10^{e} plus proche voisin de x_i.

Le seuil r est défini comme étant le quantile à 95 % de ces rayons :

$$r = Q_{95\%}(r_i).$$

Nous obtenons sur nos données un α-precision qui est égal à :

$$\alpha\text{-precision} = 97.33\%.$$

C'est un très bon score qui laisse penser que le modèle ne crée pas de données absurdes.

La figure 3.4.3 montre la projection sur les deux premiers axes principaux des deux jeux de données. On constate que les données synthétiques font sens par rapport aux données réelles. Les données synthétiques sont régulièrement proches des données réelles. Il semblerait toutefois que les données synthétiques au centre du graphe ne soient pas très proches des données réelles. Les données réelles forment deux clusters bien distincts et il semblerait que le modèle produise quelques données qui se trouvent entre ces deux clusters. La forme générale est bonne et la distribution des points sur les deux premiers axes est fortement similaire. Sur les deux premiers axes, on ne distingue pas de points de la base de données synthétique qui se trouvent à des endroits "absurdes" par rapport aux points de la base de données réelles. Cette figure confirme donc le score de la métrique α-precision.

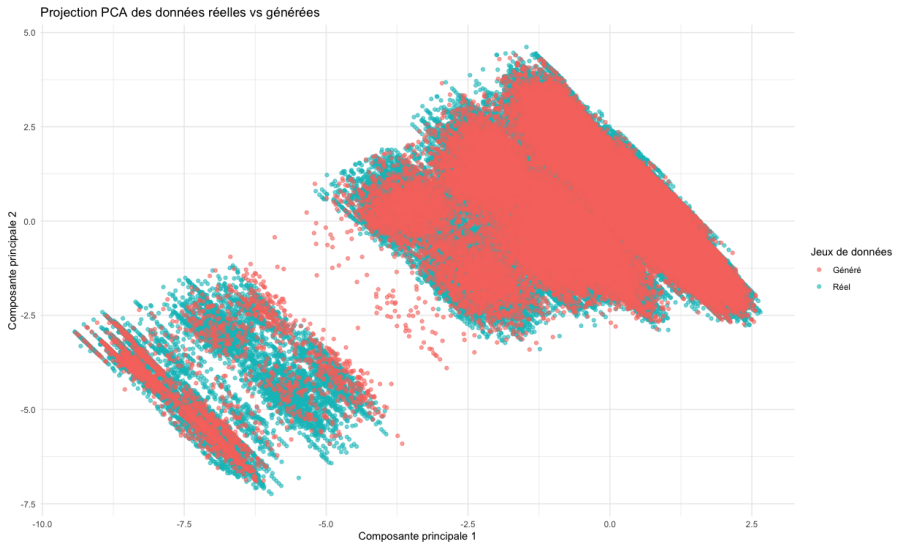


FIGURE 3.4.3 – Projection sur les deux premiers axes principaux des données générées et données réelles

3.4.4 β-Recall

La mesure β-recall est également une métrique, mais qui a un objectif différent de la métrique α-precision. Le β-recall est utile pour évaluer la complétude du modèle génératif. Elle va mesurer la capacité du modèle à couvrir toutes les données de la base de données initiale. Il se pourrait par exemple que le modèle génératif ne génère jamais de données qui possèdent une densité supérieure à 10 000, par exemple.

Le β-recall se définit comme la proportion des données initiales $\mathcal{D}_{\text{réel}}$ pour lesquelles il existe au moins une donnée synthétique $x_{\text{syn}} \in \mathcal{D}_{\text{syn}}$ située à une distance inférieure à un seuil r :

$$\beta\text{-recall} = \frac{1}{|\mathcal{D}_{\text{réel}}|} \sum_{x \in \mathcal{D}_{\text{réel}}} \mathbf{1}\left\{ \min_{x_{\text{syn}} \in \mathcal{D}_{\text{syn}}} d(x_{\text{syn}}, x) \leqslant r \right\},$$

où $d(x_{\text{syn}}, x)$ est la distance euclidienne entre x et x_{syn}, $\mathbf{1}$ est la fonction indicatrice, et $|\mathcal{D}_{\text{syn}}|$ est le nombre de données contenu dans le dataset $\mathcal{D}_{\text{syn}}$.

Cette mesure va donc vérifier que les données réelles soient bien représentées dans l'espace généré par le modèle. Si le β-recall est faible, cela voudrait laisser sous-entendre que le modèle génère uniquement une partie des données initiales, et donc ne couvre pas tous les types de polices.

Le seuil r est défini de la même manière que pour la mesure α-precision. Nous obtenons sur nos données un β-recall qui est égal à :

$$\beta\text{-recall} = 88.22\%.$$

La mesure est proche de 1, ce qui laisse penser que le modèle génère des données diversifiées et que la majorité des données du dataset initial est représentée dans le jeu de données synthétique.

La figure 3.4.3 permet de confirmer la haute valeur de la métrique β-recall. En effet, on constate sur les deux premiers axes principaux que les points des données réelles sont régulièrement "entourés" de données synthétiques. Cela laisse penser que beaucoup de points du jeu de données réelles sont représentés dans le jeu de données synthétique. On constate cependant que, pour les points de la base de données initiale qui ont sur le premier axe une valeur entre -7.5 et -5 et sur le second axe une valeur entre -7.5 et -2.5, il y a une moins grande densité de données synthétiques générées dans cette zone du graphique. Cette région représente une quantité non négligeable de points issus du dataset réel, alors qu'en proportion, les données synthétiques sont moins présentes dans cette région, bien qu'il y en ait quand même qui y soient générées. En conclusion, la figure 3.4.3 confirme le score de 88 %.

3.4.5 Distance to Closest Record (DCR)

Le DCR est une mesure qui permet de juger la qualité des données synthétiques. Elle pourrait aussi être utilisée pour comparer les performances entre des méthodes de génération de données synthétiques différentes. Le DCR mesure, pour chaque donnée synthétique, la distance la plus petite entre la donnée synthétique et le point le plus proche des données réelles. Cette mesure permet de vérifier que les données générées ne sont pas trop proches des données réelles, ce qui signifierait qu'il y a de l'overfitting et que le modèle rééchantillonne uniquement la base de données initiale. Si le DCR est trop grand, cela veut dire que les deux jeux de données sont éloignés l'un de l'autre, et que le modèle ne capture pas bien les structures de la base de données réelle.

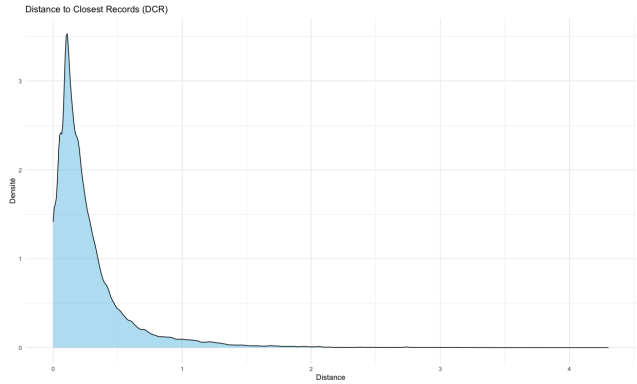
De manière formelle, soient $\mathcal{D}_{\text{réel}}$ la base de données réelle et $\mathcal{D}_{\text{syn}}$ la base de données synthétique. Le DCR moyen est défini comme :

$$\text{DCR} = \frac{1}{|\mathcal{D}_{\text{syn}}|} \sum_{x_{\text{syn}} \in \mathcal{D}_{\text{syn}}} \min_{x \in \mathcal{D}_{\text{réel}}} d(x_{\text{syn}}, x).$$

Notons, pour tout $x_i \in \mathcal{D}_{\text{syn}}$,

$$\text{DCR}_i = \min_{x \in \mathcal{D}_{\text{réel}}} d(x_i, x).$$

La figure 3.4.4 représente la densité de l'échantillon DCR_i pour les données synthétiques. On remarque que la distribution penche fortement à gauche, avec des distances autour de 0.2. Il y a uniquement 307 données qui ont une distance DCR_i égale à 0. Cela représente donc environ 0.3% des données qui sont une copie conforme des données de la base de données initiale. Cela montre que le modèle génère bien des données différentes que les données initiales. C'est une métrique qui est utile pour comparer des méthodes de générations entre elles.



(a) La densité des DCR_i

Statistique	DCR
Min.	0.0000
1er Qu.	0.1002
Médiane	0.1876
Moyenne	0.2838
3e Qu.	0.3390
Max.	4.3024

(b) Résumé statistique des distances DCR

FIGURE 3.4.4 – Comparaison graphique et statistique des distances DCR : (a) densité estimée, (b) résumé numérique.

3.4.6 Analyse de la fréquence sinistre

Le livre [2] permet de bien comprendre les pratiques utilisées par les actuaires dans la modélisation de la fréquence sinistre.

Dans l'introduction de ce mémoire, l'exemple pour motiver ce travail était celui de l'assureur. Les législations étant de plus en plus strictes, les assireurs ne peuvent plus garder l'information des polices d'assurance qui quittent leur portefeuille. Le but étant bien entendu de pouvoir utiliser les données synthétiques pour pouvoir faire leurs analyses. Ces analyses doivent donc donner des résultats similaires aux analyses faites sur le portefeuille initial. On se doute bien que les analyses seront bien plus poussées que de simples analyses descriptives comme ce fut le cas jusqu'ici dans le mémoire.

Une variable importante pour chaque assuré est de pouvoir estimer la fréquence sinistre de chaque assuré. La fréquence sinistre est définie comme

$$\text{Fréquence sinistre} = \frac{\text{Nombre de sinistres}}{\text{Exposition}}.$$

Pour pouvoir l'estimer, les assureurs utilisent un modèle prédictif. Le modèle prédictif a pour but d'estimer la fréquence sinistre de chaque assuré en utilisant les différentes caractèristiques de l'assuré. Dans le cadre de ce mémoire, on utilisera un GLM avec une loi de Poisson. Le but étant de pouvoir estimer le nombre de sinistres qui est une variable de comptage, la loi de poisson s'y prête bien. La fonction de lien dans un GLM avec un loi de poisson est la fonction exponentielle donc le modèle sera de la forme

$$\hat{\mu}_i = \text{Fréquence sinistre}_i = \text{Exposition}_i \ \exp(\alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \cdots + \alpha_n x_{ni})$$

avec l'exposition qui est mise en *offset* et en utilisant n variables.

La fonction d'erreur utilisée est la déviance. Si la fonction d'erreur quadratique moyenne est utilisée, alors le modèle ne sera pas optimal car cette métrique n'est pas consistante par rapport aux données. Cette métrique ne tient pas compte de la distribution des données. Il est préférable d'utiliser la déviance. Le choix d'une perte quadratique pour le nombre des sinistres introduirait un biais, car on ne prend pas en compte les caractéristiques de la distribution statistique de la fréquence des sinistres lors de la modélisation. La solution consiste à utiliser comme fonction de perte comme la déviance afin de traiter ce problème.

Définition 3.4.1. La déviance d'un modèle est défini comme

$$D(y, \hat{\mu}) = 2(L_{full} - L(\hat{\mu}))$$

où L_{full} est la fonction de vraisemblance logaritmique du modèle full et $L(\hat{\mu})$ celle du GLM.

Pour rappel, la fonction de densité d'une loi de poisson de paramètre λ est définie par

$$f(k) = \frac{\lambda^k}{k!} \exp(-\lambda).$$

Soient y_i le nombre de sinistres observé pour la police i et $\hat{\mu}_i$ la prédiction du modèle pour l'individu i. La fonction de déviance pour une loi de poisson est obtenue par

$$\begin{aligned} D(y, \hat{\mu}) &= 2 \sum_{i=1}^{n} \ln(\frac{{y_i}^{y_i}}{y_i!} \exp(-y_i)) - 2 \sum_{i=1}^{n} \ln(\frac{\hat{\mu}_i^{y_i}}{y_i!} \exp(-\hat{\mu}_i)) \\ &= 2 \sum_{i=1}^{n} \ln \lambda_i^{y_i} - \ln \hat{\mu}_i^{y_i} - y_i + \hat{\mu}_i \\ &= 2 \sum_{i=1}^{n} y_i \ln \frac{y_i}{\hat{\mu}_i} - y_i + \hat{\mu}_i \end{aligned}$$

avec $y_i \ln \frac{y_i}{\hat{\mu}_i} = 0$ si $y_i = 0$.

Puisque le modèle full est un modèle qui est sensé être meilleur que n'importe quel autre modèle, la fonction de déviance n'est jamais négative. Plus la déviance est grande, plus il y a des différences entre les valeurs observés et les valeurs prédites par le modèle. En résumé, la déviance mesure une distance moyenne entre un modèle particulier (dans notre cas, le GLM) et le modèlle full.

Pour vérifier la qualité des données synthétiques, nous allons construire un modèle GAM sur le training set des données initiales. Un GAM est similaire à un GLM, il traite juste les variables continues via une fonction qui est une spline. Un GAM (dans le cadre d'une loi de poisson) est donc de la forme suivante pour un modèle avec n variables catégorielles et k variables continues :

$$\hat{\mu}_i = \text{Fréquence sinistre}_i = \text{Exposition}_i \exp(\alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \cdots + \alpha_n x_{ni} + f_1(x^1_{cont,i}) + \cdots + f_k(x^k_{cont,i})).$$

Le training set contiendra 80% des données initiales et par conséquent le testing set représentera 20% des données. Le but est donc de pouvoir construire un modèle sur le trainig set et ensuite obtenir la déviance la plus petite possible sur le testing set.

Un modèle identique est construit sur l'ensemble des données synthétiques. Le but est de maintenant pouvoir comparer les prédictions des deux modèles sur le testing set des données initiales. Le modèle sélectionné est le modèle suivant

$$\begin{aligned}\hat{\mu}_i = \text{Exposition}_i \ \exp(&\alpha_{\text{Area}} x_{\text{Area},i} + \alpha_{\text{VehBrand}} x_{\text{VehBrand},i} + \alpha_{\text{VehGas}} x_{\text{VehPower},i} + \alpha_{\text{Region}} x_{\text{Region},i} \\ &+ \alpha_{\text{VehPower}} x_{\text{VehPower},i} + f_{\text{VehAge}}(x_{\text{VehAge},i}) + f_{\text{DrivAge}}(x_{\text{DrivAge},i}) + \\ &f_{\text{Density}}(x_{\text{Density},i})).\end{aligned}$$

Un tableau résumé des résultats se trouve sur la table 3.4.1.

Type de déviance	**Modèle réel**	**Modèle synthétique**
Données synthétiques	0.392566	0.395629
Training set (initiales)	0.409606	0.4155741
Testing set (initiales)	0.4104784	0.4159935

TABLE 3.4.1 – Résumé des déviances moyennes

La déviance moyenne sur le testing set pour les deux modèles est très proche l'une de l'autre. Ce qui montre que la méthode semble assez bien capturer les tendances et les interactions entre les variables afin de prédire le nombre de sinistres d'un assuré. Un constat qui est frappant est que la déviance pour les données synthétiques est la plus faible, pour le modèle construit sur celles-ci, cela peut paraître logique puisque le modèle est construit à partir des données générées. Cependant, les données synthétiques sont celles avec la plus petite déviance pour le modèle construit sur les données réelles. Une hypothèse est que les données synthétiques sont plus "propres". Peut-être, contiennent-elles moins de bruit et moins d'outliers que les données réelles. Cela laisse penser que les données synthétiques sont une version "parfaite" des données initiales. La déviance moyenne sur le training et celle sur testing set pour le modèle synthétique sont très proches. C'est assez logique puisque ce sont des données "nouvelles" étant donné qu'elles n'ont pas été utilisées lors de l'élaboration du modèle. Elles peuvent être vues comme des données nouvelles avec 2 échantillons, il est dès lors logique que les erreurs soient proches.

La figure 3.4.5 compare les prédictions de sinistralité sur le testing set entre les deux modèles. Chaque point représente une observation, et la droite rouge en pointillés correspond à la diagonale d'égalité parfaite (y = x). La majorité des points sont proches de cette diagonale, suggérant une bonne cohérence entre les deux modèles. Cependant, on observe une certaine dispersion ce qui signifie que le modèle sur données synthétiques diffère légèrement du modèle sur les données initiales. La tendance est relativement bien respectée avec une tendance linéaire entre les fréquences prédites La figure indique que les données générées reproduisent globalement bien les tendances mais elles comportent du bruit.

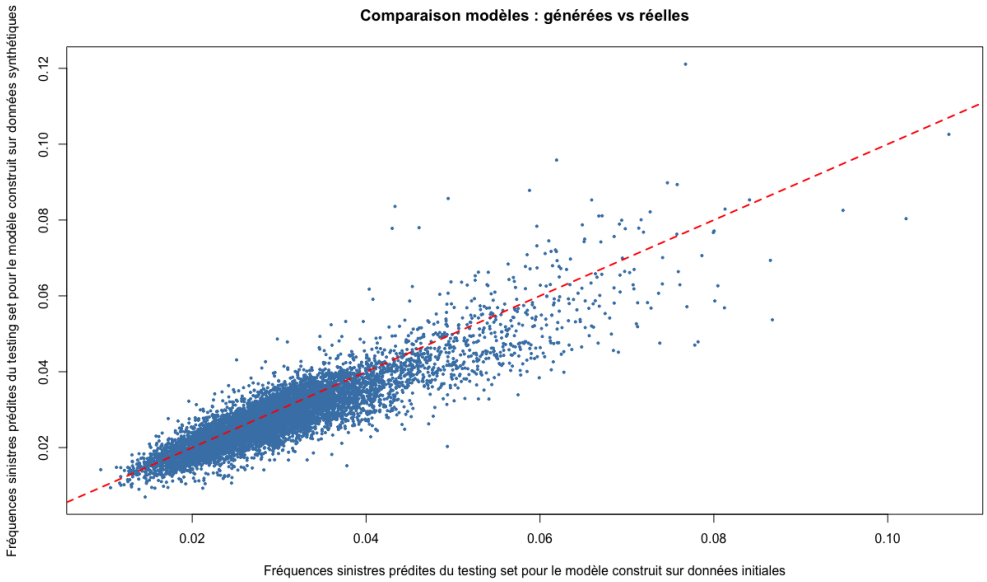


FIGURE 3.4.5 – Comparaison des prédictions de sinistralité sur le testing set entre les deux modèles (le graphique montre 10% des points pour des questions de visualisation)

La régression linéaire entre la fréquence prédite par le modèle sur les données initiales et la fréquence prédite par le modèle sur les données synthétiques est synthétisée sur la figure ci-dessous.

```
Call:
lm(formula = pred_gen ~ pred)

Residuals:
       Min          1Q      Median          3Q         Max
-0.0185478 -0.0014771 -0.0000161  0.0015351  0.0184935

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.678e-03  2.958e-05   124.4   <2e-16 ***
pred        8.906e-01  9.439e-04   943.6   <2e-16 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
       0.1          1

Residual standard error: 0.00263 on 90563 degrees of freedom
Multiple R-squared:  0.9077,    Adjusted R-squared:  0.9077
F-statistic: 9.003e+05 on 1 and 90563 DF,  p-value: < 2.2e-16
```

Cela montre que la pente est proche de 1 puisque la régression a une pente de 0.9 et que le paramètre est significatif. Le modèle calibré sur les données synthétiques prédit des fréquences sinistres légèrement inférieures aux fréquences sinistres prédites avec le modèle sur les données réelles. L'intercept du modèle de 0.004 est est significatif. Il est très proche de 0.

3.4.7 Est-il possible de retrouver les données synthétiques et réelles ?

Dans cette section, une analyse sera faite pour savoir si un modèle complexe (de Machine Learning) est capable pour une donnée de retrouver sa source : donnée réelle ou donnée synthhétique ? Un *Gradient Boosting Tree* est entrainé dans le but de retrouver l'origine des données, c'est-à-dire identifier si une donnée est issue du jeu de données généré artificiellement ou du jeu de données réel.

Pour ce faire, un jeu de données de 200 000 observations, composé de 100 000 données synthétiques et 100 000 données réelles, est construit. Le GBM sera alors entraîné sur ce jeu pour évaluer sa capacité à reconnaître la source des données à partir des variables explicatives.

Vraie / Prédite	**Synthétique**	**Réelle**
Synthétique	9400	10786
Réelle	3778	16036

FIGURE 3.4.6 – Matrice de confusion du GBM pour la classification des données réelles et synthétiques.

La figure 3.4.6 contient la matrice de confusion pour la classification sur le testing set. Le modèle parvient à relativement bien identifier les données réelles. En effet, le modèle prédit correctement 16 036 observations réelles sur un total de 19814, soit une précision de 81%. Cependant, cette performance n'est pas pleinement représentative puisque la classification des données synthétiques est nettement moins bonne : moins de 50% d'entre elles sont correctement identifiées. Le modèle a tendance à plus prédire les données réelles que les données synthétiques (car il y a légèrement plus de données réelles dans le dataset).

Ainsi, le modèle admet des difficultés à retrouver correctement la classe d'origine. Cela montre que le GBM n'arrive pas à dissocier clairement les données synthétiques des données réelles. C'est un résultat positif, car cela signifie qu'il est difficile, voire impossible, de déterminer la source d'une donnée, ce qui suggère que les données synthétiques reproduisent bien la structure des données réelles.

3.5 Discussion : limites, points positifs et améliorations possibles des modèles de diffusion

Le travail effectué sur les modèles de diffusion dans le cadre de la génération de données synthétiques présente des avantages et inconvénients. Cette section regroupera les différents

points forts et points faibles du modèle ainsi que les pistes d'amélioration et les complexifications envisageables.

3.5.1 Points positifs

Le premier point positif des modèles de diffusion dans le cadre de la génération tabulaire c'est que les données synthétiques reproduisent correctement les grandes tendances statistiques présentes dans le dataset initial. Les distributions univariées sont très bien respectées. Les interactions simples entre les variables sont bien capturées par le modèle. Même des analyses plus poussées comme l'analyse de la fréquence sinistre montrent que les données synthétiques capturent les comportements complexes observés dans les données réelles. Pour mesurer la qualité des données, les mesures α-precision et β-recall sont utiles.

L'idée derrière les modèles de diffusion est très simple à comprendre même pour une personne qui n'est pas familière à ces modèles. En effet, on part d'une distribution aléatoire connue et on applique petit à petit un processus qui cherche à retirer le bruit. Les illustrations avec la génération d'images permettent de très bien comprendre l'idée globale.

Les résultats montrent qu'il est difficile pour le modèle de distinguer clairement les données réelles des données synthétiques, ce qui est un signe de qualité pour les données générées.

Les données produites sont différentes des données initiales. De plus, il a été vérifié que les données produites ne sont pas absurdes. Par exemple, il n'y a pas de conducteurs avec des âges de -6 ou de 324 ans. Le DCR a permis de montrer que les données générées différaient des données initiales.

3.5.2 Points négatifs

Le premier point négatif est que les modèles de diffusion demandent de nombreux hyperparamètres (taux de diffusion, architecture des réseaux de neurones, etc.). C'est donc difficile de tester toutes les combinaisons possibles.

Puisque le modèle repose sur un réseau de neurones, il y a un danger d'overfitting. Un entraînement trop long ou un réseau de neurones avec trop de paramètres peut amener à un surapprentissage, réduisant la capacité de généralisation. Cela engendrerait alors un modèle qui fait uniquement un copier/coller des données originales. De plus, l'entraînement est coûteux en temps et en ressources.

Deux entraînements sur les mêmes données peuvent donner des résultats légèrement différents. Cet aspect aléatoire est d'ailleurs assez récurrent dans des modèles de machine learning (par exemple, les random forest)

Selon la forme du réseau de neurones et des paramètres choisis, les résultats peuvent être très différents.

3.5.3 Axes d'amélioration

Un premier axe d'amélioration est d'aller plus en profondeur dans la recherche des hyperparamètres et de la structure du réseau de neurones.

Une idée est de comparer différentes familles de modèles de génération pour identifier celles qui offrent le meilleur compromis entre qualité et complexité. Cela permettrait d'évaluer la qualité des modèles de diffusion par rapport à d'autres modèles génératifs.

Si l'objectif est de générer des données pour faire de la cllassification, il est possible d'envisager des modèles conditionnels qui tiennent compte d'une variable cible pour obtenir des données encore plus fidèles aux comportements attendus de la classe étudiée.

Une suite possible est de tester le modèle de génération à d'autres bases de données pour mieux évaluer la capacité de généralisation des modèles. Il faut essayer de tester le modèle pour des bases de données avec moins de lignes que celle utilisée dans ce travail, avec plus de variables explicatives, etc.

La base de données utilisée ici est "propre". Il n'y a aucune valeurs manquantes. Une piste pour la suite est d'étudier des données avec les valeurs manquantes. Pour les variables catégorielles, on pourrait créer une nouvelle catégorie mais pour les variables numériques, ce n'est pas direct. L'article [6] traite ce problème.

Conclusion

Les modèles de diffusion sont des modèles très performants dans le cadre de la génération d'images, puisqu'ils sont très populaires dans ce secteur. Le chapitre 1 montre les résultats impressionnants de ce modèle dans le cadre de la génération d'images, puisqu'il a été appliqué et modélisé sur un ensemble d'images représentant des fleurs. La première partie du mémoire met l'accent sur la théorie des modèles de diffusion et le raisonnement mathématique qui se cache derrière ce processus qui peut sembler complexe.

L'objectif de ce mémoire était d'étudier ce modèle dans le cadre de la génération de données tabulaires. Il a d'abord été appliqué à des variables continues. Cette application est relativement simple, puisque les images sont représentées par des matrices de valeurs continues. Il a d'ailleurs été montré que les résultats obtenus étaient très satisfaisants, puisque les propriétés statistiques des variables étaient bien préservées. De plus, les différentes interactions entre celles-ci étaient bien capturées par le modèle.

Le deuxième chapitre se concentre sur les variables catégorielles. Ce chapitre montre pourquoi appliquer le modèle de diffusion "classique" sur des données one-hot encoded ne fonctionne pas. Ensuite, une étude sur les modèles de diffusion multinomiaux est faite. Une introduction théorique au modèle multinomial est effectuée. Ce modèle est ensuite appliqué sur une base de données catégorielles. L'analyse de la base de données synthétiques montrera que les données générées sont de très bonne qualité.

Le dernier chapitre expliquera comment combiner les deux modèles afin de générer des données contenant des variables catégorielles et continues. Le modèle de diffusion utilisé pour les données tabulaires est appelé un modèle TaDIFF. Un rappel théorique est réalisé et ensuite une analyse approfondie des résultats, mêlant à la fois de nouvelles métriques et des résultats graphiques, est établie. Cette analyse montre que les données synthétiques produites sont de très bonne qualité, que ce soit en terme de diversification des données générées ou en termes de propriétés statistiques capturées par le modèle.

En conclusion, les modèles de diffusion sont des outils qui peuvent être utilisés par les assureurs, qui, comme expliqué dans l'introduction, se trouvent face à des réglementations de plus en plus strictes en termes de protection de données. Ils contiennent de nombreux points positifs mais également quelques points négatifs, comme expliqué dans le dernier chapitre. Les pistes de recherches et d'améliorations sont nombreuses pour ces modèles.

Appendices

.1 Annexe 1

Marque	Carburant	4	5	6	7	8	9	10	11	12	13	14	15
B1	Diesel	4%	8%	29%	12%	22%	4%	5%	16%	3%	6%	0%	0%
	Regular	29%	20%	12%	22%	6%	23%	9%	20%	5%	7%	11%	12%
B10	Diesel	0%	0%	0%	2%	6%	7%	7%	3%	30%	17%	6%	6%
	Regular	0%	0%	0%	1%	2%	3%	4%	4%	7%	10%	13%	15%
B11	Diesel	0%	0%	0%	2%	1%	1%	1%	4%	7%	6%	2%	1%
	Regular	0%	0%	0%	1%	2%	3%	2%	5%	7%	28%	14%	10%
B12	Diesel	0%	0%	1%	1%	2%	3%	11%	3%	17%	1%	3%	4%
	Regular	1%	1%	2%	2%	4%	5%	6%	3%	5%	4%	5%	23%
B13	Diesel	0%	1%	1%	1%	4%	1%	4%	1%	0%	1%	0%	0%
	Regular	1%	0%	1%	1%	1%	5%	2%	4%	2%	5%	4%	1%
B14	Diesel	0%	0%	0%	0%	1%	1%	1%	1%	0%	0%	0%	0%
	Regular	1%	0%	0%	0%	0%	1%	1%	0%	0%	1%	4%	0%
B2	Diesel	7%	20%	23%	17%	20%	3%	14%	0%	0%	0%	0%	0%
	Regular	31%	16%	9%	17%	7%	21%	8%	19%	2%	6%	10%	13%
B3	Diesel	3%	8%	5%	2%	4%	2%	5%	1%	3%	0%	0%	0%
	Regular	2%	4%	4%	4%	4%	7%	3%	6%	6%	3%	18%	11%
B4	Diesel	3%	3%	1%	1%	2%	1%	1%	1%	1%	0%	0%	0%
	Regular	4%	3%	2%	3%	2%	1%	2%	2%	1%	2%	3%	1%
B5	Diesel	1%	7%	2%	2%	1%	1%	8%	2%	0%	0%	0%	0%
	Regular	4%	4%	3%	4%	3%	2%	2%	2%	0%	2%	2%	2%
B6	Diesel	1%	1%	2%	2%	6%	2%	1%	0%	1%	0%	0%	0%
	Regular	9%	3%	1%	2%	2%	2%	2%	2%	0%	1%	5%	1%

TABLE .1.1 – Distribution en % des marques par puissance et type de carburant (les colonnes totalisent 100% par carburant)

Marque	Carburant	4	5	6	7	8	9	10	11	12	13	14	15
B1	Diesel	+1%	+1%	0%	0%	0%	+1%	+1%	+1%	0%	+2%	0%	0%
	Regular	0%	0%	+1%	-1%	+1%	0%	0%	+3%	0%	+2%	-1%	0%
B10	Diesel	0%	0%	0%	0%	0%	-1%	-1%	-1%	-2%	+2%	-1%	0%
	Regular	0%	0%	0%	-1%	0%	+1%	0%	-2%	-1%	-1%	-2%	0%
B11	Diesel	0%	0%	0%	0%	0%	+1%	0%	-1%	-2%	+3%	+2%	+1%
	Regular	0%	0%	0%	0%	-1%	0%	-1%	0%	0%	-5%	0%	-2%
B12	Diesel	0%	0%	0%	0%	0%	0%	0%	-1%	-2%	+1%	0%	+2%
	Regular	0%	0%	0%	0%	0%	0%	-1%	0%	0%	+1%	-3%	-4%
B13	Diesel	0%	0%	0%	0%	0%	0%	0%	0%	0%	+1%	0%	0%
	Regular	0%	0%	0%	0%	-1%	0%	0%	-1%	-1%	-2%	-1%	+1%
B14	Diesel	0%	0%	0%	0%	-1%	0%	0%	0%	0%	0%	0%	0%
	Regular	0%	0%	0%	0%	0%	0%	0%	0%	0%	-1%	-1%	0%
B2	Diesel	+1%	0%	-1%	0%	+1%	+1%	0%	0%	0%	0%	0%	0%
	Regular	0%	0%	0%	0%	+1%	0%	0%	0%	0%	+1%	+2%	+4%
B3	Diesel	0%	0%	0%	0%	+1%	+1%	0%	+1%	+1%	0%	0%	0%
	Regular	0%	0%	0%	0%	0%	-1%	0%	0%	+2%	0%	+3%	-4%
B4	Diesel	0%	0%	0%	0%	+1%	0%	+1%	0%	+1%	0%	0%	0%
	Regular	0%	0%	0%	0%	-1%	0%	-1%	0%	0%	+1%	0%	-1%
B5	Diesel	0%	-1%	0%	0%	0%	0%	-1%	+1%	0%	0%	0%	0%
	Regular	0%	0%	0%	0%	0%	0%	0%	0%	0%	-3%	0%	-3%
B6	Diesel	0%	0%	0%	0%	0%	0%	+1%	0%	0%	0%	0%	0%
	Regular	0%	0%	0%	0%	0%	0%	0%	0%	0%	-1%	+1%	0%

TABLE .1.2 – Différences (% simulé - % original) par marque, puissance et carburant

.2 Annexe 2

Résultats pour le réseau $\hat{x}_0(x_t, t)$

Structure	Entraînement	Validation
1028	7.71×10^{-3}	7.73×10^{-3}
1028 × 512	7.70×10^{-3}	7.73×10^{-3}
1028 × 512 × 256	7.64×10^{-3}	7.70×10^{-3}
1028 × 512 × 256 × 128	7.67×10^{-3}	7.72×10^{-3}
512 × 256 × 128 × 64	7.69×10^{-3}	7.73×10^{-3}
1024 × 1024 × 512 × 256	7.66×10^{-3}	7.72×10^{-3}
P 256×128×64×32	7.70×10^{-3}	7.71×10^{-3}
P 256×128×64	7.68×10^{-3}	7.71×10^{-3}
P 512×256×128	7.67×10^{-3}	7.70×10^{-3}
P 512×256×128×64	7.66×10^{-3}	7.71×10^{-3}

TABLE .2.1 – Erreurs d'entraînement et de validation pour différentes architectures du réseau $\hat{x}_0(x_t, t)$. "P" indique l'utilisation de branches parallèles.

Résultats pour le réseau $\epsilon(x_t, t)$

Structure	Entraînement	Validation
P 256 × 128 × 64	0.434	0.435
P 128 × 64	0.435	0.435
1024×512×256×128×64	0.432	0.434
1024×512×256×128	0.434	0.435
1024×512×256×128×64×32	0.435	0.436
512×256×128×64	0.436	0.436

TABLE .2.2 – Erreurs d'entraînement et de validation pour différentes architectures du réseau $\epsilon(x_t, t)$. "P" indique l'utilisation de branches parallèles.

Bibliographie

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan et S. Ganguli, « Deep unsupervised learning using nonequilibrium thermodynamics, » in *International conference on machine learning*, pmlr, 2015, p. 2256-2265.

[2] M. Denuit et J. Trufin, « Effective statistical learning methods for actuaries, » 2019.

[3] J. Ho, A. Jain et P. Abbeel, « Denoising diffusion probabilistic models, » *Advances in neural information processing systems*, t. 33, p. 6840-6851, 2020.

[4] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré et M. Welling, « Argmax flows and multinomial diffusion : Learning categorical distributions, » *Advances in neural information processing systems*, t. 34, p. 12 454-12 465, 2021.

[5] D. Foster, *Generative deep learning*. " O'Reilly Media, Inc.", 2022.

[6] S. Zheng et N. Charoenphakdee, « Diffusion models for missing value imputation in tabular data, » *arXiv preprint arXiv :2210.17128*, 2022.

[7] A. Kotelnikov, D. Baranchuk, I. Rubachev et A. Babenko, « Tabddpm : Modelling tabular data with diffusion models, » p. 17 564-17 579, 2023.

[8] J. Shi, M. Xu, H. Hua, H. Zhang, S. Ermon et J. Leskovec, « TabDiff : a unified diffusion model for multi-modal tabular data generation, » in *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

[9] J. Soch et al., *Statproofbook/statproofbook. github. io : The book of statistical proofs (version 2023)*, 2024.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences
Place des sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/sc**