Lois uniformes et normales de chaînes discrètes
Uniform and normal laws of discrete strings

Sébastien REBECCHI, Jean-Michel JOLION

Titre	Lois uniformes et normales de chaînes discrètes
Date	Mars-Avril-Mai 2007
Statut	**Rapport de recherche**
Mots clef	Chaînes discrètes, loi uniforme, loi normale, estimation de paramètres

Auteurs	Sébastien REBECCHI	Jean-Michel JOLION
Adresse	Laboratoire d'InfoRmatique en Images et Systèmes d'information INSA de Lyon, bâtiment Jules Verne 20, avenue Albert Einstein 69621 Villeurbanne cedex France	
Téléphones	(33) (0) 4 72 43 60 89	(33) (0) 4 72 43 87 59
Faxs	(33) (0) 4 72 43 71 17	(33) (0) 4 72 43 80 97
e-mails	sebastien.rebecchi@liris.cnrs.fr	jean-michel.jolion@liris.cnrs.fr

Table des matières

Résumé

Dans ce rapport sont proposées les extensions des lois uniformes et normales aux espaces de chaînes discrètes. Ces notions permettent la manipulation statistique d'ensembles de chaînes discrètes, via des algorithmes génératifs et prédictifs. Nous proposons également une nouvelle méthode d'estimation de paramètres de distributions de chaînes discrètes, appliquée dans ce rapport à l'estimation d'écart-type de la loi normale, et donnant des résultats encourageants. Ce rapport n'est qu'une première avancée dans l'objectif de reconnaissance statistique de formes structurées.

Mots clef

Chaînes discrètes, loi uniforme, loi normale, estimation de paramètres.

Abstract

In this report are proposed the extensions of the uniform and normal laws to spaces of discrete strings. These concepts allow statistical handling of sets of discrete strings, via generative and predictive algorithms. We also propose a new method of estimation of parameters of distributions of discrete strings, applied in this report to the estimation of standard deviation of the normal law, and giving encouraging results. This report is only one first advanced in the objective of statistical recognition of structured patterns.

Keywords

Discrete strings, uniform law, normal law, parameters estimation.

1 Introduction

1.1 Contexte et motivations

Depuis quelques années, plusieurs équipes de recherche tentent de fusionner les deux approches principales de la reconnaissance de formes, à savoir l'approche statistique et l'approche structurelle. Ce choix est motivé par l'envie de pouvoir à la fois profiter des avantages indéniables des deux méthodes tout en se détachant de leurs inconvénients respectifs.

La reconnaissance de formes statistique se base sur un codage des données sous forme de vecteurs numériques, incapable bien souvent de reproduire fidèlement la complexité des données brutes (images...). Cependant ce choix est justifié par la palette très large des outils et algorithmes statistiques parus dans la littératures et reconnus comme performant pour la manipulation de données numériques.

Le problème est exactement l'inverse pour la reconnaissance de formes structurelle, où la partie codage est très riche car s'appuyant sur des structures de données de grande expressivité (graphes, chaînes, arbres...), permettant notamment de représenter de manière adéquate toute sorte de relations intra/inter formes (réflexivité, séquentialité, hiérarchie...). Par contre, les outils liés à la manipulation de structures sont souvent bien trop restrictifs (isomorphisme de graphes, distance d'édition [Lev66, WF74]...) et pas assez robustes pour les applications spécifiques à la reconnaissance de formes. Une autre limitation vient de l'absence de formalisme structurel pour le traitement d'ensembles de données, en ce sens où les outils classiquement utilisés sont le plus souvent basés sur des opérateurs seulement unaires ou binaires. Enfin, l'association entre d'une part la taille des structures de données utilisées, et d'autre part la complexité des algorithmes mis en œuvre, tend à rejeter cette approche pour le traitement de grands volumes de données.

Pour pouvoir réconcilier ces deux approches, il est au préalable nécessaire de définir une caractérisation statistique des espaces de structures discrètes. Le premier concept statistique proposé dans cette optique est celui de médiane d'ensemble de chaînes [Koh85, JBC04], généralisé dans [Jol03] par la notion de moments d'ensemble de chaînes, et généralisé dans [JMB01] aux ensembles de graphes. Ces études traduisent une volonté d'approche directe du problème, en opposition avec l'approche indirecte qui consiste à transformer implicitement la représentation structurelle des

données, *via* l'utilisation de fonctions noyaux [Vap98], dans le but de caractériser statistiquement les ensembles de données dans l'espace vectoriel numérique implicite associé au noyau utilisé. Ces méthodes indirectes permettent certes d'utiliser certains outils statistiques dans l'espace implicite, mais restent fortement limitées, et induisent, de plus, une nouvelle perte d'information sur les données brutes, due à une seconde phase de transformation de celles-ci, en plus de la phase initiale de transformation sous forme structurelle. Un exemple d'une telle approche, s'appuyant sur des noyaux basés sur des distances d'édition à des chaînes de références, est proposé dans [NB06]. Cet exemple illustre assez bien le problème de l'approche indirecte, à savoir que dans ce cas la seule information utilisée pour chaque chaîne X est sa distance d'édition aux chaînes choisies initialement comme références, les chemins d'éditions permettant de transformer X en les chaînes de références étant quant à eux totalement ignorés. Or ces chemins représentent un gros volume d'information utile, mais malheureusement incompatible avec la représentation vectorielle numérique imposée par le fait qu'un noyau est avant tout un produit scalaire.

L'objectif de notre travail est de contribuer à la caractérisation statistique "directe" d'ensembles de structures discrètes, plus particulièrement par la notion générale de distribution statistique. Nous nous focalisons dans ce rapport sur les lois uniformes et normales de structures de type chaîne. Après avoir posé les notations utilisées tout au long de ce rapport, nous faisons un bref rappel des descripteurs statistiques, ou apparentés comme tels, proposés dans la littérature pour caractériser des ensembles de chaînes discrètes (section 2). Puis nous introduisons nos définitions des lois uniformes et normales de chaînes discrètes (section 3), munies toutes deux d'algorithmes permettant de générer des chaînes sous contrôle statistique, et de prédire la probabilité de toute chaîne. Enfin nous proposons l'application d'une méthode de recherche d'estimation d'écart-type de la loi normale, basée sur la minimisation de la distance entre la loi proposée comme estimation et un échantillon généré sous contrôle statistique de la loi à estimer (section 4).

1.2 Notations et définitions préliminaires

Soit A un alphabet fini, *i.e.* un ensemble fini non vide, dont les éléments sont appelés lettres[1]. Une chaîne (définie) sur A est une séquence de lettres de A. Notons :

- $|X|$ la longueur de la chaîne X
- X_i la i-ème lettre de X
- $|A|$ le nombre de lettres de A
- A^n l'ensemble des chaînes de longueur n sur A
- A^* l'ensemble des chaînes de longueur finie sur A
- λ la lettre ou la chaîne vide ($|\lambda| = 0$)

Pour remarque, une chaîne définie sur A est définie sur tout sur-alphabet de A (alphabet sur-ensemble de A), et λ est défini sur tous les alphabets.

Une sous-séquence X' d'une chaîne X est une chaîne obtenue en supprimant des lettres à X, tout en préservant l'ordre relatif dans X des lettres restantes : $X' = X_{i_1} \dots X_{i_m}$, avec $i_1 < \dots < i_m$ et $m \leqslant |X|$. Si $i_{j+1} = i_j + 1$ pour tout $j \in \{1, \dots, m-1\}$, alors X' est une sous-chaîne de X. Si en plus $i_1 = X_1$ (resp. $i_m = X_{|X|}$), alors X' est le préfixe (resp. suffixe) de longueur m de X.

Nous supposons l'existence de l'opérateur de concaténation entre chaînes ou lettres (chaînes de longueur 1), noté par un point, souvent omis par souci d'allégement des notations, et défini comme suit : pour tout couple (X, Y) de chaînes, $X.Y =_{\text{notation}} XY = Z$ *ssi* $Z = X_1 \dots X_{|X|} Y_1 \dots Y_{|Y|}$. Dès lors Z est défini sur l'alphabet formé de l'union des alphabets sur lesquels sont respectivement définis X et Y, et la longueur de Z est égale à la somme des longueurs respectives de X et Y. Pour remarque, $\lambda X = X\lambda = X$ pour toute chaîne X.

Une opération d'édition sur A est un couple (a, b) de lettres de $A \cup \{\lambda\}$, noté $a \to b$. Nous disons qu'une telle opération transforme a en b, ou consomme a pour produire b. Si $a = \lambda$ (resp. $b = \lambda$), il s'agit de l'insertion de b (resp. la suppression de a), et si $a \neq \lambda$ et $b \neq \lambda$, il s'agit de la substitution de a par b.

Soit $c()$ une fonction de coût d'édition sur A : $\forall (a, b) \in (A \cup \{\lambda\})^2,\ a \neq b$:

[1] La notion de lettre est à comprendre au sens large, et nous l'utilisons aussi bien pour désigner un élément de type symbolique ou numérique

- $c(a \rightarrow a) = 0$
- $c(a \rightarrow b) > 0$

Une chaîne d'édition (ou chemin d'édition) sur A est une chaîne d'opérations d'édition sur A. Soit $Z = X_1 \rightarrow Y_1 \dots X_n \rightarrow Y_n$ une telle chaîne. Nous disons que Z consomme (resp. produit) la chaîne formée de la concaténation des lettres consommées (resp. produites) par ses opérations, *i.e.* $X = X_1 \dots X_n$ (resp. $Y = Y_1 \dots Y_n$). Le coût de Z par (extension de) $c()$ est défini comme la somme des coûts de ses opérations d'édition :

$$c(Z) = \sum_{i=1}^{|Z|} c(Z_i)$$

La distance d'édition (de Levenshtein) associée à $c()$, $d(X, Y)$, entre deux chaînes X et Y sur A, est définie comme le coût minimal d'une chaîne d'édition transformant X en Y [Lev66] :

$$d(X, Y) = \min\{c(Z) | Z \text{ est une chaîne d'édition transformant } X \text{ en } Y\}$$

Une chaîne d'édition est dite optimale *ssi* son coût est égal à la distance d'édition entre sa chaîne consommée et sa chaîne produite. L'algorithme classiquement utilisé pour calculer la distance d'édition entre deux chaînes X et Y, et le chemin d'édition optimal associé, repose sur une méthode de programmation dynamique de complexité temporelle et spatiale $O(|X| \times |Y|)$, proposée initialement dans [WF74]. Il est à noter qu'il existe généralement plus d'un chemin optimal entre X et Y.

2 Descripteurs statistiques d'ensembles de chaînes discrètes

2.1 Moyenne

Le terme "moyenne" est souvent ambigu de part son association à plusieurs concepts. Il peut être utilisé pour désigner, entre autres, celui de moyenne arithmétique d'un ensemble de données, ou celui d'espérance mathématique (moyenne théorique) d'une loi de probabilité.

Le seul concept clairement traduit de manière à pouvoir être utilisé dans le domaine des chaînes est celui de moyenne pondérée de deux chaînes, proposé dans [BJAK02], et défini comme suit : une moyenne pondérée de deux chaînes X et Y est une chaîne Z vérifiant cette équation : $d(X, Z) + d(Z, Y) = d(X, Y)$. Dans ce cas, plus $d(X, Z)$ est grand (resp. petit), et plus $d(Z, Y)$ est petit (resp. grand), et alors plus le poids de Y (resp. X) est important dans Z. La méthode de construction de Z peut être résumée de la manière suivante :

1. calculer un chemin d'édition optimal O transformant X en Y ($c(O) = d(X, Y)$)
2. choisir une sous-séquence O' de O
3. appliquer O' à X (en veillant à respecter les positions des opérations de O' dans O), pour produire Z

Dans ce cas, le poids de X dans Z est égal à $[d(X, Y) - c(O')]/d(X, Y)$, et celui de Y égal à $c(O')/d(X, Y)$.

Cependant, le concept qui nous intéresse ici est celui d'espérance mathématique (ou simplement espérance) d'une distribution. Il n'existe pour le moment aucune procédure permettant de calculer cette moyenne dans le domaine des chaînes, car ceci nécessite au préalable de préciser ce que sont les probabilités et donc la notion de distribution dans ce domaine, ce qui est l'objet de cette étude.

2.2 Médiane

Sous acceptation d'une analogie géométrique, la médiane d'un ensemble fini S de chaînes est définie dans [Koh85] comme la chaîne de A^* minimisant la somme de ses distances d'édition aux chaînes de S :

$$\text{médiane}(S) = \arg\min_{P \in A^*} \sum_{Q \in S} d(P, Q)$$

Cette approche se substitue le plus souvent à l'approche probabiliste en l'absence de notion de distribution. Pour remarque, toujours selon l'analogie géométrique, on obtient une définition équivalente de la chaîne moyenne comme étant celle minimisant la somme des carrés de ses distances aux chaînes de S. Cependant certains préfèrent toujours employer le terme de chaîne médiane, avec simplement un changement de distance, passant de la distance d'édition traditionnelle de Levenshtein à une distance correspondant au carré de cette dernière, et semblant donner de légèrement meilleurs résultats en terme de classification [MJC01].

Dans tous les cas, le calcul d'une telle médiane se heurte à de plus ou moins sévères problèmes. D'une part, il a été prouvé comme étant un problème NP-difficile [dlHC00, SP03], et d'autre part, il nécessite le recours à des calculs de distances entre chaînes de A^* et donc à la mise en place de la fonction de coût $c()$ sur A.

Pour réduire la complexité des calculs, des méthodes de résolution s'appuyant sur une réduction de l'espace de recherche (l'espace complet étant A^*) ont été proposées, avec plus ou moins de succès : recherche gloutonne [CdA97, Kru99], locale [MHJC00, Koh85], dynamique [JABC03], génétique [JBC04].

Pour ce qui est de la mise en place de $c()$, il existe trois solutions. La première consiste à fixer le coût de transformation d'une lettre en une autre (différente) à une constante, généralement 1, cette méthode étant utilisée dès qu'il n'y a aucune connaissance *a priori* quant à la dissimilarité entre les lettres. La deuxième solution consiste à s'appuyer cette fois-ci sur une connaissance *a priori* relative à la dissimilarité entre les lettres, connaissance pouvant être fournie par un expert du domaine, ou simplement déduite de manière *ad hoc* (distance "géographique" sur un clavier de caractères, alphabet numérique...). Enfin, la dernière solution est celle de l'apprentissage automatique de $c()$, comme effectué dans [OS06].

2.3 Écart-type

L'écart-type d'une distribution L est défini comme l'espérance des écarts à l'espérance de L. Pour le cas spécifique des chaînes, un écart à l'espérance est traduit par une chaîne d'édition consommant l'espérance (chaîne définie sur l'alphabet $(A \cup \{\lambda\})^2$). Comme nous l'avons vu, le concept de chaîne espérance n'est pas encore utilisable, et il est par conséquent d'usage d'approximer l'écart-type d'un ensemble S de chaînes par la médiane des écarts à sa médiane, toujours sous acceptation d'une analogie géométrique, et tel qu'initialement proposé dans [Jol03]. Cependant, même dans les situtations où l'analogie géométrique pourrait être supposée valable, l'approximation de l'espérance par la médiane d'un ensemble n'est pertinente que dans le cas où l'ensemble est supposé être un échantillon généré sous contrôle d'une distribution symétrique et unimodale, ou du moins une distribution ayant une espérance supposée très proche de sa médiane.

3 Lois statistiques de chaînes discrètes

3.1 Loi uniforme

La loi uniforme sur un ensemble fini S d'éléments est celle qui associe la même probabilité $|S|^{-1}$ à tout élément de S. Nous proposons ici une définition de la loi uniforme sur l'ensemble A^n, pour tout n donné.

Proposition 1 (Loi uniforme de chaînes)
Une chaîne X est dite réalisation d'une loi uniforme sur A^n ssi chacune de ses lettres est indépendamment réalisation d'une loi uniforme sur A.

Preuve
La probabilité de toute lettre de X étant indépendamment fixée selon une loi uniforme sur A, nous avons :

$$\forall X_i, \mathrm{p}(X_i) = |A|^{-1}$$

Dès lors, nous avons :

$$\mathrm{p}(X) = \prod_{i=1}^{n} \mathrm{p}(X_i) = \prod_{i=1}^{n} |A|^{-1} = |A|^{-n}$$

Or $|A^n|^{-1} = (|A|^n)^{-1} = |A|^{-n}$.

Cette définition donne aisément lieu à la mise en place de processus génératifs et prédictifs, respectivement détaillés par les algorithmes 1 et 2. L'algorithme génératif applique simplement le processus itératif sous-entendu par la proposition 1, de complexité linéaire en le paramètre n.

Données : Un entier positif n
Résultat : Une chaîne X générée selon une loi uniforme sur A^n
début
  $X \longleftarrow \lambda$;
  pour $i \longleftarrow 1$ **à** n **faire**
    $l \longleftarrow$ tirage aléatoire uniforme d'une lettre de A;
    $X \longleftarrow X.l$;
  fin
  retourner X;
fin

Algorithme 1 – Génération d'une chaîne sous contrôle d'une loi uniforme

Données : Une chaîne X sur A, un entier positif n
Résultat : La probabilité de X suivant une loi uniforme sur A^n
début
  si $|X| = n$ **alors retourner** $|A|^{-n}$;
  retourner 0.0;
fin

Algorithme 2 – Prédiction de la probabilité d'une chaîne selon une loi uniforme

3.2 Loi normale

Il n'est pas inapproprié de rapprocher les deux concepts mathématiques de chaîne et de vecteur. En effet ceux-ci partagent un grand nombre de points communs, comme par exemple leur considération vis à vis des notions très importantes d'ordre et de duplication. Si l'on échange la position de deux lettres dans une chaîne, on obtient une chaîne différente, et il est possible d'avoir deux lettres égales à deux positions différentes d'une chaîne. Or ces deux faits se retrouvent dans le cas des coordonnées de vecteurs. La seule réelle différence entres chaîne et vecteur réside dans l'utilisation que l'on en fait. En effet, le concept de vecteur s'adresse plus classiquement aux espaces numériques, discrets ou continus, avec lesquels il est muni de la définition d'un grand nombre d'opérateurs particuliers (addition, soustraction, angle, produit scalaire, produit vectoriel...). D'un autre côté, en insérant autant de lettres λ que voulu à n'importe quelles positions où il est possible de le faire dans une chaîne X (impossible aux éventuelles positions supérieures à $|X|+1$), on obtient X. Suivant ce raisonnement, nous proposons de définir une loi normale de chaînes sur $A \cup \{\lambda\}$, et donc sur A, par une loi normale multivariée de vecteurs constitués de lettres de $A \cup \{\lambda\}$.

Or la loi normale multivariée est paramétrée par son espérance mathématique et sa matrice de covariance, et le concept de covariance nous est pour le moment inconnu dans l'espace des chaînes. Cependant, nous connaissons la traduction du concept d'écart-type dans ce domaine (*cf.* 2.3 : chaîne d'opérations d'édition consommant l'espérance). Dans le cas précis où est supposé connue l'unique donnée de l'écart-type (et par là même de l'espérance), et en supposant par conséquent l'indépendance deux-à-deux de chaque lettre composant une chaîne, nous proposons cette traduction du concept de loi normale au domaine des chaînes :

Proposition 2 (Loi normale de chaînes)
Une chaîne X est dite réalisation d'une loi normale sur A^*, d'écart-type $\sigma = M_1 \to D_1 \dots M_n \to D_n$ (d'espérance $M = M_1 \dots M_n$), ssi X est égale à la concaténation de n lettres X_i respectivement

et indépendamment réalisation d'une loi normale sur $A \cup \{\lambda\}$, d'écart-type $M_i \rightarrow D_i$ (d'espérance M_i).

Preuve
Soient $x_1, \ldots, x_n$, n variables aléatoires normales, respectivement d'espérance μ_i et écart-type σ_i, et indépendantes deux-à-deux. D'après la théorie statistique multivariée, nous savons que le vecteur $[x_1, \ldots, x_n]$ suit une loi normale (multivariée) d'espérance $[\mu_1, \ldots, \mu_n]$, et de matrice de covariance diagonale (i.e. une matrice d'indépendance) avec vecteur d'écart-type $[\sigma_1, \ldots, \sigma_n]$.

Néanmoins nous ne pouvons pas utiliser cette définition telle quelle. En effet, une fois donnés l'espérance et de l'écart-type d'une loi normale (univariée), sa densité de probabilité en tout point x dépend de la valeur de l'*écart* de x à l'espérance, écart étant traduit par exemple dans le domaine classique numérique par la valeur absolue de la soustraction de x à l'espérance. C'est ici que nous avons recours à la fonction de coût $c()$ sur A, $c(a \rightarrow b)$ traduisant effectivement une notion d'écart de b à a, en tant que coût de la tranformation de a en b ($(a, b) \in (A \cup \{\lambda\})^2$). Ce raisonnement nous permet ainsi d'affiner notre définition de la manière suivante :

Proposition 3 (Loi normale de chaînes sous considération d'une fonction de coût)
Une chaîne X est dite réalisation d'une loi normale sur A^*, d'écart-type $\sigma = M_1 \rightarrow D_1 \ldots M_n \rightarrow D_n$ (d'espérance $M = M_1 \ldots M_n$), ssi X est égale à la concaténation de n lettres X_i respectivement et indépendamment réalisation d'une distribution L_i, telle que la densité de probabilité de X_i selon L_i soit égale à la densité de probabilité de $c(M_i \rightarrow X_i)$ selon une loi normale sur l'ensemble des coûts d'édition de M_i par une lettre de $A \cup \{\lambda\}$, d'espérance $c(M_i \rightarrow M_i) = 0$ et d'écart-type $c(M_i \rightarrow D_i)$.

Par l'expression "loi normale sur l'ensemble des coûts d'édition", nous sous-entendons le recours à une distribution discrète, à savoir L_i, instanciée de manière à respecter un processus gaussien sur l'ensemble considéré. Dans notre cas cette discrétisation, car il s'agit de cela, est aisément productible. En effet, A (et donc $A \cup \{\lambda\}$) étant fini, L_i correspond simplement à la distribution associant une probabilité à toute lettre $l \in A \cup \{\lambda\}$ comme étant égale à la densité de probabilité de $c(M_i \rightarrow l)$ selon la loi normale associée, normalisée par la somme de ces densités sur l'ensemble des $c(M_i \rightarrow b)$, pour tout $b \in A \cup \{\lambda\}$.

Les processus génératifs et prédictifs sont respectivement détaillés par les algorithmes 3 et 4. L'algorithme génératif applique simplement le processus itératif sous-entendu par la proposition 3, de complexité linéaire en la longueur de la chaîne écart-type σ. Pour ce qui est de l'algorithme prédictif, la difficulté est autrement plus élevée. En effet, une chaîne X étant inchangée avec l'insertion d'autant de λ que voulu à n'importe quelles positions où il est possible de le faire, le nombre de possibilités de production de X est égal au nombre de combinaisons de $|X|$ objets parmi $|\sigma|$. Pour résoudre ce problème, nous avons recours à une méthode de programmation dynamique de complexité temporelle $O(|X| \times |\sigma|^3)$ et spatiale $O(|\sigma|)$. À chaque itération j de la boucle centrale principale, le tableau PP$[i]$ contient la probabilité de production du préfixe de longueur j de X, selon une loi normale d'écart-type le préfixe de longueur i de σ. L'initialisation est effectuée de manière à prendre en considération le fait qu'il est improbable que la première lettre de X soit produite par la loi normale d'écart-type σ_i avec $i > |\sigma| - |X| + 1$. En effet, dans ce cas, il est improbable de produire les $|X| - 1$ lettres restantes de $|X|$, différentes de λ, avec la succession d'exécutions unitaires de $|\sigma| - i < |X| - 1$ lois normales de lettres (utilisation de l'algorithme 6). La finalisation est effectuée de manière à tenir compte de toutes les possibilités de fin de production de X, *i.e.* les probabilités de fin de production de X pour chaque position de σ, eventuellement complétées par des successions de production de λ. Enfin la boucle i contenue dans la boucle j principale met à jour la probabilité de production de X_j en position i de σ, de manière à tenir compte de l'ensemble des possibilités de production de X_{j-1} en position k inférieure à i, eventuellement complétées par des successions de production de λ entre les positions supérieures à k et inférieures à i. Le cas spécial où $X = \lambda$ ne rentre pas dans ce schéma, et est donc traité à part.

3.3 Validation

Nous souhaitons pouvoir mesurer la qualité de nos modèles de lois présentés, en terme de contrôle statistique de génération d'ensembles de chaînes. Pour ce faire, nous mettons en place le

Données : Une chaîne d'édition $\sigma = M_1 \to D_1 \dots M_n \to D_n$ sur A
Résultat : Une chaîne X générée selon une loi normale sur A^*, d'écart-type σ
début
  $X \longleftarrow \lambda$;
  pour $i \longleftarrow 1$ **à** n **faire**
    $g \longleftarrow$ loi normale d'espérance 0 et d'écart-type $c(M_i \to D_i)$;
    $l \longleftarrow$ tirage aléatoire d'une lettre de $A \cup \{\lambda\}$ selon la distribution suivante :

$$\forall b \in A \cup \{\lambda\}, \mathrm{p}(b) = \frac{g\left(c(M_i \to b)\right)}{Z}$$

    avec :

$$Z = \sum_{b \in A \cup \{\lambda\}} g\left(c(M_i \to b)\right)$$

    coefficient de normalisation tel que :

$$\sum_{b \in A \cup \{\lambda\}} \mathrm{p}(b) = 1$$

    $X \longleftarrow X.l$;
  fin
  retourner X;
fin

Algorithme 3 – Génération d'une chaîne sous contrôle d'une loi normale

protocole expérimental suivant : des échantillons de chaînes sont créés *via* l'utilisation des algorithmes génératifs 1 et 3, et nous mesurons la qualité des algorithmes prédictifs 2 et 4 quant à la modélisation de la distribution sous-jacente à chaque échantillon.

Ces mesures de qualité de modélisation sont effectuées *via* des calculs de distance du χ^2. La distance du χ^2 permet de mesurer l'adéquation statistique entre un échantillon S de données et une distribution $\hat{L}$ proposée comme hypothèse de distribution génératrice de S. Plus cette distance est faible, et plus $\hat{L}$ est supposée statistiquement proche de la distribution L réellement sous-jacente à S. En théorie, la distance du χ^2 entre L et S tend vers 0 avec la taille de S, tandis que la distance entre toute $\hat{L} \neq L$ et S diverge avec la taille de S, ceci plus ou moins rapidement selon que $\hat{L}$ est statistiquement très ou peu éloignée de L. La distance du χ^2 est obtenue par comparaison des effectifs observés et théoriques (*i.e.* qui devraient être en théorie observés dans le cas où $L = \hat{L}$) de S regroupé au sein de k classes, l'union de ces classes devant constituer une partition du domaine d'existence de $\hat{L}$. Dans notre cas nous devons redéfinir ce que constituent ces classes pour l'espace des chaînes. Le processus de regroupement des chaînes d'un échantillon en $2 \leqslant k \leqslant |A|$ classes est effectué de la manière suivante :

1. A est indicé par l'intervalle $I = [1; |A|]$, ceci étant possible vu l'aspect fini de A. L'indexation en elle-même n'est pas importante et on peut tout à fait l'envisager de manière aléatoire ou par rapport à tout ordre applicable aux lettres de A.
2. I est divisé en k sous-intervalles de même taille (quotient de la division entière de $|A|$ par k), *modulo* le dernier intervalle, contenant éventuellement plus de lettres (quotient + reste de la division entière de $|A|$ par k).
3. toute chaîne X de longueur strictement positive est classée en fonction de l'indice i dans I de sa première lettre, lui-même étant associé à un indice j dans la division de I ($j \in \{1, \dots, k\}$), j définissant la classe de X. λ est toujours classée 1.

Ce processus est totalement neutre, *i.e.* indépendant à la fois de la distribution génératrice de l'échantillon et de la distribution proposée comme hypothèse. Il est aisément notable que l'espérance de toute classe j, selon la distribution proposée comme hypothèse, est donnée par la probabilité, selon cette même distribution, qu'une chaîne commence par une des lettres indicées j par le processus (exception faite de la classe 1 pour laquelle il faut ajouter la probabilité de la chaîne vide

Données : Une chaîne X sur A, une chaîne d'édition $\sigma = M_1 \to D_1 \dots M_n \to D_n$ sur A
Résultat : La probabilité de X suivant une loi normale sur A^*, d'écart-type σ
début
  // Cas spécial $X = \lambda$
  si $X = \lambda$ **alors retourner** p(λ) ; // algorithme 5
  // Initialisation : production de la première lettre de X
  pour $i \longleftarrow 1$ **à** $|\sigma| - |X| + 1$ **faire**
    // Production de X_1 à la position i de σ
    PP[i] $\longleftarrow$ p(X_1 à la position i) ; // algorithme 6
    // Complétion avec la production de λ en positions $1 \dots (i-1)$ de σ
    pour $l \longleftarrow 1$ **à** $i - 1$ **faire**
      PP[i] $\longleftarrow$ PP[i] $\times$ p(λ à la position l) ; // algorithme 6
    fin
  fin
  pour $i \longleftarrow |\sigma| - |X| + 2$ **à** $|\sigma|$ **faire**
    // Impossible de commencer la production à cette position i de σ
    PP[i] $\longleftarrow$ 0.0;
  fin
  // Itérations : production des autres lettres de X
  pour $j \longleftarrow 2$ **à** $|X|$ **faire**
    // Possibilités de production de X_j en position i de σ;
    pour $i \longleftarrow |\sigma|$ **à** 1 **faire**
      PP[i] $\longleftarrow$ 0 ; // initialisation
      pour $k \longleftarrow 1$ **à** $i - 1$ **faire**
        // Production de X_{j-1} à la position k de σ
        PLUS $\longleftarrow$ PP[k];
        // Complétion avec la production de λ en positions $(k+1) \dots (i-1)$;
        pour $l \longleftarrow k + 1$ **à** $i - 1$ **faire**
          PLUS $\longleftarrow$ PLUS $\times$ p(λ à la position l) ; // algorithme 6
        fin
        // Sommation des probabilités de production de X_{j-1}
        PP[i] $\longleftarrow$ PP[i] + PLUS;
      fin
      // Production de X_j en position i de σ;
      PP[i] $\longleftarrow$ PP[i] $\times$ p(X_j à la position i) ; // algorithme 6
    fin
  fin
  // Finalisation des possibilités de production de X
  $P \longleftarrow$ 0.0;
  pour $i \longleftarrow 1$ **à** $|\sigma|$ **faire**
    // La production de X s'est stoppée à cette position i
    PLUS $\longleftarrow$ PP[i];
    pour $l \longleftarrow i + 1$ **à** $|\sigma|$ **faire**
      // Complétion à droite avec la production de λ
      PLUS $\longleftarrow$ p(λ à la position l) ; // algorithme 6
    fin
    // Sommation des probabilités de production de X
    $P \longleftarrow P$ + PLUS;
  fin
  retourner P;
fin

Algorithme 4 – Prédiction de la probabilité d'une chaîne selon une loi normale

Données : Une chaîne d'édition $\sigma = M_1 \to D_1 \dots M_n \to D_n$ sur A
Résultat : La probabilité de λ suivant une loi normale sur A^*, d'écart-type σ
début
  $P \longleftarrow 1.0$;
  pour $i \longleftarrow 1$ **à** n **faire** $P \longleftarrow P \times$ p(λ à la position i) ; // algorithme 6
  retourner P;
fin

Algorithme 5 – Prédiction de la probabilité de λ selon une loi normale

Données : Une lettre $l \in A \cup \{\lambda\}$, une opération d'édition $o = m \to d$ sur A
Résultat : La probabilité de l suivant une loi normale sur $A \cup \{\lambda\}$, d'écart-type o
début
  $g \longleftarrow$ loi normale d'espérance 0 et d'écart-type $c(o)$;
  $p \longleftarrow$ probabilité de l selon la distribution suivante :

$$\forall b \in A \cup \{\lambda\}, \mathrm{p}(b) = \frac{g\left(c(m \to b)\right)}{Z}$$

  avec :

$$Z = \sum_{b \in A \cup \{\lambda\}} g\left(c(m \to b)\right)$$

  coefficient de normalisation tel que :

$$\sum_{b \in A \cup \{\lambda\}} \mathrm{p}(b) = 1$$

  retourner p;
fin

Algorithme 6 – Prédiction de la probabilité d'une lettre selon une loi normale

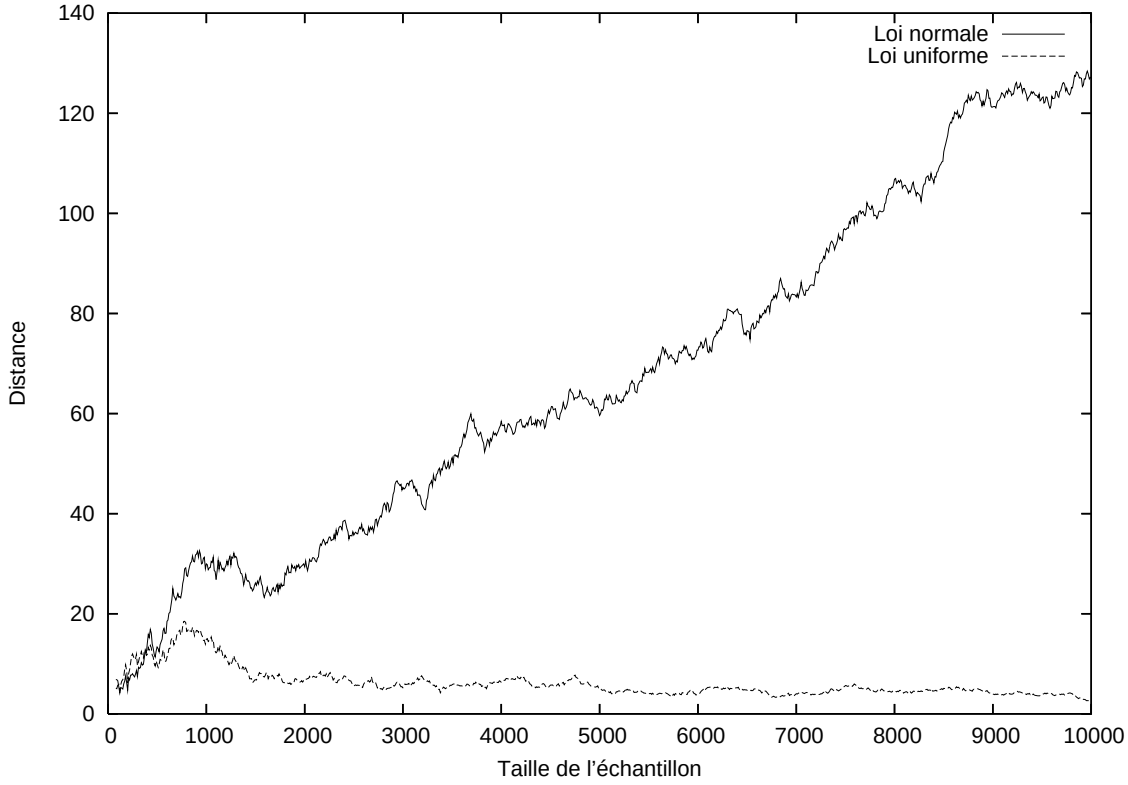


FIG. 1 – Évolution de la distance du χ^2 en fonction de la taille de l'échantillon généré selon une loi uniforme (algorithme 1)

λ).

Les figures 1 et 2 présentent les évolutions des qualités des modèles en fonction du nombre de chaînes générées (la qualité est décroissante en fonction de la croissance de la distance du χ^2). L'alphabet A et la fonction de coût d'édition $c()$ utilisés sont décrits en annexe, les chaînes générées sont de longueur 100, et le nombre de classes est fixé à 10.

Les résultats sont ceux attendus : le meilleur modèle est celui associé au processus génératif des données, et sa distance tends (lentement) vers 0 avec la taille de l'échantillon, tandis que celle de l'autre modèle diverge (rapidement). Il nous est donc effectivement possible de contrôler la distribution statistique d'un ensemble de chaînes.

4 Estimation d'écart-type d'une loi normale

La loi uniforme présentée en 3.1 étant non paramétrée, cette partie ne concerne effectivement que la loi normale présentée en 3.2.

Nous estimons l'écart-type d'une loi normale N par une chaîne d'édition $\hat{D}$ impliquant la minimisation de la distance du χ^2 entre la loi normale $\hat{N}$ d'écart-type $\hat{D}$ et un échantillon S de chaînes généré selon N. Il s'agit donc d'une méthode d'optimisation, devant être associée à un espace de recherche, celui-ci pouvant être par exemple l'ensemble des chemins d'édition optimaux consommant et produisant une chaîne de S (recherche simple), $((A \cup \{\lambda\})^2)^*$ (recherche complète), ou tout sous-ensemble particulier de $((A \cup \{\lambda\})^2)^*$ (recherche génétique, locale, gloutone, par colonie de fourmis...).

Deux types d'expérimentations sont menées, correspondant à deux niveaux d'estimation :

1. Estimation d'écart-type avec espérance connue
2. Estimation d'écart-type avec espérance inconnue

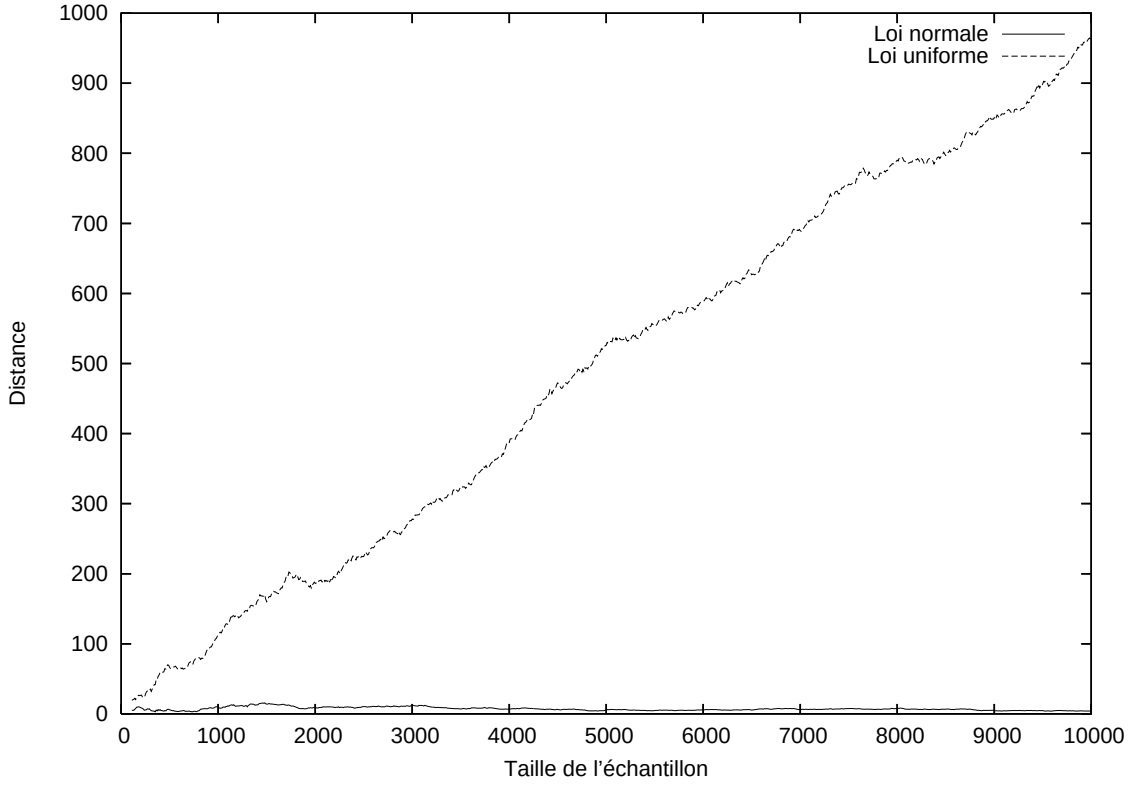


FIG. 2 – Évolution de la distance du χ^2 en fonction de la taille de l'échantillon généré selon une loi normale (algorithme 3)

Dans le premier cas seule la chaîne produite par l'écart-type est à estimer, la chaîne consommée étant bien entendue fixée à l'espérance. Par contre dans le deuxième cas la chaîne consommée par l'écart-type est également à estimer, constituant par là même une estimation de l'espérance. Dans toutes les expérimentations suivantes, la recherche de l'écart-type est limitée à l'ensemble des chemins d'édition optimaux consommant et produisant une chaîne de S (recherche simple), ce qui signifie que dans le deuxième cas, l'estimation de l'espérance est limitée à une recherche dans S.

La figure 3 présente les évolutions des qualités des modèles estimés en fonction du nombre de chaînes générées (la qualité est décroissante en fonction de la croissance de la distance du χ^2). Ces évolutions sont également comparées avec celles des qualités de ces deux modèles, utilisés ici comme repères :

- le modèle sous-jacent à la génération de l'échantillon (celui que l'on cherche à estimer)
- un modèle de loi normale dont l'écart-type a été généré totalement aléatoirement

L'alphabet A et la fonction de coût d'édition $c()$ utilisés sont décrits en annexe, les chaînes générées sont de longueur 100, et le nombre de classes est fixé à 10 (le processus de regroupement est le même que celui décrit en 3.3).

Les résultats sont positifs, à savoir qu'espérance connue ou inconnue, la loi estimée est au moins aussi proche de l'échantillon que celle l'ayant généré. Le fait qu'elle y soit même parfois plus proche peut paraître douteux, mais est simplement du au fait que notre méthode d'estimation s'appuie sur une optimisation de l'adéquation à l'échantillon, et que ce dernier est généré sous contrôle d'un processus aléatoire, pour lequel il ne peut donc pas assurer une parfaite représentativité, surtout lorsque sa taille est faible. En tous cas, nous pouvons affirmer que les estimations proposées sont très proches du paramètre réel, sans besoin de mesurer cette proximité par une quelconque distance d'édition entre chaînes d'édition, processus qui aurait nécessité l'instanciation et donc la justification d'une fonction de coût d'édition sur $(A \cup \{\lambda\})^2$. De plus cette méthode est totalement déterministe pour un échantillon donné. Enfin, l'estimation aléatoire divergeante témoigne de la difficulté de la tâche et donc également de la qualité des estimations fournies.

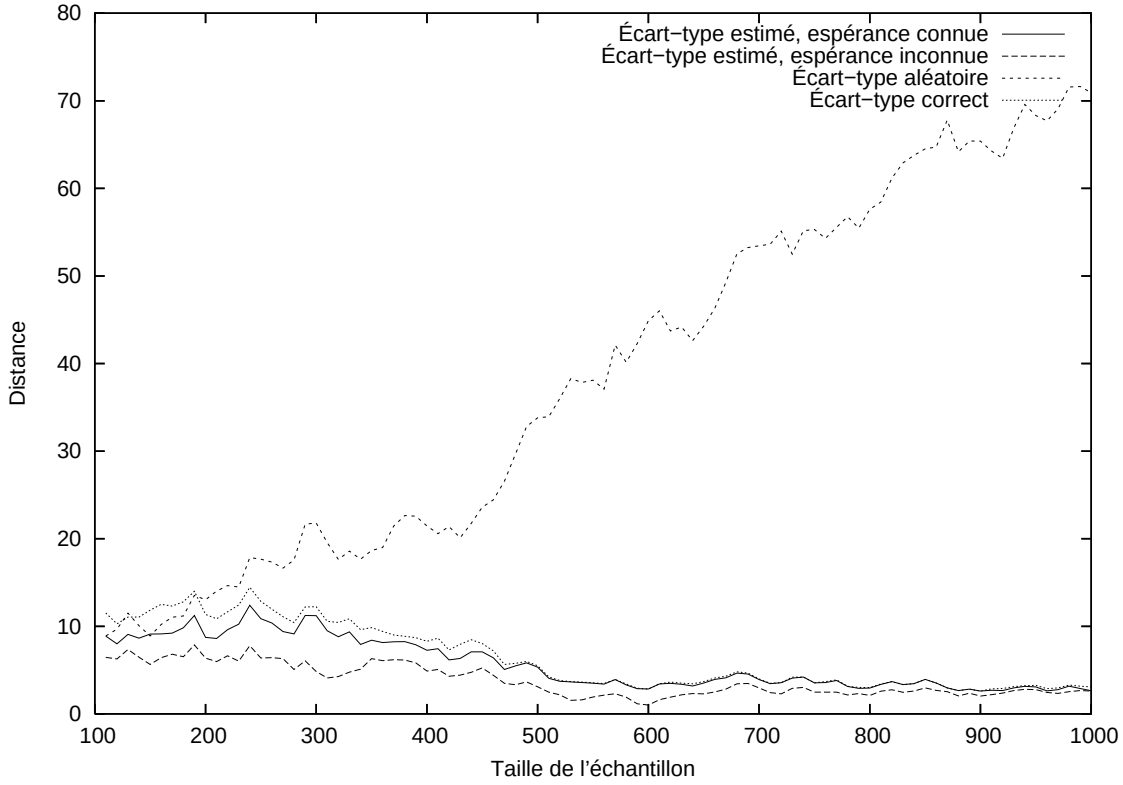


FIG. 3 – Évolution de la distance du χ^2 en fonction de la taille de l'échantillon

5 Conclusion

Nous avons proposé dans ce rapport le moyen de contrôler statistiquement la synthèse d'ensembles de chaînes discrètes par des lois normales ou uniformes, ainsi qu'une méthode d'estimation de paramètres de distributions de chaînes s'appuyant sur une recherche optimisant la distance d'un échantillon à la loi proposée. Cette méthode n'est pour le moment appliquée qu'à l'estimation de l'écart-type d'une loi normale, avec des résultats encourageants, et pourrait assez aisément être étendue à d'autres lois paramétrées, nécessitant la formalisation préalable de l'extension de ces lois au domaine des chaînes discrètes. Ce travail ne constitue donc qu'une première avancée et de nombreuses perspectives en découlent, parmi lesquelles :

1. la formalisation d'autres lois statistiques de chaînes discrètes
2. l'extension de la méthode d'estimation de paramètres aux autres lois paramétrées
3. l'estimation de la loi statistique génératrice d'un échantillon :
   (a) pour chaque loi non paramétrée : sélectionner cette loi comme estimation candidate
   (b) pour chaque loi paramétrée : chercher le(s) paramètre(s) minimisant la distance de la loi à l'échantillon et sélectionner cette loi, munis de ce(s) paramètre(s), comme estimation candidate
   (c) parmi toutes les lois sélectionnées en 3a et 3b, proposer comme estimation finale celle minimisant sa distance à l'échantillon
4. l'application à la classification de données réelles (images...) codées par des chaînes discrètes
5. la généralisation aux graphes discrets

U+	0	1	2	3	4	5	6	7
000	**[NUL]**	**[SOH]**	**[STX]**	**[ETX]**	**[EOT]**	**[ENQ]**	**[ACK]**	**[BEL]**
001	**[DLE]**	**[DC1]**	**[DC2]**	**[DC3]**	**[DC4]**	**[NAK]**	**[SYN]**	**[ETB]**
002	[SP]	!	”	#	$	%	&	’
003	0	1	2	3	4	5	6	7
004	@	A	B	C	D	E	F	G
005	P	Q	R	S	T	U	V	W
006	‘	a	b	c	d	e	f	g
007	p	q	r	s	t	u	v	w

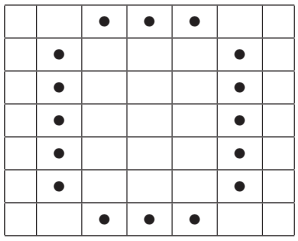
U+	8	9	A	B	C	D	E	F
000	**[BS]**	**[HT]**	**[LF]**	**[VT]**	**[FF]**	**[CR]**	**[SO]**	**[SI]**
001	**[CAN]**	**[EM]**	**[SUB]**	**[ESC]**	**[FS]**	**[GS]**	**[RS]**	**[US]**
002	(	)	*	+	,	-	.	/
003	8	9	:	;	<	=	>	?
004	H	I	J	K	L	M	N	O
005	X	Y	Z	[	\	]	^	_
006	h	i	j	k	l	m	n	o
007	x	y	z	{	|	}	~	**[DEL]**

Tab. 1 – Alphabet latin basique (ASCII), indicé selon la norme Unicode. Les caractères spéciaux (à ne pas interpréter tels quels) apparaissent entre crochets. Les caractères de contrôle C0 apparaissent en gras, et n’appartiennent pas à l’alphabet A utilisé pour les expérimentations

Annexe : alphabet et fonction de coût utilisés pour les expérimentations

L’alphabet A correspond à un sous-ensemble de l’alphabet latin basique (ou ASCII, intervalle [U+0000 ; U+007F] dans la norme Unicode), dans lequel ont été retirés les caractères de contrôle C0 (intervalle [U+0000 ; U+001F], ainsi que U+007F, dans la norme Unicode). Tous les caractères de A sont donc accessibles par n’importe quel clavier utilisant l’alphabet latin. A contient finalement 95 lettres de type caractère, exposées dans le tableau 1.

Quant à la fonction de coût $c()$, elle est purement visuelle : $\forall (a,b) \in A^2$, $c(a \to b) = c(b \to a)$, et il s’agit de la distance de Hamming [Ham50] entre les codages visuels respectifs de a et b selon une matrice binaire de dimensions 7×7. Un exemple de tel codage visuel est donné dans le tableau 2. Cette méthode de codage peut être appliquée à tout alphabet de lettres traduites sous forme de matrices binaires, comme c’est le cas par exemple pour des symboles codés par des matrices 3×3, proposé dans [SJ05] pour le codage d’images à base de contraste, et donnant de bons résultats en indexation et recherche dans des bases d’images naturelles. Les coûts des insertions et suppressions sont quant à eux tous fixés à $+\infty$, résultant en le fait que la probabilité de toute chaîne X selon une loi normale d’écart-type σ (voir 3.2) est nulle dans le cas où $|X| \neq |\sigma|$. Ce choix est motivé par le désir de forcer notre processus de génération de chaînes gaussiennes (algorithme 3) à ne produire que des chaînes de même longueur (fixée par celle de l’écart-type), ce qui est toujours le cas pour la loi uniforme (voir 3.1). De ce fait, les mesures d’adéquations entre échantillons et lois (voir 3.3) ne s’effectuent que sur des échantillons de chaînes de même longueur. Le contraire aurait été très pénalisant pour ce qui est de l’adéquation de la loi uniforme à l’échantillon gaussien, et les mesures n’auraient donc pas été d’une grande fiabilité. Ce choix implique également un aspect positif en terme de performance en prédiction de probabilité d’une chaîne selon la loi normale. Le processus général (algorithme 4) est dans ce cas simplifié à un processus itératif, détaillé dans l’algorithme 7, aligné sur celui de l’algorithme génératif (algorithme 3), et de complexité linéaire en la longueur de la chaîne écart-type σ. En effet, il est dans ce cas inutile de conserver la méthode dynamique proposée par l’algorithme général, car il n’y a qu’une seule combinaison possible de $|X| = |\sigma|$ objets parmi $|\sigma|$.



TAB. 2 – Codage visuel du caractère O par une matrice binaire de dimensions 7×7

Données : Une chaîne X sur A, une chaîne de substitutions $\sigma = M_1 \to D_1 \dots M_n \to D_n$ sur A
Résultat : La probabilité de X suivant une loi normale sur A^*, d'écart-type σ
début
  si $|X| = n$ **alors**
    $P \longleftarrow 1.0$;
    pour $i \longleftarrow 1$ **à** n **faire**
      $g \longleftarrow$ loi normale d'espérance 0 et d'écart-type $c(M_i \to D_i)$;
      $p \longleftarrow$ probabilité de X_i selon la distribution suivante :

$$\forall b \in A, \mathrm{p}(b) = \frac{g\left(c(M_i \to b)\right)}{Z}$$

      avec :

$$Z = \sum_{b \in A} g\left(c(M_i \to b)\right)$$

      coefficient de normalisation tel que :

$$\sum_{b \in A} \mathrm{p}(b) = 1$$

      $P \longleftarrow P \times p$;
    fin
    retourner P;
  fin
  retourner 0.0;
fin

Algorithme 7 – Prédiction de la probabilité d'une chaîne selon une loi normale, dans le cas spécifique où la fonction de coût interdit les insertions et suppressions

Références

[BJAK02] H. Bunke, X. Jiang, K. Abegglen, and A. Kandel. On the weighted mean of a pair of strings. *Pattern Analysis and Applications*, 5(1) :23–30, 2002.

[CdA97] F. Casacuberta and M. D. de Antonio. A greedy algorithm for computing approximate median strings. In *Proceedings of the VII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, pages 193–198, 1997.

[dlHC00] C. de la Higuera and F. Casacuberta. Topology of strings : median string is NP-complete. *Theorical Computer Science*, 230(1-2) :39–48, 2000.

[Ham50] R. W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2) :147–160, 1950.

[JABC03] X. Jiang, K. Abegglen, H. Bunke, and J. Csirik. Dynamic computation of generalised median strings. *Pattern Analysis and Applications*, 6(3) :185–193, 2003.

[JBC04] X. Jiang, H. Bunke, and J. Csirik. Median strings : a review. In *Data Mining in Time Series Databases*, pages 173–192. World Scientific, 2004.

[JMB01] X. Jiang, A. Münger, and H. Bunke. On median graphs : properties, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10) :1144–1151, 2001.

[Jol03] J.-M. Jolion. The deviation of a set of strings. *Pattern Analysis and Applications*, 6(3) :224–231, 2003.

[Koh85] T. Kohonen. Median strings. *Pattern Recognition Letters*, 3(5) :309–313, 1985.

[Kru99] F. Kruzslicz. Improved greedy algorithm for computing approximate median strings. *Acta Cybernetica*, 14(2) :331–340, 1999.

[Lev66] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8) :707–710, 1966.

[MHJC00] C. D. Martínez-Hinarejos, A. Juan, and F. Casacuberta. Use of median string for classification. In *ICPR*, pages 2903–2906, 2000.

[MJC01] C. D. Martínez, A. Juan, and F. Casacuberta. Improving classification using median string and NN rules. In *Proceedings of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, volume 2, pages 391–395, Benicàssim (Spain), 2001.

[NB06] M. Neuhaus and H. Bunke. Edit distance based kernel functions for structural pattern classification. *Pattern Recognition*, 39 :1852–1863, 2006.

[OS06] J. Oncina and M. Sebban. Learning stochastic edit distance : application in handwritten character recognition. *Pattern Recognition*, 39(9) :1575–1587, 2006.

[SJ05] I. Simand and J.-M. Jolion. Représentation d'images par chaînes de symboles : application à la recherche par le contenu, 2005. Actes du 20ème colloque GRETSI : Traitement du signal et des images, Louvain-la-Neuve, Belgique, Presses universitaires de Louvain, Diffusion universitaire CIACO (ISBN 2-87463-002-0), 2005, volume 2, pages 925-928.

[SP03] J. S. Sim and K. Park. The consensus string problem for a metric is NP-complete. *Journal of Discrete Algorithms*, 1(1) :111–117, 2003.

[Vap98] V. Vapnik. *Statistical learning theory*. Springer, 1998.

[WF74] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1) :168–173, 1974.