Génération de données binaires groupées à partitionnement contrôlé et évaluation de l'impact des méthodes de réduction de dimension sur ce partitionnement

Siwar JENDOUBI*, Ali BELLAMLIH MAMOU*
Aurélien BAELDE*

*Upskills R&D
16 Rue Marc Sangnier 94600 Choisy-le-roi, France
{siwar.jendoubi, aurélien.baelde}@upskills.ai
https://www.upskills.com/

Résumé. Les données binaires (deux valeurs possibles) sont utilisées dans plusieurs domaines de recherche tel que la modélisation des protéines en bio-informatique En particulier, certains problèmes impliquent des données binaires à partitionner. Un grand nombre de problèmes potentiellement solubles par apprentissage statistique ne peuvent l'être faute à la faible disponibilité des données réelles. Ce problème est encore plus visible dans le cas de l'apprentissage non supervisé, et notamment pour les tâches de partitionnement. D'où l'intérêt de pouvoir générer des données binaires dont le partitionnement est contrôlé, comme proposé dans cet article. Cet article détaille une méthode de génération de données binaires partitionnées, et présente de manière illustrative une comparaison de l'effet d'algorithmes de réduction de la dimension sur les caractéristiques du partitionnement généré.

1 Introduction

Les données binaires sont des données composées des vecteurs de taille arbitraire dont les modalités sont tirées dans une distribution à deux valeurs uniquement. Avec les modalités 0 ou 1, un vecteur de données binaires peut être 010111001010. Elles sont utilisées dans divers domaines d'application. Par exemple, un document de texte peut être modélisé par un vecteur binaire représentant la présence ou l'absence des termes ou de caractères. De plus, plusieurs techniques d'apprentissage automatiques ont été introduites pour l'exploitation de ces données particulières, Bouguila (2010); Wang et Kabán (2005); Ordonez (2003). Dans cet article, nous nous intéressons au problème de l'apprentissage non supervisé des données binaires. En effet, dans ce problème, les données ne sont pas labellisées et les connaissances préalables sont généralement trop faible pour être exploitables. De plus, les données réelles sont soit insuffisantes pour

valider un modèle donné, soit inexistantes. D'où l'intérêt de générer des jeux de données groupées en clusters contrôlés.

Dans cet article, un algorithme de génération des données binaires groupées par clusters de variances contrôlées est introduit. Cet algorithme permet la génération de jeux de données binaires ayant des caractéristiques connues et maîtrisées au préalable. Ces jeux de données permettent de tester et valider l'efficacité des solutions de partitionnement de données binaires. Pouvoir contrôler les caractéristiques des données générées permet aussi d'étudier l'applicabilité de ces méthodes et leurs faiblesses. Ces caractéristiques sont aussi utiles pour choisir une technique d'apprentissage automatique. En effet, le choix va être guidé par les caractéristiques des données et des tests dans des circonstances similaires aux cas réels. De plus, une étude comparant des algorithmes de réduction de dimension a été effectuée pour montrer l'utilité des données générées dans le choix d'un algorithme de réduction de dimension, sous l'angle de la conservation des caractéristiques des clusters dans l'espace réduit.

Les contributions principales de cet article sont les suivantes : 1) génération de vecteurs binaires artificiels, et groupés par clusters de variances contrôlés, 2) évaluation des données générées et leur contrôlabilité, 3) évaluation de la séparabilité des clusters, 4) évaluation des données générées à travers une étude expérimentale permettant le choix d'un algorithme de réduction de dimension qui conserve le mieux les clusters. La réduction de la dimension est considérée dans cet article comme solution pour la réduction du temps de partionnement des données.

La suite de l'article est organisée comme suit : La Section 2 présente une discussion de l'état de l'art. La Section 3 introduit l'algorithme de génération de données et évalue ses paramètres et son efficacité. La Section 4 montre une étude de cas des données générées à travers la comparaison de l'impact des méthodes de réduction de la dimension sur les clusters générés. Finalement, la Section 5 conclut l'article.

2 État de l'art

Plusieurs algorithmes de génération des données binaires, Jiang et al. (2020); Lunn et Davies (1998), visent à simuler des phénomènes de corrélation entre les variables binaires. Ces travaux cherchent à simuler un phénomène observé dans plusieurs domaines de recherche tel que la corrélation entre les variables génétiques en bioinformatique. Plusieurs algorithmes de génération de modalités binaires corrélées avec des probabilités marginales variées ont été introduits, Jiang et al. (2020); Lunn et Davies (1998). L'objectif principale de ces algorithmes est de pouvoir générer des données binaires à hautes dmension et avec des variables présentant des corrélations maitrisées entre elles. Ces algorithmes de génération de données sont adaptés pour l'étude de la corrélation des données. Cependant, les données générées avec ces solutions ne sont pas adaptées pour l'étude des approches de partitionnement des données.

Des méthodes de génération des données binaires corrélées longitudinalement ont été étudié par Farrell et Rogers-Stewart (2008). Ces méthodes sont utiles pour simuler des enquêtes empiriques importantes donnant lieu à des données binaires longitudinales corrélées où certains résultats dichotomiques sont mesurés sur les mêmes unités expérimentales au fil du temps. La génération de ces données considère l'aspect "clus-

ter" dans le processus de génération. Un cluster est définit par le fait que les mesures au sein de chaque cluster sont corrélées, mais les mesures de différents clusters sont indépendantes. Ces méthodes de génération sont différentes de notre solution. En effet, ces approches ne maîtrisent pas les distances intra et inter-clusters mais plutôt la corrélation des données dans chaque cluster.

Des données binaires synthétiques ont été utilisé pour expérimenter et valider des algorithmes de clustering comme le travail de Ordonez (2003). Cependant, la méthode de génération de données binaires utilisées n'a permis que la génération des données à haute dimension, mais ces données ne présentent pas de clusters détectables. Ainsi, ces jeux de données ont permis juste l'étude de la scalabilité de l'algorithme de clustering proposé, d'où l'intérêt de l'lagorithme introduit dans la section suivante.

3 Génération de vecteurs binaires artificiels, et groupés par clusters de variance contrôlés

Cette section explique l'algorithme de génération des données binaires (section 3.1) proposé et évalue la séparabilité des clusters générés (section 3.2).

3.1 Algorithme de génération des données

Cette section introduit l'algorithme de génération de données et l'ensemble des paramètres permettant d'en contrôler la sortie. Les valeurs des modalités binaires considérées sont soit 1 soit 0, mais d'autres couples de valeurs sont également possibles. On désigne par bit une modalité qui a exactement deux valeurs possible : 1 ou 0.

```
1 Inputs : C : Nombre de clusters. Nc : Nc = (n_1, n_2, ..., n_m) est la liste des tailles des
    clusters. D : Nombre de descripteurs. Tm : Taux de modification des bits du vecteur généré.
2 pour chaque cluster i dans C faire
3 |   Générer aléatoirement un prototype de dimension D dont les valeurs sont tirées dans
  |     une distribution de Bernouilli d'espérance 0.5 Ber(0.5).
4 |   pour chaque vecteur j dans Nc_i faire
5 |   |   Copier le prototype généré et échangé l'état de Tm % des bits aléatoirement choisis
6 |   fin
7 fin
```

Algorithme 1 : *Algorithme de génération des données binaires groupées en clusters de variance contrôlée*

Algorithme 1 introduit l'algorithme de génération des données proposé. Le taux de bits échangés, Tm, contrôle la proximité des vecteurs dans leur clusters (distance intra-cluster), tandis que la distance moyenne entre clusters (distance inter-cluster) est définie par les prototypes. Dans cet algorithme, Tm est identique pour tous les clusters. Cela signifie que la distance intra-cluster est en moyenne identique pour chaque cluster. Un nombre C de vecteurs binaires sont aléatoirement générés et sont utilisés comme les « prototypes » ou « centre » des futurs clusters. Pour chaque cluster i, Nc_i vecteurs sont générés en dupliquant le vecteur prototype, puis en échangeant la valeur d'une fraction T_m arbitrairement définie de bits choisis aléatoirement.

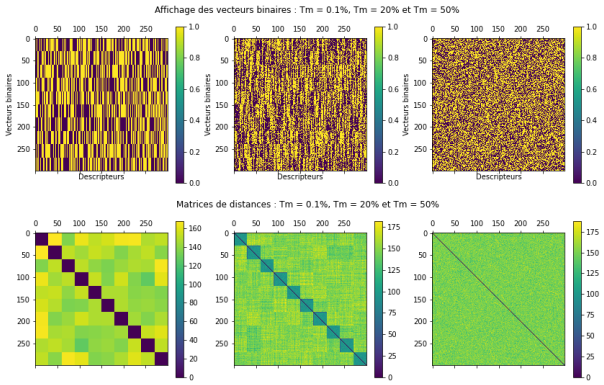

FIG. 1 – *Affichage des vecteurs binaires générés et les matrices de distances correspondantes pour trois valeurs de Tm : 0.1%, 20% et 50%. $C = 10$, $D = 300$ et $Nc = 30$.*

3.2 Évaluation de la séparation des clusters générés

L'algorithme de génération de données binaires introduit dans la section 3.1 est évalué dans cette section. En effet, afin d'utiliser ces données générées avec confiance, il est important de contrôler la qualité de leur génération. Ce contrôle de qualité est définit à travers le contrôle de la distance entre les vecteurs dans un même cluster (distance intra-cluster), et la distance entre clusters (distance inter-cluster). L'ensemble des vecteurs générés est appelé E. Cet ensemble est partitionné en C clusters dont le ième est nommé C_i.

La figure 1 présente une visualisation de trois ensembles de données et des matrices de distances correspondantes avec trois taux de modification de bits $Tm = 0.1\%$, $Tm = 20\%$ et $Tm = 50\%$ respectivement. La matrice des distances consiste à calculer la distance entre chaque couple de vecteur (x, y). La distance retenue dans cet article est la distance de Manhattan : $Man(x, y) = \sum_{i=1}^{D} |x_i - y_i|$. Les vecteurs ont une dimension $D = 300$ et 10 clusters contenant 30 vecteurs sont générés. Lorsque le taux de bits modifiés est faible ($Tm = 0.1\%$), les clusters sont aisément visibles et séparables. En cohérence, la matrice des distances montre que les vecteurs de données appartenant aux mêmes clusters sont proches. Si le taux de modification de bits augmente (et $Tm = 20\%$) les clusters restent visibles mais la distance moyenne intra-cluster augmente, menant à des vecteurs nettement plus différents les uns des autres. Avec un Tm de 50%, il n'y a plus de clusters visibles dans les matrices. Tous les vecteurs sont aléatoires, la distance moyenne intra-cluster est égale à la distance moyenne inter-cluster : la notion de cluster n'a plus de sens dans ce cas. L'expérience illustrée par la figure 2 confirme ces observations. Cette expérience consiste à tracer les histogrammes des distances intra et inter-cluster en fonction du taux moyen de modification de bits. En effet, avec un faible Tm, l'histogramme des distances intra-cluster est localisé à 0, car les vecteurs sont extrêmement similaires les uns des autres. A mesure que le taux de modification de bits Tm augmente, l'histogramme des distance intra-cluster s'élargit et

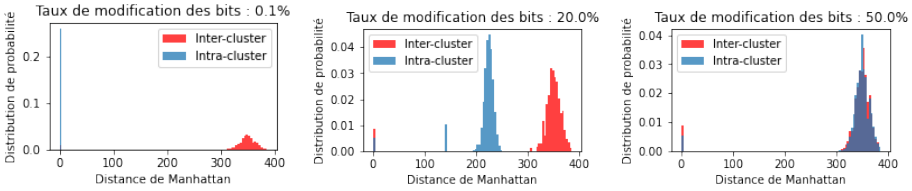

FIG. 2 – *Évolution des distributions des distances intra et inter-cluster en fonction du taux de modification des bits. $C = 30$, $D = 700$, et $Nc = 50$.*

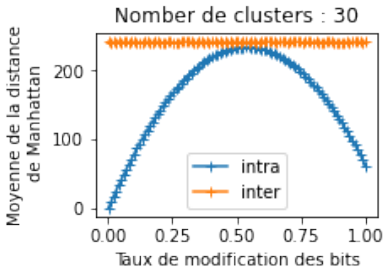

FIG. 3 – *Évolution de la distance moyenne intra et inter-cluster en fonction du taux de bits modifiés. $C = 30$, $Nc = 15$ et $D = 500$.*

se rapprochent de l'histogramme des distances inter-clusters. En effet, les clusters grossissent car le nombre de bits échangés croit. Pour un taux de modification de bits très élevé ($Tm = 50\%$), les deux histogrammes sont confondus : il n'existe plus de clusters, uniquement un nuage de points uniforme. Par ailleurs, on remarque que l'histogramme des distances inter-cluster ne dépend pas du taux Tm. En effet, ces distances sont fixées lors de génération des prototypes initiaux.

La figure 3 montre l'impact du taux de modification de bits Tm sur les distances moyennes intra et inter-cluster. En cohérence avec la figure 2, la distance moyenne inter-cluster est indépendante du taux de modification de bits Tm. De plus, la distance moyenne intra-cluster est croissante jusque $Tm = 50\%$ puis décroissante au delà. De plus, son maximum est presque confondu avec la valeur de la distance moyenne inter-cluster. Pour des Tm faibles, la distance intra-cluster est beaucoup plus faible que la distance inter-cluster : les clusters sont très compacts et éloignés les uns des autres. Lorsque le taux de modification de bits augmente, les clusters grossissent à distance inter-cluster constante. Lorsque $Tm = 50\%$, la distance intra-cluster atteint sont maximum, le rayon des clusters est proche de la distance inter-clusters, il n'y a donc plus de clusters définis. La valeur du maximum à $Tm = 50\%$ est lié au choix arbitraire de l'espérance de la loi de Bernoulli à 0.5 utilisée pour la génération des prototypes. Lorsque Tm continue à augmenter, la distance inter-cluster diminue car le potentiel d'échange de bits a été dépassé, et tout changement de bit supplémentaire conduit au vecteur

complémentaire du vecteur initial. Dans le cas où $Tm = 100\%$, tous les vecteurs sont égaux au complémentaire du prototype, sauf le prototype lui même, d'où le fait que la distance intra-cluster n'est pas strictement nulle.

4 Impact des méthodes de réduction de dimension sur la conservation des clusters générés

La section précédente présente l'algorithme de génération de données et met en lumière ses limites d'utilisation. Afin d'illustrer l'intérêt d'un tel algorithme de génération de données binaires, cette section étudie l'impact d'algorithmes de réduction de dimension sur la conservation des clusters dans l'espace réduit. En effet, l'objectif de la réduction de dimension dans cet article est de réduire les données pour minimiser leur temps de partitionnement sans perte en matière de distances entre clusters. Par conséquent, la question suivante se pose : Est-ce que réduire la dimension des vecteurs binaires modifie les clusters générés ? Pour répondre à cette question, trois méthodes de réduction des données sont considérées : NMF (Non-negative Matrix Factorization), Sra et Dhillon (2006), PCA (Principal Component Analysis), Ringnér (2008), et UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction), McInnes et al. (2018).

La figure 4 présente une visualisation des matrices de distances des données générées sans réduction et après réduction avec PCA, NMF, et UMAP respectivement. Trois valeurs de Tm ont été sélectionnées : 0.1%, 20% et 50%. Avec $Tm = 0.1\%$, la matrice des distances des données avant réduction de dimension montre des vecteurs organisés en clusters très bien définis. Après application de la réduction de la dimension avec PCA, NMF et UMAP vers un espace de dimension 100, les clusters restent visibles avec une petite perte de qualité. Ainsi, l'application d'une transformation linéaire ou non-linéaire à dimension constante ne modifie pas les clusters générés. Avec un $Tm = 50\%$, aucun cluster n'est visible sur la matrice des distances des données avant la réduction. Le même résultat est observé sur les trois matrices réduites. En effet, aucun cluster n'est présent dans les données de départ, il est donc logique qu'aucun ne soit visible en sortie des algorithmes de réduction de dimension. De plus, ces algorithmes n'ont pas introduit de biais ou de distorsion créant des clusters n'existant pas dans les données initiales. Avec $Tm = 20\%$, la matrice des distances des données avant réduction de dimension montre que les clusters sont observables mais peu compacts. PCA conserve les clusters mais il y a une perte légère en termes de distances. Après application du NMF, les clusters ne sont plus visibles sur la matrice de distance. Par ailleurs, la matrice de UMAP montre des distance plus élevé que sur la matrice des données avant réduction. Ainsi, UMAP n'a pas conservé les distances des données initiales.

Cette expérience (figure 4) a permis d'étudier le comportement des algorithmes de réduction de dimension en fonction du Tm. Selon cette expérience, PCA est l'algorithme le plus adapté pour réduire la dimension des données binaires, ayant des caractéristiques similaires à ceux générées, puisqu'il a conservé des distances proches aux distances initiales. En effet, les données générées ont été utiles pour l'aide au choix d'un algorithme de réduction de dimension selon les caractéristiques des données.

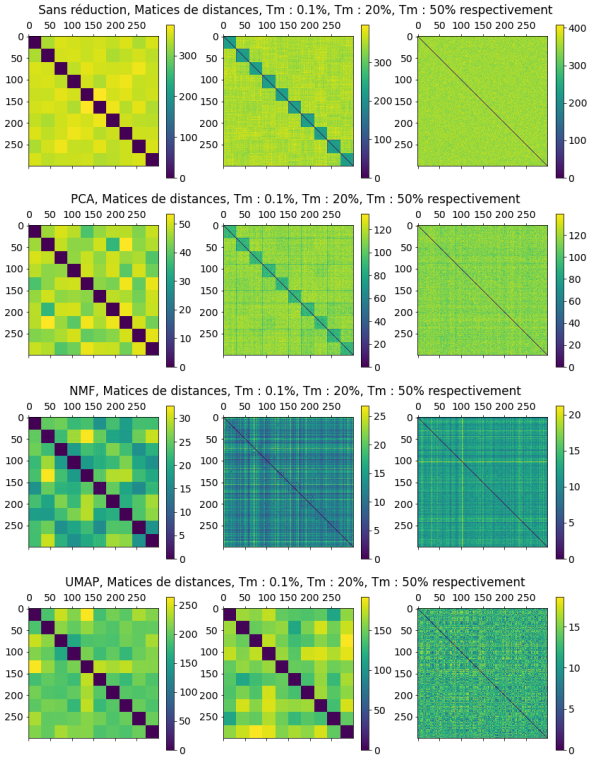

FIG. 4 – *Affichage des matrices des distances des données générées sans réduction et après réduction avec PCA, NMF et UMAP respectivement. Trois valeurs de Tm ont été considéré 0.1%, 20% et 50%.* $C = 10$, $D = 700$, *dimension de l'espace réduit* $= 100$, $Nc = 30$

5 Conclusion et perspectives

Cet article introduit un algorithme de génération de données binaires groupées à partitionnement contrôlées. Un ensemble d'expérimentation a été effectué pour prouver la capacité de l'algorithme à générer des données binaires adaptées pour étudier les approches de partitionnement de ce type de donnée. Un deuxième ensemble d'expérimentation a été effectué pour illustrer l'impact d'algorithmes de réduction de dimensions sur les données générées. Trois algorithmes ont été testés : PCA, NMF et UMAP. Ces expérimentations ont prouvé l'utilité des données générées pour aider dans le processus de choix d'un algorithme de réduction de dimension adapté à un problème donné.

Les perspectives de ce travail seront de proposer un algorithme de génération des modalités binaires corrélées avec un partitionnement maitrisé, étudier les modalités et leurs impact sur la conservation des distances, et finalement, étudier le comportement d'algorithmes de partitionnement sur les données générées.

Références

Bouguila, N. (2010). On multivariate binary data clustering and feature weighting. *Computational Statistics & Data Analysis 54*(1), 120–134.

Farrell, P. J. et K. Rogers-Stewart (2008). Methods for generating longitudinally correlated binary data. *International Statistical Review 76*(1), 28–38.

Jiang, W., S. Song, L. Hou, et H. Zhao (2020). A set of efficient methods to generate high-dimensional binary data with specified correlation structures. *The American Statistician*, 1–13.

Lunn, A. et S. J. Davies (1998). A note on generating correlated binary variables. *Biometrika 85*(2), 487–490.

McInnes, L., J. Healy, et J. Melville (2018). Umap : Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv :1802.03426*.

Ordonez, C. (2003). Clustering binary data streams with k-means. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD '03*. ACM Press.

Ringnér, M. (2008). What is principal component analysis ? *Nature biotechnology 26*(3), 303–304.

Sra, S. et I. S. Dhillon (2006). Generalized nonnegative matrix approximations with Bregman divergences. In *Advances in neural information processing systems*, pp. 283–290.

Wang, X. et A. Kabán (2005). Finding uninformative features in binary data. In *Lecture Notes in Computer Science*, pp. 40–47. Springer Berlin Heidelberg.

Summary

Binary data, data having two possible values, are widely used in several researches such as protein modelling in bioinformatics. Some problems involve clustering binary data. The availability of real data, to study the applicability of some algorithms to a given problem, is not always obvious. This issue is even more visible in the case of unsupervised learning, and for clustering problems. To resolve these issues, this paper proposes a new clustered binary data generation algorithm. Indeed, this algorithm generates clustered binary data through various parameters. These parameters are useful to generate data with known characteristics and controlled clusters. This article details a method of generating clustered binary data, and presents a comparison of the dimension reduction algorithms to show the effectiveness of the generated data in helping to choose a dimensionnality reduction algorithm that conserves clusters separability.