Université Claude Bernard Lyon 1

État de l'art (A):

Sélection de variables

Supervisée – Mono-Label

LEFEBVRE Julien

P2105454

MERCIER Loris

P1906860

Encadrant : Mr. Khalid BENABDESLEM

Unité d'Enseignement : MIF11 – Ouverture à la recherche

Année : 2022 - 2023

Table des matières

Table des illustrations

Introduction

A travers des évolutions techniques majeures dans le monde informatique, les années 90 ont marqué le début de grands jeux de données. Avec des temps de calculs de plus en plus rapides et des coûts de production de moins en moins élevés, les bases de données se sont peu à peu démocratisées pour devenir aujourd'hui un enjeu majeur de notre société pour tous les secteurs d'activité. Alors que ces jeux de données ne possédaient que quelques dizaines ou centaines de variables à cette époque, il n'est pas rare aujourd'hui de trouver des bases en possédant plusieurs dizaines de milliers, notamment dans le domaine de la biologie et de ses études génotypiques. Les premiers classifieurs de machine learning sont rapidement devenu dépassés pour des jeux de données aussi conséquents entrainant les chercheurs du monde entier à trouver des solutions optimales pour répondre à ce besoin.

C'est alors qu'est apparu le domaine de sélection de variables (*feature selection* en anglais) autrement appelée sélection de caractéristiques[1]. La sélection de variables est une technique du machine learning permettant de réduire l'espace de dimension initiale en supprimant les données redondantes ou non pertinentes en fonction de plusieurs critères prédéfinis. Cette méthode vise donc à sélectionner un sous-ensemble de variables « pertinentes » permettant d'augmenter les performances des algorithmes de classification. A noter que cette étape fait directement suite à la phase de « pré-processing » consistant entre autres à standardiser, normaliser, discrétiser les données dans le but de les rendre plus signifiantes.

De nombreux auteurs, chercheurs, ont déjà réalisé des états de l'art complet au sujet de la sélection de variables. Nous pouvons par exemple citer Guyon et Elisseef en 2003 [11] donnant une introduction complète des différentes approches de sélection de variables ou Saeys, Inza et Larranaga à travers leur article « *A review of feature selection techniques in bioinformatics* » [26].

De manière générale, trois grandes approches de la sélection de variables se retrouvent dans la littérature internationale : les méthodes de **filtre**, les méthodes **de symbioses** (ou **wrapper**[2] en anglais), et les méthodes dîtes **intégrées (embedded**[2] en anglais**)**. L'approche **par filtrage** est une méthode rapide basée principalement sur des tests statistiques. Elle est utilisée en amont de l'étape de classification. Les méthodes **wrappers,** introduites par Kohavi et John en 1994 [13], utilisent quant à elles un algorithme de classification afin d'évaluer la performance de chaque sous-ensemble variable. Enfin, les techniques **embedded** regroupent la phase de sélection de variables et de classification en une seule étape permettant d'avoir un lien fort entre le type de sélection et le type de classifieur.

C'est à travers ces 3 grandes approches que nous déroulerons notre état de l'art sur la sélection de variables. Nous prendrons le temps de présenter les différentes sous approches et algorithmes existants pour chaque partie et d'expliciter les avantages et inconvénients de chaque méthode.

[1] Nous utiliserons indifféremment ces deux appellations dans cet état de l'art

[2] Nous ferons le choix d'utiliser la dénomination usuelle des méthodes dans la littérature, à savoir les noms anglophones

1 Méthodes de filtres

Les méthodes de filtrage sont une des approches de la sélection de variables, utilisant des critères statistiques afin d'évaluer l'importance des variables. Concrètement, ces techniques permettent de classer les variables d'entrée en fonction de leur corrélation avec la variable cible en utilisant pour cela différents critères. On peut citer par exemple le test du chi2, la corrélation Pearson, le test de fisher ou encore le gain d'information… Les variables d'entrée supérieures à un seuil donné sont par la suite conservées tandis que celles inférieures sont écartées.

Ce seuil est d'ailleurs un choix important afin de ne conserver qu'un nombre optimal de variables. Pour les tests de corrélation simple, on utilise les seuils classiquement utilisés dans ces méthodes. Par exemple, si l'on regarde la p-valeur d'un test statistique, on conservera par convention les variables inférieures au seuil 0.05 comme le préconise R.Fisher dans son ouvrage *Statistical Methods for Research Workers* [7]. Pour les autres tests, ce seuil de sélection est déterminé de manière empirique.

1.1 Avantage/inconvénients

Comme le souligne I.Guyon dans *An Introduction to Variable and Feature Selection (2003)* [11], l'intérêt des méthodes de filtrage est donc la réduction de l'espace initial en ne sélectionnant que les caractéristiques les plus significatives. Contrairement aux approches *wrappers* ou *embedded* présentées dans nos sections suivantes, la sélection de variables est ici totalement indépendante du classifieur utilisé pour l'apprentissage automatique. Il n'y a donc pas de surajustement possible pouvant fausser les prédictions futures. Toutefois, comme relevé par de nombreux scientifiques comme Kohavi and John (1994) [13], les interactions avec le classifieur sont absentes car indépendantes comme l'illustre leur schéma ci-dessous.

Figure 1: Schématisation de la sélection de variables par approche de filtrage

La sélection de variables ne tient alors pas compte des performances et caractéristiques de chaque algorithme de prédiction. Il se peut donc que l'approche par filtrage enlève des variables moins signifiantes dans l'absolu, mais très performantes avec un type d'algorithme précis. C'est par exemple le cas de la méthode de filtrage *RELIEF* (voir section Multivarié) qui d'après un article de Kohavi, John et Pfleger [16] ne fonctionnerait pas avec le *Naive Bayes classifier*. Cette idée ressort également de l'article *Irrelevant Features and the Subset Selection Problem (1994)* [13], où les scientifiques indiquent clairement : « *to determine a useful subset of features, the subset selection algorithm must take into account the biases of the induction algorithm* ». Ils nous invitent alors à étudier l'approche *wrapper* résolvant directement cet inconvénient d'indépendance entre le modèle de sélection et l'algorithme de prédiction. Nous y reviendrons d'ailleurs dans la section suivante.

Néanmoins, malgré certains défauts, l'approche par filtrage est très utile de par sa simplicité et sa rapidité d'exécution. Le temps de calcul de cette méthode est en effet nettement plus rapide que les autres approches de sélection lui permettant d'être très utilisée dans les grands ensembles de données, notamment les données biologiques comme le précise *Saeys et al.* [26]. La même conclusion se retrouve d'ailleurs dans l'article *Analysis of Feature Selection Algorithms and a Comparative study on Heterogeneous Classifier for High Dimensional Data survey* [14], nous indiquant que les méthodes de filtre sont recommandées pour de grands jeux de données.

1.2 Univarié/Multivarié

Parmi les méthodes de filtrage existantes, il est possible de former deux sous-groupes au caractéristiques différentes : les méthodes univariées, et les méthodes multivariées (linéaire et non linéaire).

1.2.1 Méthodes univariées

Les approches univariées consistent à évaluer l'importance de chaque variable individuellement sans tenir compte d'éventuelles dépendances entre elles. Nous retrouvons dans cette catégorie la **corrélation de Pearson** permettant de quantifier la dépendance linéaire entre deux variables X et Y, **le test du Chi2** permettant d'évaluer la probabilité de corrélation à partir des distributions de fréquences des variables, les **tests-F** *(=tests suivant la loi de fisher*) examinant les variances de nos variables ou encore les **tests-T** *(=tests suivant la loi de Student)* évaluant les moyennes. Globalement, ce sous-ensemble repose donc sur des tests statistiques rapides et efficaces donnant des performances optimales dans de grands ensembles de données comme détaillé précédemment. Toutefois, ces méthodes examinent les variables de manière indépendante ne détectant donc pas la redondance ou des relations entre les variables. Ainsi, ces méthodes peuvent entrainer une perte d'information.

1.2.2 Méthodes multivariées

Pour pallier ce problème de relation entre les variables, de nombreux chercheurs se sont mis à proposer des approches dîtes « multivariées », considérant un sous-ensemble de variables simultanément lors de l'étape de sélection. Ces méthodes sont plus coûteuses en temps de calcul mais réduisent la perte d'information. On peut distinguer les méthodes favorisant les relations linéaires entre les variables et les méthodes non-linéaires.

1.2.2.1 Modélisation Linéaire

Parmi les relations linéaires, nous pouvons cités la **Fast Corrélation-Based Filter** (FCBF) introduite en 2002 dans un article de Yu et Liu [33]. Cet algorithme consiste à sélectionner les variables à forte corrélation avec la cible et à faible corrélation avec d'autres variables. Il s'appuie pour cela sur une métrique appelée « incertitude symétrique » (SU) rattaché aux concepts d'entropie et de gain d'information. Concrètement, l'algorithme FCBF sélectionne un ensemble de caractéristiques dont la corrélation SU est supérieure à un seuil donné puis va chercher celle possédant une corrélation prédominante. Ce concept de corrélation prédominante est introduit de toutes pièces par les deux chercheurs. Il intervient lorsque aucune caractéristique n'est plus corrélée à X que X ne l'est à la cible. Ainsi, cet algorithme prend réellement en compte les relations entre les variables tout en restant relativement rapide pour les grands ensembles de données. Ces méthodes linéaires fonctionnent particulièrement bien avec des classifieur tel que le LinearSVC ou tout autre spécifiquement conçu pour des données linéaires.

1.2.2.2 Modélisation non-linéaire

Un deuxième type de méthode multivariée est l'approche orientée non linéaire. **RELIEF** est une de ces approches multivariées introduite pour la première fois en 1992 par Kira et Rendell [15]. Plusieurs variantes sont depuis apparues formant la famille des algorithmes RBA (*Relief-based Algorithm*) comme l'algorithme **Relief-F** (KonoenKo, 1994 [17]) ou **SURF** (Greene et al., 2009 [10]). Cette approche repose sur la notion de voisin le plus proche détectant ainsi les dépendances entre les

variables. Ces algorithmes sont appréciés pour leurs performances dans des jeux de données complexes possédant des relations non linéaires entre les variables. Toutefois, les temps de calculs font partie des plus élevés de la grande famille des méthodes par filtrage.

De nombreuses autres méthodes de filtrage multivarié que nous ne détaillerons pas ici existent malgré tout comme l'approche **mRMR** (minimum-Redundancy-Maximum-relevance) introduit en 2005 par Peng, Long et Ding [22], l'algorithme **MIFS** (Mutual Information-based Feature Selection) (Battiti, 1994 [1]), la méthode Joint Mutual Information **JMI** développé en 1999 (Yang et al [32]) ou encore l'approche **CMIM,** Conditional Mutual Information Maximization (Schlittgen, 2011 [27]).

2 Méthodes Wrappers (symbioses)

Autre approche de la sélection de variables, la méthode wrapper utilise des algorithmes de classification pour évaluer l'importance de chaque caractéristique. Introduite pour la première fois par John et al. en 1994 [13], la méthode consiste à évaluer pas à pas un sous-ensemble de variables jusqu'à obtenir un ensemble optimal comme l'illustre le schéma ci-dessous.

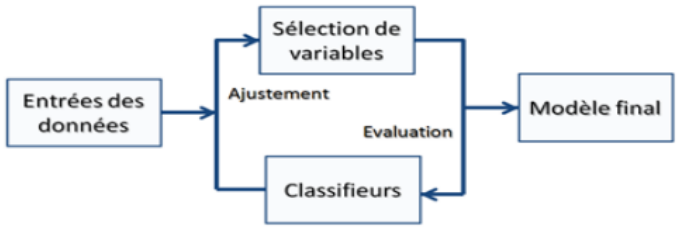

Figure 2: Schématisation de la sélection de variables par approche wrapper

L'avantage des méthodes wrappers est donc sa prise en compte des interactions entre les variables. En effet, de par ses nombreuses combinaisons testées, l'algorithme nous permet de capter les relations entre les données. De plus, contrairement à l'approche par filtrage, les algorithmes wrappers sélectionnent leur sous-ensemble de variables en lien avec le classifieur. Les caractéristiques sélectionnées sont donc spécifiques aux classifieurs choisis augmentant les performances de ces-derniers.

Néanmoins, les techniques wrappers possèdent plusieurs inconvénients. Le premier défaut de ces approches est son coût de calcul élevé. Comme le montrent plusieurs études, le nombre de sous-ensembles à tester évolue exponentiellement en fonction du nombre de variables présentes [26]. Ainsi, les approches wrappers ne peuvent être utilisées dans le cadre de grands jeux de données. De plus, il a été montré que ces techniques sont très sujettes au surapprentissage. [8]

Il est fréquent dans la littérature de classer les méthodes wrappers en deux sous-catégories : Les approches déterministes, et les approches randomisées.

2.1 Wrapper Deterministic

2.1.1 Forward Selection

Forward Selection est une des heuristiques de recherche fréquemment utilisée dans les méthodes wrappers. Elle permet de sélectionner les variables les plus pertinentes en utilisant une métrique de

performance spécifiée. Cette approche est généralement utilisée pour les modèles de régression et de classification mais ne garantit pas de trouver le sous-ensemble de variables optimal [25].

Plusieurs méthodes d'évaluation des caractéristiques cohabitent. Certaines utilisent des statistiques tandis que d'autres utilisent des métriques de performances telles que l'erreur de prédiction ou l'AUC pour la classification. Il est important de choisir une méthode convenant au classifieur choisi et s'adaptant aux types de données (linéaires, non-linéaires, normalisées ou non, etc…). Certains chercheurs [30] ont aussi proposé des méthodes pour améliorer la précision de la méthode de Forward Selection.

2.1.2 Backward Elimination

La méthode Backward Elimination est une technique de sélection de variables très similaire à la méthode Forward Selection. Il s'agit d'une méthode itérative consistant à éliminer les caractéristiques les moins pertinentes en utilisant une métrique de performance spécifiée pour évaluer les caractéristiques restantes.

La méthode démarre avec un ensemble complet de caractéristiques puis élimine à chaque étape la caractéristique ayant le moins d'impact sur la performance selon la métrique choisie. Cette étape est répétée jusqu'à ce qu'un sous-ensemble de caractéristiques optimal soit obtenu.[20] Les méthodes d'évaluation des caractéristiques sont ici les mêmes que pour la méthode précédente.

2.1.3 Exhaustive Search

La méthode de recherche exhaustive (Exhaustive Search) consiste à évaluer toutes les combinaisons possibles de variables pour trouver un sous-ensemble optimal en fonction d'une métrique de performance spécifiée. [19] Cette méthode est généralement utilisée pour les petits ensembles de données avec un nombre de variables limités.

Il est important de noter que cette approche est très coûteuse en termes de temps de calcul pour les grands ensembles de données. Cependant, le développement rapide de la technologie expérimentale et la croissance de la puissance de calcul, en particulier l'utilisation de GPU pour les calculs, permettent la recherche exhaustive possible en un temps raisonnable [9].

2.2 Wrapper Randomized

Deuxième catégorie de l'approche wrapper, les heuristiques *randomized* ajoutent la notion d'aléatoire au fonctionnement général de ces méthodes. A la différence des approches déterministes, ces algorithmes vont ici générer un certain nombre de sous-ensembles de variables de manière aléatoire avant de les évaluer avec une des métriques de performance précédemment citée [18].

L'avantage de cette heuristique est sa qualité à trouver un sous-ensemble optimal de manière aléatoire, améliorant ainsi la rapidité des résultats par rapport à d'autres méthodes de sélection. Des articles tels que *Randomized variable elimination* (Stracuzzi et al., 2004 [29]) confirment d'ailleurs cette qualité en comparant les performances de l'approche aléatoire avec celle déterministe et confirmant la supériorité des approches *randomized.*

3 Méthodes Embedded (intégrées)

Troisième grande catégorie de la sélection de variables, l'approche embedded**,** se distingue de ses concurrentes en intégrant la phase de « *sélection de variables* » directement dans la partie « training » du processus de classification (Guyon et al, [11]). Pour cela, cette approche s'appuie sur les caractéristiques des classifieurs afin de sélectionner les variables spécifiquement optimales à son fonctionnement. Ainsi, un sous-ensemble de variables optimales pour un classifieur X sera différent d'un ensemble optimal pour l'algorithme Y. Les méthodes embedded ont donc l'avantage d'avoir une grande interaction entre les variables et le classifieur là où l'approche par filtrage rend cet échange indépendant et là où l'approche par wrapper n'inclue pas cette interaction directement dans son processus de classification (Saeys et al. [26]).

Toutefois, bien que plus rapide que les wrappers, l'approche embedded reste beaucoup moins performante que les méthodes de filtre pour les grands jeux de données. Cela est principalement dû à une complexité de calcul plus élevé (Saeys et al. [26]). Une solution pour amortir cette différence serait de réduire l'espace de dimension en amont de la méthode Embedded. On trouve d'ailleurs dans la littérature plusieurs algorithmes hybrides combinant les avantages des méthodes de filtrage et embedded, à savoir la réduction rapide du nombre de variables pour l'un et l'interaction forte avec le classifieur pour l'autre. C'est par exemple le cas de recherche récente essayant de rendre plus robuste des algorithmes de *random forest* en ajoutant une étape de filtrage en amont [2].

3.1 Arbre de décision

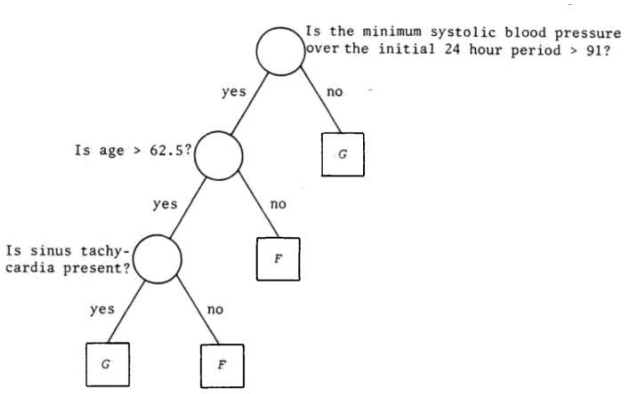

Figure 3: Schéma d'un arbre de décision CART

Cité à l'instant à travers l'approche *random forest*, les arbres de décision sont considérés comme l'une des premières approches des méthodes embedded. Existant à partir des années 60, c'est en 1984 qu'une équipe de scientifiques menée par Léo Breiman publie ses recherches au sujet d'un nouveau type de classifieur : l'arbre CART (*Classification and Regression Trees*) [4]. Comme tout arbre de décision, l'arbre CART fonctionne en divisant les données de manière itérative en fonction des valeurs de chaque variable. Chaque nœud représente un test sur une des variables de notre jeu de données tandis que les feuilles représentent la prédiction finale. L'avantage des arbres de décision réside dans sa facilité d'interprétation, il suffit de parcourir l'arbre pour retrouver les conditions amenant à un résultat. En revanche, ils sont très sujets au surapprentissage.

Chaque arbre de décision applique ses propres critères de construction (ordre de choix des variables, nombre de fils, profondeur de l'arbre, etc…). L'arbre CART produit par exemple exclusivement des arbres binaires. De nombreuses variantes à cet arbre ont par la suite été développées comme l'arbre ID3 (Quinlan, 1986) [23], puis son évolution C4.5 (Quinlan, 1993) [24].

Les années 2000 marquent ensuite l'expansion des *random forests* introduite de nouveau par Breiman en 2001 [5]. Les *random forests* sont une combinaison de classifieur d'arbres. Concrètement, cet

algorithme construit une multitude d'arbre de décision différents puis les agrège entre eux. Un individu est classé en prenant le vote majoritaire sur tous les classifieurs d'arbres dans une forêt. Il est important de souligner que cette approche est très sujette à ses hyperparamètres, c'est-à-dire les paramètres caractérisant le *randomForest*. Il est alors essentiel pour ces algorithmes de tester différentes combinaisons de paramètres afin d'en trouver une configuration performante[3].

De nombreux tests de performances ont d'ailleurs été réalisés au sujet des *randomForest*. On apprend par exemple dans un article de 2004 (Lunetta et al, [21]) que ces algorithmes surpassent « *de manière significative le test exact de Fisher en tant qu'outil de dépistage* ». Ces algorithmes semblent donc dépasser les performances des approches de filtrage. D'après une comparaison d'algorithmes réalisée en 2006 sur 10 métriques couramment utilisées (accuracy, F-score, AUC, etc…), les *randomForest* font partie des meilleurs algorithmes de prédiction [6].

Toutefois, ces conclusions sont à nuancer avec d'autres études. En effet, plusieurs chercheurs ont pointé à l'époque l'inefficacité des *randomForest* dans des grands jeux de données, concluant que cette méthode est plutôt appropriée à des ensembles de données de taille raisonnable. (Ziegler et al, 2007) [34].

Profitant des progrès techniques, de nombreuses variantes au *randomForests* de Breiman [5] ont par la suite été proposées afin d'augmenter les performances de cette approche. On peut citer par exemple l'algorithme *Random Jungle* développé en 2010 par Scharz, König et Ziegler [28] qui, obtient des résultats probants 159 fois plus rapidement.

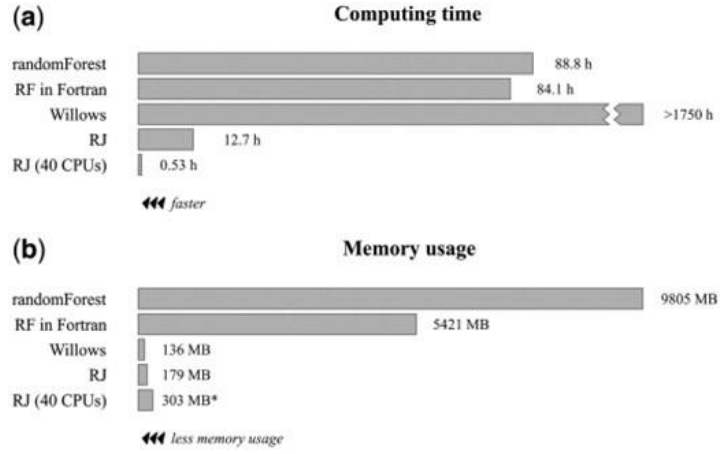

Figure 4: Comparaison des performances des approches randomForest issue de l'article sur le Random Jungle [10]

Cependant, les approches *randomForest* connaissent encore aujourd'hui quelques difficultés dans les grands jeux de données vis-à-vis de la notion d'interaction entre variables. Une étude de 2012 (Winham et al.) [31] conclue notamment que l'identification d'interactions dans les données est d'autant plus faible que le jeu de données est grand. Dans un grand espace de données, l'algorithme serait bien plus influencé par les effets marginaux entre les variables que par de réelles dépendances.

[3] Certaines bibliothèques standard de machine learning proposent des fonctions préconçues pour tester des paramètres. C'est le cas de la méthode GridSearch présente dans la bibliothèque *Scikit-Learn* de *python*.

3.2 Régularisation

Une autre technique des méthodes embedded est la sélection de variables par régularisation principalement combinée avec des classifieur linéaires tel que SVM. Introduite en 1996 par Tibshirani [30], la méthode LASSO est l'une de ces techniques, utilisant la régularisation L1. Elle consiste à minimiser la somme des valeurs absolues des coefficients, le tout pondéré par un facteur lambda. Cette méthode est plutôt robuste vis-à-vis des valeurs extrêmes

La régularisation L2 utilisé par le modèle RIDGE minimise quant à elle la somme des carrés des coefficients. Comparée à LASSO, cette méthode est moins performante sur des valeurs extrêmes mais permet de mieux apprendre sur des modèles de données complexes.

Conclusion

A travers les différentes approches présentées tout au long de cet état de l'art, il apparait que la sélection de variables est une étape clé du processus d'apprentissage ayant pour objectif d'améliorer les performances du modèle. Pour cela, les différentes méthodes tendent à réduire l'espace de données initial et le surapprentissage. L'ensemble de ces approches se découpe en trois grandes catégories ayant chacun leurs avantages et inconvénients.

Les filtres utilisent des critères statistiques pour classer et sélectionner les variables. Cette approche simple et rapide est totalement indépendante du classifieur. Néanmoins, elle ne tient pas, ou peu, compte des interactions entre les variables entrainant une perte d'information pour le processus de classification.

Les wrappers ont l'avantage de sélectionner un sous-ensemble de variables en fonction du classifieur utilisé. Les variables ainsi sélectionnées sont optimales pour la classification entrainant des meilleurs résultats de prédiction. Toutefois, ces méthodes sont coûteuses en temps et en ressources et donc peu efficace sur un grand ensemble de données malgré les avancées technologiques.

Un compromis de ces deux approches est alors les techniques Embedded embarquant la sélection de variables directement dans le processus d'apprentissage. Cette combinaison de deux phases en une entraine un gain de temps conséquent par rapport aux approches wrappers. Cela reste toutefois moins rapide que les méthodes de filtrage.

Finalement, il est difficile de tirer une conclusion quant à l'algorithme « le plus performant » d'entre tous. Plusieurs recherches sur le sujet indiquent d'ailleurs qu'il n'existe tout simplement pas de « bon » algorithme [3]. Tout dépend du contexte d'application de chaque apprentissage. Une hybridation de ces algorithmes comme évoquée dans notre section embedded pourrait alors être la piste la plus prometteuse afin d'obtenir de meilleurs résultats.

Références

[1] Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, *5*(4), 537-550.

[2] Jerbi, W., Brahim, A. B., & Essoussi, N. (2017). A hybrid embedded-filter method for improving feature selection stability of random forests. In *Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016)* (pp. 370-379). Springer International Publishing.

[3] Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and information systems*, *34*, 483-519.

[4] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). Classification and Regression Trees. *Chapman and Hall/CRC*.

[5] Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

[6] Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168).

[7] Fisher,R (1935). Statistical Methods for Research Workers, *Oliver and Boyd*, p200.

[8] González, J., Ortega, J., Damas, M., Martín-Smith, P., & Gan, J. Q. (2019). A new multi-objective wrapper method for feature selection–Accuracy and stability analysis for BCI. *Neurocomputing*, *333*, 407-418.

[9] Goudey, B., Rawlinson, D., Wang, Q., Shi, F., Ferra, H., Campbell, R. M., ... & Kowalczyk, A. (2013). GWIS-model-free, fast and exhaustive search for epistatic interactions in case-control GWAS. *BMC genomics*, *14*, 1-18.

[10] Greene, C. S., Penrod, N. M., Kiralis, J., & Moore, J. H. (2009). Spatially uniform relieff (SURF) for computationally-efficient filtering of gene-gene interactions. *BioData mining*, *2*, 1-9.

[11] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.

[12] Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (Eds.). (2008). *Feature extraction: foundations and applications* (Vol. 207). Springer.

[13] John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine learning proceedings 1994* (pp. 121-129). Morgan Kaufmann.

[14] Kassahun Azezew Ayidagn, prof. Shilpa Gite (2017) Analysis of Feature Selection Algorithms and a Comparative study on Heterogeneous Classifier for High Dimensional Data survey, *International Journal of Engineering Trends and Technology (IJETT)*, V53(2),59-63.

[15] Kira, K., & Rendell, L. A. (1992, July). The feature selection problem: Traditional methods and a new algorithm. In *Aaai* (Vol. 2, No. 1992a, pp. 129-134).

[16] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*(1-2), 273-324.

[17] Kononenko, I. (1994, April). Estimating attributes: Analysis and extensions of RELIEF. In *ECML* (Vol. 94, pp. 171-182).

[18] Mao, Y., & Yang, Y. (2019). A wrapper feature subset selection method based on randomized search and multilayer structure. *BioMed research international*, *2019*.

[19] Mnich, K., & Rudnicki, W. R. (2020). All-relevant feature selection using multidimensional filters with exhaustive search. *Information Sciences*, *524*, 277-297.

[20] Nguyen, H. B., Xue, B., Liu, I., & Zhang, M. (2014, July). Filter based backward elimination in wrapper based PSO for feature selection in classification. In *2014 IEEE congress on evolutionary computation (CEC)* (pp. 3111-3118). IEEE.

[21] Lunetta, K. L., Hayward, L. B., Segal, J., & Van Eerdewegh, P. (2004). Screening large-scale association study data: exploiting interactions using random forests. *BMC genetics*, *5*, 1-13.

[22] Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, *27*(8), 1226-1238.

[23] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*, 81-106.

[24] Quinlan, J.R. (1994). C4. 5: Programs for machine learning by j. ross quinlan. *morgan kaufmann publishers, inc.*, 1993.

[25] Reif, M., & Shafait, F. (2014). Efficient feature size reduction via predictive forward selection. *Pattern Recognition*, *47*(4), 1664-1673.

[26] Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, *23*(19), 2507-2517.

[27] Schlittgen, R. (2011). A weighted least-squares approach to clusterwise regression. *AStA Advances in Statistical Analysis*, *95*(2), 205-217.

[28] Schwarz, D. F., König, I. R., & Ziegler, A. (2010). On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data. *Bioinformatics*, *26*(14), 1752-1758.

[29] Stracuzzi, D. J., & Utgoff, P. E. (2004). Randomized variable elimination. *The Journal of Machine Learning Research*, *5*, 1331-1362.

[30] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267-288.

[31] Winham, S. J., Colby, C. L., Freimuth, R. R., Wang, X., de Andrade, M., Huebner, M., & Biernacka, J. M. (2012). SNP interaction detection with random forests in high-dimensional genetic data. *BMC bioinformatics*, *13*, 1-13.

[32] Yang, H., & Moody, J. (1999, June). Feature selection based on joint mutual information. In *Proceedings of international ICSC symposium on advances in intelligent data analysis* (Vol. 23). Rochester, NY: Citeseer.

[33] Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, *5*, 1205-1224..

[34] Ziegler, A., DeStefano, A. L., König, I. R., & Group 6. (2007). Data mining, neural nets, trees—problems 2 and 3 of genetic analysis workshop 15. *Genetic epidemiology*, *31*(S1), S51-S60.