Sur la classification d'étoiles en fonction de leur spectre d'absorption par apprentissage automatique

Patrice Béchard
Département d'informatique
et de recherche opérationelle
Université de Montréal
Montréal, QC H3T 1J4
patrice.bechard@umontreal.ca

Jean-Pascal Guévin
Département de mathématiques
et de statistique
Université de Montréal
Montréal, QC H3T 1J4
jean-pascal.guevin@umontreal.ca

IFT6390 - Fondements de l'apprentissage machine - 22 décembre 2017

Abstract

La classification de spectres stellaires provenant du *Sloan Digital Sky Survey* a été effectuée par trois algorithmes d'apprentissage, soit les machines à vecteurs de support, les réseaux de neurones de type perceptron multicouche et les réseaux de neurones convolutifs. Un taux de classification 93.40% a été obtenu avec les SVM, 94.41% avec les MLP et 94.57% avec les CNN. Une tâche similaire a été conduite avec des données d'électrocardiogramme avec des résultats moins concluants. Les taux de classification obtenus sont de 61.36% pour les SVM, 60.69% pour les MLP et 77.39% pour les CNN.

1 Introduction

L'exploration de notre univers observable a amené les astrophysiciens à observer et à catégoriser des centaines de milliers d'étoiles en fonction notamment de leur taille, de leur masse, de leur température et de leur composition. Ce processus de classification se fait, entre autre, à partir du spectre d'absorption des étoiles qui est une mesure de l'intensité du spectre électromagnétique émis par celles-ci en fonction de la longueur d'onde. Une automatisation efficace de ce processus pourrait par conséquent être un grand avantage. Nous proposons donc trois algorithmes de classification d'étoiles en fonction de leur type spectral à l'aide de méthodes d'apprentissage automatique. Pour la validation des algorithmes, des données d'électrocardiogramme, étant aussi des données corrélées en 1 dimension, seront utilisées. Les algorithmes d'apprentissage utilisés pour effectuer la classification sont les réseaux de neurones de type MLP, les réseaux de neurones convolutifs (CNN) ainsi que les machines à vecteur de support (SVM) à noyau souple. Les bases de données ainsi que les algorithmes d'apprentissages utilisés sont présentés en détails à la section 2 et les résultats obtenus sont présentés à la section 3. Les codes et les figures présentées pour l'ensemble du projet sont disponibles en ligne sur GitHub : `https://github.com/patricebechard/Machine_learning_IFT6390`.

2 Méthodes

Les données utilisées pour les spectres d'étoiles proviennent de la base de données *Sloan Digital Sky Survey* (SDSS) *Science Archive Server* (SAS) donnant gratuitement accès aux observations faites par différents télescopes. Il est évidemment nécessaire de traiter les spectres obtenus par le biais du SDSS, ceux-ci étant généralement très bruités. Un processus lissage et de normalisation utilisant notamment des moyennes mobiles permet d'en extraire l'information pertinente en éliminant le plus possible le bruit et en ne conservant que ce qui semble correspondre à des tendances plus globales. De plus, chaque spectre a été tronqué de sorte que seul la section correspondant aux log-longueurs d'onde entre 3.65 et 3.80 – correspondant aux longueurs d'onde entre ≈ 398.1 nm jusqu'à ≈ 707.9 nm, ce

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

qui représente le spectre de lumière visible – a été conservé. Nous avons aplati les spectres en ajustant une courbe de degré 3 aux données et divisé par celle-ci. Finalement, une interpolation linéaire de points a permis de diminuer le nombre de traits caractéristiques à 1000, ce que les algorithmes peuvent manipuler sans problème. Un échantillon de 60000 étoiles réparti également pour 6 types spectral différents utilisés (A, F, G, K, M, WD) a été traité. Puisque ces spectres sont les seules entrées des algorithmes essayés, le prétraitement des données a un impact majeur sur les résultats. La figure 1 présente un exemple d'un spectre d'étoile avant et après avoir subi le processus de prétraitement. Ce processus est implémenté par le module `sdss_preprocess_data` qui permet également d'extraire les données directement de SDSS.

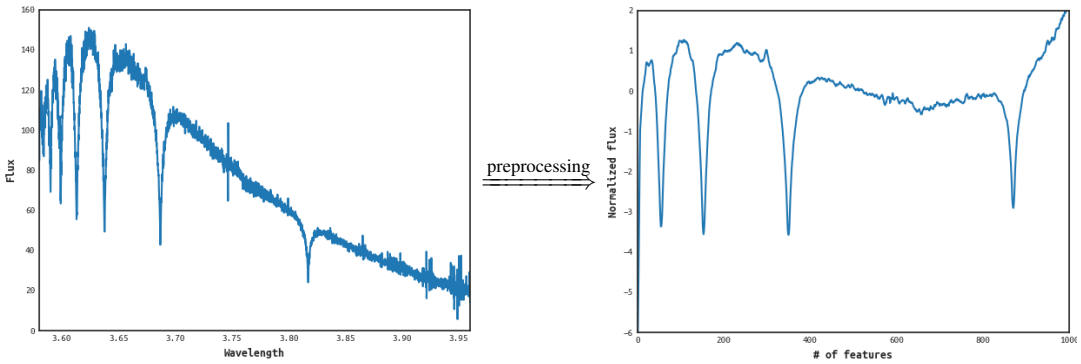

FIGURE 1 – Effet du prétraitement des spectres d'étoiles. À gauche, un exemple de données brutes fournies par le *Sloan Digital Sky Survey* est présenté. À droite, le même exemple a été normalisé et lissé.

Une première validation des algorithmes est effectuée par le biais d'une tâche connexe, soit la classification d'électrocardiogrammes selon l'état de santé du patient duquel il provient. Les électrocardiogrammes (ECG) étant fort semblables dans leurs formes à des spectres d'étoiles (tous deux étant des données corrélées dans l'espace en 1 dimension), ceci permet un premier ajustement des algorithmes en plus de nous initier au fonctionnement de ceux-ci dans le contexte d'une analyse spectroscopique. Les électrocardiogrammes utilisés proviennent du *PhysioNet/Computing in Cardiology Challenge 2017* et ont l'avantage d'être plus simple à analyser, puisqu'ils sont plus lisses et moins bruités. Une séquence de 10 secondes a été conservée pour chaque ECG et les séquences ont été classées en 2 catégories, soit un patient en santé (5050 exemples), soit un patient avec une arythmie ou un autre problème cardiaque (3478 exemples). Le nombre de traits caractéristiques pour cet ensemble de données est de 500 pour chaque exemple. Le prétraitement des données est implémenté par le module `ecg_preprocess_data`.

Trois familles d'algorithmes seront étudiées dans le cadre de ce projet. Tout d'abord, nous utiliserons des réseaux de neurones de type perceptron multicouche (MLP). L'usage de librairies telles Keras ou TensorFlow rend très simple l'implémentation de ce type d'algorithme. Le réseau créé prendra en entrée un spectre traité (centré, réduit et lissé) et retournera en sortie la classification du spectre selon un encodage *onehot*. La méthodologie utilisée pour ajuster les hyperparamètres (nombre de couches cachées, nombre de neurones dans chaque couche, régularisation, taille des lots) consiste à faire un *grid search* sur plusieurs architectures de réseaux ainsi que plusieurs types de régularisation et plusieurs tailles de lots pour trouver une ou plusieurs combinaisons prometteuses en mesurant l'erreur de classification sur l'ensemble de validation. Un processus de *fine tuning* des paramètres suivait ensuite pour améliorer le plus possible l'efficacité de l'algorithme. Finalement, un réentraînement sur l'ensemble d'entraînement ainsi que l'ensemble de validation avait lieu, accompagné d'un test sur l'ensemble de test nous a permis d'obtenir les résultats finaux. Une présentation en détails des résultats obtenus est faite à la section 3. L'utilisation d'un processeur graphique (GPU) nous a permis d'effectuer une panoplie de différentes expériences sans que celles-ci soient trop coûteuses en temps. Des travaux similaires ont été menés par Bailer-Jones et al. (1998) ainsi que Carricajo et al. (2004) pour les spectres stellaires, et par Shensheng Xu et al. (2017) pour les électrocardiogrammes.

Ensuite, puisque les points formant un spectre sont corrélés entre eux, les réseaux de neurones convolutifs (CNN) sont une approche d'apparence prometteuse. Le réseau créé calculera des convolutions unidimensionnelles sur les spectres. Le nombre de convolutions, de *feature maps* ainsi que le type de

pooling (max, moyen, etc.) souhaités sont les hyperparamètres à déterminer. Ce type d'algorithme permet de réduire le nombre de dimensions du problème en plus de prendre en compte des caractéristiques des entrées comme la connectivité locale des traits caractéristiques, tout en introduisant une invariance des traductions locales grâce au *pooling*. Hála (2014) a mené des travaux similaires pour les spectres d'étoiles. Plusieurs travaux de recherche, notamment par Rajpurkar et al. (2017) ainsi que Zihlmann et al. (2017) se sont penchés sur la classification d'électrocardiogrammes à l'aide de réseaux de neurones convolutifs.

Enfin, les machines à vecteur de support (SVM) à noyau souple seront le dernier type d'algorithme d'apprentissage utilisé pour la classification des spectres stellaires ainsi que des électrocardiogrammes. Ce genre d'algorithme a tendance à bien se débrouiller avec des entrées possédant un grand nombre de traits caractéristiques. Le noyau à utiliser sera déterminé lors de la sélection du modèle. L'implémentation des SVM a été effectuée à l'aide de la librairie Scikit Learn. Des travaux similaires ont été menés par Bu et al. (2014) pour la classification de spectres d'étoile et par Shensheng Xu et al. (2017) pour les électrocardiogrammes.

3 Présentation des résultats

3.1 Électrocardiogrammes

Contrairement à ce à quoi nous nous attendions, nous avons eu beaucoup plus de problèmes avec les données d'électrocardiogramme qu'avec les données de spectres stellaires. Tout d'abord, pour les SVM, des tests ont été effectués pour déterminer le noyau du SVM à utiliser parmi ceux implémentés par défaut dans Scikit Learn. Nous avons conservé le noyau mou de type polynomial (`poly`, $(\gamma\langle\ x,\ x'\ \rangle\ +\ r)^d$) et un noyau à fonction à base radiale (`rbf`, $\exp(-\gamma||x-x'||_2^2)$). D'autres noyaux (linéaire, sigmoid) ont également été testés, mais le faible taux de succès obtenu les ont rapidement discrédités. L'effet de trois hyperparamètres a été étudié pour cet algorithme : le noyau utilisé, le degré du polynôme dans le cas d'un noyau polynomial et la valeur de la constante de régularisation C puisqu'il s'agit d'un *soft kernel* SVM. La table 1 montre les taux de classification obtenus pour ces hyperparamètres.

TABLE 1 – Taux de classification obtenu avec l'algorithme SVM pour les trois noyaux étudiés en terme du paramétre de pénalité C.

	ENS. D'ENTRAÎNEMENT			ENS. DE VALIDATION		
NOYAU	$C = 0.75$	$C = 1.0$	$C = 1.25$	$C = 0.75$	$C = 1.0$	$C = 1.25$
RBF	74.42%	85.68%	91.24%	59.68%	60.24%	59.61%
Poly. deg. 2	85.29%	89.45%	91.73%	61.08%	61.37%	60.94%
Poly. deg. 3	93.39%	98.03%	98.84%	59.68%	58.13%	56.09%

On note d'abord à quel point cet algorithme appliqué à cet ensemble de données est à risque de sur-apprentissage. En effet, pour chacun des noyaux, le taux classifications atteint 90% et plus dès que $C \geq 1$. Pourtant, pour des valeurs de C à peine inférieure à 1, on est clairement en présence de sous-apprentissage, d'autant plus que les matrices de confusions montrent que tous les éléments sont classés dans la même catégorie. L'étroitesse de la fenêtre entre ces deux pôles à éviter s'explique par la faible de taille de l'ensemble d'entraînement. Il apparaît évident que nous aurions davantage de latitude pour l'entraînement des paramètres si nous étions en possession d'un plus grand nombre de données. Des propositions permettant de contourner seront discutés à la section 4. Il est aussi pertinent de se questionner sur la validité du prétraitement des données pour notre problème.

Une première tentative de diminuer le risque de sur-apprentissage est d'utiliser la technique d'analyse des composantes principales (PCA) afin de diminuer la dimension des traits caractéristiques, similairement à Polat and Güneş (2007). Les tables 4 et 5 montrent les taux de classification obtenus lorsque les 300 et les 100 premières composantes principales sont conservées. L'hypothèse était qu'une diminution du nombre de traits caractéristiques diminuerait le nombre de paramètre du SVM également, ce qui devrait résulter en une baisse de la capacité du SVM. Le même phénomène que sans l'usage de PCA se produit cependant : on passe de sous-apprentissage à sur-apprentissage sans que le taux de classification de l'ensemble de validation n'augmente.

Une dernière tentative d'améliorer les résultats est d'appliquer aux données une transformée de Fourier. En effet, puisque les données représentent un signal, il pourrait être avantageux d'étudier les fréquences principales de ce signal plutôt que le signal lui-même. Une démarche similaire a été conduite par Gothwal et al. (2011) pour prétraiter les entrées d'un réseau de neurones. La figure 3 en annexe montre les taux de classification obtenus pour le noyau polynomial de degré 2 pour différentes valeurs de C. La table 6 montre le taux de classification pour les différents noyaux. L'exact même phénomène se reproduit cependant : l'algorithme passe du sous-apprentissage au sur-apprentissage sans que l'erreur de classification de l'exemple de validation ne diminue. La figure 3 illustre cela : on y voit que le taux de bonnes classifications augmente pour l'ensemble d'entraînement, mais pas pour l'ensemble de validation.

On conclue que le meilleur classeur SVM à noyau *soft* est celui envoyant tous les électrocardiogrammes à la même classe, ce qui est le cas pour le noyau polynomial de degré 2 et pour le noyau de type RBF pour $C = 1$. C'est deux classeurs obtiennent des résultats d'environ 60% sur l'ensemble test, ce qui est seulement dû à la répartition des électrocardiogrammes dans les classes sain et malade. Le résultat d'un tel classeur aurait été de 50% si les répartitions avaient été égales. On conclue que le SVM s'applique mal à cet ensemble de données puisqu'il s'avère incapable de généraliser ses résultats. Notons que l'implémentation du SVM pour les données ECG est effectuée dans `ecg_svm`.

Des résultats similaires ont été obtenus avec les réseaux de neurones de type MLP. Pour l'ensemble des réseaux de neurones de ce rapport, l'optimiseur *Adam* a été utilisé, la fonction d'activation ReLU a été utilisée dans le réseau et la fonction softmax a été utilisée à la sortie. Après un *grid search* sur plusieurs architectures différentes, nous avons trouvé une architecture de réseau possédant deux couches cachées de 250 et 100 neurones, respectivement. Nous avons utilisé comme entrée des neurones les données prétraitées normalement (normalisées) ainsi que les données après y avoir effectuer une transformée de Fourier. Nous avons utilisé un terme de régularisation λ de 0.0005 pour une régularisation de type L_1 et L_2. La courbe d'apprentissage en fonction du nombre d'époques est présenté à la figure 4 en annexe. Comme on le voit, l'implémentation utilisant les données traitées par transformée de Fourier ont peine à dépasser 60% de taux de classification, alors que celles utilisant les données normalisées atteignent à peine 58%, alors qu'il classe tous les exemples dans la même catégorie. Le module `ecg_mlp` implémente les réseaux de neurones pour les données ECG.

Finalement, en utilisant un CNN pour faire la détection d'arythmies cardiaques, nous avons essayé plusieurs configurations pour maximiser la classification. En mettant un CNN et un réseau MLP complètement connecté bout-à-bout, nous avons vérifié l'erreur de classification en faisant varier le nombre de couches de convolution et de pooling. Nous avons gardé le MLP constant d'expérience en expérience avec une couche cachée de 50 neurones. Nous nous sommes limités à des tailles de fenêtres de convolution de largeur 4 et des filtres de pooling de largeur 4. Nous n'avons pas jugé nécessaire de tester le CNN avec les données prétraitées à l'aide de PCA ou d'une transformation de Fourier. Le taux de classification sur l'ensemble de validation en fonction du nombre de convolutions est résumé au tableau 7 en annexe. Notons que ces résultats correspondent au nombre optimal d'époques d'entraînement pour chaque combinaison d'hyperparamètres. Ceci sera également le cas pour tous les tableaux présentant des résultats des MLP et des CNN discutés dans cet article.

L'évolution de l'apprentissage en fonction du nombre d'époques d'entraînement pour le réseau convolutionnel avec 4 convolutions présenté au tableau 7 testé sur l'ensemble de test est présenté à la figure 2. Le module `ecg_cnn` implémente les CNN pour les données ECG.

Les résultats finaux de la précision de chaque algorithmes sur les électrocardiogrammes sont présentés au tableau 2. Les résultats présentés sont ceux pour lesquels les meilleurs résultats ont été obtenus.

TABLE 2 – Résultats pour la classification d'électrocardiogrammes sur l'ensemble de test pour chaque algorithme utilisé.

ALGORITHME	SVM	MLP	CNN
PRÉCISION	61.37%	60.69%	77.39%

Comme attendu, on remarque que le taux de classification des CNN est de plus de 16% plus élevé que celui obtenu avec les autres algorithmes. Cette situation avait été anticipée, puisque les traits caractéristiques des signaux d'électrocardiogrammes sont autocorrélées.

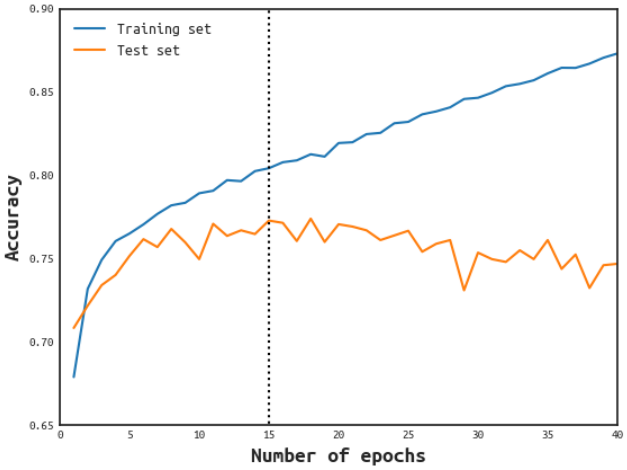

FIGURE 2 – Taux de classification sur les données d'électrocardiogramme en fonction du nombre d'époques d'entraînement pour le réseau de neurones convolutif avec 4 convolutions.

3.2 Spectres d'étoiles

Les résultats obtenus pour la classification de spectres stellaires ont été beaucoup plus concluants que ceux obtenus pour la classification d'électrocardiogrammes. Tout d'abord, avec les SVM, nous avons sélectionné un noyau RBF, puis avons essayé plusieurs valeurs pour la constante de pénalité C. Les résultats obtenus sont présentés à la figure 5 en annexe. On voit que la constante C a un effet important sur les résultats obtenus, et que la valeur optimale de C s'avère être 10. Le module `sdss_svm` implémente le SVM pour les données de SDSS.

Pour les réseaux de neurones, nous avons essayé plusieurs architectures au hasard et avons aussi modifier la taille des lots et la régularisation pour l'apprentissage. Encore une fois, les non-linéarités choisies dans le réseau étaient de type ReLU et celles à la sortie étaient de type *softmax*. La table 8 en annexe présente le taux de classification obtenu pour différentes architectures de réseau de neurones. L'effet de l'architecture sur le taux d'apprentissage était minime, ne faisant varier celui-ci que par moins de 1% à taille de lot constant (50) et sans régularisation. Les résultats de classification dépendent donc davantage du *seed* que de ces hyperparamètres.

Le second hyperparamètre à ajuster pour régler la capacité du modèle est la taille des lots. Pour le MLP avec les couches cachées $[150, 100, 50]$, plusieurs tailles de lots ont été testées et les résultats sont rapportés à la table 9. Encore une fois, le taux de classification ne varie que de très peu pour différentes tailles de lots. Finalement, nous avons essayé plusieurs types de régularisations pour le même réseau de neurones. La figure 6 présente l'effet de la régularisation sur le taux de classification maximale. Les résultats obtenus grâce à la régularisation n'ont pas non plus eu beaucoup d'influence sur le taux de classification.

Une courbe d'apprentissage pour la configuration optimale du MLP est présentée à la figure 7. Les réseaux de neurones sont implémentés pour les données de SDSS par le module `sdss_mlp`.

Enfin, un algorithme de type réseau de neurones convolutif à été étudié pour l'ensemble de données SDSS. Les hyperparamètres étudiés sont le nombre de convolution ainsi que le nombre de *feature maps*, le *batch size* et le nombre de couches cachées ainsi que le nombre de neurones les composant du réseau de neurones transmettant le vecteur de sortie du CNN à la sortie du réseau. Mentionnons également qu'un *max pooling* est effectué après chaque convolution.

D'abord, le nombre de convolutions effectuées à été étudié pour des nombres variables de *feature maps*. Pour ces essais, le réseau de neurone à la sortie du CNN a été gardé identique avec une couche cachée de taille 10 et fonctions d'activation ReLU et softmax. Les *batch size* sont également demeurés constant à 50. La table 10 en annexe montre les taux de classifications obtenus pour l'ensemble de validation pour quelques unes de ces combinaisons. Les résultats obtenus sont relativement similaires, mais tentons d'optimiser les autres hyperparamètres pour la configuration [10, 50].

Essayons différents *batch size* afin de pouvoir observer l'effet de cet hyperparamètre. La table 11 en annexe montre les résultats obtenus pour des *batch size* de respectivement 5, 50 et 500. On voit que le choix des *batch size* n'a que peu d'influence sur le taux de classification. Néanmoins, puisque ce sont des *batch size* de 50 qui donnent de (légèrement) meilleurs résultats, prenons cette valeur pour la suite des choses.

Enfin, optimisons la taille de la couche cachée du réseau de neurone en sortie du CNN. Les architectures considérées sont celles se trouvant à la table 12 en annexe. Encore une fois, on voit qu'il n'y a que très peu de différence entre les différentes valeurs de ces hyperparamètres. On note cependant que c'est une seule couche cachée de taille 50 qui a donné les meilleurs résultats. La courbe d'apprentissage du CNN pour les paramètres optimaux choisis est présentée à la figure 8. On remarque que l'apprentissage pour le CNN est beaucoup plus lisse que pour le MLP.

Concluons cette analyse en notant que les hyperparamètres n'avaient pratiquement aucun effet sur les taux de classification obtenus. Cela signifie probablement que l'ensemble de données est en général très facile à classer. On peut imaginer que les données forment des *clusters* bien définis dans l'espace des traits caractéristiques sauf pour environ 5% des spectres qui se mélangent à des *clusters* d'une autre classe que la leur. Il est donc facile d'obtenir environ 94% de bonnes classifications sur un ensemble test, mais il serait très ardu d'obtenir davantage. On remarque également que le taux classification pour l'ensemble d'entraînement, bien que plus élevé que celui de l'ensemble de validation, n'a jamais atteint 100%, ce qui est cohérent avec l'hypothèse proposée. Le module `sdss_cnn` implémente les CNN pour les données de SDSS.

Les résultats finaux de la précision de chaque algorithme sur les spectres stellaires sont présentés au tableau 3. On voit que les algorithmes permettent une meilleure généralisation pour de nouveaux exemples comparativement aux résultats obtenus plut tôt pour la classification d'électrocardiogrammes.

TABLE 3 – Résultats pour la classification de spectres d'étoiles sur l'ensemble de test pour chaque algorithme utilisé.

ALGORITHME	SVM	MLP	CNN
PRÉCISION	93.40%	94.41%	94.57%

4 Conclusion

En somme, nous avons classifier des étoiles en fonction de leur type spectral en utilisant les spectres de ces étoiles comme entrée pour nos algorithmes. Nous avons conduit une expérence similaire pour des électrocardiogrammes, donnant cependant des résultats beaucoup moins concluants. Un taux de classification 93.40% a été obtenu en utilisant les SVM, 94.41% avec les MLP et 94.57% avec les CNN lors de la classification des étoiles. Les scores étaient plutôt de 61.36% pour les SVM, 60.69% pour les MLP et 77.39% pour les CNN pour la classification d'électrocardiogrammes.

Pour améliorer le score de classification obtenu pour la classification d'électrocardiogrammes, il serait pertinent dans une expérience future d'implémenter les méthodes de *cross-validation* ou la technique de *bootstrap* pour que les différents algorithmes puissent mieux généraliser avec le nombre d'exemples disponibles dans la base de données. Évidemment, une base de données plus volumineuse permettrait de diminuer le risque de sur-entraînement.

Finalement, il serait intéressant dans des expériences futures dútiliser les spectres stellaires afin de prédire des valeurs de température et de gravité de surface par le biais de régression. Une panoplie d'autres applications de *data mining* en astronomie sont discutées par Ball and Brunner (2010).

5 Répartition et remerciements

La séparation des tâches dans le projet a été équitable. Tous les auteurs ont participés conjointement à l'élaboration des algorithmes ainsi qu'à la rédaction du présent rapport. Nous voudrions remercier Tegan Maharaj et Chehib Trabelsi pour nous avoir orienté vers les données du *PhysioNet/Computing in Cardiology Challenge 2017*.

Références

Bailer-Jones, C. A., Irwin, M., and Von Hippel, T. (1998). Automated classification of stellar spectra ?ii. two-dimensional classification with neural networks and principal components analysis. *Monthly Notices of the Royal Astronomical Society*, 298(2) :361–377.

Ball, N. M. and Brunner, R. J. (2010). Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(07) :1049–1106.

Bu, Y., Chen, F., and Pan, J. (2014). Stellar spectral subclasses classification based on isomap and svm. *New Astronomy*, 28 :35–43.

Carricajo, I., Manteiga, M., Rodríguez, A., and Dafonte, C. (2004). Automatic classification of stellar spectra. *Lecture Notes and Essays in Astrophysics*, 1 :153–164.

Gothwal, H., Kedawat, S., and Kumar, R. (2011). Cardiac arrhythmias detection in an ecg beat signal using fast fourier transform and artificial neural network. *Journal of Biomedical Science and Engineering*, 4(04) :289.

Hála, P. (2014). Spectral classification using convolutional neural networks. *arXiv preprint arXiv :1412.8341*.

Polat, K. and Güneş, S. (2007). Detection of ecg arrhythmia using a differential expert system approach based on principal component analysis and least square support vector machine. *Applied Mathematics and Computation*, 186(1) :898–906.

Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C., and Ng, A. Y. (2017). Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv :1707.01836*.

Shensheng Xu, S., Mak, M.-W., and Cheung, C.-C. (2017). Deep neural networks versus support vector machines for ecg arrhythmia classification. In *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*, pages 127–132. IEEE.

Zihlmann, M., Perekrestenko, D., and Tschannen, M. (2017). Convolutional recurrent neural networks for electrocardiogram classification. *arXiv preprint arXiv :1710.06122*.

Annexe

5.1 Tables

TABLE 4 – Taux de classification obtenu avec l'algorithme SVM pour les trois noyaux étudiés avec utilisation du PCA avec les 300 premières composantes principales conservées dans le prétraitement des données pour l'ensemble de données d'électrocardiogrammes.

	ENS. D'ENTRAÎNEMENT			ENS. DE VALIDATION		
NOYAU	$C = 0.75$	$C = 1.0$	$C = 1.25$	$C = 0.75$	$C = 1.0$	$C = 1.25$
RBF	86.51%	95.62%	98.03%	58.97%	58.55%	58.48%
Poly. deg. 2	93.56%	95.34%	96.13%	53.91%	53.48%	53.55%
Poly. deg. 3	99.70%	99.88%	99.89%	55.03%	55.17%	54.54%

TABLE 5 – Taux de classification obtenu avec l'algorithme SVM pour les trois noyaux étudiés avec utilisation du PCA avec les 100 premières composantes principales conservées dans le prétraitement des données pour l'ensemble de données d'électrocardiogrammes.

	ENS. D'ENTRAÎNEMENT			ENS. DE VALIDATION		
NOYAU	$C = 0.75$	$C = 1.0$	$C = 1.25$	$C = 0.75$	$C = 1.0$	$C = 1.25$
RBF	97.98%	99.63%	99.86%	59.11%	58.34%	58.90%
Poly. deg. 2	91.05%	92.42%	93.77%	52.99%	53.41%	53.55%
Poly. deg. 3	99.98%	99.98%	99.98%	52.15%	52.01%	51.58%

TABLE 6 – Taux de classification obtenu avec l'algorithme SVM pour les trois noyaux étudiés avec l'utilisation d'une transformée de Fourier lors du prétraitment des données pour l'ensemble de données d'électrocardiogrammes.

	ENS. D'ENTRAÎNEMENT			ENS. DE VALIDATION		
NOYAU	$C = 0.5$	$C = 1.0$	$C = 1.5$	$C = 0.5$	$C = 1.0$	$C = 1.5$
RBF	59.54%	100.00%	100.00%	58.90%	58.90%	58.90%
Poly. deg. 2	100.00%	100.00%	100.00%	55.52%	55.52%	55.52%
Poly. deg. 3	100.00%	100.00%	100.00%	59.68%	54.40%	54.40%

TABLE 7 – Taux de classification en fonction du nombre de convolutions dans le réseau de neurones convolutionnel pour les données d'électrocardiogramme.

CONFIG.	ENS. D'ENTRAÎNEMENT	ENS. DE VALIDATION
$[100]$	79.62%	68.82%
$[50, 100]$	79.17%	74.76%
$[10, 50, 100]$	79.22%	76.72%
$[10, 50, 100, 200]$	81.98%	77.92%

TABLE 8 – Taux de classification obtenu avec l'algorithme DNN pour l'ensemble de données SDSS pour différentes architectures.

CONFIG.	ENS. D'ENTRAÎNEMENT	ENS. DE VALIDATION
[50]	97.32%	93.93%
[100]	97.74%	93.89%
[200]	95.53%	93.93%
[400]	97.78%	93.96%
[200, 100]	97.83%	94.39%
[200, 50]	97.15%	94.37%
[500, 100]	97.91%	94.36%
[150, 100, 50]	97.92%	94.56%
[500, 100, 50]	98.12%	94.31%

TABLE 9 – Taux de classification obtenu avec l'algorithme DNN pour l'ensemble de données SDSS en fonction de la taille des lots pour l'architecture [150, 100, 50].

BATCH SIZE	ENS. D'ENTRAÎNEMENT	ENS. DE VALIDATION
5	96.84%	94.31%
50	97.29%	94.08%
500	97.94%	94.41%

TABLE 10 – Taux de classification obtenu avec l'algorithme CNN pour l'ensemble de données SDSS en fonction du nombre de filtres de convolution par couche de convolution.

CONFIG.	ENS. D'ENTRAÎNEMENT	ENS. DE VALIDATION
[10]	97.41%	93.75%
[50]	99.02%	94.23%
[100]	98.62%	94.47%
[10, 50]	96.00%	94.57%
[50, 100]	98.96%	94.40%
[10, 50, 100]	97.41%	94.55%

TABLE 11 – Taux de classification obtenu avec l'algorithme CNN pour l'ensemble de données SDSS en fonction de la taille des lots.

BATCH SIZE	ENS. D'ENTRAÎNEMENT	ENS. DE VALIDATION
5	94.75%	94.13%
50	94.42%	94.40%
500	95.77%	94.31%

TABLE 12 – Taux de classification obtenu avec l'algorithme CNN pour l'ensemble de données SDSS en fonction de l'architecture du réseau à la sortie du CNN.

CONFIG.	ENS. D'ENTRAÎNEMENT	ENS. DE VALIDATION
[20]	95.91%	94.25%
[50]	96.26%	94.36%
[100]	98.02%	94.27%
[20, 50]	95.79%	94.04%
[50, 100]	97.68%	94.29%
[20, 50, 100]	94.35%	94.07%

5.2 Figures

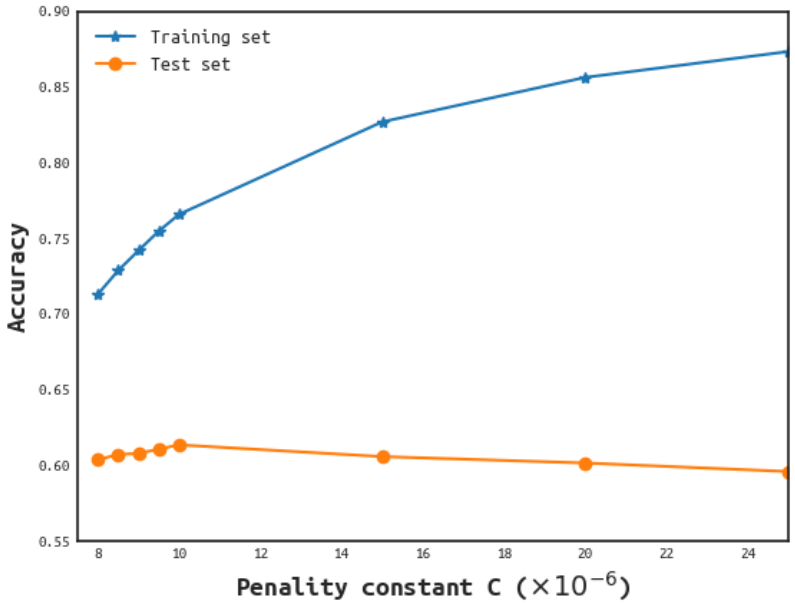

FIGURE 3 – Taux de classification sur les données d'électrocardiogramme en fonction de la constante de pénalité C pour l'algorithme SVM à noyau polynomial de degré 2 avec des données transformées à l'aide d'une transformée de Fourier.

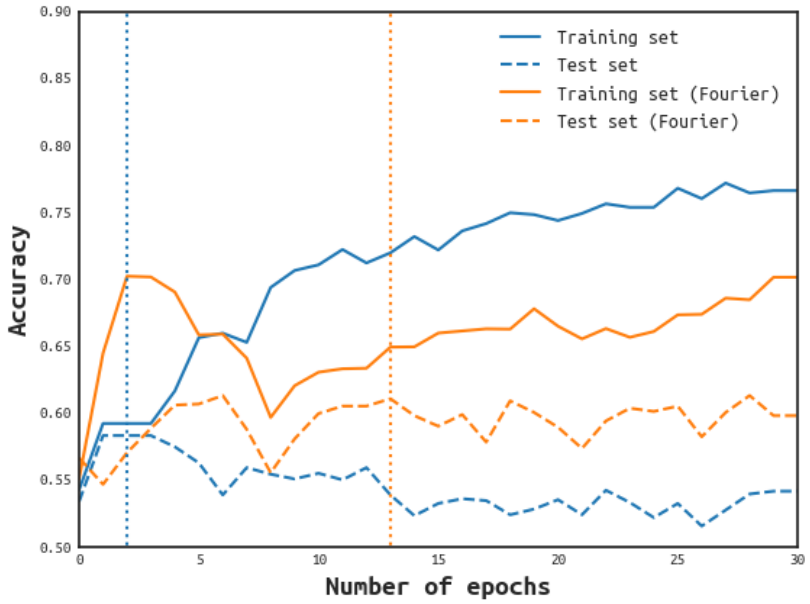

FIGURE 4 – Taux de classification sur les données d'électrocardiogramme en fonction du nombre d'époques d'entraînement pour le réseau de neurones de type MLP avec deux couches cachées de 250 et 100 neurones, respectivement.

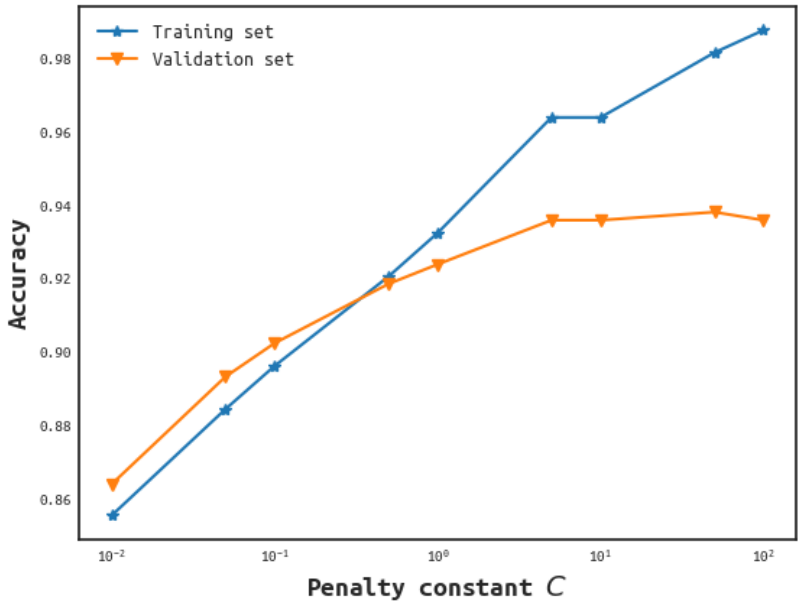

FIGURE 5 – Taux de classification sur les données de SDSS en fonction de la constante de pénalité C pour l'algorithme SVM à noyau RBF.

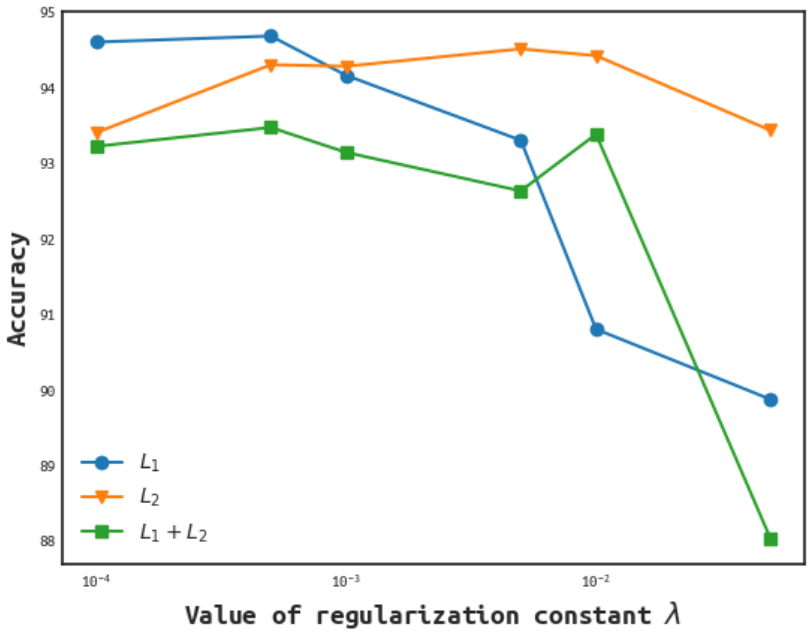

FIGURE 6 – Taux de classification sur les données de SDSS en fonction des paramètres et du type de régularisation pour l'algorithme MLP.

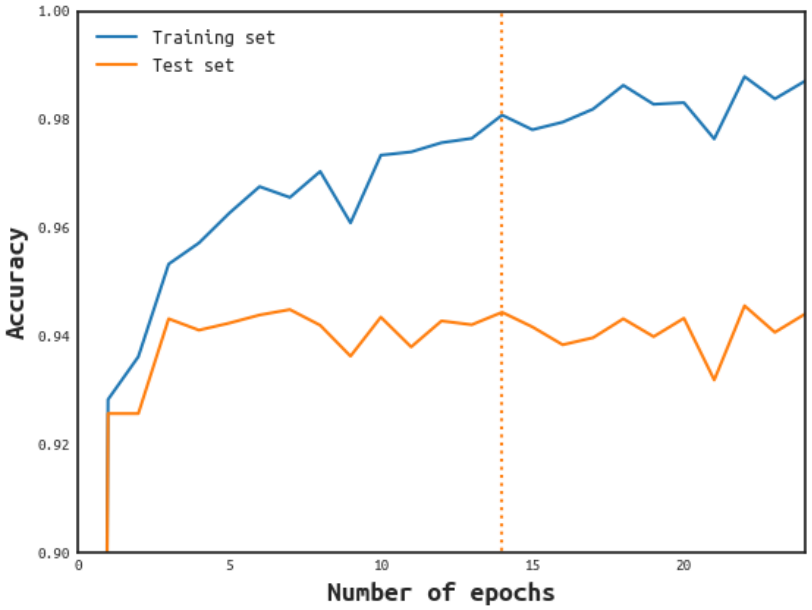

FIGURE 7 – Taux de classification sur les données de SDSS en fonction du nombre d'époques d'entraînement pour le réseau de neurones de type MLP.

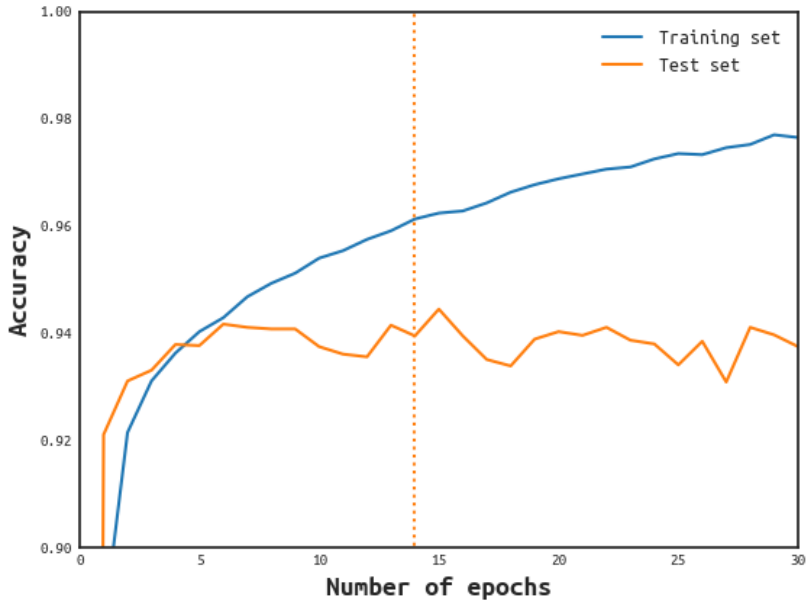

FIGURE 8 – Taux de classification sur les données de SDSS en fonction du nombre d'époques d'entraînement pour le réseau de neurones convolutif.