BBC 2019 : Projet NCI 60

Rapport

Nathan Flückiger, Vincent Guidoux et Joël Kaufmann

Introduction

Ce projet se basant sur des données provenant du programme NCI 60 a pour but la classification de différents types de cancers du set de données. Il est question en premier lieu de préparer les données de manière pertinente pour permettre dans un deuxième temps l'entraînement de modèles de machin learning. L'objectif étant que ces modèles soient capables de classifier les types de cancers avec des performances suffisamment bonnes.

NCI 60

NCI 60 un est programme de recherche sur les cancers humains, il a pour but d'être un moyen de tester les traitements anti-cancer pour les différents laboratoires pharmaceutiques dans le monde. Dans la poursuite de ce but des données concernant les différents cancers étudiés sont disponibles pour la bioinformatique.
Créé dans les années 90 le programme est constitué de 60 lignées de cellule cancéreuses humaines.
Ces lignées proviennent de prélèvements de cellules cancéreuse de 9 types de cancers : leucémie, mélanomes et cancers des poumons, du colon, du cerveau, de l'ovaire, du sein, de la prostate et des reins.
Les données nous sont mises à disposition au moyen d'un fichier *.soft* regroupant 174 échantillons (3 échantillons par lignées, 58 lignées) et l'expression de plus de 54'000 gènes.
Ces différents cancers ne sont pas tous représentés de la même façon, c'est pourquoi il a été nécessaire de faire des catégories nous permettant d'avoir des données plus grandes pour avoir des ensembles d'entraînement et de test d'une taille exploitable.

Classification des cancers grâce à la bioinformatique

Aujourd'hui, dans la bioinformatique, la classification des cancers s'effectue principalement dans les sous-types de cancers. C'est-à-dire que la bioinformatique est principalement utilisée pour savoir de quel type est un cancers après avoir détecté le cancer.
Par exemple, dans le cas d'un cancer du sein nous allons rechercher s'il est de type A ou B.

Dans nos recherches, nous avons trouvé un article[1] parlant de la catégorisation des cancers dans la même lignée que notre projet. Cet article arrivait à des résultats de 51 cancers bien classifiés sur 59.

Méthodologie

Algorithmes

Catégorisation

NCI 60 est composé de 174 échantillons, dans le but de leur classification, nous souhaitons en tirer des classes (catégories) par similarité. Dans un premier temps, cette similarité a été évaluée en effectuant des recherches sur les labels donnés aux échantillons, pour arriver à une catégorisation mélangeant définitions et terminologie de ceux-ci.

Les catégories finales sont présentées dans le tableau ci-dessous:

adenocarcinoma	carcinome	leucémie	mélanome	cerveau
60	55	18	23	18

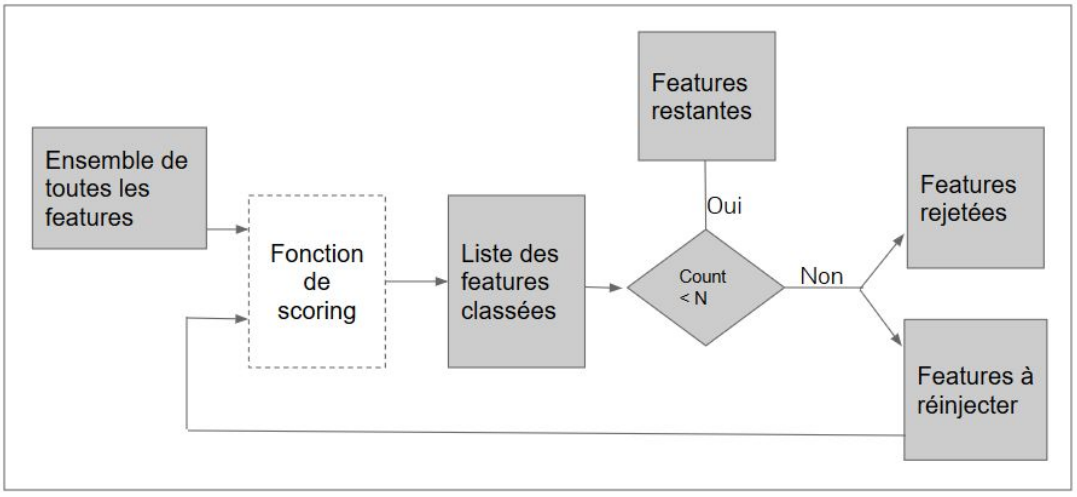
Ne disposant pas de l'expertise nécessaire pour juger de notre catégorisation, nous avons cherché un moyen de valider celle-ci au moyen d'outils statistiques. Notre approche consiste à explorer la variabilité de l'expression des gènes dans une même catégorie. L'écart-type donne une bonne appréciation de la variabilité des données, mais ne nous donne pas d'information sur quelle donnée apporte le plus de variabilité. Nous avons donc pensé à utiliser les premiers et derniers quartiles, c'est à dire les 25 premier pourcents des données et les 25 derniers pourcents de la distribution. Ainsi, nous avons pensé pouvoir mettre en évidence les valeurs d'expression de gène étant le plus loin de la moyenne.
Afin de ne pas prendre de mesures trop hâtivement, le processus est répété sur les 54'000 expressions de gènes et chaque échantillon possède un compteur qui est incrémenté pour chaque valeur dans ces quartiles.
Au final, nous avons trié les échantillons par valeur de compteur pour examiner les différences. Il s'est trouvé que ces valeurs sont relativement proches les unes des autres dans les différentes catégories. Seul un échantillons semble avoir une valeur un peu plus haute que les autres. Toutefois, comme les 2 autres échantillons similaires possèdent, au contraire, des valeurs de compteur très petites, cette catégorisation n'a pas été remise en cause.
Cette approche a toutefois des limites, puisque trouver un nombre de fois où les expressions dévient trop de la moyenne ne veut pas dire forcément que les échantillons possèdent des expressions similaires. Cela pourrait bien être quelque chose qui ressemble à un bruit blanc, parfois proche de la moyenne et parfois pas, ce qui donne une valeur moyenne pour tous les échantillons.
Ceci ne semble toutefois pas être le cas au vue des résultats finaux.

Feature Selection - Tri des caractéristiques

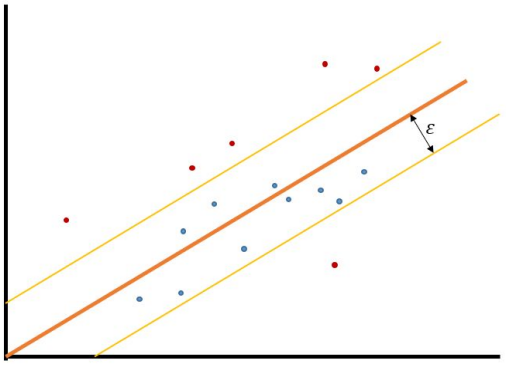
Une des méthodes que nous avons choisies utilise un **seuil de variance(Variance Threshold)** défini à 0.497, valeur qui permet de diminuer significativement le nombre de caractéristiques (features). En effet, toutes les caractéristiques possédant une variance plus grande sont éliminées, cette méthode retire 60% des données.

La **sélection univariée(Support Vector Machine)** assigne un score à chaque caractéristique indépendamment de tous les autres. Une fois que chaque caractéristique possède son score, nous faisons la sélection en ne gardant que les N meilleurs. Nous avons utilisé une fonction de régression qui lui assigne un score en fonction de sa corrélation avec la catégorie qu'il représente.

L'**élimination récursive des caractéristiques(Recursive Features Elimination)** travaille de la manière suivante :



Nous avons choisi à chaque itération de n'enlever que 0.01 % des features jusqu'à arriver à N. Cette méthode sélectionne les caractéristiques en fonction d'un scoring. Nous avons choisi la régression linéaire qui optimise le nombre de points entre la ligne centrale et les lignes qui lui sont parallèles à une distance ε.



Classification

La classification se fait selon un apprentissage supervisé. En effet, les catégories définies au début du projet l'ont été dans le but de fournir aux algorithmes de machine learning une classification attendue. C'est pourquoi il était important de valider la catégorisation. En utilisant des données labellisées, il sera ainsi possible d'évaluer les performances de nos systèmes.

Nous avons utilisé et évalué les performances de 3 algorithmes de machine learning définis ci-dessous:

- Les **Arbres de décision** sont des méthodes d'apprentissage supervisé sans paramètre, utilisée pour la classification et la régression. Le but est de créer un modèle pouvant prédire la valeur d'un variable en apprenant des décisions simple impliquée par les données.

- Les **Support Vector Machine** sont un ensemble de méthodes d'apprentissage supervisé étant utilisée pour la classification, la régression et la détection des valeurs aberrantes.

- La **Classification basée sur les voisins(Knn)** est une méthode d'apprentissage basée sur des instances ou non généralisant. Il ne crée pas de modèle interne, mais enregistre des instances des données d'entraînement. La Classification s'effectue alors par un simple vote à la majorité des voisins les plus proches de chaque points.

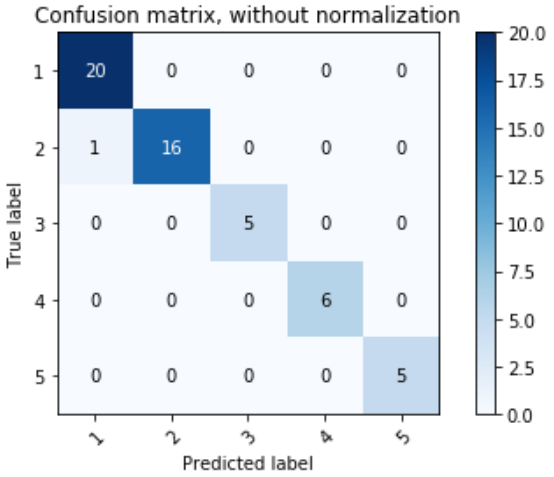
Analyse des résultats

Chacune des expériences où nous testons l'interaction d'une méthode de sélection avec une de classification a été effectuée 20 fois, nous en avons calculé un score moyen, voici les résultats:

		Variance	Recursive features elimination			Unvariant Selection			
	Rien	Threshold	500	1000	2000	500	1000	2000	
Decision Tree	0.764	0.79	0.892	0.882	0.858	0.793	0.814	0.764	0.858
Support Vector Machine	0.604	0.849	0.981	0.981	0.962	0.811	0.849	0.868	0.868
Knn	0.717	0.717	0.849	0.849	0.849	0.774	0.906	0.792	0.849
	0.6605	0.79	0.892	0.882	0.858	0.802	0.8775	0.868	

Nous pouvons observer qu'en moyenne la méthode de classification **Support Vector Machine** est la meilleure, alors que du côté des méthodes de sélection des caractéristiques c'est la **Recursive Features Elimination** qui s'en sort le mieux

Le meilleur résultat obtenu avec un seuil de 1000 caractéristiques pour la **Recursive Features Elimination** et avec la méthode de classification **Support Vector Machine** est celui-ci:



La matrice de confusion ci-dessus rend compte du meilleur résultat, obtenu par l'algorithme Support Vector Machine. Il montre une classification correcte des données de tests à une erreur près (classification d'un échantillon dans la catégorie 1 au lieu de 2).

Outils utilisés

Anaconda nous permet de mettre en place des environnement python, ce qui nous permet de transférer plus facilement notre environnement à un tier.

Jupyter Notebook nous permet de faire un notebook Python pour permettre la remise en main propre du code utiliser plus facilement.

Pandas (Python) est une librairie qui simplifie la manipulation de données par le biais de Dataframes.

SKLearn (Python) est la librairie qui met à disposition plusieurs outils de machine learning.

Conclusion

Les résultats obtenus semblent très satisfaisants. Il serait toutefois intéressant de tester les modèles obtenus sur des données provenant d'autres sources, afin de vérifier l'overfitting ou d'utiliser des algorithmes de machine learning non-supervisés afin de découvrir les catégories proposées par ceux-ci.
Un obstacle majeur rencontré dans ce travail fut de bien interpréter les données et de comprendre la direction que devait prendre le projet pour arriver à un résultat intéressant.
Bien qu'ayant beaucoup appris au travers de ce projet, cela nous a montré l'importance de pouvoir échanger et d'être suivi par des personnes compétentes dans le domaine. En effet, si nous avons maintenant quelques notions de biologie, le rôle du bioinformaticien est de travailler avec des experts, en mettant à profit ces connaissances pour mieux comprendre et traiter les problèmes liés au vivant.

Annexe

Jupyter Notebook

Vous pouvez retrouver notre notebook sur : https://nortalle.github.io/BBC_projet/

Sources

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1525183/

https://www.revmed.ch/RMS/2016/RMS-N-519/Bioinformatique-en-oncologie-une-discipline-incontournable
https://scikit-learn.org/stable/modules/tree.html#tree
https://scikit-learn.org/stable/modules/neighbors.html#classification
https://scikit-learn.org/stable/modules/svm.html