Devoir 5: Rapport

GIF-7005: Introduction à l'apprentissage machine

Stéphane Caron

12 Décembre 2018

Contents

Question 1

Dans la première question, il faut utiliser des techniques d'apprentissage non-supervisé pour en savoir plus sur les causes du cancer du sein, selon le jeu de données *Wisconsin Breast Cancer*. Ainsi, nous utiliserons le *clustering* pour tenter de former des groupes d'observations similaires entres elles et ainsi comprendre davantage les différences entre les tumeurs bénignes ou malignes.

Partie a

Dans la première partie de la question 1, nous allons utiliser l'algorithme *K-means* et évaluer les groupes formés selon 3 métriques différentes. Ces métriques nous permettrons entres autres de savoir si le regroupement nous amème de l'information pertinente quant au type de cancer. Les 3 métriques sont:

- l'indice de Rand ajusté
- le score basé sur l'information mutuelle
- la mesure V

Pour être mesure de former des groupes, l'algorithme *K-means* requiert qu'on décide d'avance le nombre de groupes qui sera formé. Pour nous permettre de faire le bon choix quant au nombre de groupes, nous allons plusieurs possibilités. Le graphique ci-dessous montre les 3 différentes métriques mentionnées plus haut en fonction du nombre de groupes.

À partir de la figure 1, on peut conclure que plus le nombre de groupes est élevé, plus cela semble améliorer les métriques. Cependant, l'amélioration devient de moins en moins importante lorsque le nombre de groupes dépasse un certain nombre. De plus, un nombre de groupes trop élevé peut causer des problèmes en généralisation alors que cela pourrait avoir un impact significatif sur la variance reliée à l'apprentissage. Ainsi, il serait approprié de garder un nombre de groupes plus faible, mais quand même performant comme $K = 6$.

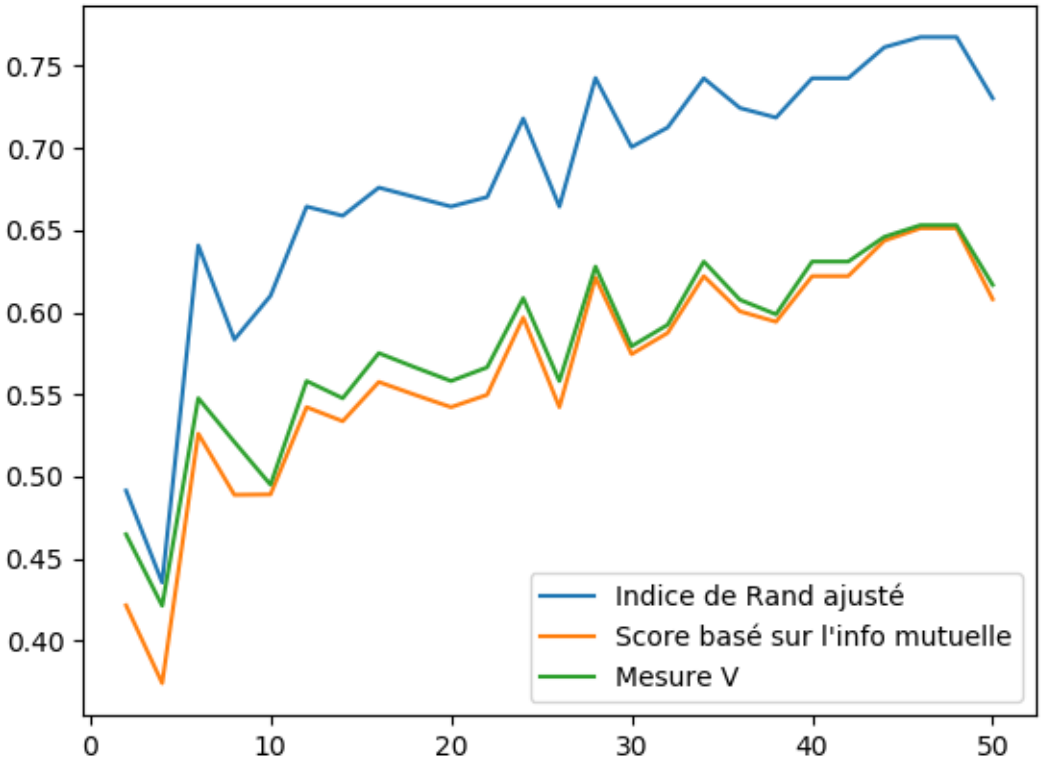

Figure 1: Valeurs de différentes métriques selon le nombre de groupes initialisé pour l'algorithme *K-Means*.

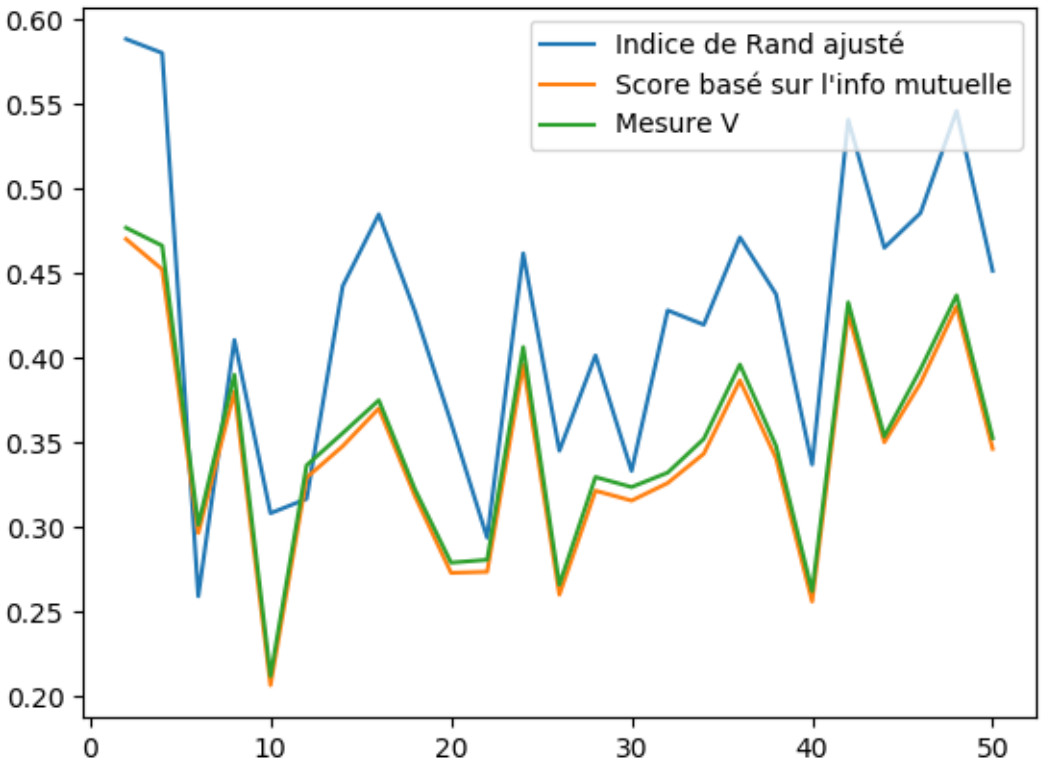

Figure 2: Valeurs de différentes métriques selon le nombre de groupes initialisé pour l'algorithme EM initialisé aléatoirement.

Partie b

Dans cette partie, on réutilise le même genre de démarche qu'à la partie a) afin de déterminer le nombre de groupes à conserver. Cependant, nous allons utiliser notre implémentation de l'algorithme EM (voir code). La figure 2 présente les 3 mêmes mesures qu'en a) pour l'algorithme EM initialisé aléatoirement.

À partir de la figure 2, on peut conclure que de choisir 2 groupes serait probablement le meilleur choix étant donné la performance élevée à ce point. On remarque d'ailleurs beaucoup de variabilité dans les résultats par la suite.

Partie c

Dans la partie précédente, nous avons initialisé l'algorithme EM aléatoirement. Dans cette partie, nous allons initialiser la paramétrisation Φ en utilisant l'algorithme *K-Means*. La figure 3 illustre les résultats des 3 différentes métriques en utilisant cette initialisation.

Comme à la partie b), il est possible de conclure que $K = 2$ nous donne les meilleures performances.

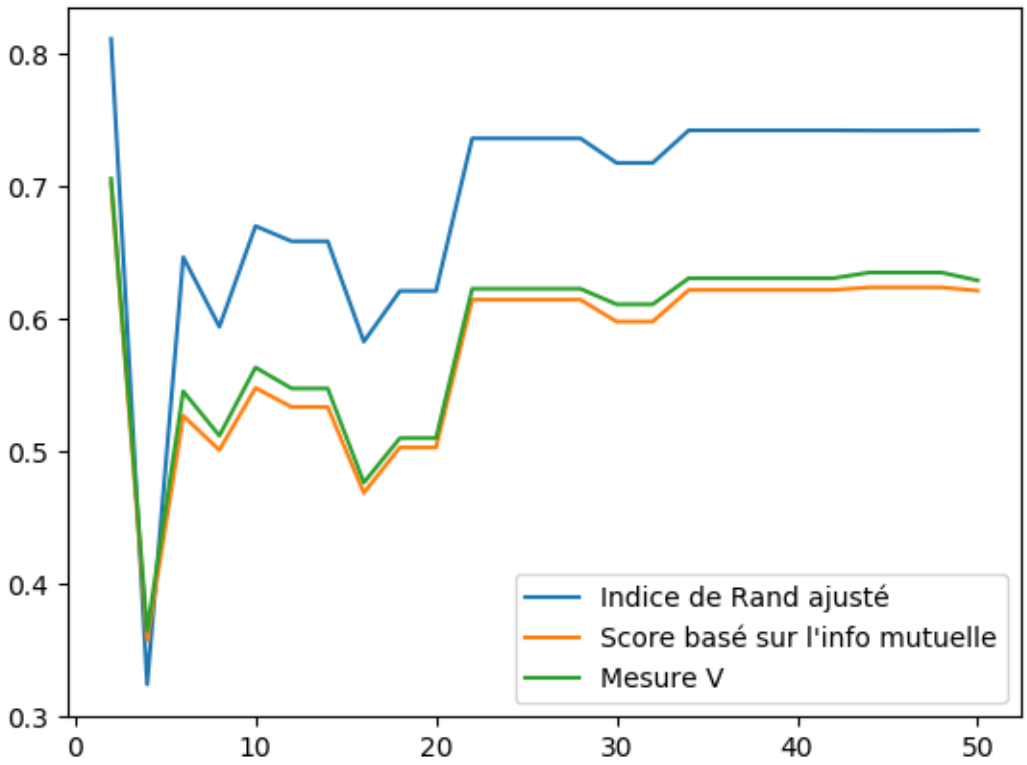

Figure 3: Valeurs de différentes métriques selon le nombre de groupes initialisé pour l'algorithme EM initialisé aléatoirement.

Partie d

En analysant les résultats des 3 sections précédentes, il est possible d'abord de conclure que l'algorithme EM fonctionne beaucoup mieux lorsqu'il est initialisé par *K-Means*.

De plus, on remarque qu'il y a un pic de performance à $K = 2$ pour EM initialisé avec *K-Means*. Ce pic est en grande partie expliqué par le fait que les données sont distributées dans une variable binaire (tumeur maligne ou bénigne). En initialisant avec l'algorithme *K-Means*, celui-ci vient recentrer les moyennes des groupes en ajoutant une composante de covariance. Si on ajoute des groupes, cela diminue les performances, car l'algorithme tente de trouver des nouveaux groupes et dilue l'information qui peut être contenue dans 2 groupes seulement.

Finalement, lorsqu'on compare l'algorithme *K-Means* avec l'algorithme EM pour une loi normale multivariée, on remarque tout d'abord que le premier est un cas particulier du deuxième. En effet, les deux algorithmes fonctionnent de manière itérative. La première étape consiste à trouver les appartenances des obersations selon une paramétrisation Φ, alors que la deuxième étape consiste à maximiser cette paramétrisation Φ. Aussi, les deux algorithmes requiert de déterminer d'avance un nombre de groupes à trouver. Cependant, l'algorithme EM détermine l'appartenance des observations en pondérant grâce aux probabilités qu'une observation $\mathbf{x}^{\mathrm{T}}$ a de se retrouver dans un *cluster* spécificique (G_j). L'algorithme *K-Means* utilise quant à lui des distances et détermine l'appertance de manière binaire.

Question 2

Dans cette question, nous allons utiliser un jeu de données textuelles pour essayer de prédire si une instance est un pourriel ou un courriel pertinent. Pour ce faire, chaque instance a été vectorisé vers 1000 variables, où chaque variable correspond à la présence d'un mot. Afin de faciliter l'apprentissage et diminuer l'effet du fléau de la dimensionnalité, il faut représenter, de manière intelligente, les données dans un nouvel espace plus petit.

Partie a

Dans a première partie, nous allons utiliser des techniques de sélection univariées comme le test du χ^2 et le critère d'information mutuelle. Une fois la sélection de variables terminée, voici les variables (ou les mots) conservées par chacune des méthodes.

Test du chi2	Critère d'information mutuelle
click	click
debian	debian
archive	archive
please	please
debian-user-request	debian-user-request
receive	rights
hibody	receive
newsletter	hibody
policy	policy
privacy	privacy

Dans le tableau ci-dessus, on remarque que les deux méthodes ont sélectionné des variables (ou mots) très similaires. Lorsqu'on compare les performances de ces modèles, on remarque qu'un *SVM* linéaire obtient des performances de 82.7% sur les jeux de données réduits (χ^2 et critère d'information mutuelle) tandis qu'avec le jeu de données initiales (1000 variables), nous obtenons des performances de 93.7%. Il est d'ailleurs intuitif de constater qu'avec le maximum d'informations, les performances sont plus élevées. Toutefois, il est intéressant

de constater ici qu'en laissant tombé 99% de l'informations (990 variables) nous obtenons des performances potentiellement très raisonnable. Le modèle est d'ailleurs beaucoup plus simple avec 10 variables.

Partie b

Dans cette partie, nous allons cette utiliser une technique de sélection de variables séquentielle. Le tableau ci-dessous illustre quelles variables ont été conservées dans le modèle final.

Selection séquentielle arrière
click
debian
hibody
stuff
newsisfree
redhat
archives
lawrence
africa
egroups

Certains mots recoupent ceux trouvés dans la partie a). Par exemple, la variable *click* est peut-être un signe que le courriel incite le destinataire à cliquer sur des liens en particulier pour vendre des choses ou souscrire à des comptes.

La performance du modèle en utilisant cette méthode de sélection de variables donne une précision de 82.9%, ce qui est très similaire en termes de performances aux autres méthodes de sélection de variables.

Partie c

Les algorithmes séquentiels de sélection de variables fonctionnent en ajoutant/enlevant une variable à la fois. Dans ce contexte, s'il y a des non linéaires complexes entres les variables, la méthode séquentielle arrière est préférable à la méthode avant. En effet, la méthode arrière aura tendance à davantage capter ces liens puisqu'elle commence en considérant toutes les variables. À l'inverse, la méthode avant commence avec un jeu de données vides pour ajouter des variables. Il est plus probable de penser que la méthode avant aura de la difficulté à capter ces liens complexes, car ces liens peuvent être fréquemment expliqués lorsque l'information est considérée de manière collective, ce que cette méthode favorise moins en partant d'un ensemble vide.

Lorsque le nombre de variables initiales (D) est grand et que le nombre de variables à conserver est significativement plus petit (K), la méthode séquentielle arrière peut devenir non-efficace. La raison est que l'entraînement est plus difficile lorsqu'il y a beaucoup de variables et dans ce cas spécifique on doit faire beaucoup d'entraînements complexes du modèle avant d'arriver à un modèle final. Dans ce cas-ci, la méthode séquentielle avant peut être plus efficace.

Question 3

Question 4