Examen de Math

Professeur : Henry LAUDE

Par Lucas BILLAUD
PSB 2020-2022

Table des matières

CRITERES D'EVALUATION

Nous nous baserons sur les critères suivants afin d'apprécier les différents travaux :

- La richesse du contenu
- La clarté des explications
- L'argumentation des expressions
- L'attractivité du sujet
- La qualité de rédaction

I. What is a Good Prediction- Issues in Evaluating General Value Functions Through Error

A. Synthèse

Ce document traite de la véracité d'une prédiction. Afin de vérifier qu'une prédiction est correcte, il est nécessaire de la « mesurer » via une approche qui atteste son résultat. Dans ce document, l'approche étudiée est RUPEE (Recent Unsigned Projected Error Estimate). RUPEE est une méthode d'estimation de l'erreur moyenne d'une GVFs (Fonctions de Valeur Générale). Il est démontré que cette approche n'est pas une solution adéquate afin d'évaluer une prédiction. D'une façon générale, l'étude mène une réflexion sur l'évaluation des prédictions pour voir si ces méthodes permettent de faire une différence entre les prédictions utiles et ordinaires.

B. Formule mathématique liée au calcul de l'erreur

$$G_t = E_\Pi(\sum_{k=0}^{\infty}(\prod_{j=1}^{k}(\gamma_{t+j})C_{t+k+1}) \quad (1)$$

$$G_t^e = E_\Pi(\sum_{k=0}^{b}(\prod_{j=1}^{k}(\gamma_{t+j})C_{t+k+1}) - V_t(\phi(o_t)) \quad (2)$$

L'équation (1) est l'expression mathématique d'une GVFs noté G_t. GVFs est une approche qui permet l'apprentissage de plusieurs autres fonctions (de prédiction, peut-être) à travers un seul flux. Sans être certain de ce que nous avançons, nous pensons que chaque terme représente une fonction de valeur qui stocke la prédiction d'un état $\prod$.

L'équation (2) est composée de 2 parties. La 1ère G_t et la seconde V_t . D'après ce que nous avons lu, nous pensons que G_t^e représente une sorte d'erreur entre la prédiction et la valeur réelle, plus généralement une mesure/évaluation de la prédiction. Nous pensons qu'elle représente la méthode RUPEE.

Nous ne saurons donner plus d'informations et expliciter précisément le rôle de chacun d'entre eux étant donné que nous ne maitrisons pas les concepts sous-jacents qu'il faut pour apprécier dans la totalité cette méthode.

C. Evaluation du travail

Bien que le sujet abordé semble très complexe, il a été approché d'une façon succincte. Même si nous ne possédons pas le background nécessaire afin d'apprécier toutes les notions abordées et comprendre complétement la représentation des formules, nous pensons avoir retiré des pistes qu'il nous faudra creuser afin de mieux capter le sujet.

Le vrai plus de ce sujet est que nous n'avions jamais entendu parler de ces notions et que cela nous ouvre d'autres portes d'exploration pour la suite.

D. Conclusion

Nous avons abordé sans rentrer dans des détails trop complexes, l'évaluation d'une prédiction. Une approche de prédiction a été présentée GVFs ainsi qu'un moyen que les auteurs du papier de recherche remettent en question (RUPEE). Nous constatons qu'il est difficile d'être objectif sur le sujet car nous n'avons pas les connaissances suffisantes afin de le traiter dans sa totalité. Cependant, nous avons découvert des notions intéressantes qu'il nous faudra creuser.

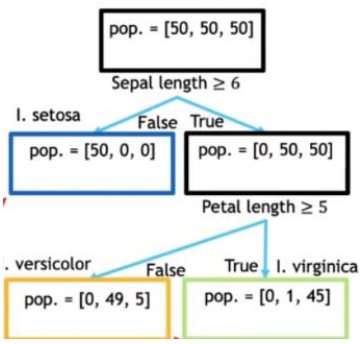
II. Apprentissage par Arbres de Décisions

A. Synthèse

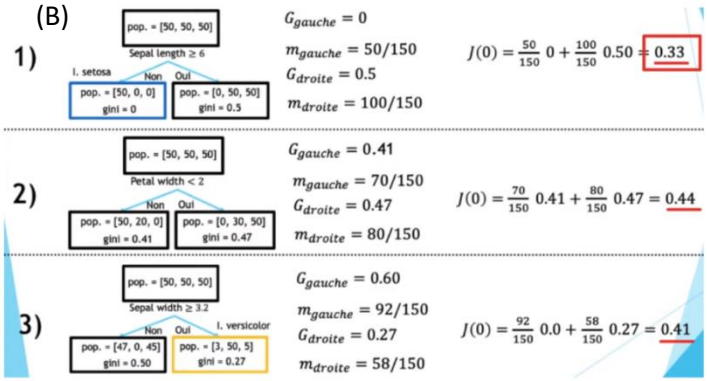
Nous abordons ici les arbres de décisions (régressions et classification). Les arbres de décision sont des algorithmes supervisés. Ils partent d'un ensemble de départ et créent des sous-ensembles basés sur des conditions afin de déterminer, après répétition du processus, la variable à prédire. Afin de créer ces conditions et de vérifier que c'est le meilleur choix de décision, il faut prendre en considération le coût des nœuds. Ce coût se mesure à l'aide de l'indice de Gini (indice de pureté d'un nœud) et de la proportion de la population des nœuds. Les arbres de décisions possèdent des limites telles que l'instabilité et le surapprentissage. C'est pourquoi on a plus souvent recours au Random forest qui est une sorte de généralisation des arbres de décisions.

B. Commentaire sur le fonctionnement des arbres de décisions et sur leur création

(A)



(B)



(*capture d'écran des exemples prise dans le document sur Github)*

(A) Nous voyons ici comment fonctionne un arbre de décision. Le problème à résoudre ici est de classer les Iris par rapport à leurs espèces respectives. Ainsi d'une façon générale, le déroulement est :

- Si la condition est respectée, alors j'appartiens à la classe A sinon à la classe B

Nous abordons aussi la pureté des nœuds mesurée par l'indice de Gini.

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

Avec $p_{i,k}^2$ qui représente la probabilité d'avoir des individus de classes différentes dans le nœud suivant. Ainsi Gi variant de 0 à 1, plus cet indice est faible plus le nœud est pur, cad que tous les individus appartiennent à la classe.

C'est le cas pour le nœud bleu. S'il y a moins de 6 pétales alors c'est forcément un sétosa.

(B) Afin de créer l'arbre (A) il faut passer par une étape qui détermine quelle est la meilleure variable de comparaison. Est-il mieux d'avoir « Sepal lenght » puis « petal length » ou le contraire. Pour faire ce choix, la notion du « Coût du nœud » est utilisée.
Ainsi celui ayant la plus petite valeur (coût le plus faible) sera en 1ere position.

$$j(k) = \left(\frac{m_{gauche}}{m}\right) G_{gauche} + \left(\frac{m_{droite}}{m}\right) G_{droite}$$

Avec G qui représente l'indice de Gini m la population total et m_gauche/droite la part de la population de chaque nœud.

C. Evaluation du travail

Le travail sur le sujet des arbres de décisions est intéressant. L'approche sur la façon dont sont construits les arbres est claire. Il y a quelques coquilles, notamment sur les exemples du calcul de l'indice de Gini, mais cela n'empêche pas la bonne compréhension du sujet traité. La partie exemple sous R est vraiment un plus. De plus, ce sujet sert de bonne base pour la compréhension des Random forest, sujet sur lequel on est redirigé à la fin de la lecture.

D. Conclusion

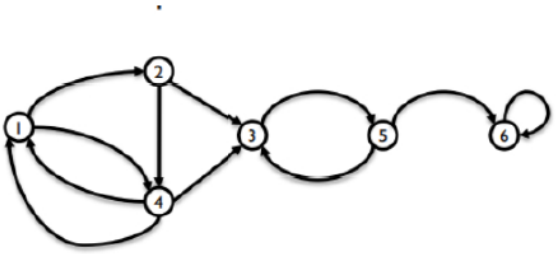
Nous avons une approche théorique et pratique du sujet (explication du fonctionnement et implémetation sous R). Le sujet est très accessible et peut être un bon point de départ pour ensuite s'attaquer au random forest. Cependant, nous aurions aimé avoir un exemple mathématique sur les optimisations des choix de valeurs.

III. Marche aléatoire

A. Synthèse

Ce document traite de la marche aléatoire qui est un processus stochastique de type chaine de Markov et de son utilisation pour le classement des page internet. Une marche aléatoire ou ramdom walk sous sa forme la plus simple est une suite de pas de taille fixe dans une direction aléatoire (par exemple sur une droite, ce sera avancer ou reculer). En utilisant les chaînes de markov et en la rendant ergodique (irréductible, récurrence positive et non périodique) Google détermine la notoriété (taux de présence, probabilité qu'un agent soit sur la page) d'un site ou d'une page web.

B. Focus sur PageRank



$$P = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 2/3 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(*capture d'écran des exemples prise dans le document sur Github)*

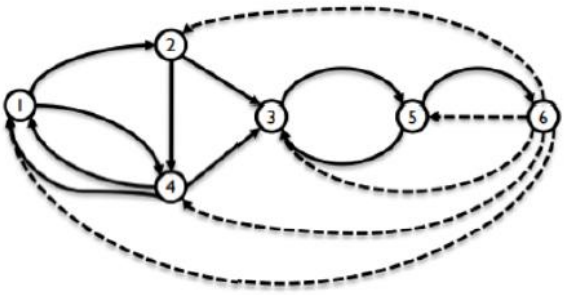
Nous avons ici, une chaine de markov représentée sous forme de graphe.

La 1ere ligne de la matrice représente le nœud 1. Il y a ½ d'aller sur le nœud 2 et ½ d'aller sur le nœud 4.

Nous avons ici une chaine non irréductible. Elle n'est pas irréductible cad que lorsque l'on arrive dans un état (6) il n'est pas possible d'y sortir. Afin de résoudre ce premier problème il faut « créer » une façon d'y sortir. Cela est fait grâce à un saut aléatoire.

Marche Aléatoire
Par William, Marko
https://github.com/WilliamRbc/PSBX/blob/main/MARCHE%20ALEATOIRE/Marche%20ALEATOIRE.pdf

Si $deg(V_i) = 0, P_ij = 1/n$ Sinon, $(i,j) \in E$, alors $P_{ij} = \frac{1}{deg(V_i)}$ sinon $P_{ij} = 0$



$$P = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 2/3 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix}$$

Le saut aléatoire est implémenté grâce à la formule ci -dessus qui affecte une répartition uniforme si un état ne communique avec personne (5 → 6, 6 est uniquement accessible depuis 5 mais une fois dedans on n'y sort plus, c'est ce que l'on appelle un puit) . Maintenant que tous les états sont liés, il faut résoudre le problème de non ergodicité afin de de leur donner une valeur de notoriété.

$$P_{pagerank} = (1-\alpha).P + \alpha.1/n.J \ \ avec \ \ \forall i, \forall j, J_{ij} = 1$$

Le problème est résolu en introduisant α=0.1 . Ainsi, $P_{pagerank}$=0.9*P+0.1*J (je n'ai pas compris comment appliquer ce calcul).

C. Evaluation du travail

Le travail est clair, de plus on voit comment le système de page rank de google est construit. On a aussi une vision du fonctionnement théorique des chaines de markov.

D. Conclusion

La façon d'expliquer la finalité du sujet (pageRank) permet d'aborder des sujets sous-jacents. En effet, l'utilisation des chaines de markov est en général vue d'un point de vue théorique. On a ici une application concrète de ce cas et de la façon de contourner certains problèmes rencontrés.

Approche mathématique sur la "Recommandation automatique et adaptative d'emojis"
Par Imen Derrouiche
https://github.com/imenderrouiche/PSBX/blob/main/Approche%20math%C3%A9matiques%20-%20%C3%89mojis.ipynb

IV. Approche mathématique sur la "Recommandation automatique et adaptative d'emojis"

A. Synthèse

Dans ce travail, on aborde un système de recommandation basé sur la distance de Leivenshtein. Elle permet de mesurer la distance entre 2 mots. Plus précisément en combien d'étape on passe du mot A au mot B (en changeant l'ordre, en supprimant ou en ajoutant des lettres). Afin de « prédire » les emojis susceptibles d'être utilisées, on se base sur la distance au sens de Leivenstein qui sépare 2 mots et de probabilité afin de connaitre quel mot a le plus de chance d'être utilisé.

B. Présentation de l'algorithme

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{si } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{sinon.} \end{cases}$$

Lev a,b (i,j) = max(i,j) représente la distance maximum par exemple entre une chaine de caractère vide est un mot quelconque.
Ensuite en fonction, elle prendra des aspects différents pour définir la distance qui sépare 2 mots.
Ce principe est appliqué ensuite pour recommander des émojis.

C. Evaluation du travail.

Le travail est clair, on aborde une nouvelle version de la distance. On apprend que des algorithmes qui dans un premier temps ne semblent pas avoir une utilité concrète sont en fait utilisés dans les tâches quotidiennes. Le sujet est abordable pour tous les niveaux.

D. Conclusion

Ici encore, on voit un sujet théorique appliqué dans un cadre concret. Au premier abord, on se demande quel pourrait être l'intérêt de mesurer la distance entre 2 mots. Cela est expliqué tout au long du document, au travers d'exemples sur la prédiction de l'utilisation des émojis.

Regression Linéaire simple et multiple
Par Nina Zoumanigui
https://github.com/Nina809/PSBX/blob/main/Regression.Rmd

V. Régression Linéaire simple et multiple

A. Synthèse

Nous traitons ici la régression linéaire. La régression linéaire (simple ou multiple) est une technique de prédiction d'une variable dépendante par rapport à une ou plusieurs variables indépendantes (par exemple prédire le salaire en fonction de l'âge, du niveau d'étude, du sexe…). Les variables indépendantes peuvent être de plusieurs types : quantitatives, qualitatives, catégorielles. La régression se base sur des hypothèses stochastiques et structurelles.

B. Forme de la régression linéaire multiple :

yi = b0 + b1xi1 + b2xi2 + · · · + bpxip + εi , i = 1, · · · , n$.

- yi : variable à prédire
- b0 : point à l'origine
- bi…bp : variables dépendantes
- Ei : erreur résiduelle hypothèse que sa distribution soit normale.

C. Evaluation du travail

On a une mise en place du contexte et de l'utilisation de la régression. On aborde en 1er lieu la régression linéaire avant de partir sur la régression multiple. On explique aussi que les problèmes de régression peuvent avoir beaucoup de variables dépendantes superflues. Cela est intéressant car implicitement on est redirigé sur un sujet de réduction de dimension.

D. Conclusion

On a ici une introduction sur la régression linéaire (simple et multiple). On passe en revue les hypothèses faites et on a sa forme générale.

Kmeans
https://github.com/Lucasblld/PSBX/tree/main/maths

Auto-Evaluation k-means.

Le fait de traiter k-mean était intéressant. Cependant, le fait de devoir résumer un papier de recherche sans réellement avoir abordé de notion mathématique en cours est un frein. De plus, il y a une perte conséquente d'informations dans le fait de résumer un papier de recherche qui explique déjà très bien le sujet. J'ai eu une approche sur des notions telles que la complexité algorithmique ou les distances. Je pense cependant qu'il serait plus judicieux de lire le papier original qui contient tous les éléments à propos du sujet. Concernant le rendu, j'ai essayé de partir à l'essentiel en expliquant l'idée du fonctionnement de k-means et je me suis appuyé sur l'idée de base du papier original qui est comment optimiser le choix des centroids ; cela afin de rendre kmeans plus performant, étant donné que cet algorithme attire plus par sa facilité d'implémentation que par ses résultats. En conclusion, je pense avoir compris la globalité du papier ce qui est assez intéressant. Et j'espère que les autres s'ils choisissent mon sujet, auront au moins les définitions de complexité et d'intuition de k-means pour pouvoir s'attaquer au papier original.