Modèles graphiques: Naïve Bayes classifier Vs. Tree Augmented Bayes classifier

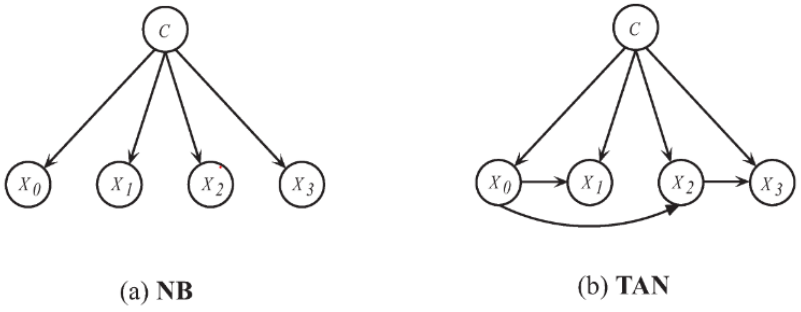
BABAH Sidiya - YECHCHI Sif-Eddine & ELJAMIY Mohamed

April 10, 2021

1 Introduction

L'inférence probabiliste est devenue une technologie de base en Intelligence Artificielle, en grande partie grâce aux développements des méthodes de la théorie des graphes pour la représentation et la manipulation des distributions de probabilités complexes (Pearl, 1988). Les modèles graphiques probabilistes ont une variété de vertus en tant que représentations de l'incertitude et ceci indépendemment de leur structure qui peut être dirigée (*Bayes Networks*) ou non dirigée (*Markov random fields*). Les modèles graphiques permettent une séparation entre les aspects qualitatifs et structurels et les aspects quantitatifs et paramétriques dans la modélisation de l'incertitude. Concrètement, les premiers sont représentés via les patterns des arêtes dans le graphe alors que les derniers sont représentés sous forme de valeurs numériques associées à des sous-ensembles de nœuds dans le graphe. Il est essentiel de préciser qu'il est aujourd'hui beaucoup plus facile d'apprendre et estimer les paramètre d'un graphe qu'apprendre sa structure qui reste encore un aspect difficile à traiter.

Dans cet article nous allons présenter l'approche de deux modèles graphiques dirigés qui sont Naïve Bayes Classifier (NB) et Tree Augmanted Naïve Bayes (TAN), étudier leurs points de similitude et de différence, les forces, les faiblesses et la complexité algorithmique de chacun d'entre eux et à la fin une comparaison numérique en terme du pouvoir prédictif de chaque modèle après les avoir appliqués sur le jeu de données *digits* de la librairie *scikit-learn*.



(a) **NB** (b) **TAN**

Figure 1: Exemples de structure de chaque modèle

Ce n'est pas par hasard que quelqu'un serait intéressé par ces deux vieux algorithmes et leur comparaison. Ces deux algorithmes sont très populaires dans le domaine de la santé pour la segmentation d'images et le diagnostic médical et divers autres domaines où il y a des tâches de regroupement ou classification de données.

2 Comparaison de ces deux modèles

Dans cette section, nous allons établir cette comparaison en fonction de plusieurs aspects qui seront détaillés par la suite.

2.1 L'approche de Naïve Bayes NB

Le modèle Naïve Bayes (NB) est généralement utilisé pour des problèmes de classification. La caractéristique importante de ce modèle est qu'il a une très forte hypothèse d'indépendance. La figure 1 - (a) représente une illustration de la structure d'un modèle NB.

Comme son nom l'indique, il est d'un part **bayésien** pour la simple raison qu'il utilise le théorème de Bayes pour calculer la probabilité conditionnelle, et **naïve** d'une autre part car il assume une indépendance entre les features conditionnellement au label ce qui n'est pas forcement établi. Cependant, l'algorithme semble fonctionner assez bien pour les tâches de classification et l'hypothèse de l'indépendance simplifie énormément des choses. Dans ce modèle, la loi a posteriori de y|X s'exprime en fonction de la vraisemblance X|y et la loi a priori de y.

$$\begin{aligned} \mathbb{P}(y|X) &= \frac{\mathbb{P}(X|y)\mathbb{P}(y)}{\mathbb{P}(X)} \\ &\propto \mathbb{P}(X|y)\mathbb{P}(y) \\ &\propto \mathbb{P}(y)\prod_{i=1}^{n}\mathbb{P}(X_i|y) \end{aligned}$$

Si on suppose que les features conditionnellement au label suivent une loi normale, le modèle s'implémente de la manière suivante:

- Prior[c] = len(X[y==c])/len(y)
- mean[c] = mean(X[y==c])
- variance[c] = var(X[y==c])
- P[,c] = normalDensity(X, mean = mean[c], cov = diag(var)) x Prior[c]
- retourne argmax(P, axis = 1)

2.2 L'approche de Tree Augmented Naïve Bayes TAN

Tree Augmented Naïve Bayes (TAN) est une extension du classifier Naïve Bayes en assouplissant l'hypothèse de l'indépendance. L'idée était que si Naïve Bayes, malgré son hypothèse d'indépendance incorrecte, fonctionne bien, alors avec la prise en compte des corrélations entre les features, la performance de la classification ne peut qu'être améliorée. Ce modèle maintient la structure du graphe Naïve Bayes et l'augmente en ajoutant des arêtes entre les variables afin de capturer les corrélations entre certaines d'elles. Par contrainte de complexité, le TAN impose une restriction sur le niveau d'interaction entre les variables à un. Par conséquent, dans un graphe TAN, chaque variable peut avoir deux parents qui sont le nœud de la classe et un autre nœud de variable, à l'exception de la variable racine qui a uniquement le noeud de la classe comme parent. La figure 1 (b) ci-dessus fournit une instance de ce à quoi un graphe TAN pourrait ressembler.

Une autre notion importante qui joue un rôle clé dans la construction de l'arbre est l'information mutuelle. Pour construire l'arbre, il est obligatoire de mesurer la corrélation entre chaque paire de variables dans les données et ajouter des arêtes uniquement entre les variables les plus corrélées. Concrètement, s'il y a N variables dans les données, alors l'arborescence correspondante aura N nœuds. Ainsi, N-1 arêtes doivent être ajoutées, pour obtenir une structure arborescente qui relie tous les nœuds du graphe. De plus, la somme des poids de toutes ces arêtes doit être le poids maximum parmi toutes ces structures. La mesure de la corrélation entre deux variables qui forme le poids d'une arête dans le graphe s'appelle l'information mutuelle et elle se calcule de la manière suivante:

$$\mathbb{I}_p(X, Y|Z) = \sum_{x,y,z} \mathbb{P}(z)\mathbb{P}(x, y|z) \log \frac{\mathbb{P}(x,y|z)}{\mathbb{P}(x|z)\mathbb{P}(y|z)}$$

Si on suppose que $X_i, X_j|Y$ suit une loi normale, alors l'information mutuelle devient:

$$\mathbb{I}_p(X_i, X_j|y) = -1/2 \log(1 - \rho^2) \geq 0$$

où ρ désigne le coefficient de corrélation entre les deux variables.

La procédure de construction de l'arbre se compose de 5 étapes principales:

- Calcul $\mathbb{I}_p(X_i, X_j|y)$ entre chaque paire de features $i \neq j$,
- Construction d'un graphe non orienté complet dans lequel les sommets sont les features $X_1, ..., X_n$ et annoter le poids d'une arête reliant X_i à X_j par $\mathbb{I}_p(X_i, X_j|y)$,
- Construction d'un arbre de recouvrement maximum, (En occurrence avec l'algorithme de Kruskal),
- Transformation de l'arbre non dirigé résultant en un arbre dirigé en choisissant au hasard une variable racine,
- Définir la direction de toutes les arêtes vers l'extérieur à partir de la racine

Concernant la prédiction, c'est toujours le même principe que celui de Bayes à l'exception de la formule de calcul qui prendra en compte l'indépendance éventuelle entre certaines features.

$$P[,c] = \mathbb{P}(y = c|X) = \mathbb{P}(y = c)\mathbb{P}(X_{root}|y = c) \prod_{\substack{i=1,\\ i\neq root}}^{n-1} \mathbb{P}(X_i|y = c, X_{parent})$$

Et puis retourner argmax(P, axis = 1) comme la classe prédite.

2.3 Complexité des deux algorithmes

2.3.1 Naive Bayes:

Considérons

- $\{A_1, A_2, ..., A_N\}$ N attributs prenant comme valeurs $\{S_1, S_2, ..., S_N\}$.
- C le nombre total d'étiquettes de classe dans l'ensemble de données.
- N le nombre total d'attributs dans l'ensemble de données.
- S_{max} le maximum de $\{S_1, S_2, ..., S_N\}$.
- R le nombre total d'instances d'apprentissage.

Supposons que, en comptant le nombre d'occurrences de chacun des libellés de classe et les états des attributs peuvent être effectués en une seule analyse sur l'ensemble des instances d'entraînement. D'où la complexité du processus est de O (R).

Pour calculer la probabilité a priori, nous devons trouver la probabilité de chacune des étiquettes de classe dans l'ensemble des données train. La complexité de ce processus est de O (C).

Pour calculer la probabilité conditionnelle de chacun des attributs, nous devons trouver la probabilité d'occurrences de chaque état de tous les attributs conditionnés sur chacune des étiquettes de classe. D'où, le nombre total de valeurs calculées dans ce processus est donné par:

$$\sum_{i=1}^{N} C \cdot S_i \leq C(N \cdot S_{max})$$

Ainsi, la complexité de l'apprentissage du classifier est déterminée par le calcul de la probabilité conditionnelle, qui est donnée par $N \cdot C \cdot S_{max}$.

Après l'entraînement, les probabilités conditionnelles et les probabilités antérieures sont stockées et peuvent être récupérées pendant le processus de classification en temps constant. Lors de la classification des données de test, le modèle doit trouver les probabilités de chaque instance d'appartenir à chacune des étiquettes de classe. Calculer chacune de ces valeurs, l'opération boucle sur chaque variable de classe, sur chacun des attributs pour chaque instance de test. Par conséquent, le processus prend une complexité temporelle C N R.

Ainsi, la complexité temporelle du modèle Naïve Bayes est principalement déterminé par la complexité de trouver les probabilités conditionnelles

2.3.2 TAN:

Les étapes de calcul intensif qui interviennent dans le modèle TAN sont les suivantes : le calcul de la probabilité antérieure, le calcul de la probabilité conditionnelle pour chaque attribut en fonction de son parent et de sa classe, et le calcul de l'information mutuelle pour chaque paire d'attributs. et de sa classe et le calcul de l'information mutuelle pour chaque paire d'attributs.

- **Probabilité antérieure:**
La complexité du calcul de la probabilité antérieure pour le TAN est la même que pour le NB (i.e $O(C)$).

-**Information mutuelle:**
Le nombre total de paires pour N attributs est $(\frac{N(N-1)}{2})$ Pour chacune des paires, nous devons calculer les probabilités de toutes les combinaisons de tous les états, que chaque attribut peut prendre. En fixant la limite supérieure du nombre d'états des deux attributs à S_{max}le nombre d'opérations à effectuer pour trouver l'information mutuelle est donné par : Nombre total d'opérations $\leq (\frac{N(N-1)}{2})S_{max}^2$

-**Probabilité conditionnelle:** Le nombre d'opérations pour chaque probabilité conditionnelle, diffère légèrement du NB. Dans le TAN, nous devons trouver la probabilité conditionnelle de chacun des attributs, conditionnée par son parent et sa classe. Par conséquent, le nombre d'états du parent doit également être multiplié. Ceci est donné par :

for i ←1 to N
    calculate $P(A_i = x \mid A_j = y, C = z)$

où A_j est le parent de A_i

- $x \leftarrow (1 \text{ à } S_i)$
- $y \leftarrow (1 \text{ à } S_j)$

Par conséquent, la complexité temporelle du calcul de la probabilité conditionnelle est donnée par :

$$\sum_{i=1}^{N} C \cdot S_i \cdot S_j \leq N \cdot S_{max}^2 \cdot C$$

Ainsi, en observant la complexité des opérations de l'arbre, on peut conclure que la complexité du modèle TAN, est fortement influencée par le calcul de l'information mutuelle, qui augmente avec l'augmentation du nombre d'attributs.

En général, il s'agit d'un compromis entre la complexité et la performance. Les modèles qui ont des niveaux de dépendance plus élevés ont tendance à être plus performants, mais ils sont très complexes. meilleures performances, mais sont très complexes. D'autre part, les modèles qui ont des dépendances plus faibles ne sont pas toujours aussi performants que les modèles complexes. ne sont pas toujours aussi performants que les modèles complexes.

3 Les deux modèles en action

Comme énoncé à l'introduction, après l'implémentation de ces deux algorithmes, il était le moment les voir en action sur un jeu de données ayant multi-classes. Notre choix était le jeu de données digits de la librairie Sickit-Learn qui se compose de 1797 lignes et 64 colonnes. Chaque ligne représente une image de 8X8 d'un chiffre manuscrit entre 0 et 9.

Dans le cadre de la préparation de données à l'entraînement, nous avons éliminé certaines colonnes qui ont eu la même valeur car ceci engendrera des erreurs quand on calcule les corrélations entre les features. De plus, nous avons décidé d'ajouter une constante à la variance dans chaque classe afin d'éviter une variance nulle. Et en fin, nous avons fait un split (70%, 30%) pour avoir des données pour l'entraînement et d'autres pour savoir le taux de précision de chaque modèle.

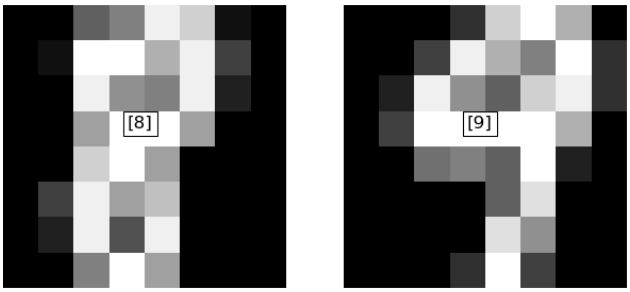


Figure 2: Le contenu du jeu de données

Performance de deux modèles:

Pour le modèle Naïve Bayes, le taux de précision était **87%** et la figure ci-dessous représente la matrix de confusion à l'issu de la prédiction. Bien sur quelqu'un pourrait considérer le roc_curve pour plus d'information sur la performance mais cet outil est limité à la tâche de classification binaire.

Figure 3: Confusion matrix (NB)

Quant au modèle TAN, plusieurs difficultés et ont été rencontrées lors de l'implémentation de ce modèle assez complexe. Par conséquent, certains compromis ont été faits et à la fin le modèle a eu une performance horrible sur le test pour le jeu de données digits, qui par sa nature, contient plusieurs aspects constants par classe. En tant que tel, nous avons considéré un autre jeu de données qui est iris. Le modèle a eu toujours un faible score par rapport NB et il a miss classifié 6 objets parmi 50 testés.

4 Conclusion

Logiquement, le TAN est censé fournir une performance meilleure que NB car la prise en compte d'une éventuelle dépendance ne peut être qu'une meilleure représentation de la réalité. Cependant, son exigence que l'arbre doive contenir N-1 arrêtes me semble contestable car, au moins dans notre cas d'application, plusieurs features ont eu aucune corrélation les une avec les autre et pourtant ont été gardées par l'algorithme de Kruskal.

La partie la plus difficile dans l'implémentation de l'algorithme TAN était la partie liée au calcul de $X_i|y, X_{parent}$ lors de la prédiction. Nous avons considéré que ce conditionnellement est entre deux loi normale et par conséquent est une loi normale mais l'itération sur X_{parent} pour évaluer la probabilité du vecteur X_i aurait été pénible et pour ça nous avons décidé d'omettre le terme $X_{parent} - \mu_{X_{parent}}$ dans cette loi conditionnelle ce qui peut être à l'origine du problème de performance du modèle.