

Intelligence Artificielle

TD 1 : Naive Bayes

Louis Annabi

27 janvier 2021

Le but de ce TP est de mettre en oeuvre une classification naïve bayésienne sur deux applications différentes : la reconnaissance visuelle de chiffres manuscrits

1 Algorithme

L'algorithme de classification naïve bayésienne se base sur un modèle probabiliste simplifié des données du problème. Une donnée du problème \mathbf{x} est décrite par différentes propriétés (x_1, \dots, x_n) , par exemple les valeurs des pixels pour une image, ou la présence ou non d'un mot dans un texte.

Le modèle probabiliste "naïf" va considérer que ces différentes propriétés sont indépendantes sachant la classe de \mathbf{x} , notée y :

$$p(\mathbf{x}|y) = p(x_1, \dots, x_n|y) = \prod_{i=1}^n p(x_i|y) \quad (1)$$

Ce modèle probabiliste simplifié, associé à la règle de Bayes, va nous permettre de classer des données avec la formule suivante :

$$p(y|\mathbf{x}) = \frac{p(y)p(\mathbf{x}|y)}{p(\mathbf{x})} \quad (2)$$

$$= \frac{p(y) \prod_{i=1}^n p(x_i|y)}{p(\mathbf{x})} \quad (3)$$

Pour une donnée \mathbf{x} , cette formule nous permet de calculer la probabilité associée à chaque classe possible, à partir des paramètres du modèle que sont $p(y)$ et les $p(x_i|y)$. On remarquera que dans cette formule, le dénominateur ne dépend pas de y . On n'a donc pas besoin de le calculer : la classe prédite sera le y pour lequel le numérateur est le plus grand.

$$\hat{y} = \operatorname{argmax}_y (p(y) \prod_{i=1}^n p(x_i|y)) \quad (4)$$

2 Application à la reconnaissance de chiffres

Dans ce premier exercice, on applique l'algorithme sur de la classification sur la base de données MNIST, téléchargeable sur <http://yann.lecun.com/exdb/mnist/>.

L'hypothèse de la classification naïve bayésienne est que les valeurs des pixels d'une image sachant la classe de l'image sont indépendants. On modélisera la loi de probabilité sur les valeurs des pixels sachant chaque classe $p(x_i|y)$ par des lois gaussiennes univariées, et la loi de probabilité a priori sur les classes $p(y)$ par une loi de Bernoulli généralisée (une probabilité associée aux 10 catégories possibles, qui se somment à 1).

1. Comment estimer les paramètres de ce modèle à partir des données d'entraînement?
2. Implémenter l'estimation des paramètres du modèle
3. Implémenter une fonction classifiant une entrée x en fonction des paramètres du modèle
4. Afficher la matrice de confusion obtenue après apprentissage
5. Quelle est la précision de l'algorithme?
6. Afficher la matrice de paramètres $p(x|y_k)$ correspondant à chaque classe y_k et commenter

3 Application à la détection de spam

Dans ce deuxième exercice, on applique l'algorithme sur de la classification binaire sur une base de données de SMS, téléchargeable sur <https://archive.ics.uci.edu/ml/datasets/sms+spam+collection>.

Dans cette tâche de classification, nous allons considérer que chaque SMS est une entrée x dont les propriétés (x_1, \dots, x_n) correspondent aux mots qu'il contient. Nous faisons donc l'hypothèse que les probabilités de présence des différents mots dans un SMS sont indépendants sachant la classe du SMS (spam ou ham). On modélisera les lois de probabilité $p(x_i|y)$ par des lois de Bernoulli, et la loi de probabilité a priori qu'un SMS soit un spam par une autre loi de Bernoulli.

1. Comment estimer les paramètres de ce modèle à partir des données d'entraînement?
2. Implémenter l'estimation des paramètres du modèle
3. Implémenter une fonction classifiant une entrée x en fonction des paramètres du modèle
4. Afficher la matrice de confusion obtenue après apprentissage
5. Quelle est la précision de l'algorithme?
6. Afficher les mots ayant la plus forte probabilité d'apparaître dans du spam

4 Rendu

Votre code et un rapport de 2 à 5 pages répondant aux questions et contenant les affichages demandés ainsi que vos analyses sont à rendre le week-end suivant le TP.