

[Next](#)
[Up](#)
[Previous](#)
[Contents](#)
[Index](#)

Suivant : [Relation avec l'unigramme multinomial Niveau supérieur](#) : [Classification de texte et Naïf](#) **Précédent :** [Le problème de classification de texte](#) [Sommaire](#) [Index](#)

Classification de texte naïve de Bayes

La première méthode d'apprentissage supervisé que nous introduisons est le *multinôme Naive Bayes* ou *NB multinomial* modèle, une méthode d'apprentissage probabiliste. La probabilité d'un document d être en classe c est calculé comme

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (113)$$

où $P(t_k|c)$ est la probabilité conditionnelle de terme t_k apparaissant dans un document de classe c . Nous interprétons $P(t_k|c)$ comme mesure de la quantité de preuves t_k contribue que c est la bonne classe. $P(c)$ est la probabilité a priori qu'un document apparaisse dans la classe c . Si les termes d'un document n'indiquent pas clairement preuve d'une classe par rapport à une autre, nous choisissons celle qui a une probabilité a priori plus élevée. $\langle t_1, t_2, \dots, t_{n_d} \rangle$ sont les jetons dans d qui font partie du vocabulaire que nous utilisons pour classement et n_d est le nombre de ces jetons dans d . Par exemple, $\langle t_1, t_2, \dots, t_{n_d} \rangle$ pour le document d'une phrase Beijing et Taipei rejoignent l'OMC pourrait être $\langle \text{Beijing}, \text{Taipei}, \text{join}, \text{WTO} \rangle$, avec $n_d = 4$, si nous traitons les termes et les comme des mots vides.

Dans la classification de texte, notre objectif est de trouver le *meilleur* classe pour le document. La meilleure classe de la classification NB est le très probablement ou *maximum a posteriori* (*CARTE*) classe c_{map} :

$$c_{map} = \arg \max_{c \in C} \hat{P}(c|d) = \arg \max_{c \in C} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c). \quad (114)$$

Nous écrivons \hat{P} pour P parce que nous ne connaissons pas le vrai valeurs des paramètres $P(c)$ et $P(t_k|c)$, mais estimez-les à partir du ensemble d'entraînement comme nous le verrons dans un instant.

Dans l'équation 114, de nombreuses probabilités conditionnelles sont multiplié, un

pour chaque position $1 \leq k \leq n_d$. Cela peut entraîner un sous-dépassement en virgule flottante. Ce vaut donc mieux faire le calcul en ajoutant logarithmes des probabilités au lieu de multiplier probabilités. La classe avec la probabilité log la plus élevée le score est toujours le plus probable ; $\log(xy) = \log(x) + \log(y)$ et la fonction logarithme est monotone. D'où, la maximisation qui est effectivement effectuée dans la plupart implémentations de NB est :

$$c_{map} = \arg \max_{c \in C} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c)]. \quad (115)$$

L'équation 115 a une interprétation simple. Chaque paramètre conditionnel $\log \hat{P}(t_k|c)$ est un poids qui indique la qualité d'un indicateur t_k est pour c . De même, le précédent $\log \hat{P}(c)$ est un poids qui indique la fréquence relative de c . Des cours plus fréquents sont plus susceptibles d'être les classe correcte que peu fréquente Des classes. le la somme des pondérations log prior et terme est alors une mesure de la façon dont il y a beaucoup de preuves que le document est dans la classe, et équation 115 sélectionne la classe pour laquelle nous avons le plus de preuves.

Nous travaillerons dans un premier temps avec cette interprétation intuitive de le modèle NB multinomial et reporter une dérivation formelle à Article 13.4.

Comment estimer les paramètres $\hat{P}(c)$ et $\hat{P}(t_k|c)$? Nous essayons d'abord l'*estimation du maximum de vraisemblance* (MLE; probtheory), qui est simplement la fréquence relative et correspond à la valeur la plus probable de chaque paramètre donné les données d'entraînement. Pour les a priori cette estimation est :

$$\hat{P}(c) = \frac{N_c}{N}, \quad (116)$$

où N_c est le nombre de documents en classe c et N est le nombre total de documents.

Nous estimons la probabilité conditionnelle $\hat{P}(t|c)$ comme la fréquence relative du terme t dans documents appartenant à la classe c :

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}, \quad (117)$$

où T_{ct} est le nombre d'occurrences de t dans documents de formation de classe c , dont plusieurs occurrences d'un terme dans un document. Nous avons fait le *hypothèse d'indépendance positionnelle* ici, dont nous parlerons plus en détail dans

la section suivante : T_{ct} est un nombre d'occurrences dans tous les postes k dans les documents de l'ensemble de formation. Ainsi, nous ne calculons pas différentes estimations pour différents positions et, par exemple, si un mot apparaît deux fois dans un document, dans des positions k_1 et k_2 , ensuite $\hat{P}(t_{k_1}|c) = \hat{P}(t_{k_2}|c)$.

Le problème avec l'estimation MLE est qu'elle est nulle pour un combinaison terme-classe qui ne s'est pas produite dans la formation Les données. Si le terme OMC dans les données de formation uniquement s'est produit dans des documents en Chine, puis les estimations de MLE pour les autres classes, par exemple UK, sera zéro:

$$\hat{P}(\text{WTO}|\text{UK}) = 0. \quad (118)$$

Maintenant, le document d'une phrase Grande-Bretagne est un membre de l'OMC obtiendra une probabilité conditionnelle de zéro pour le Royaume-Uni car nous multiplions le conditionnel probabilités pour tous les termes de Équation 113. Il est clair que le modèle doit attribuer une forte probabilité à la classe UK car le terme Bretagne se produit. Le problème est que la probabilité nulle car l'OMC ne peut pas être "conditionnée", non peu importe la force des preuves pour la classe UK à partir d'autres fonctionnalités. L'estimation est de 0 à cause de *parcimonie* : les données d'entraînement ne sont jamais assez grandes représenter adéquatement la fréquence des événements rares, par Exemple, la fréquence des OMC survenant dans Documents britanniques.

```

TRAINMULTINOMIALNB(C, ID)
1  V ← EXTRACTVOCABULARY(ID)
2  N ← COUNTDOCS(ID)
3  for each c ∈ C
4  do Nc ← COUNTDOCSINCLASS(ID, c)
5     prior[c] ← Nc / N
6     textc ← CONCATENATETEXTOFALLDOCSINCLASS(ID, c)
7     for each t ∈ V
8     do Tct ← COUNTTOKENSOFTERM(textc, t)
9     for each t ∈ V
10    do condprob[t][c] ←  $\frac{T_{ct}+1}{\sum_{t'} (T_{ct'}+1)}$ 
11 return V, prior, condprob

APPLYMULTINOMIALNB(C, V, prior, condprob, d)
1  W ← EXTRACTTOKENSFROMDOC(V, d)
2  for each c ∈ C
3  do score[c] ← log prior[c]
4     for each t ∈ W
5     do score[c] += log condprob[t][c]
6 return arg maxc ∈ C score[c]

```

Figure 13.2 : Algorithme naïf de Bayes (modèle multinomial) : Formation et tests.

Pour éliminer les zéros, on utilise *ajoute un ou Laplace lissage*, qui consiste simplement ajoute un à chaque comptage (cf. Section 11.3.2) :

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'} \quad (119)$$

où $B = |V|$ est le nombre de termes du vocabulaire. Add-one lissage peut être interprété comme un a priori uniforme (chaque terme apparaît une fois pour chaque classe) qui est ensuite mis à jour comme preuve à partir des données d'entraînement. Notez que ceci est une probabilité a priori d'occurrence d'un *terme* par opposition à la probabilité a priori d'une *classe* que nous estimons dans Équation 116 au niveau du document.

Nous avons maintenant introduit tous les éléments dont nous avons besoin pour la formation et appliquer un classificateur NB. Le complet algorithme est décrit dans Illustration 13.2.

Tableau 13.1 : Données pour le paramètre exemples d'estimation.

	docID	mots dans le document	dans $c = \bar{c}$ Chine?
ensemble d'entraînement	1	Chinois Pékin Chinois	Oui
	2	chinois chinois Shanghai	Oui
	3	Chinese Macao	Oui
	4	Tokyo Japon Chinois	non
ensemble d'essai	5	Chinois Chinois Chinois Tokyo Japon	?

Exemple travaillé. Pour l'exemple du tableau 13.1, le multinomial les paramètres dont nous avons besoin pour classer le document de test sont les antérieurs $\hat{P}(c) = 3/4$ et $\hat{P}(\bar{c}) = 1/4$ et les probabilités conditionnelles suivantes :

$$\begin{aligned} \hat{P}(\text{Chinese}|c) &= (5 + 1)/(8 + 6) = 6/14 = 3/7 \\ \hat{P}(\text{Tokyo}|c) = \hat{P}(\text{Japan}|c) &= (0 + 1)/(8 + 6) = 1/14 \\ \hat{P}(\text{Chinese}|\bar{c}) &= (1 + 1)/(3 + 6) = 2/9 \\ \hat{P}(\text{Tokyo}|\bar{c}) = \hat{P}(\text{Japan}|\bar{c}) &= (1 + 1)/(3 + 6) = 2/9 \end{aligned}$$

Les dénominateurs sont $(8 + 6)$ et $(3 + 6)$ car les longueurs de text_c et $\text{text}_{\bar{c}}$ sont 8 et 3, respectivement, et parce que la constante B dans équation 119 est 6 car le vocabulaire se compose de six termes.

On obtient alors :

$$\hat{P}(c|d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003.$$

$$\hat{P}(\bar{c}|d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001.$$

Ainsi, le classificateur attribue le document d'essai à $c = \text{Chine}$. La raison pour cette décision de classement est que les trois occurrences du positif indicateur chinois en d_5 l'emportent sur les occurrences de les deux indicateurs négatifs Japon et Tokyo. **Fin de l'exemple travaillé.**


Tableau 13.2 : Temps de formation et de test pour NB.

	mode	complexité temporelle	
	formation	$\Theta(D L_{ave} + C V)$	
	essai	$\Theta(L_a + C M_a) = \Theta(C M_a)$	

Quelle est la complexité temporelle de NB ? La complexité de calculer les paramètres est $\Theta(|C||V|)$ car le jeu de paramètres se compose de $|C||V|$ probabilités conditionnelles et $|C|$ antérieurs. le prétraitement nécessaire au calcul des paramètres (extraire le vocabulaire, compter les termes, etc.) peuvent être fait en un seul passage à travers les données de formation. Le temps la complexité de ce composant est donc $\Theta(|D|L_{ave})$, où $|D|$ est le nombre de documents et L_{ave} est la longueur moyenne d'un document.

Nous utilisons $\Theta(|D|L_{ave})$ comme notation pour $\Theta(T)$ ici, où T est la longueur du recueil de formation. C'est non standard ; $\Theta(\cdot)$ n'est pas défini pour une moyenne. Nous préférons exprimer le temps complexité en termes de $|D|$ et L_{ave} parce que ce sont les principales statistiques utilisées pour caractérisent les collections de formation.

La complexité temporelle de A PPLY M ULTINOMIAL NB in figure 13.2 est $\Theta(|C|L_a)$.

L_a et M_a sont les nombres de jetons et types, respectivement, dans le test documenter . A PPLY M ULTINOMIAL NB peut être modifié pour être $\Theta(L_a + |C|M_a)$ (Exercice 13.6). Enfin, en supposant que la longueur des documents de test est bornée, $\Theta(L_a + |C|M_a) = \Theta(|C|M_a)$ car $L_a < b|C|M_a$ pour une constante fixe b . 

Le tableau 13.2 résume les complexités temporelles. En général, nous avons $|C||V| < |D|L_{ave}$, donc les deux formations et la complexité des tests sont linéaires dans le temps qu'il faut pour scanner les données. Parce que nous devons regarder les données à moins une fois, on peut dire que NB a un temps optimal complexité.

Son efficacité est l'une des raisons pour lesquelles NB est une méthode de classification de texte populaire.

Sous-sections

- [Relation avec le modèle de langage unigramme multinomial](#)

[Next](#) [Up](#) [Previous](#) [Contents](#) [Index](#)

Suivant : [Relation avec l'unigramme multinomial Niveau supérieur : Classification de texte et Naïf](#) **Précédent :** [Le problème de classification de texte](#) [Sommaire](#)
[Index](#)

© 2008 Cambridge University Press

Ceci est une page générée automatiquement. En cas d'erreurs de formatage, vous pouvez consulter l' [édition PDF](#) du livre.

2009-04-07