Classification Bayésienne Naïve

Classification Bayésienne Naïve

Introduction au théorème de Bayes

Les problèmes de Classification

Les problèmes de classification visent à assigner des *instances* à des *classes*, en utilisant des *propriétés* des instances. Par exemple, en connaissant les mesures de différentes parties de la fleur (les *propriétés*), peut on assigner un échantillon (une *instance*) à une espèces du genre *Iris* (une *classe*)?

Le raisonnement générale des méthodes de classification est le suivant: plus les propriétés d'une instance sont proches des propriétés d'une classe, plus la probabilité que cette instance appartienne à cette classe est élevée. De manière plus mathématique, on définit une instance comme:

$$\mathbf{x} = (x_1, \ldots, x_n)$$

$\mathbf{x}$ est l'instance, et x_k est sa k-ième propriété.

Le problème que l'on cherche a résoudre est donc de mesurer

$$P(C_m|\mathbf{x}) = P(C_m|x_1, x_2, \ldots, x_n)$$

Ici,

$$C_m$$

dénote le fait d'appartenir à la m-ième classe.

La Classification Bayésienne Naïve

La Classification Bayésienne Naïve (CBN) permet de résoudre des problèmes de Classification, en utilisant une approche Bayesienne, et en faisant des hypothèses qui sont Naïves (mais bien utiles). Nous allons procéder en deux étapes. D'abord, construire le modèle probabiliste qui décrit le problème de classification; ensuite, construire le classificateur, qui permet de passer de la probabilité à une *décision* sur la classe à laquelle une instance appartient.

Construction du modèle probabiliste

Dans la CBN, on utilise le théorème de Bayes pour résoudre les problèmes mentionnés plus haut. La question centrale de la classification peut donc être ré-exprimée comme:

$$P(C_m|\mathbf{x}) = \frac{P(C_m) \times P(\mathbf{x}|C_m)}{P(\mathbf{x})}$$

Dans la pratique, la valeur du dénominateur (*preuve*) n'est pas intéressante, puisqu'elle ne dépend pas de la classe en question (et les valeurs de $\mathbf{x}$ sont connues et constantes). On peut donc simplifier le problème comme étant:

$$P(C_m|\mathbf{x}) \propto P(C_m) \times P(\mathbf{x}|C_m)$$

Cette quantité est équivalente à

$$P(\mathbf{x}, C_m) = P(x_1, \ldots, x_n, C_m)$$

et en appliquant les règles de calcul sur les probabilités, on peut ré-écrire ceci comme

$$P(x_1|x_2, \ldots, C_m)P(x_2, \ldots, C_m)$$

puis comme

$$P(x_1|x_2, \ldots, C_m)P(x_2|x_3, \ldots, C_m)P(x_3, \ldots, C_m)$$

jusqu'à avoir

$$P(x_1|x_2, \ldots, C_m) \ldots P(x_{n-1}|x_n, C_m)P(x_n|C_m)P(C_m)$$

Ce qui, à première vue, complexifie considérablement les choses. **Mais!** On utilise une approche dite *naïve*, dans laquelle on suppose que toutes les propriétés des instances sont *indépendantes*. Autrement dit, si x_1 n'a aucun effet sur $x_2, \ldots$, alors on peut écrire

$$P(x_1|x_2, \ldots, C_m) = P(x_1|C_m)$$

Et par conséquent, l'expression ci-dessus se simplifie:

$$P(x_1|C_m)P(x_2|C_m) \ldots P(x_n|C_m)P(C_m)$$

On peut donc finalement simplifier le problème de classification de manière considérable:

$$P(C_m|\mathbf{x}) \propto P(C_m) \prod_{i=1}^{n} P(x_i|C_m)$$

Construction du classificateur

Dans les étapes précédentes, nous avons construit le modèle probabiliste, c'est à dire que nous sommes en mesure de calculer la forme générale du problème $P(C_m|\mathbf{x})$, c'est à dire la probabilité que $\mathbf{x}$ appartienne à la m-ième classe sachant les propriétés de $\mathbf{x}$. Un *classificateur* utilise cette information pour retourner une valeur unique, qui correspond au numéro de la classes à laquelle $\mathbf{x}$ appartient.

Dans la CBN, on utilise en générale un classificateur très simple: *argmax*. On calcule la probabilité que $\mathbf{x}$ appartienne à chacune des classes possibles, et on assigne $\mathbf{x}$ à la classe qui a la probabilité la plus élevée. Plus formellement, on représente par $\hat{y}$ la classe à laquelle l'instance $\mathbf{x}$ appartient,

$$\hat{y} = \text{argmax}_{k \in 1 \ldots K} P(C_m) \prod_{i=1}^{n} P(x_i|C_m)$$

Application

Nous allons tenter de déterminer à quelle espèce du genre *Iris* une plante dont on connaît les mesures appartient. R possède un jeu de données avec toutes les informations nécessaires:

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Nous allons garder la première ligne pour l'identifier, et utiliser l'ensemble des autres lignes pour faire le travail de prédiction.

```
problem <- iris[1,]
training <- iris[-1,]
```

Exercice

Quelle est l'instance, quelles sont les classes, et quelles sont les propriétés?

La première étape consiste a déterminer comment on peut estimer

$$P(x = v|C_m)$$

, c'est à dire la probabilité d'observer une valeur

$$c$$

pour la propriété

$$x$$

si on appartient à la classe

$$m$$

. Toutes les propriétés sont représentées par des variables quantitatives, et on va supposer que la distribution de ces variables est suffisament normale pour être représentée par une Gaussienne.

La formule donnant la distribution de probabilité d'une loi normale, paramétrée par sa moyenne et sa variance, est (comme nous le savons tous sans aller le chercher sur Wikipédia)

$$P(x = v|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \times \exp\left[-\frac{(v - \mu_C)^2}{2\sigma_C^2}\right]$$

Dans cette formule,

$$\mu_C$$

est la valeur moyenne de la propriété

$$x$$

pour les instance de la classe

$$C$$

, et

$$\sigma_C^2$$

est leur variance.

Il faut donc calculer ces deux quantités pour chaque espèce. Nous allons donc passer le jeu de données au format long.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1.9000     v purrr   0.3.2
## v tibble  2.1.3          v dplyr   0.8.3
## v tidyr   0.8.3          v stringr 1.4.0
## v readr   1.3.1          v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
long_training <- training %>%
  gather(variable, value, Sepal.Length:Petal.Width)


summary_statistics <- long_training %>%
  group_by(Species, variable) %>%
  summarize(mean = mean(value),
            sd = sd(value))

head(summary_statistics)
```

```
## # A tibble: 6 x 4
## # Groups:   Species [2]
##   Species    variable      mean    sd
##   <fct>      <chr>        <dbl> <dbl>
## 1 setosa     Petal.Length 1.46  0.175
## 2 setosa     Petal.Width  0.247 0.106
## 3 setosa     Sepal.Length 5.00  0.356
## 4 setosa     Sepal.Width  3.43  0.383
## 5 versicolor Petal.Length 4.26  0.470
## 6 versicolor Petal.Width  1.33  0.198
```

Une fois cette étape effectuée, on peut calculer la probabilité qu'une mesure donnée appartienne à une classe, avec la fonction `dnorm` (`R` a énormément de fonctions déjà écrites pour mesurer les densités de probabilités).

La fonction `dnorm` (`?dnorm` pour en savoir plus) prend trois arguments: la valeur, la moyenne de la distribution, et la déviation standard. Nous allons commencer par une illustration: quelle est la probabilité que la longeur du pétale `Petal.Length` de notre instance appartienne à l'espèce `setosa`?

```
setosa_petal_length <- filter(summary_statistics,
                              Species == "setosa",
                              variable == "Petal.Length")

setosa_petal_length
```

```
## # A tibble: 1 x 4
## # Groups:   Species [1]
##   Species variable      mean    sd
##   <fct>   <chr>        <dbl> <dbl>
## 1 setosa  Petal.Length  1.46 0.175
```

```
proba_petal_length_setosa <- dnorm(
  problem$`Petal.Length`,
  setosa_petal_length$mean,
  setosa_petal_length$sd
  )

proba_petal_length_setosa
```

```
## [1] 2.13302
```

Remarquez au passage que cette valeur est supérieure à 1 – c'est parce qu'il s'agit d'une densité de probabilité, et pas de la probabilité elle même.

On voudrait maintenant automatiser un petit peu ce processus… La quantité que l'on cherche a mesurer est toujours la même: quelle est la probabilité de la classe sachant la valeur de la propriété. Nous allons résoudre ce problème avec une fonction:

```
proba_class_knowing_feature <- function(class_name, feature_name, summary, problem){
  class_feature <- subset(summary,
    (Species == class_name) & (variable == feature_name)
    )
  proba_feature_class <- dnorm(
    problem[1, feature_name],
    class_feature$mean,
    class_feature$sd)
  return(proba_feature_class)
}

proba_class_knowing_feature("setosa", "Petal.Length", summary_statistics, problem)
```

```
## [1] 2.13302
```

On peut vérifier que la fonction retourne la même valeur que lorsque nous avons fait le calcul étape par étape.

On peut maintenant aller calculer l'ensemble des probabilités par propriété et par classe:

```
proba_class_feature <- summary_statistics %>%
  mutate(id = map2_dbl(Species, variable, proba_class_knowing_feature,
                       summary = summary_statistics, problem = problem))

head(proba_class_feature)
```

```
## # A tibble: 6 x 5
## # Groups:   Species [2]
##   Species    variable      mean    sd            id
##   <fct>      <chr>        <dbl> <dbl>         <dbl>
## 1 setosa     Petal.Length 1.46  0.175 2.13
## 2 setosa     Petal.Width  0.247 0.106 3.41
## 3 setosa     Sepal.Length 5.00  0.356 1.08
## 4 setosa     Sepal.Width  3.43  0.383 1.02
## 5 versicolor Petal.Length 4.26  0.470 0.00000000768
## 6 versicolor Petal.Width  1.33  0.198 0.000000184
```

À ce stade, nous avons *presque* toutes les informations pour prédire la classe à laquelle notre échantillon appartient. Il reste seulement a mesurer la probabilité de chaque classe. Cette étape fait appel à notre intuition biologique et à notre connaissance du problème. Si on veut ne pas prendre de décision, on peut supposer que les trois espèces (`setosa`, `versicolor` et `virginica`) ont la même probabilité (par exemple,

dans un peuplement d'Iris, on trouvera les trois avec la même fréquence). On peut aussi regarder dans les données combien de chaque espèce on a, et utiliser ceci comme une information sur leur abondance – si on a mesuré trois fois plus de *I. seticolor*, peut-être que cette espèce est plus abondante dans la nature, et qu'un échantillon pris au hasard a une plus forte probabilité d'appartenir à cette espèce.

```
table(training$Species)
```

```
##
##     setosa versicolor  virginica
##         49         50         50
```

Hmm. Les trois espèces semblent équiprobables dans ce jeu de donnée. Nous allons donc donner à chaque classe m la même probabilité:

$$P(C_m) = \frac{1}{3}$$

.

Notez au passage que c'est une force des méthodes Bayesiennes: on peut intégrer de l'information prioritaire sur les données, qui reflète notre connaissance biologique.

Nous allons maintenant construire le classificateur. La formule exacte du classificateur est:

```
P(species) P(petal length | species) P(petal width | species) ...
```

En `R`, on peut l'écrire sous la forme suivante:

```
species <- as.character(unique(iris$Species))

species_proba <- rep(1/length(species), length(species))

proba_class_feature %>%
  knitr::kable(.)
```

Species	variable	mean	sd	id
setosa	Petal.Length	1.4632653	0.1752307	2.1330196
setosa	Petal.Width	0.2469388	0.1062663	3.4052402
setosa	Sepal.Length	5.0040816	0.3558787	1.0810197
setosa	Sepal.Width	3.4265306	0.3828487	1.0230248
versicolor	Petal.Length	4.2600000	0.4699110	0.0000000
versicolor	Petal.Width	1.3260000	0.1977527	0.0000002
versicolor	Sepal.Length	5.9360000	0.5161711	0.2082111
versicolor	Sepal.Width	2.7700000	0.3137983	0.0849365
virginica	Petal.Length	5.5520000	0.5518947	0.0000000
virginica	Petal.Width	2.0260000	0.2746501	0.0000000
virginica	Sepal.Length	6.5880000	0.6358796	0.0405935
virginica	Sepal.Width	2.9740000	0.3224966	0.3271303

Et c'est maintenant le moment de vérité – on peut aller regarder dans la variable `problem` à quelle espèce notre échantillon appartenait:

```
as.character(problem$Species)
```

```
## [1] "setosa"
```

Ça marche!