WSI-7-naive-bayes-classifier

Piotr Lenczewski

Styczeń 2024

1 Opis algorytmu

Naiwny klasyfikator bayesowski to algorytm klasyfikacji oparty na teoretycznych podstawach statystycznych, zwłaszcza na regule Bayesa. Klasyfikatory tego typu są "naiwne", ponieważ zakładają niezależność między cechami, co oznacza, że każda cecha w próbce jest traktowana jako niezależna od pozostałych.

1.1 Cel algorytmu

Załóżmy, że mamy zbiór danych składający się z różnych klas (etykiet) i cech opisujących te dane. Naiwny klasyfikator bayesowski służy do przewidywania klasy nowego obiektu na podstawie jego cech.

1.2 Sposób przewidywania klasy obiektu

Wyliczone zostają prawdopodobieństwa, że obiekt należy do poszczególnych klas pod warunkiem jego cech. Oznaczmy klasę jako C i cechy jako $x_1, x_2, \ldots, x_n$. Prawdopodobieństwo warunkowe, że obiekt X należy do klasy C, można zapisać jako:

$$P(C|x_1, x_2, \ldots, x_n) = \frac{P(C) \cdot P(x_1|C) \cdot P(x_2|C) \cdot \ldots \cdot P(x_n|C)}{P(x_1) \cdot P(x_2) \cdot \ldots \cdot P(x_n)}$$

Naiwny klasyfikator bayesowski zakłada, że cechy są niezależne, co skraca powyższe równanie do:

$$P(C|x_1, x_2, \ldots, x_n) = P(C) \cdot \prod_{i=1}^{n} P(x_i|C)$$

Wynikiem klasyfikacji jest najbardziej prawdopodobna klasa.

1.3 Wykorzystanie danych treningowych

Aby skożystać z powyższych wzorów należy najpierw, korzystając z danych testowych, wyliczyć poszczególne prawdopodobieństwa: klasy, cech pod warunkiem klasy. By to osiągnąć przyjmę, że rozkładem cech obiektu jest rozkład normalny. Załóżmy, że mamy N przykładów treningowych, z których N_C należy do klasy C. Wtedy prawdopodobieństwo klasy:

$$P(C) = \frac{N_C}{N}$$

Aby obliczyć prawdopodobieństwa cech pod warunkiem klas należy najpierw obliczyć wartość oczekiwaną, oraz wariancję:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Wtedy prawdopodobieństwo cechy pod warunkiem klasy możemy policzyć ze wzzoru:

$$P(x_i|C) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right)$$

1.4 Wykorzystanie logarytmu w obliczeniach

Aby usprawnić obliczenia wykorzystuje się logarytm naturalny. Wtedy:

$$\ln\left(P(C|x_1, x_2, \ldots, x_n)\right) = \ln\left(P(C)\right) + \sum_{i=1}^{n} \ln\left(P(x_i|C)\right)$$

Gdy dla $C_1, C_2, \ldots, C_m$, $\ln(P(C_m|x_1, x_2, \ldots, x_n))$ ma największą wartość, to $P(C_m|x_1, x_2, \ldots, x_n)$ również ma największą wartość. Pozawala to na uniknięcie operacji mnożenia.

2 Testowanie algorytmu

2.1 Założenia

Do testów wykorzystam zbiór danych `https://archive.ics.uci.edu/ml/datasets/iris`. Zostanie on podzielony na zbiór treningowy oraz testowy w stosunku 8:2. Dla uśrednienia wyników algorytm zostanie przetestowany dla 100 różnych podziałów zbioru danych.

2.2 Testowane wartości

Zamierzam obliczyć kilka wskaźników dla wyniku działania algorytmu:

- accuracy (dokładność)
$$\text{Accuracy} = \frac{TP_{\text{c1}} + TP_{\text{c2}} + TP_{\text{c3}}}{\text{Total Instances}}$$
- precision (precyzja)
$$\text{Precision}_{\text{ci}} = \frac{TP_{\text{ci}}}{TP_{\text{ci}} + FP_{\text{ci}}}$$

$$\text{Precision}_{\text{weighted}} = \frac{(\text{Instances}_{\text{c1}} \times \text{Precision}_{\text{c1}}) + (\text{Instances}_{\text{c2}} \times \text{Precision}_{\text{c2}}) + (\text{Instances}_{\text{c3}} \times \text{Precision}_{\text{c3}})}{\text{Total Instances}}$$

- recall (czułość)

$$\text{Recall}_{\text{ci}} = \frac{TP_{\text{ci}}}{TP_{\text{ci}} + FN_{\text{ci}}}$$

$$\text{Recall}_{\text{weighted}} = \frac{(\text{Instances}_{\text{c1}} \times \text{Recall}_{\text{c1}}) + (\text{Instances}_{\text{c2}} \times \text{Recall}_{\text{c2}}) + (\text{Instances}_{\text{c3}} \times \text{Recall}_{\text{c3}})}{\text{Total Instances}}$$

- F1 score (miara F1)

$$\text{F1}_{\text{ci}} = 2 \times \frac{\text{Precision}_{\text{ci}} \times \text{Recall}_{\text{ci}}}{\text{Precision}_{\text{ci}} + \text{Recall}_{\text{ci}}}$$

$$\text{F1}_{\text{weighted}} = \frac{(\text{Instances}_{\text{c1}} \times \text{F1}_{\text{c1}}) + (\text{Instances}_{\text{c2}} \times \text{F1}_{\text{c2}}) + (\text{Instances}_{\text{c3}} \times \text{F1}_{\text{c3}})}{\text{Total Instances}}$$

- confusion matrix (macierz pomyłek)

		Predicted		
		Class 1	Class 2	Class n
Actual	Class 1	n_{11}	n_{21}	n_{m1}
	Class 2	n_{12}	n_{22}	n_{m2}
	...	...	...	...
	Class n	n_{1m}	n_{2m}	n_{mm}

3 Wyniki testów

3.1 Przykładowe wartości współczynników

Wartości wskaźników dla różnych podziałów tych samych danych wejściowych

Wskaźnik	Wartości		
accuracy	0.9333	1.0000	0.9667
precision	0.9436	1.0000	0.9704
recall	0.9333	1.0000	0.9667
f1 score	0.9324	1.0000	0.9667

3.2 Uśrednione wartości współczynników

Średnia wartość wskaźników dla 100 podziałów tych samych danych wejściowych

Wskaźnik	Średnia wartość
accuracy	0.9537
precision	0.9583
recall	0.9537
f1 score	0.9537

3.3 Uśredniona macierz pomyłek

Średnia ilość obiektów danej klasy. Obiekty są tutaj podzielone na klasy faktyczne (wiersze), przydzielone przez algorytm (kolumny)

		Predicted		
		Iris-setosa	Iris-versicolor	Iris-virginica
Actual	Iris-setosa	9.95	0	0
	Iris-versicolor	0	9.33	0.63
	Iris-virginica	0	0.76	9.33

4 Wnioski

- W zależności od podziału zbioru danych algorytm jest bezbłędny lub myli się w pojedyńczych przypadkach,
- Uśredniając algorytm jest skuteczny w ok. 95% przypadków
- Niektóre klasy są trudniejsze do klasyfikacji od innych. Iris-setosa jest zawsze klasyfikowana prawidłowo, w przeciwieńswie do pozostałych,
- Może to wskazywać na większe podobieństwo w obrębie danych cech gatunków Iris-versicolor i Iris-virginica.