Metody systemowe i decyzyjne w informatyce

Laboratorium – Python – Zadanie nr 2

κ-NN i Naive Bayes

autorzy: M. Zięba, J.M. Tomczak, A. Gonczarek, S. Zaręba, J. Kaczmar

Cel zadania

Celem zadania jest implementacja klasyfikatorów κ-NN oraz Naive Bayes w zadaniu analizy dokumentów tekstowych.

Zadanie klasyfikacji dokumentów tekstowych

Rozważmy problem klasyfikacji dokumentu tekstowego $\mathcal{T}$ do jednej z kategorii tematycznych. Każdy dokument tekstowy opisany jest za pomocą wektora cech $\mathbf{x} = (\phi^1(\mathcal{T}), \ldots, \phi^D(\mathcal{T}))^{\mathrm{T}}$, gdzie każda cecha $\phi^d(\mathcal{T}) \in \{0, 1\}$ określa, czy d-te słowo występuje w dokumencie $\mathcal{T}$, tj. $\phi^d(\mathcal{T}) = 1$, czy też nie, $\phi^d(\mathcal{T}) = 0$. Dla każdego dokumentu należy rozwiązać problem klasyfikacji z wieloma klasami $y \in \{0, 1, 2, 3\}$, gdzie każda wartość określa grupę tematyczną (0 – *computer*, 1 – *recreation*, 2 – *science*, 3 – *talk*).

Zadanie klasyfikacji nowego dokumentu tekstowego $\mathbf{x}^{new}$ do jednej z grup tematycznych polega na wyznaczeniu prawdopodobieństwa $p(y|\mathbf{x}^{new})$, a następnie wyboru tej klasy, dla której prawdopodobieństwo warunkowe jest największe:

$$y^* = \arg\max_y p(y|\mathbf{x}^{new}). \tag{1}$$

Kluczową wielkością w problemie klasyfikacji jest rozkład warunkowy $p(y|\mathbf{x})$, dlatego jest on celem modelowania. Zauważmy, że wielkość tę można modelować co najmniej na dwa sposoby:

- **Podejście generujące**: zauważmy, że rozkład warunkowy $p(y|\mathbf{x})$ można wyznaczyć korzystając ze wzoru Bayesa:

$$\begin{aligned} p(y|\mathbf{x}) &= \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \\ &= \frac{p(\mathbf{x}|y)p(y)}{\sum_{y'} p(\mathbf{x}|y')p(y')} \end{aligned}$$

  W celu poznania rozkładu warunkowego $p(y|\mathbf{x})$ będziemy modelować wielkości $p(\mathbf{x}|y, \theta)$ i $p(y|\pi)$, gdzie θ i π oznaczają parametry modelu.

- **Podejście dyskryminujące**: rozkład warunkowy $p(y|\mathbf{x})$ modelujemy wprost za pomocą modelu $p(y|\mathbf{x}, \theta)$, gdzie θ oznacza parametry modelu.

Podejście generujące: Naive Bayes

Model W podejściu generującym naszym celem jest modelowanie rozkładów $p(\mathbf{x}|y,\theta)$ i $p(y|\theta)$. Rozkład na grupę tematyczną wyrażać będziemy za pomocą rozkładu wielopunktowego:

$$p(y|\pi) = \mathrm{M}(y|\pi), \tag{2}$$

gdzie $\pi = (\pi_0, \ldots, \pi_{K-1})$, a π_k oznacza prawdopodobieństwo *a priori* k-tej grupy tematycznej.

W rozważanym przypadku cechy opisujące dokument są binarne, dlatego odpowiednim rozkładem byłby taki rozkład, który każdej możliwej konfiguracji słów przyporządkowuje wartość prawdopodobieństwa. Zwróćmy jednak uwagę, że takich konfiguracji jest 2^D, a zatem model musiałby posiadać $2^D - 1$ parametrów. Przykładowo, dla $D = 100$ wyuczenie takiego modelu jest w praktyce niemożliwe. Dlatego dalej przyjmować będziemy, że występowanie słów jest niezależne od siebie, wówczas rozważany model będzie posiadał jedynie D parametrów. Naturalnie w ten sposób tracimy możliwość modelowania współzależności między występowaniem słów, ale zyskujemy możliwość wyuczenia takiego modelu. Model, który zakłada niezależność cech, nazywa się **Naive Bayes** i wyraża się następująco:

$$p(\mathbf{x}|y,\theta) = \prod_{d=1}^{D} p(x_d|y,\theta) \tag{3}$$

iloczyn prawd dla każdego słowa

gdzie dla rozpatrywanego zadania rozkład warunkowy na cechy modelujemy za pomocą rozkładu dwupunktowego:

$$p(x_d|y=k,\theta) = \mathrm{B}(x_d|\theta_{d,k}) \tag{4}$$

$$= \theta_{d,k}^{x_d}(1-\theta_{d,k})^{1-x_d}. \tag{5}$$

Uczenie Celem uczenia w przypadku modelu Naive Bayes jest oszacowanie prawdopodobieństw $\{\pi_k\}_{k=0\ldots,3}$ oraz prawdopodobieństw $\{\theta_{d,k}\}_{\substack{d=1,\ldots,D\\k=0,\ldots,3}}$ w oparciu o dane uczące $\mathcal{D}$.

Korzystając z metody największej wiarygodności (estymator ML) wielkości te możemy wyznaczyć w następujący sposób:

$$\boxed{\pi_k = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}(y_n = k),} \tag{6}$$

wektor prawdopod a priori
prawd. wystąp kategori sum = 1

$$\theta_{d,k} = \frac{\sum_{n=1}^{N} \mathbb{I}(y_n = k, x_{n,d} = 1)}{\sum_{n=1}^{N} \mathbb{I}(y_n = k)}, \tag{7}$$

gdzie $\mathbb{I}(\cdot)$ oznacza indykator, który zwraca wartość 1, gdy wszystkie warunki logiczne, które są jego argumentami, są prawdziwe i wartość 0 – w przeciwnym przypadku.

Często w praktyce może wystąpić problem, że pewne słowo może nie pojawić się w danych uczących lub posiadamy zbyt mało danych, aby dostatecznie dobrze oszacować interesujące nas prawdopodobieństwo. Wówczas stosuje się dodatkowy rozkład *a priori* na słowa, dla których określamy *założoną* wartość występowania słowa a oraz jego niewystępowania b. W rozważanym przypadku, dla cech binarnych, wygodnym rozkładem *a priori* jest rozkład beta:

$$p(\theta_{d,k}) = \text{Beta}(\theta_{d,k}|a, b), \tag{8}$$

gdzie $a, b > 0$ są tzw. *hiperparametrami*. Wówczas można wyznaczyć estymator maksymalnej a posteriori (MAP) dla $\theta_{d,k}$:

$$\boxed{\theta_{d,k} = \frac{\sum_{n=1}^{N} \mathbb{I}(y_n = k, x_{n,d} = 1) + a - 1}{\sum_{n=1}^{N} \mathbb{I}(y_n = k) + a + b - 2}.} \tag{9}$$

Podejście dyskryminujące: κ-NN

Model κ-Nearest Neighbors (κ-NN) jest przykładem modelu dyskryminującego oraz modelu nieparametrycznego, tzn. takiego, dla którego parametrami modelu są dane uczące. Rozkład warunkowy dla grupy tematycznej pod warunkiem dokumentu tekstowego określa się w następujący sposób:

$$\boxed{p(y|\mathbf{x}, \kappa) = \frac{1}{\kappa} \sum_{i \in N_\kappa(\mathbf{x}, \mathcal{D})} \mathbb{I}(y_i = y)} \tag{10}$$

gdzie κ oznacza liczbę najbliższych sąsiadów, $N_\kappa(\mathbf{x}, \mathcal{D})$ oznacza zbiór indeksów κ najbliższych sąsiadów dla dokumentu $\mathbf{x}$ w zbiorze treningowym $\mathcal{D}$.

Zauważmy, że model κ-NN zależy od zbioru treningowego oraz wartości parametru κ, czyli liczby sąsiadów. Wartość κ musi być zadana przed dokonaniem predykcji.

Kluczowym pojęciem dla κ-NN jest **odległość** za pomocą której wyznacza się najbliższych sąsiadów. W rozważanym przypadku do czynienia mamy z dokumentemi tekstowymi opisanymi za pomocą D cech binarnych określających występowanie słów w dokumencie. W celu wyznaczenia odległości między dwoma dokumentami posłużymy się **metryką Hamminga**, która określa liczbę miejsc, na których dwa ciągi różnią się od siebie. Na przykład, dla $\mathbf{x}_1 = (1, 0, 0, 1)$ i $\mathbf{x}_2 = (1, 1, 0, 0)$ odległość Hamminga między $\mathbf{x}_1$ i $\mathbf{x}_2$ wynosi 2:

$$\begin{array}{ccccc} 1 & 0 & 0 & 1 & \\ 1 & 1 & 0 & 0 & \\ \hline 0 & 1 & 0 & 1 & = 2 \end{array}$$

Selekcja modelu

W rozważanym problemie mamy do czynienia z trzema wielkościami, których nie wyuczamy w oparciu o dane, tj. liczbę sąsiadów κ dla κ-NN oraz wartości rozkładu *a priori* dla Naive Bayes. W przypadku, gdy dysponujemy zbiorem walidacyjnym $\mathcal{D}_{val}$ o długości N_{val}, możemy przeprowadzić selekcję tych wartości. W celu oceny modelu w oparciu o wspomniane wielkości, stosować będziemy miarę **błąd klasyfikacji**:

$$\boxed{E(\mathcal{D}_{val};\alpha) = \frac{1}{N_{val}}\sum_{n=1}^{N_{val}} \mathbb{I}(y_n \neq \hat{y}_n),} \tag{11}$$

gdzie α jest hiperparametrem κ w przypadku κ-NN lub (a, b) dla Naive Bayes, oraz $\hat{y}_n$ jest predykowaną przez model wartością klasy dla n-tego przykładu ze zbioru walidacyjnego.

Algorithm 1: Procedura selekcji modelu dla modelu κ-NN lub Naive Bayes.

Wejście : Zbiór walidacyjny $\mathcal{D}_{val}$, zbiór wartości hiperparametru(-ów) Λ

Wyjście: Wartość α

```
for α ∈ Λ do
    if Naive Bayes then
        Znajdź estymatory dla π i θ z użyciem a i b ;
        Policz wartość E(D_val; (a, b)) ;
    else if κ-NN then
        Policz wartość E(D_val; κ) ;
end
Zwróć wartość α, dla której E(D_val; α) jest najniższa.
```

~~Testowanie poprawności działania~~

~~Do sprawdzania poprawności działania zaproponowanych rozwiązań służy funkcja `main` w pliku `main.py`.~~

~~**W pliku `main.py` nie wolno czegokolwiek zmieniać ani dopisywać.**~~

~~Dodatkowo, aby program zadziałał, należy zainstalować pakiet `wordcloud`. W Windowsie można zrobić to w następujący sposób:~~

1. ~~Zainstalować Visual C++ 2015 Build Tools ze strony:~~
   ~~http://landinghub.visualstudio.com/visual-cpp-build-tools~~
2. ~~Uruchomić linię poleceń Start -> cmd i wpisać:~~
   ~~`pip install wordcloud`~~

Instrukcja wykonania zadania

Instrukcja: Należy zaimplementować wszystkie funkcje w pliku `content.py`

1. Zaimplementować funkcję `hamming_distance` liczącą odległości Hamminga. Funkcja przyjmuje dwie macierze rzadkie reprezentujące dwa zbiory obiektów i wyznacza macierz zawierającą odległości Hamminga pomiędzy obiektami z jednego i drugiego zbioru.

2. Zaimplementować funkcję `sort_train_labels_knn` liczącą macierz posortowanych etykiet klas względem macierzy odległości. Dla danej macierzy odległości i zadanych etykiet klas należy zbudować macierz, która w każdym wierszu zawiera etykiety klas posortowane zgodnie z odległościami z tego samego wiersza w macierzy odległości.[1]

3. Zaimplementować funkcję `p_y_x_knn` wyznaczającą macierz prawdopodobieństw przynależności do każdej z klas dla modelu KNN (10).

4. Zaimplementować funkcję `classification_error` liczącą błąd klasyfikacji (11). Jeżeli dla danego przykładu $\mathbf{x}$ prawdopodobieństwo $p(y = k|\mathbf{x})$ dla kilku klas k jest maksymalne, to jako predykcję modelu wybieramy klasę o najwyższym numerze k.

5. Zaimplementować funkcję `model_selection_knn` dokonującą selekcji modelu KNN dla zadanych wartości κ.

6. Zaimplementować funkcję `estimate_a_priori_nb` liczącą estymator ML dla klas, π_k (6), dla modelu NB.

7. Zaimplementować funkcję `estimate_p_x_y_nb` liczącą estymator MAP dla cech, $\theta_{d,k}$ (9), dla modelu NB.

8. Zaimplementować funkcję `estimate_p_y_x_nb` wyznaczającą macierz prawdopodobieństw przynależności do każdej z klas dla modelu NB.

9. Zaimplementować funkcję `model_selection_nb` dokonującą selekcji modelu NB dla zadanych wartości parametrów a i b.

UWAGA! Wszelkie nazwy funkcji i zmiennych w pliku `content.py` muszą pozostać zachowane.

Pytania kontrolne

1. Proszę wyznaczyć estymator największej wiarygodności dla rozkładu wielopunktowego.

2. Proszę wyznaczyć estymator największej wiarygodności dla rozkładu dwupunktowego.

3. Proszę wyznaczyć estymator maksymalnego *a posteriori* dla rozkładu dwupunktowego.

4. Dlaczego stosujemy założenie o niezależności cech określających wystąpienie słowa w dokumencie? Jaka jest korzyść z takiego podejścia, a jaka jest strata?

[1]PRZYKŁAD: macierz odległości: `[2 5 3; 6 7 1]`, zadane etykiety klas: `[0 3 2]`, macierz posortowanych etykiet: `[0 2 3; 2 0 3]`.

5. Jaka jest interpretacja parametrów θ? Ile jest takich parametrów dla D cech i K klas?
6. Jaka jest interpretacja parametrów π? Ile jest takich parametrów dla D cech i K klas?
7. Jaka jest interpretacja hiperparametru κ? Za co odpowiada? Jaka jest jego interpretacja geometryczna? Jak jego wartość wpływa na rozwiązanie?
8. W jaki sposób wyznaczamy sąsiedztwo w modelu κ-NN?
9. Czy model κ-NN jest modelem generującym, czy dyskryminującym? Czy jest to model parametryczny, czy nieparametryczny?
10. Czy model Naive Bayes jest modelem generującym, czy dyskryminującym? Czy jest to model parametryczny, czy nieparametryczny?