

Dokumentacja wstępna ZUM

Temat:

Tworzenie modeli klasyfikacji wieloetykietowej przez zastosowanie dekompozycji na wiele powiązanych zadań klasyfikacji jednoetykietowych zgodnie z metodą bayesowskiego łańcucha klasyfikatorów. Porównanie z algorytmami klasyfikacji wieloetykietowej dostępnymi w środowisku R lub Python.

(Bayes chain classifier for multi-label classification)

Spis treści

1.	Interpretacja tematu projektu	1
2.	Opis części implementacyjnej	2
2.1	Przygotowanie danych	2
2.2	Preprocessing danych	2
2.3	Budowa klasyfikatorów jednoetykietowych	2
2.4	Budowa sieci bayesowskiej	3
2.5	Implementacja łańcucha klasyfikatorów	3
2.6	Połączenie metod	4
2.7	Analiza wyników	4
3.	Plan badań - testowanie modelu:	4
3.1	Cel badań,	4
3.2	Charakterystyka zbioru danych:	4
3.3	Procedura oceny modeli	5
3.4	Plan eksperymentów	6

1. Interpretacja tematu projektu

Głównym celem projektu jest implementacja algorytmu służącego do klasyfikacji przykładów posiadających więcej niż jedną etykietę. Każdy przykład ma przypisany zestaw binarnych (lub wieloklasowych) znaczników. Jednym ze sposobów stworzenia

modelu umożliwiającego predykcje na bazie przykładów wieloetykietowych jest dokonanie transformacji tej bazy zgodnie z metodą łańcucha klasyfikatorów. Metoda ta polega na wykorzystaniu koncepcji związanej z sekwencyjnym tworzeniem modeli klasyfikacji binarnej, w której to dla każdego kolejnego modelu klasyfikacji dodaje się do zbioru atrybutów predykcje etykiet poprzednich modeli wykorzystanych przy wcześniej rozważanych etykietach, w poprzednich ogniwach łańcucha. Dzięki temu każdy kolejny model estymuje tylko jedną etykietę dla zbioru przykładów. Tym samym wykorzystana jest informacja o zależności między poszczególnymi etykietami. Kolejność predykcji dla następujących po sobie etykiet jest istotna i zwykle ma wpływ na końcowy wynik. Dobór kolejności ogniw łańcucha dla predykcji kolejnych etykiet będzie ustalana na podstawie sieci bayesowskiej utworzonej na podstawie zależności między poszczególnymi etykietami.

Aby dopełnić projekt, wstępnie wybranym klasyfikatorem dla każdego z modeli będzie gaussowski klasyfikator bayesa.

2. Opis części implementacyjnej

Projekt będzie realizowany w języku Python. Poniżej przedstawiono kolejne etapy implementacji:

2.1 Przygotowanie danych

W tym etapie wczytane zostaną przykłady z plików o rozszerzeniu .csv. Następnie dane te zostaną przejrane oraz dokonana zostanie ich wstępna analiza w celu np. znalezienia i wyeliminowania przykładów z brakującymi wartościami. Po wstępnej analizie dane zostaną podzielone na 2 części - zestaw atrybutów oraz zestaw etykiet.

2.2 Preprocessing danych

Kolejnym etapem projektu będzie przekształcenie danych w taki sposób, aby miały one odpowiedni format i były gotowe do bezpośredniego przekazania modelu klasyfikacji. Zostanie zaimplementowana funkcja `split_data()`, która podzieli dane na zbiór treningowy i testowy w proporcji 4:1. Funkcja ta będzie zwracać 2 zestawy danych. Następnie przewiduje się kodowanie etykiet za pomocą enkodera `LabelEncoder()` z biblioteki `sklearn.preprocessing`. Operacja ta jest niezbędna do poprawnego działania algorytmu klasyfikacji danych.

2.3 Budowa klasyfikatorów jednoetykietowych

Klasyfikatorami jednoetykietowymi będą głównie gaussowskie naiwne klasyfikatory bayesa, ponieważ w wybranych zbiorach danych atrybuty mają charakter ciągły. Planuje się wykorzystanie zaimplementowanych już klasyfikatorów dostępnych w bibliotece `sklearn.naive_bayes`. Zostanie wykorzystana metoda `fit()` – uczenie modelu oraz `predict()` - predykcja modelu. Naiwny klasyfikator bayesowski bazuje na wyznaczaniu

prawdopodobieństwa przynależności do danej klasy na podstawie prawdopodobieństwa przynależności do danej klasy poszczególnych atrybutów. Klasyfikator Bayesa korzysta bezpośrednio z twierdzenia Bayesa, z którego można wyliczyć prawdopodobieństwo warunkowe zaistnienia pewnego zdarzenia, pod warunkiem zajścia innego zdarzenia:

$$P(c = d | a_1 = v_1, \dots, a_n = v_n) = \frac{P(c = d) \cdot P(a_1 = v_1, \dots, a_n = v_n | c = d)}{P(a_1 = v_1, \dots, a_n = v_n)}$$

gdzie: a to zestaw cech, c to badana hipoteza, czyli etykieta. $P(c=d)$ to prawdopodobieństwo a'priori, $P(c=d | a_1=v_1, \dots, a_n=v_n)$ to prawdopodobieństwo a'posteriori.

2.4 Budowa sieci bayesowskiej

Sieć bayesowska będzie tym składnikiem całego programu, który reprezentuje relacje zależności pomiędzy etykietami. Sieć bayesowska to acykliczny graf skierowany, którego węzłami będą numery etykiet, a skierowane krawędzie będą odzwierciedlać zależność przyczynową.

Procedura implementacyjna sieci Bayesowskiej będzie korzystać z algorytmu znanego w literaturze pod nazwą Chow & Liu. Składa się on z następujących kroków:

- 1) Wyznaczenie informacji wzajemnej między wszystkimi etykietami.
- 2) Wyznaczenie drzewa rozpinającego o maksymalnej sumie wag – wykorzystanie algorytmu Kruskala
- 3) Ustalenie zwrotów krawędzi od wybranego węzła początkowego

Odczytując węzły z grafu w kolejności topologicznej otrzymamy wejściowe parametry dla łańcucha klasyfikatorów, mianowicie kolejność etykiet do predykcji.

2.5 Implementacja łańcucha klasyfikatorów

W tym etapie zaimplementowany zostanie łańcuch klasyfikatorów, który będzie składał się z klasy `ClassifiersChain`. Klasa ta będzie składała się z metod `fit()` służącej do nauki całych łańcuchów oraz metody `predict()` służącej do dawania predykcji całego modelu. Aby utworzyć łańcuch będzie trzeba podać nazwę klasyfikatora służącego do predykcji kolejnych etykiet, a także listę, w której to zostanie wskazana kolejność etykiet będącymi kolejnymi ogniwami łańcucha.

Łańcuch będzie działał tak, że na wejściu dostanie zestaw przykładów oraz kolejne numery etykiet przekazane w liście w odpowiedniej kolejności, podyktowanej przez sieć bayesowską. Następnie w tejże kolejności łańcuch będzie estymował klasę każdej etykiety, przy czym dla każdego kolejnego ogniwa danego łańcucha wynik poprzedniej predykcji będzie traktowany jako dodatkowy atrybut danego przykładu i wiedza ta zostanie wykorzystana do wyliczenia wartości kolejnej etykiety danego łańcucha.

Innymi słowy dla przykładu składającego się z wektora atrybutów $X = (x_1, x_2, x_3)$ i etykiet $Y = (y_1, y_2)$, pierwszy model stworzony zostanie dla przykładu składającego się jedynie z wektora atrybutów $X = (x_1, x_2, x_3)$ i etykiety $Y = (y_1)$. Model za sprawą

klasyfikatora Bayesa dokona predykcji wartość y_1 . W kolejnym podejściu stworzony zostanie drugi model, tym razem już dla przykładu składającego się z wektora atrybutów $X = (x_1, x_2, x_3, y_1)$ i etykiety $Y = (y_2)$. Analityczny opis tej metody wygląda następująco:

$$p(\mathbf{y}|\mathbf{x}) = p(y_1|\mathbf{x}) \prod_{j=2}^L p(y_j|\mathbf{x}, y_1, \dots, y_{j-1})$$

2.6 Połączenie metod

Kod wykonawczy będzie pobierał dane, przygotowywał je i wykonywał na nich preprocessing. Następnie na wydzielonych danych treningowych będzie tworzona sieć bayesowska. Korzystając z niej zbudowana zostanie odpowiednią liczbą łańcuchów klasyfikatorów, które wyestymują poszukiwane klasy. Następnie wyniki predykcji zostaną połączone.

2.7 Analiza wyników

Zostanie dostarczona w dokumentacji końcowej

3. Plan badań - testowanie modelu:

3.1 Cel badań,

Z racji na implementacyjny charakter projektu postanowiono zbadać i zweryfikować jedynie podstawowe czynniki i parametry modelu mogące mieć wpływ na ostateczne wyniki klasyfikacji. Planuje się:

- Zbadać wyniki klasyfikacji dla przynajmniej 2 zestawów danych testowych, podając do łańcucha klasyfikatorów jedynie jedną etykietę (zbadanie jakości zwykłych predykcji)
- wyznaczyć czas tworzenia sieci i uczenia modelu łańcucha,
- porównać uzyskany model z modelem łańcucha klasyfikatorów, dla którego etykiety dobierane są w sposób losowy (bez sieci bayesowskiej),
- sprawdzić model dla wybranego innego klasyfikatora jednoetykietowego,
- sprawdzić wpływ zmiany korzenia w sieci bayesowskiej

3.2 Charakterystyka zbioru danych:

Zdecydowano się na wybór danych do testowania modeli z portalu UCI Machine Learning Repository: <https://archive.ics.uci.edu/dataset/528/amphibians>. Zbiór danych zawiera 189 przykładów, 22 atrybuty i 4 etykiety oraz reprezentuje on informacje o płazach, pochodzące z portalu GIS oraz informacji satelitarnych, a także z informacji zebranych z inwentaryzacji przyrodniczych w Polsce. Atrybuty każdego przykładu opisują środowisko przyrodnicze danej okolicy, np. liczba zbiorników wodnych w okolicy, typ zbiorników, obecność podmokłych łąk, stawów, itp. Etykiety opisują z kolei gatunki

płazów, które występują w danej okolicy; każda etykieta odpowiada osobnemu gatunkowi płaza. Zbiór danych nie zawiera brakujących wartości. Składa się z atrybutów numerycznych, porządkowych, a także kategoriowych.

Jako drugi zestaw danych wybrano również dane biologiczno-przyrodnicze: <https://archive.ics.uci.edu/dataset/406/anuran+calls+mfccs>. Dane umożliwiają rozpoznawanie gatunków anuranów (żab) na podstawie dźwięków, które wydają. Ten zestaw danych został utworzony na podstawie segmentacji 60 nagrań audio należących do 4 różnych rodzin, 8 rodzajów i 10 gatunków. Każdy dźwięk odpowiada jednemu okazowi (przykładowi). Dane składają się z ponad 7000 przykładów. Każdy przykład posiada 3 etykiety określające gatunek, rodzaj i rodzinę płazów. Przykłady składają się z 22 atrybutów - współczynników MFCC, które to są wynikami analizy sygnału dźwiękowego. Podane dane mają charakter numeryczny. Spodziewane jest istnienie silnej zależności pomiędzy poszczególnymi etykietami klas.

Families:		
	Bufonidae	68
	Dendrobatidae	542
	Hylidae	2165
	Leptodactylidae	4420
Genus:		
	Adenomera	4150
	Ameerega	542
	Dendropsophus	310
	Hypsiboas	1593
	Leptodactylus	270
	Osteocephalus	114
	Rhinella	68
	Scinax	148
Species:		
	AdenomeraAndre	672
	AdenomeraHylaedact...	3478
	Ameeregatrivittata	542
	HylaMinuta	310
	HypsiboasCinerascens	472
	HypsiboasCordobae	1121
	LeptodactylusFuscus	270
	OsteocephalusOopha...	114
	Rhinellagranulosa	68
	ScinaxRuber	148

Rysunek 1 Etykiety drugiego zestawu danych

3.3 Procedura oceny modeli

Przede wszystkim planuje się wyznaczyć dokładność predykcji modelu. W tym celu wybrany zbiór danych zostanie podzielony na zbiór treningowy i testowy, następnie wytrenowany na zbiorze treningowym, a poszczególne wyniki klasyfikacji zostaną porównane z etykietami zbioru testowego. Dokładność wyników dla każdej z etykiet (np. korzystając z miary F) zestawie się w tabeli zbiorczej. Planuje się zagregować wyliczenia jakości za pomocą makro-uśredniania.

Dodatkowo przewiduje się wyznaczenie straty Hamminga dla klasyfikacji wieloetykietowej.

3.4 Plan eksperymentów

W ramach eksperymentów planuje się przeprowadzenie testów dla różnych zbiorów danych, różnych proporcji podziału na zbiór treningowy i testowy, a także porównanie algorytmu a z algorytmem łańcucha klasyfikatorów, w którym kolejne etykiety do predykcji wybierane są w kolejności losowej.