Wojciech Makos
Jakub Kliszko

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

Głównym celem projektu jest implementacja algorytmu służącego do klasyfikacji przykładów posiadających więcej niż jedną etykietę. Każdy przykład ma przypisany zestaw binarnych 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 zachowana jest relacja zależności między etykietami. Kolejność predykcji 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 naiwny klasyfikator bayesowski.

2. Opis części implementacyjnej oraz lista algorytmów, bibliotek, klas, funkcji

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

2.1 Przygotowanie danych

W tym etapie wczytane zostaną przykłady z plików o rozszerzeniu .arff. Następnie dane te zostaną przejrzane 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. Operacja ta będzie możliwa dzięki zaimplementowaniu klasy Data dziedziczącej z klasy NamedTuple. Cała realizacja odbędzie się w ramach metody read_data().

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 konwersję danych kategorycznych na wersję numeryczną za pomocą enkodera (prawdopodobnie wybranym enkoderem bedzie OneHotEncoder(), którego użyjemy korzystając z sklearn).

2.3 Budowa klasyfikatorów jednoetykietowych

Klasyfikatorami jednoetykietowymi będą naiwne klasyfikatory bayesowskie, które będą zaimplementowane jako oddzielna klasa o nazwie `NaiveBayes`, w której to pojawią się takie metody jak `fit()` - uczenie modelu, `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 bazuje bezpośrednio na twierdzeniu Bayesa, z którego można wyliczyć prawdopodobieństwo warunkowe zaistnienia pewnego zdarzenia, pod warunkiem zajścia innego zdarzenia:

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

gdzie: `a` to zestaw cech, `c` to badana hipoteza, czyli etykieta. `P(c=d)` to prawdopodobieństwo a'priori, `P(c=d|a1=v1,...,an=vn)` to prawdopodobieństwo a'posteriori. Warto zauważyć, że klasyfikator bayesowski zakłada niezależność cech, która to obrazuje się następującym wzorem:

$$P(a_1 = v_1 \ldots, a_n = v_n|c = d) = \prod P(a_i = v_i|c = d)$$

Powyższy zabieg jest oczywiście nieprawdziwym uproszczeniem, ale w praktyce jest w stanie dawać wyniki z dość wysoką dokładnością. Przy implementacji klasyfikatora Bayesa zastosujemy również wygładzanie laplace'a z możliwością zmiany parametru alpha.

2.4 Budowa sieci bayesowskiej

Sieć bayesowska będzie tym składnikiem, który reprezentuje relacje zależności pomiędzy etykietami. Wybrana sieć bayesowska będzie jedną z prostszych wersji sieci o reprezentacji drzewa nieskierowanego MWST. Implementacja wygląda tak, że:

2.5 Implementacja łańcucha klasyfikatorów

W tym etapie zaimplementowany zostanie łańcuch klasyfikatorów, który będzie składał się z klasy `ClassifierChain`. Klasa ta będzie się składała z metod: .....

Łańcuch będzie działał tak, że na wejściu dostanie zestaw przykładów składających się z atrybutów oraz numery etykiet 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 = (x1, x2, x3) i etykiet Y = (y1, y2), pierwszy model stworzony zostanie dla przykładu składającego się jedynie z wektora atrybutów X = (x1, x2, x3) i etykiety Y = (y1). Model za sprawą klasyfikatora Bayesa dokona predykcji wartość y1. W kolejnym podejściu stworzony zostanie drugi

model, tym razem już dla przykładu składającego się z wektora atrybutów X = (x1, x2, x3, y1) i etykiety Y = (y2). Analityczny opis tej metody wygląda następująco:

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

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 liczba łańcuchów klasyfikatorów, które wyestymują poszukiwane klasy. Następnie wyniki predykcji zostaną połączone. JAK?

2.7 Testowanie modelu

2.8 Analiza wyników

3. Plan badań - testowanie modelu:

3.1 Cel badań,

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

- wyznaczyć dokładność wyników klasyfikacji dla przynajmniej 2 zestawów danych testowych,
- wyznaczyć czas uczenia modelu,
- 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 kategorycznych.

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. 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

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 wyniki klasyfikacji zostaną porównane z etykietami zbioru testowego. Ta sama operacja przeprowadzona zostanie dla obu zbiorów danych, a dokładność wyników porówna się ze sobą i zestawi w tabeli zbiorczej. Wyniki predykcji porównane zostaną również z gotowymi implementacjami algorytmów, zaimplementowanych w bibliotece scikit-learn. Dodatkowo do tworzenia klasyfikatora wieloetykietowego przyda się biblioteka scikit-multilearn. Ponadto w projekcie na potrzeby przetwarzania danych planuje się wykorzystanie biblioteki pandas oraz numpy.

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 modelu z modelem, gdzie kolejne etykiety do predykcji wybierane są w kolejności losowej.