Multi-label classification

Wojciech Bełka
Patryk Smól

Czym jest multi-label classification?

Klasyfikacja wielo-etykietowa to problem, w którym dla danych wejściowych rozważamy więcej niż jedno prawidłowe przyporządkowanie klasy.

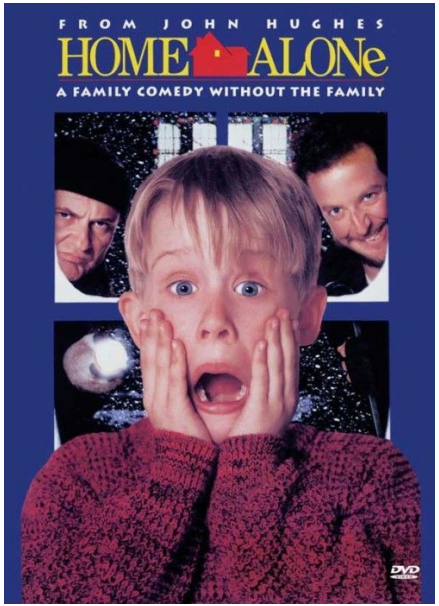
$$\boldsymbol{x} = (x_1, x_2, \ldots, x_p) \xrightarrow{\boldsymbol{h}(\boldsymbol{x})} \boldsymbol{y} = (y_1, y_2, \ldots, y_m) \in \mathcal{Y} = \{0, 1\}^m$$

Przykład

- drzewo
- budynek

Przykład

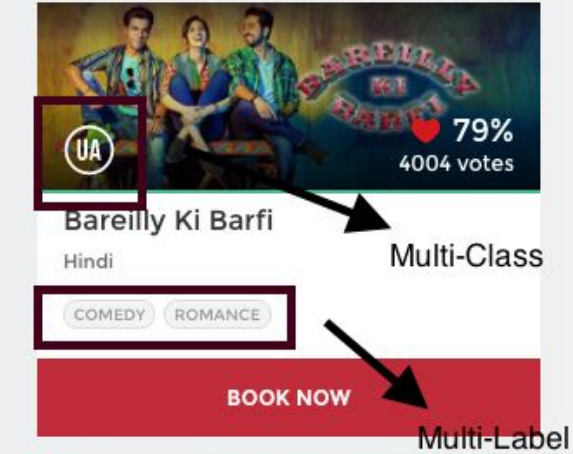
- komedia
- rodzinny



Zastosowanie

- kategoryzacja zdjęć
- bioinformatyka (klasyfikacja genów)
- kategoryzacja tekstów

Multi-Label a Multi-Class



Metody transformacji problemu wieloetykietowego

- The binary relevance (BR)
  - One versus all (one against all)
  - One versus one
- Classifier Chains
- The label powerset (LP)

The binary relevance (BR)

Przekształcenie problemu wielu etykietowego do wielu problemów binarnych (jeden problem-jeden klasyfikator). Każdy klasyfikator przewiduje istotność jednej z etykiet.

Sposób ten nie modeluje zależności pomiędzy poszczególnymi cechami w wektorze.

One versus all - jeden przeciw wszystkim

Klasyfikator zostaje przeszkolony na podstawie próbek należących do klasy opisującej problem oznaczonych jako pozytywne i próbek należących do innych klasy oznaczonych jako negatywne

One versus one - jeden przeciw jednemu

Każdy klasyfikator szkolony jest na podstawie par próbek należących do dwóch klas. W czasie prognozy stosowany jest schemat głosowania: wszystkie klasyfikatory są stosowane do próbki, a klasa, która uzyskała największą liczbę prognoz, zostanie przewidziana przez połączony klasyfikator.

Classifier Chains

Klasyfikatory budują łańcuch, w którym każdy z nich uczy się binarnej klasyfikacji pojedynczej etykiety wykorzystując metodę BR. Cechy przekazane dla każdego klasyfikatora są rozszerzone o wartości binarne wskazujące, które z poprzednich etykiet zostały przypisane do predykcji poprzedzającego klasyfikatora.

f(x1,x2,x3,x4)=**y1**

f(x1,x2,x3,x4,**y1**)=**y2**

f(x1,x2,x3,x4,**y1**,**y2**)=**y3**

f(x1,x2,x3,x4,**y1**,**y2**,**y3**)=**y4**

The label powerset (LP)

Tworzy jest jeden binarny klasyfikator dla każdej kombinacji etykiet obecnej w zbiorze treningowym.

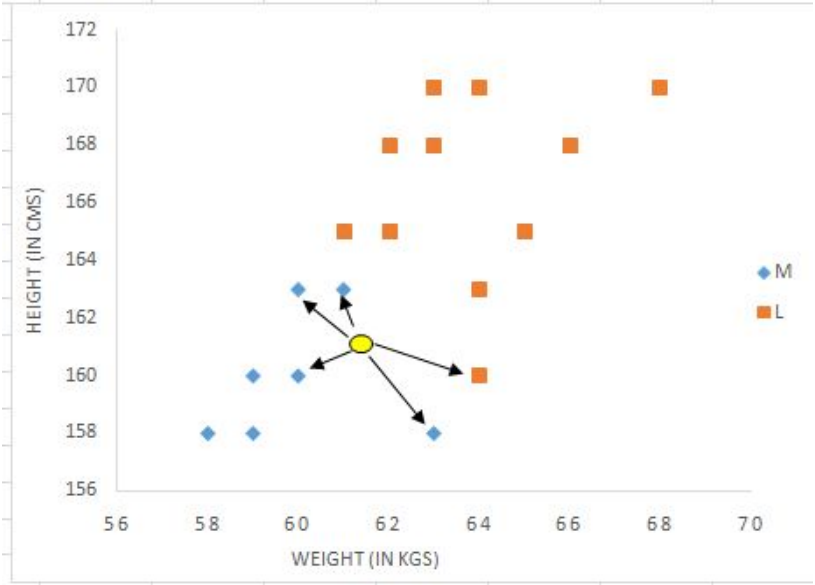
Klasy: A, B, C

[0 0 0],
[1 0 0],
[0 1 0],
[0 0 1],
[1 1 0], → przykład dla etykiety A i C
[1 0 1],
[0 1 1],
[1 1 1]

Algorytmy wspierające klasyfikacje wieloetykietową

- K najbliższych sąsiadów
- Drzewa decyzyjne
- Sieci neuronowe
- Boosting
- Kernel methods for vector output (SVM)

K najbliższych sąsiadów (ML-KNN)

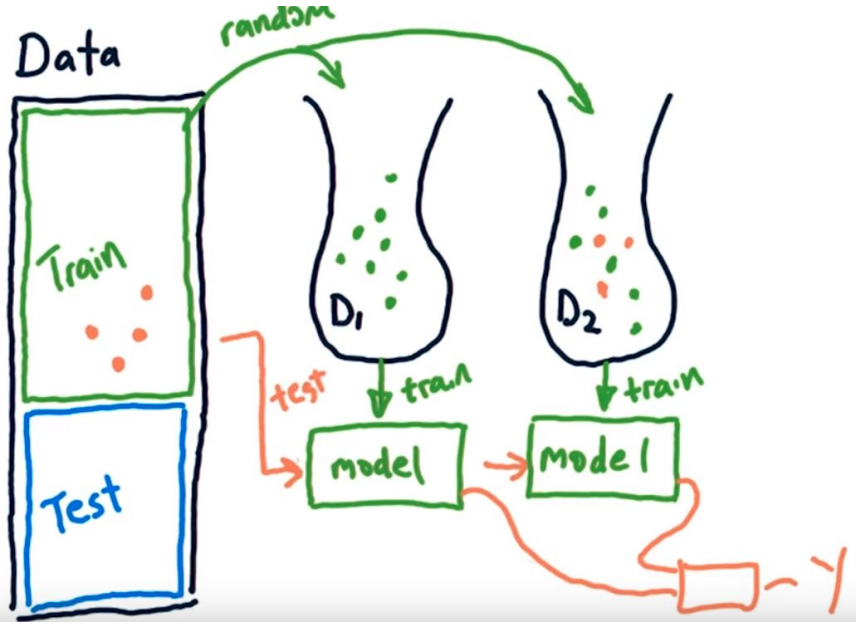


- zasada maximum a posteriori (MAP)

```python
from sklearn.datasets import fetch_mldata
import numpy as np
from sklearn.neighbors import KNeighborsClassifier


mnist = fetch_mldata('MNIST original')
X, y = mnist["data"], mnist["target"]
X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]
shuffle_index = np.random.permutation(60000)
X_train, y_train = X_train[shuffle_index], y_train[shuffle_index]
y_train_large = (y_train >= 7)
y_train_odd = (y_train % 2 == 1)
y_multilabel = np.c_[y_train_large, y_train_odd]
knn_clf = KNeighborsClassifier()
knn_clf.fit(X_train, y_multilabel)
some_digit = X[36000]
print(knn_clf.predict([some_digit]))
```

Boosting



- uczenie “słabych” klasyfikatorów
- uczenie na części zbioru
- lepsze skorelowanie etykiety z danymi
- AdaBoost

Neural network

$$X = \{x_1, \ldots, x_n\}$$

$$y = \{y_1, \ldots, y_n\}$$

$$\hat{y}_i = \operatorname{argmax}_{j \in \{1,2,3,4,5\}} P(c_j | x_i).$$

$$P(c_j | x_i) = \frac{1}{1 + \exp(-z_j)}.$$

Narzędzia

- Mulan i Meka bazujące na Weka - dla języka Java
- scikit-learn - dla Python i Julia
- mlr - dla języka R
- popularne datasety: http://mulan.sourceforge.net/datasets.html

DZIĘKUJEMY