Metody Odkrywania Wiedzy

Nie-całkiem-naiwny klasyfikator Bayesa

Dokumentacja końcowa

Mateusz Jamiołkowski, Michał Uziak

28 maja 2013

Spis treści

1 Wstęp

Celem projektu jest implementacja algorytmu budowy sieci bayesowskiej, która będzie uwzględniać zależności między atrybutami klasyfikowanych obiektów. Na podstawie tak zbudowanego modelu zostanie zaimplementowany alogrytm, którego działanie zostanie porównane z dostępnymi modułami klasyfikacji języka R.

2 Algorytm

Na podstawie analizy dostępnych materiałów zdecydowano, że budowa sieci bayesowskiej zostanie zaimplementowana z wykorzystaniem drzew TAN (tree-augmented naive Bayesian Network).

2.1 Opis działania algorytmu tworzącego sieć bayesowską

Opis algorytmu tworzenia drzew TAN:

1. Obliczenie informacji wzajemnej $I_p(A_i, A_j|C)$ dla każdej pary atrybutów A_i, A_j takich, że $i \neq j$, gdzie $I_p(A_i, A_j|C) = \sum_{c \in C} P(A_i, A_j, c) \log \frac{P(A_i, A_j|c)}{P(A_i|c)P(A_j|c)}$,C - klasa obiektu.

2. Budowa grafu zupełnego nieskierowanego, którego wierzchołkami będą atrybuty $A_1, ..., A_n$ natomiast waga krawędzi pomiędzy atrybutami A_i i A_j będzie równa $I_p(A_i, A_j)$

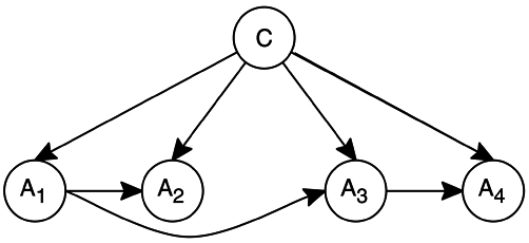
3. Budowa minimalnego drzewa rozpinającego z wykorzystaniem algorytmu Kruskala.

4. Stworzenie drzewa skierowanego, poprzez wybór jednego wierzchołka A_p. Kierunek każdej krawędzi w drzewie zostanie tak dobrany, aby, przechodząc zgodnie z kierunkiem krawędzi, odległość od wierzchołka A_p rosła.

5. Dodanie wierzchołka C(reprezentującego klasę obiektu) do drzewa oraz utworzenie krawędzi skierowanych "od" wierzchołka C pomiędzy tym wierzchołkiem a już istniejącymi wierzchołkami $A_1, A_2...A_n$.

2.2 Opis konstrukcji klasyfikatora

Działanie klasyfikatora będzie polegało na wyborze klasy, która jest najbardziej prawdopodobna dla danego zestawu wartości atrybutów. W odróżnieniu od naiwnego klasyfikatora Bayesa, budowany klasyfikator będzie uwzględniał wzajemne zależności atrybutów. W zbudowanym drzewie krawędź wchodząca do węzła(atrybutu) oznacza zależność od innego węzła(atrybutu). Na rysunku 5.2 przedstawiono przykładowy model drzewa TAN. Zgodnie z przedstawioną zasadą łatwo zauważyć, że atrybut A_2 jest zależny od atrybutu A_1 oraz klasy C.



Rysunek 2.1: Przykładowy model drzewa

Klasyfikator będzie wyznaczał klasę danego obiektu zgodnie z poniższym wzorem:

$$h(x) = \arg\max_{c \in C} P(c|a_1, ..., a_n) = \frac{P(c, a_1, a_2, ..., a_n)}{P(a_1, a_2, ..., a_n)} = \frac{P(c) \cdot \prod_{i=1}^{n} P(a_i|c, a_j)}{P(a_1, a_2, ..., a_n)}$$

Wyrażenie $P(a_i|c, a_j)$ odzwierciedla zależności w zbudowanym drzewie TAN - atrybut A_i jest zależny od A_j
Implementacja algorytmu i powyższy opis zostały stworzone na podstawie artykułu [1]

3 Implementacja

W formie pakietu języka R o nazwie *notSoNaiveBayes* zostały udostępnione następujące funkcje:

- *notSoNaiveBayes* - funkcja domyślna, która na podstawie danych uczących buduje model, który jest zwracany jako obiekt języka R.
- *predict* - funkcja, która na podstawie modelu uzyskanego w wyniku działania funkcji *notSoNaiveBayes* i wektora (macierzy) atrybutów próbek przypisuje im klasę.

Dodatkowo na potrzeby implementacji zostały napisane następujące funkcje:

- *buildDirectedTree* - funkcja, która na wejściu przyjmuje graf nieskierowany zakodowany w formie macierzy sąsiedztwa oraz numer wierzchołka grafu, który ma zostać korzeniem drzewa jakie powstanie jako argument wyjściowy tej funkcji.
- *kruskal* - jest to implementacja klasycznego algorytmu Kruskala, z jedną różnicą,napisana funkcja szuka maksymalnego drzewa rozpinającego.

4 Testy

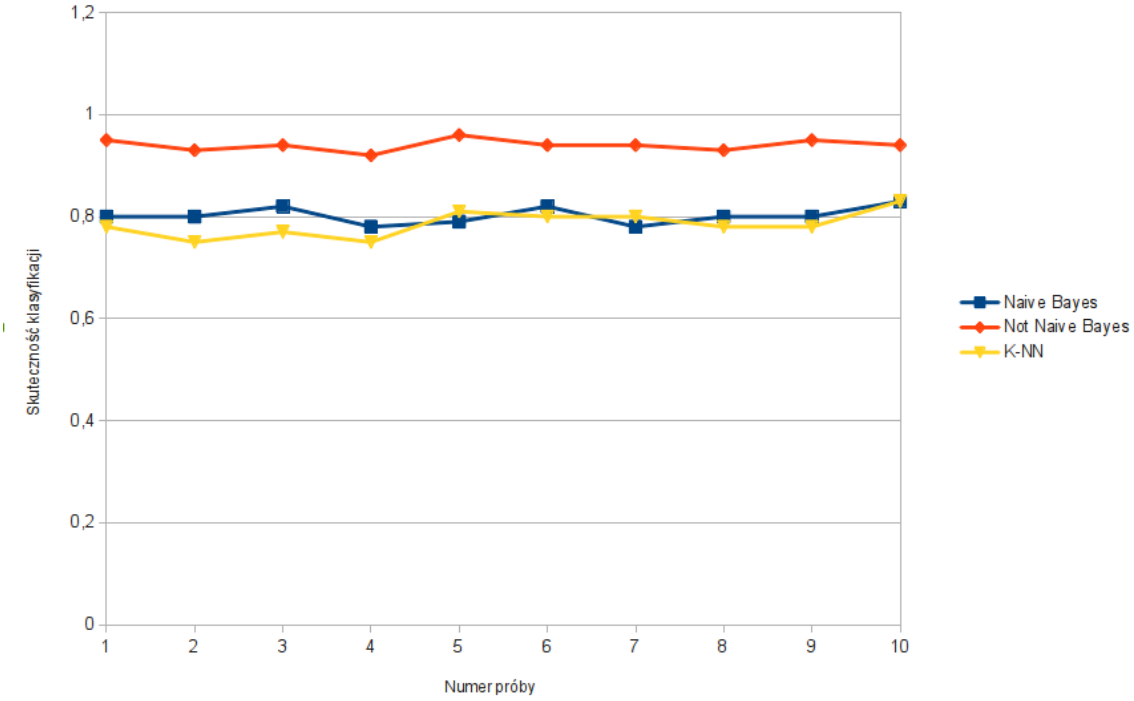
W celu sprawdzenia jakości zbudowanego klasyfikatora jego działanie zostało porównane z innymi algorytmami klasyfikacji dostępnymi w modułach języka R:

- naiwnym klasyfikatorem Bayesa - moduł

- algorytmem kNN - biblioteka RWeka [2]

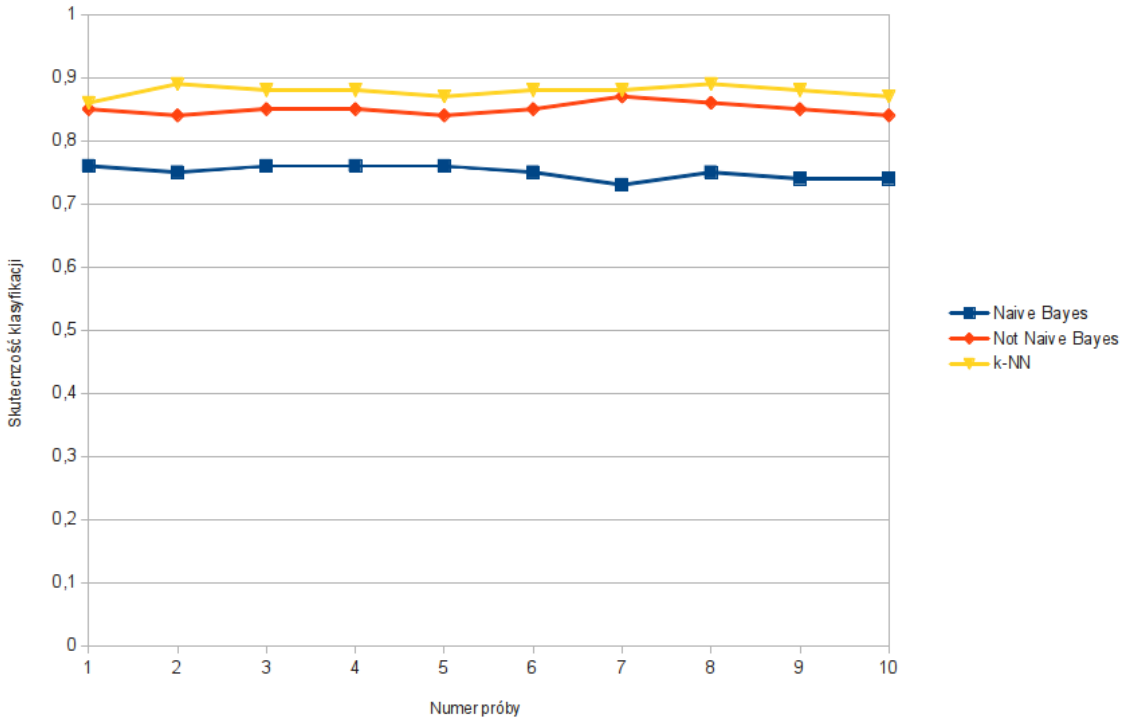
Do testów zostaną wykorzystane dane należące do repozytorium Uniwersytetu Kalifornijskiego w Irvine (Machine Learning Repository, University of California, Irvine), dostępne pod adresem: `http://archive.ics.uci.edu/ml/datasets.html`.
Testy zostały przeprowadzone za pomocą metody Bootstrap. Dla każdego zbioru danych rozlosowano dwie próbki - testową

5 Porównanie jakości klasyfikacji



Rysunek 5.1: Wyniki dla zbioru testowego car

6 Wnioski



Rysunek 5.2: Wyniki dla zbioru testowego nursery

Literatura

[1] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayessian Network Classifiers. *Machine Learning*, (29):131–163, 1997.

[2] Kurt Hornik. Package 'rweka'. *RWeka documentation*, 2013.