Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych
Informatyka

Metody Odkrywania Wiedzy

Klasyfikator bayesowski typu AODE

Nie-całkiem-naiwny klasyfikator bayesowski typu AODE
(averaged one-dependence estimators).
Porównania ze standardowym naiwnym klasyfikatorem
bayesowskim i innymi algorytmami klasyfikacji dostępnymi w R.

Wstępne założenia

Wykonali:
Paweł Guz
Mateusz Kędrzyński

Prowadzący:
dr inż. Paweł Cichosz

Warszawa, 10 IV 2015

1 Interpretacja tematu projektu

1.1 Wprowadzenie

Naiwny Klasyfikator Bayesa nie jest trudny w implementacji i często używany przy zagadnieniach klasyfikacji. Skuteczność jego działania zależy od dokładności estymowanych prawdopodobieństw atrybutów, które bazują na wzajemnej niezależności, co często nie jest spełnione. Jego odmiany: LBR jak i TAN polepszają jego działanie w zamian za znaczny koszt obliczeniowy. AODE ma być algorytmem zyskujący podobne rezultaty, bez nadmiernego narzutu obliczeniowego.

1.2 Naiwny klasyfikator Bayesa

Przyjmijmy następujące oznaczenia:

$x = \langle x_1, x_2, ..., x_n \rangle \in X$: przykład podlegający klasyfikacji

$y \in \{c_1, c_2, ..., c_k\}$: klasy klasyfikacji

$\hat{}$: wielkość związana z danymi trenującymi

T: zbiór zmiennych trenujących

Ze wzoru Bayesa otrzymujemy:

$$\Pr(y|\mathbf{x}) = \frac{\Pr(y, \mathbf{x})}{\Pr(\mathbf{x})} \tag{1}$$

a stąd:

$$\arg\max_y(\Pr(y|\mathbf{x})) = \arg\max_y(\Pr(y, \mathbf{x})) \tag{2}$$

Zakładając, że dane trenujące stanowią reprezentatywną próbkę, częstość wystąpienia danego zdarzenia w próbce będzie z wystarczająco dobrą aproksymacją równe prawdopodobieństwu tego zdarzenia.

$$\Pr(y, \mathbf{x}) \approx \hat{\Pr}(y, \mathbf{x}) \tag{3}$$

Jednakże, zakładając, że liczba atrybutów jest dostatecznie duża, możliwość wystąpienia próbki $(y, \mathbf{x})$ będzie względnie mała. Jedną z metod obejścia tego ograniczenia, jest estymacja tego prawdopodobieństwa poprzez inne prawdopodobieństwa, które możemy z większą pewnością uzyskać z próbek. Z definicji prawdopodobieństwa warunkowego otrzymujemy:

$$\Pr(y, \mathbf{x}) = \Pr(y) \Pr(\mathbf{x}|y) \tag{4}$$

W przypadku względnie małej liczby klas oraz względnie dużej liczby próbek, wartość $\Pr(y)$ jesteśmy w stanie wystarczająco dokładnie estymować. Jednakże estymowanie wartości $\Pr(\mathbf{x}|y)$ wciąż będzie kłopotliwe.

Naiwny Klasyfikator Bayesa zakłada niezależność atrybutów, co daje:

$$\Pr(\mathbf{x}|y) = \prod_{\mathbf{i=1}}^{\mathbf{n}} \Pr(\mathbf{x_i}|y), \tag{5}$$

Opierając się na wyżej wymienionych zależnościach, wybieramy klasę dla danej próbki na podstawie:

$$\arg\max_y(\hat{\Pr}(y) \prod_{i=1}^{n} \hat{\Pr}(\mathbf{x_i}|y)) \tag{6}$$

W trakcie treningu Naiwny Klasyfikator Bayesa potrzebuje dwie tablice: pierwsza przechowująca estymowane prawdopodobieństwa klas oraz druga przechowująca estymowane prawdopodobieństwa dla danej klasy w zależności od wartości atrybutu. Pierwsza z nich jest jednowymiarowa, druga natomiast dwuwymiarowa. Złożoność przestrzenna wyniesie: $\mathcal{O}(knv)$, gdzie v to średnia liczba wartości atrybutu, k liczba klas, a n to liczba atrybutów. Złożoność czasowa jest równa $\mathcal{O}(tn)$, gdzie t jest liczbą próbek trenujących.
W trakcie klasyfikacji, zaklasyfikowanie pojedynczego elementu posiada złożoność czasową $\mathcal{O}(kn)$ oraz przestrzenną: $\mathcal{O}(knv)$ - utrzymanie tablicy otrzymanej w wyniku treningu.

1.3 Alternatywne podejścia oparte na klasyfikatorze Bayesa

Naruszenie założenia niezależności atrybutów może prowadzić do nieakceptowalnych błędów. Alternatywne podejścia takie jak: LBR oraz TAN pozwalają na rozluźnienie tego założenia.

Przy podejściu LBR dla każdego przykładu $\mathbf{x}$ (klasyfikator tworzony w trakcie klasyfikacji) wyznaczany jest zbiór W wartości pewnych atrybutów (wykorzystując specjalizowane algorytmy zachłanne). Następnie zakłada się niezależność spośród pozostałych atrybutów (nie wchodzących w skład W) dla danego W oraz y. Stąd $\mathbf{x}$ jest klasyfikowane poprzez:

$$\arg\max_y(\hat{\Pr}(y|W)\prod_{i=1}^{n}\hat{\Pr}(x_i|y,W)) \tag{7}$$

W trakcie treningu złożoność pamięciowa i czasowa, niezbędna do przechowywania danych: $\mathcal{O}(tn)$, natomiast w trakcie klasyfikacji złożoność czasowa wynosi: $\mathcal{O}(tkn^2)$

Przy podejściu TAN konstruowana jest funkcja $p(x_i)$, która wyznacza atrybut zależny. x jest klasyfikowane poprzez:

$$\arg\max_y(\hat{\Pr}(y)\prod_{i=1}^{n}\hat{\Pr}(x_i|y,p(x_i))) \tag{8}$$

W trakcie trenowania tworzona jest trzywymiarowa tablica: do tablicy dwuwymiarowej tworzonej przy naiwnym klasyfikatorze Bayesa dodawany jest jeden wymiar atrybut-wartość, pomocny przy późniejszym konstruowaniu funkcji p. Stąd złożoność pamięciowa wynosi: $\mathcal{O}(k(nv)^2)$, natomiast złożoność czasowa $\mathcal{O}(tn^2)$. Następnie do wytworzenia funkcji p dochodzi rozważanie każdej pary atrybutów dla poszczególnych klas (złożoność wynosi $\mathcal{O}(k(nv)^2)$) oraz następnie generowane jest maksymalne drzewo rozpinające $\mathcal{O}(n^2 \log n)$.

Wyżej wymienione podejścia dają dobre rezultaty, jednakże są kosztowne obliczeniowo. Jest to spowodowane, głównie:
- wyborem modelu: zbiór W (LBR) oraz funkcja p (TAN)
- estymacją prawdopodobieństwa: w trakcie klasyfikacji (LBR), poprzez trójwymiarową tablicę (TAN)

1.4 Averaged One-Dependence Estimators (AODE)

Kolejnym algorytmem pozwalającym na osłabienie założenia o niezależności atrybutów oraz, ponadto, pozbawionym niektórych wad algorytmów TAN oraz LBR, jest Averaged One-Dependence (AODE). Opiera się na wybraniu atrybutów zależnych, których wartość dla danej próbki x w zbiorze trenującym występuje przynajmniej przyjęte m razy.

Wykorzystując zależność:

$$\Pr(y, \mathbf{x}) = \Pr(y, x_i) \Pr(\mathbf{x}|\mathbf{y}, \mathbf{x_i}) \tag{9}$$

oraz sumując po wszystkich atrybutach występujących w danych testowych dla danej wartości odpowiednią liczbę razy otrzymujemy ($\geqslant m$), otrzymujemy:

$$\Pr(y, \mathbf{x}) = \frac{\sum_{i:1 \leqslant i \leqslant n \ \wedge \ m \leqslant F(x_i)} \Pr(y, x_i) \Pr(\mathbf{x}|y, x_i)}{|i : 1 \leqslant i \leqslant n \ \wedge \ m \leqslant F(x_i)|} \tag{10}$$

Ponieważ mianownik jest taki sam dla każdej klas, wybór odpowiedniej klasy sprowadza się do rozważenia następującego zagadnienia:

$$\arg\max_y \Big(\sum_{i:1 \leqslant i \leqslant n \ \wedge \ m \leqslant F(x_i)} \hat{\Pr}(y, x_i) \prod_{j=1}^{n} \hat{\Pr}(x_j|y, x_i) \Big) \tag{11}$$

Złożoność pamięciowa podczas treningu jak i klasyfikacji jest taka sama i sprowadza się do utrzymywania tablicy trójwymiarowej: $\mathcal{O}(k(nv)^2)$, złożoność czasowa w trakcie treningu: $\mathcal{O}(tn^2)$, a w trakcie klasyfikacji: $\mathcal{O}(kn^2)$.

Oczekujemy lepszych rezultatów od algorytmu AODE ze względu na mniejszy nacisk na niezależność argumentów. Ponadto, zaletą AODE jest możliwość inkrementalnego nauczania. W przypadku dodatkowych danych albo aktualizacji liczby m należy jedynie zaktualizować tabelę.

2 Implementacje algorytmów

2.1 Implementacja algorytmu AODE

Tematem projektu będzie implementacja algorytmu AODE. Jego główna idea została przedstawiona w punkcie 1.4. W celu zwiększenia jakości algorytmu następujące kwestie zostaną rozważone:

wartości ciągłe
Implementacja zostanie zrealizowana w wersji akceptującej dane dyskretne. W przypadku atrybutów ciągłych niezbędna jest wcześniejsza dyskretyzacja.

zerowe i małe prawdopodobieństwa
W przypadku gdy dane testowe nie pokrywają wszystkich możliwych kombinacji, prawdopodobieństwo: $\Pr(x_j|y, x_i)$ lub $\Pr(y, x_i)$ we wzorze (**??**) może być równe 0. Operator iloczynu powoduje propagację zera, co może prowadzić do błędnych wyników związanych z niedostateczną ilością danych uczących i mniejszą zdolnością modelu do uogólniania. Aby temu zapobiec będziemy stosować m-estymację tych prawdopodobieństw:

$$\Pr(y = d, x_i = v_i) = \frac{|T_{x_i=v_i}{}^{\mathrm{d}}| + 1}{|T^{\mathrm{d}}| + k|x_i|} \tag{12}$$

$$\Pr(x_j = v_j|y = d, x_i = v_i) = \frac{|T^{\mathrm{d}}{}_{x_i=v_i, x_j=v_j}| + 1}{|T^{\mathrm{d}}{}_{x_i=v_i}| + k|x_i||x_j|} \tag{13}$$

brakujące wartości atrybutów
Jeżeli w danych trenujących będzie atrybut niesprecyzowany to przy estymacji nie będzie on brany pod uwagę.

2.2 Wykorzystane algorytmy porównawcze

Działanie algorytmy AODE zostanie porównane z wyżej wymienionymi algorytmami: naiwny klasyfikator Bayesa, LRB, TAN.

Ponadto, porównamy działanie powyższych algorytmów do innej klasy algorytmów klasyfikacji, jakimi są drzewa decyzyjne. Zbiór tych algorytmów polega na tworzeniu drzew w których każdy węzeł oznacza testowanie atrybutu natomiast każdy liść reprezentuję klasę decyzyjną. Proces klasyfikacji polega na przejściu od korzenia do liścia testując poszczególne warunki w węzłach. Jednym z popularniejszych algorytmów klasyfikacji za pomocą drzew decyzyjnych jest algorytm C.45. Drzewa tworzone są przy użyciu kryterium "gain ratio". Algorytm tworzy kolejne węzły dopóki liczba obiektów do podziału jest mniejsza niż pewna wartość progowa. Po utworzeniu drzewa można je przyciąć, tzn. niektóre węzły zastąpić liśćmi.

wykorzystane pakiety
Implementacje algorytmów będą pochodziły z następujących pakietów:

- Naiwny klasyfikator Bayesa - pakiet e1071
- C.45 - pakiet RWekka
- TAN - pakien bnlearn
- LRB - pakiet RWekka

3 Plan badań

3.1 Cel eksperymentów

Pierwszym celem przeprowadzenia eksperymentów jest walidacja poprawności implementacji algorytmu. Sprawdzenie będzie wykonane poprzez stworzenie modelu na podstawie prostych danych trenujących, a następnie sprawdzenie czy model wykonuje akceptowalną klasyfikację.

Drugim celem jest porównanie działania zaimplementowanego algorytmu AODE z innymi algorytmami klasyfikacji: naiwnym klasyfikatorem Bayesa, TAN, LRB oraz algorytmem wykorzystującym drzewa decyzyjne: C.45. Badany będzie wpływ liczby atrybutów na poprawność działania algorytmów, a także wpływ parametrów poszczególnych algorytmów.

3.2 Charakterystyka zbiorów danych

Wykorzystywane zbiory danych będą pochodziły z UCI Machine Learning Repository. Zbiory będą podzielone na trzy kategorie w zależności od ilości atrybutów:

- mała ilość atrybutów (do 10)
- średnia ilość atrybutów (od 10 do 100)
- duża ilość atrybutóW (od 100)

Zbiory danych zostaną podzielone na dane treningowe oraz dane testowe poprzez losowanie w proporcji 2:1. Następnie dla każdej pary algorytm - zbiór danych zostanie utworzony możliwie

najlepszy model, oparty na tych samych trenujących, poprzez dobranie odpowiednich parametrów algorytmów.

Założono, że algorytm AODE zostanie zaimplementowany jedynie w wersji akceptującej skończony zbiór wartości atrybutów, dane zawierające atrybuty ciągłe będą musiały być zdyskretyzowane.

3.3 Parametry algorytmów, których wpływ na wyniki będzie badany

Algorytm AODE ma następujące parametry:

m - minimalna ilość przykładów w danych uczących które zawierają atrybut o wartości x_i (patrz wzór (**??**))

3.4 Miary jakości i procedury oceny modeli

Jakość modelu będzie oceniana poprzez wyznaczenie błędu jako:

$$e = \frac{liczba\ blednie\ sklasyfikowanych\ przykladow}{liczba\ przykladow} \tag{14}$$

,
a także poprzez wybrane później wskaźniki jakości:

- misclassification error
- accuracy
- true positive rate
- false positive rate
- precision
- recall

Literatura

[1] Paweł Cichosz, *Materiały do wykładu z MOW*

[2] Geoffrey I. Webb Janice R. Boughton Zhihai Wang, *Not so naive Bayes: Aggregating one-dependence estimators*. School of Computer Science and Software Engineering

[3] Paweł Cichosz, *Data Mining Algorithms: Explained Using R*