Dokumentácia algoritmu KAD

Autori: Ivan Tkachenko, Roman Dzhulai

Obsah

1 Popis algoritmu

Kombinačná analýza dát (KAD) je štatistická metóda, ktorá dokáže generovať pravidlá pre znalostné systémy (ZS). Je založená na frekvenciách kombinácií atribútov. Vyberá spoľahlivé kombinácie najprv na základe fundovanosti. Kombinácia sa musí v súbore údajov vyskytovať v dostatočnom počte, aby prešla testom fundovanosti. Hodnotíme stupeň platnosti implikácie fundovanych kombinácií, musí byť vyššia ako nami stanovená hranica. Hľadáme vhodné kombinácie dĺžky od 1 do počtu atribútov. Pri každej iterácii porovnávame nové kombinácie s pravidlami, ktoré sú už zaznamenané v baze znalosti (BZ). Ak je platnosť implikácie nového pravidla väčšia ako pravidlá v BZ, pridáme nové pravidlo do BZ.

2 Popis dat

Dataset, ktorý sme použili, je dataset Mushroom Classification z Kaggle. Dataset obsahuje popisy vzoriek zodpovedajúcich 23 druhom žiabrových húb v hubách rodu Agaricus a Lepiota čerpaných z The Audubon Society Field Guide to North American Mushrooms (1981). Každý druh je identifikovaný ako určite jedlý alebo určite jedovatý. Pre našu implementáciu KAD sme zo súboru údajov vybrali triedy húb a atribúty cap-shape, cap-surface, cap-color, gill-size, stop-shape, závoj-typ, ring-number, population. Na demonštráciu sme vybrali 1000 záznamov z pôvodného datasetu. Údaje sme rozdelili na 70% pre trenovanie a 30% pre testovanie.

3 Implementácia

3.1 Funkcie pre prácu s kombináciami

3.1.1 Funkcia `get_combinations`

Táto funkcia akceptuje:

- `dataframe`: Dataframe s atribútmi.
- `length`: Parameter určujúci dĺžku kombinácií, ktoré sa majú generovať.

Pomocou modulu `itertools` (zabudovaného v Pythone) funkcia generuje

všetky možné kombinácie atribútov a hodnôt daného rozsahu. Kombinácie sú vrátené vo forme dataframe, kde:

- Každý riadok predstavuje jednu kombináciu atribútov.
- Hodnoty atribútov sú `None`, ak kombinácia daný atribút nepoužíva, inak obsahujú konkrétne hodnoty atribútu.

3.1.2 Funkcia `count_combinations_per_class`

Táto funkcia prijíma:

- `combinations`: Dataframe vygenerovaných kombinácií.
- `dataframe_attributes`: Dataframe atribútov.
- `dataframe_classes`: Dataframe tried.

Vstupné dataframy sú konvertované na numpy polia pre rýchlejšiu manipuláciu. Funkcia vypočíta počet výskytov každej kombinácie pre jednotlivé triedy a výsledky vráti ako počty výskytov.

3.1.3 Funkcia `get_reliable_combinations`

Parametre funkcie:

- `combinations`: Dataframe obsahujúci kombinácie.
- `combination_frequencies`: Zoznam frekvencií výskytov kombinácií.
- `rcr`: Požiadavka na fundovanosť, t.j. minimálny počet výskytov kombinácie v dátach.
- `n_records`: Počet analyzovaných záznamov.

Funkcia prechádza kombinácie a identifikuje tie, ktoré spĺňajú podmienku `rcr`. Vráti dva zoznamy:

- Indexy fundovaných kombinácií.
- Samotné fundované kombinácie.

3.1.4 Funkcia `get_reliable_combinations_stat`

Parametre:

- `combination_frequencies`: Frekvencie výskytu kombinácií.
- `class_counts`: Frekvencie výskytu kombinácií pre jednotlivé triedy.
- `reliable_indexes`: Indexy fundovaných kombinácií.
- `riv`: Požiadavka na platnosť implikácie.

Funkcia vypočíta platnosť implikácie (`iv`) pre každú kombináciu ako podiel najvyššej frekvencie triedy a celkovej frekvencie kombinácie. Kombinácie s $\texttt{iv} \geq \texttt{riv}$ sa pridávajú do zoznamu vhodných kombinácií. Návratová hodnota obsahuje:

- Index kombinácie v dataframe.
- Kombináciu vo forme zoznamu atribútov.
- Hodnotu `iv`.

3.1.5 Funkcia `elems_comp`

Porovnáva dve pravidlá (a, b) na základe atribútov. Vráti:

- `True`, ak je kombinácia b podmnožinou a.
- `False`, inak.

3.2 Trieda `KAD`

Trieda `KAD` implementuje algoritmus na jednoduchú inicializáciu a použitie. Pre vytvorenie inštancie je potrebné poskytnúť:

- `X_train`: Záznamy atribútov.
- `Y_train`: Záznamy tried.

Premenné triedy:

- `rules`: Uložené pravidlá.
- `attributes`: Záznamy atribútov.
- `classes`: Záznamy tried.
- `REQUIRED_COMBINATION_RELIABILITY (rcr)`: Parameter fundovanosti.

- `REQUIRED_IMPLICATION_VALIDITY (riv)`: Parameter platnosti implikácie.
- `N_RECORDS`: Počet záznamov v tréningových údajoch.
- `N_ATTRIBUTES`: Počet atribútov.

3.2.1 Funkcia `process`

Hlavná funkcia algoritmu:

1. Vygeneruje kombinácie pomocou `get_combinations()`.
2. Spočíta frekvencie kombinácií pre rôzne triedy pomocou `count_combinations_per_class()`.
3. Vyberie fundované kombinácie cez `get_reliable_combinations()`.
4. Identifikuje kombinácie s platnou implikáciou cez `get_reliable_combinations_stat()`.
5. Nové kombinácie porovná s existujúcimi pravidlami a pridá ich, ak sú vhodné.

3.3 Testovanie

3.3.1 Funkcia `format_rule`

Pomocná funkcia na formátovanie pravidiel. Vytvorí reťazcovú reprezentáciu pravidla na vizualizáciu v konzole.

3.3.2 Proces inicializácie a testovania

1. Načítanie dát do dataframe.
2. Rozdelenie dát na atribúty a triedy.
3. Rozdelenie dát na tréningovú (70%) a testovaciu (30%) sadu.
4. Inicializácia `KAD` a generovanie pravidiel pomocou `process()`.
5. Usporiadanie pravidiel podľa `iv`, formátovanie a vizualizácia.
6. Testovanie pravidiel na testovacej sade a vyhodnotenie pomocou `sklearn`.

Funkcia tiež podporuje klasifikáciu vlastných prípadov na základe vygenerovaných pravidiel.

4 Výsledky testovania

Extrahované pravidlá

Pravidlá
IF gill-size:n AND stalk-shape:t THEN CLASS:p
IF gill-size:n AND stalk-shape:t AND veil-type:p THEN CLASS:p
IF gill-size:n AND stalk-shape:t AND ring-number:o THEN CLASS:p
IF gill-size:n AND stalk-shape:t AND population:v THEN CLASS:p
IF gill-size:n AND stalk-shape:t AND veil-type:p AND ring-number:o THEN CLASS:p
IF gill-size:n AND stalk-shape:t AND veil-type:p AND population:v THEN CLASS:p
IF gill-size:n AND stalk-shape:t AND ring-number:o AND population:v THEN CLASS:p
IF gill-size:n AND stalk-shape:t AND veil-type:p AND ring-number:o AND population:v THEN CLASS:p
IF gill-size:b AND stalk-shape:t THEN CLASS:e
IF gill-size:b AND stalk-shape:t AND veil-type:p THEN CLASS:e
IF gill-size:b AND stalk-shape:t AND ring-number:o THEN CLASS:e
IF gill-size:b AND stalk-shape:t AND veil-type:p AND ring-number:o THEN CLASS:e
IF gill-size:n THEN CLASS:p
IF cap-shape:x AND gill-size:b THEN CLASS:e
IF gill-size:b THEN CLASS:e

Tabuľka 1: Súbor pravidiel získaných modelom.

Confusion Matrix

	Predikovaná trieda	Skutočná trieda
	e	**p**
e	140	10
p	59	91

Tabuľka 2: Confusion matrix.

Confusion matrix ukazuje nasledujúce:

- **Pravdivé pozitíva (TP)**: 140 vzoriek bolo správne klasifikovaných ako trieda ’e’.
- **Falošné pozitíva (FP)**: 59 vzoriek triedy ’p’ bolo nesprávne klasifikovaných ako trieda ’e’.
- **Pravdivé negatíva (TN)**: 91 vzoriek bolo správne klasifikovaných ako trieda ’p’.
- **Falošné negatíva (FN)**: 10 vzoriek triedy ’e’ bolo nesprávne klasifikovaných ako trieda ’p’.

Classification report

Trieda	Presnosť	Záchyt	F1-Skóre	Podpora
e	0.70	0.93	0.80	150
p	0.90	0.61	0.73	150
Accuracy			0.77	300
Priemerný (macro)	0.80	0.77	0.76	300
Priemerný (weighted)	0.80	0.77	0.76	300

Tabuľka 3: Klasifikačná správa.

Klasifikačná správa ukazuje nasledujúce kľúčové body:

- **Presnosť**: Pomer pravdivých pozitív medzi všetkými pozitívnymi predikciami. Pre triedu ’e’ je presnosť 0.70, čo znamená, že 70% predikcií triedy ’e’ je správnych.
- **Záchyt**: Pomer pravdivých pozitív medzi všetkými skutočnými pozitívnymi vzorkami. Pre triedu ’e’ je záchyt 0.93, čo naznačuje, že model správne identifikuje 93% všetkých vzoriek triedy ’e’.
- **F1-Skóre**: Harmonický priemer presnosti a záchytu. Pre triedu ’e’ je F1 skóre 0.80, čo znamená dobrú rovnováhu medzi presnosťou a záchytom.
- **Podpora**: Počet skutočných výskytov každej triedy v dátach (150 pre obe triedy ’e’ a ’p’).

Model dosahuje celkovú Accuracy 77%, pričom má relatívne vysoký záchyt pre triedu ’e’, ale môže získať lepšie výsledky v oblasti presnosti pre triedu ’e’ a záchytu pre triedu ’p’.

5 Záver

V tejto práci sme implementovali algoritmus KAD na extrakciu pravidiel zo súboru dát týkajúceho sa klasifikácie húb. Cieľom bolo využiť pravidlá na efektívnu klasifikáciu a predikciu tried húb na základe ich atribútov.

Pomocou algoritmu KAD sme získali sadu pravidiel, ktoré popisujú vzory a závislosti medzi atribútmi húb a ich triedami. Tieto pravidlá umožňujú rýchlu a interpretovateľnú analýzu dát a slúžia ako nástroj na klasifikáciu nových vzoriek.

Výsledky ukazujú, že model dosiahol celkovú Accuracy 77% pri klasifikácii dvoch tried húb. Taktiež sme vyhodnotili výkon modelu pomocou Confusion matrix a Classification report, čo nám poskytlo cenné informácie o silných a slabých stránach modelu.

Celkovo sme preukázali, že algoritmus KAD môže byť úspešne aplikovaný na klasifikáciu.