RPH. Naive Bayes spam filter

Vadym Ostapovych, Roman Novikov

4. ledna 2022

Úvod

Úkolem dané úlohy je vytvoření spam filtru, který bude umět odfiltrovat spam a korektní emaily. Pro její řešení bylo rozhodnuto vytvořit klasifikátor založený na Bayesove podmíněné pravděpodobnosti.

1.Matematický princip filtru

Naive Bayes je trénovací filtr, který vyžaduje pevnou sadu trénovacích dat pro určení parametrů klasifikátoru. Za nich budeme považovat klasifikované emaily(třídy)

- `SPAM` - email je spamový.
- `OK` - email je korektní(ham).

Hlavní idea uvedeného spam filtru je z sady trénovacích emailů zformovat svůj slovník "rysových slov"(feature words), taký k nim mohou patřit nějaké větší fráze. Pod slovníkem budeme rozumět dvě množiny slov, které budou patřit nebo v spam, nebo v ham emailech. Pro konkrétní slovo budeme počítat dva typy slovníků:

1) Kolikrát slovo se objevuje v konkrétní třídě(class), např. při účení slovo "slave"mezi třemi spam emaily, objevilo 5-krát., proto do toho slovníku přidáme s klíčem "slave"bude číslo 5.
2) Kolikrát slovo se objevuje v konkrétní třídě na počet emailů tohoto classu. Uvažujeme minulý příklad se slovem "slave", ale nechť z těch 3 spam emailu, jen ve dvou z nich se objevily ty 5 slov, proto do toho slovníku přidáme číslo 2.

Z dvou uvedených slovníku musíme vypočítat parametry filtru pro dalsi klasifikaci emailu. V nasim pripade to budou podminene pravdepodobnosti "likelihoods"pro jednotlivá slova, že email je spam, resp. ham, za podminky, že obsahuje nějaké slovo. Uvedemé vzorec pro výpočet spamovosti emailu

$$P(S|W) = \frac{N_{W|S}}{N_{spam\ words} + N_{uniq}} \cdot \frac{N_{W|per\ spam\ email}}{N_{spams}} \tag{1}$$

kde

- $P(S|W)$ - pravděpodobnost, ze email je spam za podmínky, ze obsahuje slovo W.
- $N_{W|S}$ - počet slova W ve všech emailech.
- $N_{spam\ words}$ - celkový počet objevovaní rysových slov v spam emailech.
- N_{uniq} - celkový počet rysových(unikátních) slov, klíčů.
- $N_{W|per\ spam\ email}$ - v koliká spamových emailech se objevuje dané slovo.
- N_{spam} - celkový počet spamových emailů.

Z rovnice lze vidět, že podmíněná pravděpodobnosti se skládá z dvou jednotlivých zlomků, které identifikuji Bayesove pravdepodobnosti objevováni slova vůči celkovému poctu spamových slov a vůči celkovému poctu spamových emailů. Stejný vzorec použijeme pro vypočet pro hamovosti slova, kde budeme počet už vůči hamovým slovům a hamovým emailem. Po vypočtu dostaneme dva slovníky s pravděpodobností pro hamové a spamové slova F_{hams}, F_{spams}. Z nich už budeme klasifikovat nové emaily.

V novém emailu nalezneme unikátní nebo rysová slova a přidáme do nejaké množiny např. F_{email}, počet slov v emailu nás nezajímá. Pak z té množiny vypočítáme

$$F_{email} \cap F_{spams} - \text{pravdepodbnosti slov, ktere se objevily v spamu behem trenovani} \quad (2)$$

$$F_{email} \cap F_{hams} - \text{pravdepodbnosti slov, ktere se objevily v hamu behem trenovani} \quad (3)$$

Pro každou z nalezených množin spočítáme pravděpodobnost, že email je spam, resp. ham. Ukážeme výpočet pro spam, pro ham je totiž stejný.

$$P_{S_{email}} = P(S) \cdot P(S|W_1) \cdot P(S|W_2) \cdot ... \cdot P(S|W_n \quad (4)$$

kde

- $P(S|W_i)$ - pravděpodobnost, že email je spam za podmínky, že obsahuje slovo W_i.
- P_S -pravděpodobnost spamu v novému emailu, muže byt vypočten z poctu spam a ham emailů nebo rozumně nastaven.

Z pravděpodobností $P_{S_{email}}$, $P_{H_{email}}$ už dáváme novy email do třídy podle vyšší pravděpodobnosti

$$\text{Email class} = argmax(P_{S_{email}}, P_{H_{email}}) \quad (5)$$

2.Implementace

Všechna trénovacé část je implementována v trénovacím corpusu. V třídě `Filter` jen zavoláme metodu z trénovacího corpusu pro nastavení parametrů filtru.

Vyber rysových slov

Pro vypočet rysových slov potřebujeme dva slovníky, v každém z nich klíčem je jednotlivý class SPAM nebo OK. Hodnotami klíčů jsou Countery, které identifikuji počet slov. První slovník, který představuje celkový počet slov v jednotlivých třídách slov se jmenuje `vocabluary`, druhy představuje počet těch slov na každý email, jmenuje `words_per_email` a oba jsou atributy trénovacího corpusu.
Budeme postupně procházet jednotlivé emaily každé třídy, pro parsing emailu použijeme modul email, který vyčleňuje hlavni content. Z toho hlavního contentu pomoci regulárních vyrazů nalezneme adresy urls, názvy emailu, které se zmiňují a typy contentu, které jsou současti modulu email. To můžeme povazovat za jendnotlivé fráze. Dále spočitáme unikátni slova v tomto emailu, pomoci regulárního vyrazu odstraníme všechna písmena, která nejsou z anglické abecedy, a nahradime je prázdným symbolem, uděláme všechna písmena malými a pomoci funkce split() zformujeme list slov. Taky dáváme pozor na minimální a maximální délku slov a jestli to slovo není v stop words, slova, od kterých v anglickém jazyce veta nemění moc smysl. Délky byly nastavené na [2, 100] a stop slova byly použite stejné jako v modulu nltk. Uvedeme příklad vyber unikátních slov na věte `Van is a Dungeon Master since 1990`

- is, a jsou stop wordy.
- 1990 není patří do anglické abecedy, je to číslo. áme ostatní slova do malého písmena a jelikož respektuji délky, uděláme list [”van”, ”dungeon”, ”master”]

Counter pomáhá při výpočtu objevovaní slov. Výsledkem by bylo Counter(”van”:1, ”dungeon”:1, ”master”:1). Na zaklade toho uděláme dva slovníky.

Výpočet podmíněných pravděpodobností z slovníků

Pro vypočet podmíněných pravděpodobností byl použit princip laplace smoothing pro lepší predikci. Např. my představujeme, ze slovo, které se objevilo pouze v hamu se taky objevilo v k-krat spamových emailu, stejně pro ham. Pokud A - množina spamových slov, B - množina hamových slov, lze to zapsat následně pro k = 1, co je v podstatě defaultní konstanta

$$A_{new} = A \cup (A \setminus B) \quad (6)$$

$$B_{new} = B \cup (B \setminus A) \quad (7)$$

Z toho zápisu se objevuje, ze délka klíče nebo slov pro oba nové slovníky musí být stejná. A proto ve vzorci pro vypočet unikátních slov N_u můžeme použit libovolný z Counteru bud’ pro ham nebo spam.

Predikce třídy

Pro predikci classu budeme povazovat

$$P(S) = P(H) = 0.5 \tag{8}$$

jelikož nelze předpokládat z trénovacích dat, ze pravděpodobnost, ze novy email patří do nejaké třídy je stejná jako v trénovacích datech. Taky pro vypočet predikční pravděpodobnosti byly použité logaritmy, abychom se vyhnuli floating-point underflow. Proto prakticky se koná srovnáni logaritmu

$$log(P_{S_{email}}) = log(P(S)) + \sum_{i=1}^{n} log(P(S|W_i)) \tag{9}$$

$$log(P_{H_{email}}) = log(P(H)) + \sum_{i=1}^{n} log(P(H|W_i)) \tag{10}$$

A proto rozhodnutí se koná jako

$$\text{Email class} = argmax(log(P_{S_{email}}), log(H_{S_{email}})) \tag{11}$$

3.Trenovani klasifikatoru

Evaluace klasifikatoru se konala na datasetech 1, 2 uvedenych v zadani. Trenovani se konalo taky na uvedenych datasetech. Výsledky jsou v tabulce

Trénovaci dataset	Evaluační dataset	Kvalita filtru q
1	1	0.953
1	2	0.931
2	1	0.836
2	2	0.97

Tabulka 1: Porovnaní kvality filtru

Lze videt, ze kvalifikator má dostatečně výsokou kvalitu, možný problém neúplné klasifikace je nesprávná detekce kličových slov nebo malý počet kličových fraz. Různá možná vylepšení jako počet unikátních slov v novém emailu nebo výpočet $P(S), P(H)$ z trenovaci davalo horší výsledky, proto uvedená implementace je považovana za nejlepší dosahnutou. Na Brutu byly dosanuté nasledující výsledky Z vysledku turnaje se objevilo,

Trénovaci dataset	Kvalita filtru q	Misto v turnaji
1	0.932	6
2	0.855	8
3	0.853	13

Tabulka 2: VÝsledky turnaje

ze filter má velký počet False Negative a malý počet False Positives, co ukazuje, že nesprávně detekuje ham emaily a dost čast je považuje za spam, co taky potvrzuje možnou nespravnou detekci unikátních slov.

Vysledek prace

Přinos práce každého z učastniků byl rovnocenný 50 na 50. Každý udělal něco pro jednotlivé části práce. Návrh filtru je docela správná, a detekce je vysoká, ačkoliv není idealní. Možné zlepšení je dodělat detekci správných slov, co lip by šlo použitím knihovny nltk. Jinak pro tuto úlohu byl dosahnut maximální počet bodů.

Pouzita literatura

- Wikimedia Foundation. (2021, December 3). Naive Bayes spam filtering. Wikipedia. Retrieved January 4, 2022, from https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering.
- Spam filter in Python: Naive bayes from scratch. KDnuggets. (n.d.). Retrieved January 4, 2022, from https://www.kdnuggets.com/2020/07/spam-filter-python-naive-bayes-scratch.html

- Motwani, S. (2020, August 8). Email spam filtering using naive Bayes classifier. Springboard Blog Email Spam Filtering Using Naive Bayes Classifier Comments. Retrieved January 4, 2022, from https://in.springboard.com/blog/email-spam-filtering-using-naive-bayes-classifier/