Spam Filter

Princip fungování

# Úvod

Základ fungování filtru zní víceméně jednoduše. V základu máme nějaký naivní filtr, který „krmíme“ sety testovacích e-mailů. Poté co je filtr aplikován na celý na celý set se spočítá confusion matrix, která určí, jak kvalitní aktuální podoba filtru je. Po každé iteraci se nějak upraví podoba filtru tak, aby byl při příštím opakování výsledek pokud možno lepší. Iterujeme tak dlouho, dokud se nám nepodaří dosáhnout nějakého daného „sweet spotu“.

# Statistické okénko

Základem našeho spam filtru je takzvaný „naivní Bayesův algoritmus“, který se zakládá na Bayesově větě. Tato slouží k nalezení pravděpodobnosti jevu na základě jeho dřívějšího výskytu. V případě aplikace na jev, ve kterém se vyskytuje více různých pravděpodobností je ideální, protože bere každý z jevů jako nezávislý, a tedy jako jev, který má stejný vliv na konečný výsledek, jako všechny ostatní. Neexistuje jev, kterému by byla přiznávána větší váha než nějakému jinému jevu

# Určení „Bag of words“

Teď vlastně hned upravím tvrzení z předchozího odstavce. V případě spam filtru je každé jedno slovo „jev“, který má svou vlastní pravděpodobnost, tedy pokud se jedno slovo vyskytne vícekrát, celková váha tohoto slova bude ***n*** násobek původní pravděpodobnosti kdy ***n*** je počet výskytů slova v textu.

Zpočátku však nelze určit, v jakých případech je slovo spam a v jakých ne (záleží totiž na tematice daného textu, v případě e-mailů se může například u jednotlivých uživatelů lišit oblast témat, kterých se týkají jednotlivé spam e-maily). Z tohoto důvodu je nejdříve nutné provést sérii cvičných hodnocení, na kterých se filtru „řekne“, jaké e-maily byly spam.

Tayloring pravděpodobností

Filtr si na základě informací sám analyzuje obsah e-mailu a „společné jmenovatele“ zařadí mezi seznam nežádoucích slov. Tento seznam a celkový výpočet pravděpodobnosti výskytu pak bude logicky přesnější s každým dalším testem. V konečné fázi (po sladění confusion matrix na hodnoty, které se co nejvíce blíží ideálu, tzn. nejmenší možný výskyt false positive a false negative výsledků) budou určitá slova označena za „vysoce pravděpodobná“ vzhledem k častějšímu výskytu v e-mailech označených jako spam (pravděpodobně uděláme v nějaké číselné formě (10 - vysoce pravděpodobně spam?) a některá jako „téměř neškodná“.

# Celková pravděpodobnost e-mailu

Konečné vyhodnocení, zda se jedná o ,SPAM’ flagged e-mail, nebo ne, závisí na celkovém výskytu slov s vyšší pravděpodobností spamu. V tento moment přichází na scénu tzv. Bayesova pravděpodobnost [[1]](#footnote-2). Pro tento výpočet upravíme základní Bayesův vzorec[[2]](#footnote-3) a spočítáme pravděpodobnost pro všechna slova e-mailu[[3]](#footnote-4). Bayesova pravděpodobnost se dělí na dva typy: apriorní a posteriorní. Oba budou v našem případě použity. Apriorní aplikace pravděpodobnosti bude aplikována poté, co filtr projde „učící“ fází, spočívá ve vycházení z předchozích zkušeností (tedy z učících datasetů). Číslo udávající apriorní pravděpodobnost bude Obsah obrázku text, snímek obrazovky, číslo, Písmo

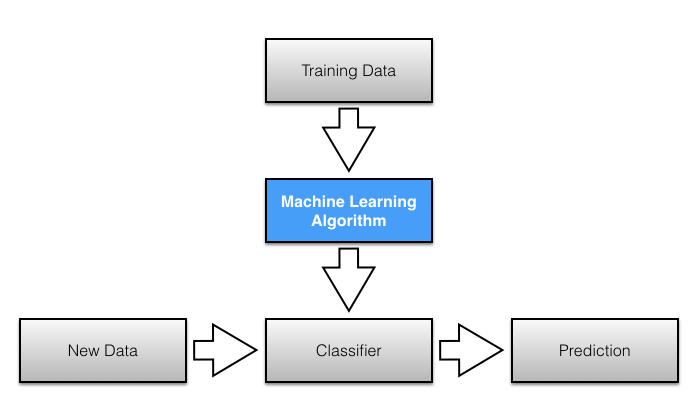
Popis byl vytvořen automatickypodíl ,SPAM‘ flagged e-mailů vůči jejich ,OK‘ protějškům. Hodnotu apriorní pravděpodobnosti má také každý jeden jev (tedy každé slovo). viz obr. 1

Pokud tato překročí určitou hranici (to je další moment, kdy to může být trochu tricky), vyhodnotí filtr daný e-mail jako spam.

Obrázek

Figure

# Inicializace filtru a jeho postupný trénink

Jako základní filtr lze zvolit prakticky jakýkoli zcela jednoduchý filtr, úplně první iterace bude vždy testovací. Filtr si po prvním načtení všech mailů a následném porovnání s confusion matrix spočítá pravděpodobnosti u jednotlivých slov a upraví své chování.

Obrázek

Figure

# Práce s rozdílnými datasety

Vzhledem k možné rozdílnosti tematiky ,SPAM’ e-mailů u jednotlivých datasetů je na místě, aby byly pravděpodobnosti u jednotlivých slov defaultně nastaveny relativně neutrálně a filtr se rozhodoval až po první iteraci. Pokud filtr zachytí slovo, které ještě nikdy nezaznamenal, tzv. vzácné slovo, jsou dvě metody, jak s touto situací naložit. První je slova s nulovou pravděpodobností jednoduše ignorovat. Druhá, složitější, je aplikace tzv. Laplaceova vyhlazení (Laplace Smoothing).[[4]](#footnote-5)

# Další ignorovaná slova

Rozhodně je nezbytné určit si tzv. běžná slova, tj. slova, bez kterých jednoduše nelze poskládat jakákoli věta (typicky jednoslabičná slova). Tato slova budou mít, vzhledem k jejich prakticky neustálému výskytu, ,SPAM‘ pravděpodobnost cca. 0,5. Do celkového vzorce pro výpočet je tedy nejvhodnější brát pouze ta slova, která se svou pravděpodobností blíží jednomu, či druhému kraji spektra (tzn. slova s pravděpodobností kolem 0.0, či slova ≅ 0,8)

# Nevalidní data

Tzv. Bayesian poisoning je situace, kdy jsou do ,SPAM‘ e-mailů přidávána náhodná slova s cílem degradovat účinnost filtru jeho zmatením. Je možné, že tato situace v průběhu testování nastane, ale nepředpokládám to.

# Implementace v Pythonu

První kroky (to, co se odevzdávalo individuálně) krásně připravily půdu pro finální implementaci. Víceméně všechnu tools potřebné pro vytvoření testovacího a filtrovacího prostředí už máme. Pro pochopení vytváření se slovy doporučuji [tohle](https://towardsdatascience.com/naïve-bayes-spam-filter-from-scratch-12970ad3dae7) „naivní“ řešení. Jedná se o aplikaci všeho výše zmíněného do kódu. Případně [tenhle](https://github.com/jieren123/SpamFilter-NavieBayes) projekt, tam ale spatřuji lehké nedostatky, zejména absenci ukládání do souboru, která v našem případě bude nutná.

# Zdroje

<https://en.wikipedia.org/wiki/Bayes%27_theorem>

<https://en.wikipedia.org/wiki/Posterior_probability>

<https://medium.com/@dataproducts/stop-spam-in-its-tracks-create-a-naive-bayes-spam-filter-with-python-a5b72b8b4b8f>

<https://towardsdatascience.com/laplace-smoothing-in-naïve-bayes-algorithm-9c237a8bdece>

<https://towardsdatascience.com/naïve-bayes-spam-filter-from-scratch-12970ad3dae7>

<https://github.com/jieren123/SpamFilter-NavieBayes>

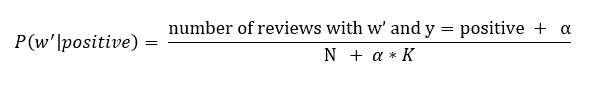
1. Obsah obrázku černá, tma

   Popis byl vytvořen automaticky

   A a B jsou jednotlivé události, kdy P(B) ≠ 0, P(A|B) je pravděpodobnost A je podmínková pravděpodobnost, tedy P(A), pokud B ≠ 0 [↑](#footnote-ref-2)
2. 

   Pr(S|W) je pravděpodobnost, že e-mail je spam, pokud obsahuje dané slovo (W). Pr(H) představuje pravděpodobnost, že zpráva není spam, Pr(H|W) je pravděpodobnost, se kterou se dané slovo vyskytuje v e-mailech, které nejsou spam. [↑](#footnote-ref-3)
3. 

   Kdy v tomto vzorci p je celková pravděpodobnost, že zpráva je spam a (p1, …, pn) jsou dílčí pravděpodobnosti pro jednotlivá slova v textu. Kvůli reprezentaci extrémně malých čísel v CPU se vyplatí použít spíše upravený vzorec: kdy Obsah obrázku černá, tma

   Popis byl vytvořen automaticky [↑](#footnote-ref-4)
4. 

   Kdy w‘ je slovo s nulovou pravděpodobností, K počet výskytů w‘ v aktuálním datasetu, N počet hodnocení kdy y je pozitivní. Alfa je nějaký vyhlazovací koeficient, většinou se používá 1 pro úplnou eliminaci nulové pravděpodobnosti. [↑](#footnote-ref-5)