Proste algorytmy klasyfikacji tekstu (kNN)

*Maks-ym Styczeń 2017*

1. **Opis zadania**

W ramach zadania - ”Proste algorytmy klasyfikacji tekstu (kNN). Porównania ze standardowymi algorytmami klasyfikacji dostepnymi w R.” zaimplementowano klasyfikator który nastepnie został wykorzystany w zagadnieniu rozpoznawania spam wiadomosci SMS.

Do nauki i oceny klasyfikatorów wykorzystano zbiór *SMS Spam Collection* z repozytorium UCI.

1. **Opis algorytmu**

Algorytm k-najbliższych sąsiadów (k-NN) jest metodą nie parametryczną, stosowaną do klasyfikacji i/lub regresji. W obu przypadkach, wejście składa się z k najbliższych przykładów szkoleniowych w przestrzeni. Wynik wyjściowy zależy od tego, czy K-NN stosuje się klasyfikacji lub regresji.

Będziemy rozpatrzywali stosowanie algorytmu k-NN w klasyfikacji.

W klasyfikacji wyjściem jest przynależność przykładu do klasy (predykcja). Obiekt zostaje sklasyfikowany większością głosów swoich sąsiadów, a obiekt jest przypisany do klasy najczęściej spotykanej wśród swoich *k* najbliższych sąsiadów (k jest dodatnią liczbą całkowitą, zazwyczaj małe). Jeśli *k* = 1, po czym przedmiot jest po prostu przypisany klasie tego jednego najbliższego sąsiada.

k-NN jest rodzajem *lazy learning*, w którym funkcja jest aproksymowana tylko lokalnie, a wszystkie obliczenia odłożone są do momentu klasyfikacji. Algorytm k-NN jest jednym z najprostszych  algorytmów uczenia maszynowego.

Sąsiedzi są pobierane z zestawu obiektów, dla których klasa (sklasyfikowane *k‑*NN) lub wartość właściwości obiektu (dla regresji *k*‑NN) jest znana. To może być traktowane jako zestaw szkoleniowy dla algorytmu, choć nie jest wymagany żadny wyraźny etap szkolenia.

Najbliższe sąsiedzi są wyszukiwane przez obliczenia metryk dla danego przykładu ze zbioru testującego z każdym przykładem z modeli, wykorzystanej dla klasyfikacji. Te metryki pokazują stopień podobieństwa między dwoma przykładami.

W danym projekcie były wykorzystane metryki Euklidowa oraz kosinusowa.

Metryka Euklidesowa:

,

gdzie *v2*, *v1* – przykłady, dla których szukamy normę.

Metryka kosinusowa:

1. **Opis eksperymentów**

Celem zadania było odniesienie (określenie klasy) przekazanej wiadomości SMS do jednej z klas: „spam” albo „ham”.

1. **Charakterystyka zbioru danych**

Danymi do uczenia, testowania i oceny klasyfikatorów był wybrany zbiór SMS Spam Collection archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection.

Wybrany zbiór danych jest zbiorem etykietowanych wiadomości które zostały zebrane na potrzeby badania spamu wysyłanego SMS'ami. Zbiór składa się z 4827 zwykłych wiadomości oraz z 747 wiadomości zawierających spam. Cała kolekcja wiadomości jest umieszczona w jednym pliku, wśród nich występują zestawy wiadomości oznaczonych jako niepożądane ("spam") oraz zestawy wiadomości pożądanych ("ham").

Ponizej kilka przykładów:

* ham Why are u up so early?
* ham Ya that one is slow as poo
* spam Bloomberg -Message center +447797706009 Why wait? Apply for your future http://careers. bloomberg.com
* ham "Im on gloucesterroad what are uup to later?"
* ham Yes:)here tv is always available in work place..
* spam YES! The only place in town to meet exciting adult singles is now in the UK. Txt CHAT to 86688 now! 150p/Msg.
* ham Lol no ouch but wish i'd stayed out a bit longer

1. **Implementacja algorytmu**

Algorytm k-NN implementowano w pliku knn.R

W ramach wstępnego przetwarzania danych, zostały wykonane następujące kroki:

* zmniejszyliśmy tekst do małych liter;
* wyodrębniono strony internetowe, wszystkie strony zastąpiono na „wbst” (od słowa „website”);
* ciągi numeryczne z 11 znaków wyodrębniono jako numer teleponu, taki ciągi są zmienione na „phnnmbr” (od słów „phone number”);
* usunęliśmy wszystkie znaki interpunkcji;
* wyodrębniono końcówki SMS’ów „xxx” jako „msgend” (od słów „message ending”), w tych miejscach na końcach używane były ciągi symbolów „x” zamias imieni nadawcy
* wyodrębniono „xxx” w środku powiadomień
* usunięte nadmierne spacje

Z wczytanego pliku tekstowego jest tworzony data frame, w którym są 2 kolumny. Pierwsza zawiera klasy powiadomień (ham/spam), druga – treść SMS. Pod czas odczytu pliku i tworzenia data frame’u również są usunięte tak nazywane „stop words”. Dla tego jest stworzona metoda read\_lines\_classified.

Dalej jest utworzona macierz zer i jedynek, w której kolumny to są słowa (swojego rodzaju słownik naszego modelu), a indeksy wierszy odpowiadają SMS’om w data frame’ie. Dla tego służy metoda create\_word\_matrix.

Potem obieramy podział naszego zbioru na trenujący oraz testowy. Niestety, przez bardzo wielką złożoność algorytmu nie udało się wybierać dość duże zbioru trenujące i testujące. Dlatego dla tworzenia zbioru trenującego wykorzystano jest 250 przykładów, a testowano nie więcej 25 przykładów.

Wlaśnie, sam plik był utworzony tak, żeby dla jego uruchomienia trzeba było robić jak najmniej dodatkowych działań.

1. **Wyniki**

Dla oszacowania wydajności utworzonego algorytmu on był uruchomiony z różnymi zakresami danych.

*Tabela 2. Wyniki testowania algorytmu k-NN*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | metric | k | overall | ham | spam |
| 1 | e | 10 | 0.857 | 0.920 | 0.5 |
| 2 | c | 10 | 0.643 | 0.750 | 0.0 |
| 3 | e | 10 | 0.760 | 0.750 | 0.8 |
| 4 | c | 10 | 0.560 | 0.700 | 0.0 |
| 5 | e | 6 | 0.762 | 0.789 | 0.5 |
| 6 | c | 4 | 0.714 | 0.789 | 0.0 |

*Tabela 3. Zakresy, wykorzystane dla tworzenia modeli oraz testowania*

|  |  |  |
| --- | --- | --- |
|  | train | test |
| 1 | 1:250 | 5561:5574 |
| 2 | 420:740 | 51:75 |
| 3 | 4250:4600 | 1265:1285 |

W wyniku mamy taką sytuację, że dla danej implementacji i testowych zbiorów ilość sąsiadów nie wpłynęła na dokładność klasyfikacji, a korzystając z metryki kosinusowej, nie potrafiliśmy prawidłowo wyznaczyć przynajmniej jedno powiadomienie SPAM. Z danych testów nie można na pewnie powiedzieć, że klasyfikacja za pomocą metryki kosinusowej jest zła i niepoprawna, dlatego, że nam mogły prosto trafić takie dane. Na testach, prowadzonych dla sprawdzania działalności kolejnych metod w pliku knn.R było widać, że metryka kosinusowa działa i obliczenia odbywają się poprawnie. W teorii metryka kosinusowa musiałaby dawać lepszy wynik, niż Euklidysowa, ale z tabeli wyżej uzyskaliśmy odwrotne efekty.

Dla przyspieszania algorytmu były usunięte niektóre dodatkowe przetwarzania wierszy powiadomień oraz normy wektorów nie są obliczane w każdej iteracji predykcji, co dało możliwość korzystać z tego algorytmu.

Dla polepszenia wyników predykcji można jeszcze użyć steming tekstu, co jest dość przydatne i często korzystanie w obróbce tekstów dla tworzenia dobrego słownika modeli i bardziej skutecznego procesu klasyfikacji. Ta się odbywa dlatego, że po stemingu zostają tylko słowa semantyczne i oni są w formie bezokolicznika (dla czasowników) oraz tylko korzeń słów (dla rzeczowników).

Także można korzystać z innych wartości atrybutów niż binarne, dla obliczenia metryk. Wtedy wynik będzie bardziej odpowiedni.

kNN — prosty algorytm klasyfikacji, a więc dla zastosowania w prawdziwych problemach jest często nieskuteczny. Oprócz dokładności klasyfikacji problemem również jest jego wydajność obliczeniowa. Jeśli w zbiorze trenującym jest N obiektów, w testującym — M i wymiarowość przestrzeni — K, to ilość operacji klasyfikacji badanej próby można ocenić jako *O*(*K\*M\*N*). Niemniej jednak Algorytm k-NN jest dobrym przykładem do odkrywania Machine Learning.