Klasyfikacja danych algorytmem KNN

Jakub Adamczyk
Student kierunku Informatyka
Wydział Informatyki, Elektroniki i Telekomunikacji
Akademia Górniczo-Hutnicza w Krakowie

Algorytm K najbliższych sąsiadów (K nearest neighbours, KNN) jest prostym algorytmem statystycznego uczenia maszynowego z nadzorem służącym do klasyfikacji lub regresji danych. Stanowi on często pierwszy algorytm uczenia maszynowego przedstawiany studentom uczelni wyższych. Pomimo swojej prostoty daje on często zadowalające w praktyce wyniki. Artykuł ten przedstawia sposoby jego modyfikacji, które pozwalają osiągnąć za jego pomocą rezultaty mogące rywalizować z bardziej zaawansowanymi algorytmami uczenia maszynowego. Porównano także wyniki działania zaimplementowanego zmodyfikowanego algorytmu KNN z wcześniejszymi osiągnięciami w tej dziedzinie.

1 Wprowadzenie

Klasyfikacja danych jest ważnym zagadnieniem dziedzin analizy danych i nauczania maszynowego. Ma ona liczne zastosowania w różnych dziedzinach nauki, m. in. przy rozpoznawaniu obrazów [1], kategoryzacji tekstów [2] i innych, w których zachodzi potrzeba przypisania zmiennej objaśnianej jednej z określonych klas na podstawie zapewnionych danych. Sprawdza się dobrze w problemach, gdzie zależności między danymi są nieliniowe i trudne do określenia.

Algorytm KNN (zaproponowany w [3]) jest nieparametryczny [4], tzn. nie ma parametrów niezależnych od danych - optymalne wartości wszystkich parametrów zależą od konkretnego zbioru do klasyfikacji. Nie czyni on przy tym żadnych założeń co do rozkładów danych z klasyfikowanych zbiorów, co czyni go przydatnym przy badaniu zbiorów, co do których nie mamy żadnej wcześniejszej wiedzy (np. zakresów wartości).

Do zalet algorytmu należą jego prostota, brak fazy treningu (przechodzi od razu do klasyfikacji, co może stanowić jednak czasami wadę), możliwość łatwego zrównoleglenia obliczeń oraz dobra precyzja. Sprawiają one, że algorytm ten może być także w prostej implementacji używany jako część bardziej złożonych technik nauczania maszynowego.

Główne wady to koszty obliczeniowe i pamięciowe - nie należy on do najszybszych, a ponadto wymaga wykorzystania wszystkich danych ze zbioru treningowego. Zrównoleglanie obliczeń pozwala co prawda zmniejszyć te trudności (przyspiesza obliczenia oraz pozwala na wczytywanie danych kawałkami do pamięci), jednak ich nie eliminuje.

Algorytm ma 3 parametry, które trzeba zoptymalizować dla danego zbioru: liczbę k najbliższych sąsiadów, metrykę do obliczania odległości między sąsiadami oraz sposób ważenia sąsiadów, tak, aby ci bliżsi punktowi mieli większy wpływ na ostateczną klasyfikację.

Sposób działania algorytmu, możliwości dobrania optymalnych wartości parametrów oraz wcześniejsze wyniki badań na tym polu zostały omówione w pierwszej części artykułu. W dalszej przedstawiono sposób implementacji klasyfikatora metodą KNN mającego osiągnąć lepsze wyniki od dotychczasowych, używając tych technik. Na koniec przedstawiono wyniki, wnioski oraz omówiono perspektywy rozwoju projektu.

2 Działanie algorytmu

KNN w podstawowej wersji przyjmuje 4 argumenty: zbiór treningowy, zbiór do klasyfikacji, parametr klasyfikowany oraz liczbę sąsiadów k. Oblicza odległość euklidesową między klasyfikowanym punktem a wszystkimi pozostałymi, bierze k najbliższych sąsiadów i klasyfikuje punkt głosowaniem większościowym (przyporządkowuje punkt do tej klasy, w której jest najwięcej sąsiadów). Na rys. 1 przedstawiono przykład dla k=3 i k=5, gdzie większościowo przyporządkowuje się zielony punkt jako niebieski kwadrat lub czerwony trójkąt.

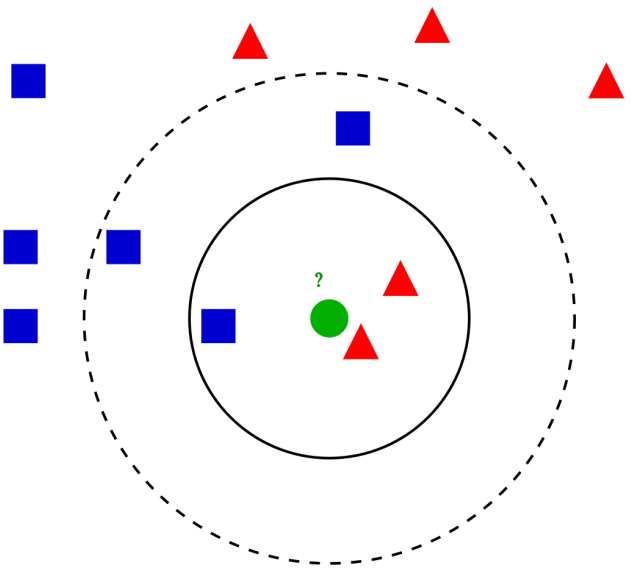


Fig. 1 Dla k=3 (linia ciągła) wynik klasyfikacji dla zielonego punktu z ważeniem stałym to czerwony trójkąt. Dla k=5 (linia przerywana) jest to już niebieski kwadrat.

Optymalny dobór liczby sąsiadów k to nadal badany problem. Niskie k pozwala wyraźnie zaznaczyć granice między klasami (uwzględnia tylko najbliższe, najważniejsze klasy), natomiast wysokie k filtruje szum (zmniejsza ryzyko wzięcia tylko albo w większości zaszumionych i błędnie bliskich punktów). Zaproponowano różne rozwiązania, takie jak wzięcie wszystkich sąsiadów z różnymi wagami [5] lub użycie algorytmu z podejściem *ensemble learning* [6]. Przyjęło się na podstawie badań empirycznych, że często dobrą wartością k jest pierwiastek kwadratowy z liczby obserwacji (liczby wierszy) zbioru treningowego, nie udało mi się jednak znaleźć oparcia teoretycznego dla tej praktyki. Problem znalezienia algorytmu optymalnego doboru k (innego niż wypróbowanie wielu wartości) pozostaje problemem otwartym.

Dobór metryki zależy w znacznej mierze od rodzaju danych, na których operujemy oraz od ich wymiaru (liczby kolumn). Metryka euklidesowa, tak jak wiele innych, cierpi na “klątwę wymiaru” - zjawisko, w którym w wysokowymiarowych przestrzeniach (przyjmuje się często, że > 30) wektory układają się prawie idealnie na kuli o promieniu jednostkowym. W przypadku algorytmu KNN, gdzie znalezienie sąsiadów o najmniejszych odległościach odgrywa kluczową kwestię może być to decydujące.

Powinno się zatem używać metryk odpornych na klątwę wymiaru, np. cosinusowej lub Mahalanobisa. Rodzaj danych wpływa na dobór metryki, bo np. do klasyfikacji tekstów przydatna może być odległość Hamminga (związana ze swojej natury z danymi tekstowymi). Bardzo wiele badań związanych z algorytmem KNN skupiało się na sprawdzaniu różnych metryk, w tym największe do tej pory badanie, które udało mi się znaleźć [7]. Metryki zostały omówione w dalszej części artykułu.

Ważenie sąsiadów opiera się na idei, że bliżsi sąsiedzi są ważniejsi, bo są bardziej podobni do naszego punktu. Podstawową wersję algorytmu można traktować jak stałe ważenie - wszystkie punkty o wadze 1. W [5] zaproponowano ważenie sąsiadów zgodnie z odwrotnością indeksu (metoda odwróconych indeksów), tzn. pierwszy (najbliższy) sąsiad ma wagę 1, drugi $\frac{1}{2}$, trzeci $\frac{1}{3}$ etc., czyli $\frac{1}{i}$. Pewną modyfikację tego sposobu stanowi zaproponowany w [8] schemat ważenia $\frac{1}{log_2(1+i)}$.

3 Metryki

W tej części omówione zostaną zaimplementowane metryki wraz z ich potencjalnym użyciem. Kompleksowe badanie można znaleźć w [7].

Canberra
Zaproponowana w [9] i zmodyfikowana w [10]. Stanowi pewną formę ważonej metryki Manhattan. Jest bardzo czuła na zmiany w okolicach zera, co widać szczególnie w wysokowymiarowych przestrzeniach. Nadaje się zatem do użycia przy danych rozrzuconych wokół jednego punktu centralnego.

$$D(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

Czebyszewa
Znana także jako metryka maksimum. Jest to specyficzna wersja metryki Minkowskiego, w której $p = \infty$.

$$D(x,y) = \max_i |x_i - y_i|$$

Chi-kwadrat
Znana także jako "Chi-kwadrat do kwadratu" dla odróżnienia od innych metryk typu Chi-kwadrat.

$$D(x,y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{|x_i + y_i|}$$

Cosinusowa
Wykorzystuje miarę podobieństwa cosinusowego, zdefiniowanego jako cosinus kąta między wektorami - im ten kąt jest mniejszy, tym bardziej podobne one są. Ważną cechą jest, że nie zwraca uwagi na długość wektorów. Jest używana szczególnie w klasyfikacji tekstów (np. przy wektorach liczb wystąpień poszczególnych słów). Aby uzyskać metrykę cosinusową odejmuje się podobieństwo cosinusowe od 1.

$$D(x,y) = 1 - \sphericalangle(x,y) = 1 - \frac{x \cdot y}{||x|| * ||y||} = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}}$$

Hassanata
Metryka ta została zaproponowana w [8] i specjalnie zaprojektowana na potrzeby algorytmu K najbliższych sąsiadów. Przyjmuje ona zawsze wartości z przedziału $[0, 1)$.

$$D(x,y) = \sum_{i=1}^{n} D(x_i, y_i)$$

$$D(x_i, y_i) = \begin{cases} 1 - \frac{1+\min(x_i,y_i)}{1+\max(x_i,y_i)} & \min(x_i, y_i) \geq 0 \\ 1 - \frac{1+\min(x_i,y_i)+|\min(x_i,y_i)|}{1+\max(x_i,y_i)+|\min(x_i,y_i)|} & \min(x_i, y_i) < 0 \end{cases}$$

Euklidesowa
Najprostsza z metryk, szczególnie narażona na klątwę wymiaru. Jest jednak przydatna w przestrzeniach o bardzo małej liczbie wymiarów, gdzie dobrze różnicuje odległości między punktami.

$$D(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Euklidesowa normalizowana
Wersja metryki euklidesowej znormalizowana podzieleniem przez długość pierwszego z wektorów. Zmniejsza to wpływ skrajnie różnych wielkości atrybutów na odległość.

$$D(x,y) = \frac{D_{euclidean}(x,y)}{||x||}$$

Euklidesowa standaryzowana
Wersja metryki euklidesowej, w której każdy element wektora jest pomnożony przez $\frac{1}{var(i\_col)^2}$ tj. odwrotność kwadratu wariancji dla danej kolumny ze zbioru treningowego. Zmniejsza to wpływ atrybutów o skrajnie dużej wariancji (np. błędnie zmierzonych) na odległość.

$$D(x,y) = \sum_{i=1}^{n} \frac{D_{euclidean}(x_i, y_i)}{var(i\_col)^2}$$

Hamminga
Jest to metryka wykorzystywana przede wszystkim w porównywaniu tekstów, w której sumuje się liczbę współrzędnych o różnych wartościach.

$$D(x,y) = \sum_{i=1}^{n} (x_i \neq y_i)$$

Mahalanobisa
Metryka zdefiniowana na potrzeby statystyki, konkretnie wyznaczania podobieństwa między nieznanym wektorem losowym a wektorem ze znanego zbioru. Została stworzona wiele lat przed powstaniem algorytmu K najbliższych sąsiadów (czy uczenia maszynowego w ogóle), ale znajduje szerokie zastosowanie w analizie danych ze względu na użycie macierzy kowariancji (a więc uwzględnienie wariancji między danymi). Poniższy zapis zakłada, że wektory x i y są pionowe, a znak $*$ to mnożenie macierzy.

$$D(x,y) = \sqrt{(x-y)^T * C^{-1} * (x-y)}$$

Manhattan
Znana także jako metryka taksówkowa, sprawdza się dobrze przy danych zbliżonych do rozkładu normalnego.

$$D(x,y) \sum_{i=1}^{n} |x_i - y_i|$$

Pearsona
Wykorzystuje wartość bezwzględną współczynnika korelacji Pearsona, odejmując go od 1. Reprezentuje stopień zależności liniowej wektorów (ich korelację).

$$D(x,y) = 1 - r_{xy}$$

Spearmana
Analogiczna do metryki Pearsona, ale wykorzystuje współczynnik korelacji Spearmana (wskazujący jak dobrze można opisać relację między wektorami funkcją monotoniczną).

$$D(x,y) = 1 - \rho_{xy}$$

4 Sposób testowania algorytmu

Do sprawdzania precyzji klasyfikacji algorytmów nauczania maszynowego wykorzystuje się często technikę walidacji skrośnej (sprawdzianu krzyżowego). Polega ona na wzięciu zbioru, w którym znamy wszystkie wartości (także atrybutu przewidywanego) oraz podziale go na zbiór uczący i zbiór walidacyjny. W walidacji n-krotnej zbiór walidacyjny to $\frac{1}{n}$ całego zbioru, a reszta to zbiór uczący. Używa się algorytmu, przyjmując kolejne części zbioru wielkości $\frac{1}{n}$ jako zbiory walidacyjne, dla każdego z nich znajduje się procent poprawnych wyników i uśrednia się wyniki. Pozwala to uzyskać średnią precyzję algorytmu dla danych parametów w danym zbiorze testowym.

Najpopularniejsze zbiory testowe pochodzą z UCI Machine Learning Repository. Wykorzystanie ich pozwala na łatwe porównanie wyników algorytmu z wynikami innych badań.

Testy zaimplementowanego algorytmu przeprowadzono na wybranych zbiorach z tego repozytorium, opisanych poniżej. Wykorzystano 5-krotną walidację skrośną.

5 Implementacja algorytmu

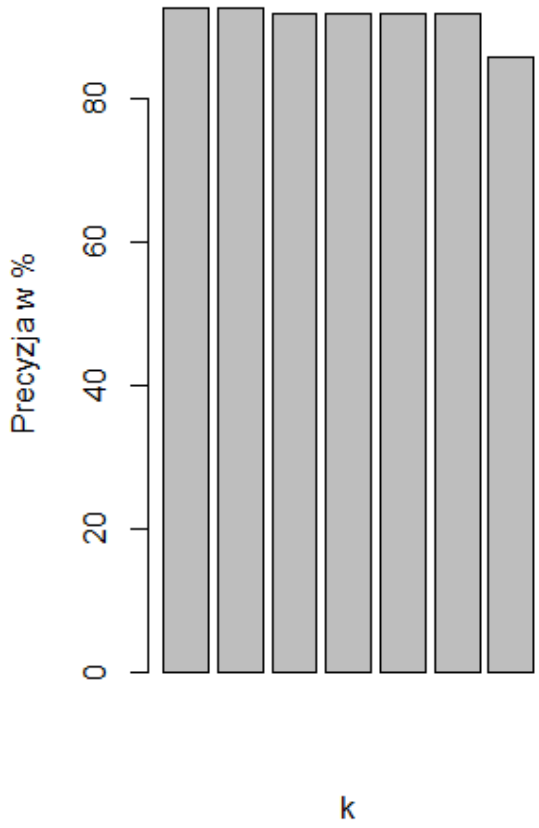
Opisany powyżej algorytm K najbliższych sąsiadów zaimplementowano w języku R. Domyślną wartością k jest pierwiastek z liczby obserwacji, z możliwością zmiany na liczbę lub wartość "all" (wszystkie obserwacje). Dostępne są wszystkie metryki opisane powyżej, domyślną jest metryka Hassanata (w [7] pokazano, że nadaje się dobrze na wartość domyślną). Możliwe sposoby ważenia to 3 opisane wcześniej.

6 Zbiory danych

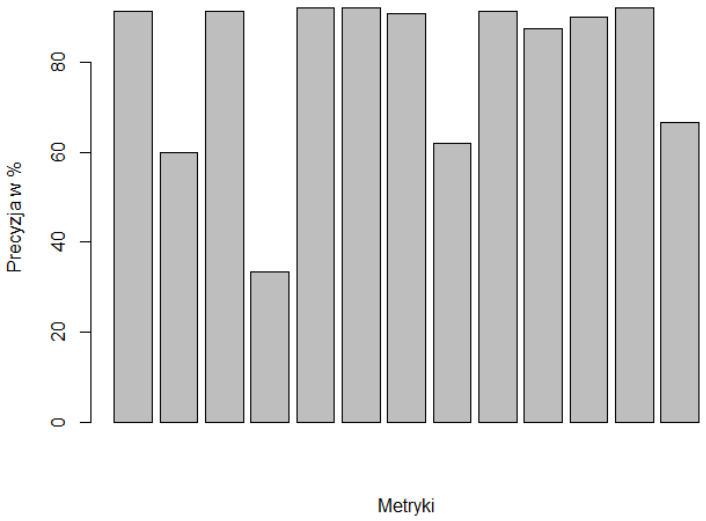
Nazwa	Obserwacje	Atrybuty (bez klasy)	Klasy	Opis
Banknote	1372	4	2	Dane zebrane ze zdjęć banknotów prawdziwych i sfałszowanych; sprawdza się autentyczność banknotu.
Cancer	569	30	2	Zbiór diagnostyki medycznej, dane badań kobiet po operacji usunięcia złośliwego raka piersi; sprawdza się, czy nowotwór powróci po operacji.
Car	1728	6	4	Dane o autach (cena, jakość wykonania, komfort korzystania); wycenia się opłacalność zakupu.
Glass	214	9	6	Skład chemiczny i pochodzenie szkła (np. szyba pojazdu, zastawa stołowa, szyba budynku) znalezionego na miejscach zbrodni w USA; sprawdza się pochodzenie szkła.
Haberman	306	2	2	Przeżywalność pacjentów po operacji nowotworu piersi; sprawdza się, czy pacjent przeżyje dłużej niż 5 lat.
Ionosphere	351	34	2	Dane z anten radiowych badających strukturę jonosfery; sprawdza się, czy sygnał przejdzie przez jonosferę.
Iris	150	4	3	Budowa płatków kwiatów irysów; klasyfikuje się gatunek irysa.
MONK2	601	6	2	Sztuczny zbiór stworzony do porównywania jakości klasyfikacji algorytmów nauczania maszynowego.
Parkinson	195	21	2	Zbiór biomedycznych pomiarów głosu pacjentów chorych na Parkinsona; sprawdza się, czy pacjent ma tę chorobę.
Wine	178	12	3	Analiza składu chemicznego win włoskich; klasyfikuje się odmianę wina.

7 Wyniki

Dla sprawdzenia wpływu wartości k i dobranej metryki na wyniki zmierzono precyzję (*accuracy*, standardowy sposób sprawdzania dokładności algorytmów nauczania maszynowego) i przedstawiono (w procentach) na poniższych diagramach. W tabeli poniżej zebrano najlepsze wyniki dla każdego ze zbiorów oraz odpowiadające im wartości parametrów.



Na powyższym wykresie dla stałej metryki euklidesowej i ważenia odwrotnymi indeksami przedstawiono precyzję dla różnych k dla zbioru irysów.



Na powyższym wykresie metryki to od lewej: Canberra, Czebyszewa, Chi-kwadrat, cosinusowa, Hassanata, euklidesowa, euklidesowa normalizowana, euklidesowa standardyzowana, Hamminga, Mahalanobisa, Manhattan, Pearsona i Spearmana. Wartość k była stała i wynosiła pierwiastek z liczby obserwacji dla zbioru irysów (ważenie odwrotnymi indeksami).

Zbiór danych	k	Metryka	Sposób ważenia	Poprawność	Wyniki poprzednich badań
Banknote	“all”	Mahalanobisa	Stałe wagi	100%	100%
Cancer	7	Canberra	Odwrócone indeksy	95,78%	95,62%
Car	Pierw. z liczby obserwacji	Chi-kwadrat	Stałe wagi	78,12%	55,37%
Glass	1	Manhattan	Stałe wagi	37,71%	51,15%
Haberman	“all”	Manhattan	Odwrócone indeksy	73,85%	38,87%
Ionosphere	1	Manhattan	Stałe wagi	89,14%	58,12%
Iris	5	Canberra	Stałe wagi	94,66%	95,85%
MONK2	1	Canberra	Stałe wagi	98,66%	78,10%
Parkinson	11	Mahalanobis	Stałe wagi	78,97%	99,97%
Wine	3	Hassanat	Stałe wagi	93,30%	74,08%

Jak widać, zgodnie z twierdzeniem “za wszystko trzeba płacić” (*no free lunch theorem*) nie ma jednej optymalnej kombinacji k, metryki i sposobu ważenia. Co ciekawe, algorytm najbliższego sąsiada (k=1), teoretycznie najbardziej narażony na szum, okazał się najlepszy w dwóch zestawach realnych danych pomiarowych, które, zdawałoby się, są najbardziej narażone na zaszumienie. Metryka Hassanata nie potwierdziła swoich doskonałych wyników z [8], ustępując zaskakująco często miejsca prostym metrykom Canberra i Manhattan. Sposób ważenia z [5] także nie okazał się aż tak dobry, za to zwykłe głosowanie większościowe było zwykle wystarczające.

W kilku przypadkach udało się uzyskać wyniki lepsze od dotychczasowych (np. z [7], [11]), np. dla zbiorów Car czy Ionosphere poprawność była zdecydowanie większa. Dla zbiorów danych, dla których uzyskano wyższe wyniki może to wynikać z niezaimplmentowania odpowiedniej ilości metryk, np. dla zbioru Parkinson w [7] uzyskano przedstawiony w tabeli wynik dla pewnej metryki, z której nie skorzystałem w swoim badaniu.

8 Wnioski

Zaimplementowany ulepszony algorytm K najbliższych sąsiadów uzyskał zdecydowanie zadowalające wyniki, w niektórych przypadkach uzyskując poprawność klasyfikacji znacznie przewyższającą porównywane badania. Pokazano także, jak ważne są: optymalny dobór liczby sąsiadów k, metryki oraz sposobu ważenia. Przy ich odpowiednim dobraniu można uzyskać wyniki bardzo dobre, porównywalne ze znacznie bardziej zaawansowanymi algorytmami.

9 Perspektywy rozwoju

Z pewnością można jeszcze poprawić uzyskane wyniki poprzez rozszerzenie implementacji algorytmu.

Optymalna wartość k może być wyznaczana osobnym algorytmem - chociaż jest to problem otwarty, to należałoby sprawdzić istniejące próby rozwiązania (np. heurystyczna aproksymacja wartości optymalnej).

Istnieją także algorytmy tworzenia metryki dla zbioru danych. Są one jednak z natury dość skomplikowane i dlatego nie zaimplementowano ich na potrzeby tego badania, jednak można by porównać wyniki takich rozwiązań z tymi dla standardowych metryk.

Istnieje wiele różnych sposobów ważenia sąsiadów poza tymi przetestowanymi - ich zaimplementowanie mogłoby zwiększyć precyzję.

Można ważyć także nie tylko obliczone odległości, ale same atrybuty - istnieją specjalne algorytmy ważenia atrybutów dla uwzględnienia ich właściwości (wariancji, skali wartości w porównaniu do pozostałych etc.).

Dzięki wykorzystaniu technik programowania funkcyjnego i wektoryzacji operacji (mechanizmu specyficznego m. in. dla użytego języka R) udało się przyspieszyć algorytm do zadowalających osiągów. Nie zrównoleglono jednak operacji - dla naprawdę dużych zbiorów danych z pewnością dałoby to znaczący zysk w szybkości obliczeń.

Literatura

[1] Wu, X., Kumar, V., Ross, Q., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z., Steinbach, M., Hand, D., and Steinberg, D., 2008, “Top 10 algorithms in data mining,” Knowledge and Information Systems, **14**(1), pp. 1–37.

[2] Manne S., S. F. S., Kotha S.K., 2012, “Text Categorization with K-Nearest Neighbor Approach.” Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012. Advances in Intelligent and Soft Computing, **132**.

[3] Silverman, B. W. and Jones, M. C., 1989, “E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951),” International Statistical Review / Revue Internationale de Statistique, **57**(3), pp. 233–238.

[4] Kataria, A. and Singh, M. D., 2013, “A Review of Data Classification Using K-Nearest Neighbour Algorithm,” .

[5] Jirina, M. and Marcel Jirina, J., “Classifiers Based on Inverted Distances,” New Fundamental Technologies in Data Mining, pp. 369–386.

[6] Hassanat, A. B., Abbadi, M. A., Altarawneh, G. A., and Alhasanat, A. A., 2014, “Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach,” CoRR, **abs/1409.0919**, 1409.0919

[7] Prasath, V. B. S., Alfeilat, H. A. A., Lasassmeh, O., and Hassanat, A. B. A., 2017, “Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier - A Review,” CoRR, **abs/1708.04321**, 1708.04321

[8] Alkasassbeh, M., Altarawneh, G., and Hassanat, A., 2015, “On enchancing the performance of nearest neighbour classifiers using Hassanat distance metric,” Canadian Journal of Pure and Applied Sciences, **9**.

[9] Lance, G. N. and Williams, W. T., 1966, “Computer Programs for Hierarchical Polythetic Classification (Śimilarity Analyses"),” Comput. J., **9**, pp. 60–64.

[10] Lance, G. N. and Williams, W. T., 1967, “Mixed-Data Classificatory Programs I - Agglomerative Systems.” Australian Computer Journal, **1**, pp. 15–20.

[11] Parvin, H., Alizadeh, H., and Minaei, B., 2008, “MKNN: Modified k-nearest neighbor,” Lecture Notes in Engineering and Computer Science, **2173**.