# Metody sztucznej inteligencji 2
## Projekt 1. — Algorytm $k$ najbliższych sąsiadów
## Raport

Bartłomiej Dach, Tymon Felski

27 maja 2018

Niniejszy dokument zawiera raport z implementacji algorytmu $k$ najbliższych sąsiadów ($k$-NN, ang. *$k$ nearest neighbors*) oraz analizę efektywności jego działania dla dostarczonych danych treningowych.

## 1 Algorytm $k$ najbliższych sąsiadów

Algorytm $k$ najbliższych sąsiadów jest jedną z wielu metod rozwiązywania **problemu klasyfikacji**, czyli predykcji wartości zmiennych jakościowych (zwanych również zmiennymi kategorycznymi lub dyskretnymi) na podstawie przykładowych **danych treningowych** [1, s. 9–10]. Klasycznym przykładem tego typu problemu jest klasyfikacja gatunków irysów z użyciem pomiarów rozmiarów działek kielicha oraz płatków kwiatu.

Klasyfikatory oparte na metodzie $k$ najbliższych sąsiadów operują na punktach w przestrzeni $n$-wymiarowej. Proces klasyfikacji danego punktu $x_0 \in \mathbb{R}^n$ odbywa się w następujący sposób:

1. Ze zbioru treningowego wybierane jest $k$ punktów znajdujących się najbliżej punktu $x_0$ [2, s. 261].
2. Jeżeli większość z wybranych $k$ punktów należy do jednej klasy, to punkt wejściowy jest przypisywany do tej samej klasy.
3. Potencjalne remisy w punkcie (2) są rozstrzygane losowo [1, s. 463–464].

Liczba punktów wybieranych ze zbioru treningowego (wartość $k$) stanowi parametr algorytmu. Jakość działania metody w dużym stopniu zależy od odpowiedniego doboru tego parametru dla danego zadania klasyfikacji [1, s. 468–470].

Do parametrów algorytmu można zaliczyć również metrykę używaną do obliczania odległości między punktami. W zaimplementowanym algorytmie uwzględnione zostały następujące metryki:

1. **odległość euklidesowa**:

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

2. **odległość taksówkowa (miejska, Manhattan)**:

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

3. **odległość Czebyszewa (maksimum)**:

$$d(x, y) = \max_{i=1,\ldots,n} |x_i - y_i|$$

4. **odległość Minkowskiego** z parametrem $p$, stanowiąca uogólnienie powyższych:

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

Ponieważ algorytm $k$-NN wymaga zapamiętania całości danych treningowych do swojego działania, zaliczany jest on do klasy klasyfikatorów **opartych na pamięci** (ang. *memory-based*) [1, s. 463].

# 2 Sposób analizy efektywności algorytmu

Efektywność działania $k$-NN była mierzona z użyciem dwóch dostarczonych dwuwymiarowych zbiorów danych oraz zastosowaniem metody kroswalidacji, zalecanej do doboru parametrów metody [1, s. 470].

## 2.1 Kroswalidacja

**$k$-krotna kroswalidacja (walidacja krzyżowa, sprawdzian krzyżowy)** to technika ewaluacji metod klasyfikacji z użyciem zebranych danych. W tej technice zbiór danych treningowych $X$ dzielony jest losowo na $k$ rozłącznych podzbiorów $X_1, \ldots, X_k$ podobnej liczności.

Po podziale danych wykonywane jest $k$ serii testowych. W $i$-tej serii testowany klasyfikator jako dane treningowe otrzymuje zbiór $X \setminus X_i$. Zadaniem klasyfikatora jest wyznaczenie klas dla punktów ze zbioru $X_i$. Następnie dla każdego punktu $x_j \in X_i$ następuje porównanie etykiety $y'_j$ wyznaczonej przez testowaną metodę z etykietą faktyczną $y_j$ [1, s. 241–243].

Główną metryką dokładności algorytmu przy tym porównaniu jest **proporcja błędnej klasyfikacji**, obliczana wzorem

$$e_i = \frac{1}{|X_i|} \cdot |\{y_j \neq y'_j : j = 1, 2, \ldots, |X_i|\}|$$

Końcowa proporcja błędnej klasyfikacji dla danej metody jest obliczane poprzez uśrednienie uzyskanych $k$ wyników.

## 2.2 Rozważane zbiory treningowe

Algorytm został przetestowany na dwuwymiarowych zbiorach `simple` i `three_gauss`, w których większość punktów zawiera się w kwadracie $[-1, 1] \times [-1, 1]$. Oba zbiory mają dość regularną charakterystykę, dzięki czemu możliwe jest również zastosowanie metod statystycznych do oceny efektywności klasyfikatora $k$-NN.

### 2.2.1 Zbiór `simple`

Zbiór `simple` składa się z punktów o rozkładzie zbliżonym do jednostajnego wzdłuż obu współrzędnych Punkty w tym zbiorze podzielone są na dwie klasy wzdłuż prostej o równaniu $x + y = 0$. Docelowo klasyfikator powinien więc jak najwierniej odwzorować podział punktów wzdłuż tej prostej.

### 2.2.2 Zbiór `three_gauss`

Zbiór `three_gauss` zawiera trzy klasy, częściowo nakładające się na siebie. Każda z trzech klas charakteryzuje się rozkładem podobnym do dwuwymiarowego rozkładu normalnego:

$$f(\mathbf{x}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)}{2\pi\sqrt{|\boldsymbol{\Sigma}|}}$$

Na podstawie zbioru zawierającego 30 000 punktów wyznaczone zostały przybliżone parametry rozkładów:

1. Dla klasy 1. mamy

$$\boldsymbol{\mu}_1 \approx \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_1 \approx \begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix},$$

2. Dla klasy 2. mamy

$$\boldsymbol{\mu}_2 \approx \begin{bmatrix} -0.6 \\ 0.2 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_2 \approx \begin{bmatrix} 0.04 & 0 \\ 0 & 0.0025 \end{bmatrix},$$

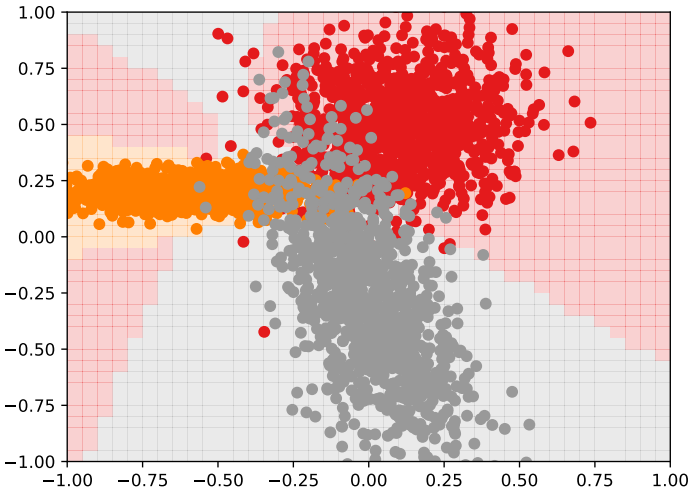
3. Dla klasy 3. mamy

$$\boldsymbol{\mu}_3 \approx \begin{bmatrix} 0 \\ -0.3 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_3 \approx \begin{bmatrix} 0.03 & -0.045 \\ -0.045 & 0.16 \end{bmatrix}.$$

Używając powyższych parametrów, można podzielić kostkę $[-1, 1] \times [-1, 1]$ na trzy obszary, obliczając w każdym jego punkcie **odległość Mahalanobisa** od poszczególnych rozkładów wzorem

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

i przypisując dany punkt do rozkładu, do którego jest mu najbliżej. Podział ten dla zbioru `three_gauss` ukazuje rysunek 1.




Rysunek 1: Podział obszaru $[-1, 1] \times [-1, 1]$ na klasy z użyciem odległości Mahalanobisa

# 3 Wyniki eksperymentów

Zgodnie z opisem zawartym w poprzednim rozdziale, jakość klasyfikacji przy pomocy algorytmu $k$-NN została przetestowana metodą kroswalidacji przy użyciu dwóch zbiorów treningowych i różnych wartości parametrów. Zbadano zbiory `simple` oraz `three_gauss` zawierające odpowiednio 1000 i 3000 elementów. Pod uwagę wzięto wartości parametru $k$ ze zbioru $\{1, 3, 5, 7, 9, 11, 13\}$ oraz pięć metryk:

- odległość taksówkową,
- odległość euklidesową,
- odległość Czebyszewa,
- odległość Minkowskiego z parametrem $p$ wynoszącym 1.5,
- odległość Minkowskiego z parametrem $p$ wynoszącym 3.

Zastosowano pięciokrotną kroswalidację, tzn. dane treningowe zostały losowo podzielone na 5 rozłącznych podzbiorów. We wszystkich próbach ziarno generatora liczb losowych zostało ustalone, aby podział był ten sam dla wszystkich permutacji wartości parametrów algorytmu.

## 3.1 Zbiór `simple`

### 3.1.1 Kroswalidacja

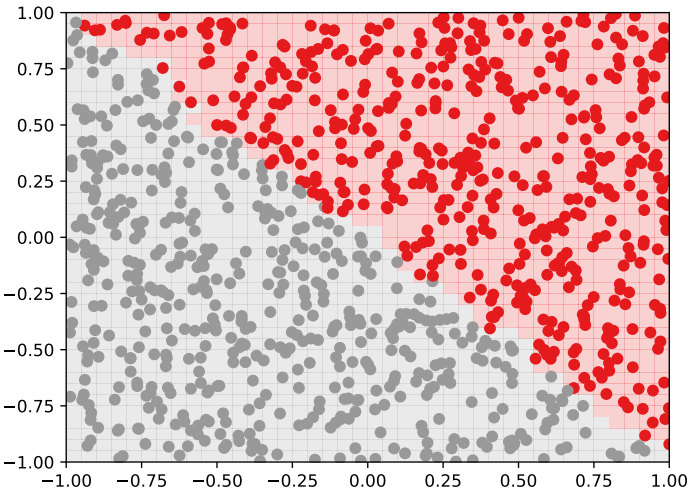
Wyniki testów jakości klasyfikacji przy pomocy pięciokrotnej kroswalidacji dla zbioru `simple` znajdują się w tabeli poniżej. Liczby w tabeli oznaczają średnią proporcję poprawnej klasyfikacji ($1 - \overline{e_i}$). Kolorem zielonym zaznaczono wynik najlepszy, a kolorem czerwonym — najgorszy. Poza najlepszym i najgorszym wynikiem, na pomarańczowo wyróżniono także dwa zestawy parametrów z wynikiem bardzo zbliżonym do najlepszego.

| | | **Wartość parametru $k$** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 3 | 5 | 7 | 9 | 11 | 13 |
| **Użyta metryka** | taksówkowa | 0.9850 | 0.9880 | 0.9870 | 0.9800 | 0.9820 | 0.9800 | 0.9810 |
| | euklidesowa | 0.9850 | 0.9860 | 0.9820 | 0.9820 | 0.9830 | 0.9830 | 0.9790 |
| | Czebyszewa | 0.9830 | 0.9860 | 0.9820 | 0.9810 | 0.9800 | 0.9810 | 0.9820 |
| | Minkowskiego (1.5) | 0.9850 | 0.9870 | 0.9840 | 0.9800 | 0.9830 | 0.9810 | 0.9800 |
| | Minkowskiego (3) | 0.9840 | 0.9850 | 0.9840 | 0.9820 | 0.9800 | 0.9810 | 0.9800 |

Tablica 1: Wyniki kroswalidacji algorytmu $k$-NN dla zbioru treningowego `simple`

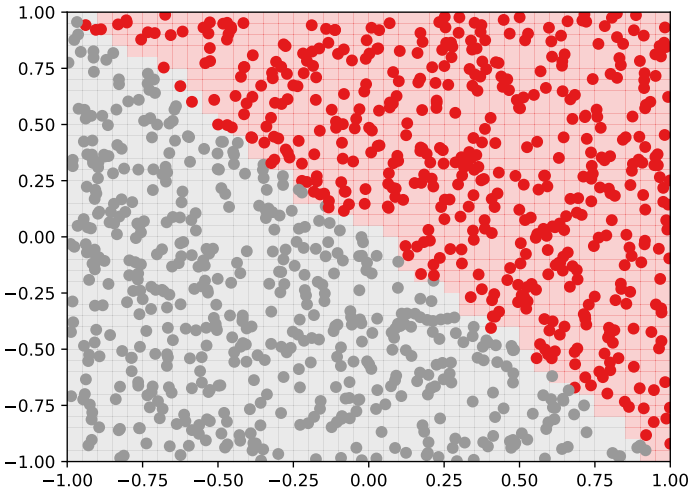
W przypadku powyższych wyników dla zbioru `simple` można zauważyć tendencję osiągania lepszej klasyfikacji dla mniejszych wartości parametrów $k$. Co prawda jest to bardzo mała różnica, ponieważ różnica pomiędzy najlepszym a najgorszym wynikiem nie wynosi nawet 1 punktu procentowego, jednak jest ona widoczna. Jest to zgodne z przewidywaniami – dla zbioru, w którym dane są liniowo separowalne, co skutkuje ostrym brzegiem pomiędzy klasami, rozważanie mniejszej liczby sąsiadów pozytywnie wpływa na ostateczny wynik, ponieważ nie rozmywa brzegu. Najlepiej wypadła tutaj klasyfikacja algorytmem $k$-NN dla $k = 3$ i z metryką taksówkową, jednak takie samo wywołanie dla $k = 5$ oraz wywołanie z $k = 3$ i metryką Minkowskiego przy $p = 1.5$ dały bardzo zbliżone wyniki. Równomierny rozkład punktów w obszarze niweluje różnice w wynikach spowodowane wyborem metryki.

Po ustaleniu najkorzystniejszych wartości parametrów algorytmu $k$-NN dla powyższego zbioru, zdecydowano się zbadać cały obszar $[-1, 1] \times [-1, 1]$ z gęstością próbkowania $\delta = 0.05$ i ustalić przynależność środkowych punktów każdego z kwadratów o boku $\delta$, zawartych w tym obszarze, do jednej z klas zbioru treningowego. Poniższy wykres jest graficzną reprezentacją otrzymanych wyników dla $k = 3$ oraz metryki będącej odległością taksówkową.




Rysunek 2: Klasyfikacja punktów z obszaru $[-1, 1] \times [-1, 1]$ dla $k = 3$ i odległości taksówkowej

Dla kontrastu, podobną analizę przeprowadzono dla parametrów, które zapewniły najgorszą klasyfikację, czyli dla $k = 13$ i odległości euklidesowej.




Rysunek 3: Klasyfikacja punktów z obszaru $[-1, 1] \times [-1, 1]$ dla $k = 13$ i odległości euklidesowej

Zamieszczone powyżej wykresy 2 oraz 3 potwierdzają opisane wcześniej przypuszczenia. Rozważanie aż 13 sąsiadów powoduje, że brzeg punktów ze zbioru nie jest poprawnie wyznaczony. Zjawisko rozmycia jest szczególnie widoczne w okolicy punktu $(-0.6, 0.7)$, w drugiej ćwiartce wykresu. Wokół punktów z klasy czerwonej, które znajdują się na brzegu, znajduje się więcej punktów z klasy szarej, co skutkuje zaklasyfikowaniem obszaru jako szary.

### 3.1.2 Analiza powstałych obszarów

Dla wyznaczonego w procesie kroswalidacji parametru $k = 3$ przetestowano wpływ wyboru metryki na odwzorowanie kształtu brzegów między obszarami wyznaczonymi przez klasyfikator. Po podziale obszaru $[-1, 1] \times [-1, 1]$ dokonano próbkowania klasyfikatora dla równoodległych punktów z interwałem $\delta = 0.05$ i porównania wynikowych etykiet z idealnymi. Wyniki tego eksperymentu znajdują się w tabeli 2.

| **Użyta metryka** | **Jakość odwzorowania brzegu** |
|---|---|
| taksówkowa | 0.9786 |
| Minkowskiego ($p = 1.5$) | 0.9798 |
| euklidesowa | 0.9810 |
| Minkowskiego ($p = 3$) | 0.9827 |
| Czebyszewa | 0.9833 |

Tablica 2: Wyniki porównania odwzorowania brzegów między zbiorami przez algorytm $k$-NN z odwzorowaniem dokładnym dla zbioru `simple`

Różnice między poszczególnymi metrykami nie są znaczne (rozstęp 0.5 punkta procentowego różnicy), lecz najlepszym wynikiem wykazała się metryka Czebyszewa.

## 3.2 Zbiór `three_gauss`

### 3.2.1 Kroswalidacja

Pięciokrotną kroswalidację badanego klasyfikatora przeprowadzono także dla zbioru `three_gauss`. Jej wyniki znajdują się w tabeli poniżej. Komórki tabeli wyróżnione kolorami: zielonym, czerwonym i pomarańczowym mają tutaj takie same znaczenia.

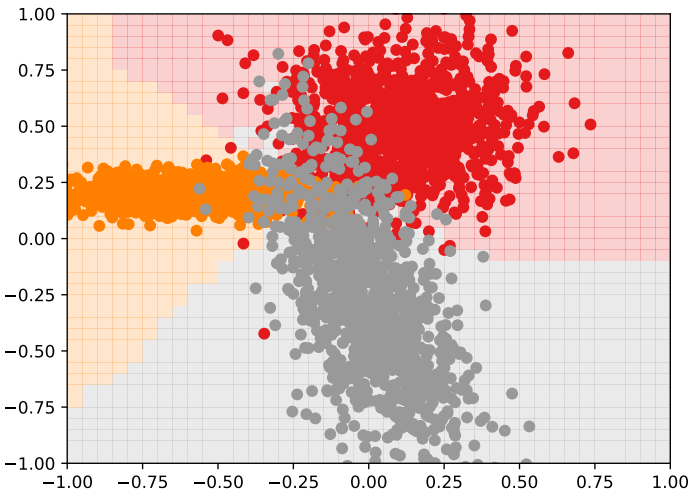
| | | **Wartość parametru $k$** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 3 | 5 | 7 | 9 | 11 | 13 |
| **Użyta metryka** | taksówkowa | 0.9053 | 0.9233 | 0.9257 | 0.9283 | 0.9333 | 0.9280 | 0.9290 |
| | euklidesowa | 0.9073 | 0.9220 | 0.9230 | 0.9303 | 0.9290 | 0.9290 | 0.9267 |
| | Czebyszewa | 0.9113 | 0.9213 | 0.9217 | 0.9253 | 0.9277 | 0.9307 | 0.9300 |
| | Minkowskiego (1.5) | 0.9070 | 0.9237 | 0.9253 | 0.9290 | 0.9307 | 0.9280 | 0.9263 |
| | Minkowskiego (3) | 0.9090 | 0.9210 | 0.9230 | 0.9300 | 0.9297 | 0.9297 | 0.9297 |

Tablica 3: Wyniki kroswalidacji algorytmu $k$-NN dla zbioru treningowego `three_gauss`

Jak już zostało to wcześniej opisane, klasy punktów w zbiorze `three_gauss` częściowo na siebie nachodzą; ponadto mamy w nim do czynienia z tzw. obserwacjami odstającymi, czyli punktami z klasy, które znajdują się daleko od pozostałych. Wybór większego

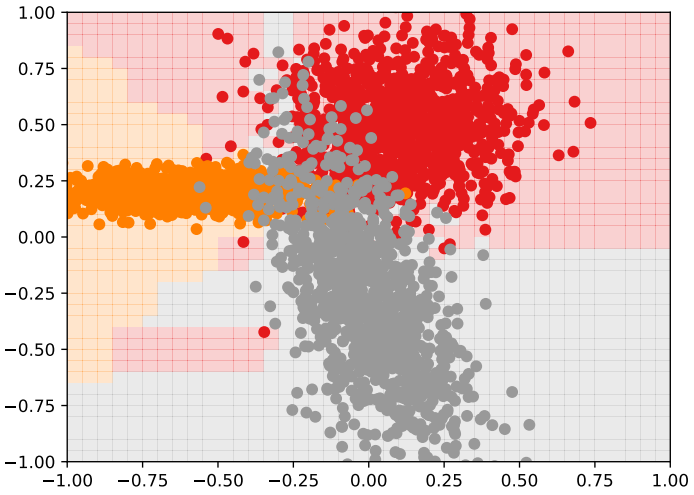
parametru $k$ pozwala na zniwelowanie wpływu obserwacji odstających na działanie algorytmu, co demonstrują dane doświadczalne. Lepsze wyniki klasyfikacji można dostrzec w prawej części tabeli. Najlepiej wypadło tutaj wywołanie dla $k = 9$ z metryką taksówkową, jednak niewiele gorszy wynik można osiągnąć dla takiej samej wartości $k$ z metryką Minkowskiego dla $p = 9$ lub dla $k = 11$ i odległości Czebyszewa. Różnica pomiędzy najgorszym a najlepszym wynikiem wynosi w przypadku zbioru `three_gauss` już aż prawie 3 punkty procentowe.

Dla wyróżnionych parametrów algorytmu $k$-NN podczas klasyfikacji zbioru `three_gauss` przeprowadzono analogiczną analizę, jak w przypadku zbioru `simple`. Punkty z obszaru $[-1, 1] \times [-1, 1]$ z gęstością próbkowania $\delta = 0.05$ zostały sklasyfikowane, w celu wizualizacji przyporządkowania. Poniższy wykres zawiera wyniki uzyskane dla wywołania algorytmu $k$-NN dla $k = 9$ i odległości taksówkowej.




Rysunek 4: Klasyfikacja punktów z obszaru $[-1, 1] \times [-1, 1]$ dla $k = 9$ i odległości taksówkowej

Podobnie jak w poprzednim podrozdziale, dla zbioru `three_gauss` również zwizualizowano wynik działania algorytmu $k$-NN dla najgorszego zestawu parametrów, czyli $k = 1$ i odległości taksówkowej. Przedstawia to poniższy wykres.




Rysunek 5: Klasyfikacja punktów z obszaru $[-1, 1] \times [-1, 1]$ dla $k = 1$ i odległości taksówkowej

Zamieszczone powyżej wykresy 4 oraz 5 pokazują znaczenie poprawnego dobrania wartości parametru $k$ dla zbioru o takiej charakterystyce. Rozważanie większej liczby sąsiadów (w tym przypadku dziewięciu) powoduje, że punkty z danej klasy znajdujące się daleko od pozostałych nie wpływają na klasyfikację w swojej okolicy, jeżeli zdecydowana większość sąsiadów należy do innej klasy. Dobrym przykładem tego zjawiska jest punkt z klasy czerwonej znajdujący się w okolicy punktu $(-0.3, -0.45)$. Dla $k = 9$ został on pominięty i znajduje się w obszarze należącym do klasy szarej. W przypadku klasyfikacji z $k = 1$ punkt ten znacznie wpłynął na ostateczny wynik i spowodował przyporządkowanie sporego obszaru trzeciej ćwiartki wykresu do swojej klasy.

### 3.2.2 Analiza powstałych obszarów

Analogicznie jak w przypadku zbioru `simple`, kształt obszarów wyznaczonych przez $k$-NN porównano z wzorcowym (w tym przypadku jest to podział wyznaczony przez odległość Mahalanobisa, ukazany na rys. 1). Wyniki tego eksperymentu znajdują się w tabeli 4.

| **Użyta metryka** | **Jakość odwzorowania brzegu** |
|---|---|
| taksówkowa | 0.7829 |
| Minkowskiego ($p = 1.5$) | 0.7948 |
| euklidesowa | 0.7995 |
| Minkowskiego ($p = 3$) | 0.8061 |
| Czebyszewa | 0.8126 |

Tablica 4: Wyniki porównania odwzorowania brzegów między zbiorami przez algorytm $k$-NN z odwzorowaniem wyznaczonym przez metrykę Mahalanobisa dla zbioru `three_gauss`

Jakość odwzorowania w tym przypadku znacząco spada, ponieważ podział przedstawiony na wykresie 1 nie jest intuicyjny z geometrycznego punktu widzenia. Algorytm $k$-NN w dużej mierze opiera się na miarach w celu wyznaczenia sąsiedztwa, więc szczególnie w drugiej ćwiartce układu współrzędnych (blisko klasy 2, oznaczonej na rysunkach kolorem pomarańczowym) rozbieżności są duże. Natomiast i w tym przypadku zauważalny jest wzrost jakości odwzorowania wraz z wzrostem parametru $p$ w normie Minkowskiego.

# 4 Wnioski

Algorytm $k$ najbliższych sąsiadów jest jednym z najprostszych klasyfikatorów pod względem zasady działania. Testy z dostarczonymi zbiorami pokazały jednak, że jest on w stanie osiągać bardzo dobre wyniki dla danych o odpowiedniej regularności. Dużym minusem algorytmu jest jednak konieczność zapamiętania całego zbioru treningowego, co jest w przypadku dużych ilości danych źródłem problemów zarówno pamięciowych, jak i wydajnościowych.

Z przeprowadzonych eksperymentów wynika, że odpowiedni dobór parametrów klasyfikatora stanowi kluczową rolę w jego dokładnym działaniu. Przy doborze parametru $k$ należy brać pod uwagę czynniki takie, jak separowalność klas, zaszumienie danych oraz występowanie obserwacji odstających, zaś wybór metryki ma znaczący wpływ na kształty obszarów, na które klasyfikator dzieli przestrzeń $\mathbb{R}^n$.

## Literatura

[1] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Nowy Jork: Springer-Verlag, 2009. [Online] Dostępne: `https://web.stanford.edu/~hastie/ElemStatLearn/`. [Dostęp 26 lutego 2018]

[2] V.N. Vapnik, *Statistical learning theory*. Nowy Jork: John Wiley and Sons, 1998.