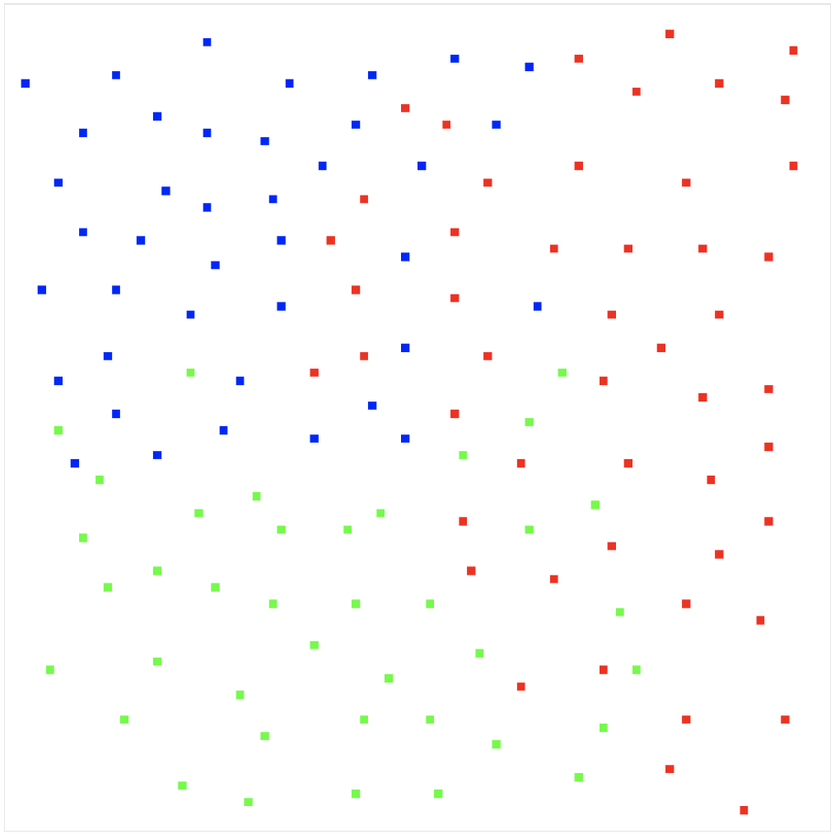
Metric Learning

Bartłomiej Szałach

30 października 2017

1 Wstęp

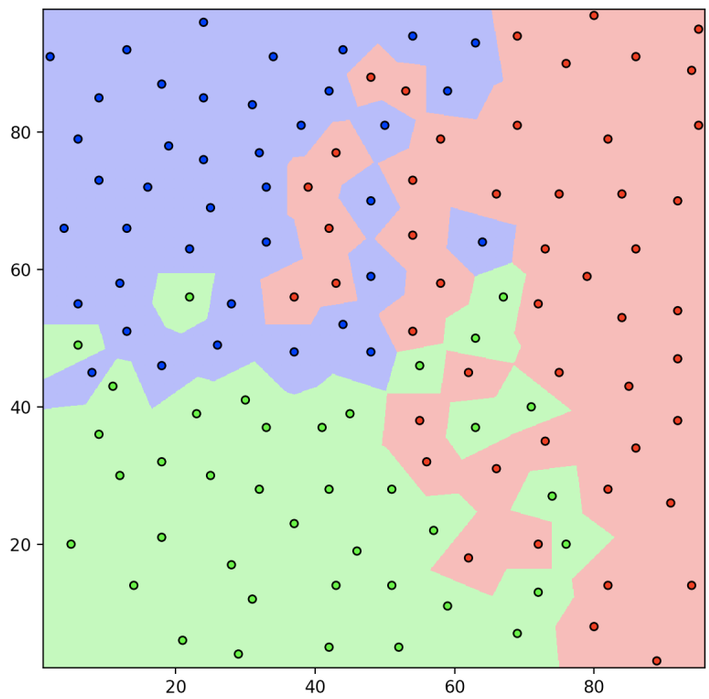
Celem zadania jest obserwacja jak zmiana wykorzystywanej przez klasyfikator k-NN metryki wpływa na kształt granicy decyzyjnej, a więc w efekcie na jego skuteczność. W programie graficznym został przygotowany przykładowy zbiór danych z "obserwacjami" należącymi do 3 różnych klas. Obrazek został zapisany jako 256-kolorowa mapa bitowa. Do zadania użyto języka Python z bibliotekami: *sklearn, numpy, matplotlib, PIL (pillow)*



Rysunek 1: Wygenerowane dane

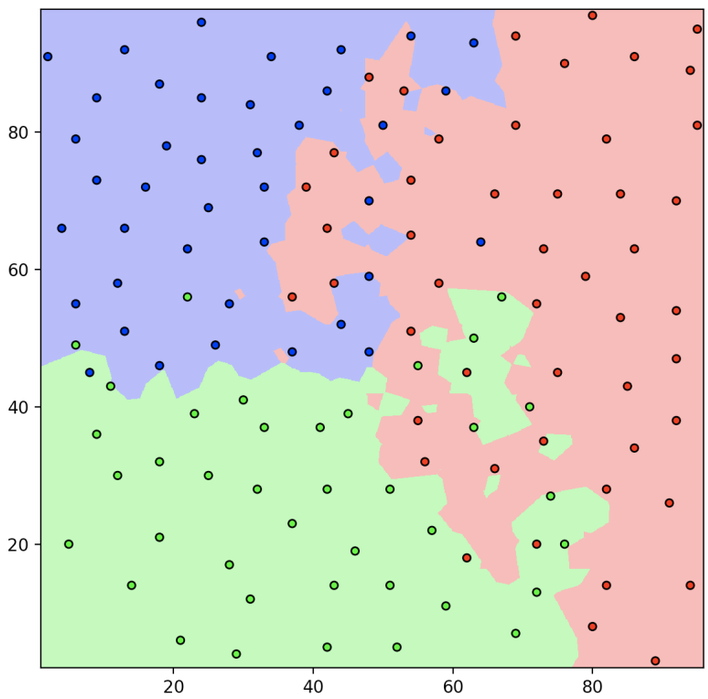
2 Algorytm k-NN z metryką Euklidesa i k = 1

Granice podziału są poszarpane dla k = 1. Jest tak dlatego, że każdy z punktów tworzy swoją wysepkę, w której jest on najbliższy dla dowolnego punktu znajdującego się w niej.



Rysunek 2: k = 1, metryka Euklidesa

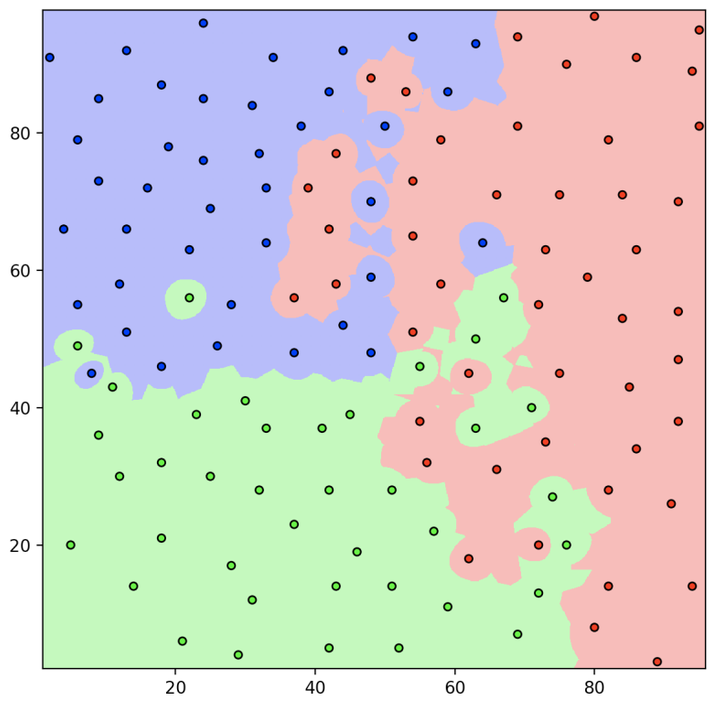
3 Algorytm k-NN z metryką Euklidesa i k = 3



Rysunek 3: k = 3, metryka Euklidesa

Wartość k = 3 jest zazwyczaj wartością wystarczającą do dobrej klasyfikacji. Na obrazku widać istotną poprawę granic obszarów decyzyjnych. Ze względu na (celowo przygotowane) nieregularne dane, możemy zaobserwować kilka ciekawych zjawisk. Po lewej stronie dwa zielone punkty znajdują się na niebieskim tle, gdyż w dowolnym ich otoczeniu przeważać będą niebiescy sąsiedzi (1 zielony i 2 niebieskie). Podobna sytuacji występuje z punktami niebieskimi na czerwonym tle. Interesującym zjawiskiem są również małe zielone wyspy na czewonym tle. Wraz ze wzrostem k znikają one na rzecz

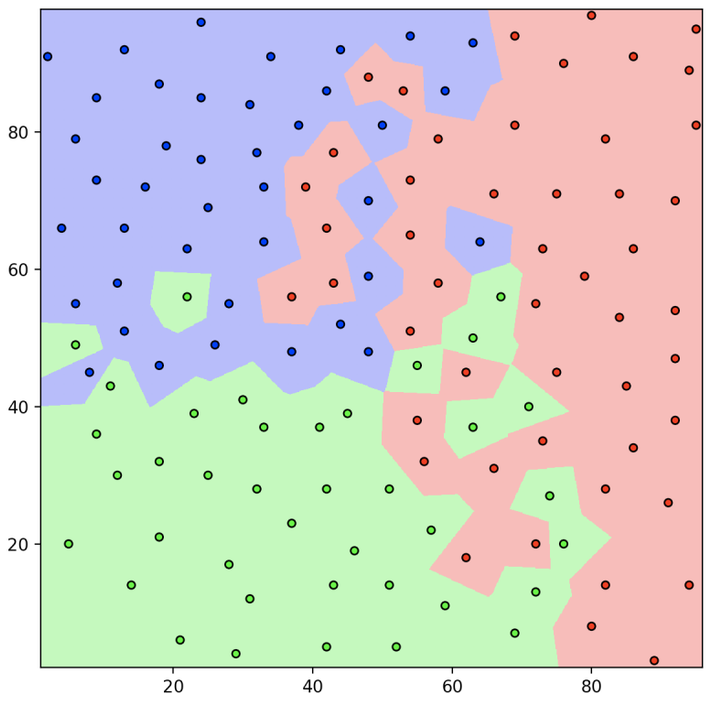
klasyfikacji do klasy czerwonej. Dla k = 10 praktycznie już nie występują.
Gdy zastosujemy wagi punktów (tzn. bliższe punkty mają większe znaczenie niż te dalsze) to uzyskujemy ciekawą klasyfikację.



Rysunek 4: k = 3, metryka Euklidesa, waga dystansu

4 Algorytm k-NN z metryką Mahalanobisa

Metryka Mahalanobisa różnicuje wkład poszczególnych składowych w odległość pomiędzy punktami. Do jej uzyskania używamy macierzy kowariancji, która określa charakter rozkładu punktów. Niestety, z nieznanych autorowi przyczyn, granice klas klasyfikacji z metryką Mahalanobisa (k = 1) wygląda niemalże identycznie jak te uzyskane przy użyciu metryki Euklidesa z k = 1.



Rysunek 5: k = 3, metryka Mahalanobisa

5 Skuteczność klasyfikatorów

Poprzednie wykresy przedstawiały klasyfikację w oparciu o cały zbiór danych. Podczas oceny dokładności podzielono zbiór na dwa podzbiory: treningowy oraz testowy. Zbiór treningowy stanowi ok. 75% danych, natomiast testowy ok. 25%.
Ponieważ punkty do zbiorów są wybierane pseudolosowo, to na potrzeby oceny skuteczności klsyfikacji zmieniono skrypt, aby zamiast rysować granice klasyfikacji, zwracał średnią z 10 wyliczonych skuteczności. Skuteczność klasyfikatorów wyniosła:

- Metryka Euklidesa k = 1: 69.41%
- Metryka Euklidesa k = 3: 70.65%
- Metryka Euklidesa k = 20: 76.57%
- Metryka Euklidesa k = 3 z wagami: 69.18%
- Metryka Mahalanobisa k = 1: 70.79%

6 Wnioski i podsumowanie

- Dla klasyfikatora kNN wybór odpowiedniej metryki ma duże znaczenie i wpływ na jego skuteczność
- Zwiększając wartość k linie granic decyzyjnych ulegają wygładzeniu przestając być poszarpane. Zwiększa się też skuteczność
- Nie istnieje jedna najlepsza metryka, ani wartość k, muszą one być dobrane w oparciu o charakterystykę problemu oraz własne obycie w obszarze z którego pochodzą dane
- Ułożenie punktów ma kluczowe znaczenie na skuteczność. Przy mniej pomieszanych zbiorach skuteczność jest wyższa

- Klasyfikacja z metryką Mehalanobisa z k = 1 wygląda bardzo podobnie do klasyfikacji z metryką Euklidesa z k = 1. Skormnie uważam, że jest to jednak bardziej prawdopodobnie błąd autora lub zły dobór danych, niż rzeczywiste zachowanie. Jednakże, pomiar skuteczności klasyfikacji z metryką Mahalanobisa trwał kilkadziesiąt razy dłużej niż w poprzednich przypadkach, co pozwala przypuszczać, że jest to poprawna klasyfikacja, pomimo, że nie wygladała na taką.
- Różnice w skuteczności są mało widoczne, co niepokoi autora tej pracy. Dopiero po zwiększeniu wartości k do 20, skuteczność znacznie wzrosła.