UMAP

Uniform Manifold Approximation and Projection for Dimension Reduction

Mateusz Przyborowski

17 kwietnia 2020 r.

Główna motywacja – redukcja wymiaru

Główna motywacja – redukcja wymiaru

Metody redukcji wymiaru możemy podzielić na:

Główna motywacja – redukcja wymiaru

Metody redukcji wymiaru możemy podzielić na:

1. feature selection
   - algorytmy przeszukiwania (np. wyżarzania, zachłanne)
   - kryteria stopu (np. AIC, BIC)

Główna motywacja – redukcja wymiaru

Metody redukcji wymiaru możemy podzielić na:

1. feature selection
   - algorytmy przeszukiwania (np. wyżarzania, zachłanne)
   - kryteria stopu (np. AIC, BIC)
2. feature extraction
   - metody liniowe (np. PCA, LDA)
   - metody nieliniowe (np. t-SNE, UMAP)

Schemat działania t-SNE:

Schemat działania t-SNE:

1. liczenie gaussowskich rozkładów p-stwa pomiędzy przykładami,

Schemat działania t-SNE:

1. liczenie gaussowskich rozkładów p-stwa pomiędzy przykładami,
2. liczenie rozkładów t-Studenta dla embeddingów,

Schemat działania t-SNE:

1. liczenie gaussowskich rozkładów p-stwa pomiędzy przykładami,
2. liczenie rozkładów t-Studenta dla embeddingów,
3. minimalizacja dywergencji Kullbacka-Leiblera pomiędzy rozkładami za pomocą algorytmu gradientowej optymalizacji

Schemat działania UMAP:

Schemat działania UMAP:

1. liczenie topologicznej reprezentacji danych przy nieznanej rozmaitości, na której są one zadane,

Schemat działania UMAP:

1. liczenie topologicznej reprezentacji danych przy nieznanej rozmaitości, na której są one zadane,
2. liczenie topologicznej reprezentacji embeddingów na znanej (niskowymiarowej) rozmaitości,

Schemat działania UMAP:

1. liczenie topologicznej reprezentacji danych przy nieznanej rozmaitości, na której są one zadane,
2. liczenie topologicznej reprezentacji embeddingów na znanej (niskowymiarowej) rozmaitości,
3. minimalizacja entropii krzyżowej pomiędzy reprezentacjami za pomocą algorytmu gradientowej optymalizacji

Podstawowe pojęcia

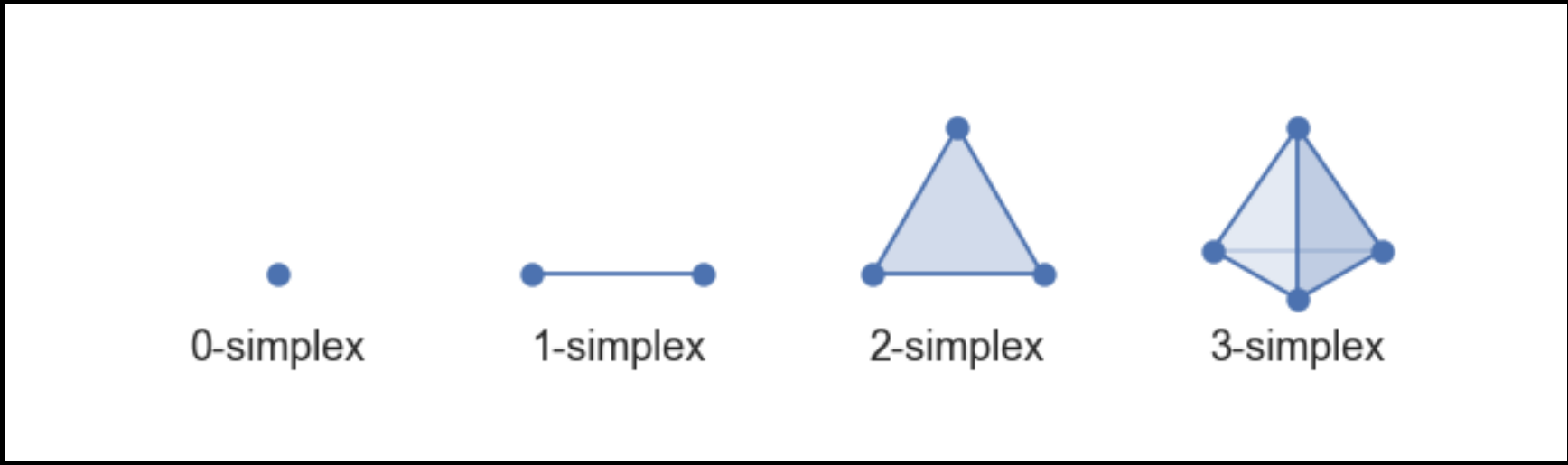
Sympleks to n-wymiarowy wielościan, który jest otoczką wypukłą swoich n+1 wierzchołków.

Sympleks to n-wymiarowy wielościan, który jest otoczką wypukłą swoich n+1 wierzchołków.

Intuicyjnie jest to uogólnienie takich obiektów jak prosta lub trójkąt na więcej wymiarów.

Sympleks to n-wymiarowy wielościan, który jest otoczką wypukłą swoich n+1 wierzchołków.

Intuicyjnie jest to uogólnienie takich obiektów jak prosta lub trójkąt na więcej wymiarów.



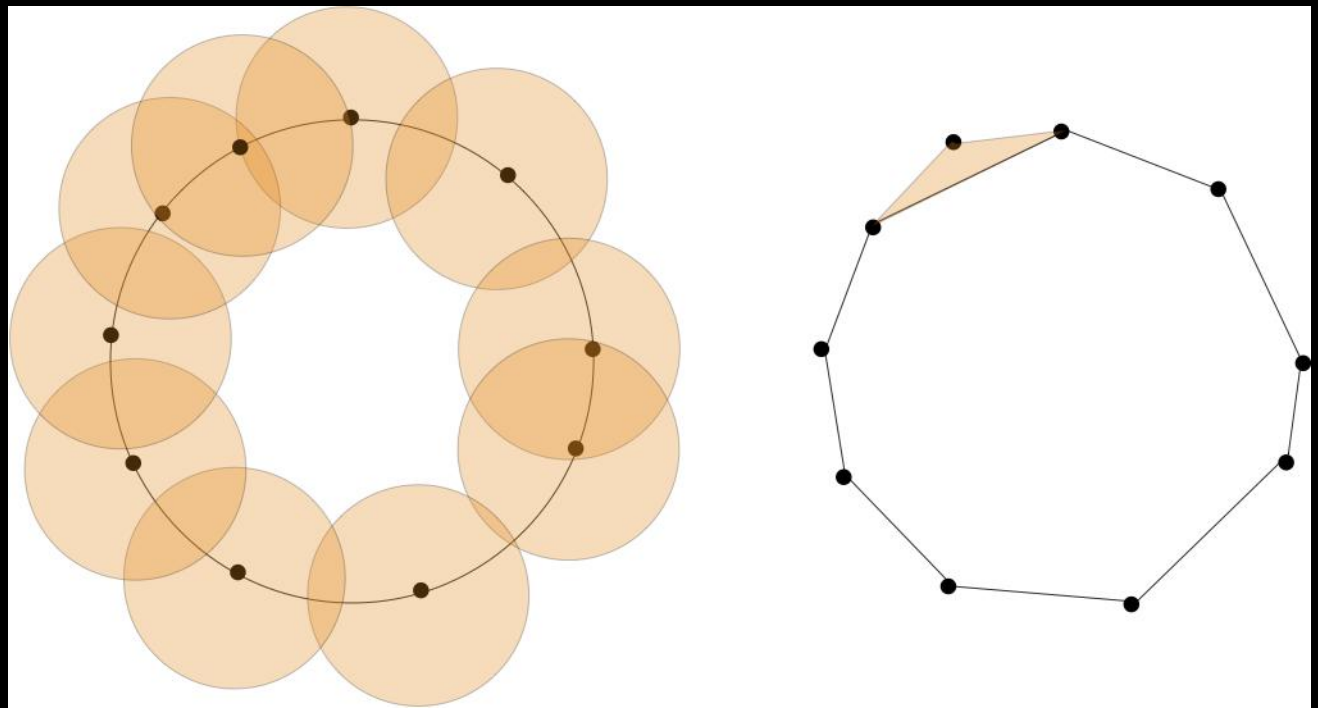
Sklejanie sympleksów wzdłuż ich (hiper)krawędzi daje nam obiekt o nazwie **kompleks symplicjalny**.

Sklejanie sympleksów wzdłuż ich (hiper)krawędzi daje nam obiekt o nazwie **kompleks symplicjalny**.

Mając zadane otwarte pokrycie przestrzeni topologicznej, poprzez konstrukcję kompleksu Cecha, uzyskujemy kompleks symplicjalny tej przestrzeni.

Sklejanie sympleksów wzdłuż ich (hiper)krawędzi daje nam obiekt o nazwie **kompleks symplicjalny**.

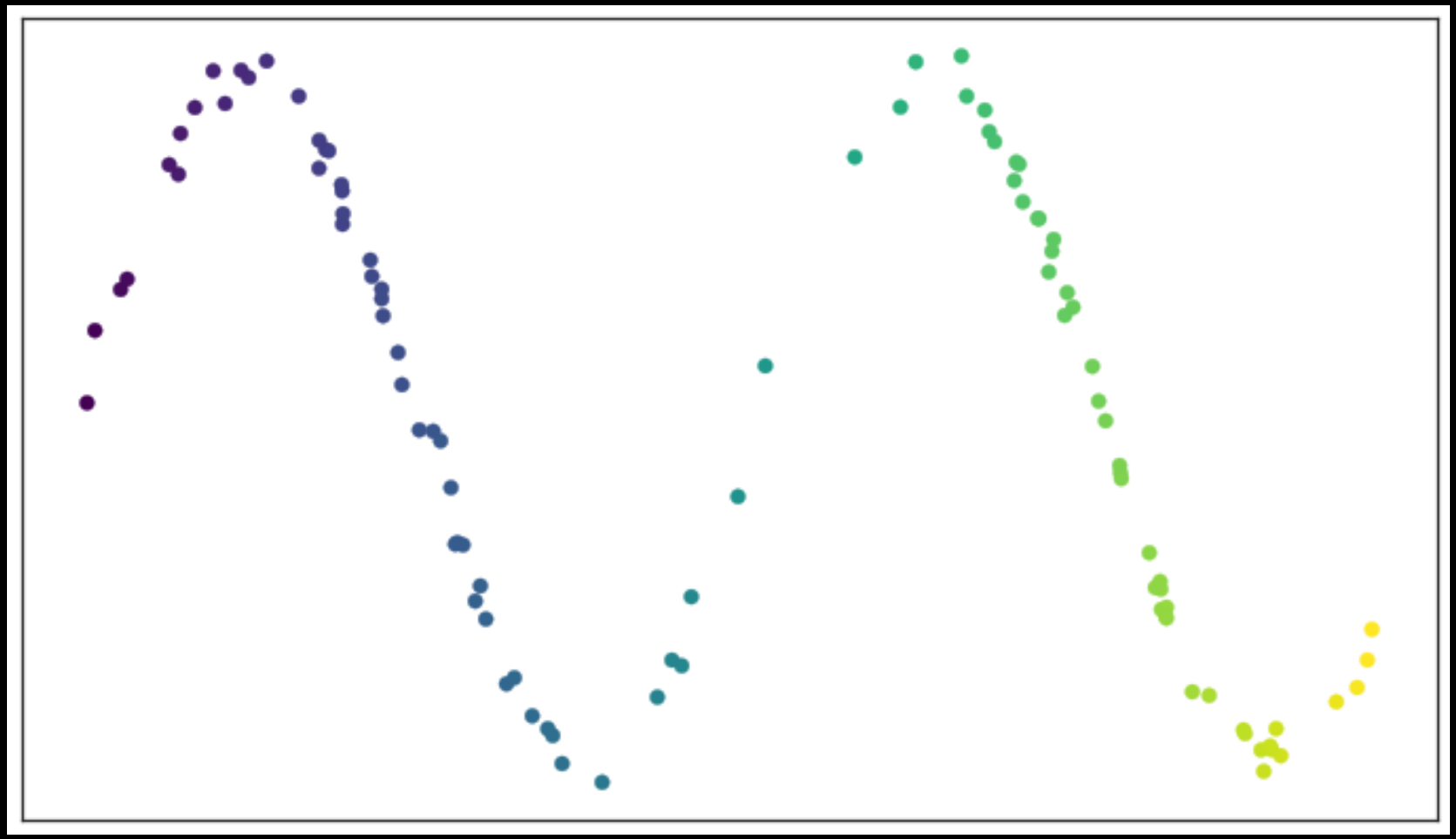
Mając zadane otwarte pokrycie przestrzeni topologicznej, poprzez konstrukcję kompleksu Cecha, uzyskujemy kompleks symplicjalny tej przestrzeni.

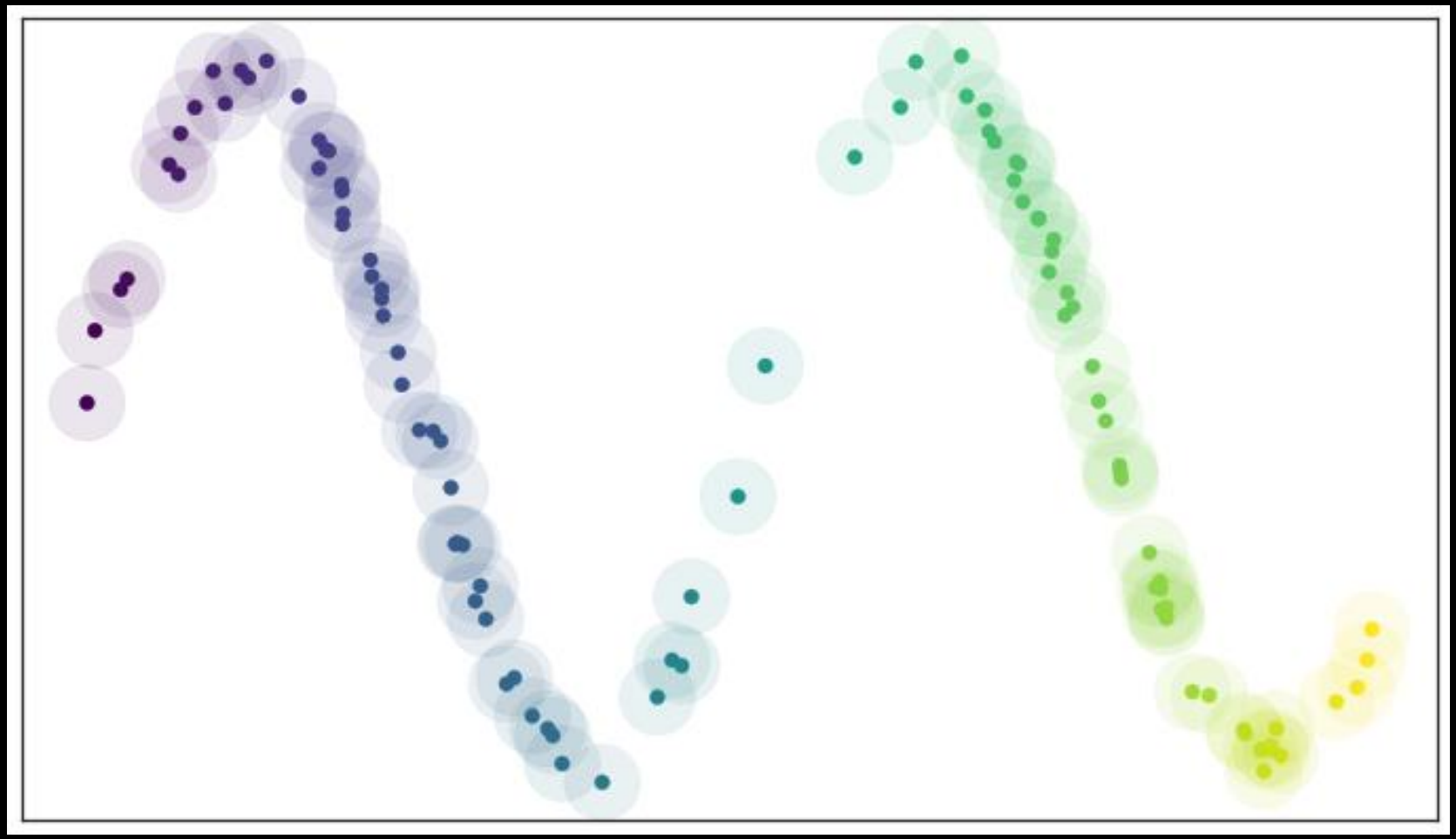


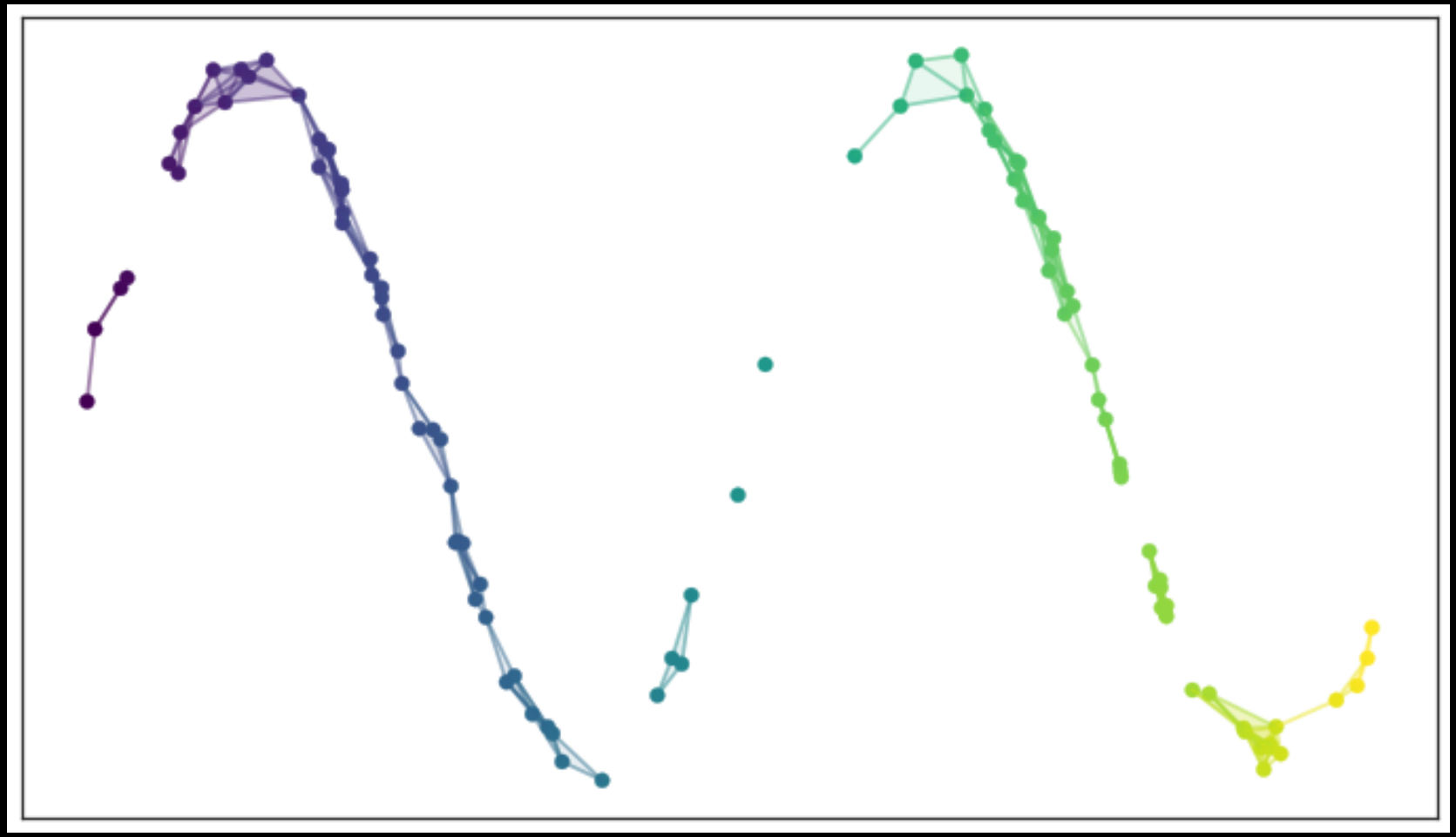
Z twierdzenia o nerwie wiemy, że tak skonstruowany kompleks symplicjalny zawiera istotne informacje o własnościach wyjściowej przestrzeni topologicznej, np. ma ten sam typ homotopii.

Z twierdzenia o nerwie wiemy, że tak skonstruowany kompleks symplicjalny zawiera istotne informacje o własnościach wyjściowej przestrzeni topologicznej, np. ma ten sam typ homotopii.

Jeśli założymy, że nasze dane leżą w pewnej przestrzeni metrycznej, to możemy aproksymować pokrycie poprzez kule jednostkowe wokół poszczególnych punktów.







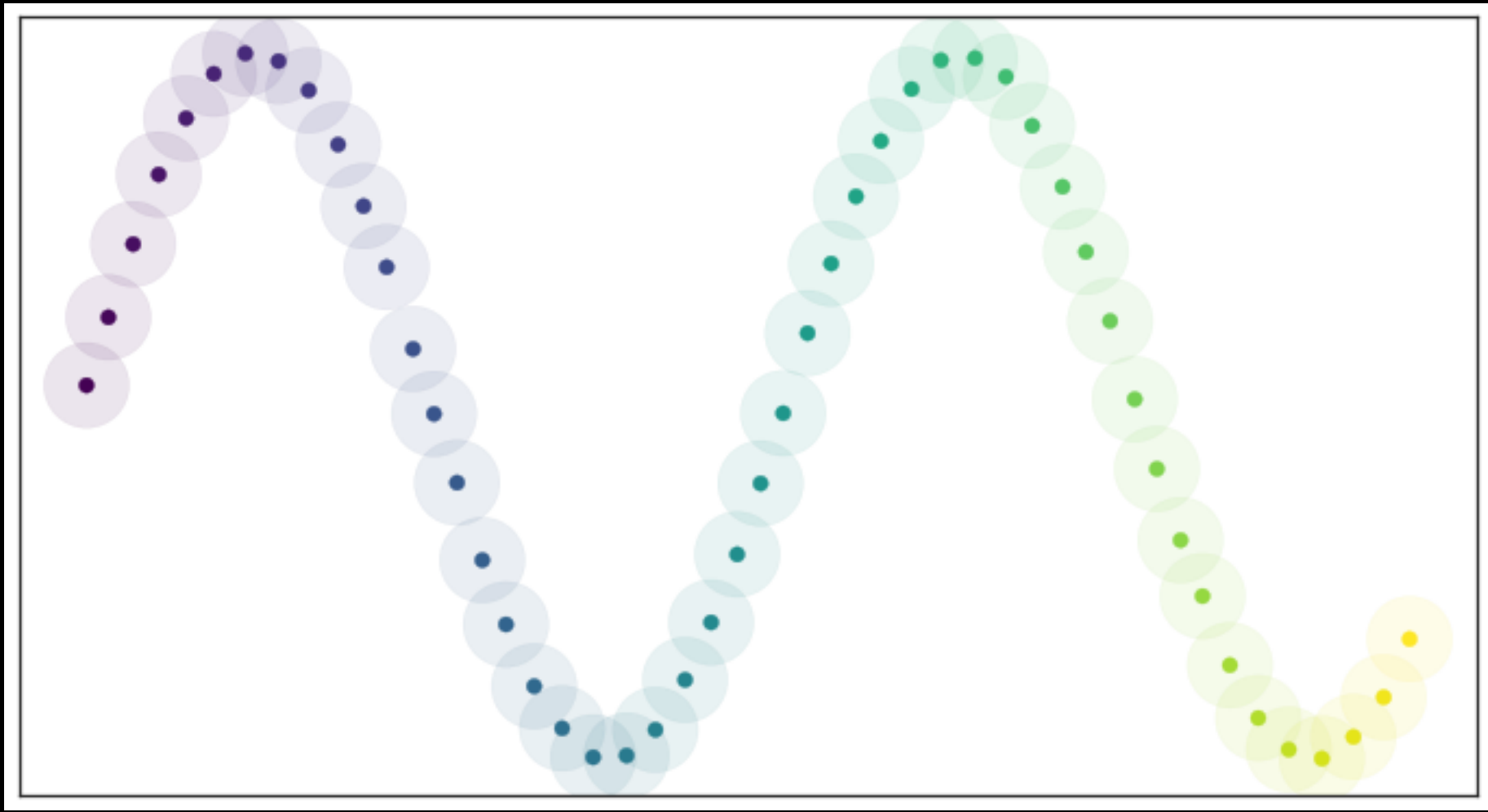
Tutaj pojawia się pierwszy istotny problem:

Tutaj pojawia się pierwszy istotny problem:

Jak dobrać promień tych kul?

Zauważmy, że bylibyśmy w stanie łatwo znaleźć promień w sytuacji gdyby dane były rozmieszczone jednostajnie na całej rozmaitości.

Zauważmy, że bylibyśmy w stanie łatwo znaleźć promień w sytuacji gdyby dane były rozmieszczone jednostajnie na całej rozmaitości.



Ale na ogół nie są.

Ale na ogół nie są.

Mimo to, załóżmy, że dane są jednolicie rozmieszczone na pewnej rozmaitości.

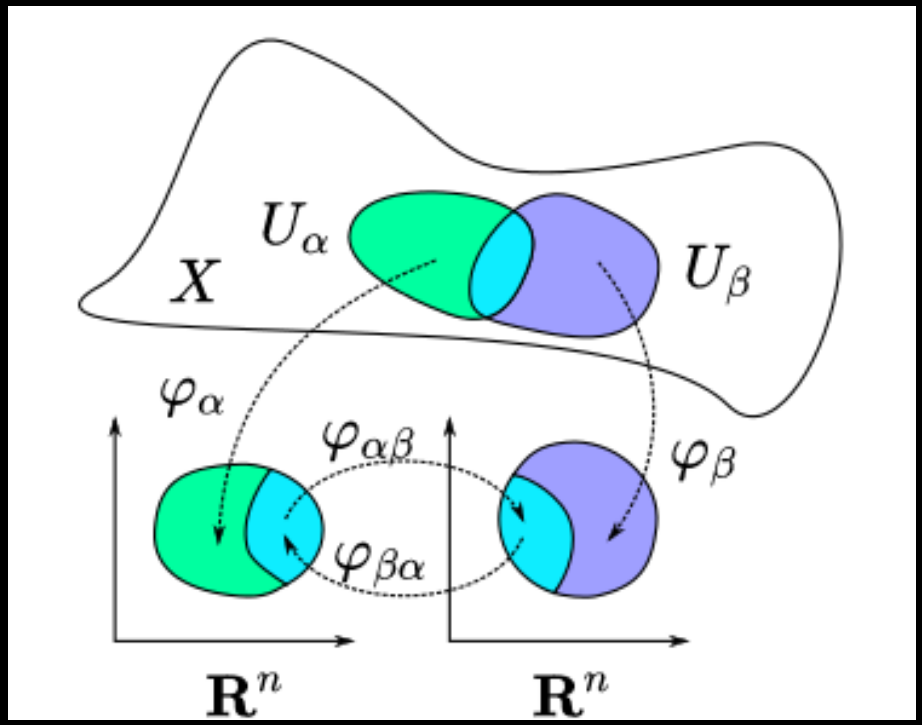
Ale na ogół nie są.

Mimo to, załóżmy, że dane są jednolicie rozmieszczone na pewnej rozmaitości.

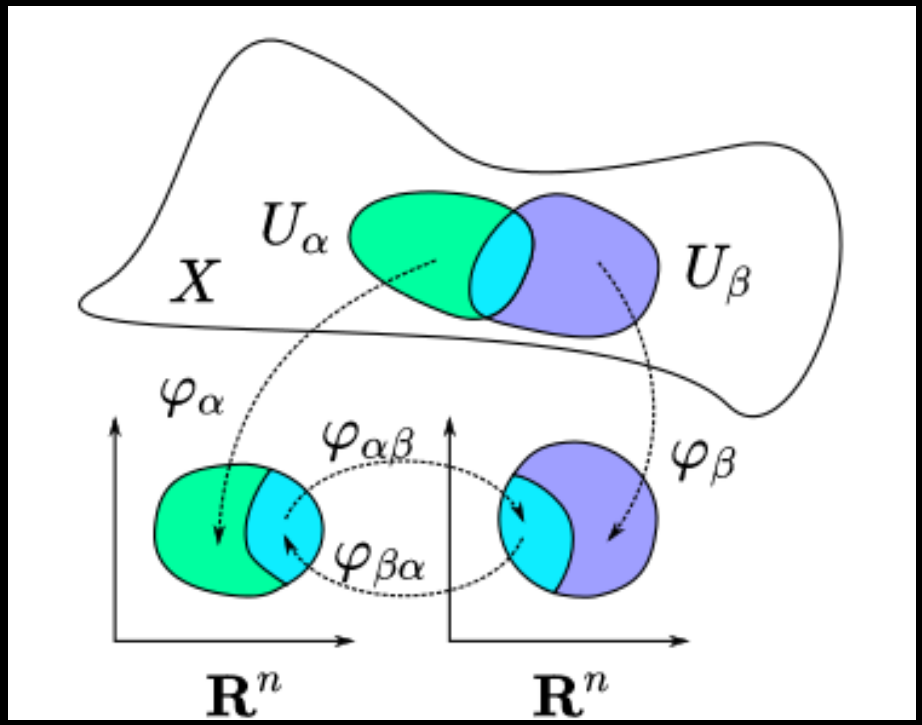
Co podówczas możemy powiedzieć o tej rozmaitości?

Jeśli dane nie są rozmieszczone jednolicie to możemy uznać, że pojęcie dystansu (metryka) zmienia się na powierzchni rozmaitości.

Jeśli dane nie są rozmieszczone jednolicie to możemy uznać, że pojęcie dystansu (metryka) zmienia się na powierzchni rozmaitości.



Jeśli dane nie są rozmieszczone jednolicie to możemy uznać, że pojęcie dystansu (metryka) zmienia się na powierzchni rozmaitości.

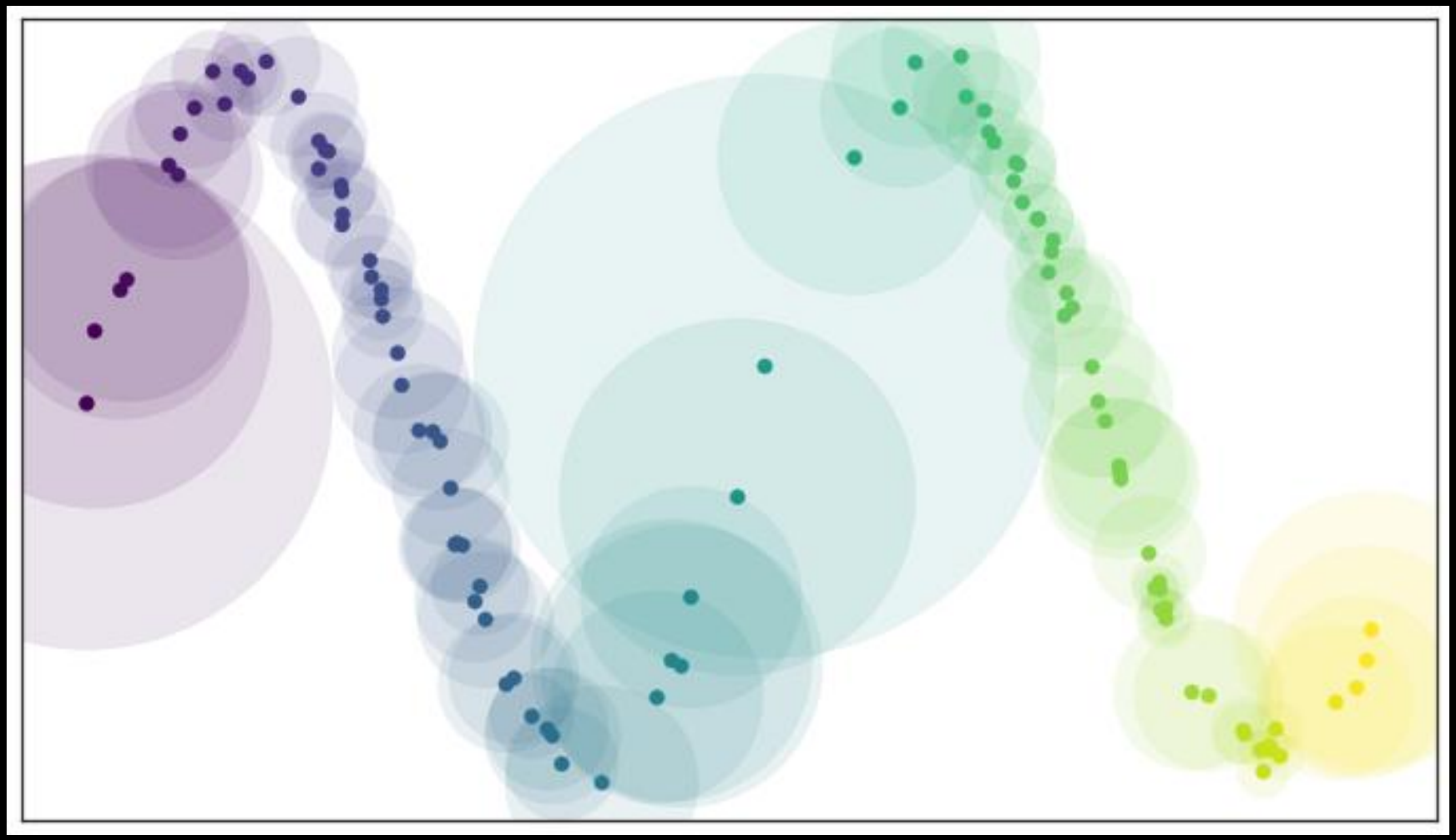


Możemy wówczas przybliżać lokalną metrykę - np. przez KNN - i na jej podstawie budować pokrycia otwarte

Problem jest rozwiązany o tyle, o ile pozostaje do ustalenia wartość K algorytmu KNN.

Problem jest rozwiązany o tyle, o ile pozostaje do ustalenia wartość K algorytmu KNN.

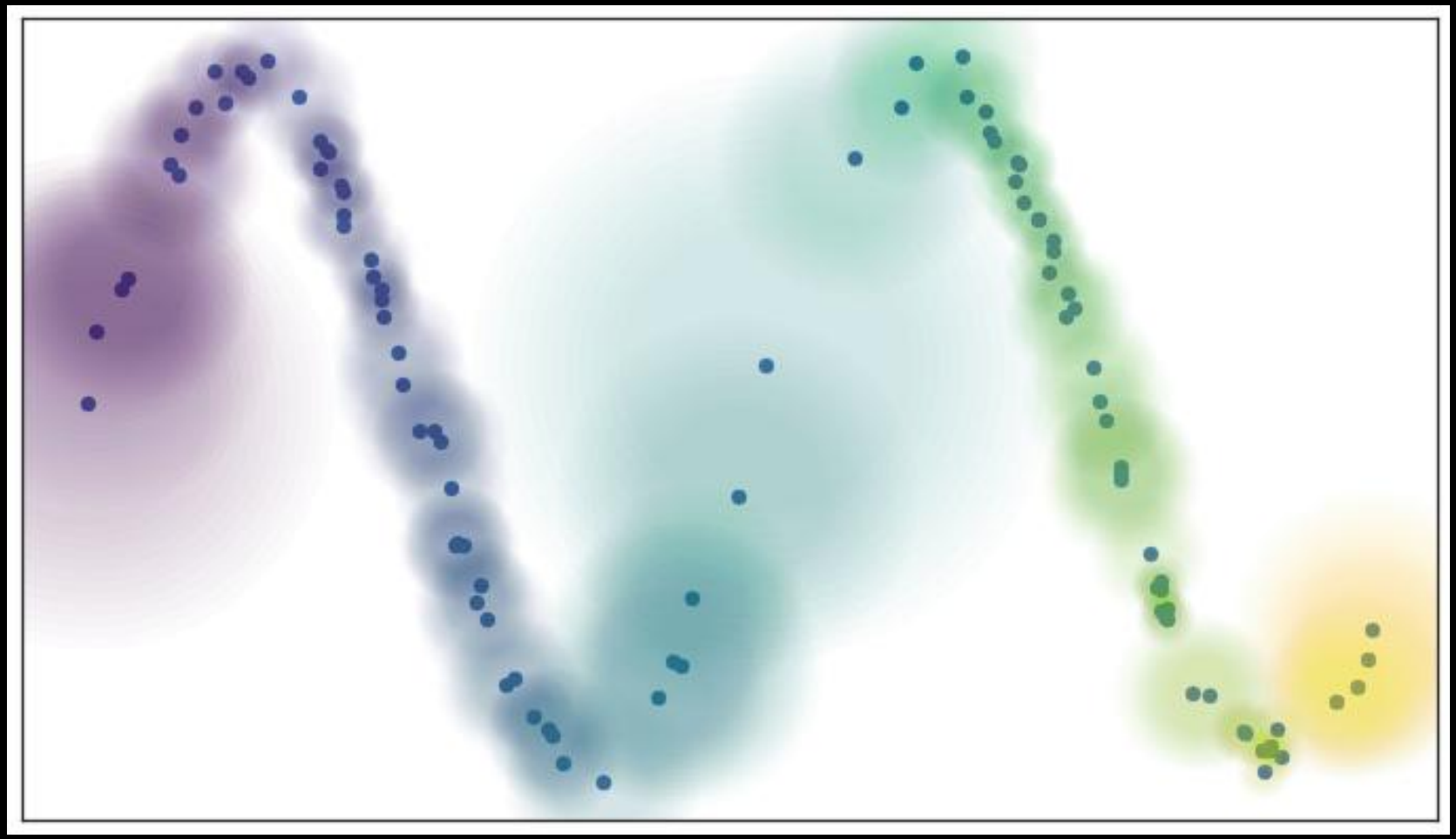
Parametr K pozwala nam kontrolować lokalność reprezentacji (im mniejszy tym bardziej lokalna).



Mając przybliżenie metryki riemannowskiej tej rozmaitości możemy mierzyć dystans pomiędzy punktami, w szczególności stworzyć ważony graf podobieństwa pomiędzy punktami danych.

Mając przybliżenie metryki riemannowskiej tej rozmaitości możemy mierzyć dystans pomiędzy punktami, w szczególności stworzyć ważony graf podobieństwa pomiędzy punktami danych.

Ma to prostą interpretację w sensie zbiorów rozmytych: określamy stopień przynależności do danego zbioru otwartego.



Tu pojawia się nowy problem:

Tu pojawia się nowy problem:

Wiele punktów jest izolowanych!

Ten problem rozwiązuje pojęcie **lokalnej spójności**: wokół każdego punktu rozmaitości istnieje pewne spójne otoczenie.

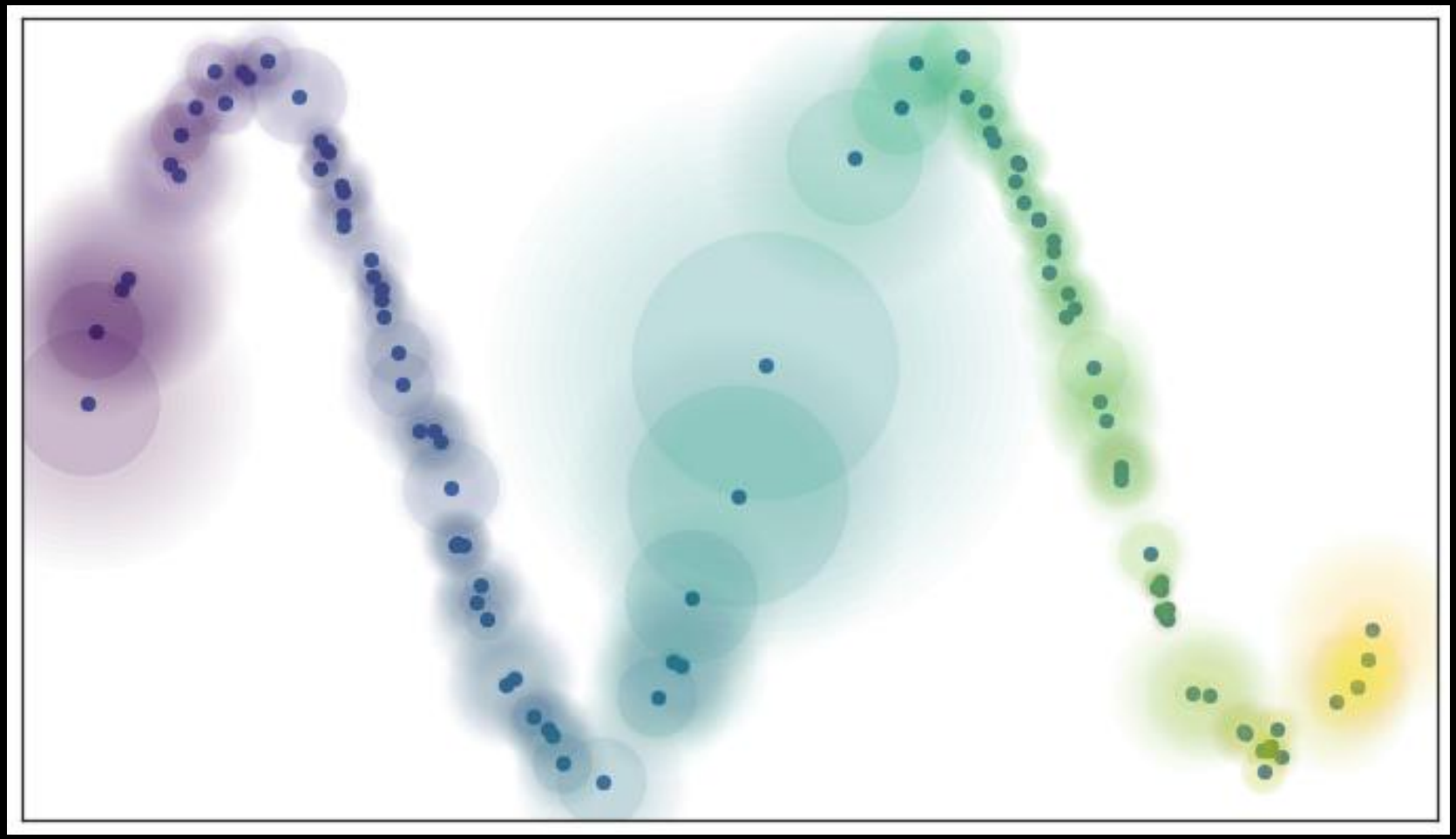
Ten problem rozwiązuje pojęcie **lokalnej spójności**: wokół każdego punktu rozmaitości istnieje pewne spójne otoczenie.

A w sensie zbiorów skończonych –
nie ma punktów izolowanych.

Ten problem rozwiązuje pojęcie **lokalnej spójności**: wokół każdego punktu rozmaitości istnieje pewne spójne otoczenie.

A w sensie zbiorów skończonych –
nie ma punktów izolowanych.

Dla zbiorów rozmytych oznacza to tyle, że otoczenie do najbliższego sąsiada ma stopień przynależności równy 1.



Kolejny problem:

Kolejny problem:

Te metryki nie są spójne.

Kolejny problem:

Te metryki nie są spójne.

Według metryki punktu A jego odległość od punktu B może być inna niż w sensie metryki punktu B.

Kolejny problem:

Te metryki nie są spójne.

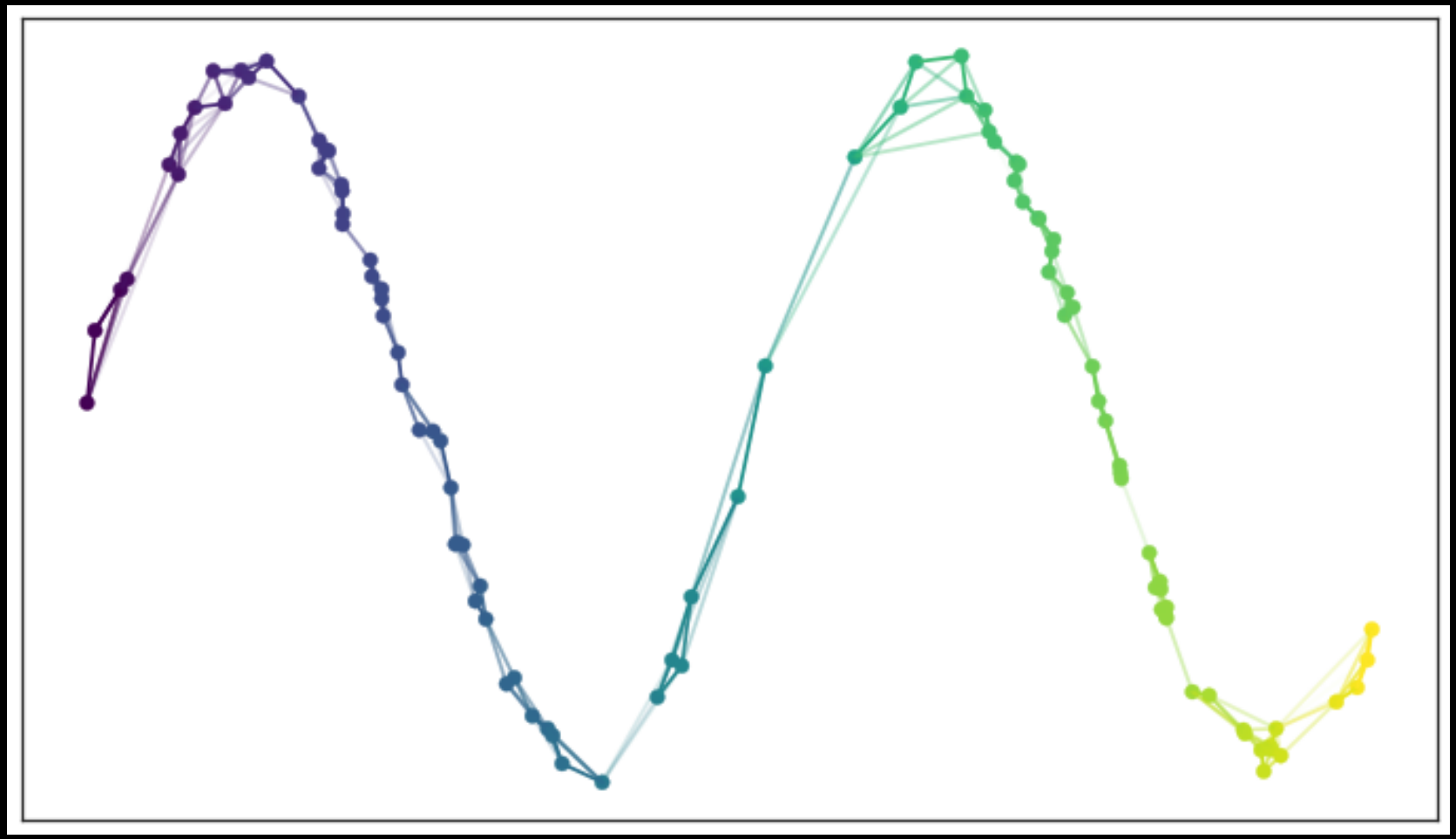
Według metryki punktu A jego odległość od punktu B może być inna niż w sensie metryki punktu B.

Jak rozstrzygać takie konflikty?

Przyjmując odległości jako p-stwa istnienia krawędzi, definiujemy odległość pomiędzy punktami jako p-stwo że przynajmniej jedna z krawędzi istnieje, a więc dla wag **a** i **b** kładziemy **d(a,b) = a+b–a*b**

Przyjmując odległości jako p-stwa istnienia krawędzi, definiujemy odległość pomiędzy punktami jako p-stwo że przynajmniej jedna z krawędzi istnieje, a więc dla wag **a** i **b** kładziemy **d(a,b) = a+b–a*b**

Dla tak zdefiniowanych krawędzi reprezentacja topologiczna danych jest rozmytym kompleksem symplicjalnym, który możemy reprezentować za pomocą ważonego grafu.



Kolejny krok to znalezienie embeddingu, który ma możliwie podobną reprezentację topologiczną.

Kolejny krok to znalezienie embeddingu, który ma możliwie podobną reprezentację topologiczną.

Reprezentację topologiczną możemy wyliczać w taki sam sposób jak dla danych. Różnica polega na tym, że dla embeddingu nie przybliżamy metryki riemannowskiej, bo wiemy jak ma wyglądać metryka rozmaitości - ma być euklidesowa (liczba wymiarów jest parametrem).

Musimy umieć porównywać reprezentacje danych i embeddingów.

Musimy umieć porównywać reprezentacje danych i embeddingów.

Dane i embeddingi dzielą 0-sympleksy.

Musimy umieć porównywać reprezentacje danych i embeddingów.

Dane i embeddingi dzielą 0-sympleksy.

Jeśli uznamy wagi za p-stwa istnienia krawędzi, to porównywanie 1-sympleksów sprowadza się do porównywania dwu wektorów prawdopodobieństw.

Musimy umieć porównywać reprezentacje danych i embeddingów.

Dane i embeddingi dzielą 0-sympleksy.

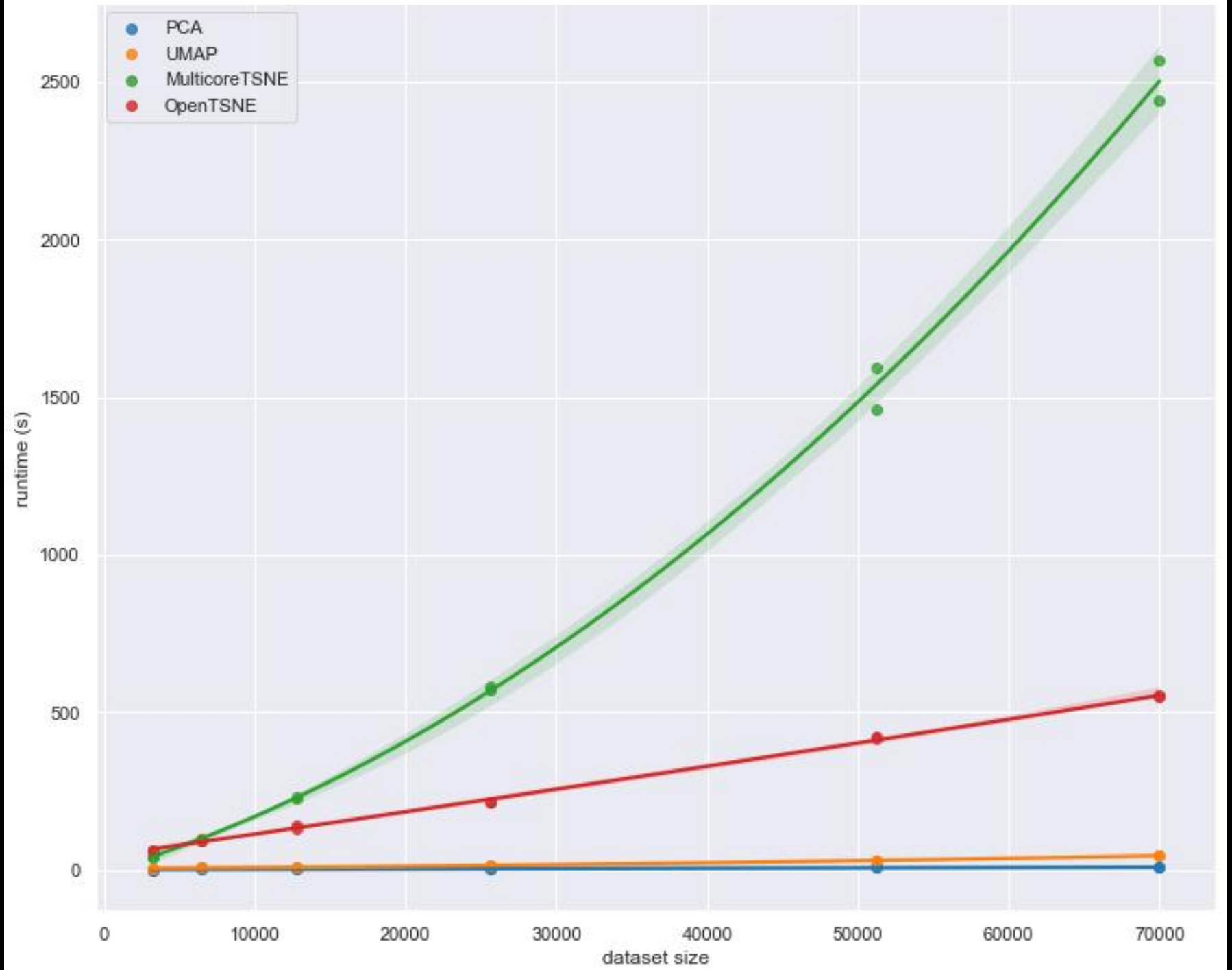
Jeśli uznamy wagi za p-stwa istnienia krawędzi, to porównywanie 1-sympleksów sprowadza się do porównywania dwu wektorów prawdopodobieństw.

Przy założeniu, że mają one rozkład zero-jedynkowy, za miarę podobieństwa możemy wziąć entropię krzyżową.

Część odpowiedzialna
za skupienia

$$\sum_{e \in E} w_h(e) \log\left(\frac{w_h(e)}{w_l(e)}\right) + (1 - w_h(e)) \log\left(\frac{1 - w_h(e)}{1 - w_l(e)}\right)$$

Część odpowiedzialna
za przerwy


PCA
UMAP
MulticoreTSNE
OpenTSNE
runtime (s)
2500
2000
1500
1000
500
0
0
10000
20000
30000
40000
50000
60000
70000
dataset size

Zalety

Zalety

- UMAP działa szybciej niż t-SNE i lepiej się skaluje tak z licznymi, jak i wielowymiarowymi danymi;

Zalety

- UMAP działa szybciej niż t-SNE i lepiej się skaluje tak z licznymi, jak i wielowymiarowymi danymi;
- oprócz redukcji wymiaru, UMAP uczy się również metryki embeddowanej przestrzeni, co można wykorzystać do zadań uczenia nadzorowanego;

Zalety

- UMAP działa szybciej niż t-SNE i lepiej się skaluje tak z licznymi, jak i wielowymiarowymi danymi;
- oprócz redukcji wymiaru, UMAP uczy się również metryki embeddowanej przestrzeni, co można wykorzystać do zadań uczenia nadzorowanego;
- UMAP uwzględnia tak lokalną strukturę danych, jak i globalną.

Zalety

- UMAP działa szybciej niż t-SNE i lepiej się skaluje tak z licznymi, jak i wielowymiarowymi danymi;
- oprócz redukcji wymiaru, UMAP uczy się również metryki embeddowanej przestrzeni, co można wykorzystać do zadań uczenia nadzorowanego;
- UMAP uwzględnia tak lokalną strukturę danych, jak i globalną.
- dostępne są implementacje algorytmu UMAP w językach Python, R, Julia

Wady

Wady

- Podstawowe założenie UMAPa – że dane są położone na pewnej rozmaitości – sprawia, że UMAP jest mało odporny na szum w danych (szczególnie dotyczy to małych zbiorów danych);

Wady

- Podstawowe założenie UMAPa – że dane są położone na pewnej rozmaitości – sprawia, że UMAP jest mało odporny na szum w danych (szczególnie dotyczy to małych zbiorów danych);
- chociaż globalna struktura danych jest brana pod uwagę, to jednak UMAP głównie odwzorowuje lokalne zależności;

Wady

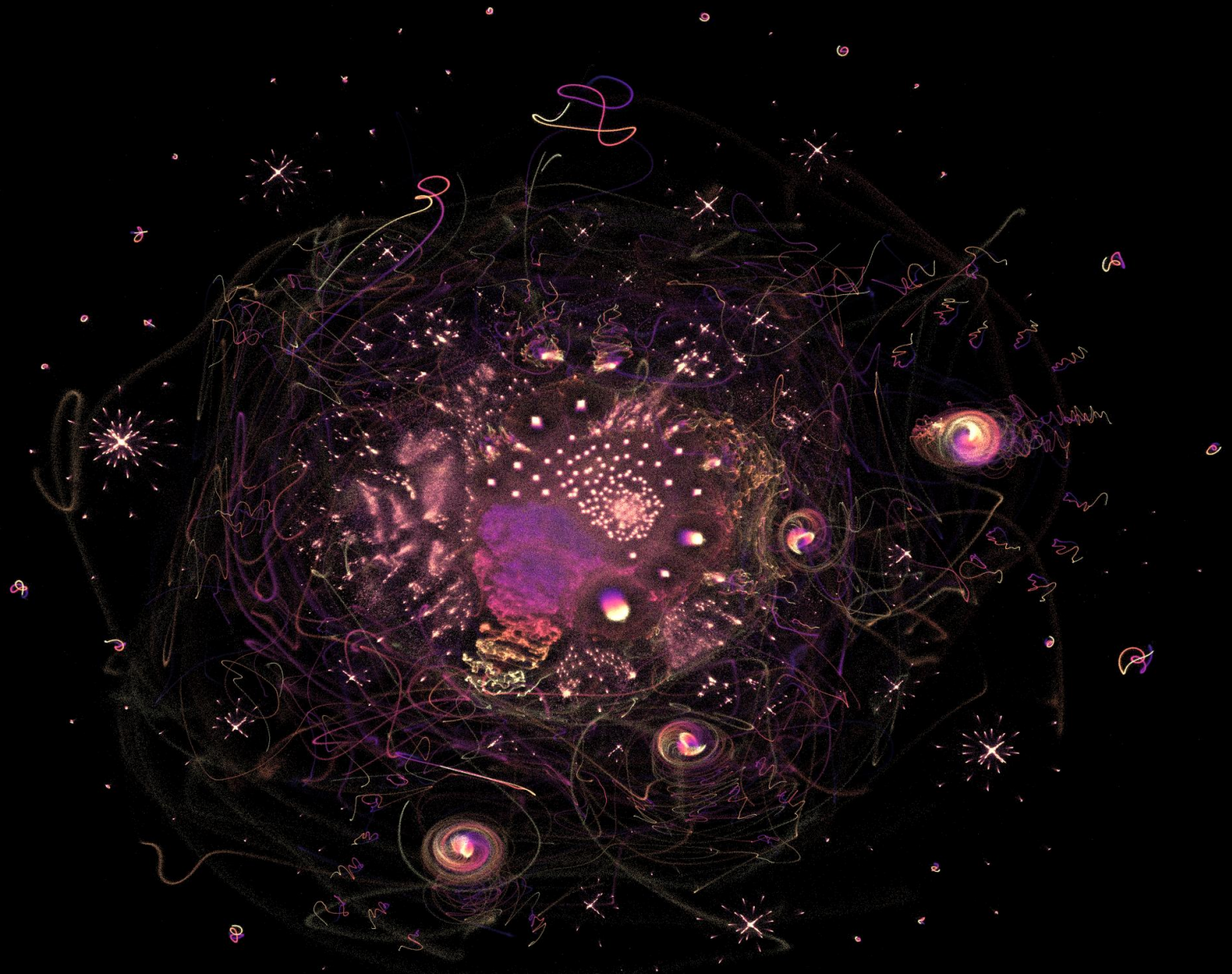
- Podstawowe założenie UMAPa – że dane są położone na pewnej rozmaitości – sprawia, że UMAP jest mało odporny na szum w danych (szczególnie dotyczy to małych zbiorów danych);
- chociaż globalna struktura danych jest brana pod uwagę, to jednak UMAP głównie odwzorowuje lokalne zależności;
- UMAP nie produkuje interpretowalnych wyników;

Wady

- Podstawowe założenie UMAPa – że dane są położone na pewnej rozmaitości – sprawia, że UMAP jest mało odporny na szum w danych (szczególnie dotyczy to małych zbiorów danych);
- chociaż globalna struktura danych jest brana pod uwagę, to jednak UMAP głównie odwzorowuje lokalne zależności;
- UMAP nie produkuje interpretowalnych wyników;
- UMAP bazuje na licznych aproksymacjach, np. do liczenia KNN, stąd nie nadaje się do małych zbiorów danych (<500);

Bibliografia

- McInnes, L, Healy, J, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, ArXiv e-prints 1802.03426, 2018.
- David I Spivak, *Metric realization of fuzzy simplicial sets,* Self published notes, 2012.
- dokumentacja UMAP w wersji języka Python jest dostępna pod linkiem: umap-learn.readthedocs.io



Dziękuję za uwagę