O szukaniu dziury w całym

Czyli analiza danych za pomocą topologii

Bartosz Furmanek

Program na dzisiaj

Dlaczego?

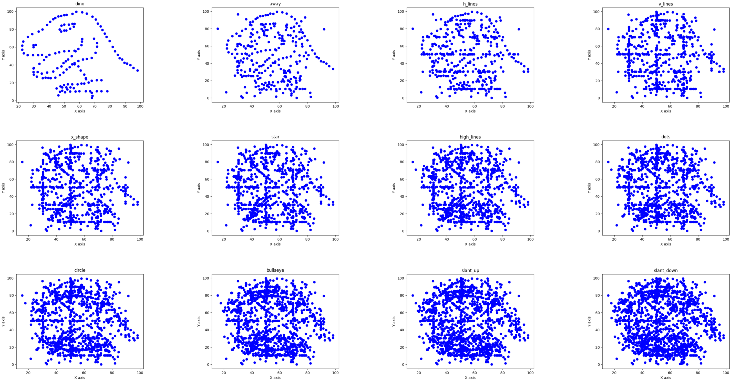
Kompleksy

- Wstęp
- Kompleks kostkowy
- Filtracja

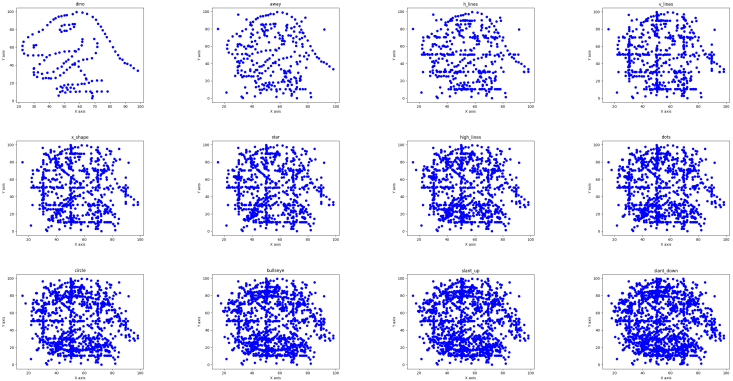
Stabilność diagramów persystencji

Uwaga: fioletowe komentarze służą do tego, żebym nie zapomniał niczego. Proszę zwracać mi uwagę, gdybym je zignorował!!!
Wszystkie materiały są dostępne na GitHubie:
https://github.com/BartTheBartender/o-szukaniu-dziury-w-calym.

Datasaurus Dozen

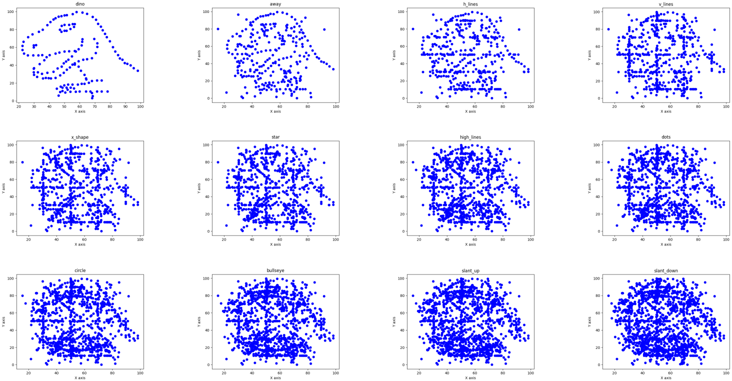


Datasaurus Dozen



Jupter notebook: datasaurus – bez bottleneck distance na razie.

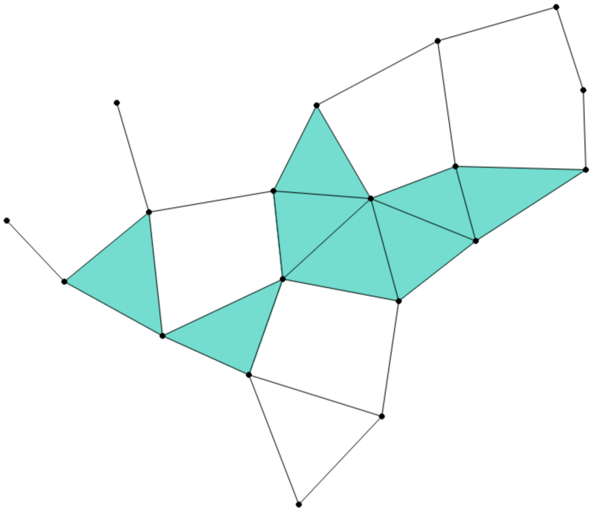
Datasaurus Dozen



Jupter notebook: datasaurus – bez bottleneck distance na razie.
Jupter notebook: cat – tylko pokazać.

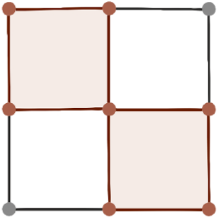
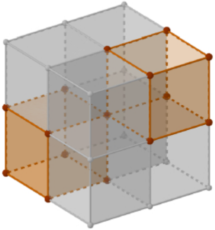
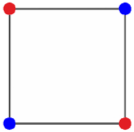
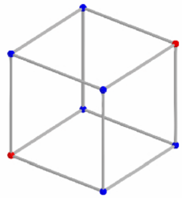
Kompleks symplicjalny

Kompleks symplicjalny zbudowany jest z *sympleksów*, czyli punktów, odcinków, trójkątów, czworościanów, itd. ...



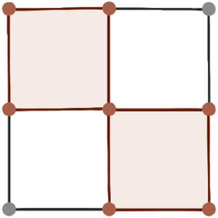
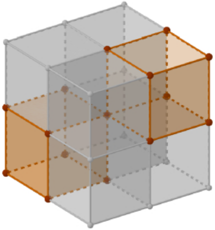
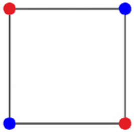
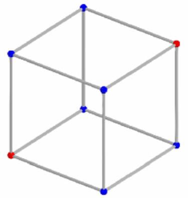
Kompleks kostkowy

... natomiast *kompleks kostkowy* zbudowany jest z *kostek*, czyli punktów, odcinków, kwadratów, sześcianów, itd.

Kompleks kostkowy

... natomiast *kompleks kostkowy* zbudowany jest z *kostek*, czyli punktów, odcinków, kwadratów, sześcianów, itd.

Jak wygląda tutaj liczenie homologii?

Porównanie

Porównanie

Kompleks symplicialny

Porównanie

Kompleks symplicialny
Dowolny zbiór punktów na płaszczyźnie, w przestrzeni ... można *ztriangulować*, czyli podzielić na sympleksy

Porównanie

Kompleks symplicialny
Dowolny zbiór punktów na płaszczyźnie, w przestrzeni ... można *ztriangulować*, czyli podzielić na sympleksy

Kompleks kostkowy

Porównanie

Kompleks symplicialny
Dowolny zbiór punktów na płaszczyźnie, w przestrzeni ... można *ztriangulować*, czyli podzielić na sympleksy

Kompleks kostkowy
Posiada naturalny sposób indeksowania kostek, co więcej często wystarcza pamiętać najwyżejwymiarowe kostki.

Filtracja

Jeśli mamy kompleks $\mathcal{K}$, to *filtracją na* $\mathcal{K}$ nazywamy ustawienie sympleksów/kostek po kolei, tak, żeby każdy sympleks/kostka pojawiał(a) się po wszystkich swoich ścianach.

Filtracja

Jeśli mamy kompleks $\mathcal{K}$, to *filtracją na* $\mathcal{K}$ nazywamy ustawienie sympleksów/kostek po kolei, tak, żeby każdy sympleks/kostka pojawiał(a) się po wszystkich swoich ścianach. Przykład filtracji – tak naprawdę to już znacie.

Filtracja

Jeśli mamy kompleks $\mathcal{K}$, to *filtracją na* $\mathcal{K}$ nazywamy ustawienie sympleksów/kostek po kolei, tak, żeby każdy sympleks/kostka pojawiał(a) się po wszystkich swoich ścianach. Przykład filtracji – tak naprawdę to już znacie. Mniej oczywisty przykład: jupyter notebook cubical – bez bottleneck na razie.

Kompleks Vietorisa-Ripsa

Niech $A = a_0, a_1 \ldots a_n \subseteq \mathbb{R}^n$. Może to być zbiór punktów w dowolnie wymiarowej przestrzeni. Ustalamy promień $r \geqslant 0$. *Kompleks Vietorisa-Ripsa dla A i r* składa się z takich sympleksów $S \subseteq A$, że każde dwa punkty $x, y \in S$ odległe są o co najwyżej r.

Kompleks Vietorisa-Ripsa

Niech $A = a_0, a_1 \ldots a_n \subseteq \mathbb{R}^n$. Może to być zbiór punktów w dowolnie wymiarowej przestrzeni. Ustalamy promień $r \geqslant 0$. *Kompleks Vietorisa-Ripsa dla* A *i* r składa się z takich sympleksów $S \subseteq A$, że każde dwa punkty $x, y \in S$ odległe są o co najwyżej r.
Python: complex visualization: Vietoris Rips

Kompleks Čecha

Niech $A = a_0, a_1 \ldots a_n \subseteq \mathbb{R}^n$. Może to być zbiór punktów w dowolnie wymiarowej przestrzeni. Ustalamy promień $r \geqslant 0$. *Kompleks Čecha dla A i r* składa się z takich sympleksów $S \subseteq A$, że wszystkie kule o środkach w punktach $s \in S$ i promieniu r mają wspólny punkt.

Kompleks Čecha

Niech $A = a_0, a_1 \dots a_n \subseteq \mathbb{R}^n$. Może to być zbiór punktów w dowolnie wymiarowej przestrzeni. Ustalamy promień $r \geqslant 0$. *Kompleks Čecha dla A i r* składa się z takich sympleksów $S \subseteq A$, że wszystkie kule o środkach w punktach $s \in S$ i promieniu r mają wspólny punkt.

Python: complex visualization: Čech

Porównanie

Porównanie

Kompleks Vietorisa-Ripsa

Porównanie

Kompleks Vietorisa-Ripsa
Oblicza się dużo szybciej niż kompleks Čecha.

Porównanie

Kompleks Vietorisa-Ripsa

Oblicza się dużo szybciej niż kompleks Čecha.

Kompleks Čecha

Porównanie

Kompleks Vietorisa-Ripsa
Oblicza się dużo szybciej niż kompleks Čecha.

Kompleks Čecha
Zachodzi *Twierdzenie Čecha o nerwie*: kompleks Čecha jest homotopijnie równoważny z sumą kul z których powstał.

Porównanie

Kompleks Vietorisa-Ripsa
Oblicza się dużo szybciej niż kompleks Čecha.

Kompleks Čecha
Zachodzi *Twierdzenie Čecha o nerwie*: kompleks Čecha jest homotopijnie równoważny z sumą kul z których powstał.

Jupyter Notebook: cat – do końca

Porównanie

Kompleks Vietorisa-Ripsa
Oblicza się dużo szybciej niż kompleks Čecha.

Kompleks Čecha
Zachodzi *Twierdzenie Čecha o nerwie*: kompleks Čecha jest homotopijnie równoważny z sumą kul z których powstał.

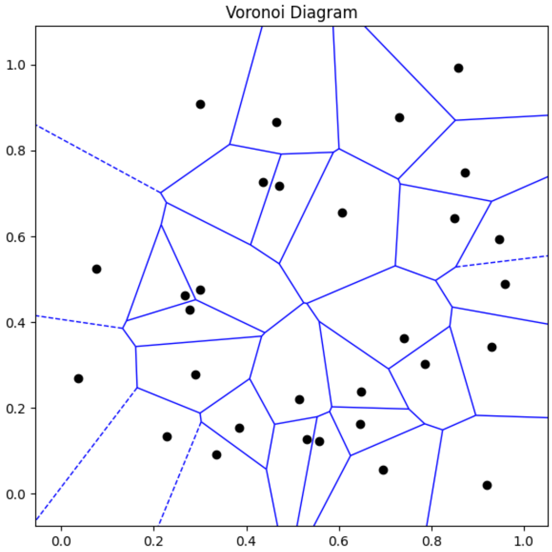
Jupyter Notebook: cat – do końca

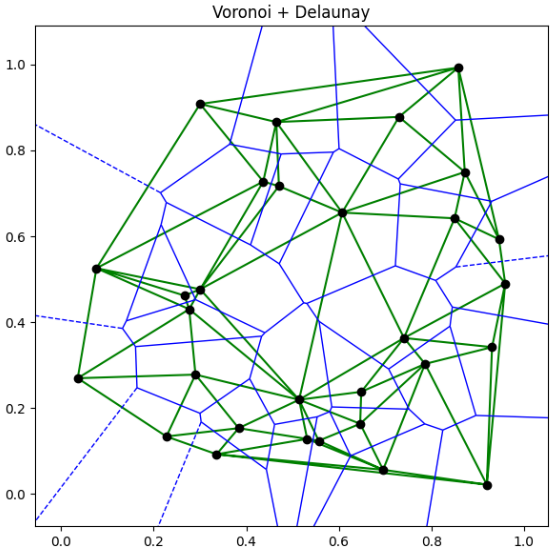
Pojawia się jednak bardzo ważne pytanie ...

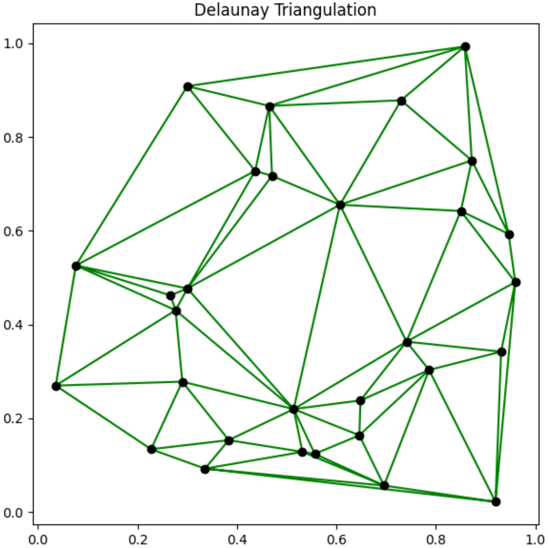
ILE WYMIARÓW MAJĄ TE KOMPLEKSY?

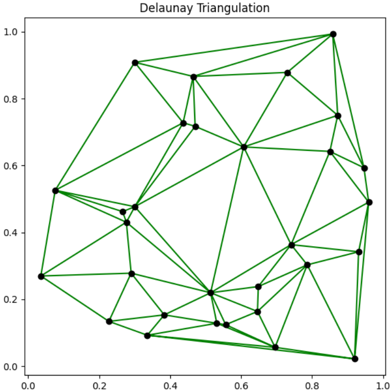
ILE WYMIARÓW MAJĄ TE KOMPLEKSY?

Rozwiązania są dwa: albo ignorujemy sympleksy wysokich wymiarów, albo w przypadku kompleksu Čecha modyfikujemy konstrukcję.

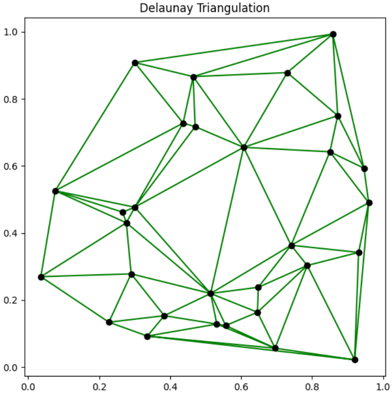

Voronoi Diagram
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0




Delaunay Triangulation
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



Delaunay Triangulation
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



Teraz możemy zdefiniować *alpha complex* dla zbioru $A \subseteq \mathbb{R}^n$ oraz $r \geqslant 0$. Sympleksami w nim są takie sympleksy z triangulacji Delaunaya, że wszystkie kule o środkach w wierzchołkach tych sympleksów i promieniu r mają wspólny punkt.

Niech $\mathcal{K}$ będzie kompleksem symplicjalnym. *Filtr na* $\mathcal{K}$ to funkcja $\mathcal{K} \to \mathbb{R}$, która zadaje filtrację. Przykłady to (w prezentacji były dwa):

Niech $\mathcal{K}$ będzie kompleksem symplicjalnym. *Filtr na* $\mathcal{K}$ to funkcja $\mathcal{K} \to \mathbb{R}$, która zadaje filtrację. Przykłady to (w prezentacji były dwa):

- Natężenie jasności piksela.

Niech $\mathcal{K}$ będzie kompleksem symplicjalnym. *Filtr na* $\mathcal{K}$ to funkcja $\mathcal{K} \to \mathbb{R}$, która zadaje filtrację. Przykłady to (w prezentacji były dwa):

- Natężenie jasności piksela.
- Wartość r, dla którego w kompleksie Vietorisa-Ripsa/Čecha/alpha dodawany jest sympleks

Niech $\mathcal{K}$ będzie kompleksem symplicjalnym. *Filtr na* $\mathcal{K}$ to funkcja $\mathcal{K} \to \mathbb{R}$, która zadaje filtrację. Przykłady to (w prezentacji były dwa):

- Natężenie jasności piksela.
- Wartość r, dla którego w kompleksie Vietorisa-Ripsa/Čecha/alpha dodawany jest sympleks

Idea: mała zmiana filtra mało wpływa na zmianę diagramu persystencji.

Jeśli f, g są filtrami na $\mathcal{K}$, to chcemy sensownie zdefiniować odległość między nimi. Przyjmujemy

$$d(f,g) := \max_{\sigma \in \mathcal{K}} |f(\sigma) - g(\sigma)|.$$

Przykład?

Jeśli f, g są filtrami na $\mathcal{K}$, to przez $\mathrm{Dgm}(f)$, $\mathrm{Dgm}(g)$ oznaczymy ich diagramy persystencji. Jak policzyć odległość między nimi?

Jeśli f, g są filtrami na $\mathcal{K}$, to przez $\mathrm{Dgm}(f)$, $\mathrm{Dgm}(g)$ oznaczymy ich diagramy persystencji. Jak policzyć odległość między nimi?

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) := \min_{\phi:\mathrm{Dgm}(f)\to\mathrm{Dgm}(g)} \max_{x\in\mathrm{Dgm}(f)} \|x - \phi(x)\|$$

Nazywamy to *bottleneck distance*.

Jeśli f, g są filtrami na $\mathcal{K}$, to przez $\mathrm{Dgm}(f)$, $\mathrm{Dgm}(g)$ oznaczymy ich diagramy persystencji. Jak policzyć odległość między nimi?

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) := \min_{\phi:\mathrm{Dgm}(f)\to\mathrm{Dgm}(g)} \max_{x\in\mathrm{Dgm}(f)} \|x - \phi(x)\|$$

Nazywamy to *bottleneck distance*.

Przykład? – Jupyter Notebooks: datasaurus & cubical ale najpierw

...

Niech f, g będą filtrami na $\mathcal{K}$ oraz

$$d(f, g) := \max_{\sigma \in \mathcal{K}} |f(\sigma) - g(\sigma)|,$$

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) := \min_{\phi : \mathrm{Dgm}(f) \to \mathrm{Dgm}(g)} \max_{x \in \mathrm{Dgm}(f)} \|x - \phi(x)\|$$

Niech f, g będą filtrami na $\mathcal{K}$ oraz

$$d(f, g) := \max_{\sigma \in \mathcal{K}} |f(\sigma) - g(\sigma)|,$$

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) := \min_{\phi : \mathrm{Dgm}(f) \to \mathrm{Dgm}(g)} \max_{x \in \mathrm{Dgm}(f)} \|x - \phi(x)\|$$

Zachodzi wówczas

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) \leqslant d(f, g),$$

z czego wynika, że

Niech f, g będą filtrami na $\mathcal{K}$ oraz

$$d(f, g) := \max_{\sigma \in \mathcal{K}} |f(\sigma) - g(\sigma)|,$$

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) := \min_{\phi : \mathrm{Dgm}(f) \to \mathrm{Dgm}(g)} \max_{x \in \mathrm{Dgm}(f)} \|x - \phi(x)\|$$

Zachodzi wówczas

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) \leqslant d(f, g),$$

z czego wynika, że

- mała zmiana filtra nie zaburza nam znacznie diagramu persystencji – robimy to od samego początku !!!

Niech f, g będą filtrami na $\mathcal{K}$ oraz

$$d(f, g) := \max_{\sigma \in \mathcal{K}} |f(\sigma) - g(\sigma)|,$$

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) := \min_{\phi : \mathrm{Dgm}(f) \to \mathrm{Dgm}(g)} \max_{x \in \mathrm{Dgm}(f)} \|x - \phi(x)\|$$

Zachodzi wówczas

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) \leqslant d(f, g),$$

z czego wynika, że

- mała zmiana filtra nie zaburza nam znacznie diagramu persystencji – robimy to od samego początku !!!
- Jeśli $d(\mathrm{Dgm}(f), \mathrm{Dgm}(g))$ jest duże, to filtry muszą znacznie się różnić.

Niech f, g będą filtrami na $\mathcal{K}$ oraz

$$d(f,g) := \max_{\sigma \in \mathcal{K}} |f(\sigma) - g(\sigma)|,$$

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) := \min_{\phi : \mathrm{Dgm}(f) \to \mathrm{Dgm}(g)} \max_{x \in \mathrm{Dgm}(f)} \|x - \phi(x)\|$$

Zachodzi wówczas

$$d(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) \leqslant d(f,g),$$

z czego wynika, że

- mała zmiana filtra nie zaburza nam znacznie diagramu persystencji – robimy to od samego początku !!!
- Jeśli $d(\mathrm{Dgm}(f), \mathrm{Dgm}(g))$ jest duże, to filtry muszą znacznie się różnić.

Jupyter Notebooks: datasaurus & cubical