Clusters and graphs

Rick Jardine

Fields Institute

October 23, 2018

Finding star clusters

William and Caroline Herschel (late 1700s): count stars in regions of space.

Still how it's done today: sophisticated cameras/instruments (eg. ESA spacecraft Gaia, at L2 Lagrange point, online in 2013), with computers doing the star counts from images or data sets.

First data had evidence of a new cluster (Gaia 1) near Sirius.

Detection method finds regions with high densities of stars ("stellar over-densities"). Primitive method of topological data analysis.

Relative to the "big picture", these are small, very dense collections of stars — anomalies.

Big picture item: structure of cosmic background microwave radiation.

Bank data

Record of transactions processed by a major bank (eg. Scotiabank, RBC, ...) over some time frame is a vast trove of data.

There are big pictures and small pictures associated with this data:

Big: Large scale market fluctuations, relations to world events, possible basis for predictions.

Small: Dense patterns of small transactions ($\leq$ \$10,000) in an account or between a small group of accounts could indicate money laundering.

The idea, at both a macro and micro level, is to find "clusters" in the data.

Finding clusters is a form of "unsupervised machine learning".

Topological Data Analysis

Clusters are collections of data points in relative close proximity, according to some finite list of parameters.

In practice, the parameters are real variables.

A **data cloud** is a finite set of points $X \subset \mathbb{R}^N$.

Basic idea: Analyze regions of the data cloud X, by density.

Various ways of saying what a cluster is:

- K-means clustering: Find centres of regions (Voronoi cells) of nearest neighbours to a given set of points. Use these centres to recalculate, repeatedly. Initial points chosen by eyeballing the data. Algorithm partitions the data set.
- Clusters are **subsets** of the data set that are relatively close together (distance $< s$ apart, for some variation of s), eg. DBSCAN, heirarchical clustering, HDBSCAN. Unsupervised. Isolates regions of interest.

Movie time

Video displays a run of a DBSCAN algorithm.

We have a data set $X \subset \mathbb{R}^2$ (a list). Start with first point $x_0 \in X$, a distance s, and number of vertices $k = 4$.

If x_0 has at least 4 other points in a disc of radius s centred at x_0, then x_0 is in a cluster. Perform same analysis for all new points in the disc to make the cluster grow, if possible.

If x_0 has less than 4 other points in the disc, move on to next point in the list X.

Tunable parameters: the choice of distance s and the number of other points k required in a disc. One-point clusters are noise.

This algorithm splits up (partitions) the points of X into "connected components".

https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

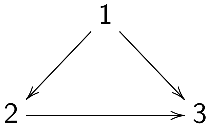
Graphs

A **graph** Γ consists of a set Γ_0 of **vertices**, a set Γ_1 of **edges**, and two functions $s, t : \Gamma_1 \to \Gamma_0$, called **source** and **target**.

Examples:

$\underline{N} = \{1, \dots, N\}$. The **complete graph** $K(\underline{N})$ has vertices all $1 \leq i \leq N$, ie, $K(\underline{N})_0 = \underline{N}$. The edges of this graph are the pairs of numbers $1 \leq i < j \leq N$.

- $K(\underline{1})$ is the vertex 1 with no edges.
- $K(\underline{2})$: $1 \to 2$.
- $K(\underline{3})$:



Example: Vietoris-Rips graphs

Start with a **data cloud** (a list) $X \subset \mathbb{R}^N$. Suppose that $s > 0$.

Rips graph: $V_s(X)$ has vertices $V_s(X)_0 = X$ and edges $V_s(X)_1$ consisting of elements $x, y \in X$ with $x < y$ in the list and $d(x,y) < s$.

- If $s < t$, then $V_s(X) \subset V_t(X)$
- $V_s(X) = X$ is discrete for s small, and is the complete graph $K(|X|)$ for for s big.

$X \subset \mathbb{R}^N$ is finite, so

- there is an s (small) so that every ball of radius s centred at $x \in X$ contains only x, ie. $V_s(X)$ has no edges.
- there is an R (big) such that every ball of radius R centred at $x \in X$ contains all elements of X, ie. $V_R(X) = K(|X|)$.

Any sequence $0 < s_1 < s_2 < \dots < s_r$ with s_1 sufficiently small and s_r sufficiently large determines a family of graphs (a "filtration")

$$X = V_{s_1}(X) \subset V_{s_2}(X) \subset \dots \subset V_{s_r}(X) = K(|X|).$$

Path components

Say that points x, y are in the same **path component** of a graph Γ (write $x \sim y$) if they can be joined by a string of edges

$$\begin{tikzcd}
& x_1 \arrow[dr, leftrightarrow] & & x_3 \arrow[dr, leftrightarrow] & & {} & \dots & {} \arrow[dr, leftrightarrow] & \\
x_0 \arrow[ur, leftrightarrow] & & x_2 \arrow[ur, leftrightarrow] & & x_4 \arrow[ur, leftrightarrow] & & \dots & & x_n
\end{tikzcd}$$

Picture is a path of edges of Γ between $x = x_0$ and $y = x_n$.

The path component relation $\sim$ splits up the vertices Γ_0 into a collection $\pi_0(\Gamma)$ of subsets.

NB: $[x]$ is an element of $\pi_0(\Gamma)$ **and** the component containing x.

$$X_0 = \bigsqcup_{[x] \in \pi_0(\Gamma)} [x].$$

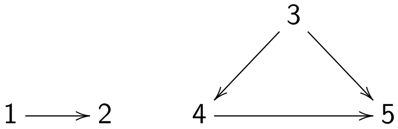
$\pi_0(\Gamma)$ is the set of **path components** of Γ.

Examples

1) Any two vertices of the complete graph $K(\underline{N})$ are in the same path component: $[i] = [j]$ if $i < j$.

$K(\underline{N})$ is **connected**. $\pi_0(K(\underline{N})) = *$.

2) Here's $K(\underline{2}) \sqcup K(\underline{3})$:



This graph has **two** path components: $\{1, 2\}$, $\{3, 4, 5\}$. Otherwise there would be an edge between the two pieces.

3) $X \subset \mathbb{R}^N$: $x, y \in X$ are in the same path component of $V_s(X)$ if there is a series of short hops (of length $< s$) through points of X.

$\pi_0 V_s(X) = X$ for s sufficiently small and $\pi_0 V_R(X) = *$ for R sufficiently large.

Varying the parameter s

For $x, y \in X$, write $x \sim_s y$ if x, y are in the same path component $[x]_s$ of $V_s(X)$.

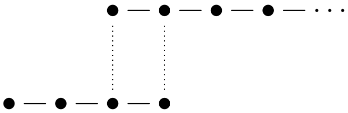
If $s < t$ and $x \sim_s y$, then $x \sim_t y$:

Hops of length $< s$ are of length $< t$.

There is a function of equivalence classes (path components)

$$\pi_0 V_s(X) \to \pi_0 V_t(X),$$

which is induced by the inclusion of graphs $V_s(X) \subset V_t(X)$.



What's going on?

Given $X \subset \mathbb{R}^N$, and $0 < s$, $\pi_0 V_s(X)$ defines a partition

$$X = \bigsqcup_{[x] \in \pi_0(V_s(X))} [x]$$

This partition is an "old style" clustering of X.

If $s < t$, every path component $[x]_s$ of $V_s(X)$ is contained in a path component of $V_t(X)$, namely $[x]_t$, which could be a bigger subset of X. $[x]_s \subset [x]_t$ as subsets of X.

Every path component of $V_t(X)$ is a union of path components of $V_s(X)$. The partition given by $\pi_0 V_s(X)$ is a **refinement** of that given by $\pi_0 V_t(X)$.

For $0 < s_1 < s_2 < \dots s_r$ the string of functions

$$\pi_0 V_{s_1}(X) \to \pi_0 V_{s_2}(X) \to \cdots \to \pi_0 V_{s_k}(X)$$

defines a **heirarchical** clustering of X, with progressively coarser partitions.

New cluster definition — first steps

In the general setup of Rips graphs $V_s(X)$ associated to a data cloud $X \subset \mathbb{R}^N$, we have:

- all $V_s(X)$ have the same set of vertices, namely X.
- each set of path components $\pi_0 V_s(X)$ defines a partition of X
- for $s < t$, the function $\pi_0 V_s(X) \to \pi_0 V_t(X)$ is defined by $[x]_s \mapsto [x]_t$, with
$$[x]_s \subset [x]_t \subset X.$$

A **cluster** for $X \subset \mathbb{R}^n$ is defined by a path component $[x]_s$ such that $[x]_s = [x]_t$ for some $t > s$.

The graph $\Gamma(F)$

$X \subset \mathbb{R}^N$. Choose $0 < s_1 < s_2 < \dots < s_r$. Set $V_i(X) = V_{s_i}(X)$.

We have a sequence of functions between path component sets

$$\pi_0 V_1(X) \to \pi_0 V_2(X) \to \cdots \to \pi_0 V_k(X)$$

Abstractly: $F: \ F(1) \xrightarrow{\alpha} F(2) \xrightarrow{\alpha} \dots \xrightarrow{\alpha} F(k) \xrightarrow{\alpha} \dots$

Graph $\Gamma(F)$: vertices (x,i), $x \in F(i)$, edges $(x,i) \to (\alpha(x), i+1)$.

$$\begin{tikzcd}
(y,1) \arrow[r, "\alpha"] & (y,2) \arrow[dr, "\alpha"] & & \\
& & (u,3) \arrow[r, "\alpha"] & \cdots \\
(z,1) \arrow[r, "\alpha"'] & (z,2) \arrow[ur, "\alpha"'] & &
\end{tikzcd}$$

A **branch point** is a vertex (x,i) with more than one incoming edge $(y, i-1) \to (x,i)$.

The cluster graph

Remove all edges of $\Gamma(F)$ terminating in branch points to construct subgraph $\Gamma_0(F) \subset \Gamma(F)$

$\Gamma_0(F)$ is the **cluster graph** for F.

Graphs have path components, and the **clusters** are the path components of $\Gamma_0(F)$, ie. elements of $\pi_0\Gamma_0(F)$.

Alternatively: A cluster of F is a path

$$(x_0, i) \to (x_1, i+1) \to \cdots \to (x_p, i+p)$$

of max length in $\Gamma(F)$ st no $(x_j, i+j)$ is a branch point for $j > 0$.

NB: (x_0, i) is a branch point, or x_0 has no preimage in $F(i-1)$.

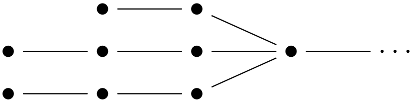
Example: For the sequence

$$\pi_0 V_*(X) : \ \pi_0 V_1(X) \to \pi_0 V_2(X) \to \cdots \to \pi_0 V_k(X)$$

the cluster graph $\Gamma_0(\pi_0 V_*(X))$ has edges $([x]_i, i) \to ([x]_{i+1}, i+1)$ such that the inclusion $[x]_i \to [x]_{i+1}$ is a *bijection*.

Small clusters

The isolated groups of bright objects define “small” clusters. They join other clusters at some parameter value, which could be large.



The small clusters are **noise** or **anomalies**.

Two ways to filter out noise:

1) Every element of $x_s \in \pi_0 V_s(X)$ has a cardinality $|x_s| = |[y]|$, where $x_s = [y]$. Score each cluster

$$P : \ (x_s, s) \to (x_{s+1}, s+1) \to \cdots \to (x_{s+p}, s+p)$$

by setting $\sigma(P) = |x_s| \cdot p$. Compare scores of clusters.

2) Throw away the path components of sufficiently small size during the computation process.

Comments

1) Components with big voids around them define clusters with higher scores than components of same size surrounded by smaller voids.

2) Scoring is relatively expensive. It can only be done after all other calculations.

3) Throwing away small path components (eg. isolated points, small groups) is brutal but computationally effective — can be done before constructing the cluster graph.

Higher dimensional persistence

The Rips graph $V_s(X)$ has subgraphs ("Lesnick graphs")

$$\dots \subset L_{s,k+1}(X) \subset L_{s,k}(X) \subset \dots L_{s,0}(X) = V_s(X)$$

defined by valence of vertices, and natural in s.

$x \in L_{s,k}(X)_0$ if it is a member of at least k edges of $V_s(X)$ ie. a ball of radius s centred on x contains at least k other members of X — a type of density measure.

For $s < t$, have a rectangular array of inclusions of graphs

$$\begin{tikzcd} L_{s,k}(X) \arrow[r] & L_{t,k}(X) \\ L_{s,k+1}(X) \arrow[u] \arrow[r] & L_{t,k+1}(X) \arrow[u] \end{tikzcd}$$

all with potentially different vertices.

Cluster graph

There is an induced array of path components

$$\begin{tikzcd}
\pi_0 L_{s,k}(X) \arrow[r] & \pi_0 L_{t,k}(X) \\
\pi_0 L_{s,k+1}(X) \arrow[r] \arrow[u] & \pi_0 L_{t,k+1}(X) \arrow[u]
\end{tikzcd}$$

and a graph $\Gamma(\pi_0 L_{*,*}(X))$ with vertices and edges

$$([x],(s,k)),\ \ [x] \in \pi_0 L_{s,k}(X),$$

$$([x],(s,k)) \to ([x],(t,k)),\ \ ([x],(s,k+1)) \to ([x],(s,k)).$$

The **cluster graph** $\Gamma_0(\pi_0 L_{*,*}(X))$ has edges which **preserve the size** of path components $[x]$.

The **clusters** of $L_{*,*}(X)$ are the path components of the cluster graph $\Gamma_0(\pi_0 L_{*,*}(X))$.

Remarks

Consider the picture

$$\begin{tikzcd}
\pi_0 V_{s_1}(X) \arrow[r] & \pi_0 V_{s_2}(X) \arrow[r] & \dots \arrow[r] & \pi_0 V_{s_r}(X) \\
\pi_0 L_{s_1,k}(X) \arrow[r] \arrow[u] & \pi_0 L_{s_2,k}(X) \arrow[r] \arrow[u] & \dots \arrow[r] & \pi_0 L_{s_r,k}(X) \arrow[u]
\end{tikzcd}$$

1) Running the "cluster algorithm" along the bottom row gives clusters for a higher density part of the data cloud X. Can tune the density of clusters by varying k

... except the meaning is not the same. Define the cluster graph for the bottom row as in the full 2-dimensional case above — can't just define the cluster graph by throwing away edges.

2) Admission: The algorithm of the smiley face video calculates $\pi_0 L_{s,k}(X)$ for a fixed s and k (s is whatever, and $k = 4$).

Scoring

A cluster P for $\{L_{s,k}(X)\}$ is a connected graph consisting of a set of vertices $(x,(s,k))$ with suitable edges.

For each vertex $(x,(s,k))$, the element x is a path component (a set of vertices) in $L_{s,k}(X)$.

The path component x has finite cardinality $|x|$, and this number is the same ($|x| = |y|$) for all points $(y,(t,i))$ in the cluster, by definition.

The **score** $\sigma(P)$ of the cluster P is defined by

$$\sigma(P) = \sum_{(x,(s,k))\in P} |x| = |x| \cdot |P|.$$

We deal with noise by throwing away clusters with low scores, or by throwing away points $(x,(s,k))$ with $|x|$ small, or both.

Patching

We would like, for programming purposes, to have a method of determining clusters from patches: $X = X_1 \cup X_2$.

Not simple, because it is **not** true that $V_s(X) = V_s(X_1) \cup V_s(X_2)$.

Example: Suppose that X_1 = blue dots and X_2 = red dots in the picture below, with X = all dots. Suppose s is slightly larger than the distance between adjacent dots.

Then $\pi_0 V_s(X_i) = X_i$ for $i = 1, 2$, but $\pi_0 V_s(X) = *$.

But ... can make "cluster trees" with edges given by clusters, and the cluster tree for X "is" a union of the cluster trees for X_1 and X_2. Don't know how to formalize this yet.

References

John Healy and Leland McInnes.
Accelerated heirarchical density clustering.
Preprint, arXiv: 1705.07321v2 [stat.ML], 2017.

M. Lesnick and M. Wright.
Interactive visualization of 2-d persistence modules.
Preprint, arXiv: 151.00180v1 [math.AT], 2015.

Afra Zomorodian and Gunnar Carlsson.
Computing persistent homology.
Discrete Comput. Geom., 33(2):249–274, 2005.