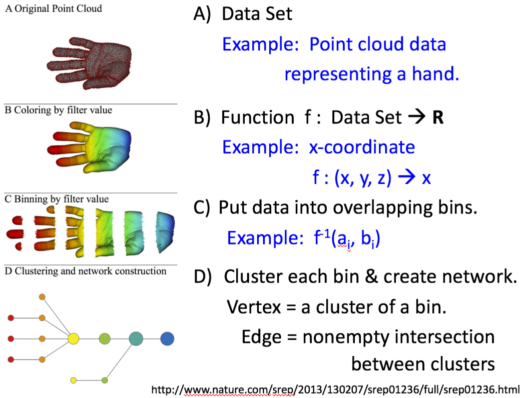

A Original Point Cloud
B Coloring by filter value
C Binning by filter value
D Clustering and network construction

A) Data Set

Example: Point cloud data representing a hand.

B) Function f : Data Set → **R**

Example: x-coordinate

$f : (x, y, z) \to x$

C) Put data into overlapping bins.

Example: $f^{-1}(a_i, b_i)$

D) Cluster each bin & create network.

Vertex = a cluster of a bin.

Edge = nonempty intersection between clusters

http://www.nature.com/srep/2013/130207/srep01236/full/srep01236.html

Figure: Here we follow the standard convention by assigning a specific color to each set in the covering $\mathcal{C}$ and then using the same color for the nodes in the Mapper nerve.

The statistical version of Mapper

We must now describe a method for transporting this construction from the setting of topological spaces to the setting of point clouds. The notion of a covering makes sense in the point cloud setting, as does the definition of coverings of point clouds using maps from the point cloud to a reference metric space, by 'pulling back' a predefined covering of the reference space.

The notion which does not make immediate sense is the notion of constructing connected components of a point cloud. Clustering turns out to be the appropriate analogue. A good example of such a clustering algorithm is the *single linkage clustering*. It is defined by fixing the value of a parameter ϵ, and defining blocks of a partition of our point cloud as the set of equivalence classes under the equivalence relation generated by the relation $\sim_\epsilon$ defined by $x \sim_\epsilon x'$ if and only if $d(x, x') \leq \epsilon$. This way each 'cluster' corresponds to the set of vertices in a single connected component: given any binary relation R on X, the equivalence relation generated by R is the smallest equivalence relation containing R.

The algorithm for generating a statistical Mapper for a data cloud.

- Define a reference map $f\colon X \to Z$, where X is the given point cloud and Z is the reference metric space.
- Select a covering $\mathcal{U}$ of Z.
- If $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$, then construct the subsets $X_\alpha = f^{-1}\mathcal{U}_\alpha$.
- Select a value ϵ as input to the single linkage clustering algorithm above, and construct the set of clusters obtained by applying the single linkage algorithm with parameter value ϵ to the sets X_α. At this point, we have a covering of X parametrized by pairs (α, c), where $\alpha \in A$ and c is one of the clusters of X_α.
- Construct the simplicial complex whose vertex set is the set of all possible such pairs (α, c), and where a family $\{(\alpha_0, c_0), (\alpha_1, c_1), \dots, (\alpha_k, c_k)\}$ spans a k-simplex if and only if the corresponding clusters have a point in common.

This construction is a plausible analogue of the continuous construction described above. We note that it depends on the reference map, a covering of the reference space, and a value for ϵ.

Example

Consider a point cloud data which is sampled from a noisy circle in $\mathbb{R}^2$, and the filter $f(x) = \|x - p\|^2$, where p is the leftmost point in the data.

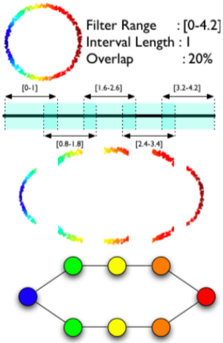


Figure: The vertices are colored by the average filter value.

An important question, of course, is how to generate useful reference maps. If our reference space Z is the Euclidean space $\mathbb{R}^n$ then this means simply generating real valued functions on the point cloud. To emphasize the way in which these functions are being used, we refer to them as *filters* or *filter functions*. Frequently one has interesting filters, defined by a user, which one wants to study. However, in other cases one simply wants to obtain a geometric picture of the point cloud, and it is important to generate filters directly from the metric which may reflect interesting properties of the point cloud. Here are some important examples.

Kernel density estimator. Consider any density estimator applied a point cloud X. It will produce a non-negative function on X, which reflects useful information about the data set. Often, it is exactly the nature of this function which is of interest.

Data depth. The notion of data depth refers to any attempt to quantify the notion of nearness to the center of a data set. It does not necessarily require the existence of an actual center in any particular sense, although a point which minimizes the quantity in question could perhaps be thought of as a choice for a center. Quantities of the form

$$e_p(x) = \frac{1}{\#X} \sum_{x' \in X} d(x, x')^p, \quad x' \in X$$

are referred to as *eccentricity functions*. Other notions could equally well be used. The main point is that the Mapper output based on such functions can identify the qualitative structure of a particular kind.

Eccentricity. This function $\epsilon(x)$ is the maximal distance of another data point from x.

Principal metric SVD filters. Given a matrix of data points (here we really mean Euclidean vectors placed as columns in a matrix) one can apply singular value decomposition in order to obtain the k-th eigenvector of a distance matrix, for example the principal eigenvector corresponds to the largest eigenvalue in magnitude. Projecting data points onto, for example, the principal eigenvector is a way for achieving dimensionality reduction; this projection can serve as a filter function and we can therefore produce a topological summary. Another projection yields a different filter function and therefore possibly a different-looking topological summary compared to the previous one.

Visualizing the Mapper

The dimension of the nerve of the covering of Z determines the dimension of the Mapper complex. The standard choice usually involves intervals in $\mathbb{R}$ with only double overlaps. This forces the 1-dimensional nature of most Mappers you see in applications. It is possible to also visualize the 2-dimensional Mapper obtained from using finitely many rectangles in $\mathbb{R}^2$ with only triple overlaps, similar to the brick wall pattern.

The colors in the Mapper

The colors you see in the Mapper diagram are indicating the values of the chosen filter. Usually the blue end of the spectrum denotes the smaller values and the red end the larger values. There must be other ways to use this ‘extra dimensional’ feature to better advantage.

Unknown stability properties of the Mapper are an obstruction to using faithful measurements in the diagrams. This is in contrast to the stability properties of persistent homology that we saw.

Just a remark for appreciation of the following phenomenon. If one wants to dynamically alter the parameters that build the Mapper, that is fine and creates a movie-like experience with frames corresponding to a smoothly changing parameter. The only variable that is not so well-behaved is the choice of the covering of Z. Even continuous deformations of the covering would usually result in abrupt changes in the Mapper making this not a good explorative tool. There are discontinuous choices that may be made for a relatively consistent experience.

This remark is important for the spirit of TDA. The guiding principle seems to be that instead of committing to a feature or a projection, etc. the recurring idea in TDA is to consider all options at once and learn to explore the moduli space.

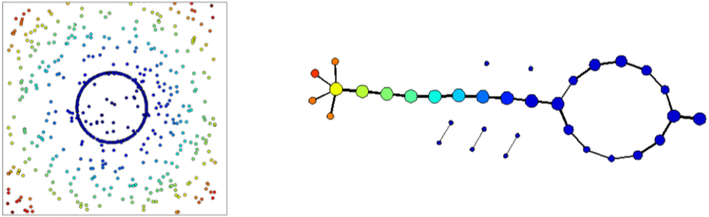


Figure: The diagram produced from a noisy sampled circle by using SVD.

How robust is Mapper?

It's not clear. There are no theorems. There seems to be no reason for it to be robust but under some circumstances it seems to be robust.

How robust is Mapper?

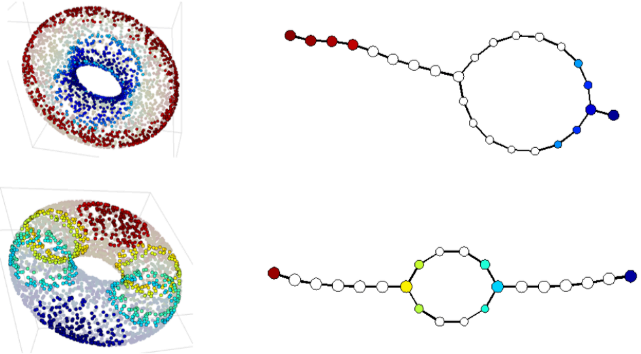


Figure: The summary produced from a sampled torus using SVD with different choices of the projection vectors.

How robust is Mapper?

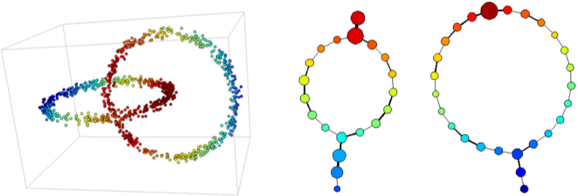


Figure: The summary produced from linked circles recognizes two distinct connected components and their shapes.

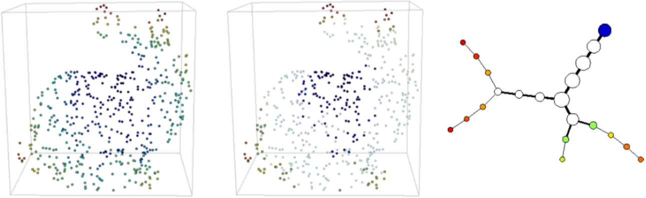


Figure: A really faint (=sparsely) sampled rabbit, but the quality of the Mapper summary is unchanged.

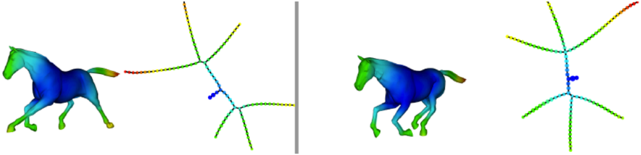


Figure: The integrity of the horse Mapper is preserved throughout the frames of the movement.

Applications of Mapper

G. M. Reaven and R. G. Miller performed a study at Stanford University in the 1970s. 145 patients who had diabetes, a family history of diabetes, who wanted a physical examination, or to participate in a scientific study participated in the study. For each patient, six quantities were measured: age, relative weight, fasting plasma glucose, area under the plasma glucose curve for the three hour glucose tolerance test (OGTT), area under the plasma insulin curve for the (OGTT), and steady state plasma glucose response.

This created a 6 dimensional data set, which was studied using projection pursuit methods, obtaining a projection into three dimensional Euclidean space, under which the data set appears as in the slide. Miller and Reaven noted that the data set consisted of a central core, and two ‘flares emanating from it. The patients in each of the flares were regarded as suffering from essentially different diseases, which correspond to the division of diabetes into the adult onset and juvenile onset forms.

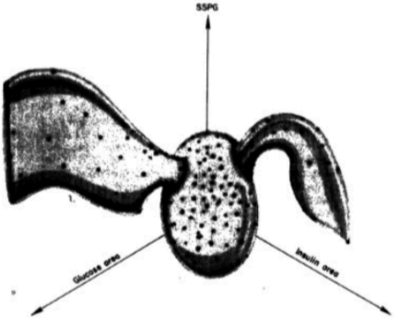


Figure: This is how an artist depicted the dataset in question.

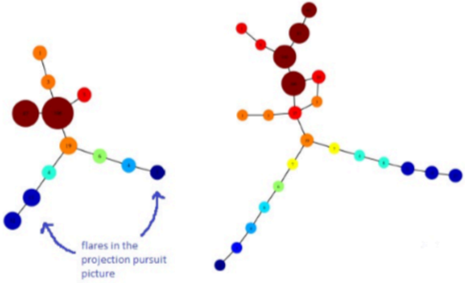


Figure: The diagram produced by the Mapper.

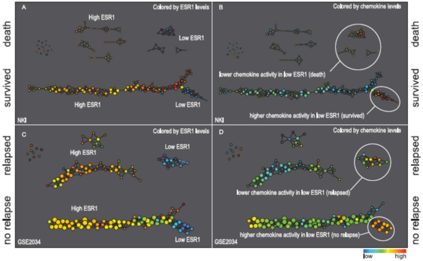
The filter in this case is a density estimator, and high values occur at the dark nodes at the top, while low density values occur on the lower flares. At both scales, there is a central dense core, and two 'flares' consisting of points with low density. The core consists of normal or near-normal patients, and the two flares consist of patients with the two different forms of diabetes.

For one of the most famous examples of the use of mapper so far, see Nikolau, Levine, Carlsson, *Topology based data analysis...* which identifies a subgroup of breast cancers with a unique mutational profile and excellent survival.

Feature selection:

Example:

Data: breast cancer patients that went through specific therapy.


A
Colored by ESR1 levels
High ESR1
Low ESR1
High ESR1
Low ESR1
NKI
B
Colored by chemokine levels
lower chemokine activity in low ESR1 (death)
higher chemokine activity in low ESR1 (survived)
NKI
C
Colored by ESR1 levels
High ESR1
Low ESR1
High ESR1
Low ESR1
GSE2034
D
Colored by chemokine levels
lower chemokine activity in low ESR1 (relapsed)
higher chemokine activity in low ESR1 (no relapse)
GSE2034
death
survived
relapsed
no relapse
death
survived
relapsed
no relapse
low
high

Extracting insights from the shape of complex data using topology, Lum et al., Nature, 2013

f : eccentricity, $N = 30$, $g = 0.33$

Goal: detect variables that influence survival after therapy in breast cancer patients

Figure: An application of the Mapper to feature selection. Cancer patient group with good survival rates can be identified.

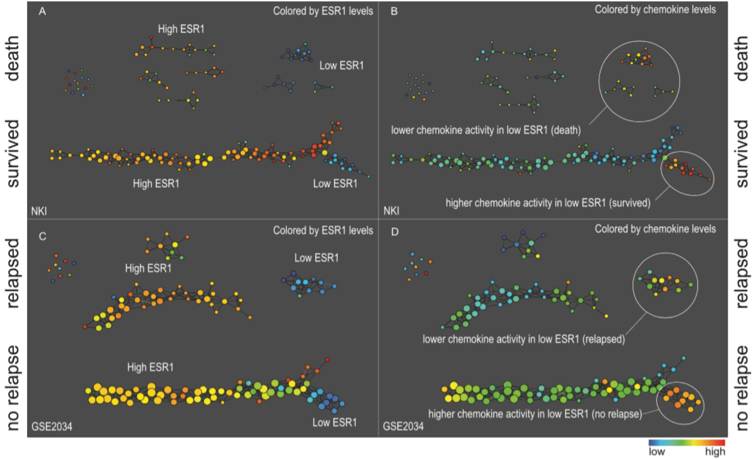

A
Colored by ESR1 levels
High ESR1
Low ESR1
High ESR1
Low ESR1
NKI
B
Colored by chemokine levels
lower chemokine activity in low ESR1 (death)
higher chemokine activity in low ESR1 (survived)
NKI
C
Colored by ESR1 levels
High ESR1
Low ESR1
High ESR1
Low ESR1
GSE2034
D
Colored by chemokine levels
lower chemokine activity in low ESR1 (relapsed)
higher chemokine activity in low ESR1 (no relapse)
GSE2034
death
survived
relapsed
no relapse
low
high

Figure: Better resolution.

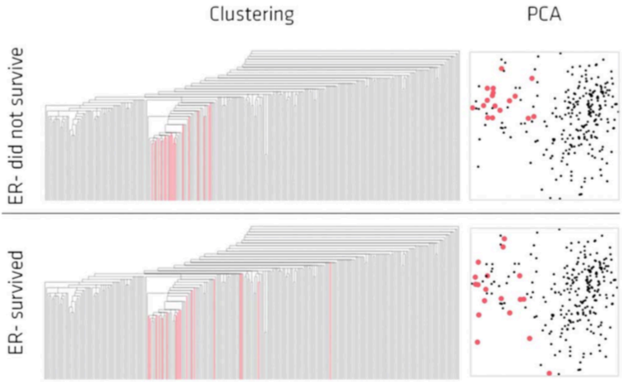


Figure: Classical single linkage hierarchical clustering approaches cannot easily detect these biologically relevant sub-groups because by their nature they end up separating points in the data set that are in fact close..

The following is from Alagappan's classification of NBA players according to 13 "positions".

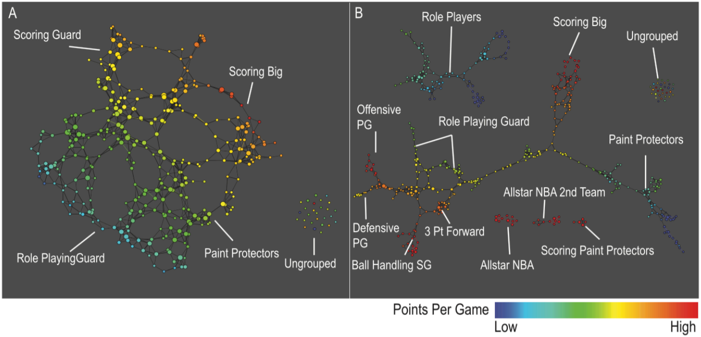


Figure: Here the distinction is in the resolution. On the left 20 intervals were used, on the right 30 intervals for the principal SVD value decomposition.

Applications of Mapper in Machine Learning

The Mapper can be used in conjunction with machine learning for feature selection. This goes through the following stages. (1) Build a Mapper graph/complex from data. This stage of course has a lot of flexibility and available choices. (2) Find interesting structures (loops, flares, distinguished coloring of a group of nodes). This is done by hand unless the structure is a computation such as persistent homology. (3) Select the features/variables that best discriminate the data in these structures.

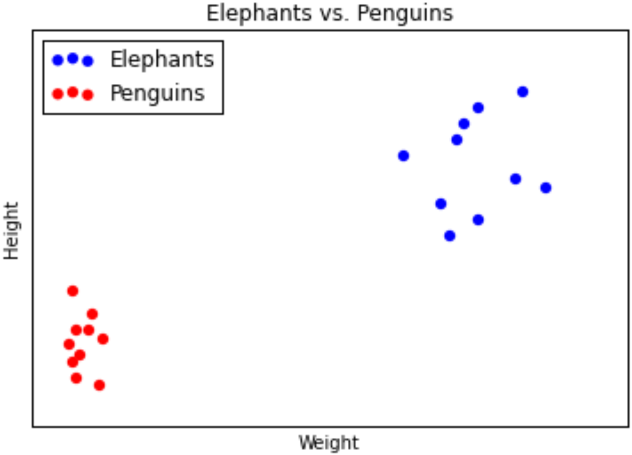
Machine learning pipeline

Supervised learning: the goal is to learn the outcomes of a given process, treated as a black box, so as to be able to predict the outcomes for new inputs.

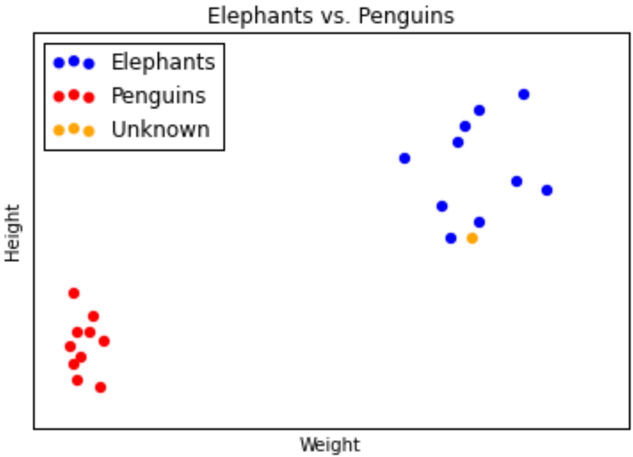
The data set is called the *training set*. The input parameters are *features*. Same as *covariate* in statistics. Persistence diagrams can be used to produce such features. A *model* is a function with undetermined parameters learned from the training set that can now be used to make predictions.

The simplest to describe problem is *classification*. The values of the function are 0 and 1.

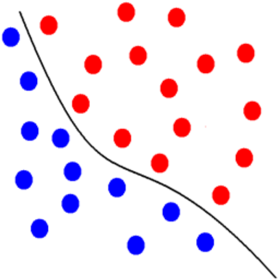
A simple planar data set



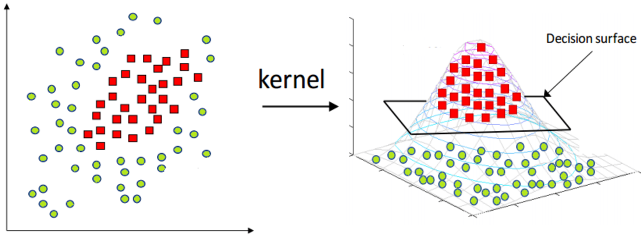
Classification of the unknown animal



Harder classification problem



SVM: the linear method



SVM, PCA, etc. are insufficient or costly in many modern ML applications.

Garbage in $\longrightarrow$ garbage out

A major problem in ML is feature selection and feature generation. Practitioners usually worry about bias in data but it's clear that bias in feature selection is as important.

Example: house pricing. Number of rooms vs number of families with last name Edison living in the neighborhood.