LSSTC DFSP

Unsupervised machine learning

Matthew J. Graham
Center for Data-Driven Discovery, Caltech (and NOAO)
mjg@caltech.edu / graham@noao.edu

January 23, 2017

CENTER FOR DATA-DRIVEN DISCOVERY

What is unsupervised machine learning?

"Inferring a function to describe hidden structure from unlabeled data"

=> letting the data speak for itself

Data may reside in clusters or on a lower dimensional manifold

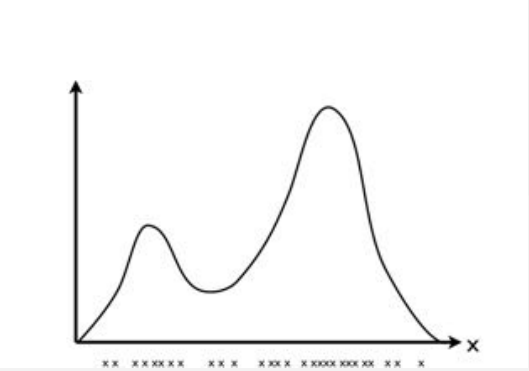
Types of activity:

- Clustering
- Density estimation
- Auto-encoding/deep neural networks
- Dimensionality reduction
- Symbolic regression

…

Density estimation: histograms

- Build a *probability density function* (pdf) from the data



How to choose the bin size/number of bins:

- Assume the underlying distribution is Gaussian, Scott's rule:

$$h = \frac{3.5\sigma}{N^{1/3}}$$

where σ is the sample standard deviation and *N* is the number of data points

- For non-Gaussian distributions, Freedman-Diaconis rule:

$$h = \frac{2IQR}{N^{1/3}}$$

where *IQR* is the interquartile range ($q_{75} - q_{25}$)

Knuth rule and Bayesian blocks

Treat histogram as a piecewise constant model of the underlying density function

- Knuth rule: optimizes a Bayesian fitness function across fixed-width bins

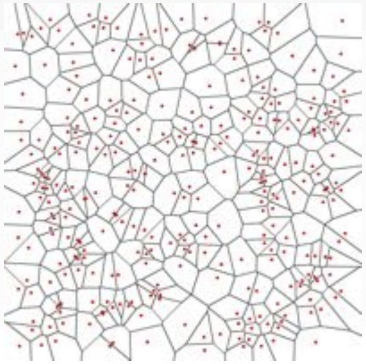
$$N \log M + \log\left[\Gamma\left(\frac{M}{2}\right)\right] - M \log\left[\Gamma\left(\frac{1}{2}\right)\right] - \log\left[\Gamma\left(N + \frac{M}{2}\right)\right] + \sum_{k=1}^{M} \log\left[\Gamma\left(n_k + \frac{1}{2}\right)\right]$$

for M bins with n_k measurements in bin *k*.

- Bayesian blocks: optimizes a Bayesian fitness function across an arbitrary configuration of bins (optimal binning)

$$F(N_i, T_i) = N_i(\log N_i - \log T_i)$$

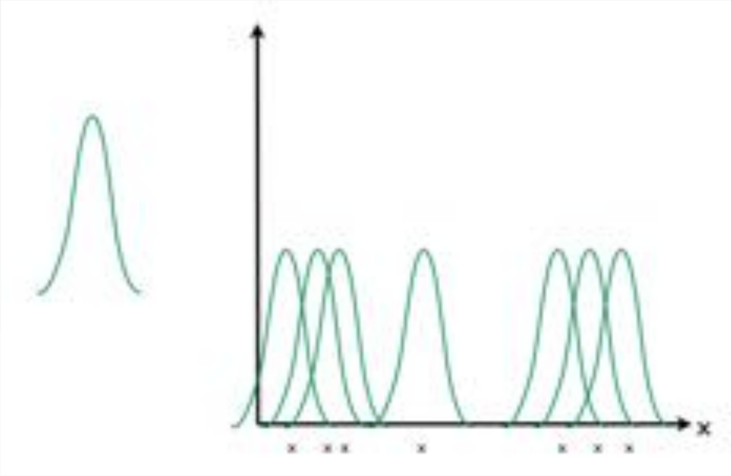
  - For higher dimensions, calculate Voronoi tesselation and then use 1-D Bayesian block rule to identify (non)-contiguous blocks

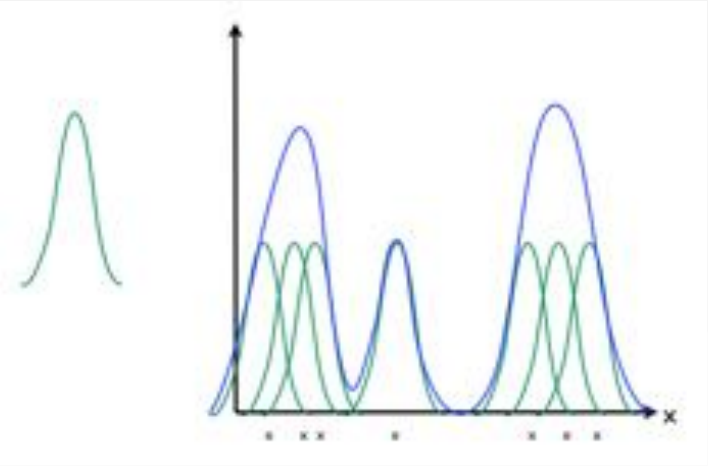


Kernel density estimation

- Nonparametric density estimation
- Each data point is described by a *kernel*
- The pdf is estimated as the sum of the kernels:

$$\hat{f}_h(x) = \frac{1}{n}\sum K_h(x - x_i) = \frac{1}{nh}\sum K\left(\frac{x - x_i}{h}\right)$$





Bandwidth choice

- Minimize the mean integrated squared error:

$$MISE_h = \int (\hat{f}_h(x) - f(x))^2\, dx = \int \hat{f}_h^2\, dx - 2\int \hat{f}_h f(x)\, dx + \int f^2(x)\, dx$$

$$\int \hat{f}_h f(x)\, dx = E_x\left(\hat{f}_h(x)\right)$$

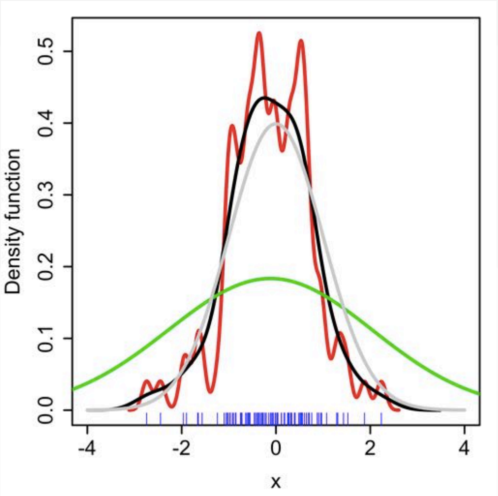
$$\hat{E}_x(\hat{f}_h(x)) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}_{h,-i}(x_i)$$

- Define cross-validation least-square score:

$$CV(h) = \int \hat{f}_h^2(x)\, dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{h,-i}(x_i)$$

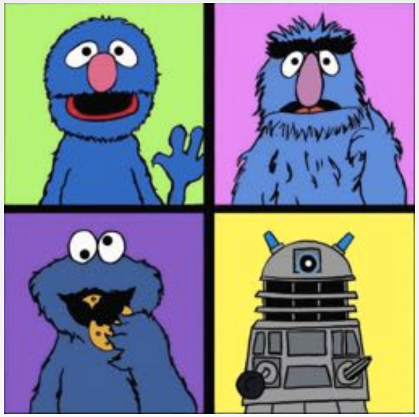
- Optimum value of *h*:

$$\hat{h}_C V = \arg\min_h CV(h)$$



Clustering

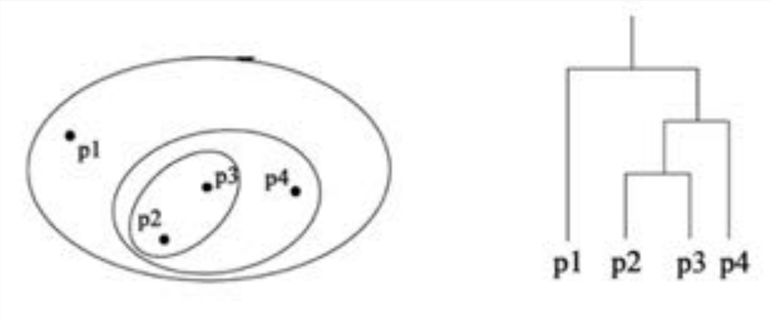
- A *cluster* is a collection of objects which are *similar* between them and are *dissimilar* to the objects belonging to other clusters



Types of clustering:

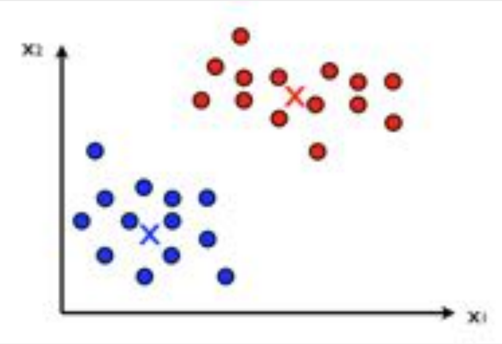
- Partition clustering

- Hierarchical clustering



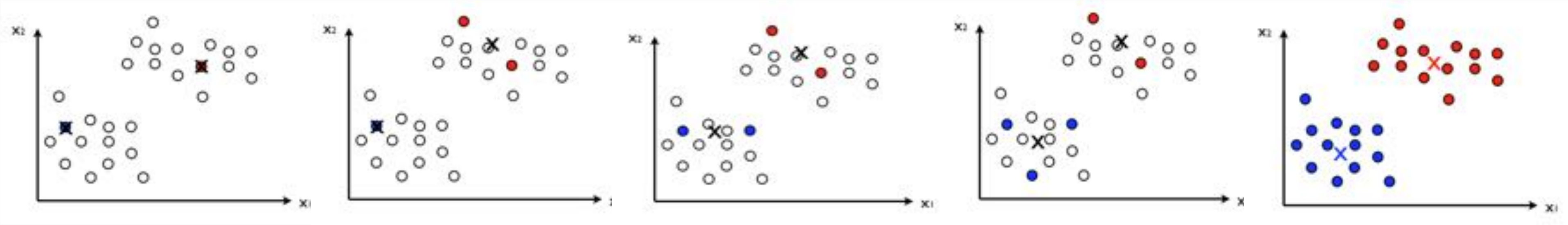
K-means

- Divide objects in *k* clusters
- Each cluster is described by a centroid
- Each object is associated to the closest centroid



<u>How to define the centroids</u>:

- Choose initial centroids (randomly, clusters depend on initial centroids)
- Randomly pick a new object and associate it with nearest centroid
- Centroids are re-defined as the mean of the objects in the cluster
- Convergence after *i* iterations (subject to some measure)

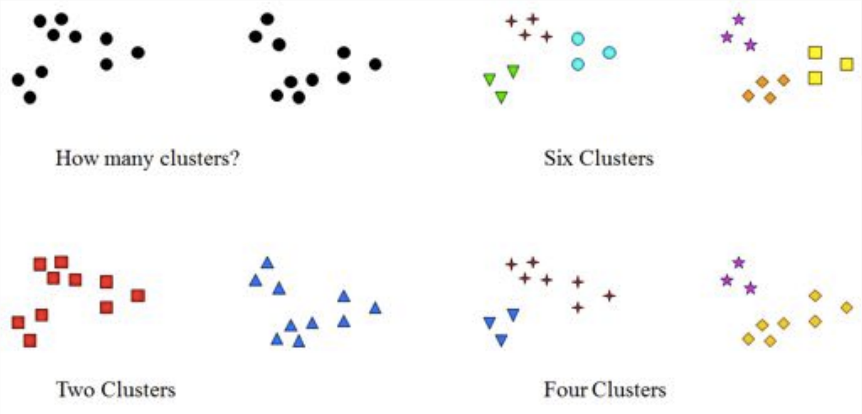


How many clusters?

- If a likelihood function can be defined, use *Akaike information criterion* (AIC), *Bayesian information criterion* (BIC), or *Deviance information criterion* (DIC)
- Mean *silhouette* of data:
  - *a:* mean distance to objects within its cluster (cohesion)
  - *b:* minimum mean distance to objects within other clusters (separation)

$$s = \frac{b-a}{\max(a,b)}$$

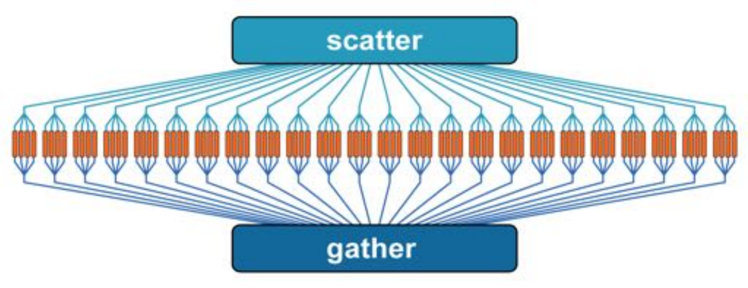
  - Want positive *s* *(a<b)* close to one (small *a*)
- Cross validation: mean value of some objective function over *x* partitions for *k* clusters
- Similarity matrix: eigenvalues/ eigenvectors



Parallelize k-means

Analysis:

- Large amounts of data that do not need to be sent around processors
- Minimum processor intercommunication
- Data set needs to be read for each iteration but each point only needs to be read by one processor



Solution:

- Map:
  - Divide data amongst processors
  - Each processor reads previous iteration's cluster centers and assigns its data to the clusters
  - Each processor then calculates new centers for its data
- Reduce:
  - True cluster centers for this iteration are weighted average of new centers from each processor

Stream k-means

Analysis:

- Data are too large to store on available resources or hold in memory
- Data are not persistent so no later processing possible
- Rough-and-ready results required for data exploration
- Time-dependent results to check convergence, data quality

Solution:

Make initial guesses for the centers $w_1, w_2, \ldots, w_t$

Set the counts $n_1, n_2, \ldots, n_t$ to zero

Loop until interrupted:

Acquire the next example, x

If w_i is closest to x:

Increment n_i

Replace w_i by $w_i + (1/n_i)*(x - w_i)$

Stochasticize k-means

Analysis:

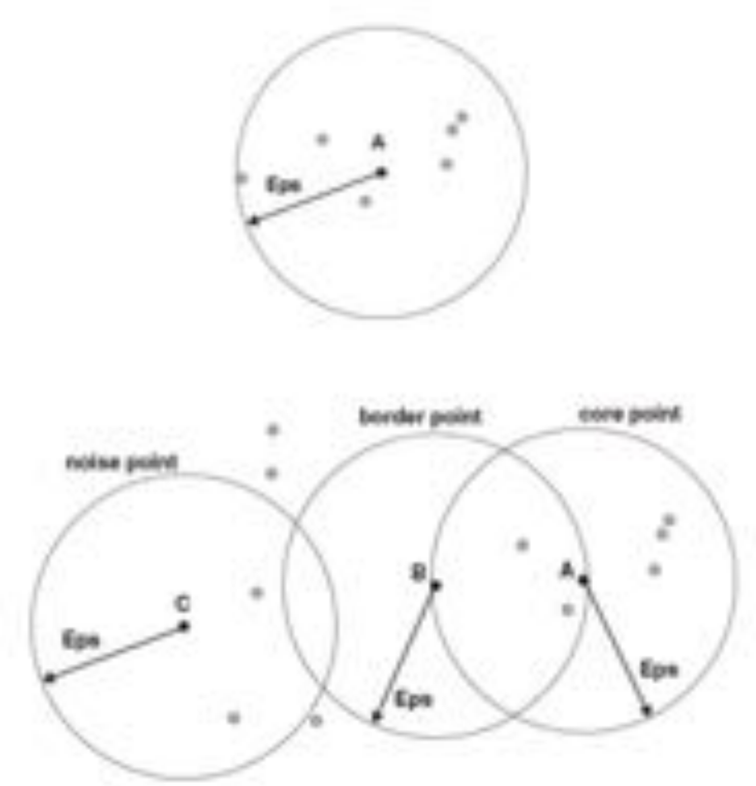
- k-means is prone to local minima and sensitive to initial clusters
- Normally repeat several times
- Stochastic algorithm can reach (global) minimum quicker:
  - (Nominally) works with subset of the data
  - Relative position of clusters found very quickly
  - Terminal convergence slowed down by stochastic noise implied by random choice of points
  - Great learning algorithm but hopeless optimization algorithm

Solution:

- The right choice of learning rate (replace scalar with inverse Hessian of loss) gives much better convergence
- This is just the online version

DBSCAN

- Center-based density: number of points with a specified radius, eps
- Classification of points according to this density:
  - Core point (CP): at least minpts within an eps radius
  - Border point (BP): not a CP but in the neighborhood of a CP
  - Noise point (NS): neither of the above

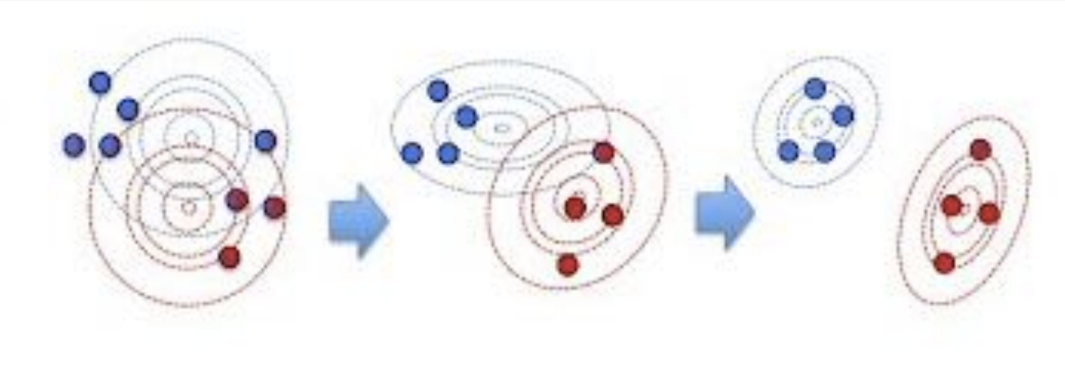


<u>How to pick eps and minpts</u>:

- minpts ≥ D+1
- Plot the distance to k=minpts nearest neighbor – where plot shows a sharp bend indicates noise

Gaussian mixture models

- Assumes all data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters
- Nonparametric density estimation
- GMMs can be thought of as a generalization which incorporates information about the covariance structure of the data as well as the centers of the latent Gaussians
- Number of clusters can be considered a fine tuning parameter and selected via BIC or assume Dirichlet process prior



Hierarchical clustering

- Each point starts its own cluster and pairs of clusters are merged as one moves up the hierarchy

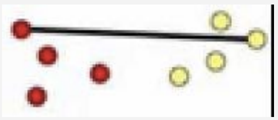
<u>Linkage criteria</u>:

- Single link ("friends-of-friends")

$$D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

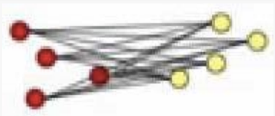
- Complete link

$$D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$



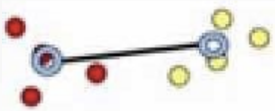
- Average link

$$D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$$



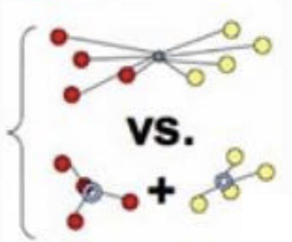
- Centroids

$$D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|} \sum_{x \in c_1} \vec{x}\right), \left(\frac{1}{|c_2|} \sum_{x \in c_2} \vec{x}\right)\right)$$



- Ward's method

$$TD_{c_1 \cup c_2} = \sum_{x \in c_1 \cup c_2} D(x, \mu_{c_1 \cup c_2})^2$$



Similarity metrics

- The most common similarity metric is Euclidean distance
- Manhattan distance:

$$D(x,y) = \sum_i |x_i - y_i|$$

- Minkowski distance:

$$D(x,y,p) = \left(\sum_i |x_i - y_i|^p \right)^{1/p}$$

- Mahalanobis distance:

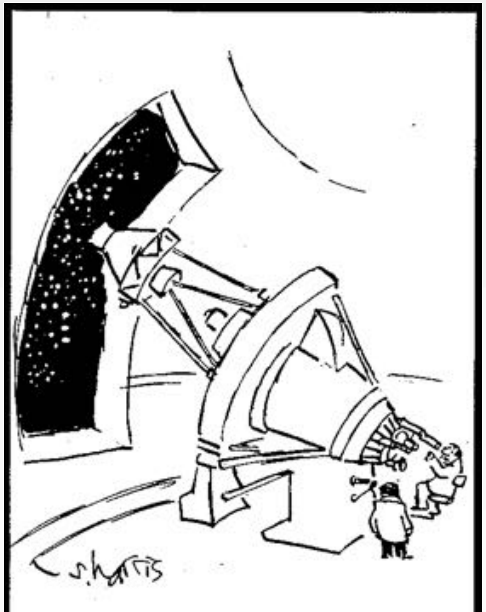
$$D(x,y) = \sqrt{(x-y)^T S^{-1} (x-y)}$$

- Cosine distance:

$$D(x,y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$

- Jaccard distance:

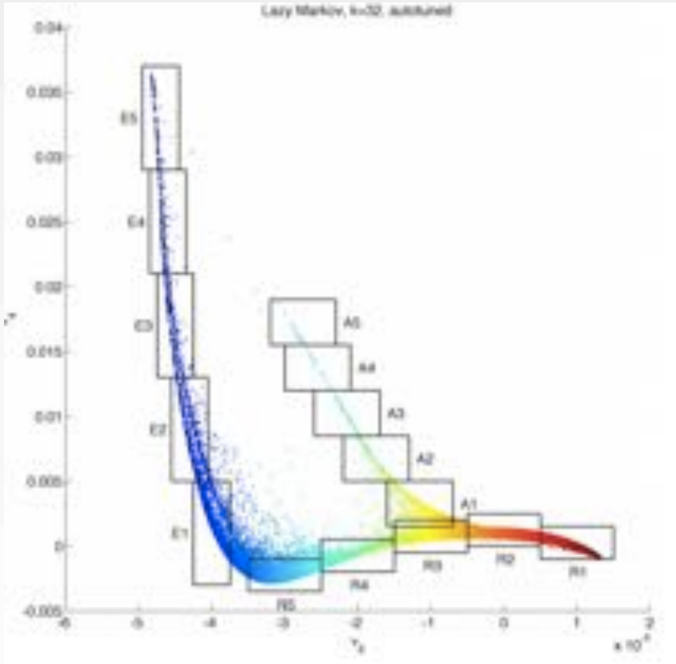
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$



"Actually they all look alike to me."

What are the limits to clustering?

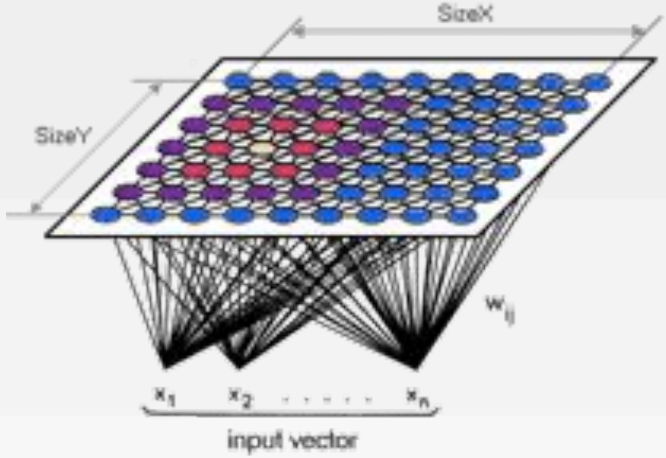
- Clustering is *obvious* when dealing with a feature set (0 < D < 100)
- What about higher dimensions?
  - Spectra: Locally-biased semi-supervised eigenvectors
  - Time series: Dynamic time warping
  - Images: Convolutional neural networks
  - Documents: Latent Dirichlet allocation

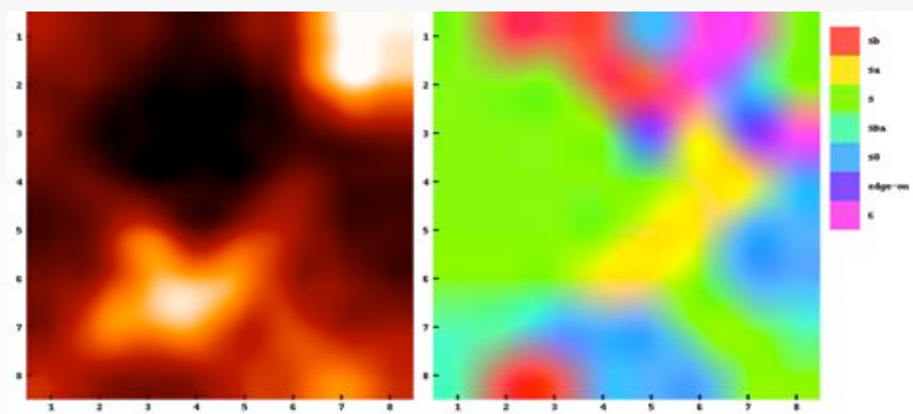


- Very high dimensions are sparse: clusters occupy increasingly small volumes

Self-organizing maps (SOMs)

- Type of artificial neural network trained to produce a low-dimensional representation of the input space of the training samples

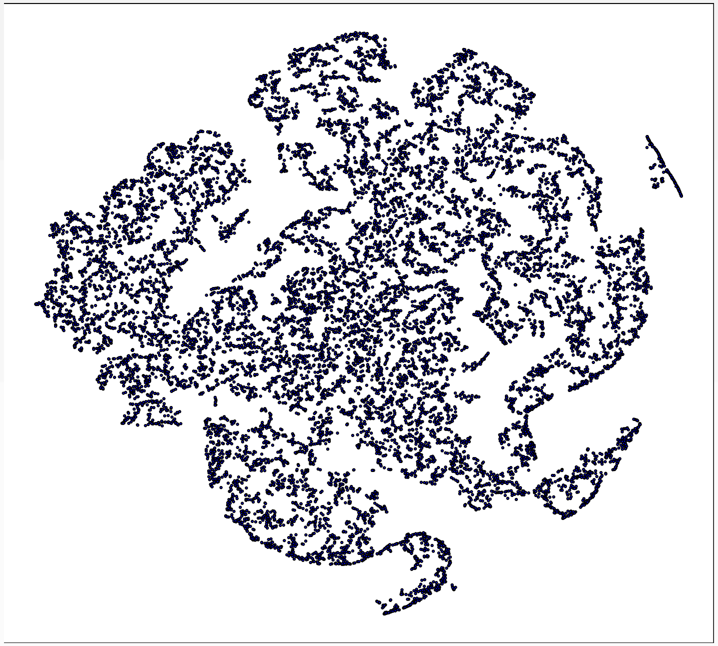


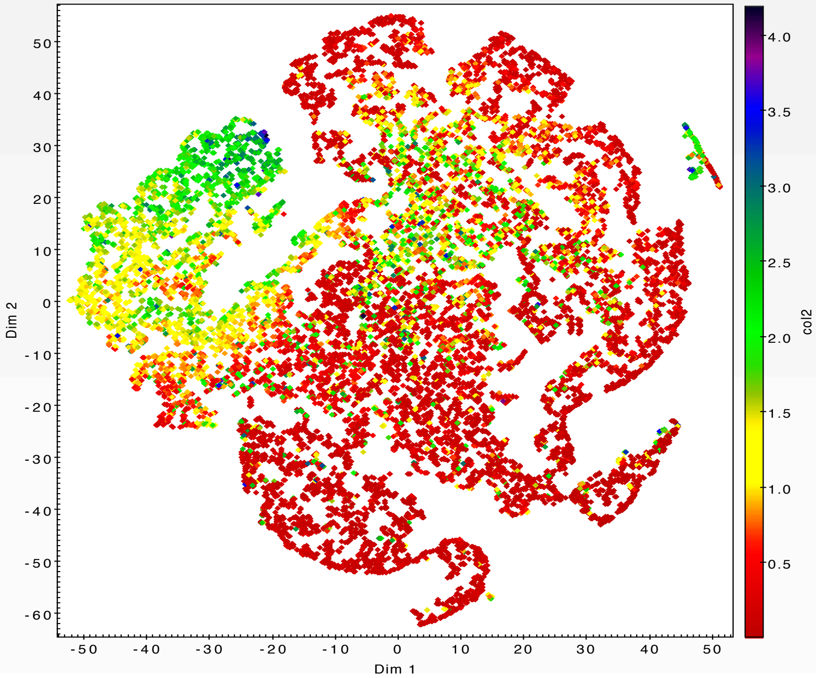


- Carrasco Kind & Brunner (2014) apply it to photometric redshifts

t-distributed stochastic neighbor embedding (TSNE)

- State-of-the-art dimensional reduction technique:
  - Probability distribution, *P*, over pairs of high-dimensional objects in data identifying similar and dissimilar objects
  - Probability distribution, *Q*, over low dimensional map similarly
  - Minimize information theoretic constraint (Kullback-Leibler divergence) of *Q* from *P*

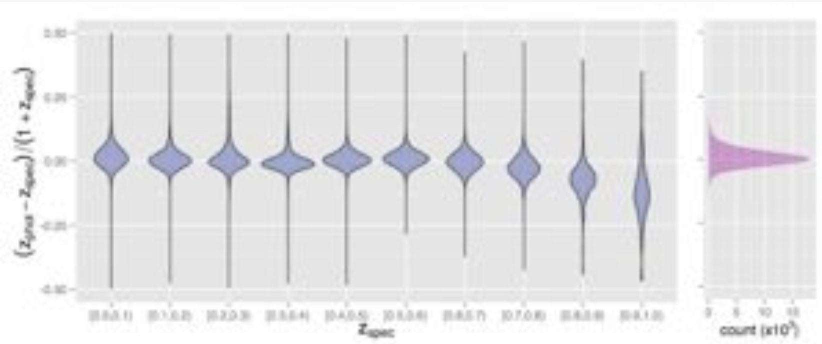




Symbolic regression

- Find a function, in symbolic form, that fits the data
- Specific type of building block to be used in fit:
  - Algebraic operators, analytical functions (trig, exp/log, power), constant, Boolean, switching, squashing, state variables
- Evolutionary algorithm explores a metric space constructed from numerical partial derivatives of pairs of variables in data set looking for best match to predicted candidate function
- Produces a small set of final candidate analytical expressions on accuracy-parsimony Pareto front
- Krone-Martins, Ishida & Souza (2013):

$$z_{phot} = \frac{0.4436r - 8.261}{24.4 + (g-r)^2 (g-i)^2 (r-i)^2 - g} + 0.5152(r-i)$$



The endpoint of unsupervised learning