# Chapter 6

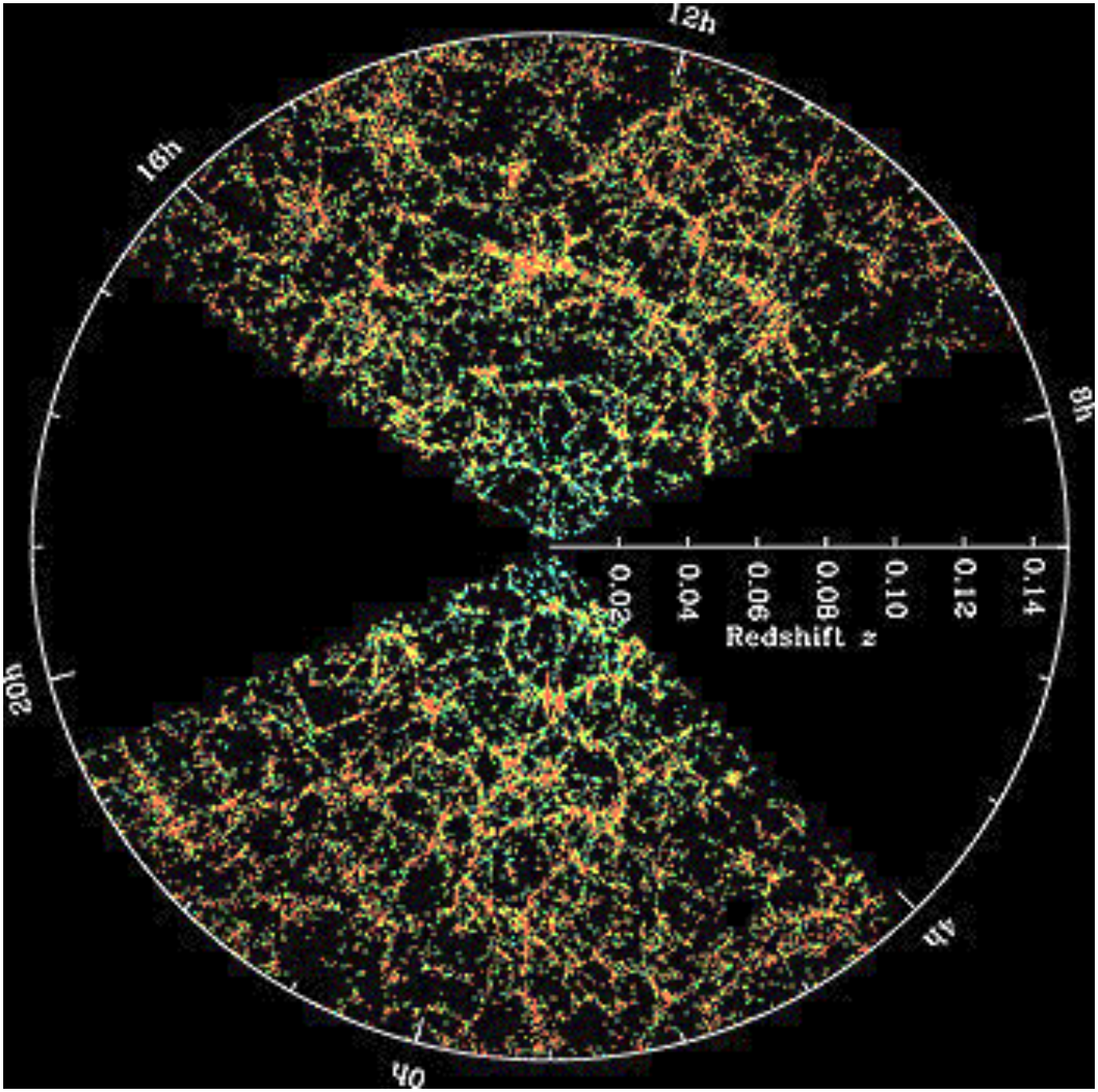
Structure in point data

Robert LIndner

## Point data:

SDSS Galaxy Redshifts

(ra, dec, *z*)



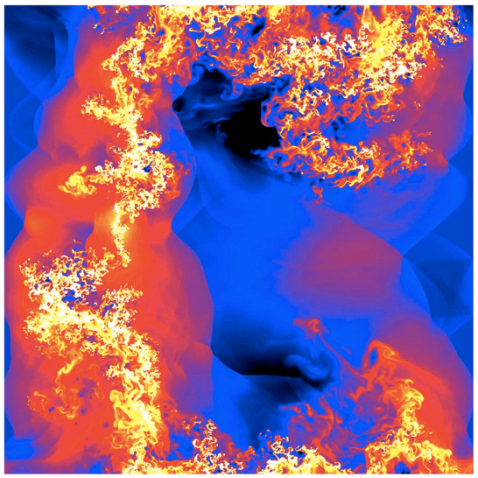
More point
Data.
(X, Y, Z, t)

**Anything that can be organized into a catalog is point data.**

| ID | Source Name | $S_\nu^{\rm Best}$ (mJy) | $S_\nu^{\rm Full}$ (mJy) | $S_\nu^{\rm Deboosted}$ (mJy) | $P(<0$ |
|---|---|---|---|---|---|
| 1 | MM J104700.1+590109 | $3.7 \pm 0.8$ | $4.1 \pm 0.6$ | $3.5^{+0.6}_{-0.6}$ | <0.0 |
| 2 | MM J104627.1+590546 | $4.5 \pm 0.8$ | $4.7 \pm 0.7$ | $3.8^{+0.7}_{-0.7}$ | <0.0 |
| 3 | MM J104631.4+585056 | $6.1 \pm 1.8$ | $4.7 \pm 0.7$ | $3.8^{+0.8}_{-0.7}$ | <0.0 |
| 4 | MM J104607.4+585413 | $2.8 \pm 0.8$ | $3.2 \pm 0.5$ | $2.7^{+0.5}_{-0.5}$ | <0.0 |
| 5 | MM J104725.2+590339 | $4.9 \pm 0.9$ | $5.2 \pm 0.8$ | $4.0^{+0.8}_{-0.9}$ | <0.0 |
| 6 | MM J104638.4+585613 | $3.1 \pm 0.7$ | $2.7 \pm 0.5$ | $2.3^{+0.4}_{-0.4}$ | <0.0 |
| 7 | MM J104700.1+585439 | $2.8 \pm 0.7$ | $2.8 \pm 0.5$ | $2.3^{+0.4}_{-0.5}$ | <0.0 |
| 8 | MM J104633.1+585159 | $4.5 \pm 1.2$ | $3.4 \pm 0.6$ | $2.7^{+0.6}_{-0.6}$ | <0.0 |
| 9 | MM J104704.9+585008 | $5.6 \pm 1.5$ | $5.1 \pm 0.9$ | $3.8^{+1.0}_{-0.9}$ | <0.0 |
| 10 | MM J104622.9+585933 | $3.6 \pm 0.7$ | $2.9 \pm 0.5$ | $2.4^{+0.5}_{-0.5}$ | <0.0 |
| 11 | MM J104556.5+585317 | $3.5 \pm 0.9$ | $3.4 \pm 0.6$ | $2.7^{+0.6}_{-0.6}$ | <0.0 |
| 12 | MM J104448.0+590036 | $5.1 \pm 0.9$ | $3.5 \pm 0.6$ | $2.7^{+0.6}_{-0.7}$ | <0.0 |
| 13 | MM J104609.0+585826 | $2.7 \pm 0.7$ | $2.7 \pm 0.5$ | $2.1^{+0.5}_{-0.5}$ | <0.0 |
| 14 | MM J104636.1+590749 | $4.3 \pm 0.8$ | $4.3 \pm 0.8$ | $3.0^{+0.9}_{-0.9}$ | <0.0 |
| 15 | MM J104738.3+585213 | $5.1 \pm 1.1$ | $4.5 \pm 0.9$ | $3.0^{+0.9}$ | |

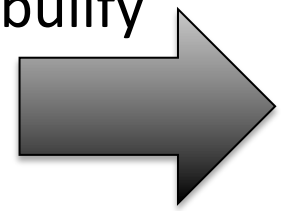
# Probably all data is point data

- Images, time streams, audio recordings can all be recast in tabular form as a function of timestamp, pixel coordinate, frequency, ….



A&H2007

Tabulify



Xpix, Ypix, Value

| | | | | | |
|---|---|---|---|---|---|
| 2 | MM J104627.1+590546 | $4.5 \pm 0.8$ | $4.7 \pm 0.7$ | $3.8^{+0.7}_{-0.7}$ | <0.0 |
| 3 | MM J104631.4+585056 | $6.1 \pm 1.8$ | $4.7 \pm 0.7$ | $3.8^{+0.8}_{-0.7}$ | <0.0 |
| 4 | MM J104607.4+585413 | $2.8 \pm 0.8$ | $3.2 \pm 0.5$ | $2.7^{+0.5}_{-0.5}$ | <0.0 |
| 5 | MM J104725.2+590339 | $4.9 \pm 0.9$ | $5.2 \pm 0.8$ | $4.0^{+0.8}_{-0.9}$ | <0.0 |
| 6 | MM J104638.4+585613 | $3.1 \pm 0.7$ | $2.7 \pm 0.5$ | $2.3^{+0.4}_{-0.4}$ | <0.0 |
| 7 | MM J104700.1+585439 | $2.8 \pm 0.7$ | $2.8 \pm 0.5$ | $2.3^{+0.4}_{-0.5}$ | <0.0 |
| 8 | MM J104633.1+585159 | $4.5 \pm 1.2$ | $3.4 \pm 0.6$ | $2.7^{+0.6}_{-0.6}$ | <0.0 |
| 9 | MM J104704.9+585008 | $5.6 \pm 1.5$ | $5.1 \pm 0.9$ | $3.8^{+1.0}_{-0.9}$ | <0.0 |
| 10 | MM J104622.9+585933 | $3.6 \pm 0.7$ | $2.9 \pm 0.5$ | $2.4^{+0.5}_{-0.5}$ | <0.0 |
| 11 | MM J104556.5+585317 | $3.5 \pm 0.9$ | $3.4 \pm 0.6$ | $2.7^{+0.6}_{-0.6}$ | <0.0 |
| 12 | MM J104448.0+590036 | $5.1 \pm 0.9$ | $3.5 \pm 0.6$ | $2.7^{+0.6}_{-0.7}$ | <0.0 |
| 13 | MM J104609.0+585826 | $2.7 \pm 0.7$ | $2.7 \pm 0.5$ | $2.1^{+0.5}_{-0.5}$ | <0.0 |
| 14 | MM J104636.1+590749 | $4.3 \pm 0.8$ | $4.3 \pm 0.8$ | $3.0^{+0.9}_{-0.9}$ | <0.0 |

# Common tasks for point data

- **Density estimation**

  "What PDF describes my data?"

- **Cluster finding**

  "Are there interesting subgroups or structures in my data?"

- Statistical description

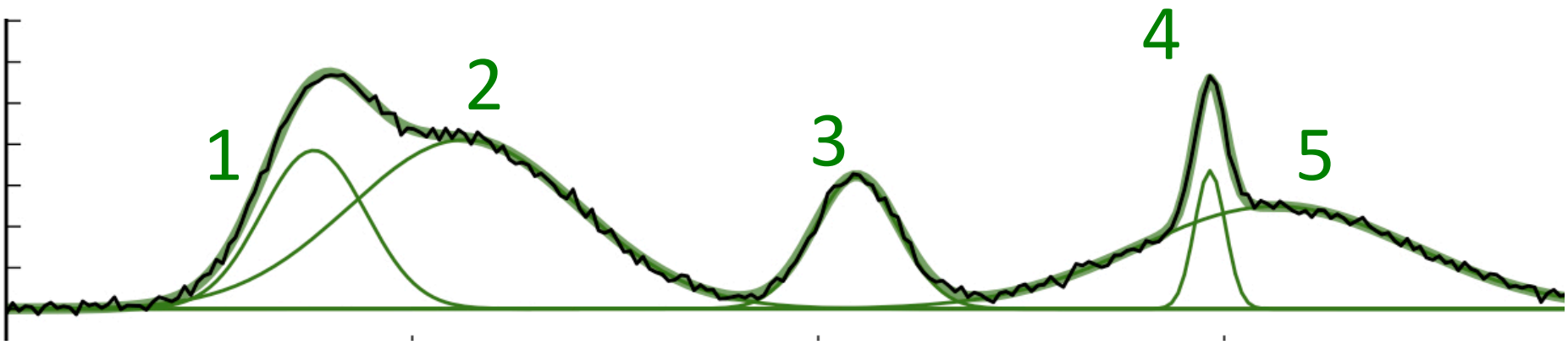
  "How can I compare my data in a statistical way to other datasets?"

# Density estimation

- Parametric
  - "Fit these data using these parameters"
  - Must specify fitting function. E.g., Gaussian, binomial, Poisson, …
- Non-parametric
  - "Fit these data *however you can*"
  - Specify (almost) nothing.
  - More computationally expensive,
  - Less physical insight

# Parametric density estimation

- Fit your favorite model
- Gaussian Mixture Model
  (Covered by Karen Lewis)



# Non-parametric density estimation

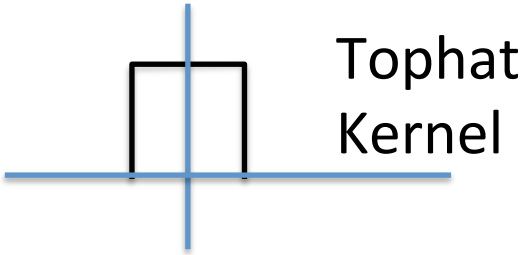
- Histograms (previously discussed)
  - Arbitrary choice of edges. Results could change when edges change.
  - Original data can be "thrown away", as histogram contains all information. This is good.
- **Kernel Density Estimators (KDE)**
  - No arbitrary edge choices. Specification of kernel uniquely specifies the distribution.
  - Good for data containing interesting things of a given scale
  - Original data need to be "carried around" and used every time the density function is to be estimated.
- Nearest Neighbor Density estimator
  - KDE benefits + can find structures on any scale.

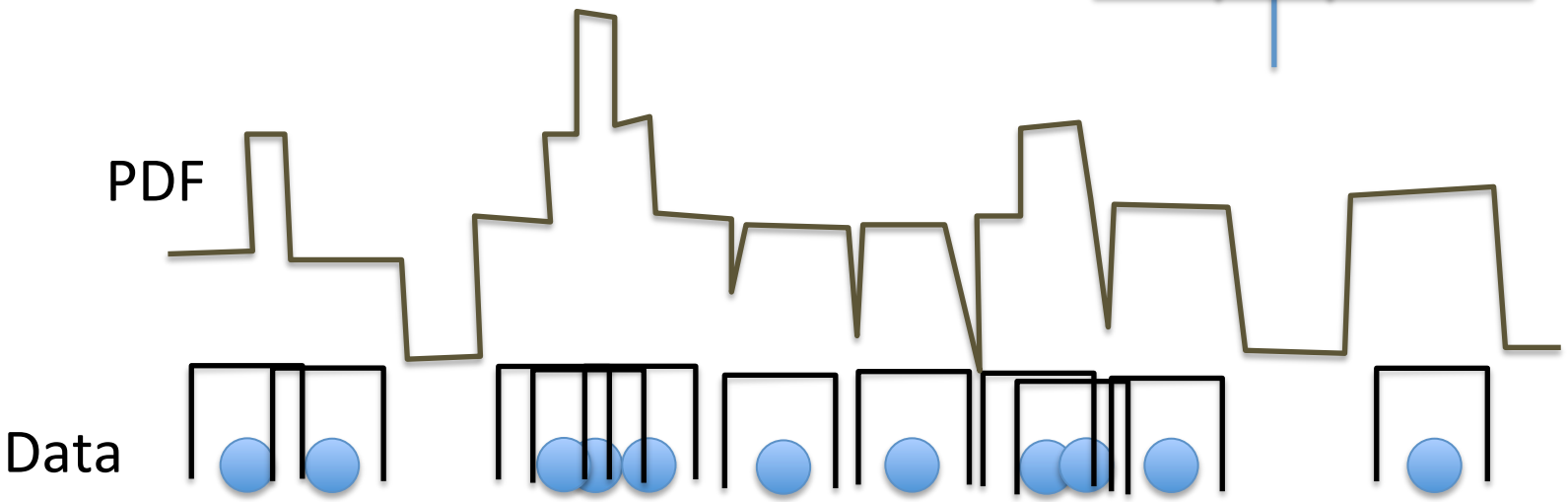
# Kernel Density estimators

$$\text{estimator}(x) = \frac{1}{Nh} \sum_{x'} K\left(d(x - x')/h\right)$$

$\text{kernel K}$ = All values positive, unity normalization, zero mean

Mathematically similar to convolution.



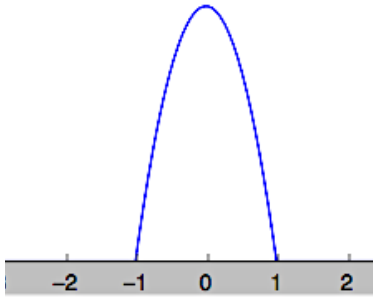
# Common kernels, could use anything satisfying

= All values positive, unity normalization, zero mean

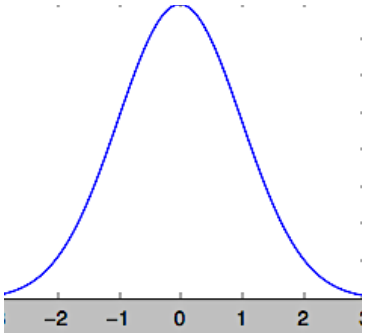
Epanechnikov kernel

$$\frac{3}{4}\left(x^2-1\right) \qquad -1<x<1$$

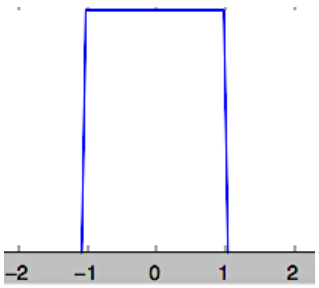
$$0 \qquad \text{Otherwise}$$




Gaussian kernel




Top hat kernel



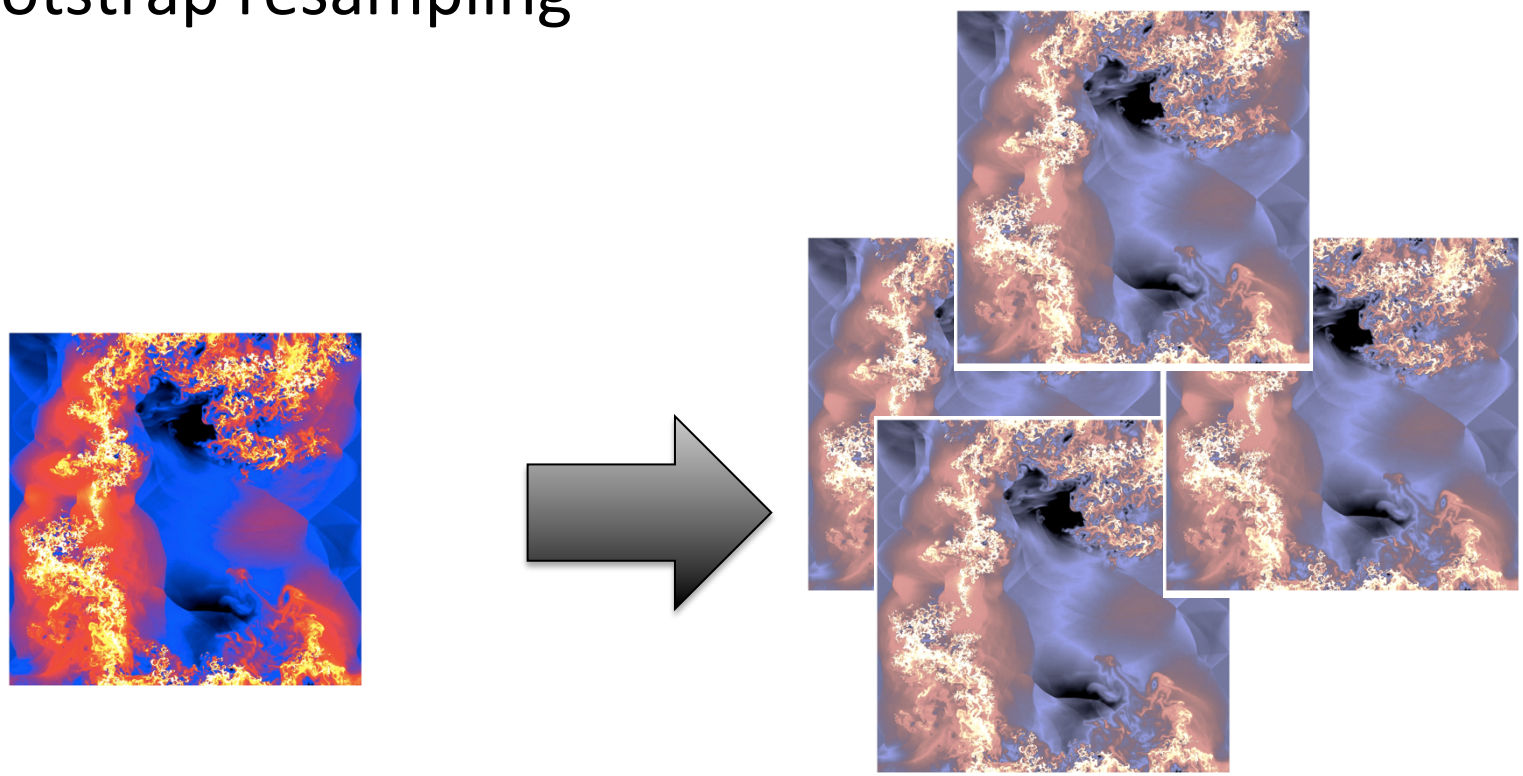
# How to choose optimal scale *h*?

- Cross validation
  - Optimize the model hyper parameters on an independent (out of sample) set of "cross validation" data
    - E.g., number of Gaussians in Gaussian mixture model, the shape of kernel in kernel density estimator, architecture of neural network, number of trees in random forest

i.e., "Guess and check"…

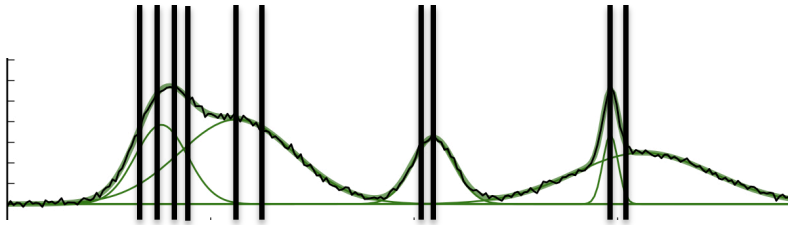
# How to choose optimal scale *h*?

- How to obtain cross validation data
  - Leave-one-out cross validation
  - Bootstrap resampling

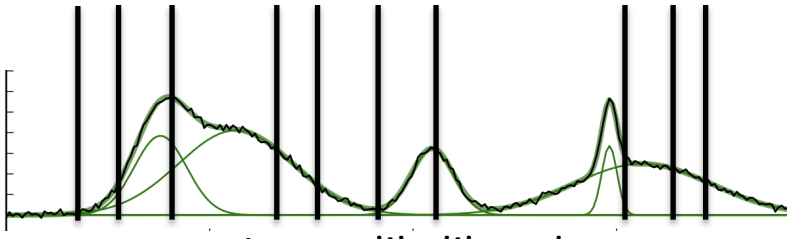


# How to choose optimal scale *h*?

- Choose the parameter *h* which produces the **maximum likelihood** of generating the given data, $\mathcal{L} = \prod_i \hat{f}(x_i)$ .
- Equivalently, minimize $-\sum_i \log \hat{f}_h^{CV}(x_i)$ as a function of *h*.



High likelihood



Lower likelihood

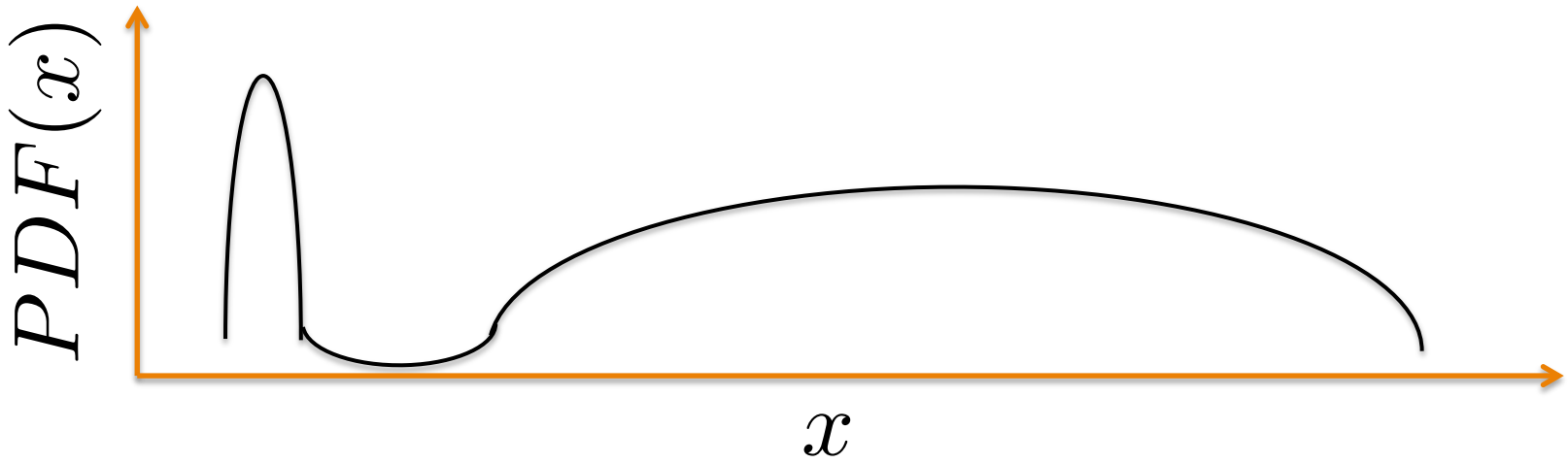
# How to choose optimal scale *h*?

- Using only the Maximum Likelihood condition will produce over-fit results without tons of data
- Can use ensemble technique to reduce over-fitting (see coding example).
  - Solve for *h* which maximizes Likelihood over *many* bootstrapped datasets simultaneously.

# Computer Time

# Problem with Kernel technique of density estimation

- We are stuck with only *one* value of the kernel width!
- Not flexible to allow for a varying spatial scale. E.g., what about trying to model this PDF:

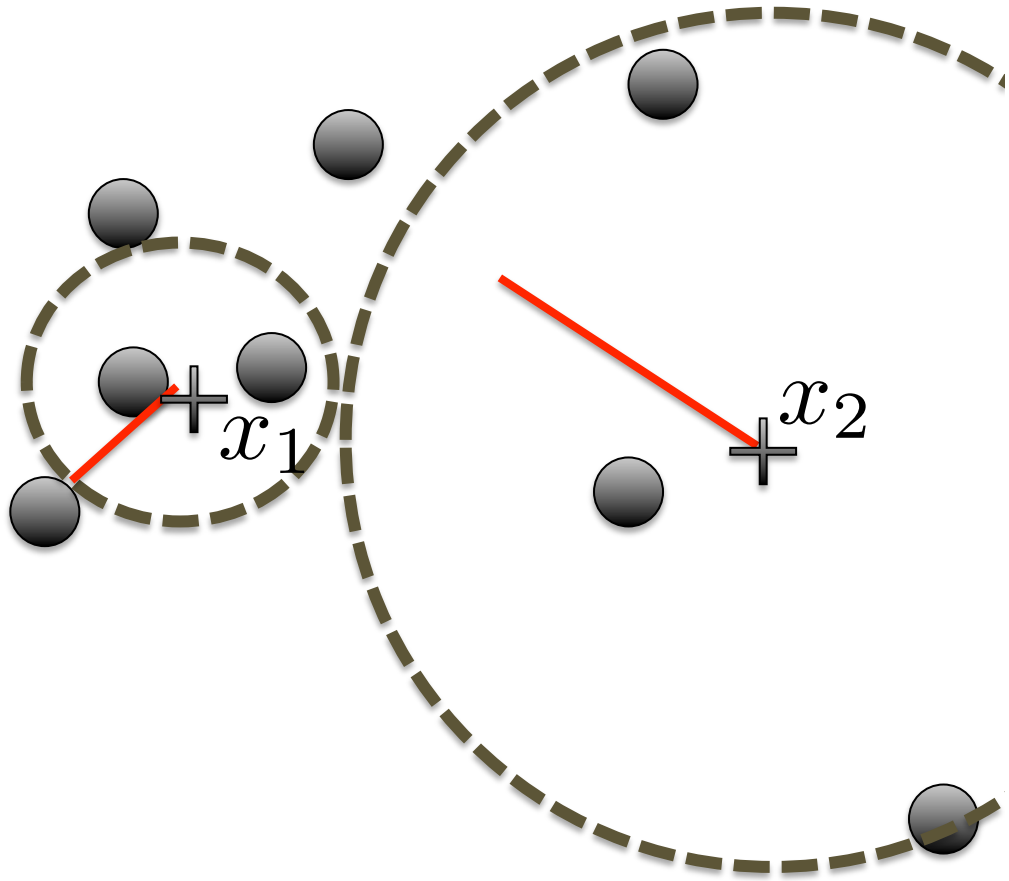


# Nearest Neighbor Density Estimation

$$\hat{f}_k(x) = \frac{K}{V_D(d_k)}$$

$$\vec{f}_3(x_2) = \frac{K}{\pi d_3^2(x_2)}$$

$$\vec{f}_3(x_1) = \frac{K}{\pi d_3^2(x_1)}$$

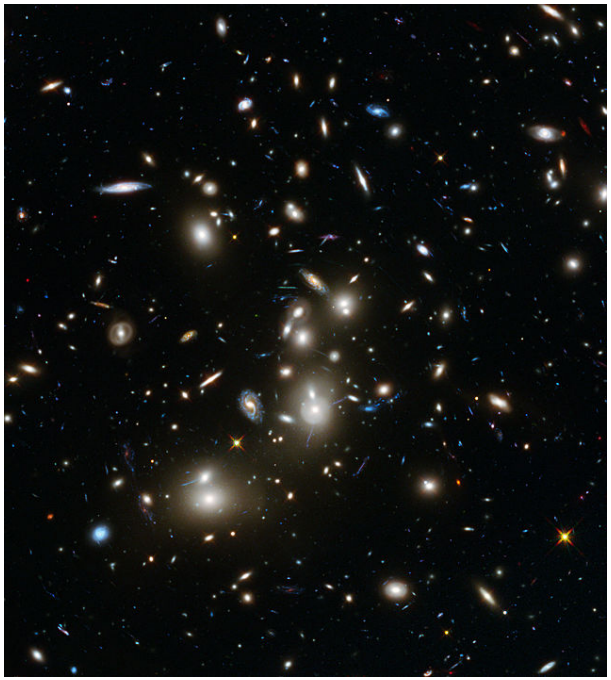


# Nearest Neighbor (NN) Density Estimation

- NN technique has power to capture variation on any scale in the data
- Both nearest neighbor and KDE suffer from "data baggage"
- Both also suffer from potentially $N^2$ calculations, and worse for cross validation
  - (oct,quad,…)trees can help the algorithmic speed (see Bob Benjamin's slides on trees)

# Clustering

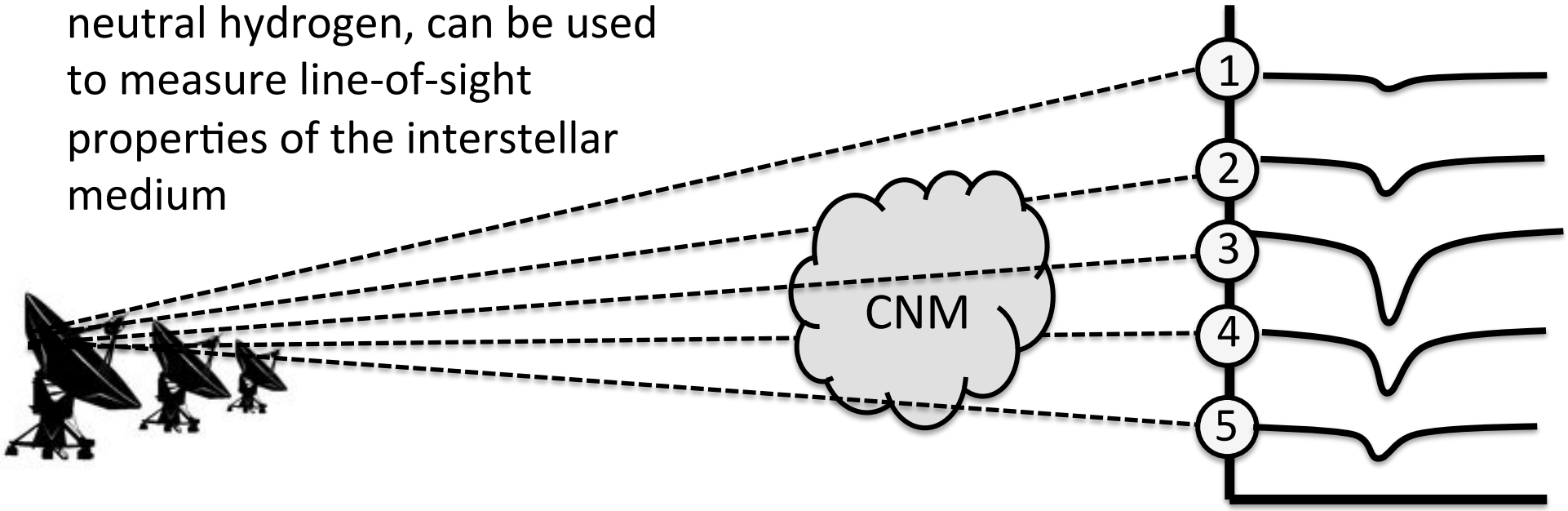
- Unsupervised learning
  - No training data
- Searching for “structures” in point data
- Used often as data exploration tool to discover groupings of data with similar properties



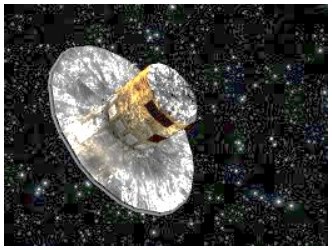
Galaxy cluster

# Clustering can be done any any data: Searching for distinct ISM clouds

21cm observations detect neutral hydrogen, can be used to measure line-of-sight properties of the interstellar medium

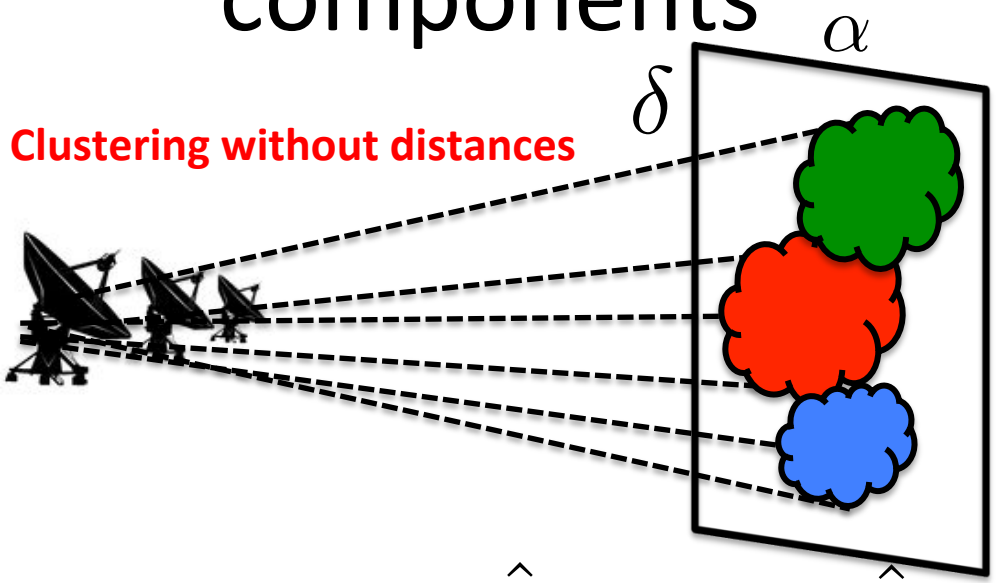



- Begin with 5D space of ra, dec, amplitude, width, and velocity.
- Search for clusters of points to find physics associations (clouds)



Gaia spacecraft

# Finding clouds from Gaussian components




- Distance is critical for ISM physics

  Cloud size: $L \propto D$

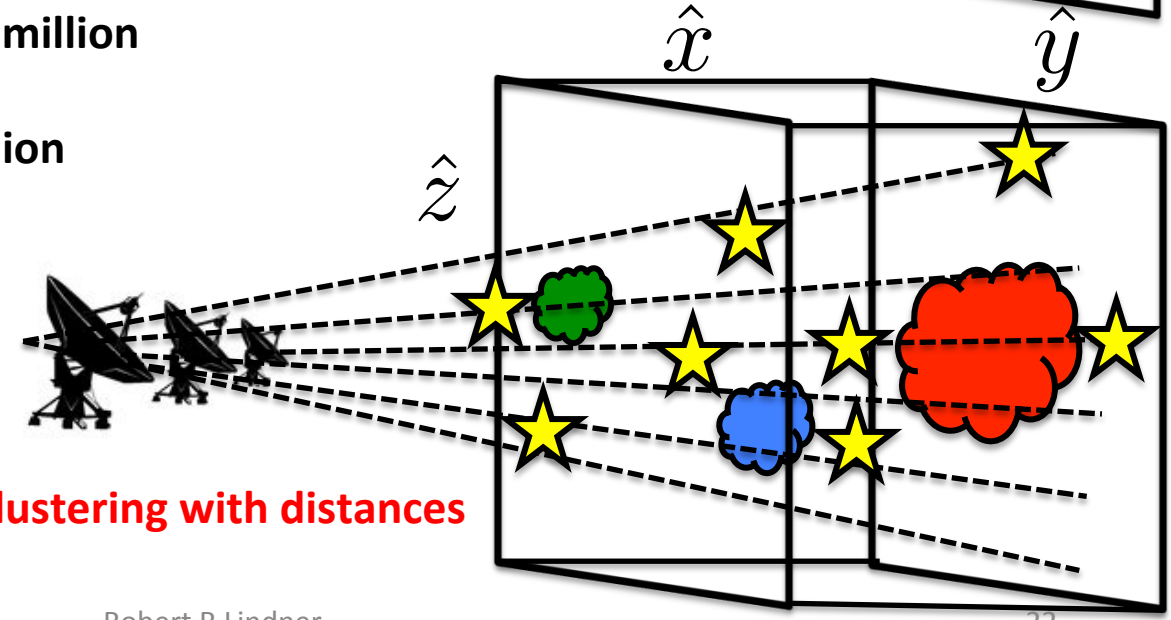
  Cloud mass: $M \propto D^2$

- HI Spectra in GALFA-HI survey: **3 million**

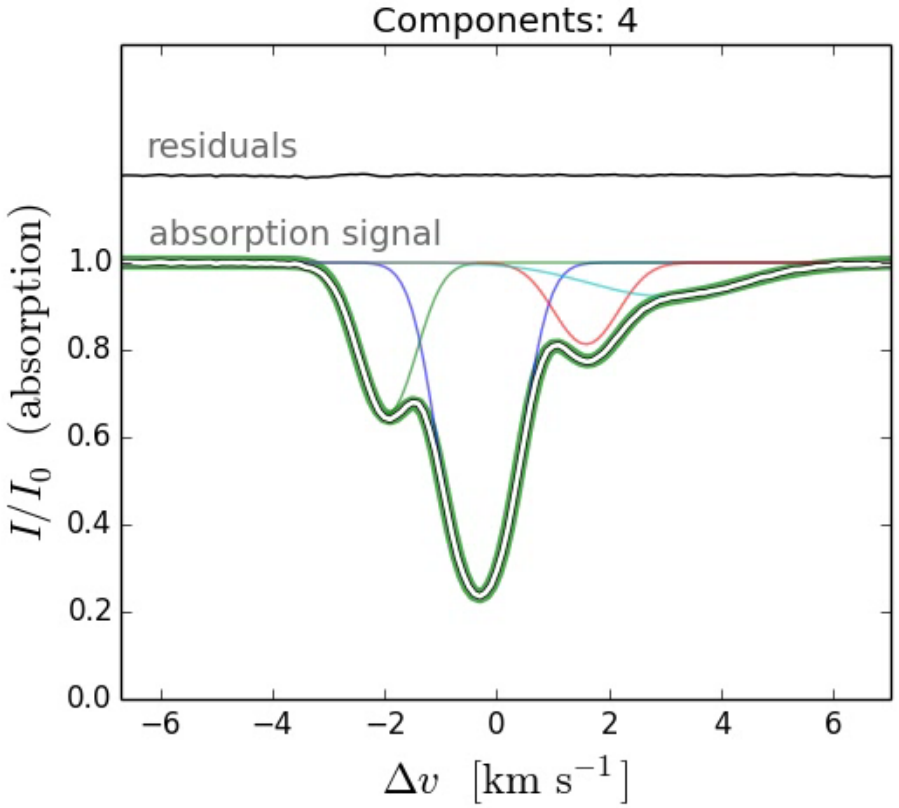
  **+**

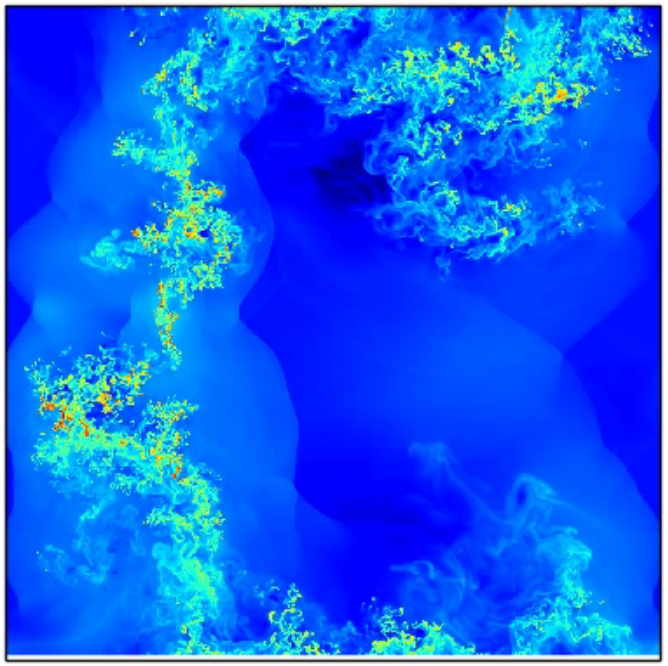
- Stellar distances from Gaia: **1 billion**

= 3D Atlas of neural gas in the Milky Way



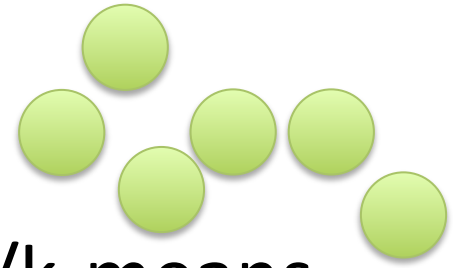
# Example: data generation = extraction of Gaussian components

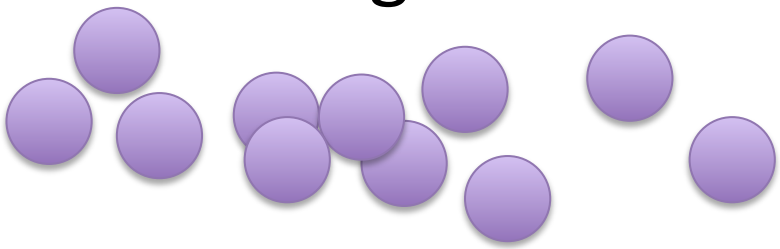


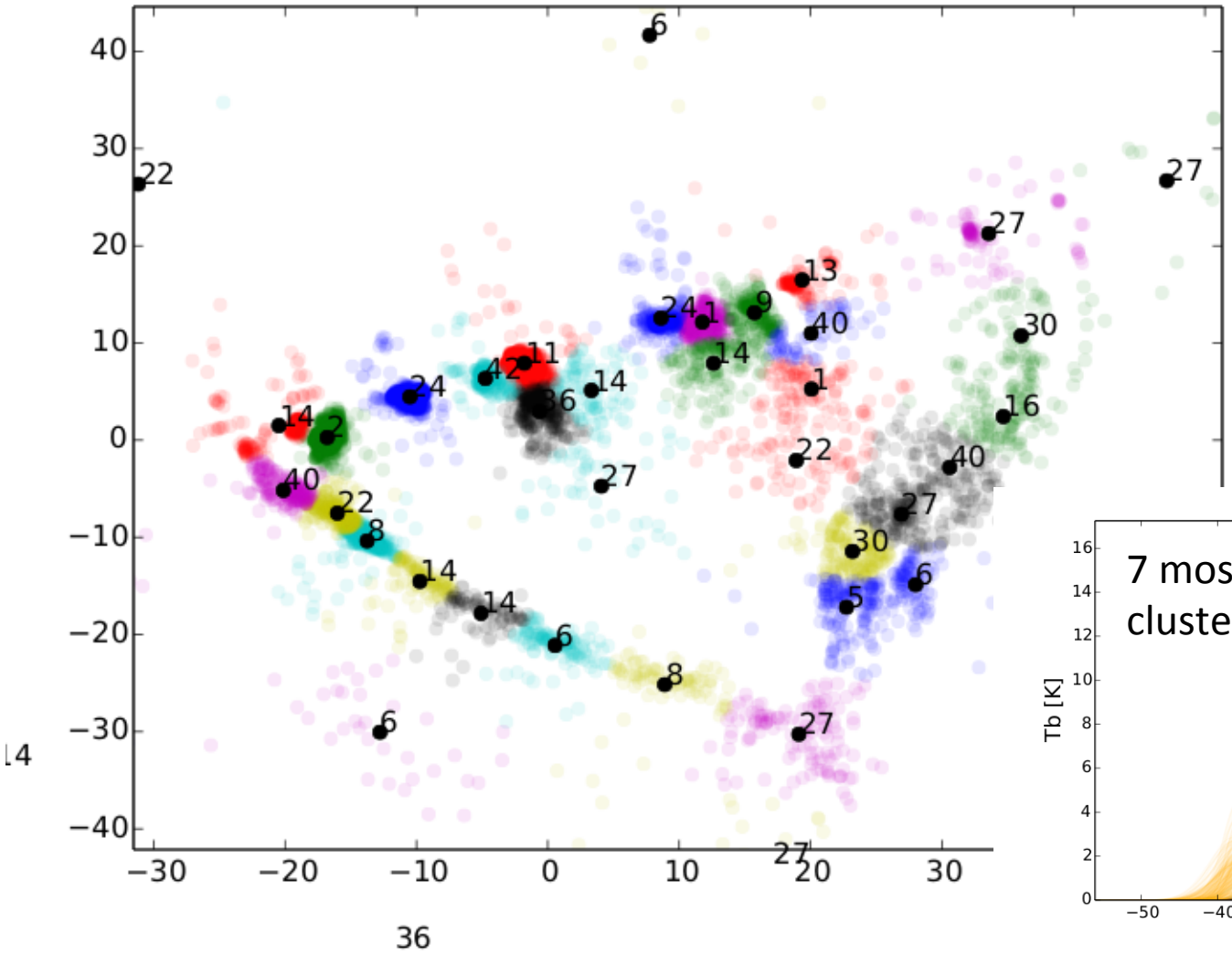
# Clustering: K-means

- Expectation Maximization Algorithm:
  - 1. Assign initial cluster locations
  - 2. Assign labels to all points based on distance
  - 3. Re-compute cluster locations using mean of all points in a given cluster
  - Go to 2.
- http://www.bytemuse.com/post/k-means-clustering-visualization/



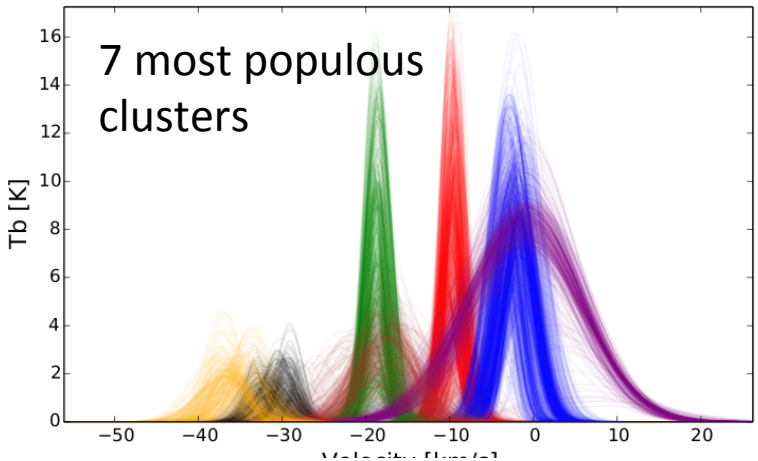


# Example, K-means clustering



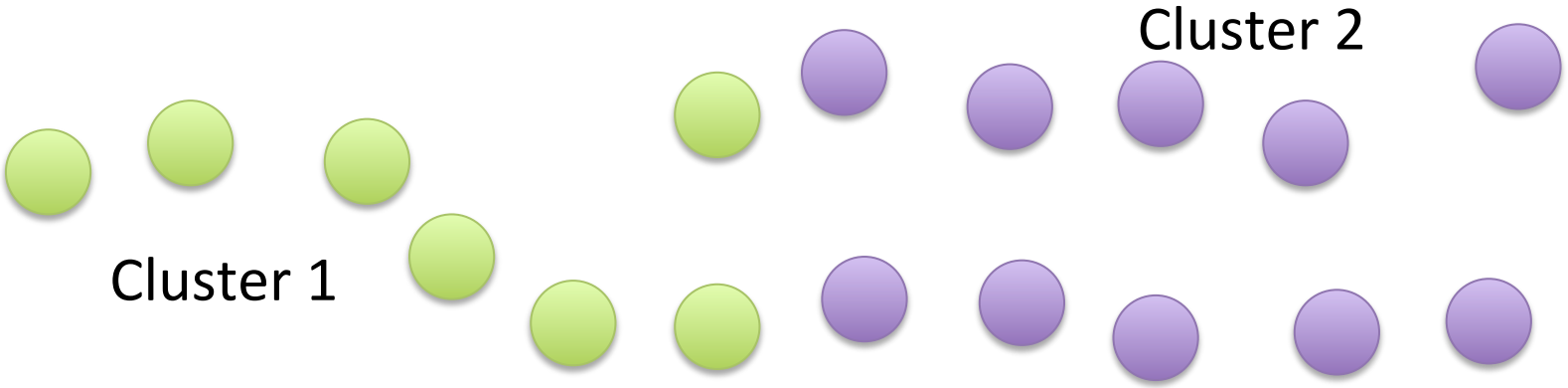

21cm Emission Data
of quasar
3C454.3 (credit C. Heiles)

400 pixels, ~10k components



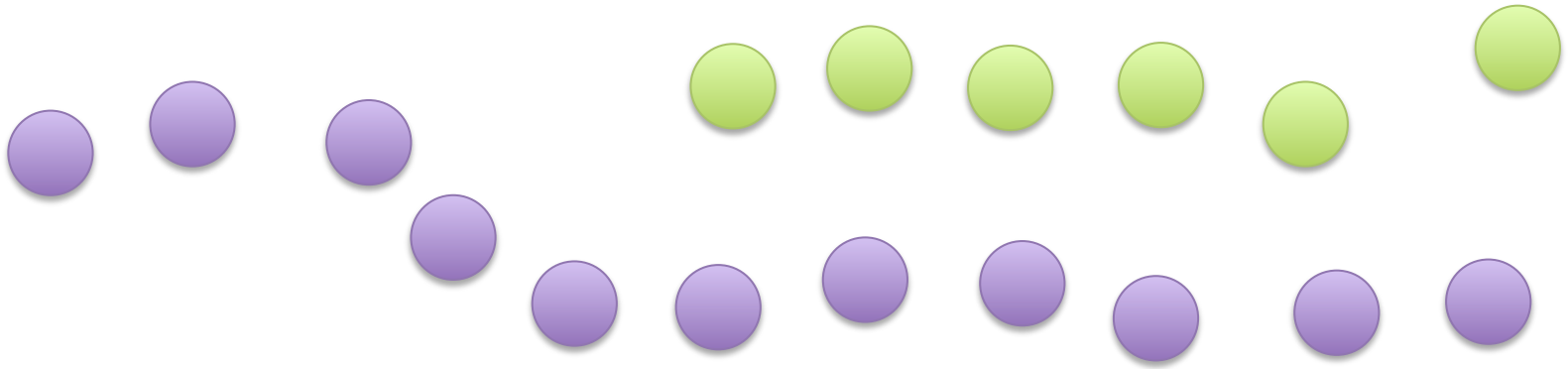
# K-means not optimal for “non-clumps”

E.g., K-means might decide to do this:



# Hierarchical clustering

- Procedural algorithm (lot's of "if-then" decisions)
  - Less mathematically elegant, still meaningful, very powerful
    - 1. Initialize data of N samples to have N distinct clusters
    - 2. Merge nearest pairs of clusters until desired number of clusters is reached. (Merging subject to nearest neighbor constraints)



# Example, Hierarchical clustering

2D hydrodynamic numerical simulation of Interstellar Medium. Audit & Hennebelle (2007)