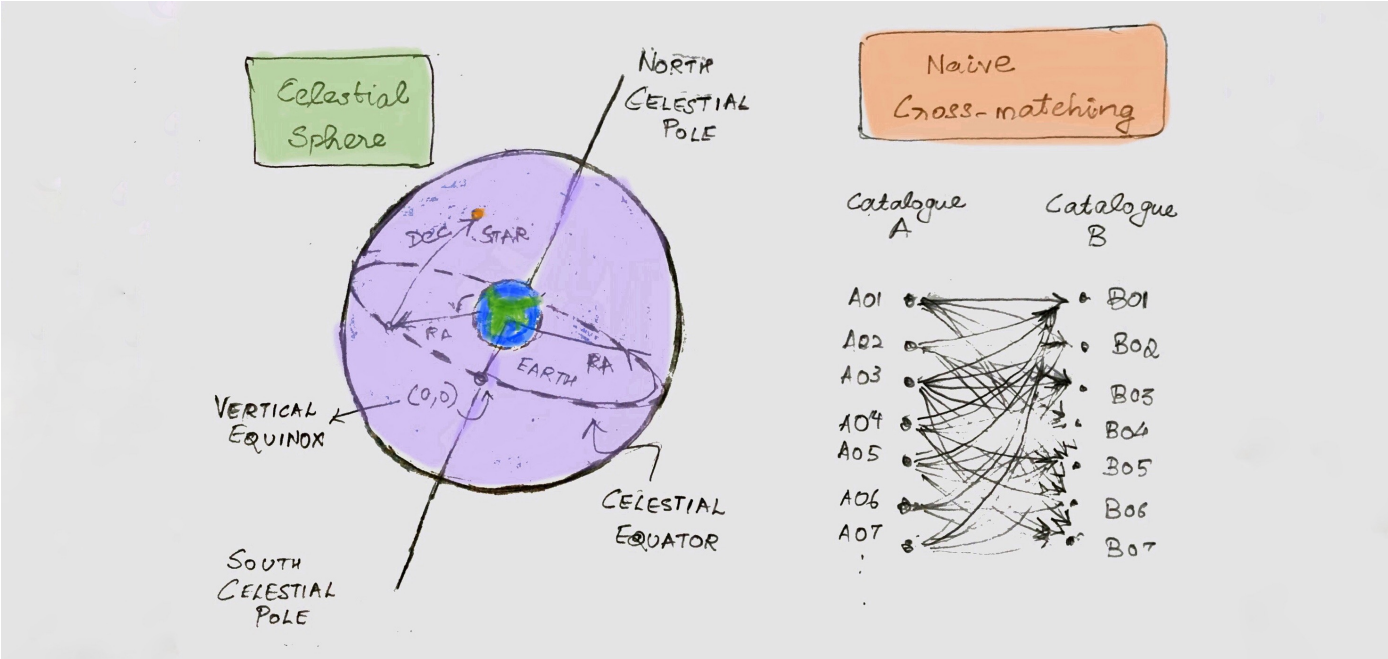
A naive cross-matching algorithm on galaxy catalogues

Rohith Krishna
11 August 2020



In this article, I present a naive cross-matching algorithm. An astronomer is interested in cross-matching objects discovered in different surveys. With billions of galaxies and the enormous data that it comes with, it is necessary to have an efficient algorithm that does this without blowing up in exponential time. Here, some fundamentals of astronomy is laid out first, followed by the implementation of the naive cross-matcher. In the real world too, business analysts in retail and ecommerce industries require cross-matching products in their shopping catalogues to provide consumers a display of their products without repetition of identical products.

This is part 1 of a series of articles on this topic of cross-matching. The naive cross-matcher here is an algorithm that is extremely slow and teaches us exactly what not to do. Seriously, do not implement this in comparing your catalogues.

Supermassive black holes and active galactic nuclei (AGN)

Galaxies are made up of stars, gas, dust and at its centre is a supermassive black hole - billions of times the mass of our sun, squeezed into a tiny space smaller than our solar system. These black holes influence the formation and growth of the galaxy. In some galaxies where gaseous and other matter present in the central region, this matter gravitationally accumulated towards the center by a process that astrophysicists call accretion. In the formation of these accretion disks surrounding supermassive back holes, a lot energy is released in the process. This is region at the center of a galaxy with high luminosity and energy radiations in some portions of the electromagnetic spectrum is called an **active galactic nucleus.**

The active galactic nuclei (AGN) are steady sources of enormous amounts of electromagnetic radiation in the universe. Their luminosity is so high that they outshine even all the stars in the galaxy combined. A typical range of emission of nuclei of nearby galaxies[1] is about $10^{33}\mathrm{Joule}\cdot\mathrm{s}^{-1}$ to $10^{40}\mathrm{Joule}\cdot\mathrm{s}^{-1}$. Such accretion also lead to huge jets of radiationand ionised matter emanating from the poles of the black hole. These jets travel at the speed of light and are usually dubbed *relativistic jets* and stretch to several hundred thousands of light years along that axis.

> ### An unsolved problem in physics
>
> In fact, the answer to why such accretion disks surrounding supermassive blackholes emit relativistic jets through their polar axes is still an unsolved problem is physics. The study of the nature of such radiation goes a long way in understanding jets from active galactic nuclei.

These jets are also associated with large magnetic fields and fast moving electrons spiralling around these jets produce strong and enormous radio emissions. The different sizes and structures in the radio jets can give us information about the different epochs in the accretion activity.

In 2006, a study by astronomers at Yale, found infrared emissions that throw light into the nature of quasar jets that emanate out of AGNs. This study particularly involves the jet of quasar 3C273, first discovered in 2006. They had reportedly developed a false-color image of quasar jet 3C273 based on the infrared data they observed, and to this they cross-matched with other emissions from radio waves to X-rays. These emissions were spread over more than hundreds of thousands of light years. [2]

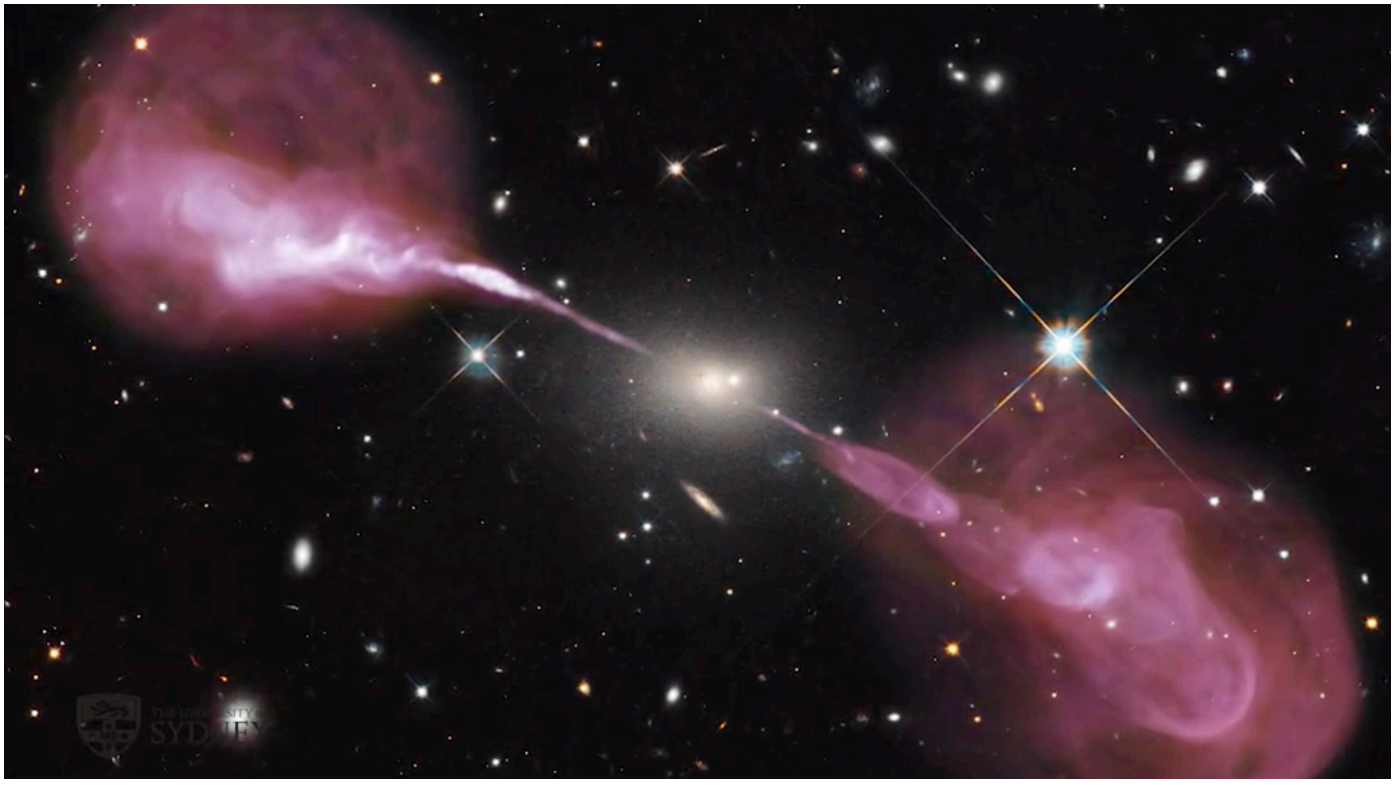
The need for cross-matching in astronomy

Astronomers use radio telescopes in deserts, optical telescopes in mountains and x-ray and gamma ray telescopes in space, to capture electromagnetic radiation at all possible frequencies.

Light from each part of the EM-spectrum gives a different insight into the physics of stars, galaxies and black holes. For example:

- Optical light of a ~ 100nm show where energy is being emitted by stars.
- Radio waves of ~1cm - 1m wavelengths show the presence of electrons accelerated by magnetic fields.

There is a need therefore to combine information from both optical and radio surveys, by matching objects in order to see if measured objects correspond to actual physical ones. This is done by creating a catalogue of objects from the surveys and then performing a **positional cross-match**. Cross-matching is the process of identifying different rows that may be in the same table or different tables, that refer to the same astronomical object.



The image above is Hercules A - a radio galaxy located 2100 million light years away. At its core is a super-massive black-hole and powerful jets transport energy to the galactic outskirts and beyond. From the optical image are seen the stars above that make up the elliptical galaxy 3c348.

Although positional cross-matching is a seemingly simple process, key insights in data science can be drawn by thinking in terms of *time complexity* and *scaling of algorithms.*

Positional cross-matching on images of galaxies

In this section positional cross-matching between catalogues is introduced. Source Extractor is a popular package used for performing this. The images to be cross-matched are first collected. In astronomy, the images are matrices with RGB or grayscale values for each pixel. The cross-matching exercise works as follows:

- Take an image. The program runs through the pixels in an image and finds statistically significant peaks.
- Then the surrounding pixels are grouped and a function called the point spread function is fit. The point spread function (PSF) describes the impulse response of an imaging system to a point object. [3] This is because the image produced by a point object is never a point but rather a patch of light distributed over a region of space along the plane of the image - commonly referred to an 'airy' pattern in diffraction. [4]
- The result is a list of astronomical objects, each with a position, an angular size, and intensity measurement.
- Now we employ cross-matching by searching the second catalogue and finding a counterpart for each object in the first catalogue.

Note.

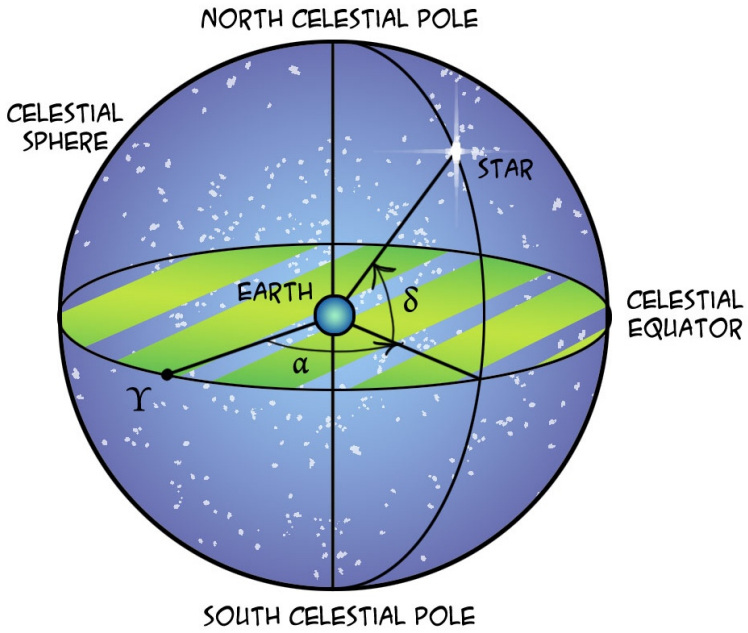
1. Points in an image are objects in space. In a 3-dimensional space, the distance between two points on the image (say, two galaxies) cannot be obtained by calculating the Euclidean distance. Instead, the distance has to be measured over the greater circle, as one would compute distances between objects on a spherical surface.
2. The component of statistical significance arises here, for there are uncertainties associated with the values measured from the images. This is because of noise in the image, telescope calibration or because of characterization of the telescope's PSF.

Before we go ahead with the cross-matching exercise, it is important to know the position variables that are relevant to us, and to define the angular distance, which is the measure of closeness between astronomical objects.

Coordinate system in astronomy

In astronomy one uses either equatorial or galactic coordinate system to position objects in space. The equatorial coordinate system is defined so:

- Objects in space are on a celestial sphere. The Earth's equator projected into the celestial sphere forms the celestial equator.
- Analogous to lattitudes and longitudes for measuring positions on Earth's surface/air, we use the angles **right ascension (RA)** and **declination (Dec)** to measure positions of objects on the celestial sphere.
- Right ascension is measured in hours, mins and seconds east from the position where the celestial equator intersects the ecliptic, i.e, the vertical equinox.
- Declination is measured in degrees, arcminutes and arcseconds and indicates the postion of the object north or south of the celestial equator.
- Since Earth's orientation in space changes over a period of 26000 years, we factor in a change in our coordinate system with **epochs.** The current epoch used is J2000.



Note. Since RA and Dec are given HMS and DMS notations, it is often useful to convert it to standard decimal degrees. A full circle of 360° corresponds to 24 hours. Hence, 1 hour in HMS corresponds to 15°. Dec on the other hand ranges from $+90^\circ$ to -90°. Thus, we have

$$RA_{decimal} = 15 \times (hours + \frac{mins}{60} + \frac{sec}{3600}) \tag{1}$$

$$Dec_{decimal} = \begin{cases} (hours + \frac{mins}{60} + \frac{sec}{3600}), & \text{for angle}>0 \\ (-1) * (hours + \frac{mins}{60} + \frac{sec}{3600}), & \text{for angle}<0 \end{cases} \quad (2)$$

In our algorithm, we make use of coordinates RA and Dec in the respective formats. Hence we define the following functions.

```
import numpy as np
def hms2dec(hours,mins,sec):
  return 15*(hours+(mins/60)+(sec/3600))

def dms2dec(hours,arcmin,arcsec):
  if hours>=0:
    return (hours+(arcmin/60)+(arcsec/3600))
  elif hours<0:
    return (-1)*((-1*hours)+(arcmin/60)+(arcsec/3600))
  else:
    return "Enter valid coordinates"
```

Measuring angular distances

Angular distance is defined as the projected angle between objects on the celestial sphere, as seen from Earth. We make use of the haversine formula here,[5] however, other measures of distances of points on the Great Circle might also be used.

Consider two points on the celestial sphere with right ascension and declination denoted as (α_1, δ_1) and (α_2, δ_2) respectively. The angular distance between these points d is,

$$d = 2 \arcsin \sqrt{\sin^2 \frac{|\delta_1 - \delta_2|}{2} + \cos \delta_1 \cos \delta_2 \sin^2 \frac{|\alpha_1 - \alpha_2|}{2}}$$

As is evident, angular distance is measured in degrees. We also define the function to calcaulate angular distances.

```
def angular_dist(r1, d1, r2, d2):
  b = np.cos(d1)*np.cos(d2)*np.sin(np.abs(r1 - r2)/2)**2
  a = (np.sin(np.abs((d1-d2)/2)))**2
  d = 2*np.arcsin(np.sqrt(a + b))
  return d
```

Loading the datasets

In this exercise, we shall use a naive positional cross-matching algorithm to check if astronomical objects in the first catalogue match with one in the second catalogue. We make use of the following data sources for our catalogue:

- AT20G Bright Source Sample (BSS) catalogue. This catalogue contains a list of brightest objects from the AT20G radio survey. Of the two, this is shorter and has only about 320 objects. The variables include positions in equatorial coordinate system, luminosities and other notes.
- SuperCOSMOS all-sky galaxy catalogue. This has a list of galaxies observed in visible light surveys. It has about 241 million observations - a large dataset of about 8.2GB in sheer size.

We extract the data from the respective pages in `.csv` format. The first dataset `bss.csv` contains objects from the radio survey and is described below:

```
=== bss.csv ===
Full, [MEM2008], n_, RAJ2000"h:m:s", DEJ2000"d:m:s", l_, S20Jy, S8.6Jy,
S4.8Jy, S1.4Jy, z, m20%, m8.6%, m4.8%
1,00,04,35.65,-47,36,19.1,0.87,0.97,0.90,1.7,3.2,2.8
2,00,10,35.92,-30,27,48.3,0.74,0.72,0.63,0.315,1.190,4.1,2.0,1.4
3*,00,11,01.27,-26,12,33.1,0.64,0.82,0.69,0.210,1.096,1.3,1.0,0.9
```

The first row is the ID which uniquely identifies the objects. There are natural numbers from 1, and can be left out, for they are the same as indices. The star (*) suffix is a correlation with another table and can be ignored here. The 2-4 columns are RA in HMS and 5-7 are Dec in DMS notations respectively. The remaining columns are luminosties or spectral densities. Only columns 1-7 are relevant for our analysis, as those are the only coordinates that deal with position.

The second 'large' dataset, called `super.csv` contains objects from the optical survey.

```
=== super.csv ===
RA,Dec,sigRA,sigDec,epoch,muAcosD,muD,sigMuAcosD,sigMuD,chi2,classMagB,clas
sMagR1,classMagR2,classMagI,meanClass,classB,classR1,classR2,classI,ellipB,
ellipR1,ellipR2,ellipI,qualB,qualR1,qualR2,qualI
1.0583407,-52.9162402,1.2605071E-05,1.3178029E-
05,1990.9344,-14.794838,-22.16756,7.242738,7.881182,5.027039,14.072,12.997,
13.293,12.74,1,1,1,1,1,0.182453,0.234902,0.213206,0.19472,16,16,16,16
2.6084425,-41.5005753,2.0626481E-05,2.0626481E-
05,1990.0508,-1.144597,-0.50977,10.397644,11.014809,0.245407,18.84,18.834,1
8.387,18.929,2,2,2,2,2,0.106605,0.112284,0.137899,0.091846,0,0,0,0
```

Here, the first two columns correspond to RA and Dec, while other columns correspond to shape of the galaxy. We shall extract these columns alone using the following function.

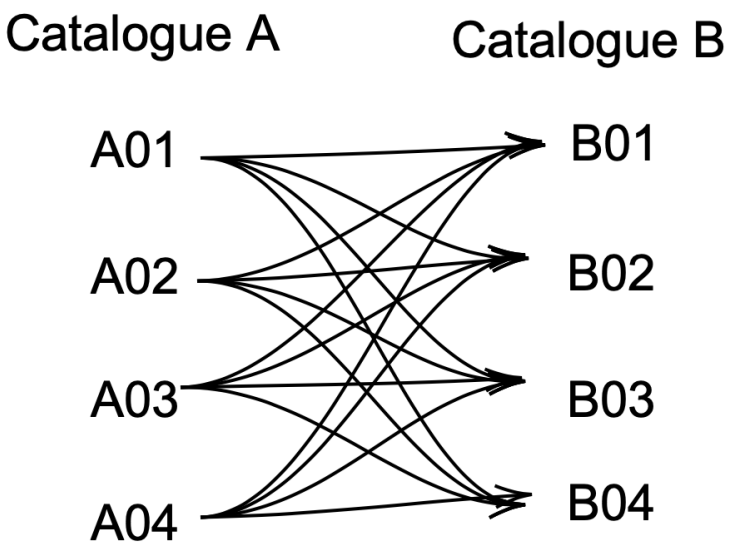
```
# importing bss.csv catalogue
def import_bss():
  res = []
  data = np.loadtxt('bss.dat', usecols=range(1, 7))
  for i, row in enumerate(data, 1):
    res.append((i, hms2dec(row[0], row[1], row[2]), dms2dec(row[3], row[4],
row[5]))) # conversion used
  return res # returns tuples of the form (id, RA, Dec)

# importing super.csv catalogue
def import_super():
  data = np.loadtxt('super.csv', delimiter=',', skiprows=1, usecols=(0, 1))
  res = []
  for i, row in enumerate(data, 1):
    res.append((i, row[0], row[1]))
  return res # returns tuples of the form (id, RA, Dec)
# Note: loadtxt works only so long as there aren't any NA's
```

A naive cross-matching algorithm

Now that, we've imported the relevant position variables from the two catalogues, we can implement a naive algorithm to cross-match. It is naive because, the following program, as we'll see later simply wouldn't scale.

1. Pick object in catalogue A.

2. Go through all objects in catalogue B and find the one closest to the chosen object.
3. If the objects are close enough (subject to a predefined maximum distance parameter), then record the match.
4. Repeat steps 1-3 for all other objects in the first catalogue.



As it can be seen in this algorithm every object in A is compared with every object in B. We first implement this algorithm keeping 160 objects in A and 500 objects in B. The function `crossmatch` is defined thus, keeping a particular level of `max_dist`.

```
def crossmatch(cat1, cat2, max_radius):
    matches = []
    no_matches = []
    for id1, ra1, dec1 in cat1:
        closest_dist = np.inf
        closest_id2 = None
        for id2, ra2, dec2 in cat2:
            dist = angular_dist(ra1, dec1, ra2, dec2)
            if dist < closest_dist:
                closest_id2 = id2
                closest_dist = dist
        # if match is outside max_radius, then ignored
        if closest_dist > max_radius:
            no_matches.append(id1)
        else:
            matches.append((id1, closest_id2, closest_dist))
    return matches, no_matches
```

```
if __name__ == '__main__':
  import time
  bss_cat = import_bss()
  super_cat = import_super()
  start = time.time()
  # Cross-matching with a max_dist of 40 arcseconds
  max_dist = 40/3600 # also implemented with other values.
  matches, no_matches = crossmatch(bss_cat, super_cat, max_dist)
  print(len(no_matches), len(matches))
  print(time.time()-start,"seconds")
```

Results

Under a 40 arcsecond maximum distance, we find that 151 objects in `bss_cat` matched with those in `super_cat` catalogue. When the permissible maximum distance was reduced to 5 arcseconds we find that there were 120 matches, leaving 40 unmatched. Further, we find that each cross-matching on 160x500 catalogues, takes about 1.5 seconds to compute. This clearly indicates that the algorithm would not scale.

On time complexity

In any analysis involving algorithms and computations, it is essential to deduce the time complexity of the algorithm in order to determine if it would perform the task in linear or constant time, or would blow up taking exponential time. In this context, we have say, n galaxies in catalogue A and m galaxies in catalogue B. The time complexity of our naive algorithm goes as $O(mn)$, which isn't a good worst-case time complexity.

In our case, with 160 objects in cat. A and 500 in cat B, it takes roughly 1.5 seconds to find all matches. If we were to use surveys with hundreds of millions of galaxies and implement this naive cross-matching, it would take days to compute! Hence, we certainly cannot use this frustratingly slow algorithm for cross-matching and need faster solutions. We shall explore these in the next part of this article.

Insights in Business Analytics

This would a good point to digress to the real world, and see what all this means for a business analyst. We have in the previous sections essentially written an algorithm to match objects in two catalogue. Oh! Where can we find applications for this in the real world? Obviously, in the retail and ecommerce industry which have millions of products to offer and millions to compare and ship. Consider an e-commerce giant wishes to procure products from a seller and list the seller's catalogue on their website. It has to compare the products on the seller's list with their original catalogue. If not, they'd be entering every single item sold by every single seller, resulting in buyers drowning in webpages of identical products offered by different sellers. Thus there is a need to efficiently cross-match objects in the catalogues, with real business consequences, that we shall discuss in a subsequent article.

References

1. Active galactic nuclei. Fabian. 1999. ↩

2. Shedding New Light on the 3C 273 Jet with the *Spitzer Space Telescope*, Astrophysical J. Sept 2006. ↩

3. Point Spread Functions. Astr 511/O'Connell Lec 13. ↩

4. Point Spread Function. Notes here. ↩

5. Haversine Formula for measuring distances on the Great Circle. ↩