Cross match between APOGEE and TGAS based on KDTree

Shengqi Yang

Department of Physics
NewYork University

Final Project Presentation of Computational Physics, 2016

Outline

1. Introduction
   - APOGEE and TGAS
   - KDTree Algorithm
2. Cross Match Results
   - Compare the self-defined KDTree with KDTree in Scipy

Outline

1. Introduction
   - APOGEE and TGAS
   - KDTree Algorithm
2. Cross Match Results
   - Compare the self-defined KDTree with KDTree in Scipy

APOGEE and TGAS

Two large datasets

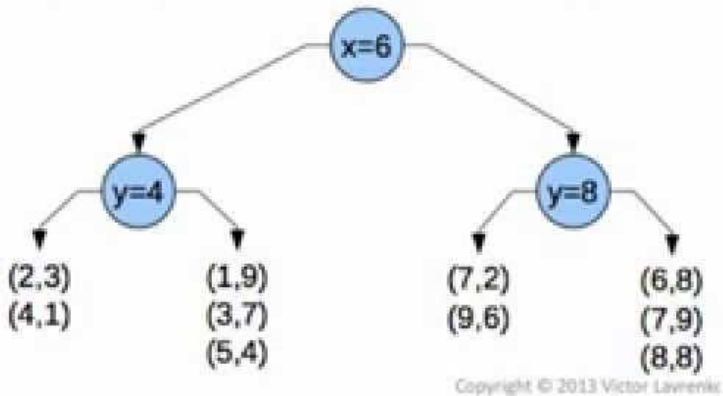
- APO Galactic Evolution Experiment (APOGEE): A survey contains information of 155632 red giant stars in Milky Way Galaxy, including high-resolution, high signal-to-noise position, bands magnitudes and so forth.
- The Tycho-Gaia Astrometric Solution (TGAS): A survey contains information about 2057050 stars in Tycho-2 catalogue, including their positions, parallaxes, proper motions and so forth.

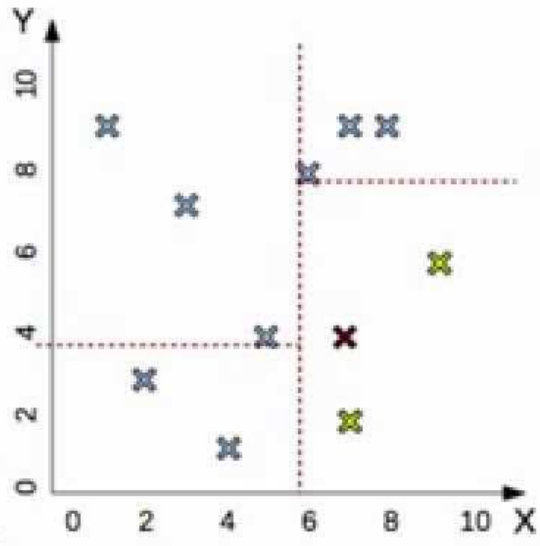
Outline

1. Introduction
   - APOGEE and TGAS
   - KDTree Algorithm
2. Cross Match Results
   - Compare the self-defined KDTree with KDTree in Scipy

K-D tree example

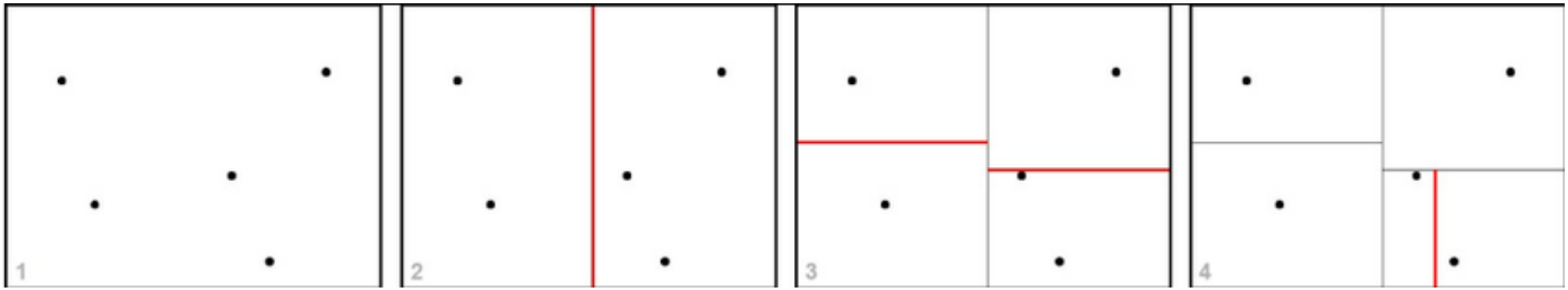
- Building a K-D tree from training data:
  - {(1,9), (2,3), (4,1), (3,7), (5,4), (6,8), (7,2), (8,8), (7,9), (9,6)}
  - pick random dimension, find median, split data, repeat
- Find NNs for new point (7,4)
  - find region containing (7,4)
  - compare to all points in region





KDTree Algorithm

Midpoint split



Outline

1. Introduction
   - APOGEE and TGAS
   - KDTree Algorithm
2. Cross Match Results
   - Compare the self-defined KDTree with KDTree in Scipy

My KDTree vs Scipy KDTree

Rules

Using midpoint split method and APOGEE data (volume is smaller) to construct KDTree and use TGAS data to inquiry the tree. Leaf size is 10 data points. Matching criterion is the angular separation is less than 2 arcsec.

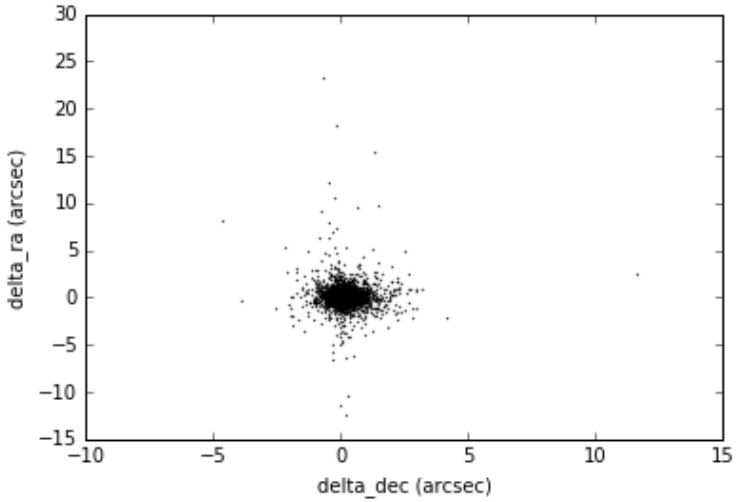
Overview

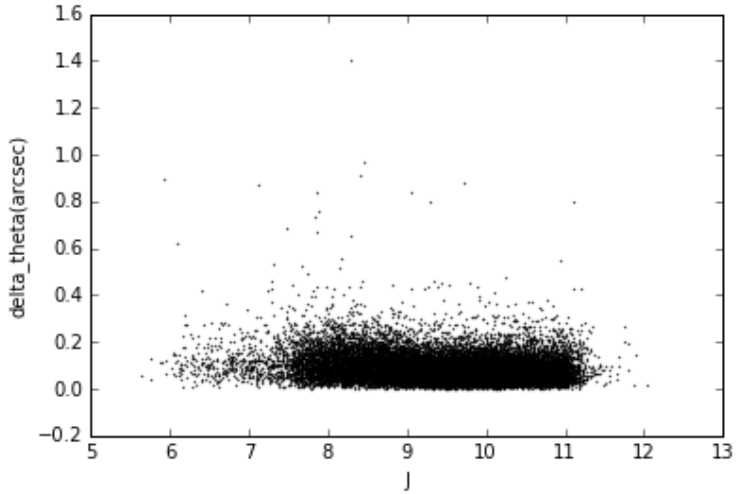
Scipy KDTree finds 20113 pairs (set A). My KDTree finds 20112 pairs (set B). Among them 300 pairs in set B is out of set A. 301 pairs in set A is out of set B.

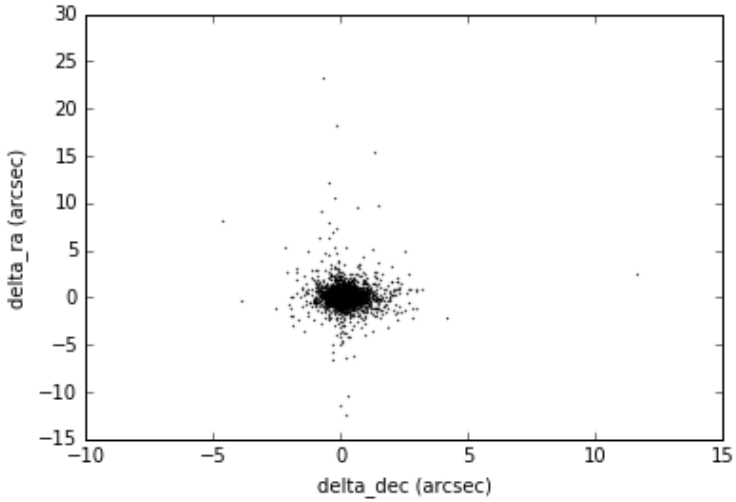
Further checking shows that APOGEE data contains a lot of repeat points. Data points in the mismatching sets are actually identical but with different index. In another word, Scipy KDTree and my KDTree give identical matching result.

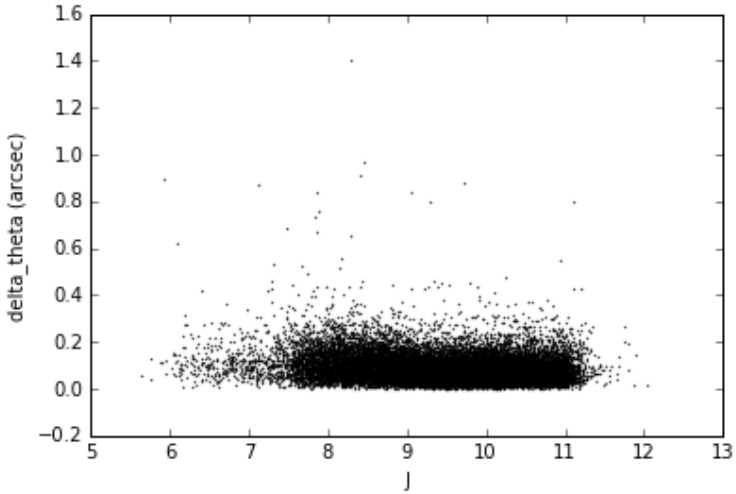
Scipy KDTree uses 131.96 seconds to complete cross-match, while it takes my KDTree 98.98 seconds.

My KDTree vs Scipy KDTree









Summary

- My KDTree gives extremely similar result with Scipy. It can efficiently pick out close data points in two huge datasets. It can be extended to satisfy various two dimensional cross match needs. It is 1.33 times faster than Scipy KDTree.
- My KDTree does not ensure a one-to-one mapping (neither does Scipy KDTree). My KDTree will generate empty leaves when processing higher dimensional data, but Scipy KDTree use Midpoint-Sliding split method, it will save some space.

For Further Reading I

Songrit Maneewongvatana and David M. Mount.
On the Efficiency of Nearest Neighbor Searching with Data Clustered in Lower Dimensions.
1990.

Source codes.
Scipy KDTree
Gaia tools .xmatch
Astropy .match to catalog sky