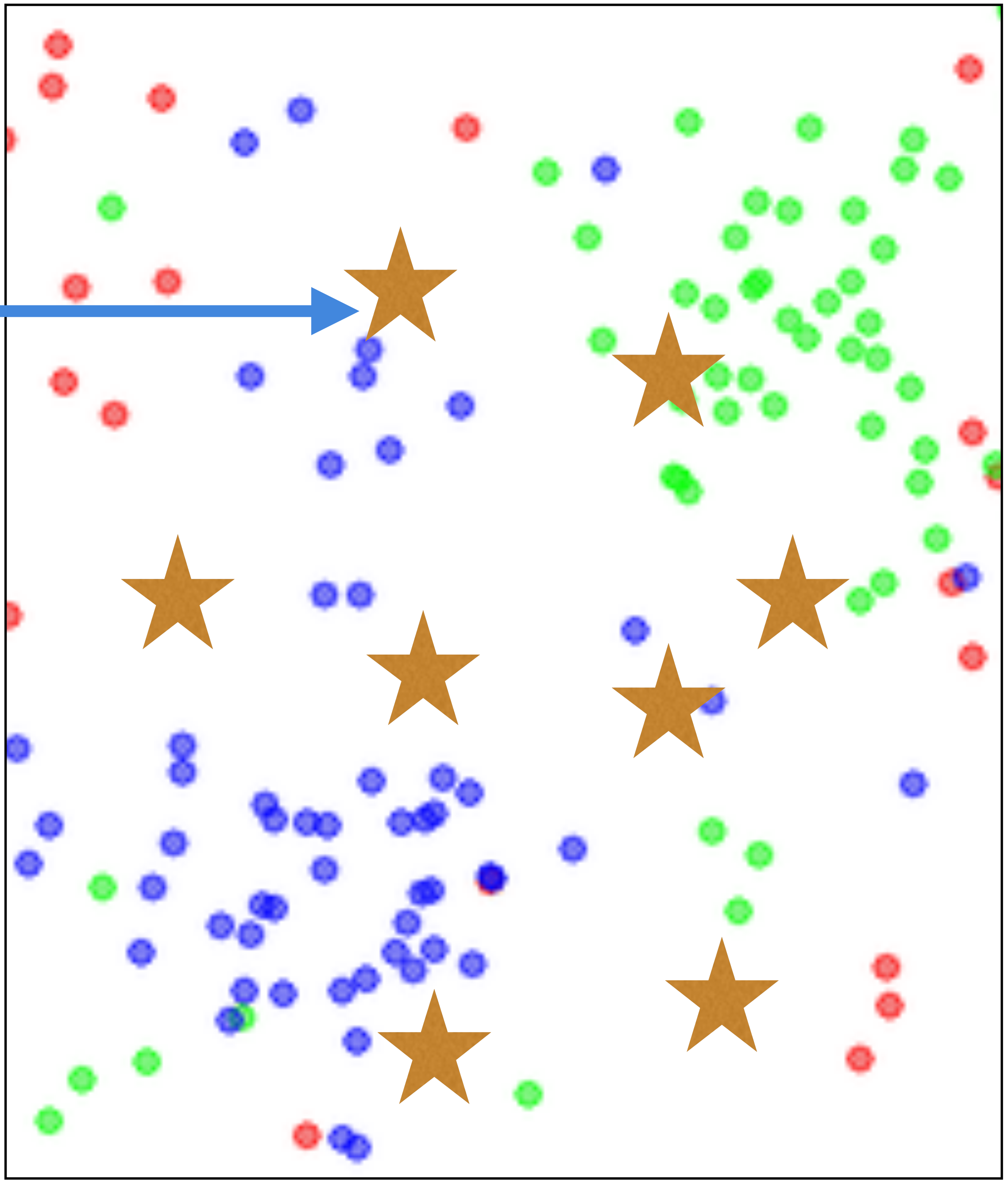
Quadtree-kNN: Improving Flink's kNN with a quadtree

Daniel Blazevski

Flink INSIGHT

kNN

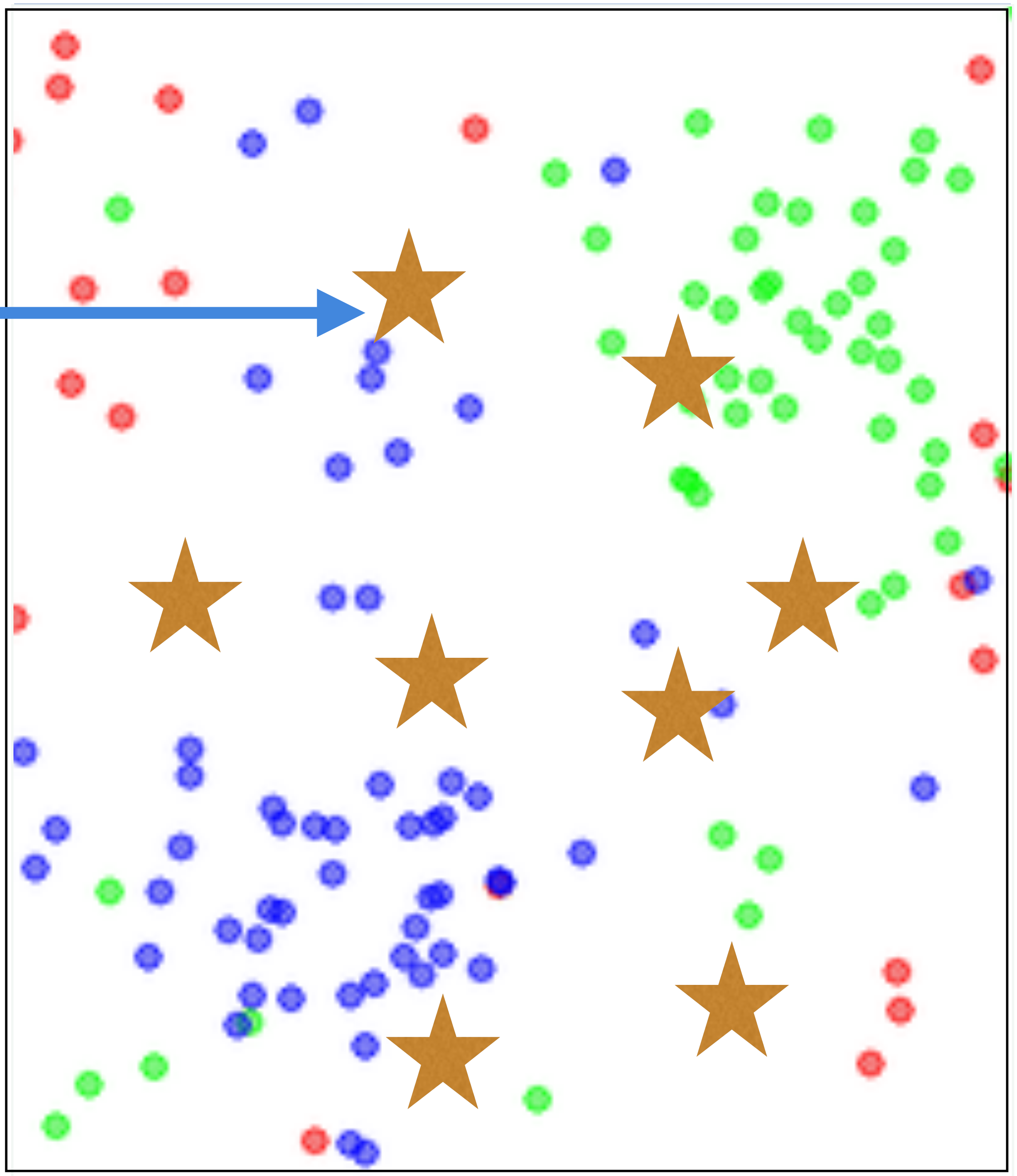
kNN: Which class do the gold stars belong to?



kNN

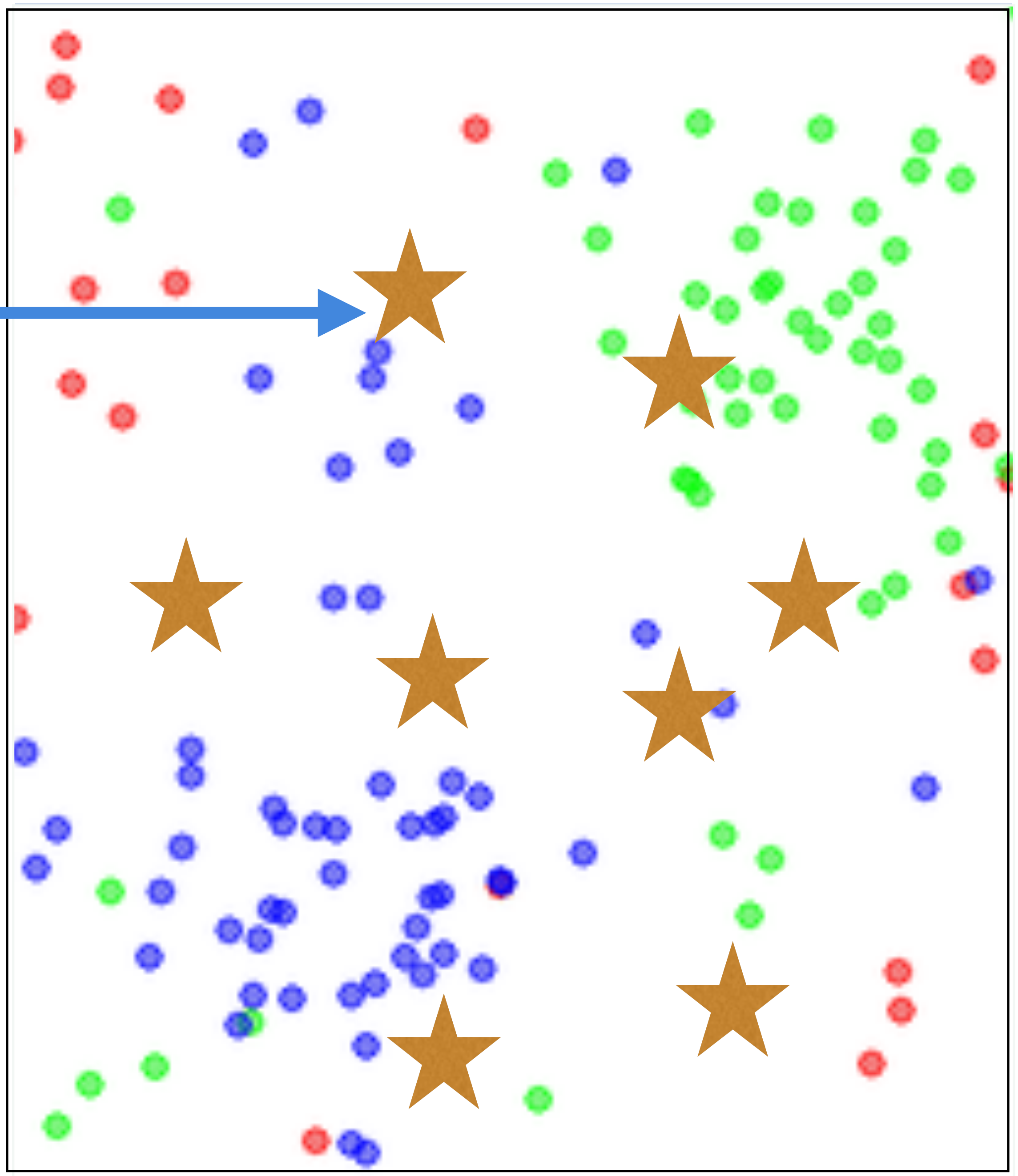
kNN: Which class do the gold stars belong to?

- Look at the k-nearest neighbors (kNN)



kNN

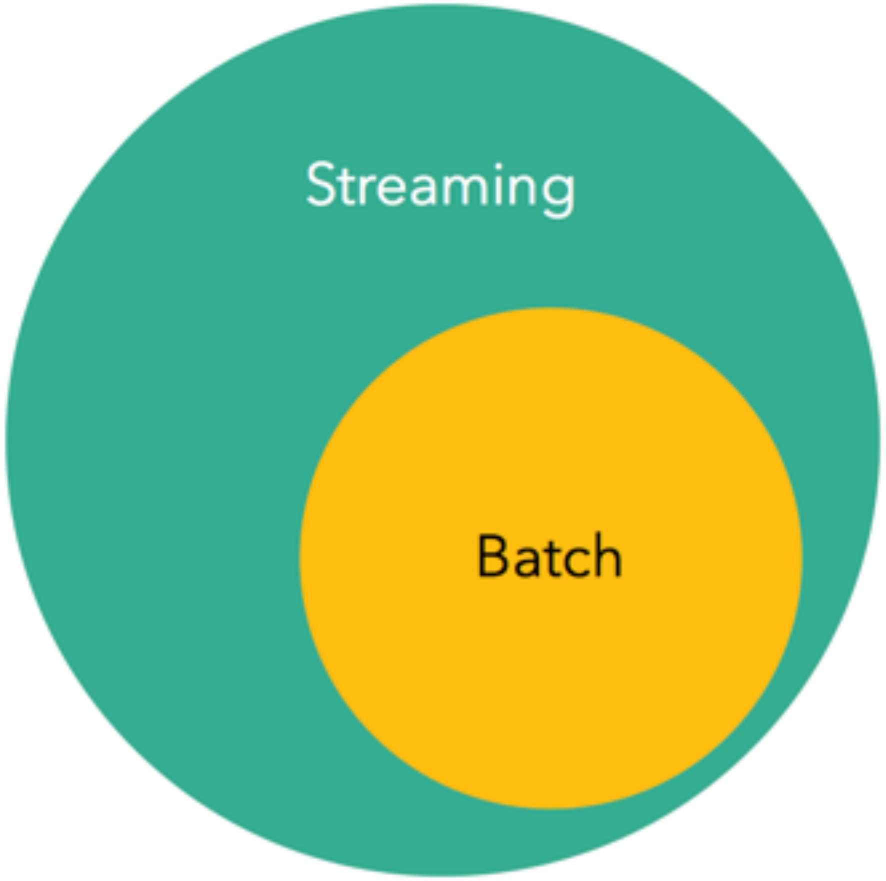
kNN: Which class do the gold stars belong to?



- Look at the k-nearest neighbors (kNN)
- Wide array of applications in data science

Flink

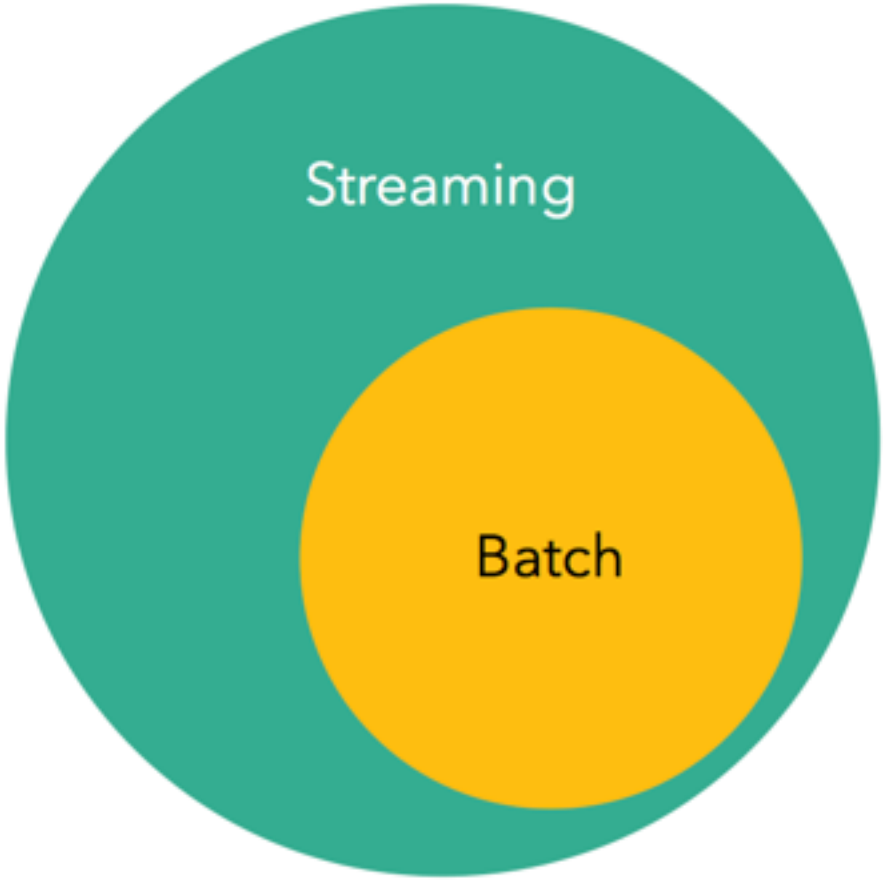
Philosophy: Batch is a subset of Streaming

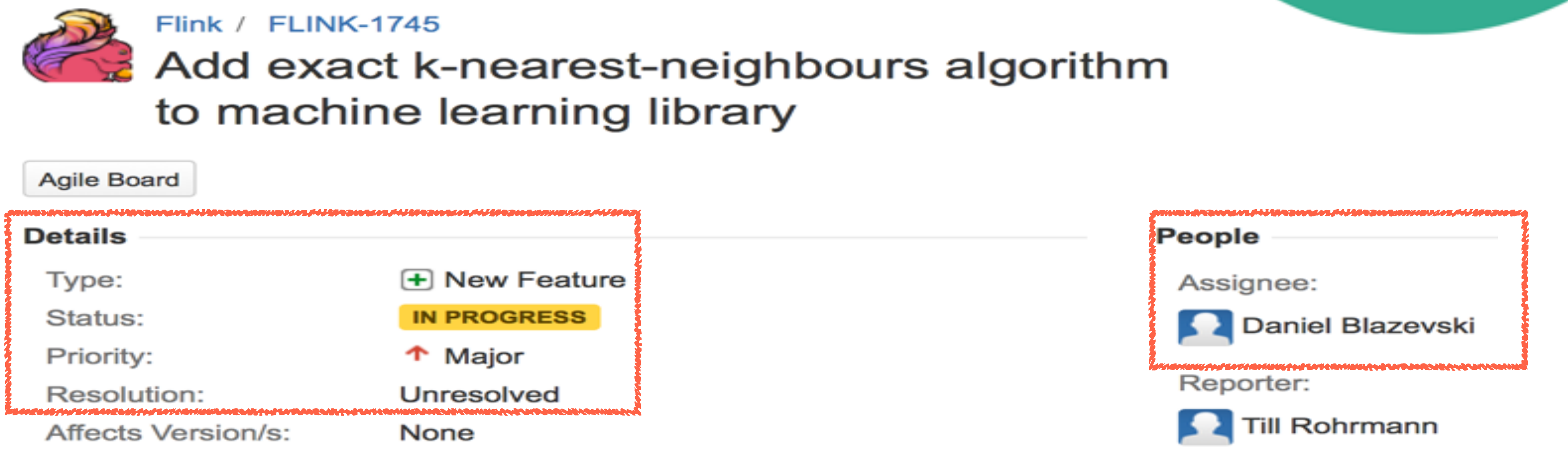


Flink

Philosophy: Batch is a subset of Streaming

- Distributed data processing tool
- December 2014: Became a top-level Apache project!





Flink / FLINK-1745

Add exact k-nearest-neighbours algorithm to machine learning library

Details

Type:	⊞ New Feature
Status:	IN PROGRESS
Priority:	↑ Major
Resolution:	Unresolved
Affects Version/s:	None

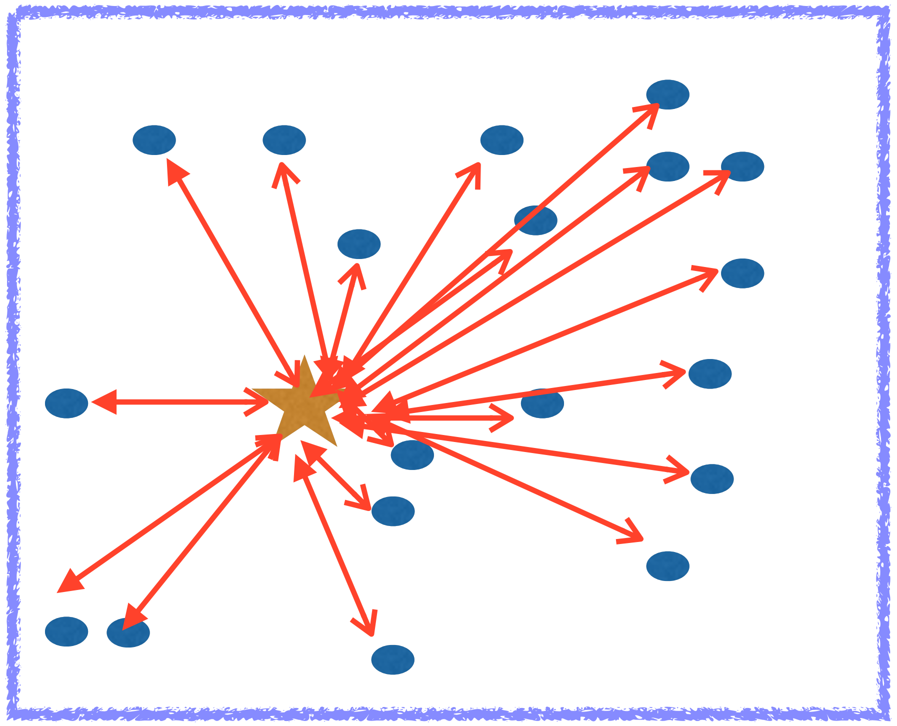
People

Assignee:
Daniel Blazevski

Reporter:
Till Rohrmann

Strategy

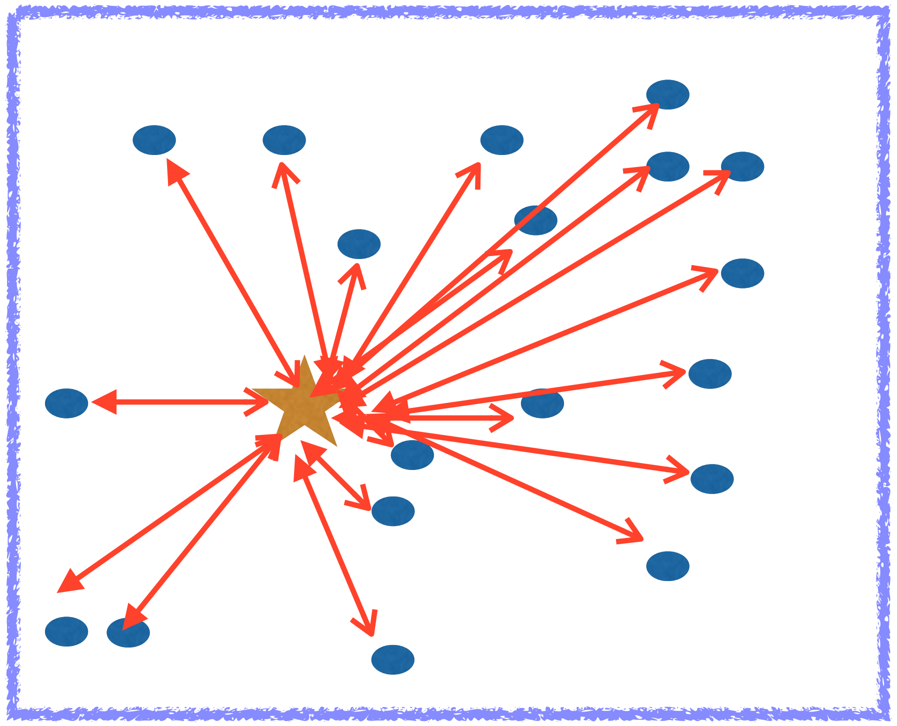
- Currently in Flink: Compute all pairwise distances — expensive!

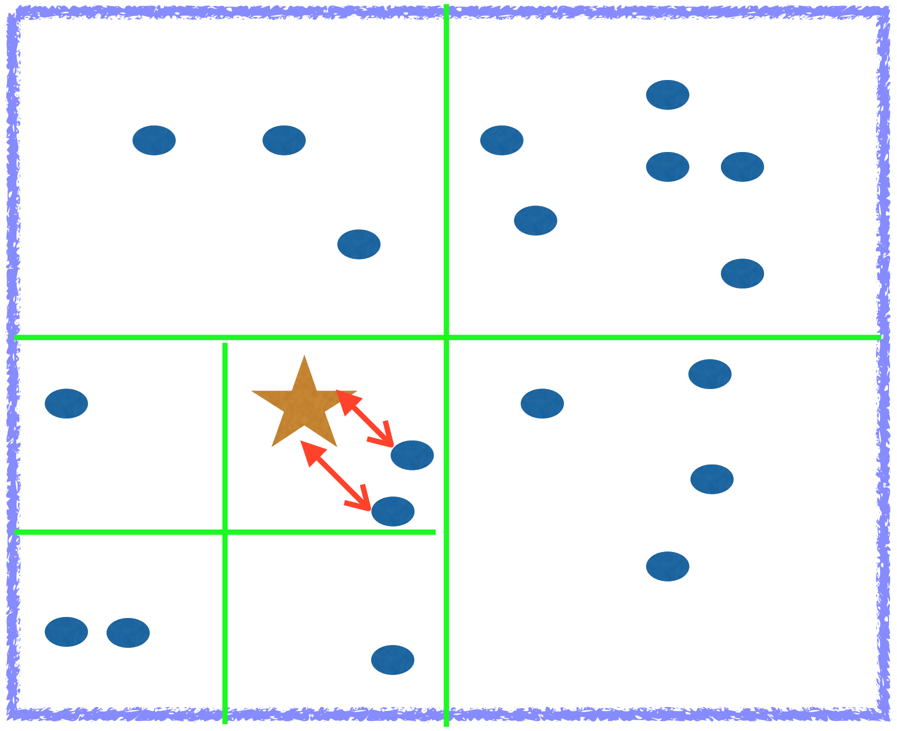


Strategy

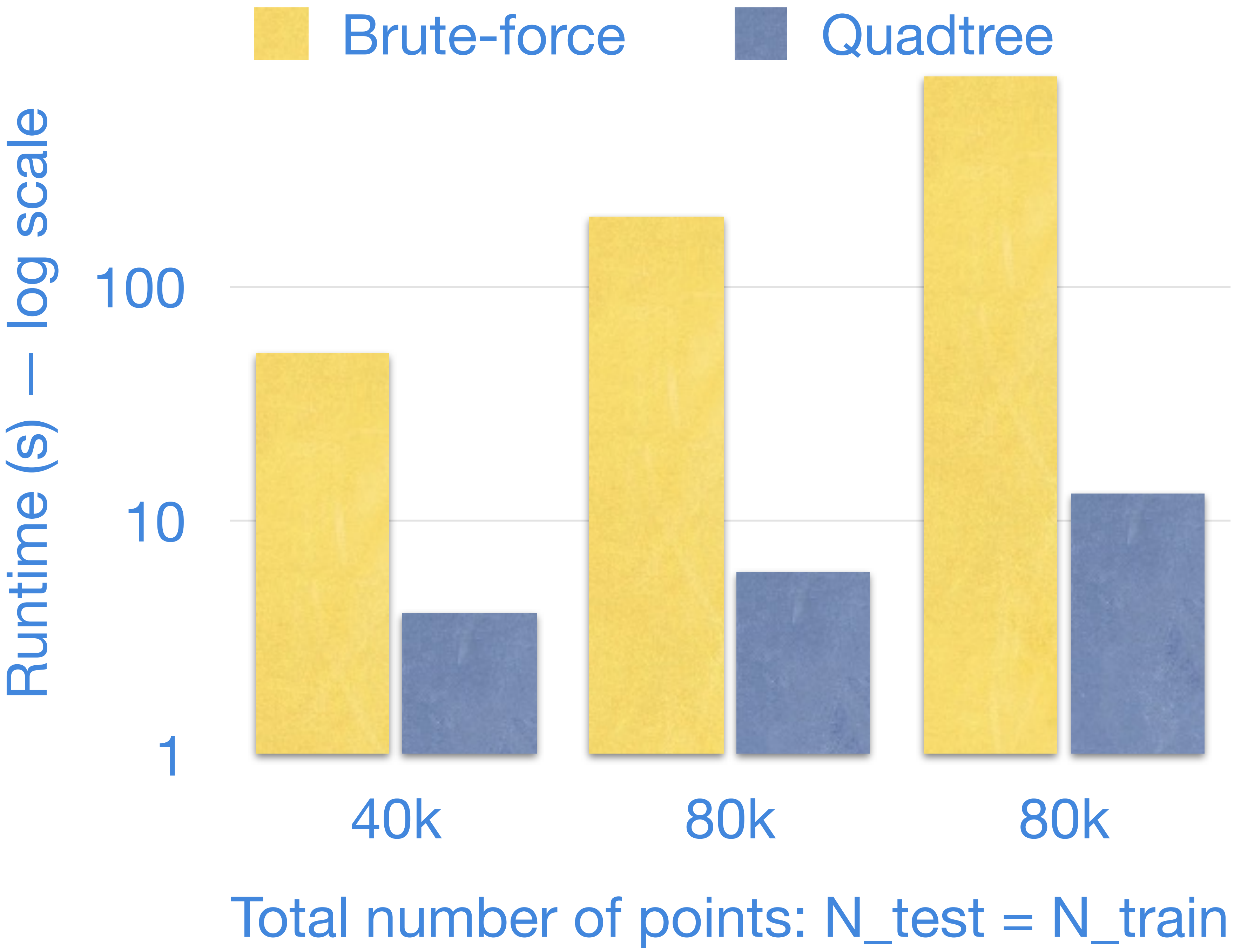
- Currently in Flink: Compute all pairwise distances — expensive!

- My work: Partition training set using a Quadtree

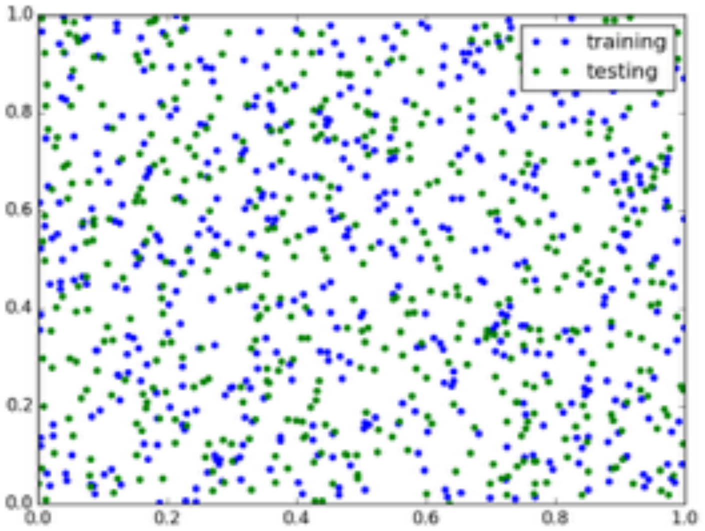




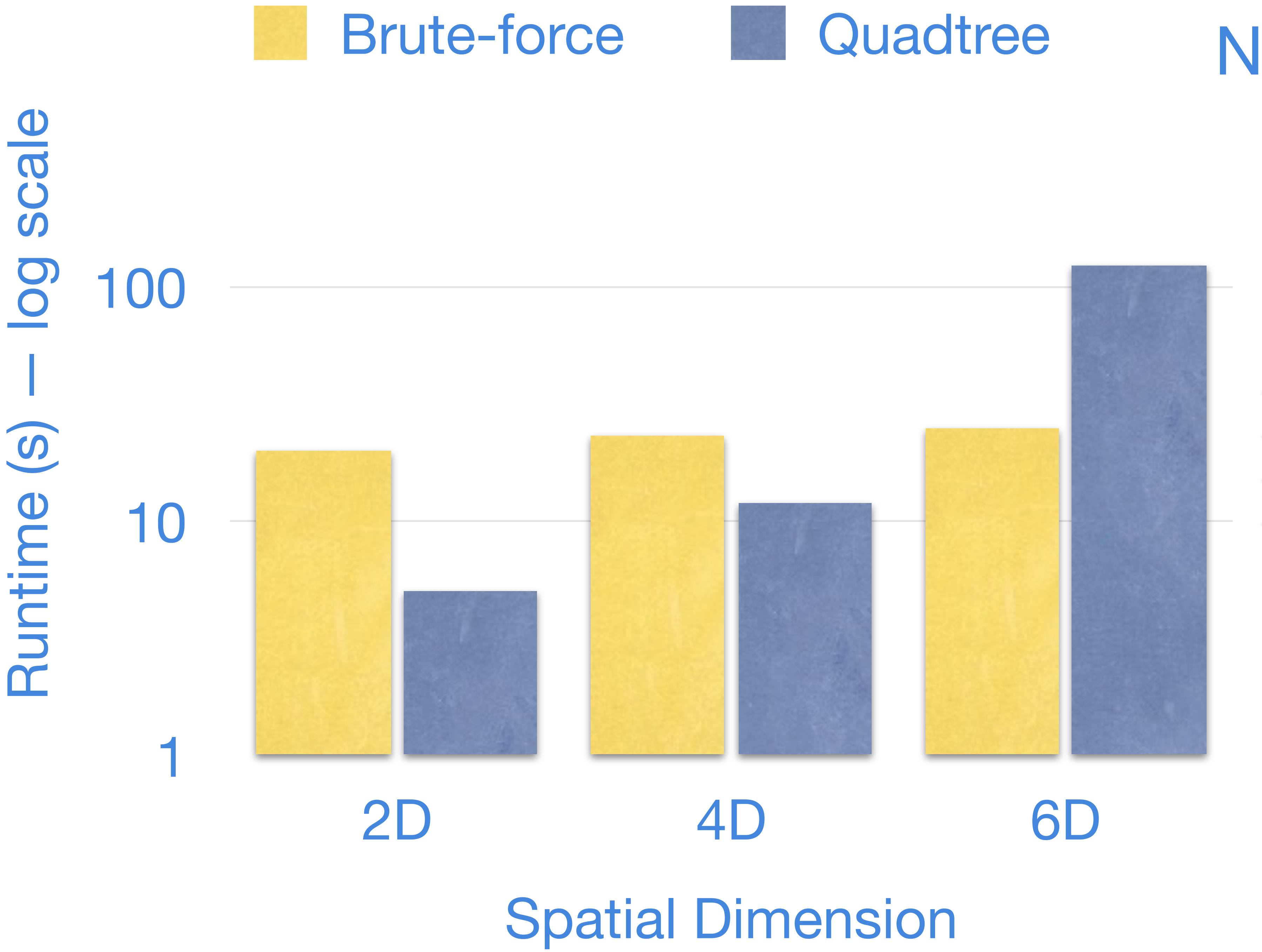
Performance



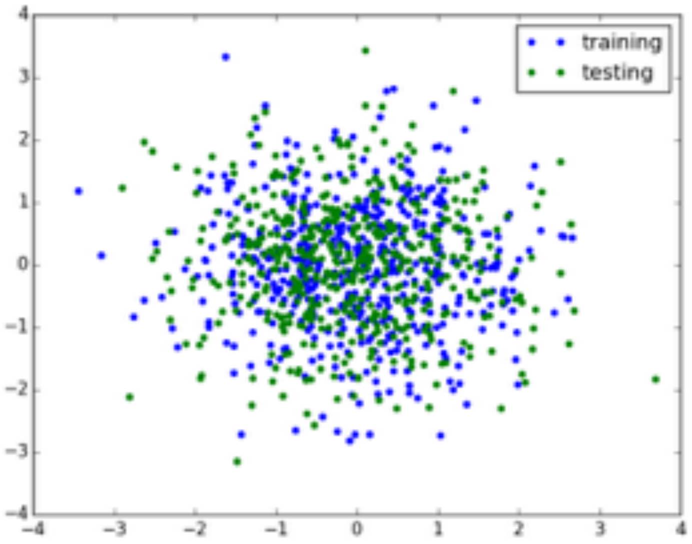
2D uniform data



Performance

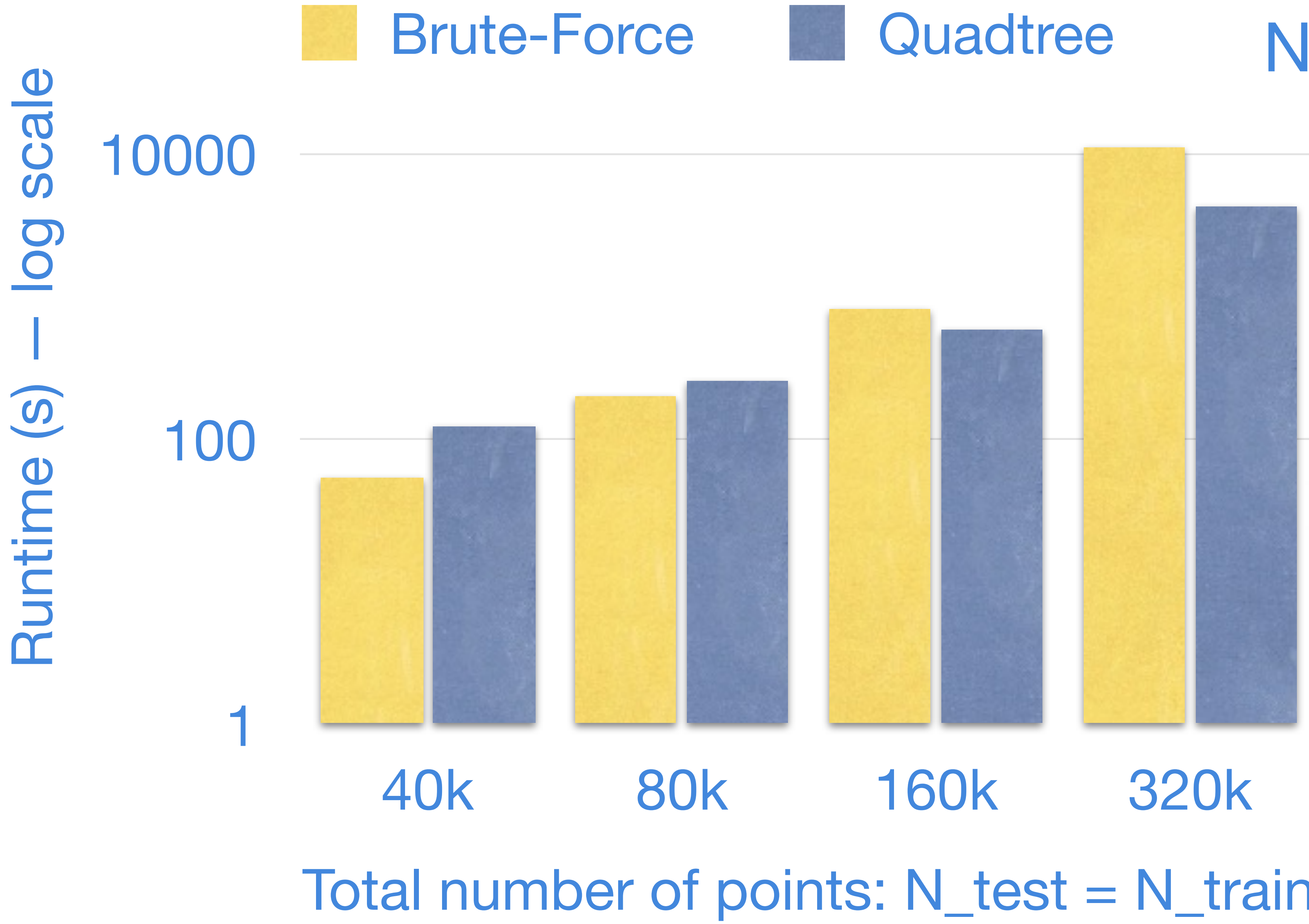


Normal distribution

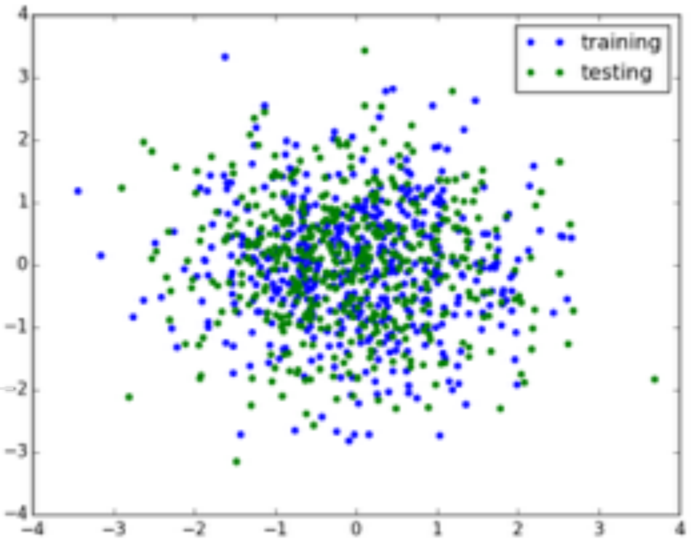


N_test = N_train= 20k

Performance



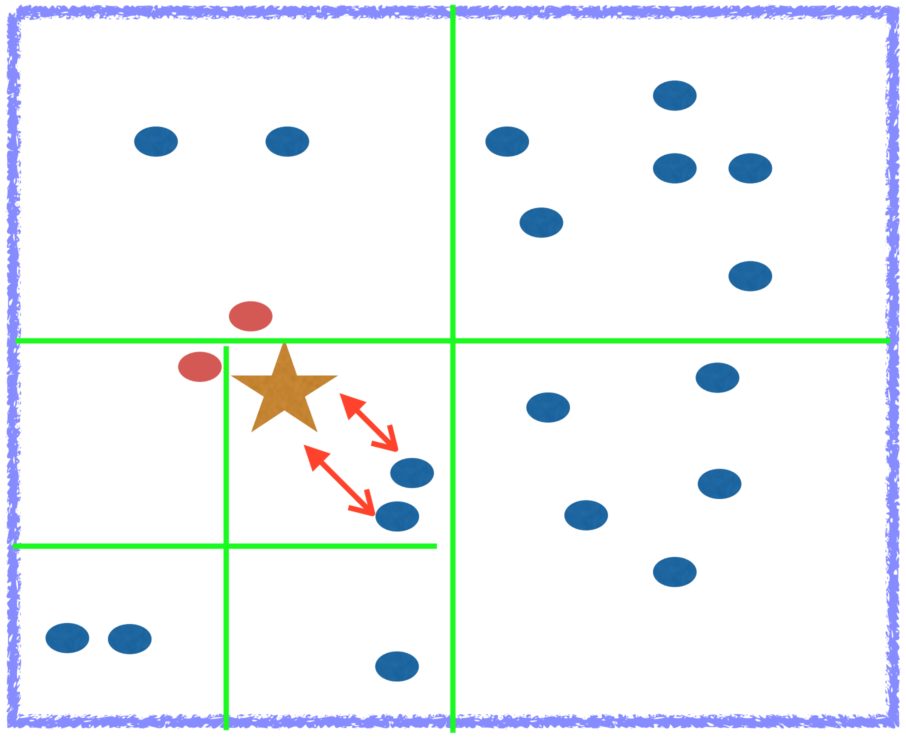
Normal distribution



Spatial dimension = 6

Challenge

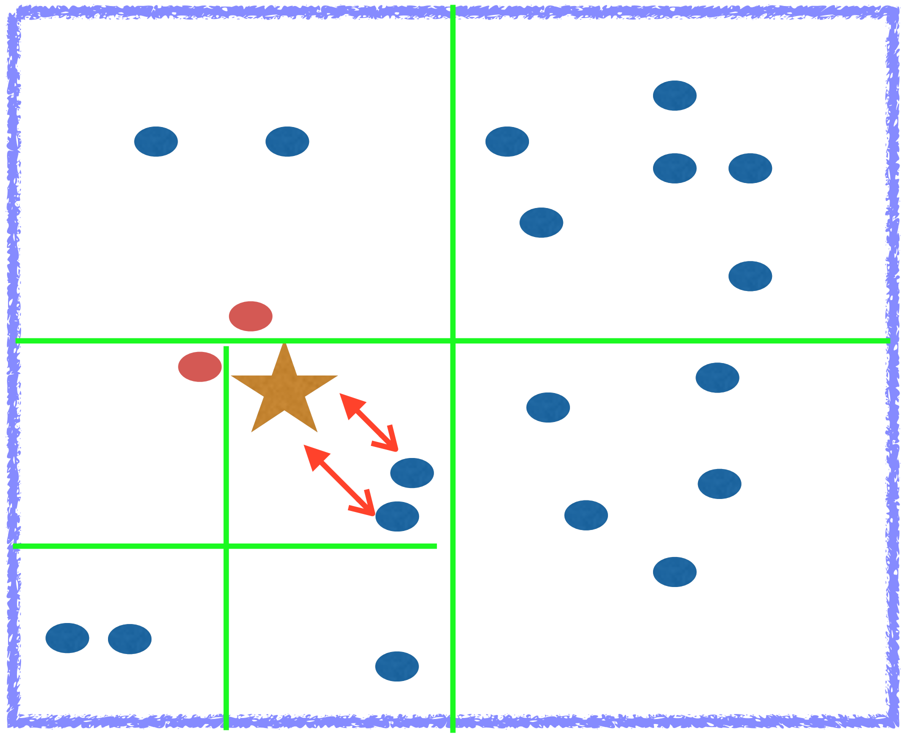
Nearest neighbors may not be in minimal bounding box



Scala Flink

Challenge

Nearest neighbors may not be in minimal bounding box

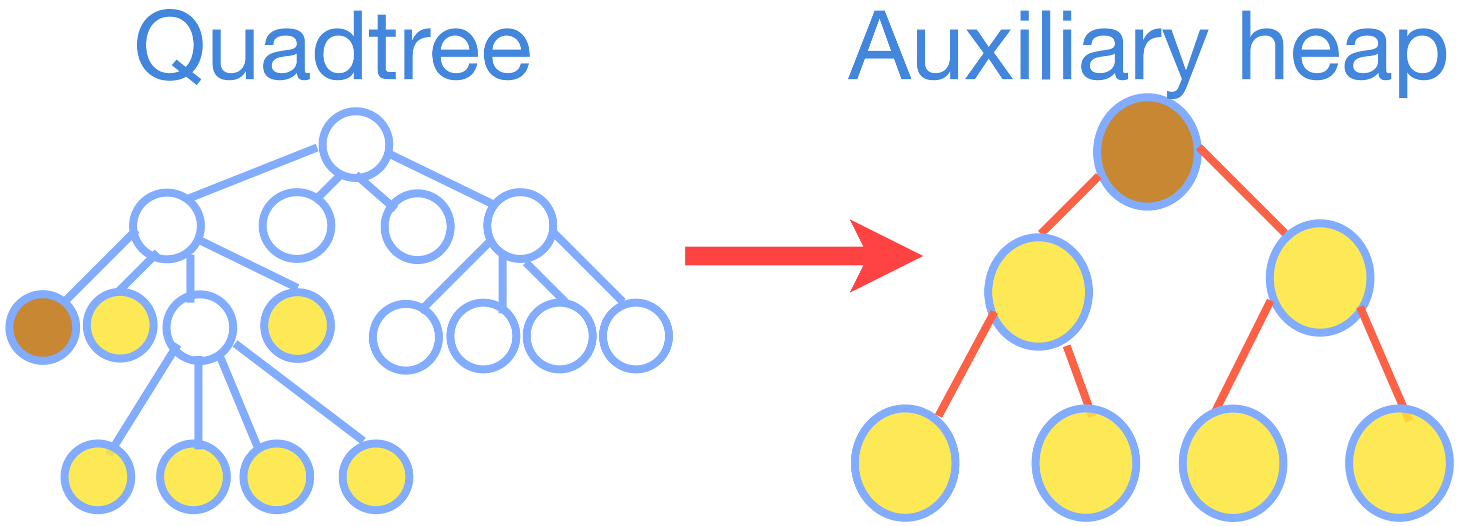


- min-heap on siblings' leaf nodes:

```
PriorityQueue[(Double, Node)]
```

- Priority = "Dist(star,box)"

```
node.minDist(obj:DenseVector)
```



Scala Flink

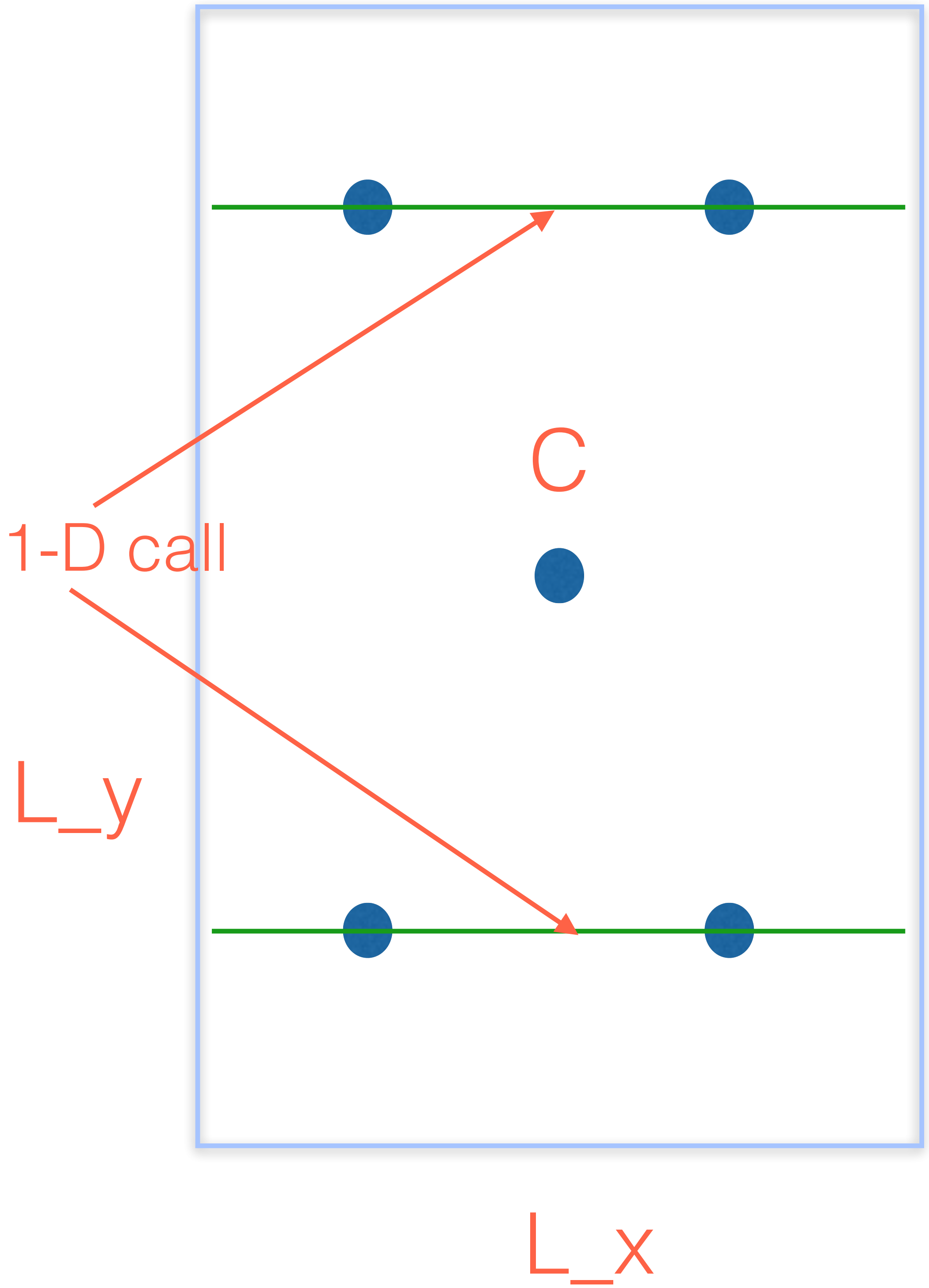
About me

- PhD in Math from UT Austin
  - aerospace and fusion energy
- Most recently: Oak Ridge National Laboratory
- Enjoy the outdoors

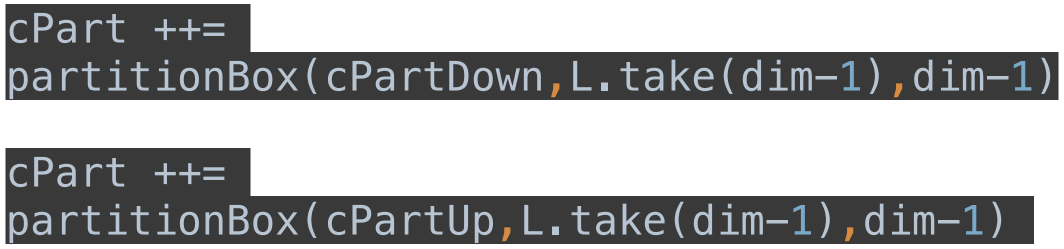
- daniel.blazevski@gmail.com
- github.com/danielblazevski
- project website: bit.ly/quadtree-flink

Back-up slides

Partitioning n-dim Box



- Box defined by center C, and Length vector L
- When partitioning, new L is easy: L <— L/2
- Have 2^(d) new centers!
- Use recursion by shifting up and down in the last coordinate



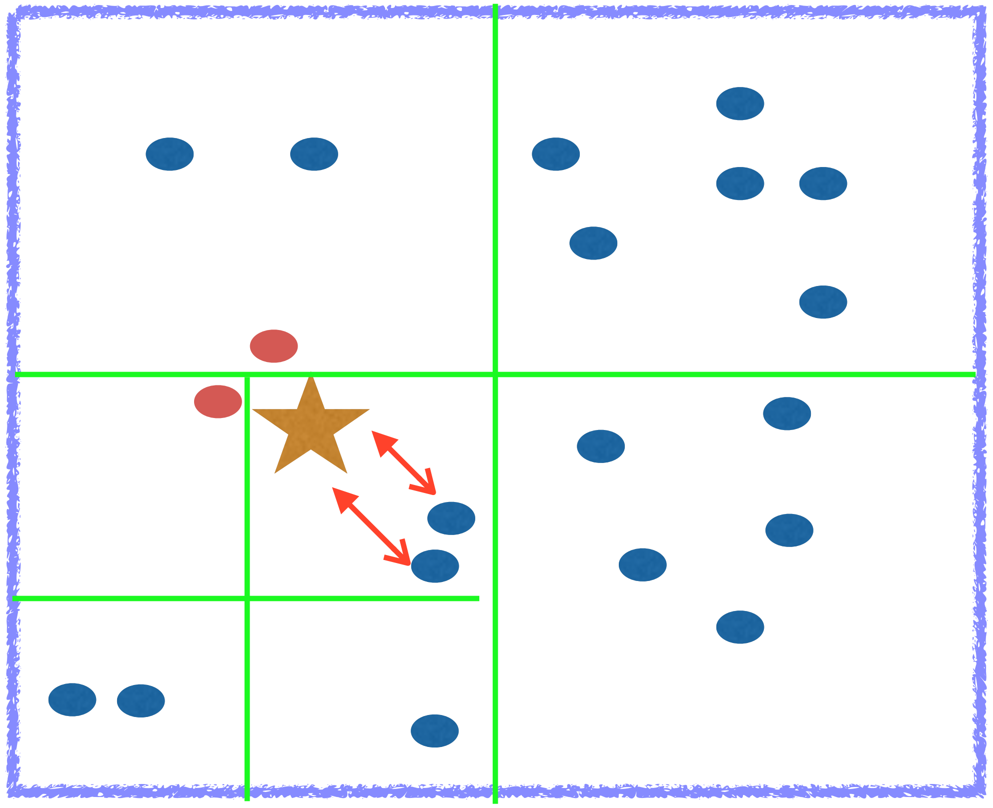
```
cPart ++=
partitionBox(cPartDown,L.take(dim-1),dim-1)

cPart ++=
partitionBox(cPartUp,L.take(dim-1),dim-1)
```

Scala Flink

Challenges

Nearest neighbors may not be in minimal bounding box



- Use a min-heap on leaf nodes:

```
PriorityQueue[(Double, Node)]
```

- Priority = “Dist(star,box)”

```
node.minDist(obj:DenseVector)
```

- Pop leaf nodes + objects until at least k “near” points; look at MAX distance, R

- Then search all boxes with Dist(star,box)<R

Scala Flink

Implementation

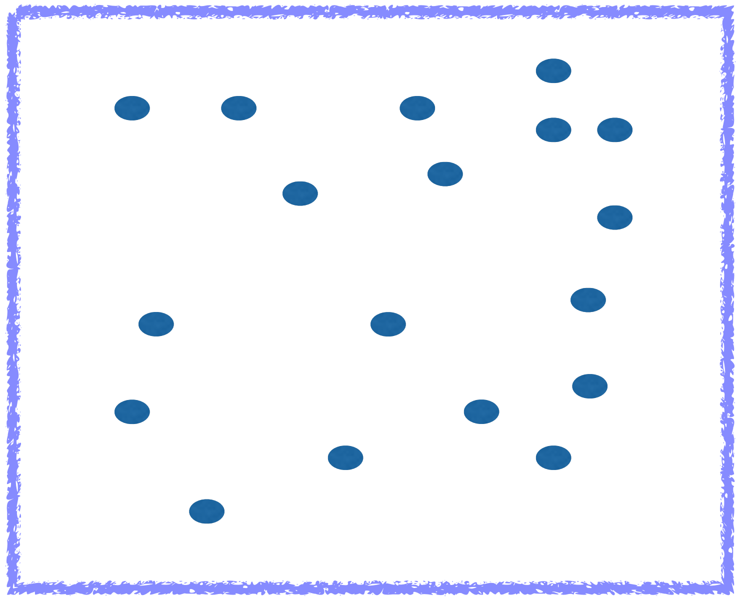
- Scala code; utilized Flink's existing structures

```
class QuadTree
(minVec:ListBuffer[Double],
maxVec:ListBuffer[Double])
```

```
def insertRecur(obj:DenseVector,
n:Node)
```

```
def searchRecurSiblingQueue
(obj:DenseVector,n:Node,
nodeBuff:ListBuffer[Node])
```

Scala Flink amazon web services EC2



Partition training set on nodes, then form a quadtree on each node