Music, Neighbors and tracking

Sabrina Amrouche, Tobias Golling, Andreas Salzburger, Moritz Kiehn

c.amrouche@cern.ch
Motivation

- High-luminosity LHC (and future FCC) will bring very high pile-up scenarios
- Average 200 proton-proton collisions for HL-LHC (up to 1000 proton-proton collisions for FCC)
- Combinatorics becomes very challenging
The approach

- Reduce combinatorial complexity: define regions
- Cluster inside regions
(1) Translate dataset to small sets

(2) Tracking in small set
(1) Translate **dataset** to small sets  
(2) Tracking in small set

100K points  
1 M - 960 dim
(2) Tracking in Small set
(2) Tracking in Small set
El Meu Poble
Txaṅgo · Som Riu

Governant
Txaṅgo · Som Riu

Corazón Viajero
Txaṅgo · Som Riu

Camp de Batalla
Txaṅgo · Som Riu

Com Dues Gotes d'Aigua
Txaṅgo · Som Riu

Tant de Bo
Txaṅgo · Som Riu

No t'Adormis
Txaṅgo · Som Riu

La Dansa del Vestit
Txaṅgo · Benvinguts al Llarg Viatge

Batega
Txaṅgo · Som Riu

Compta amb Mi

Valencià.
irinemercury
• “Many millions of songs”
• < 0.1ms to get n similar songs

[high-dimensional space]
• Unsupervised
Approximate Nearest Neighbors (ANN)
A **bucket** of neighbors

5 nearest neighbors in N dimensions

Traditionally, we do an exhaustive search - brute force
A **bucket** of neighbors

5 nearest neighbors in N dimensions

Traditionally, we do an exhaustive search - brute force
Bucketing mechanism
A **bucket** of neighbors
A **bucket** of neighbors
A **bucket** of neighbors
A **bucket** of neighbors
A bucket of neighbors
A **bucket** of neighbors
A **bucket** of neighbors

$O(\log(n))$
Building a Binary Tree
Data structures and algorithms
Data structure and Algorithm

- Some data structures associate with algorithm: trees, heaps, graphs...
- Some structures represent some concepts better than others
- The data structure can significantly reduce complexity

“Algorithms are like verbs and data structures are like nouns. An algorithm is just a method for doing something on a computer; a data structure is a layout for memory that represents some sort of data.”

Quora, What’s the difference between data structures and algorithms?
Search complexity vs
Dataset

- 10K particles, 100K points
- 5 features : global x, y, z, local $\theta$, $\Phi$
- Hashing library used : **HNSW**, Annoy (spotify)
- Buckets parameters
  - bucket size
  - nb trees (graphs)

Simulated with ACTS, Ttbar event, mu 200
A bucket of neighbors

$O(\log(n))$
A **bucket** of neighbors

query point

$O(\log(n))$
A **bucket** of neighbors

$O(\log(n))$
Efficiency

- Using truth information:
  - Count number of trackable particles found in a bucket
  - “Trackable particles” = 8 points (avg 11)
  - Efficiency as:
    
    \[
    \frac{\text{nb\_trackable\_particles\_found}}{\text{total\_trackable\_particles}} \approx 9K
    \]
0.07 ms per query
Data structure … again

Finding & removing particles

Rebuilding data structure
Rebuilding the data structure

Efficiency evolution, 6 trees of 20K queries, 8 hits min

- 0.07 ms per query
- 20 hits per bucket

- 20 hits per bucket, 0.32
- \( p_T > 0.4 \text{ GeV} \rightarrow 0.5 \)
- \( p_T > 0.5 \text{ GeV} \rightarrow 0.63 \)
- \( p_T > 0.6 \text{ GeV} \rightarrow 0.8 \)
- \( p_T > 1 \text{ GeV} \rightarrow 0.89 \)
Scaling of bucketing

![Diagram showing throughput versus dataset size for different bucket sizes. The x-axis represents dataset size (in # of hits), and the y-axis represents throughput (in buckets/s). The graph includes two sets of data points: black circles for 20 hits per bucket and purple circles for 50 hits per bucket. The throughput generally increases with dataset size.]
(1) Translate dataset to small sets  (2) Tracking in small set
Clustering inside buckets
Clustering inside buckets

- Complexity low enough for (any) technique to work fast...but
- Data driven approach has - many - advantages
  - A single technique to work hand in hand with data structure
  - Independance towards physics
    - expected to work as well for displaced tracks
    - Low pT, impact parameter...etc
  - Towards non-parametric approaches
Proposed approach

**Supervised** model trained on **buckets**

evaluating if **N** hits belong together
Supervised learning

Training

Test (new event)
Distribution of learnt distances

**Learnt metric**
Preliminary results

ROC

True Positive Rate

False Positive Rate

NN (area = 0.94)

5 Buckets - N=30 hits
3K segments/predictions
Buckets and data structure

Event

100K hits, 200 μ

|bucket| = 50 hits

| layer | = 15 hits

Layers

Clustering

~ 1 particle per bucket
Conclusion and next steps

● Faster and efficient data structures for tracking: reading tracks

● Open source hashing /ANN libraries:

  github.com/nmslib/hnswlib and github.com/spotify/annoy

Next

● Robust against physics parameters change: displaced tracks, Trigger.

● Implementation into Athena
Backup