Probabilistic Kinetic Data Structures

F. Betul Atalay **Sorelle A. Friedler** Dianna Xu

Motivation

- maintaining geometric structures online
- handling motion that is not known in advance and is not polynomial
- handling uncertainty in motion and in the resulting geometric structure
- real-world applicable framework
  - probabilistically predicted point motion

Kinetic Data Structures (KDS)

- assumes known point trajectories that are polynomials of bounded degree
- geometric structure is maintained in an online manner assuming these known trajectories
- *certificates* guarantee geometric properties
- *events* are scheduled and queued at predetermined certificate failure times
- rules are given to update certificates on failure

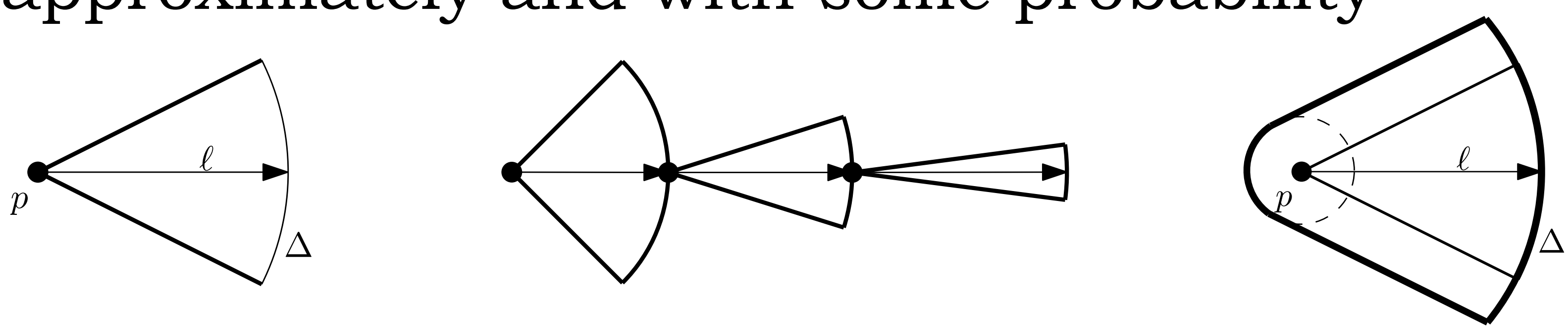
Kinetic Data Structures Analysis

small: O(poly log n) or $O(n^{\varepsilon})$ for arbitrarily small ε

- Responsiveness – certificate update time
  - responsive if the time is small
- Compactness – total number of certificates at any time
  - compact if only a small factor more than O(n)
- Locality – certificates per point
  - local if the number is small
- Efficiency – total certificate failures (compared to combinatorial changes)
  - efficient if the ratio is small

Our PKDS Framework

- user-given confidence threshold Φ
- point location is determined by querying a *model* which returns a distribution over
  - location
  - direction
  - confidence over time
- geometric structure is maintained approximately and with some probability



1D-maximum example

Problem: maintain the point with maximum value as points move in 1D

KDS solution: maintain a heap – the certificates are the parent-child relationships

PKDS solution:

- schedule *structural events* at the first intersection of the pie slices
- schedule additional *time events* at the end of the pie slices

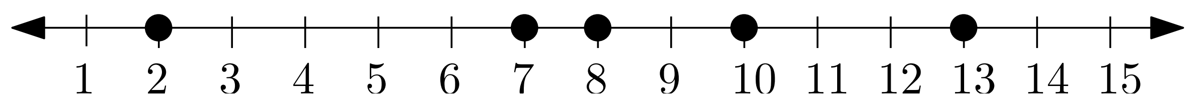
Approximately Correct Geometric Structures

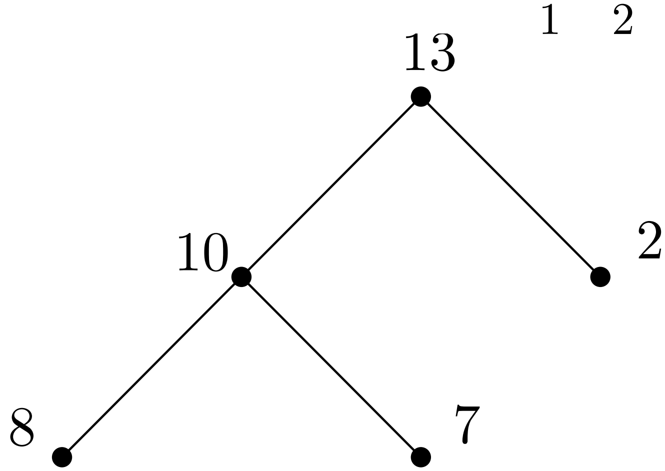
What does it mean for an algorithm to maintain a geometric structure approximately?

- Φ of the time the structure is 100% correct
- the structure is Φ correct at all times
  - relative error model – the structure is within Φ of the correct structure
  - absolute error model – the structure is within some fixed value away from the correct structure
  - robust error model – Φ of the points are correctly maintained within the structure

Results: Certificate-based error model

- Φ of the certificates are correct at all times
- The certificate-based error model implies the robust error model for these problems:
  - 1D maximum
  - Convex Hull





Results: PKDS framework

- The PKDS translation of a KDS solution to a problem is:
  - local
  - compact
  - responsive – assuming that a single future event can be calculated in O(poly log n) time
  - efficient or close to efficient – depends on the rate at which the pie slices lengthen under predictable motion
  - Φ^k correct under the certificate-based error model, where k is the maximum number of points per certificate

Results: Efficiency Analysis Details

Assumptions:

- original KDS structural events are divided evenly among the n points
- times between structural events are evenly spaced
- point motion is unpredictable at most a constant number of times between structural events
- when a point is undergoing predictable motion, the model increases the pie slice length at a rate that is between linear and doubling

Linear: $O(KDS) * \sqrt{(\text{total time steps} * n / KDS)}$

Doubling: $O(KDS) * \log(\text{total time steps} * n / KDS)$

Future work

To show new problems hold within PKDS:

- the certificate-based error model implies the robust error model for that problem
- future events can be calculated in O(poly log n) time