Compressing Kinetic Data From Sensor Networks

Sorelle A. Friedler and David M. Mount

University of Maryland, College Park

Motivation #1

- Kinetic data: data generated by moving objects
- Sensors collect data
- Large amounts of data
- Want to analyze it later
- Don't know what questions we'll want to ask in advance
- Lossless compression

Entropy

Consider the string generated by a random process…

- Entropy: The information content of a string or a measurement of the predictability of the random process

  $- \Sigma_x \, pr(x) \log pr(x)$

  - Example: A weighted coin that's always heads vs. a normal coin:

  $-(1 \log 1) = 0$ vs. $-(\frac{1}{2} \log \frac{1}{2} + \frac{1}{2} \log \frac{1}{2}) = 1$

- Joint entropy: The entropy based on joint probabilities of a set of events
- Normalized entropy: for strings of length n, 1/n entropy
  - bits to encode each character

Data Compression

- Encoded strings are compressed to a shorter length than the original

Consider the string generated by a random process…

- Optimal compression algorithms over a string achieve a per bit encoding rate equal to the normalized entropy

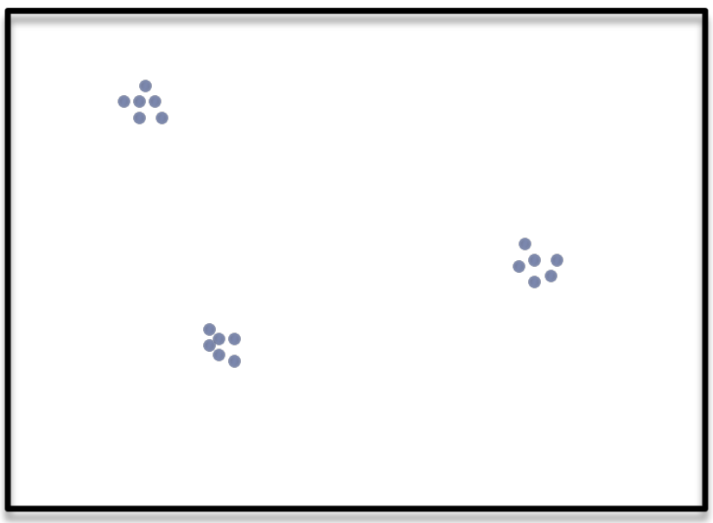
- Optimal compression algorithms over a set of strings achieve a per bit encoding rate equal to the normalized joint entropy

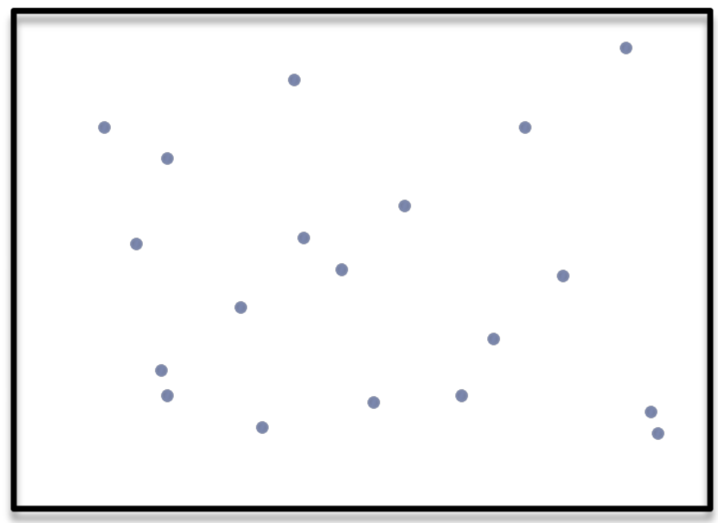
Data Compression Options

Lossless	Lossy
Data is completely retrievable	Some data may be lost
Compression bounds are theoretically provable	Can compress the data to fit in the space you have
Sliding-window Lempel-Ziv algorithm 1977	

Motivation #2

- Develop a framework for kinetic data from sensors
  - No advance object motion knowledge
  - No restrictions on object motion
  - Reasonable assumptions of what a sensor can know
  - Efficiency analysis that is <u>motion sensitive</u>



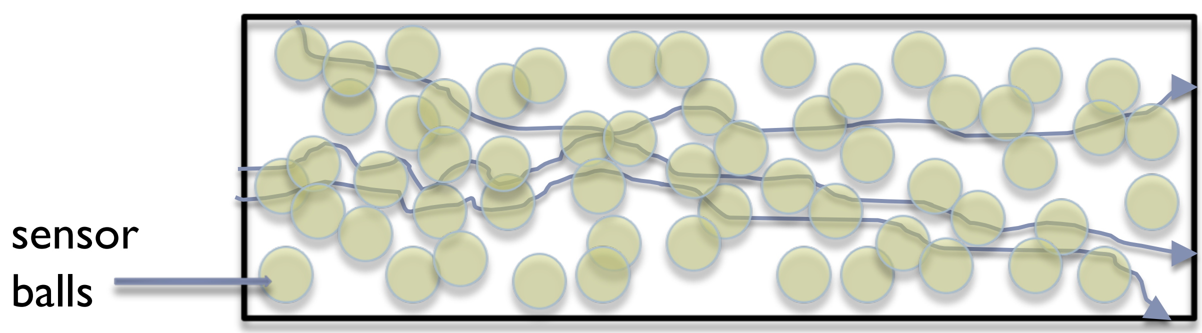


Existing Frameworks for Kinetic Data

- Kinetic Data Structures [BaschGuibasHershberger97]
  - Each point has a flight plan (algebraic expression)
  - Flight plans may change (with notice)
  - Computational structure (e.g. Delauney triangulation, lower envelope, etc.) is maintained online
  - Certificates guarantee boolean properties
  - Certificate failure times are computed and put in a priority queue. Rules are given to update the property on failure.
- Framework for sensor placement [NikoleteasSpirakis08]
  - Possible object trajectories are 3D curves over space and time
- Minimal sensor assumptions [GandhiKumarSuri08]
  - Sensors can count objects within their detection region

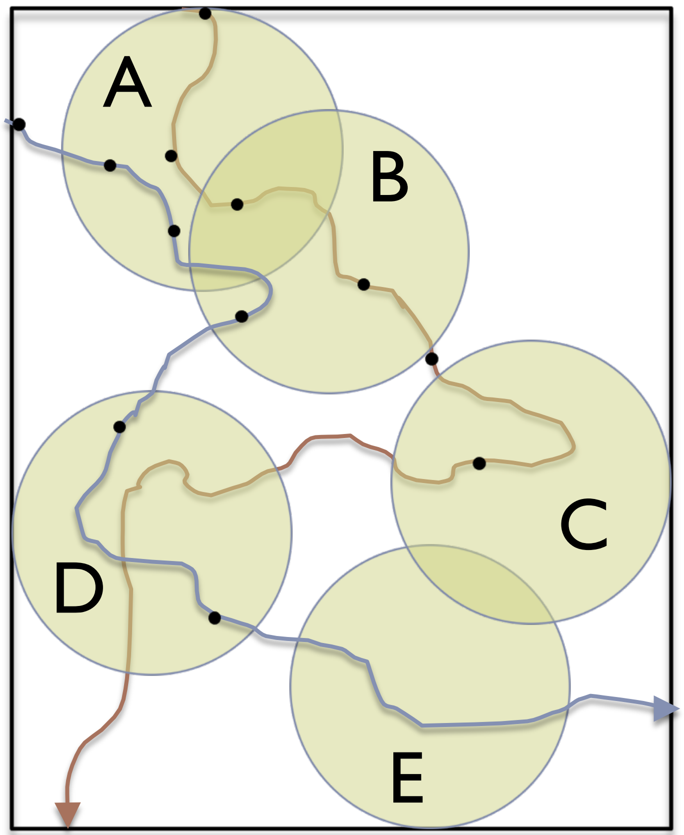
Our Framework

- Detection region around each sensor (stationary sensors)
- Point motion unrestricted
- No advance knowledge about motion
- Each sensor reports the count of points within its region at each synchronized time step
- <u>*k*-local</u>: Sensor outputs statistically only dependent on *k* nearest neighbors



Data Collection

Data based on underlying geometric motion



Sensor data streams

A	B	C	D	E
1	0	0	0	0
2	0	0	0	0
2	1	0	0	0
0	2	0	0	0
0	0	0	1	0
0	0	1	1	0

time

What is Optimal?

- Joint entropy chain rule ($\mathbf{X} = \{X_1, X_2, \ldots, X_S\}$):
  - $H(\mathbf{X}) = H(X_1)+H(X_2 | X_1)+\ldots+H(X_S | X_1,\ldots,X_{S-1})$

- *k*-local entropy (H_k): normalized joint entropy of a set of streams that are only dependent on up to *k* streams from their *k* nearest neighbors

- Optimal compression of sensor streams is $H(\mathbf{X}) = H_k(\mathbf{X})$

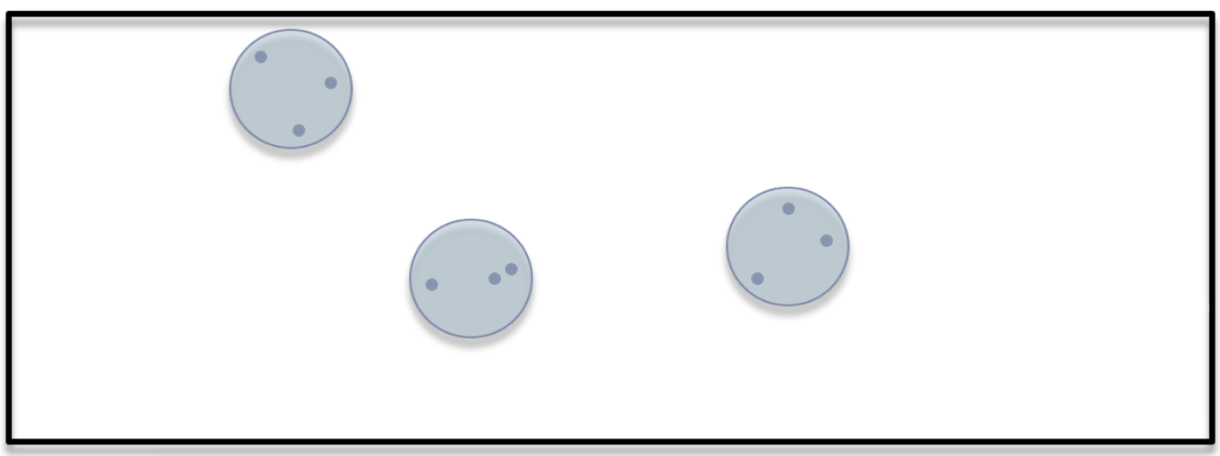
Data Compression Algorithm

- The optimal bound is the joint entropy of the set of streams
- Compressing each separately doesn't reach this bound
- Compressing all together reaches bound, but the window size necessary to achieve the needed repetition is too large to be practical
- Since $H(\mathbf{X})=H_k(\mathbf{X})$ we want groups of roughly k streams

Data Compression Algorithm: Partitioning Lemma

- *k*-clusterable: A point set that can be clustered into subsets of size at most $k+1$ so that if p and q are among each other's k nearest neighbors then they are in the same cluster.

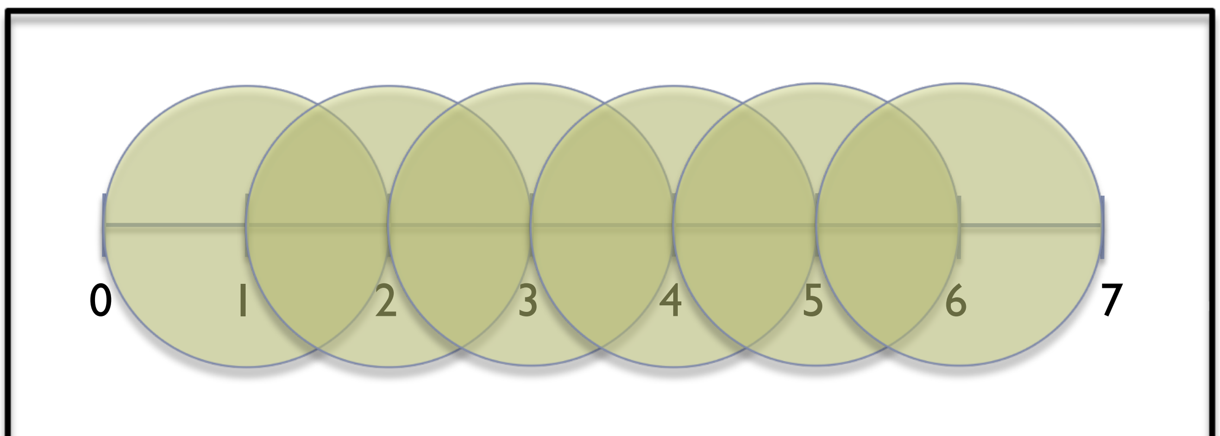
2-clusterable example



Data Compression Algorithm: Partitioning Lemma

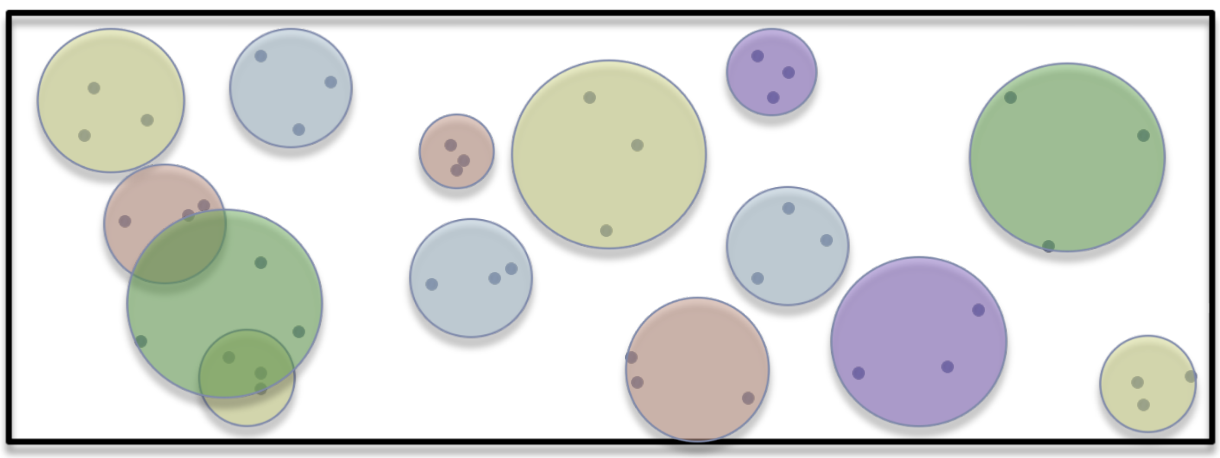
- *k*-clusterable: A point set that can be clustered into subsets of size at most $k+1$ so that if p and q are among each other's k nearest neighbors then they are in the same cluster.

not 2-clusterable example



Data Compression Algorithm: Partitioning Lemma

- Lemma: There exists an integral constant c such that for all $k>0$ any point set can be partitioned into c partitions that are each k-clusterable.



Partitioning Algorithm

for all points find

  $r_k(p)$ = distance from p to its k^{th} nearest neighbor

  $NN_k(p)$ = k nearest neighbors of p

while P is nonempty

  unmark all points in P

  create a new empty partition P_i

  while there are unmarked points

    r = minimum $r_k(p)$ for unmarked p

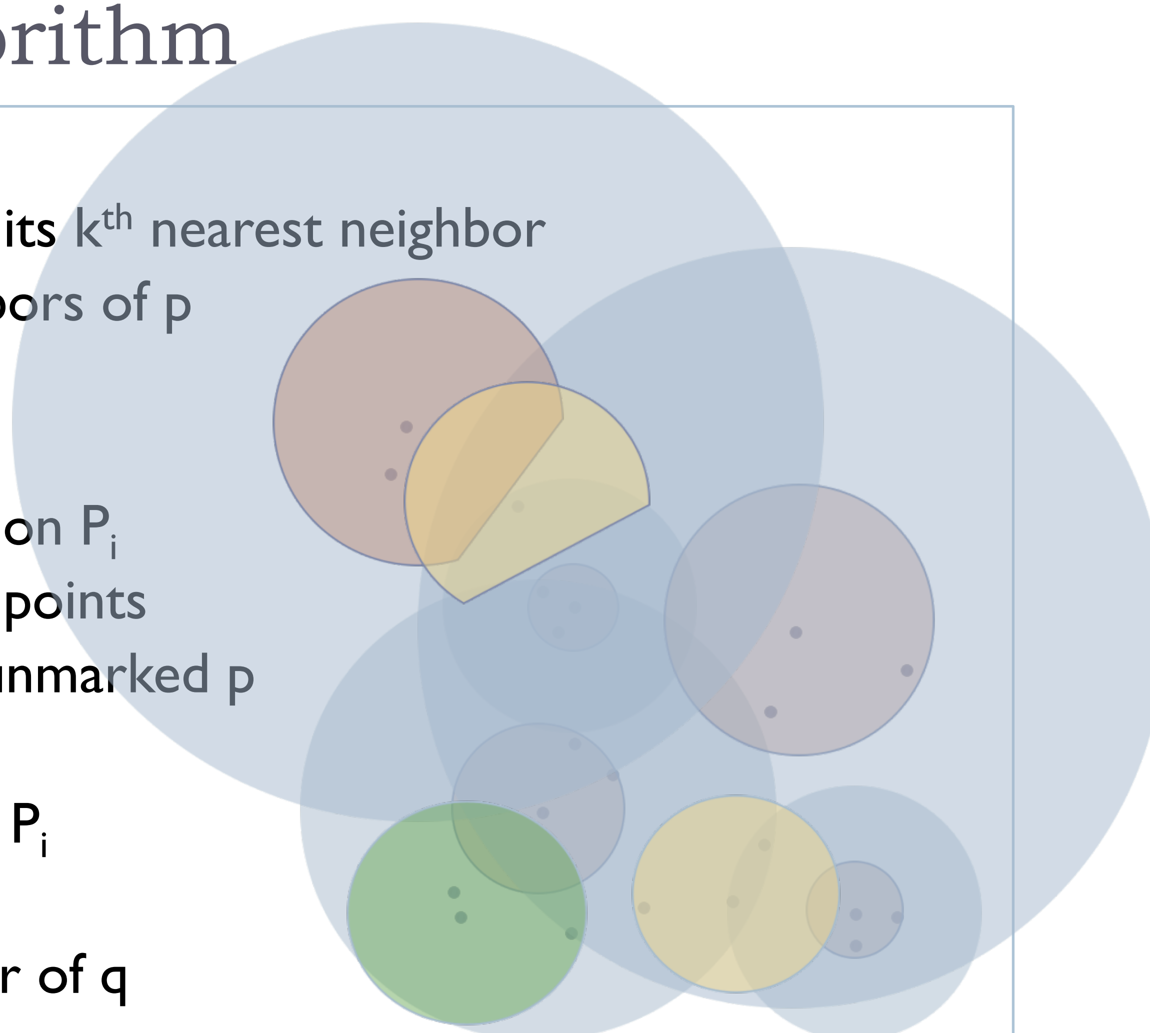
    q = point with $r_k(q)$ = r

    add q and its $NN_k(q)$ to P_i

      and remove from P

    mark all points within 3r of q

return $\{P_1, P_2, \ldots, P_c\}$



Partitioning Lemma Proof Sketch

- By nature of marking and order of clustering, all partitions are *k*-clusterable
  - No points in the partition are within 2r of a radius r cluster
  - Increasing $r_k(p)$ choices ensure non-mutual $NN_k(p)$ relations are separated into different clusters
- There are *c* partitions
  - In each round every point is either marked or removed from P
  - A point *p* is marked only by points within 12 $\min(\{r_k(p)\})$
  - Points that mark *p* are separated by distance $\min(\{r_k(p)\})$
  - Packing argument bounds the number of times a point can be marked to $c = O(1 + 12^{O(1)}) = O(1)$

Data Compression Algorithm

- Partition and cluster the sensors, then compress

```
for each partition P_i
    for each cluster in P_i
        combine the cluster's streams into
        one with longer characters
return the union of the compressed streams
```

1120…
1003…

(11)(10)(20)(03)…

- Proof Sketch:
  - The joint entropy of the streams is the optimal length
  - Sensor outputs are *k*-local, so each compressed partition is the optimal length
  - There are *c* partitions, so the total length is *c* times optimal

Summary of Results

- Framework for kinetic sensor data
  - No assumptions about object motion or advance knowledge
  - Motion sensitive analysis
  - Relies on minimal sensor abilities

- Lossless compression algorithm that compresses the data to c H(X), which is O(optimal)
  - Assumes the sensor outputs are only dependent on their k nearest neighbors
  - Assumes the sensor outputs can be modeled by an underlying random process

Recent and Future Work

- Extend analysis of compression algorithm to consider empirical entropy (no underlying random process) ✓
- Retrieval without decompressing the data ✓
  - Range searching
  - E.g. given a time period and spatial range, what is the aggregated count?
- Statistical analysis without decompressing the data
- Lossy compression
- Experimental evaluation
- Application in non-sensor contexts

Thank you!

Questions?