Catching Elephants with Mice

Sparse Sampling for Monitoring Sensor Networks

S. Gandhi, S. Suri, E. Welzl
Outline

- Introduction
- VC-Dimension and $\varepsilon$-nets
- Catching Elephants ...
  - ... in Theory
  - ... in Practice
- Simulation Results
- Conclusion & Discussion
Outline

- Introduction
- VC-Dimension and $\varepsilon$-nets
- Catching Elephants ...  
  - ... in Theory  
  - ... in Practice
- Simulation Results
- Conclusion & Discussion
Introduction

- **Sensor Networks**
  - Ideally: tiny, inexpensive, allowing real-time and fine-grained monitoring
  - Applications
    - Mostly surveillance or environmental monitoring
    - e.g. tracking pollution level in a habitat

- **Issues**
  - Local and temporal variations
  - Natural faults, adversarial attacks
Introduction – Our Goal

- Detect significant events
  - Monitoring only a small subset of all nodes
  - Using a scheme that scales relatively

- Estimate the size of these events

- Terminology
  - Elephants: "large" events, defined by what fraction $\varepsilon$ of the network is affected
  - Mice: the monitoring set we use to detect ("catch") these elephants
Introduction – Assumptions

- No low level issues
  - Reliable communication from nodes to base station
  - Idealized sensing

- Base station knows locations of all sensors
  - But no assumptions over the distribution

- At most one event at any time

- Event-geometry can be described in "nice" ways
  - → Vapnik-Chervonenkis dimension
Outline

- Introduction
- VC-Dimension and \( \varepsilon \)-nets
- Catching Elephants ...
  - ... in Theory
  - ... in Practice
- Simulation Results
- Conclusion & Discussion
VC-Dimension

- Why is a circle simpler than a rectangle?
- \((X, R)\) is called a range space
  - \(X\) is a ground set
  - \(R\) is a set of subsets (ranges) of \(X\)
- Definitions
  - \(A \subseteq X\) is shattered by \(R\) if all possible subsets of \(A\) can be obtained by intersecting \(A\) with an \(r \in R\)
  - The VC-Dimension of \((X, R)\) is the cardinality of the largest set \(A\) that can be shattered by \(R\)
VC-Dimension

- Range spaces with infinite VC dimension
  - e.g. if \( R \) is the set of all convex polygons
- Relevance for our scheme
  - We will see that we can choose a set of
    \[ O\left(\frac{d \log d}{\varepsilon} \log \frac{d \log d}{\varepsilon}\right) \]
    mice for our scheme to work
  - Thus for events that can be approximated by simple geometric shapes of \textbf{constant} VC-dimension the sample size is
    \( O\left(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon}\right) \)
ε-nets

- **Definition**
  - $B \subseteq X$ is an $\varepsilon$-net for $X$ if an event $r \in R$ that affects $\geq \varepsilon|X|$ nodes also affects $B$ (i.e. $r \cap B \neq \emptyset$)

- **Construction**
  - If we choose $m \geq \max \left( \frac{8d}{\varepsilon} \log \frac{8d}{\varepsilon}, \frac{4}{\varepsilon} \log \frac{2}{\delta} \right)$ nodes from the network at random, we will, with probability of $(1 - \delta)$, have an $\varepsilon$-net
  - $\Rightarrow O\left(\frac{d}{\varepsilon} \log \frac{d}{\varepsilon}\right)$ nodes
**ε-nets**

- Looking at the definition, ε-nets seem to be just the right tool for our problem

- Two major drawbacks if applied directly
  - False alarms
  - No size estimation

- Using ε-nets in another way, we can remedy both problems without increasing the asymptotic size
Outline

- Introduction
- VC-Dimension and $\varepsilon$-nets
- Catching Elephants ...
  - ... in Theory
  - ... in Practice
- Simulation Results
- Conclusion & Discussion
Catching Elephants in Theory

- The symmetric difference
  - \( D_1 \oplus D_2 := (D_1 \cup D_2) \setminus (D_1 \cap D_2) \)

- If \((X, R)\) has VC dimension \(d\)
  - \(R' := \{r_1 \oplus r_2 \mid r_1, r_2 \in R\}\)
  - Then \((X, R')\) has VC dimension \(d' := O(d \log d)\)

- Now we can use the following algorithm
  - \(S\): the set of all sensors in our network
  - \(d\): the maximum VC dimension of the elephants
  - \(\varepsilon\): the fraction that defines if an event is an elephant
Catching Elephants in Theory - Algorithm

- **CatchElephants**(S, d, ε)
  1. d' := O(d log d)
  2. Construct ε/4-net M for S (the mice)
      - w.r.t. the symmetric difference ranges of dimension d'
  3. Let T ⊆ M be the "dead mice"
  4. Compute a disk D
      - containing T and excluding M \ T
  5. K := |S ∩ D| sensors lie inside D
      - If K ≥ 3εn / 4, then the event is an elephant of size K
      - Otherwise the event is not an elephant
Catching Elephants in Theory - Essence

- Constructing the special $\varepsilon/4$-net
  - The number of nodes in $D \oplus D^*$ is at most $\varepsilon n/4$
  - Size approximation error of $\pm \varepsilon n/4$

- Checking if $K \geq 3\varepsilon n / 4$
  - Two-sided guarantee
    - Every elephant is reported
    - The algorithm never reports events of size $\leq \varepsilon n/2$
  - False alarms only for events of size $(\varepsilon n/2, \varepsilon n)$
    - This is the approximation gray zone
Catching Elephants in Practice

- Estimating Theoretical Pessimism
  - Determine empirically what's "good enough" in practice

- Redundancy-aware Sampling
  - Choose the mice not too close to each other

Lake (2500 nodes)
Catching Elephants in Practice

![Graph showing monitoring set size vs. error for theoretical number, random sampling, and RA sampling.](image-url)
Outline

- Introduction
- VC-Dimension and \( \varepsilon \)-nets
- Catching Elephants ... 
  - ... in Theory
  - ... in Practice
- Simulation Results
- Conclusion & Discussion
Simulation Results

- **Parameters**
  - 5 different networks with $n \in [1'000, 45'000]$  
  - Geometries: circles, ellipses, axis-aligned rectangles  
  - Event-size chosen randomly in $[0.1n, 0.3n]$  

- 2000 events for each of the 15 pairs of datasets and event geometries $\Rightarrow$ 30'000 tests

- Monitoring sets constructed by using Redundancy-aware sampling
Simulation Results
Simulation Results

- Number of mice vs. total number of nodes
Outline

- Introduction
- VC-Dimension and $\varepsilon$-nets
- Catching Elephants ...
  - ... in Theory
  - ... in Practice
- Simulation Results
- Conclusion & Discussion
Conclusion

- The idea of using a random sample is not particularly novel
- Core contributions
  - Quantifying the sample size, using the VC dimension
  - Bridging the gap between theory and practice
- Key idea of the scheme
  - Using ε-nets w.r.t. symmetric difference ranges
  - two-sided guarantee and size estimation
Discussion – Personal Thoughts

- Interesting ideas
- Sometimes confusing
  - Mixing of asymptotic and plain terms
  - Some (core) details not explained too thoroughly
- Practical focus is notable
  - Why no simulations with polygons?
  - "Quantifying": Most terms are asymptotic
  - "Bridging the gap": Maybe the bridge was built from the theory side
Discussion – Your Turn

- Comprehension questions
- Your opinion
  - Do you think this results are useful?
  - Or do you see difficulties in practical applications?

- How would you extend the scheme for a network with nodes of different importance?