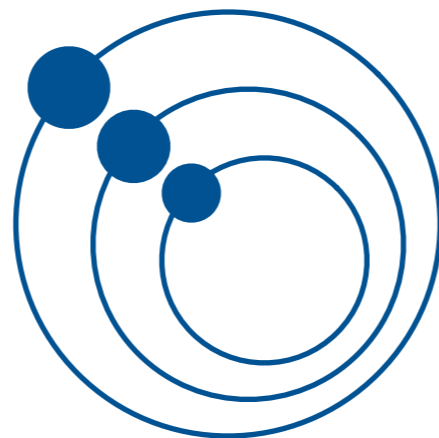


# Community Epidemic Detection Using Time-Correlated Anomalies

Adam J. Oliner, Ashutosh V. Kulkarni, and Alex Aiken  
Department of Computer Science  
Stanford University



# Today's Talk

---

1. Intrusion Detection
2. *Syzygy*
3. Deployments



(1)

# Intrusion Detection

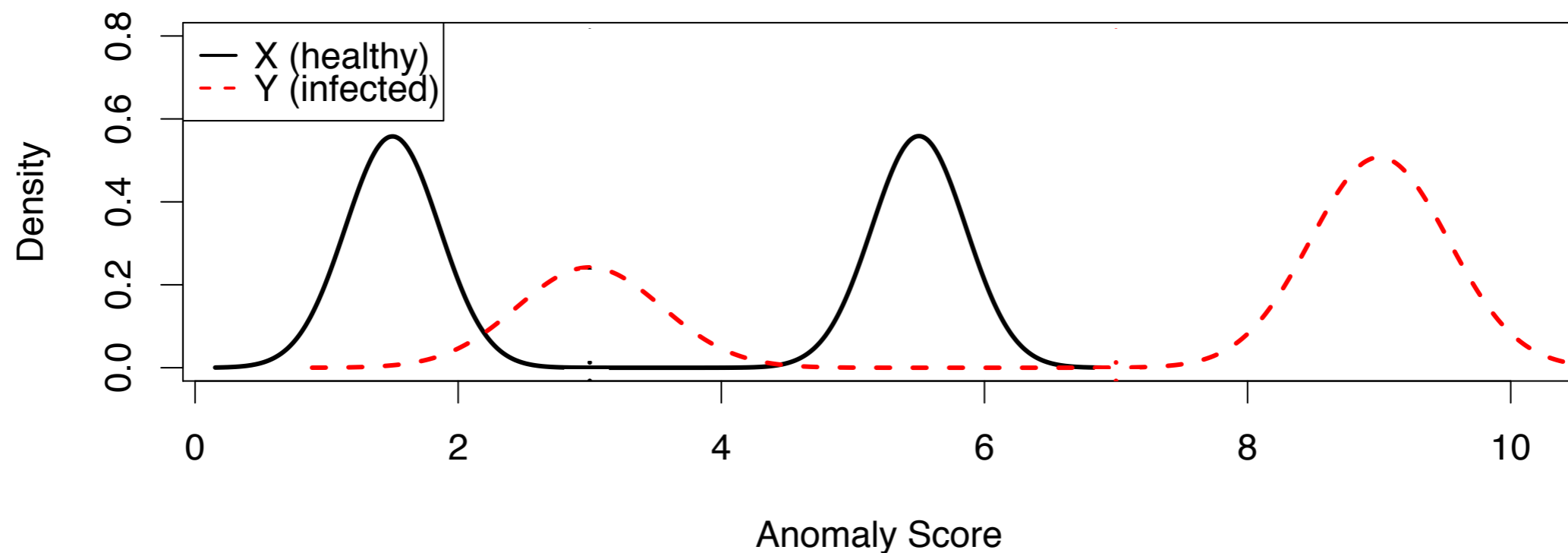
# Client Detection



- Model generates *anomaly scores*
- Generate an alarm (✗) when the score exceeds some threshold
- e.g., [Debar92][Eskin00][Feng03][Forrest96][Gao04][Gao06][Giffin02][Hofmeyr98][Javitz91][Malan05][Mutz06][Paxson99][Porras97][Sebring88][Smaha88][Tan02][Vaccaro89][Xie04]

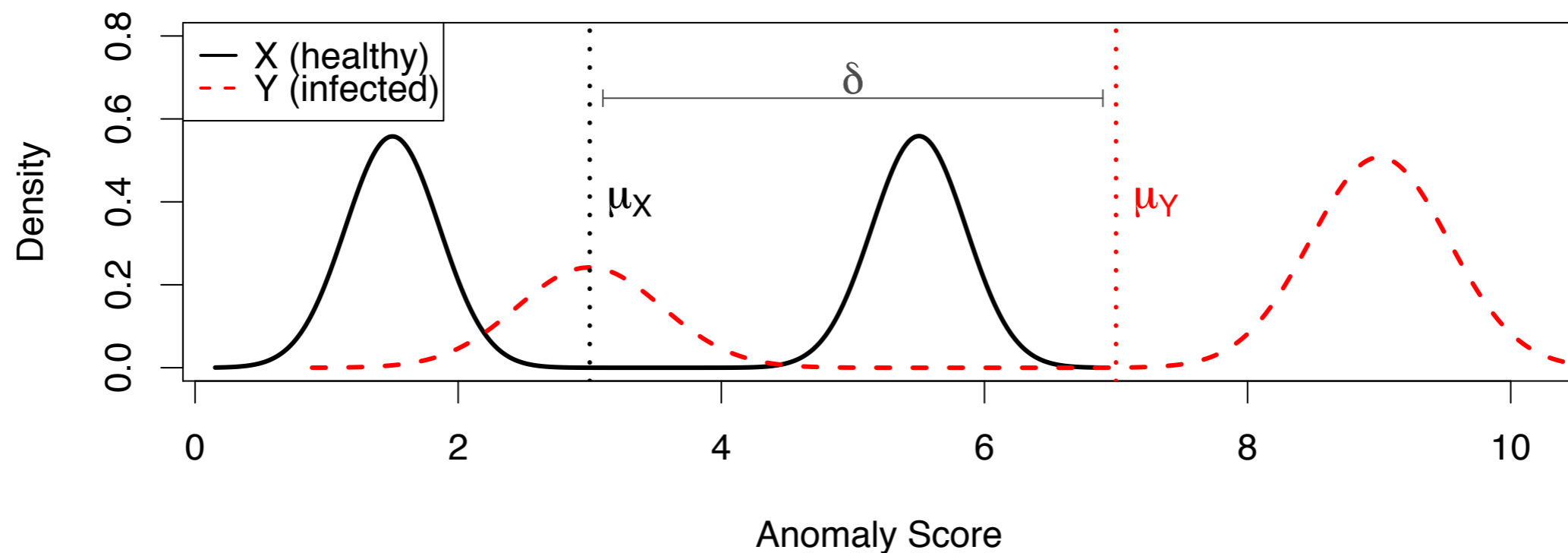
# Anomaly Scores

- Average infected score is higher
- There is no perfect threshold



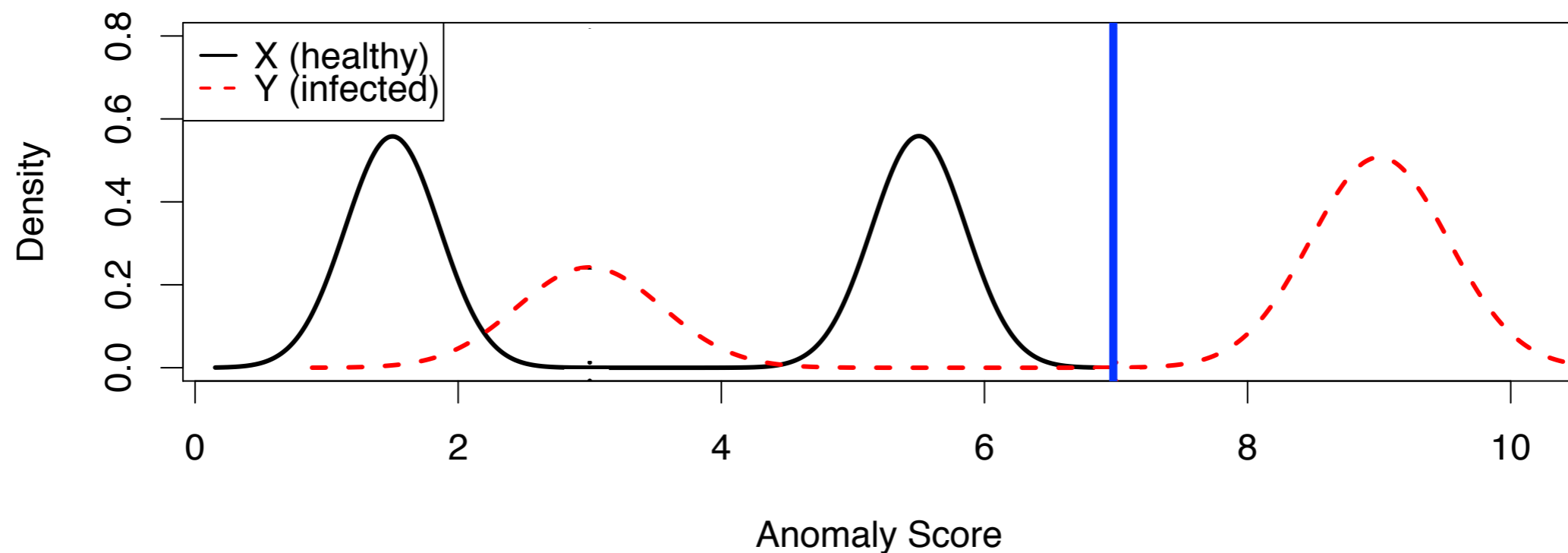
# Anomaly Scores

- Average infected score is higher
- There is no perfect threshold



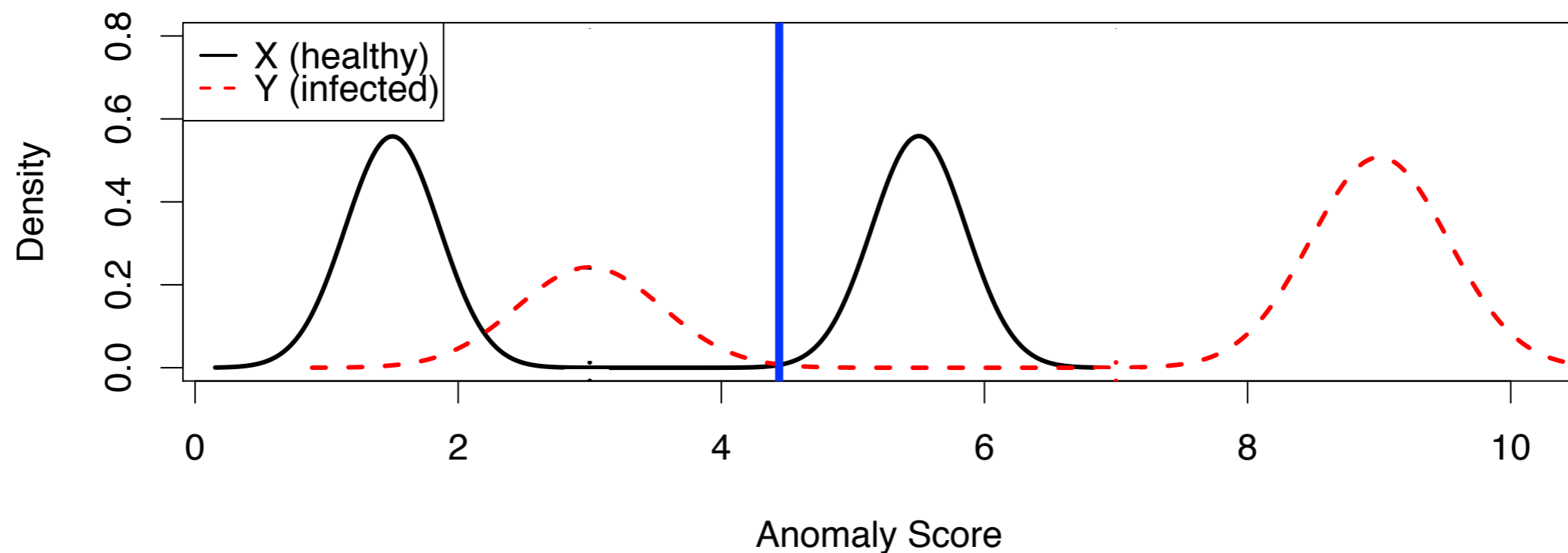
# Anomaly Scores

- Average infected score is higher
- There is no perfect threshold



# Anomaly Scores

- Average infected score is higher
- There is no perfect threshold



# Community

---

- Set of instances of an application
  - “all Firefox browsers in a company”
  - “all Apache servers in a cluster”
- Each client generates anomaly scores
- Are  $>0$  infected? (*epidemic*)



# Previous Approaches

① → ✓/✗

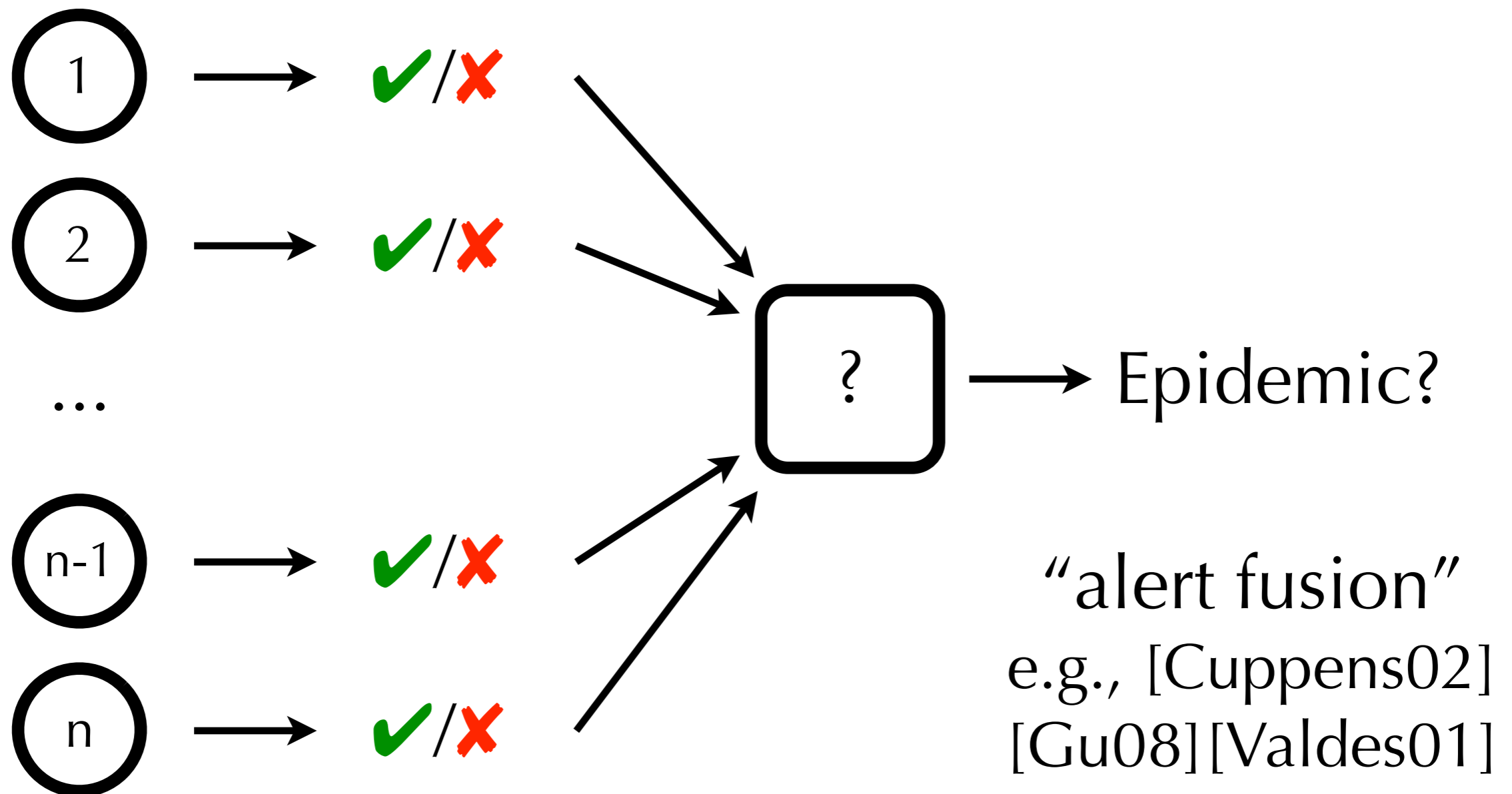
② → ✓/✗

...

③ → ✓/✗

④ → ✓/✗

# Previous Approaches



(2)

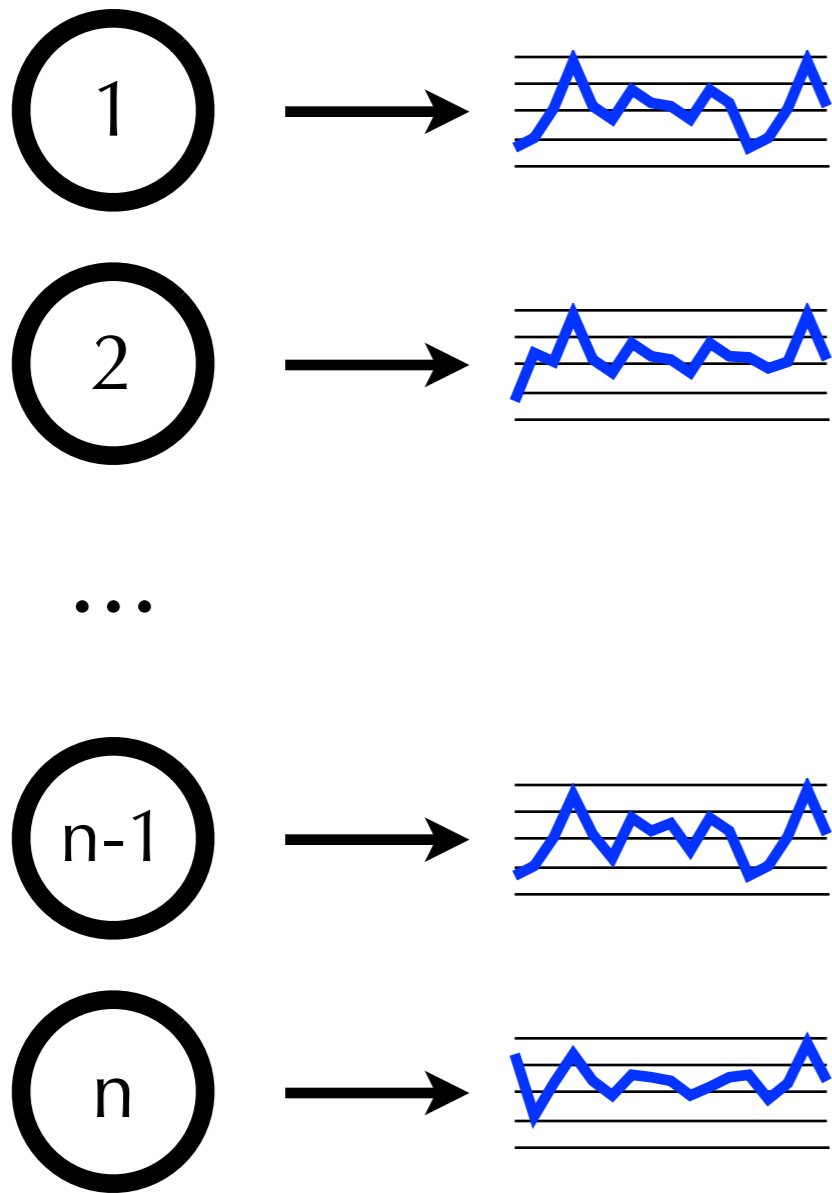
*Syzygy*

# Syzygy

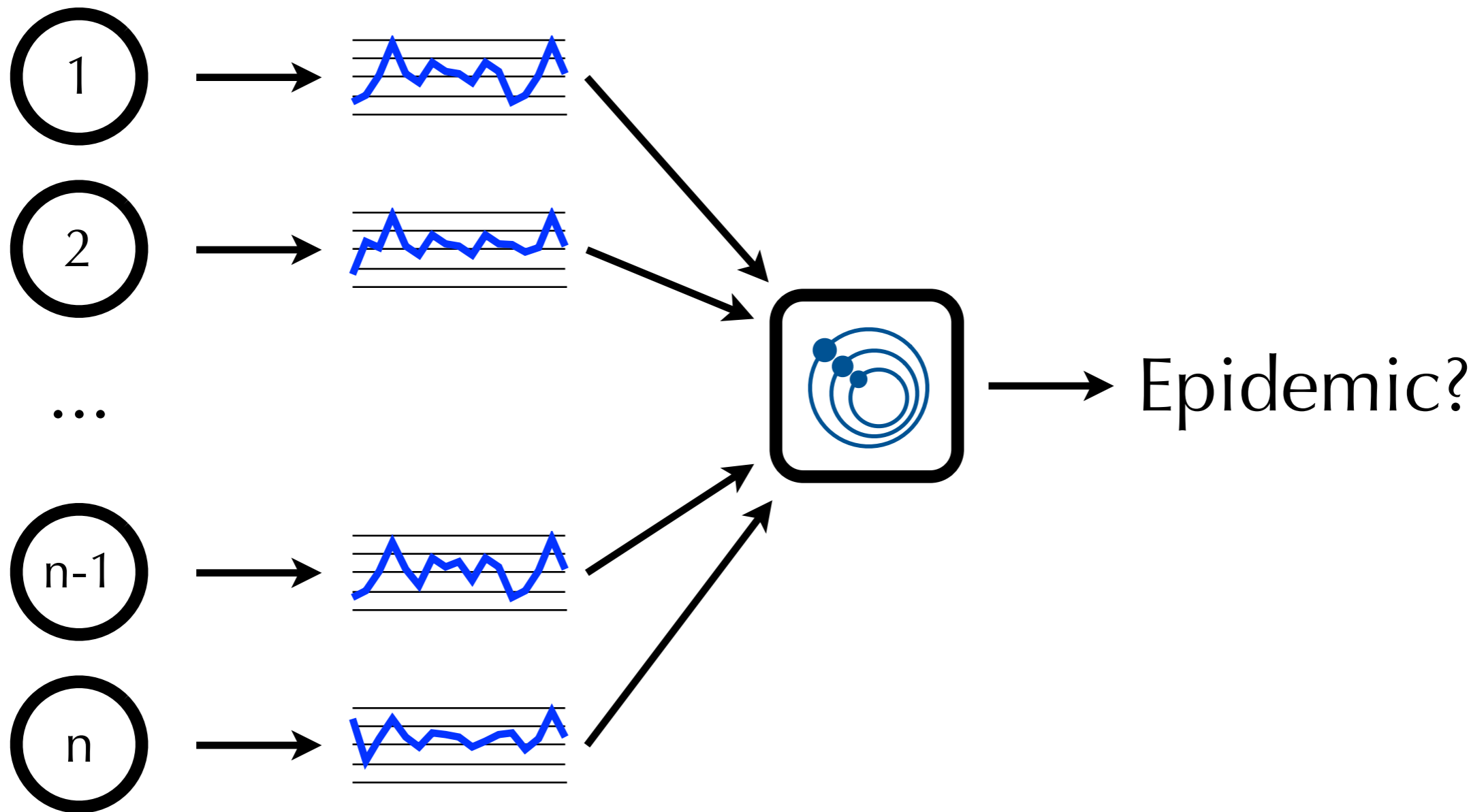
1	→	0.47
2	→	0.28
...		
n-1	→	1.13
n	→	0.19



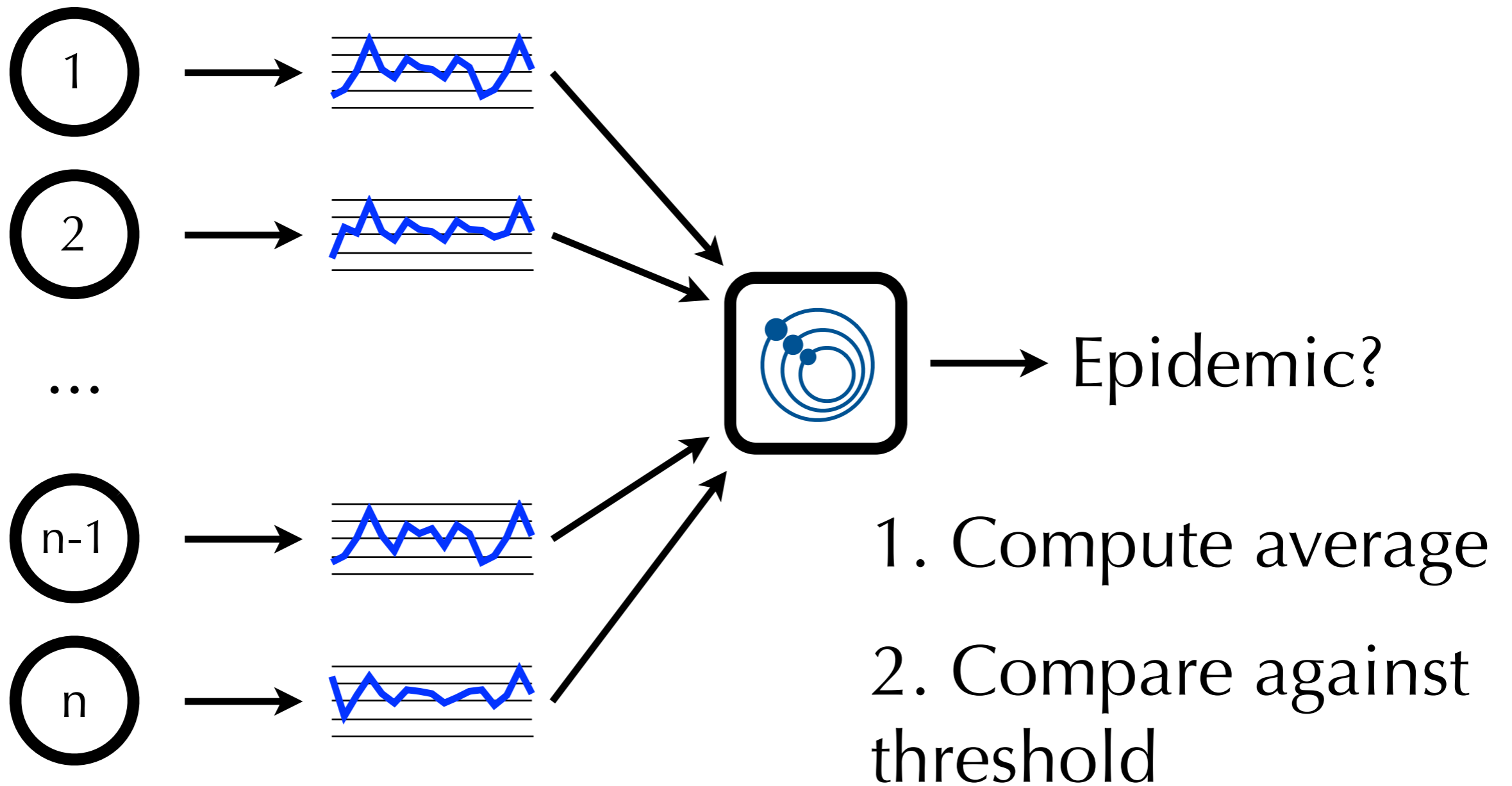
# Syzygy



# Syzygy

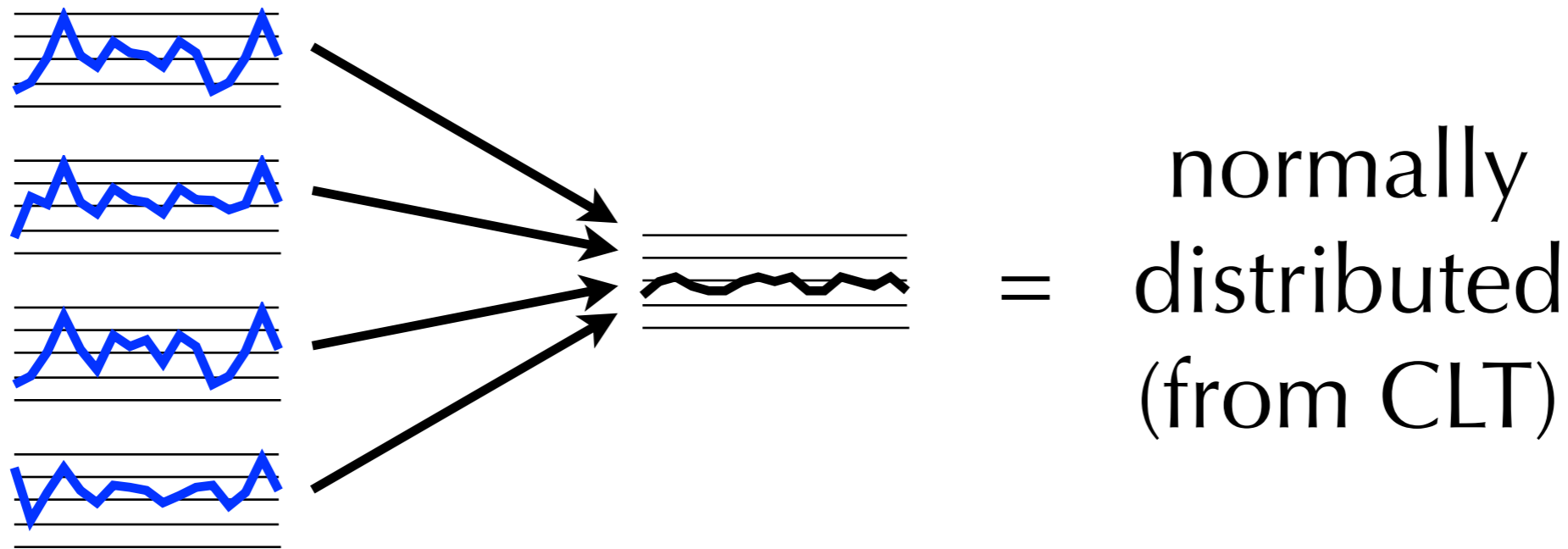


# Syzygy



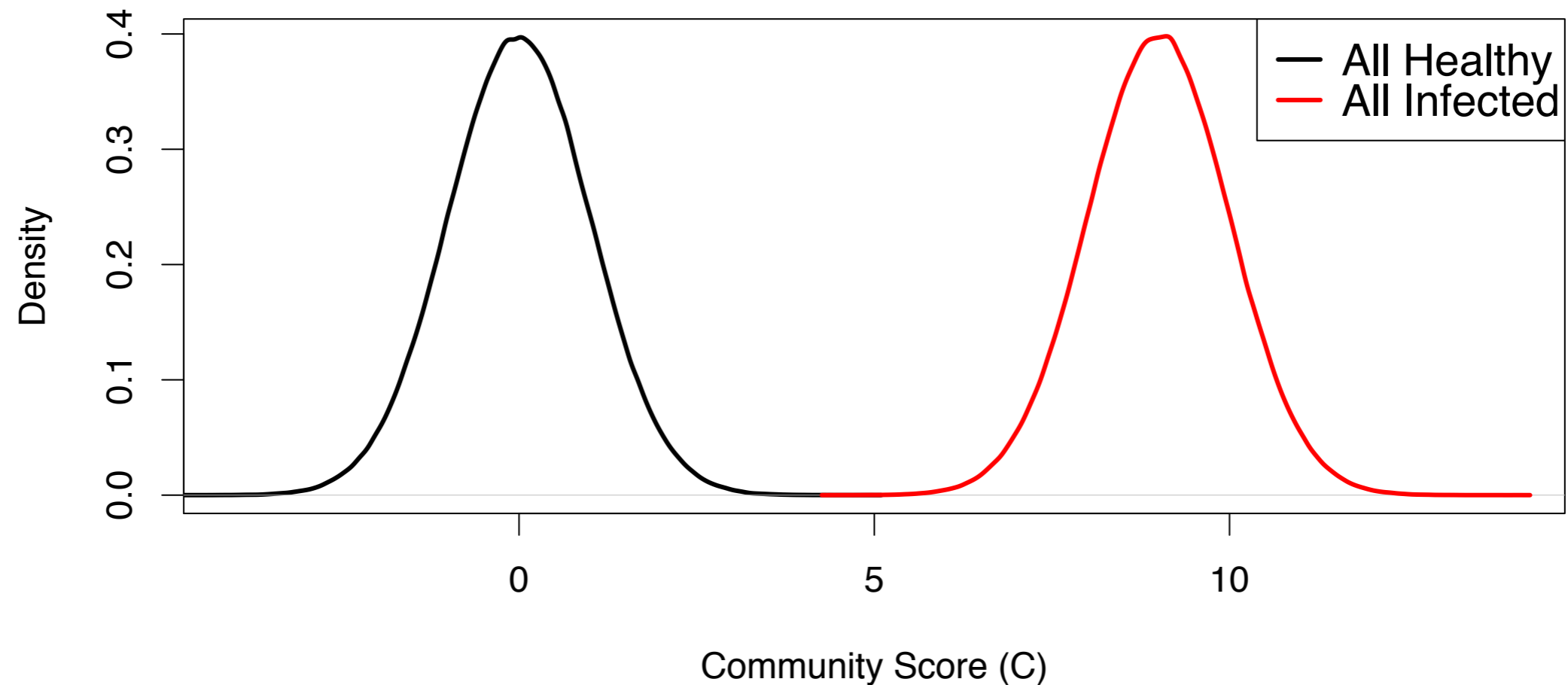
# Community Score

- Average of current anomaly scores
- $n$  i.i.d. random variables



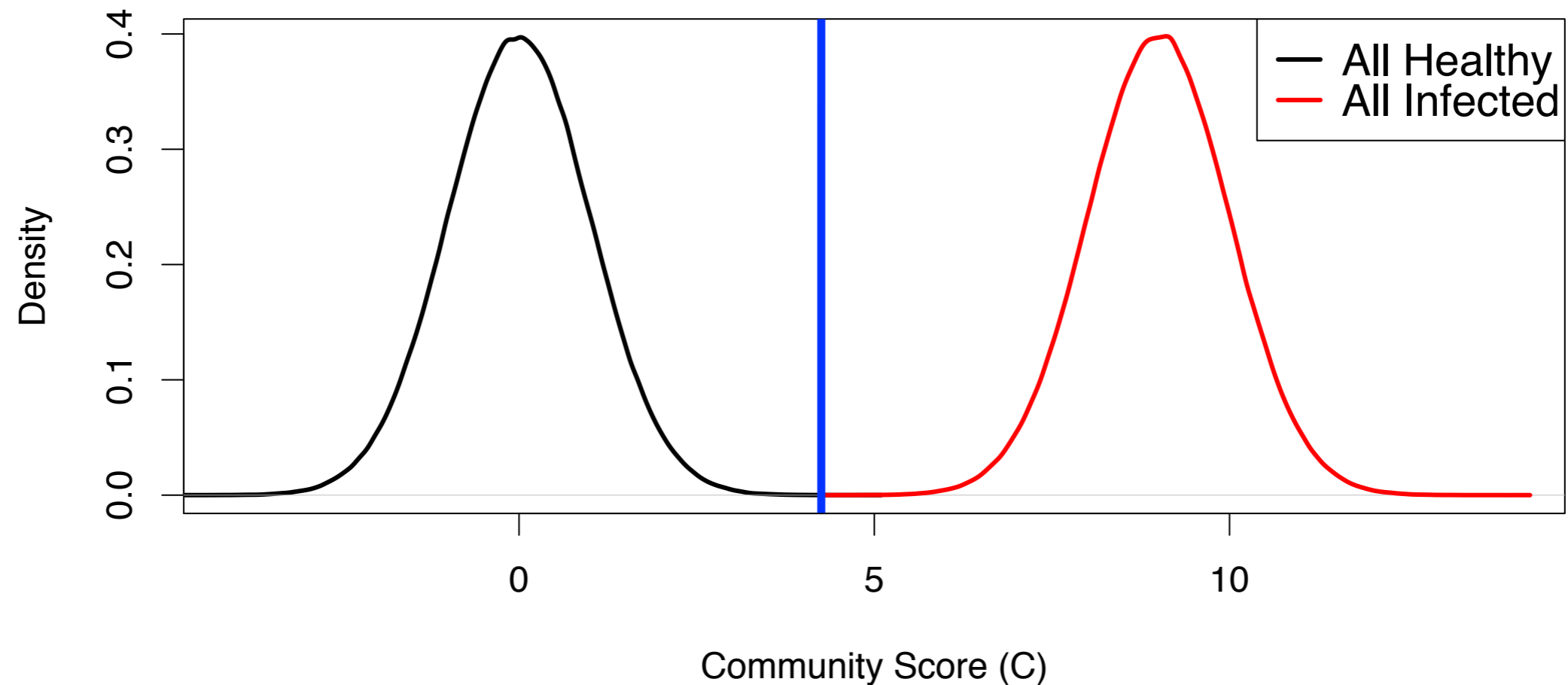
# Community Score

- Clearer separation



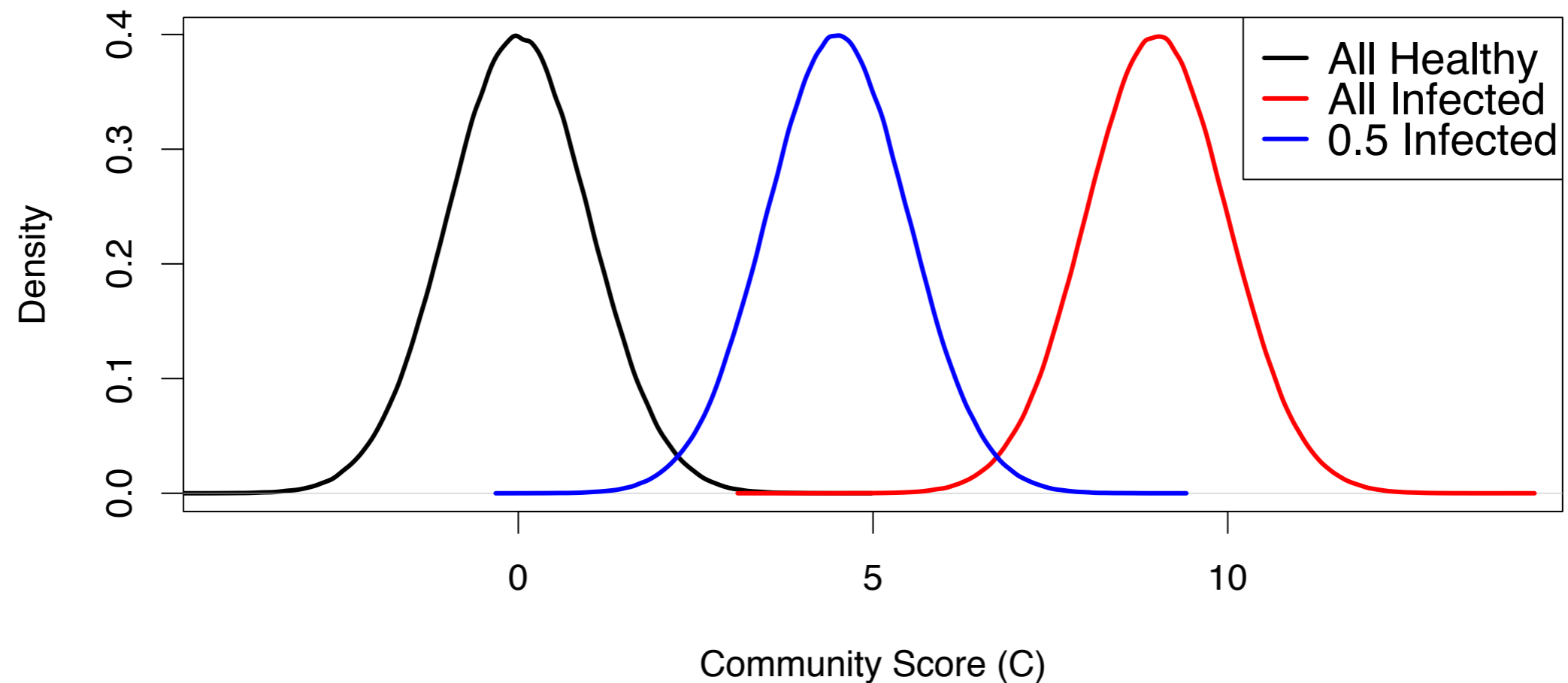
# Community Score

- Clearer separation



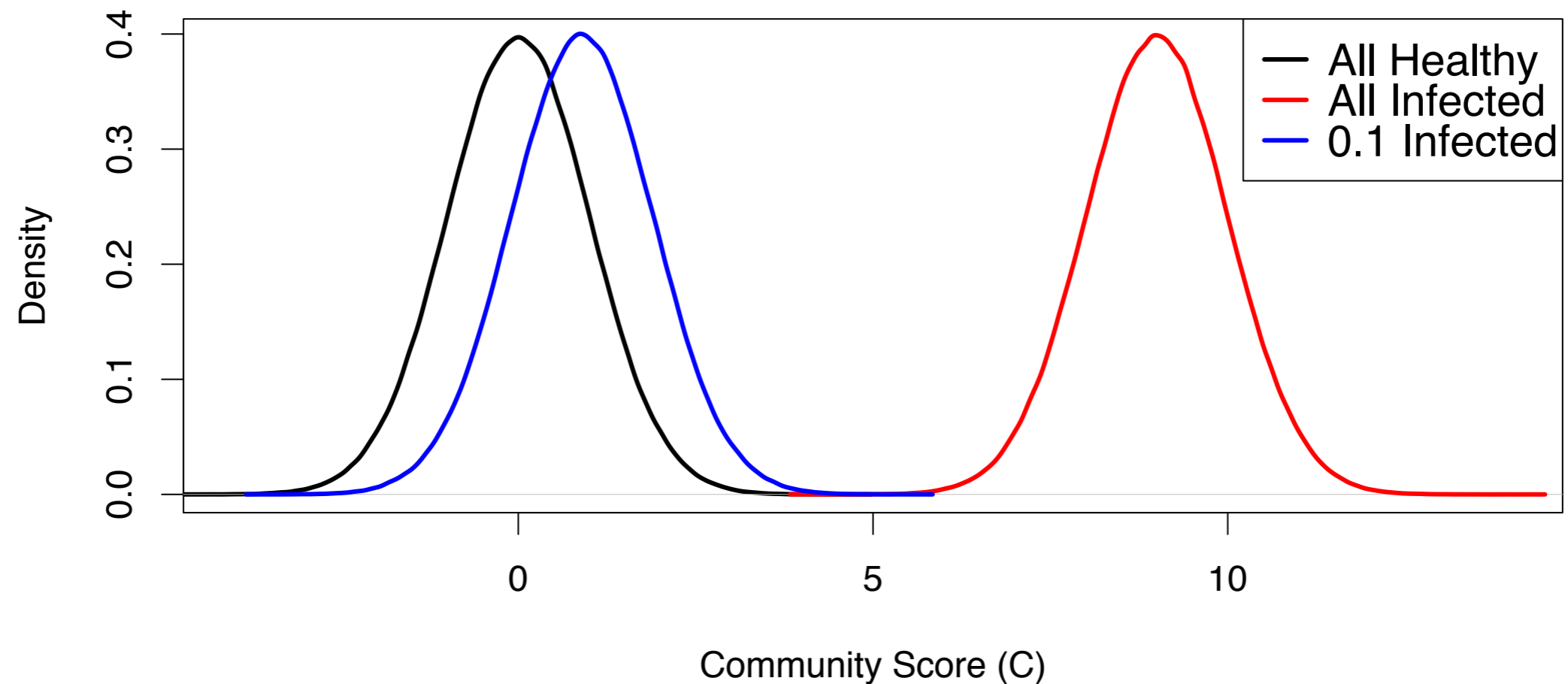
# Community Score

- Clearer separation



# Community Score

- Clearer separation



# Properties

---

- Weak requirements
  - Independence
  - Measurable effect
- Strong theoretical guarantees
  - Tunable false positive rate
  - Asymptotically perfect detection



(3)

Deployments

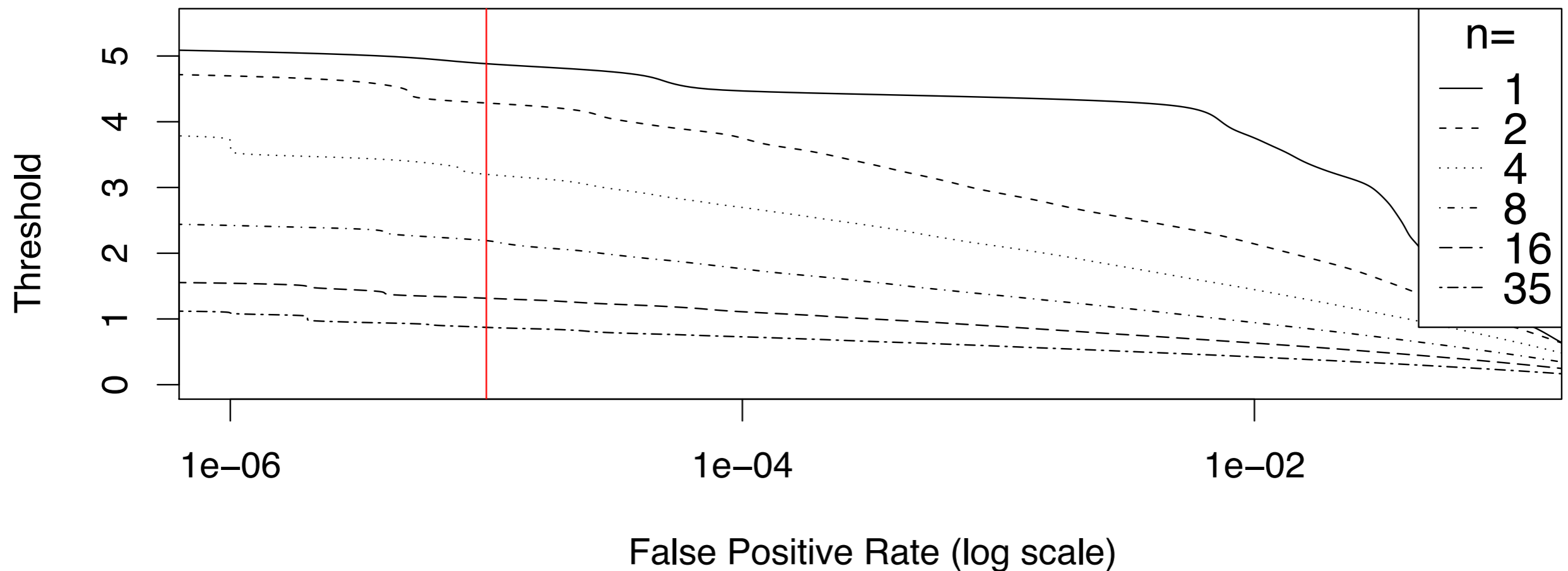
# Deployment #1

- Firefox in Stanford CS Building
- 35 clients over two weeks
- Model uses distributions of system call sequences
- > Ten billion anomaly scores
- No false positives



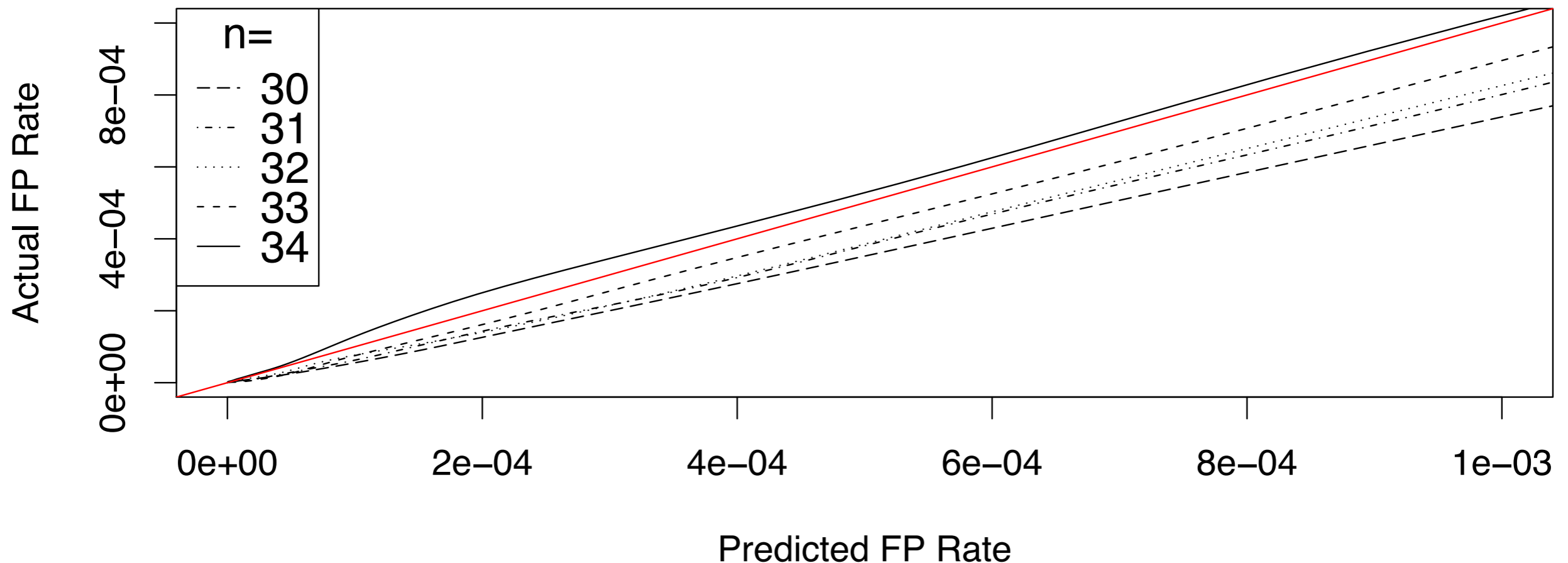
# Setting FP Rate

- Automatically sets the threshold



# Threshold Accuracy

- Can pick FP rate in a live deployment



# Deployment #2

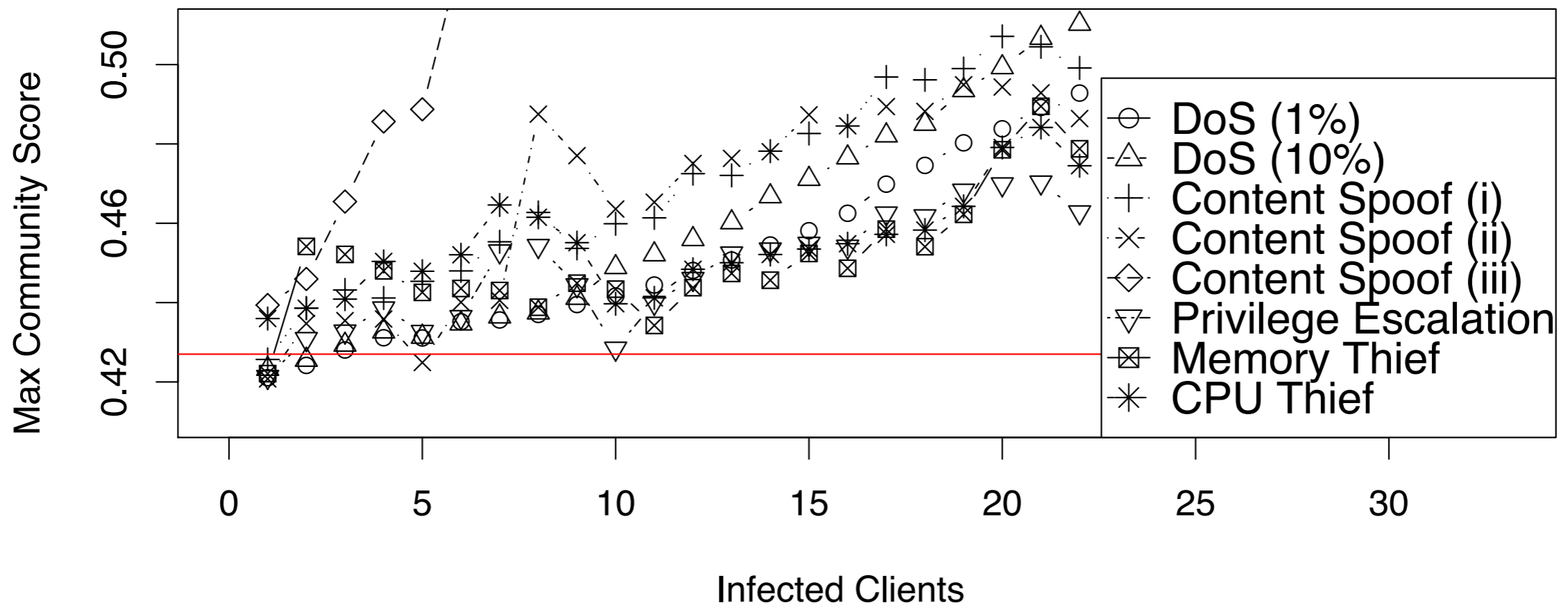
---

- Apache server cluster
- 22 clients driven by workload generator
- Model uses response time distributions
- Attacks: DoS, resource exhaustion, content spoofing, privilege escalation



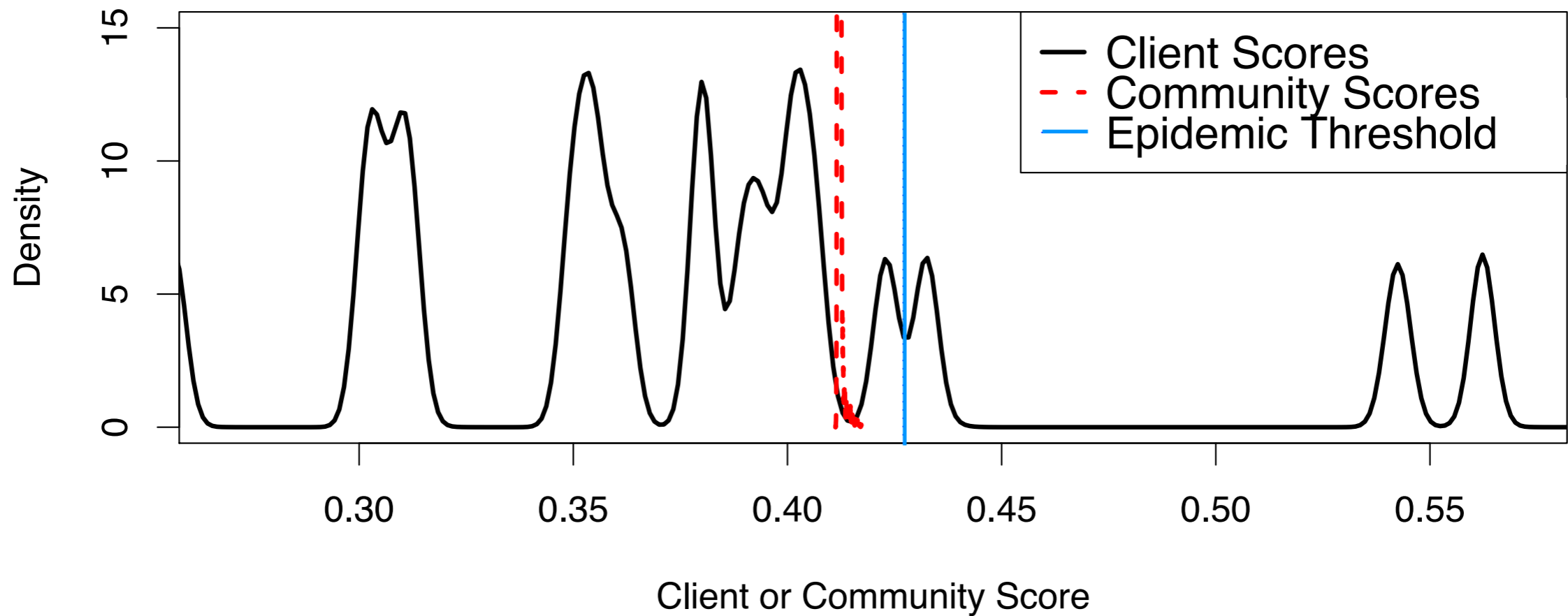
# Detection

- All attacks detected, no false positives



# Convergence

- Even at low  $n$  (22)



# Contributions

---

- Proposed Syzygy epidemic detector
- Proved strong guarantees
- Verified robustness claims
- Demonstrated practicality



# Fin

---

You have reached the end of the presentation. Please turn back.



# Data Sets

---

- Four desktop applications
- Four exploits
  - More powerful than those in the wild  
(in a technical sense, see paper)
- Model uses system call sequences



# Experiments

---

- Attack detection
- Tainted training data
- Mimicry
- Client variation
- Parameter sensitivity



# Simulation Summary

*Syzygy* is robust to noisy or incomplete client models, tainted training data, exploits that exhibit advanced behavior like mimicry, and a variety of other adverse conditions.

