Outside the Closed World: On Finding Intrusions with Anomaly Detection

Robin Sommer

International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org

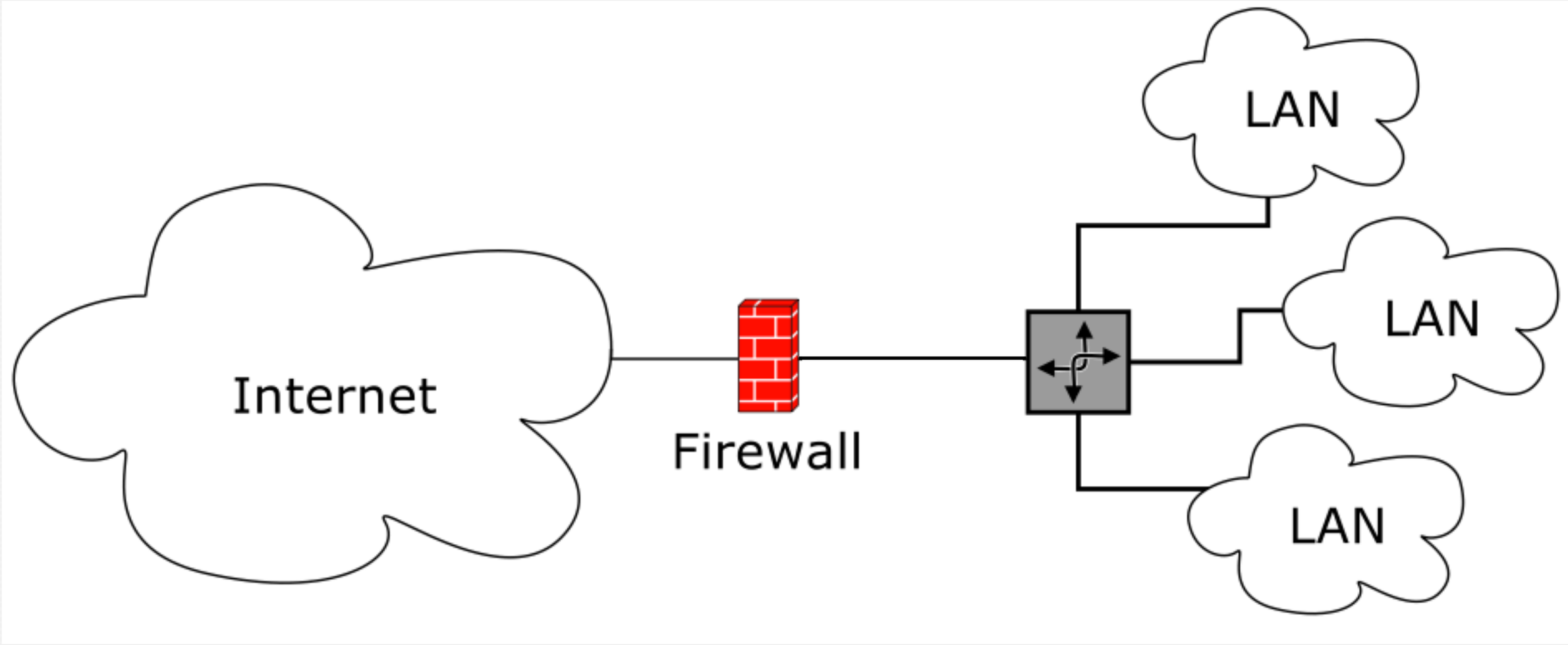
Security Seminar
UC Davis

June 2011

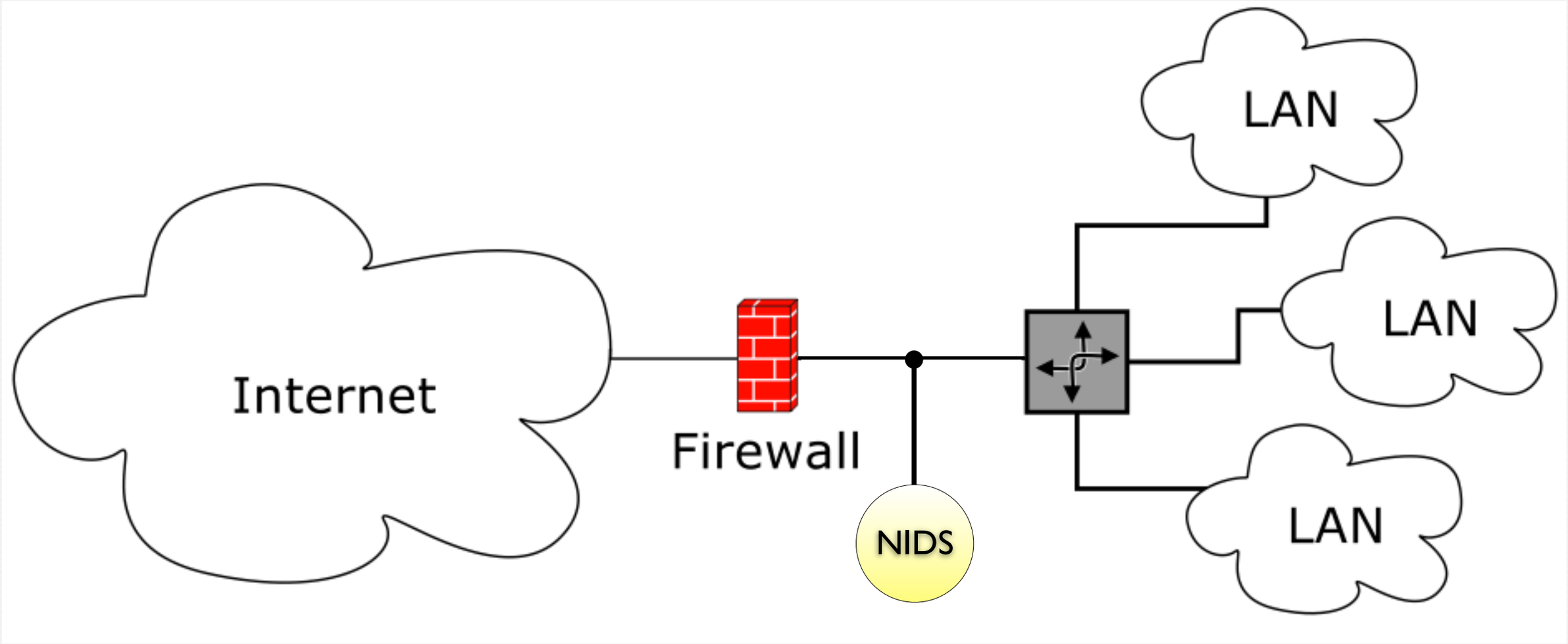
Monitoring For Intrusions

- Too many bad folks out there on the Net.
  - Constantly scanning the Net for vulnerable systems.
  - When they mount an attack on your network, you want to know.
- Operators deploy systems that monitor their network.
  - Intrusion detection or intrusion prevention systems (IDS/IPS).
- Key question: *How* does an IDS find the attack?

Achieving Visibility



Achieving Visibility



How Can an IDS Find Attacks?

Misuse detection (aka signature-/rule-based)

Searching for what we *know* to be bad.

How Can an IDS Find Attacks?

Misuse detection (aka signature-/rule-based)

Searching for what we *know* to be bad.

Snort Signature Example

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139
    flow:to_server,established
    content:"|eb2f 5feb 4a5e 89fb 893e 89f2|"
    msg:"EXPLOIT x86 linux samba overflow"
    reference:bugtraq,1816
    reference:cve,CVE-1999-0811
    classtype:attempted-admin
```

Anomaly Detection

Aims to find novel, previously unknown attacks

Anomaly Detection

Aims to find novel, previously unknown attacks

Assumption: Attacks exhibit characteristics different from normal traffic, for a suitable definition of normal.

Anomaly Detection

Aims to find novel, previously unknown attacks

Assumption: Attacks exhibit characteristics different from normal traffic, for a suitable definition of normal.

Detection has two components:

(1) Build a profile of normal activity (often offline).
(2) Match activity against profile and report what deviates.

Anomaly Detection

Aims to find novel, previously unknown attacks

Assumption: Attacks exhibit characteristics different from normal traffic, for a suitable definition of normal.

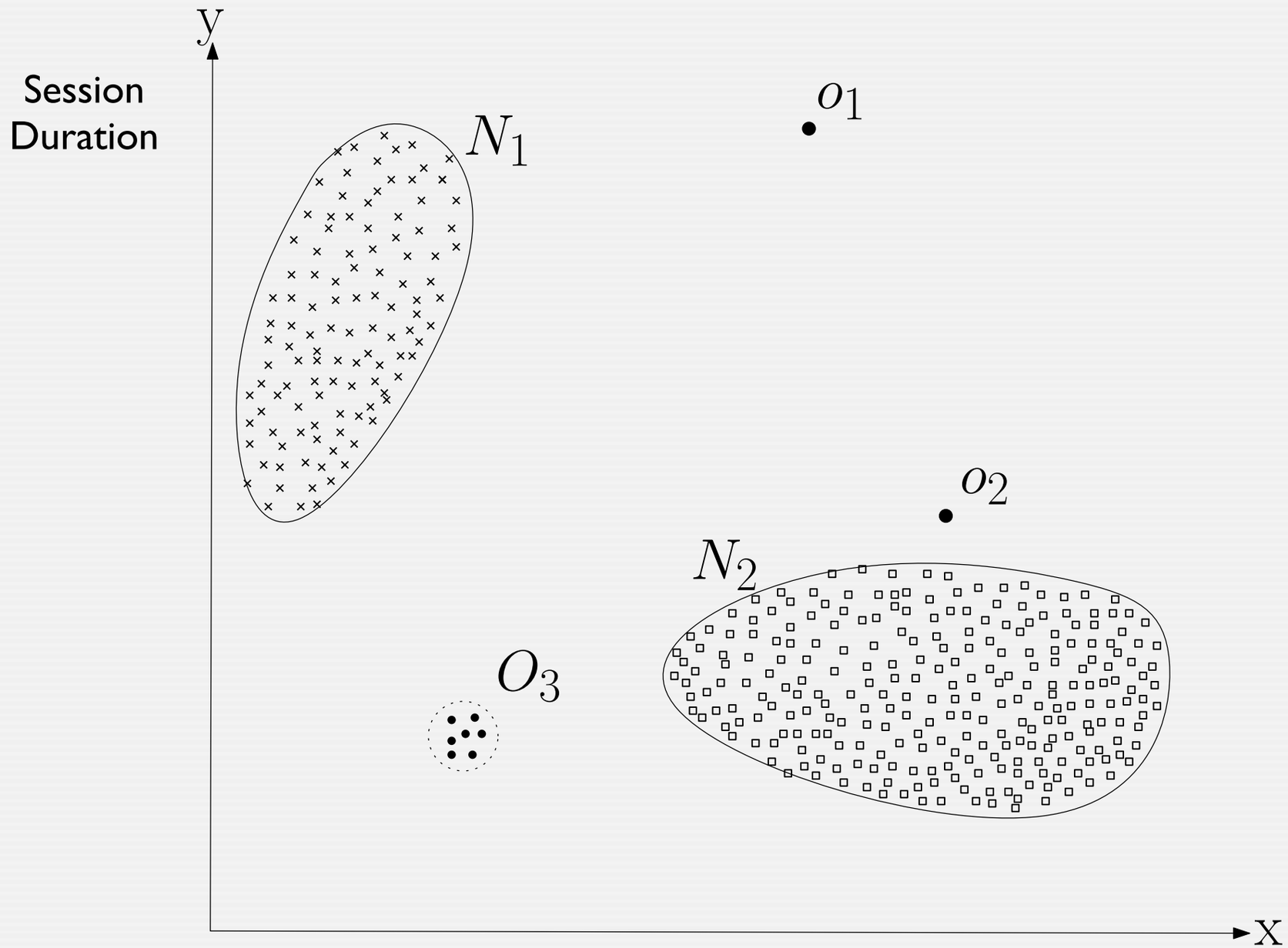
Detection has two components:

(1) Build a profile of normal activity (often offline).
(2) Match activity against profile and report what deviates.

Originally introduced by Denning's IDES in 1987:

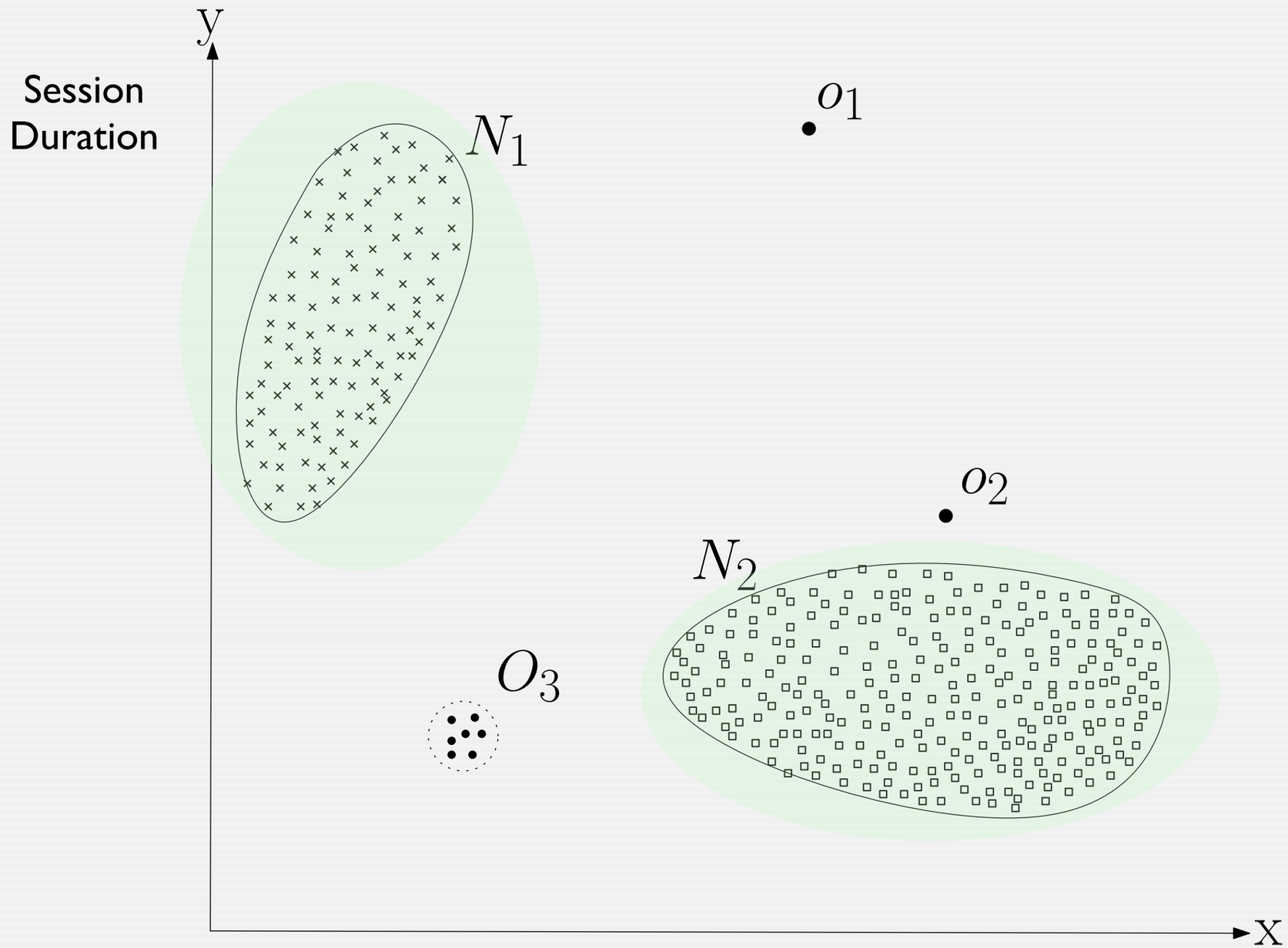
- Host-level system building per-user profiles of activity.
- Login frequency, password failures, session duration, resource consumption.
- Build probability distributions for attribute/user pairs.
- Determine likelihood that new activity is outside of the assumed model.

A Simple 2D Model of Normal



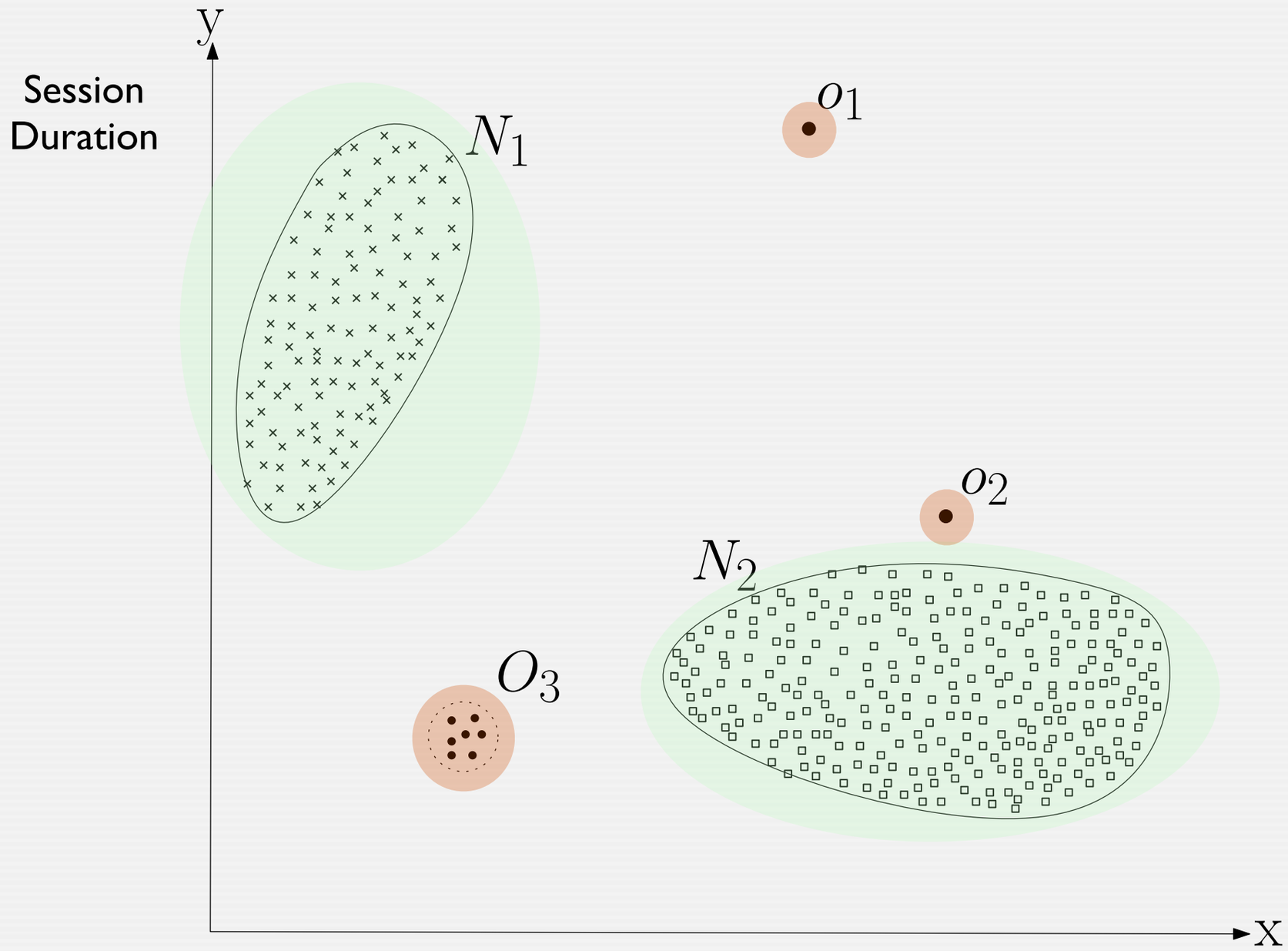
Source: Chandola et al. 2009

A Simple 2D Model of Normal



Source: Chandola et al. 2009

A Simple 2D Model of Normal



Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998]
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling	Section 7.2.2	Chow and Yeung [2002]
Bayesian Networks	Section 4.2	Siaterlis and Maglaris [2004], Sebyala et al. [2002], Valdes and Skinner [2000], Bronstein et al. [2001]
Neural Networks	Section 4.1	HIDE [Zhang et al. 2001], NSOM [Labib and Vemuri 2002], Smith et al. [2002], Hawkins et al. [2002], Kruegel et al. [2003], Manikopoulos and Papavassiliou [2002], Ramadas et al. [2003]
Support Vector Machines	Section 4.3	Eskin et al. [2002]
Rule-based Systems	Section 4.4	ADAM [Barbara et al. 2001a; Barbara et al. 2003; Barbara et al. 2001b], Fan et al. [2001], Helmer et al. [1998], Qin and Hwang [2004], Salvador and Chan [2003], Otey et al. [2003]
Clustering Based	Section 6	ADMIT [Sequeira and Zaki 2002], Eskin et al. [2002], Wu and Zhang [2003], Otey et al. [2003]
Nearest Neighbor based	Section 5	MINDS [Ertoz et al. 2004; Chandola et al. 2006], Eskin et al. [2002]
Spectral	Section 9	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theoretic	Section 8	Lee and Xiang [2001],Noble and Cook [2003]

Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

Examples of Past Efforts

Technique Used	Section	References
Statistical Profiling using Histograms	Section 7.2.1	NIDES [Anderson et al. 1994; Anderson et al. 1995; Javitz and Valdes 1991], EMERALD [Porras and Neumann 1997], Yamanishi et al [2001; 2004], Ho et al. [1999], Kruegel at al [2002; 2003], Mahoney et al [2002; 2003; 2003; 2007], Sargor [1998]
Parametric Statistical Modeling	Section 7.1	Gwadera et al [2005b; 2004], Ye and Chen [2001]
Non-parametric Statistical Modeling	Section 7.2.2	Chow and Yeung [2002]
Bayesian Networks	Section 4.2	Siaterlis and Maglaris [2004], Sebyala et al. [2002], Valdes and Skinner [2000], Bronstein et al. [2001]
Neural Networks	Section 4.1	HIDE [Zhang et al. 2001], NSOM [Labib and Vemuri 2002], Smith et al. [2002], Hawkins et al. [2002], Kruegel et al. [2003], Manikopoulos and Papavassiliou [2002], Ramadas et al. [2003]
Support Vector Machines	Section 4.3	Eskin et al. [2002]
Rule-based Systems	Section 4.4	ADAM [Barbara et al. 2001a; Barbara et al. 2003; Barbara et al. 2001b], Fan et al. [2001], Helmer et al. [1998], Qin and Hwang [2004], Salvador and Chan [2003], Otey et al. [2003]
Clustering Based	Section 6	ADMIT [Sequeira and Zaki 2002], Eskin et al. [2002], Wu and Zhang [2003], Otey et al. [2003]
Nearest Neighbor based	Section 5	MINDS [Ertoz et al. 2004; Chandola et al. 2006], Eskin et al. [2002]
Spectral	Section 9	Shyu et al. [2003], Lakhina et al. [2005], Thottan and Ji [2003],Sun et al. [2007]
Information Theoretic	Section 8	Lee and Xiang [2001],Noble and Cook [2003]

Features used

- packet sizes
- IP addresses
- ports
- header fields
- timestamps
- inter-arrival times
- session size
- session duration
- session volume
- payload frequencies
- payload tokens
- payload pattern
- ...

Examples of techniques used for network intrusion detection.

Source: Chandola et al. 2009

The Holy Grail ...

The Holy Grail ...

- Anomaly detection is extremely appealing.
  - We find novel attacks without anticipating any specifics ("zero-day").
  - It's plausible: machine-learning works so well in many other domains.

The Holy Grail ...

- Anomaly detection is extremely appealing.
  - We find novel attacks without anticipating any specifics ("zero-day").
  - It's plausible: machine-learning works so well in many other domains.
- Many research efforts have explored the notion.
  - Numerous papers have been written ...

The Holy Grail ...

- Anomaly detection is extremely appealing.
  - We find novel attacks without anticipating any specifics ("zero-day").
  - It's plausible: machine-learning works so well in many other domains.
- Many research efforts have explored the notion.
  - Numerous papers have been written ...
- But guess what's used *in operation*? Snort.
  - We find hardly any machine-learning-based NIDS in real-world deployments.

The Holy Grail ...

- Anomaly detection is extremely appealing.
  - We find novel attacks without anticipating any specifics ("zero-day").
  - It's plausible: machine-learning works so well in many other domains.
- Many research efforts have explored the notion.
  - Numerous papers have been written ...
- But guess what's used *in operation*? Snort.
  - We find hardly any machine-learning-based NIDS in real-world deployments.
- Could anomaly detection be harder than it appears?

Prerequisites

- My definition of "anomaly detection" is intrusion detection based on a machine-learning algorithm.
  - Technically, the terminology is more fuzzy but that's what people associate.
- I'll focus on network-based approaches.
  - But much of the discussion applies to host-based systems as well.
- I won't tell you how machine-learning works.
- Intrusion detection is all about the real-world.
  - Nothing is perfect; all these systems are based on a set of heuristics.
  - Whatever helps the operator is good.
- Target are medium to large environments
  - 10,000s of users and hosts

Why Is Anomaly Detection Hard?

Intrusion Detection Is Different

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Intrusion Detection Is Different

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Outlier detection and the high costs of errors

How do we find the opposite of normal?

Interpretation of results

What does that anomaly *mean*?

Evaluation

How do we make sure it actually works?

Training data

What do we train our system with?

Evasion risk

Can the attacker mislead our system?

Outlier Detection

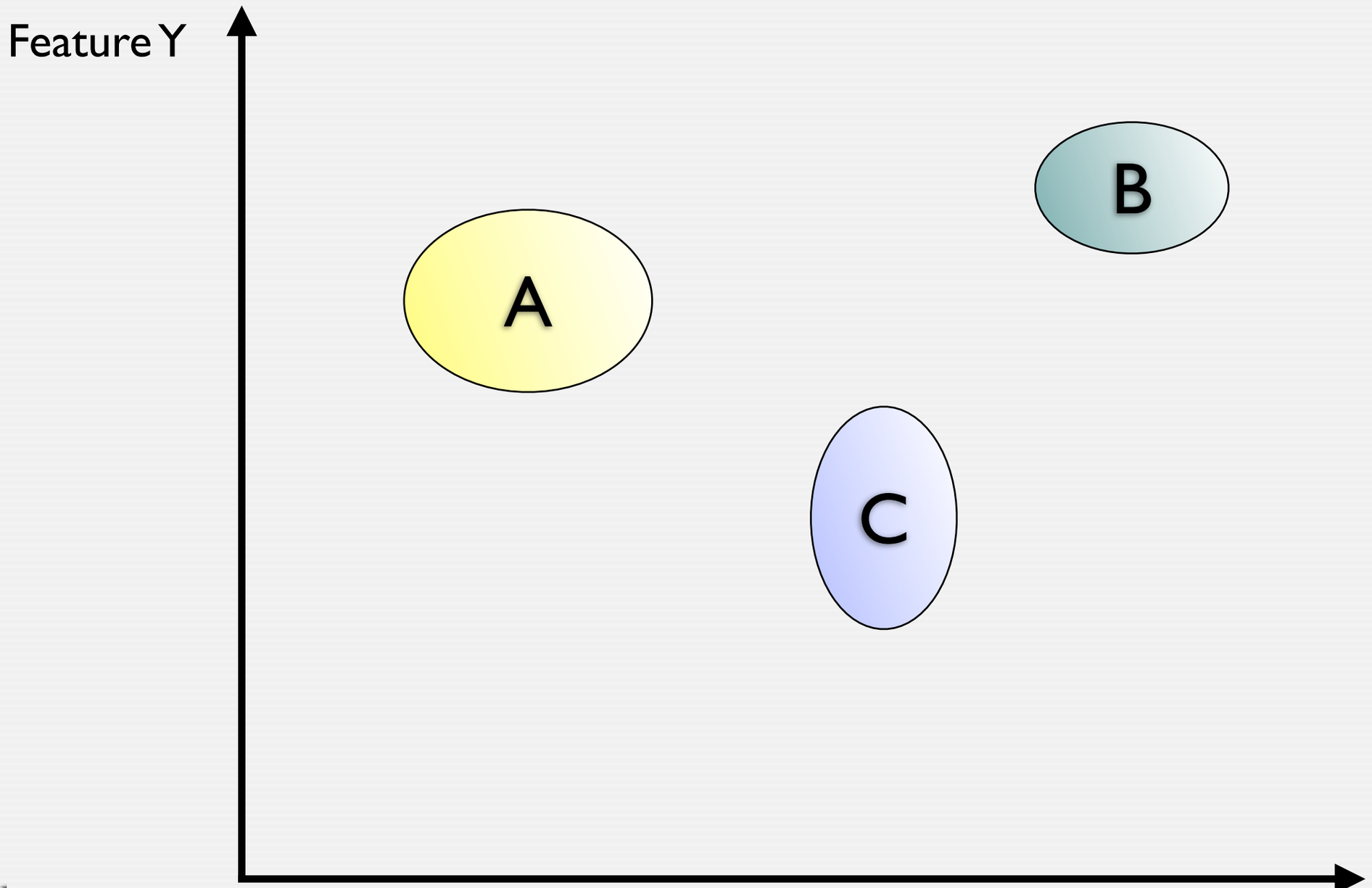
- Anomaly detection is *outlier detection*.
  - Machine-learning builds a model of its normal training data.
  - Given an observation, decide whether it fits the model.

Outlier Detection

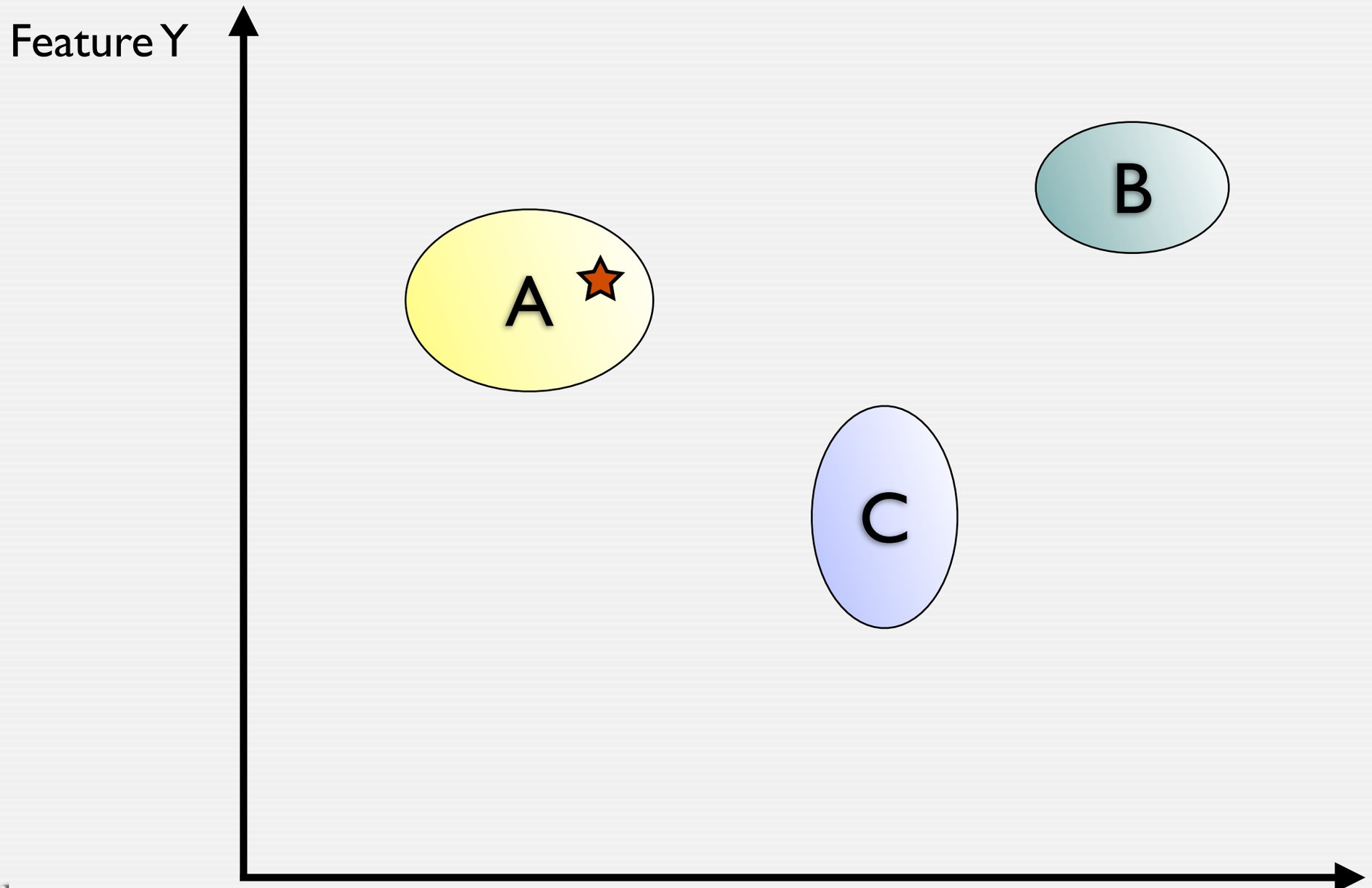
- Anomaly detection is *outlier detection*.
  - Machine-learning builds a model of its normal training data.
  - Given an observation, decide whether it fits the model.

- Problem: Machine-learning is not that good at this.
  - It's better at finding similarity than abnormality.
  - The classical machine-learning application is *classification*.

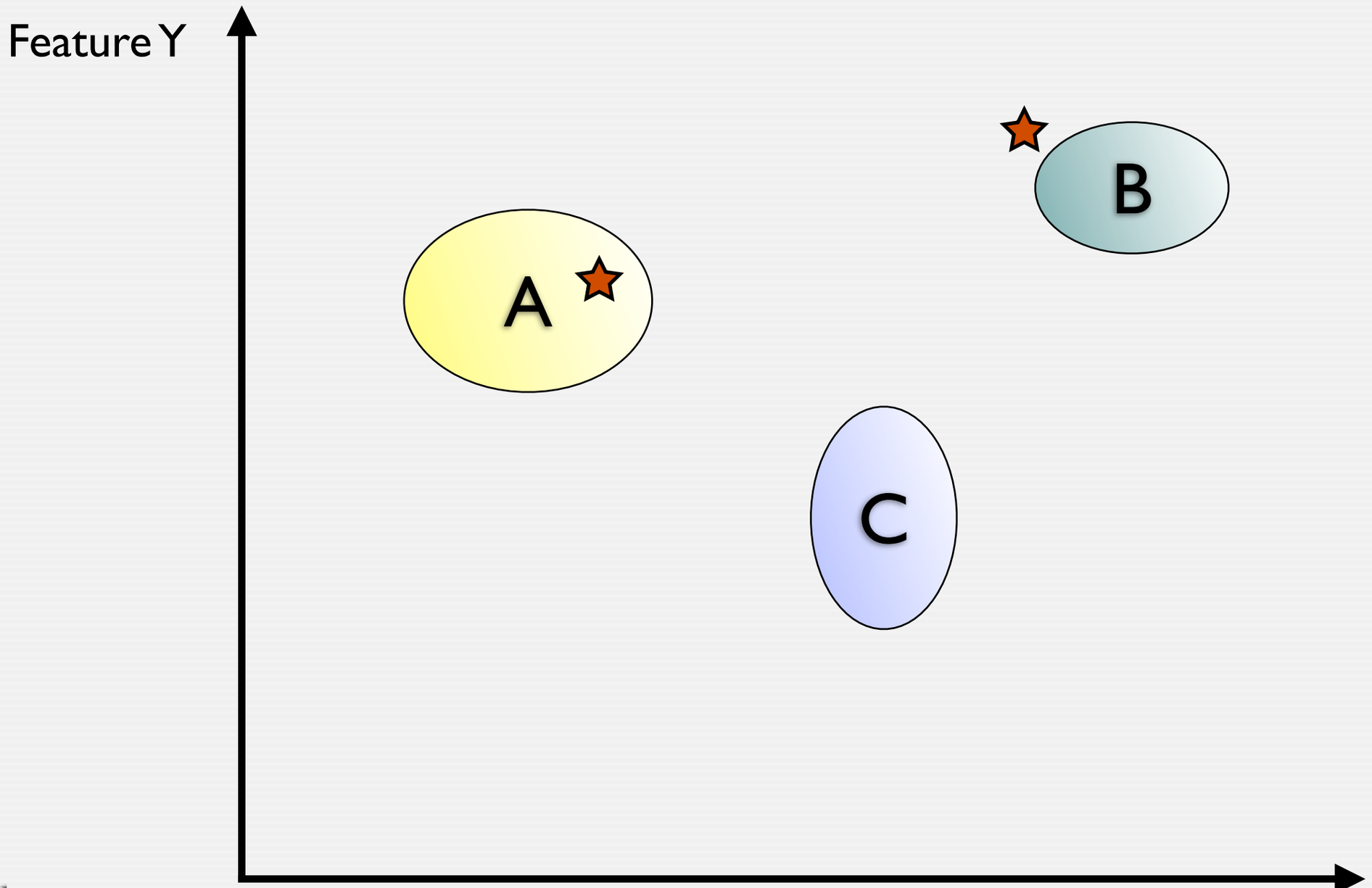
Classification Problem



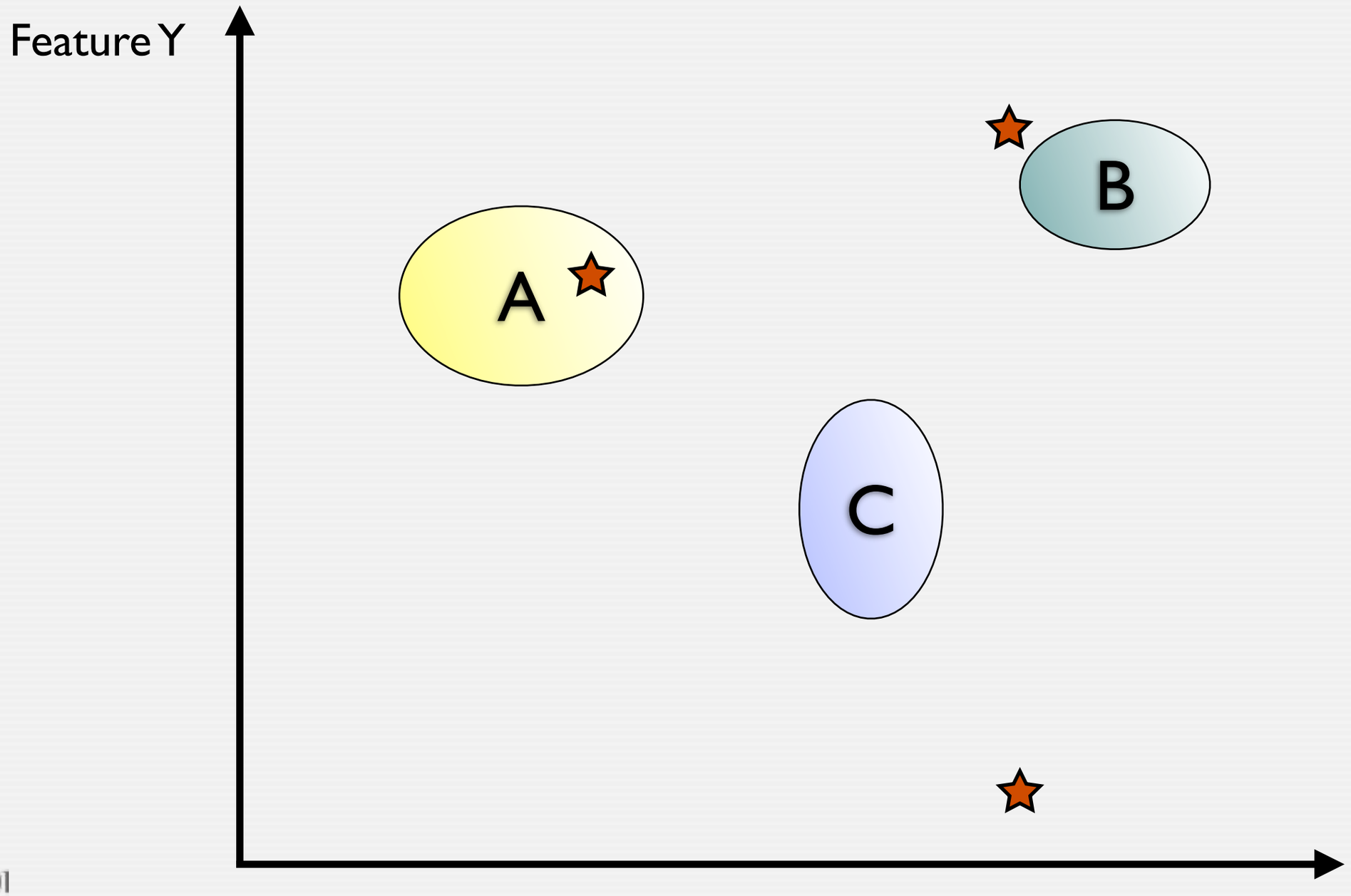
Classification Problem



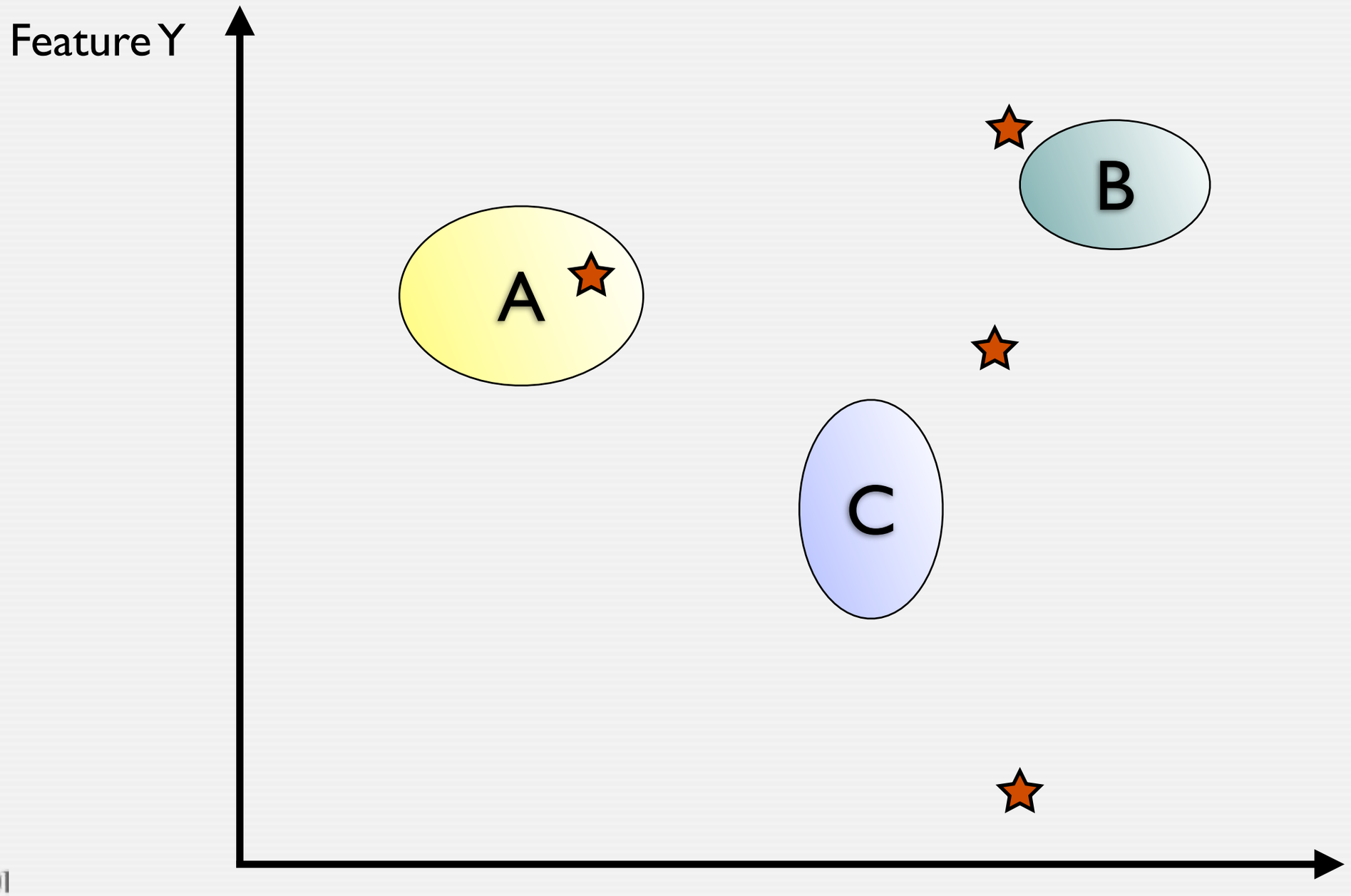
Classification Problem



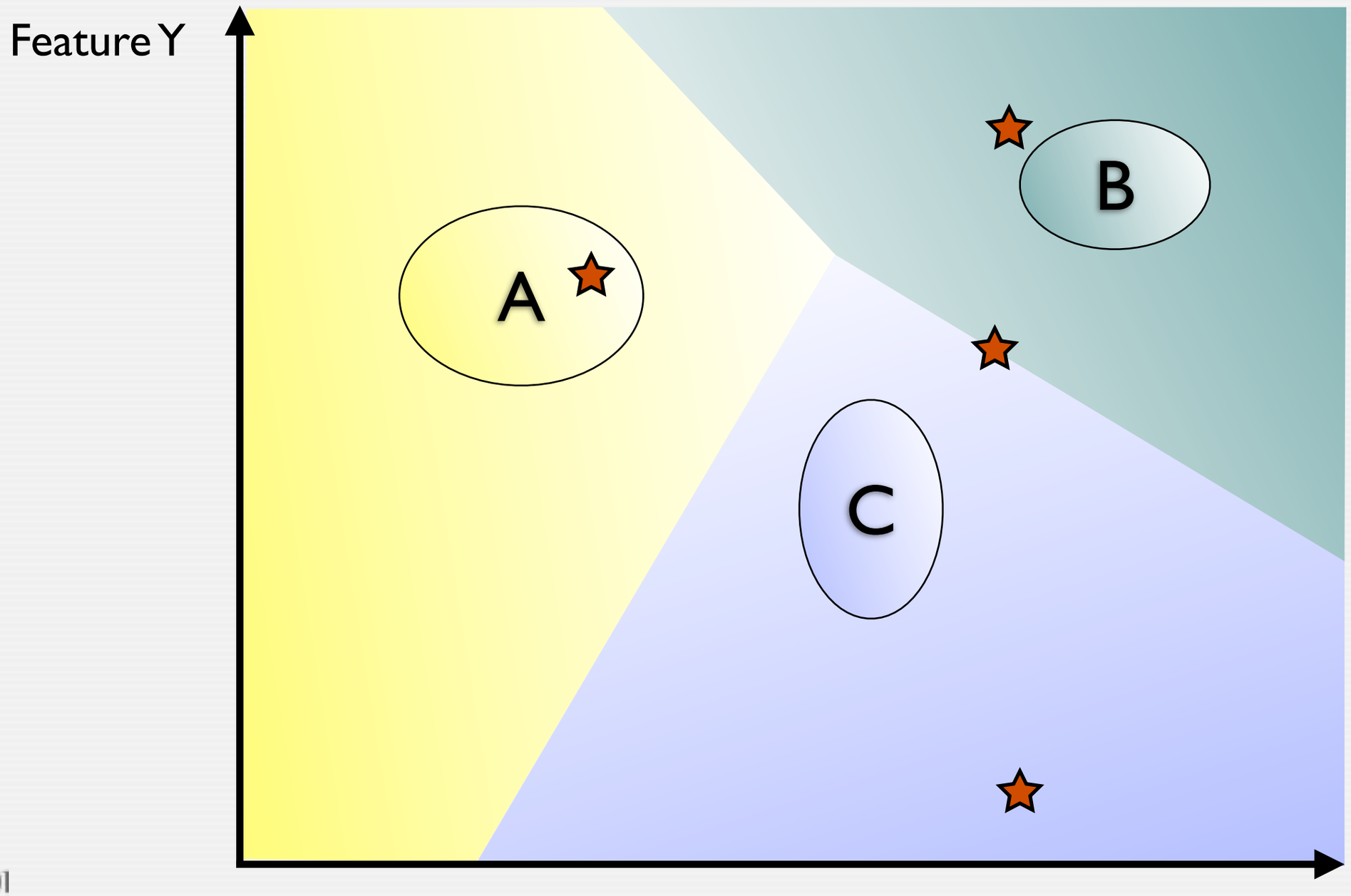
Classification Problem



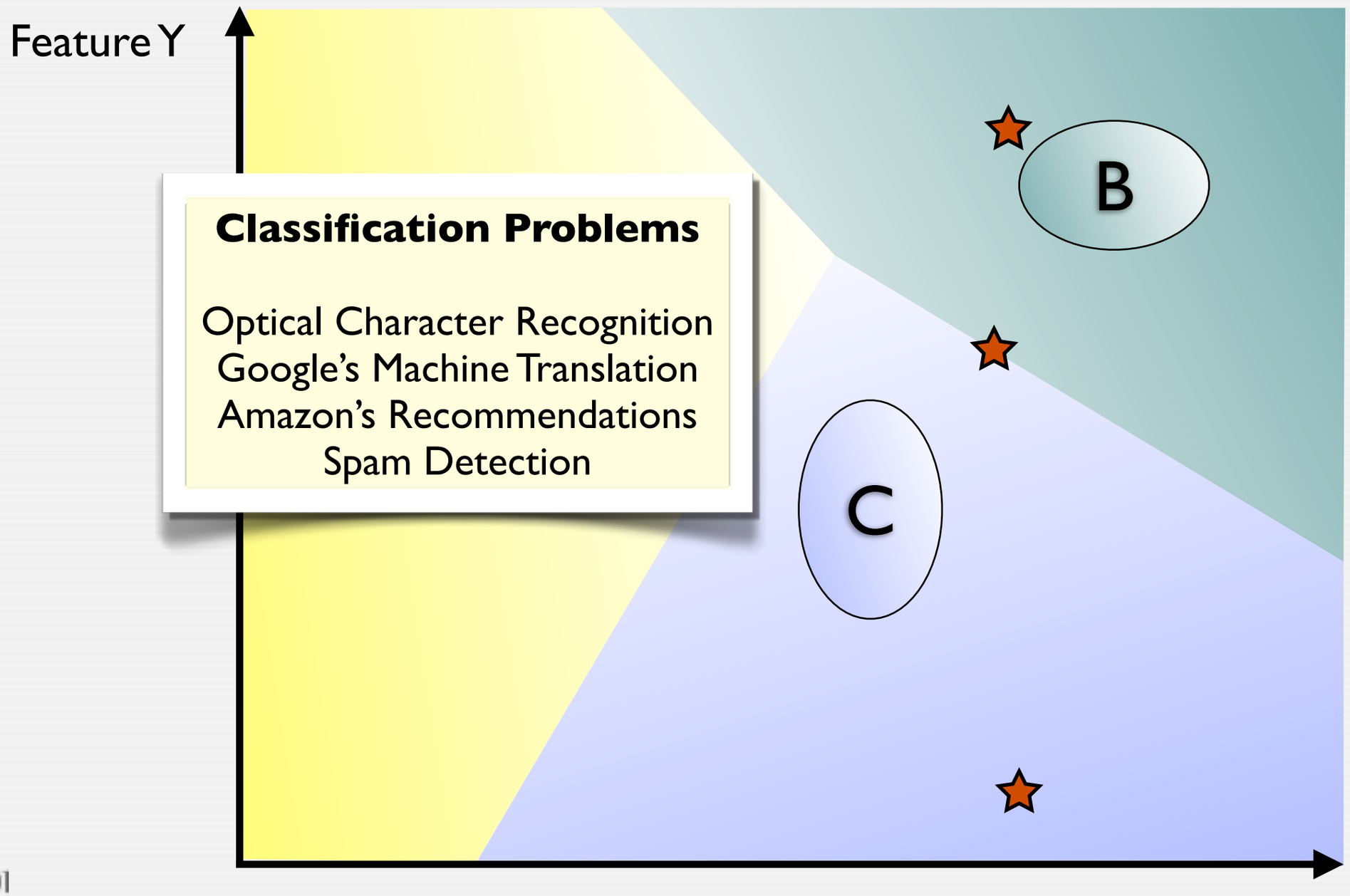
Classification Problem



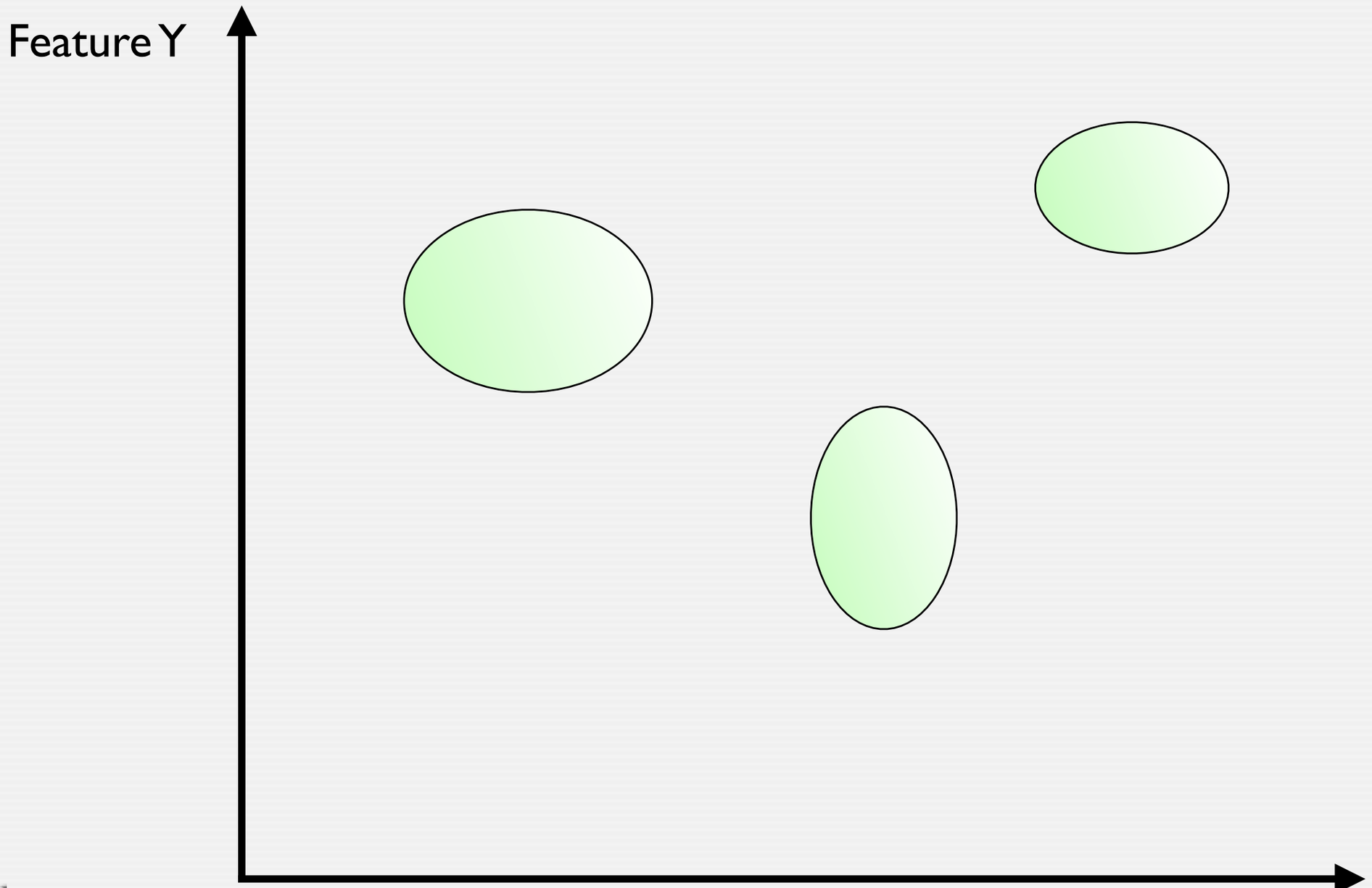
Classification Problem



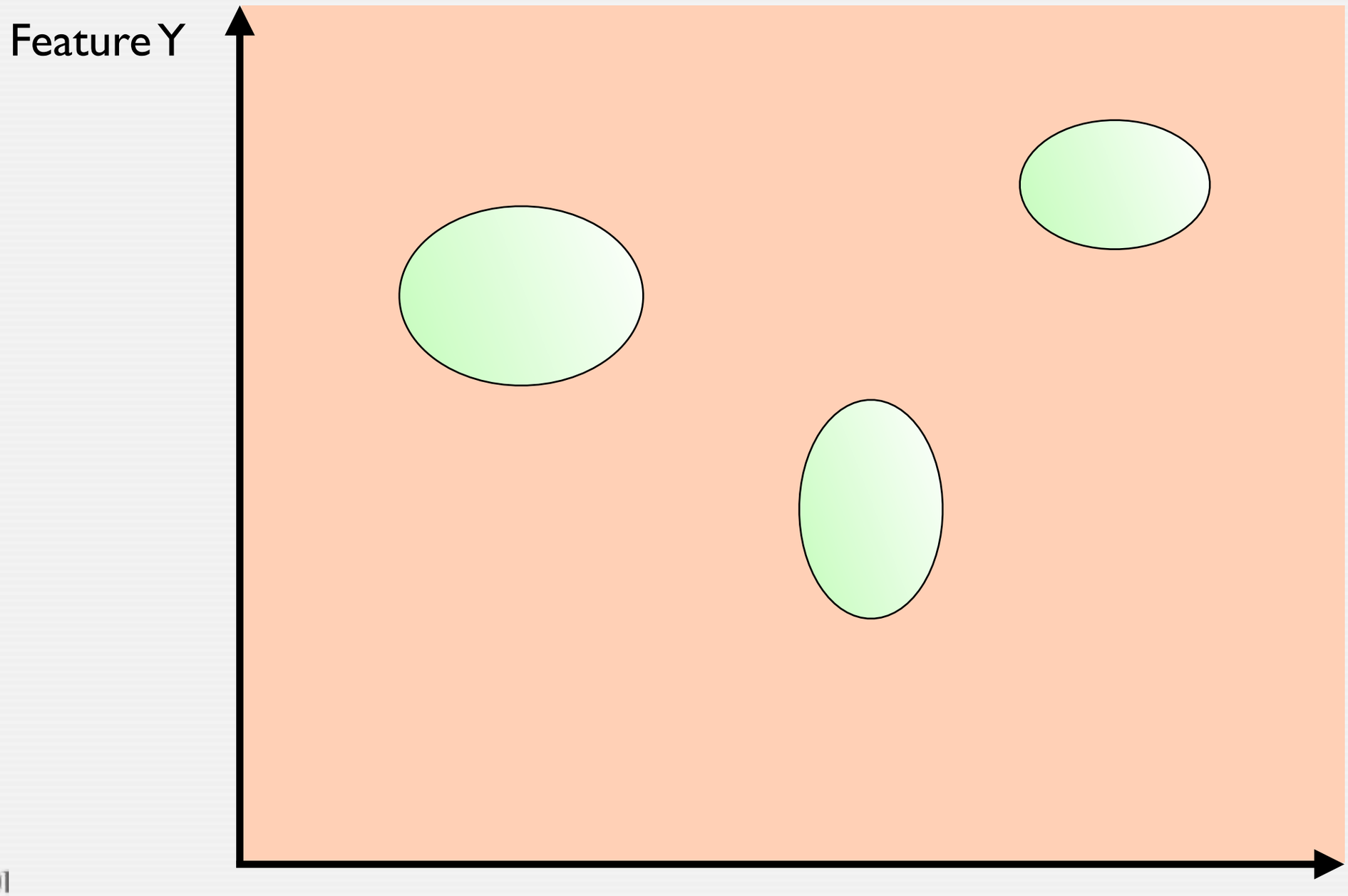
Classification Problem



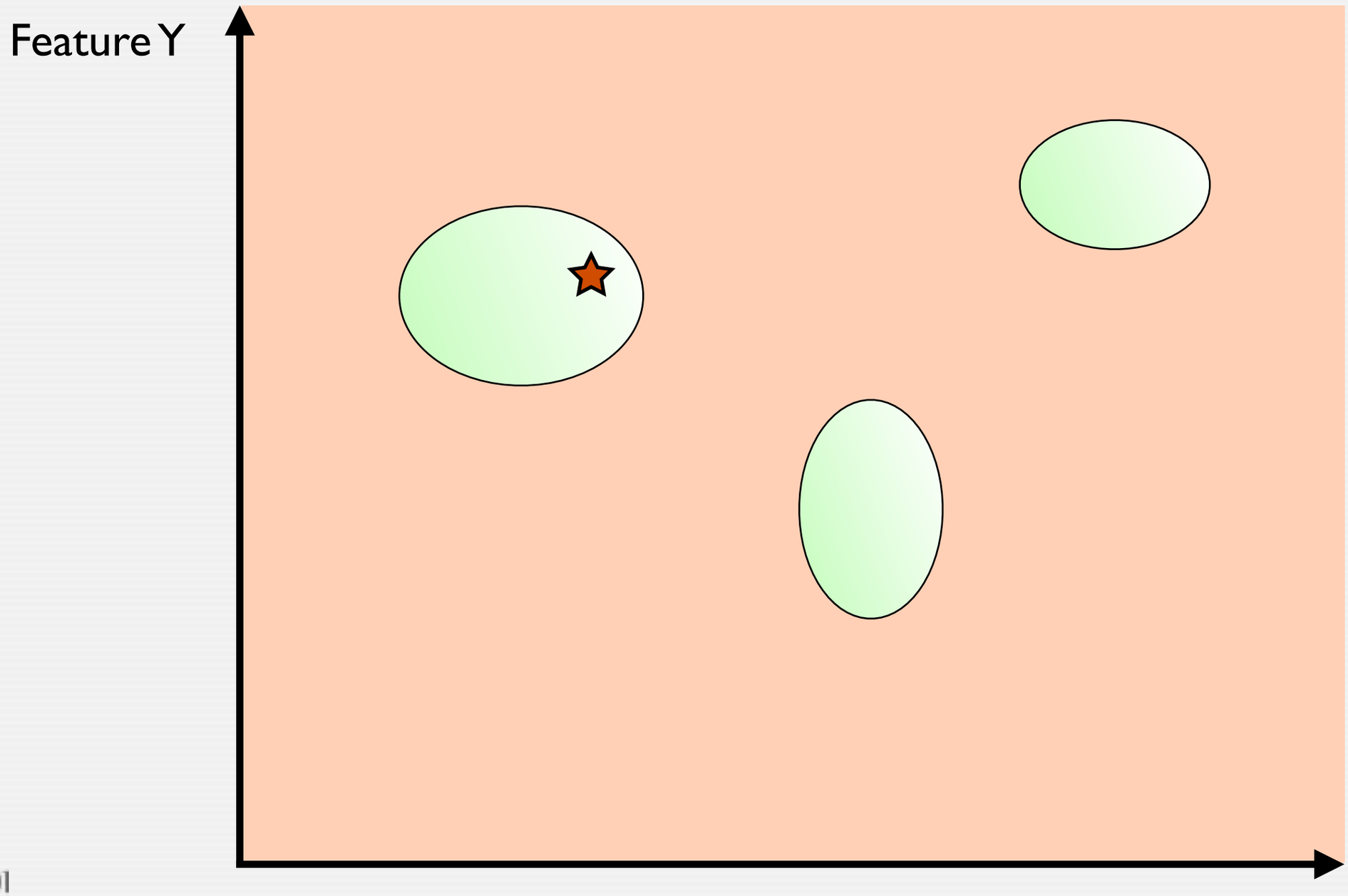
Outlier Detection



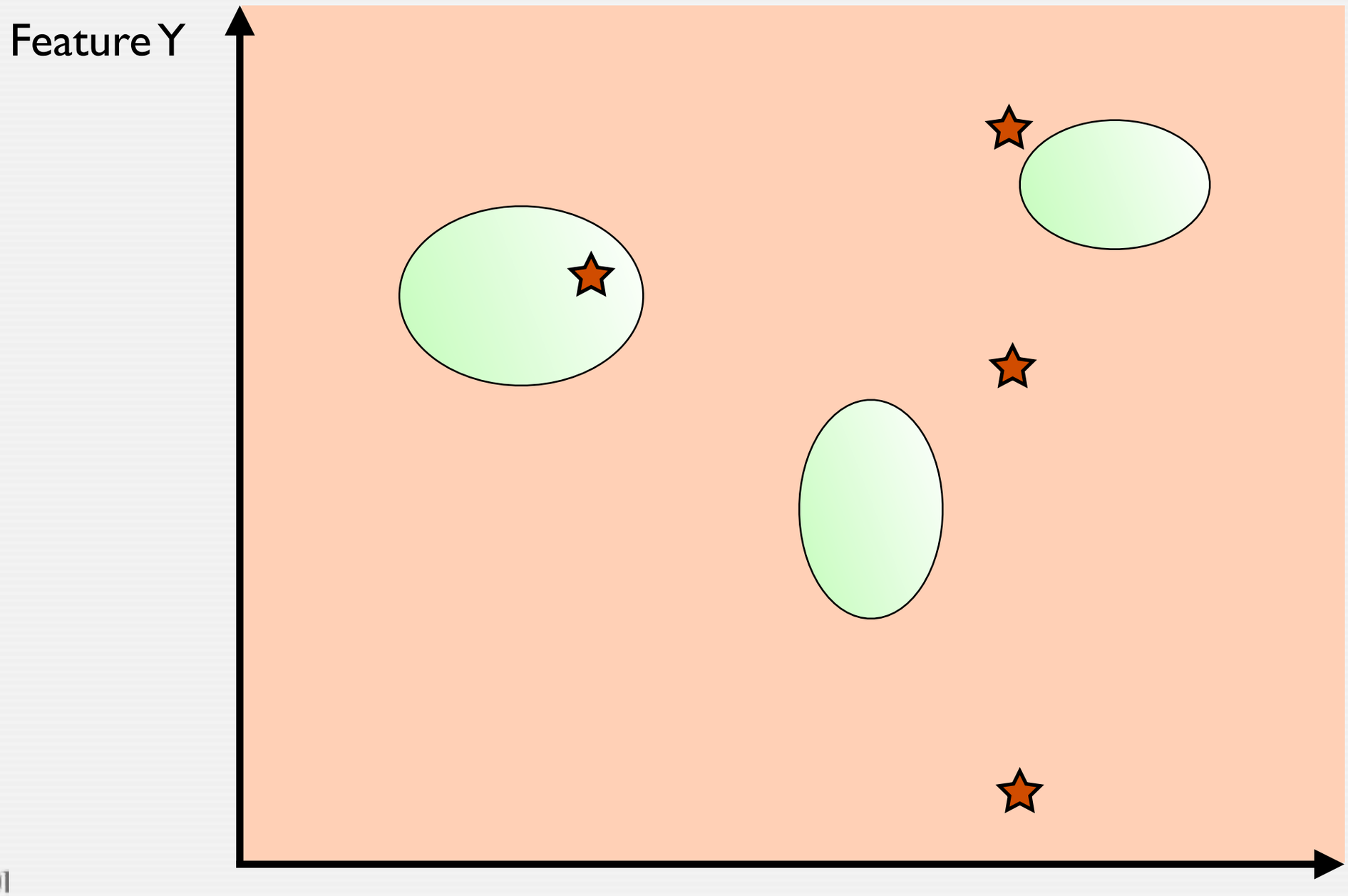
Outlier Detection



Outlier Detection



Outlier Detection



Outlier Detection

- Assumes a *Closed World:*
  - Specify only positive examples.
  - Adopt standing assumption that the rest is negative.

Outlier Detection

- Assumes a *Closed World:*
  - Specify only positive examples.
  - Adopt standing assumption that the rest is negative.
- Real-life problems rarely involve "closed" worlds.
  - One needs to cover all positive cases to avoid misclassifications.

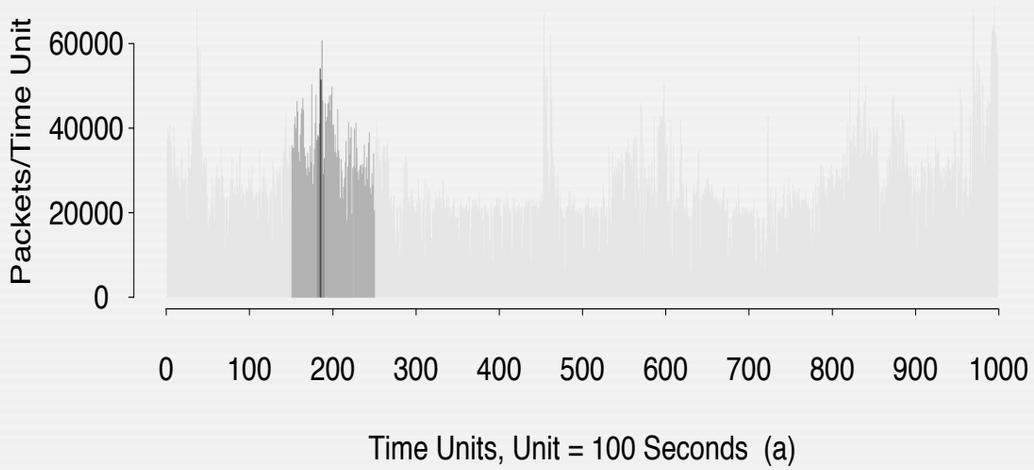
Outlier Detection

- Assumes a *Closed World:*
  - Specify only positive examples.
  - Adopt standing assumption that the rest is negative.
- Real-life problems rarely involve "closed" worlds.
  - One needs to cover all positive cases to avoid misclassifications.
- Can be used successfully if the model is "good enough".
  - Feature space is of low dimensionality and/or variability.
  - Mistakes are cheap.
  - Examples: fraud detection (credit cards, insurances); image analysis.

Outlier Detection

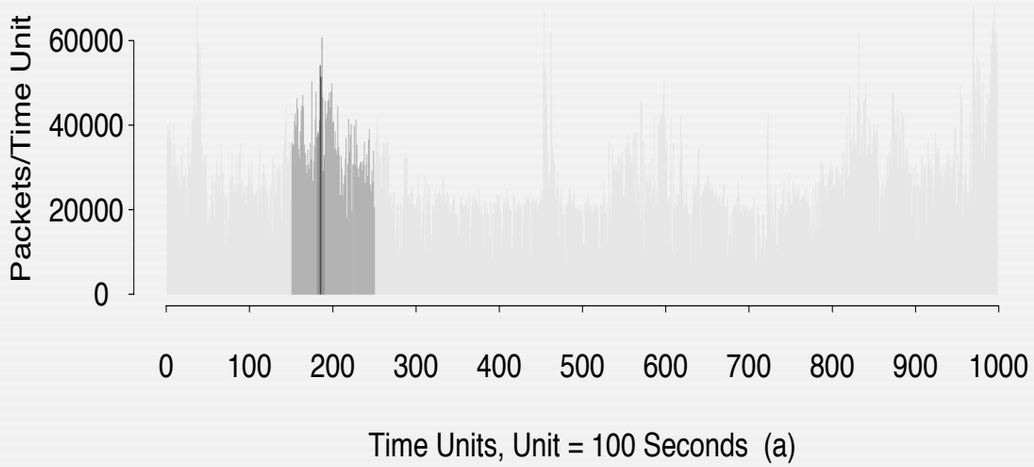
- Assumes a *Closed World:*
  - Specify only positive examples.
  - Adopt standing assumption that the rest is negative.
- Real-life problems rarely involve "closed" worlds.
  - One needs to cover all positive cases to avoid misclassifications.
- Can be used successfully if the model is "good enough".
  - Feature space is of low dimensionality and/or variability.
  - Mistakes are cheap.
  - Examples: fraud detection (credit cards, insurances); image analysis.
- Tends to be hard to do for intrusion detection
  - Network activity is extremely diverse at all levels of the protocol stack.
  - ... and that's already without any malicious activity.

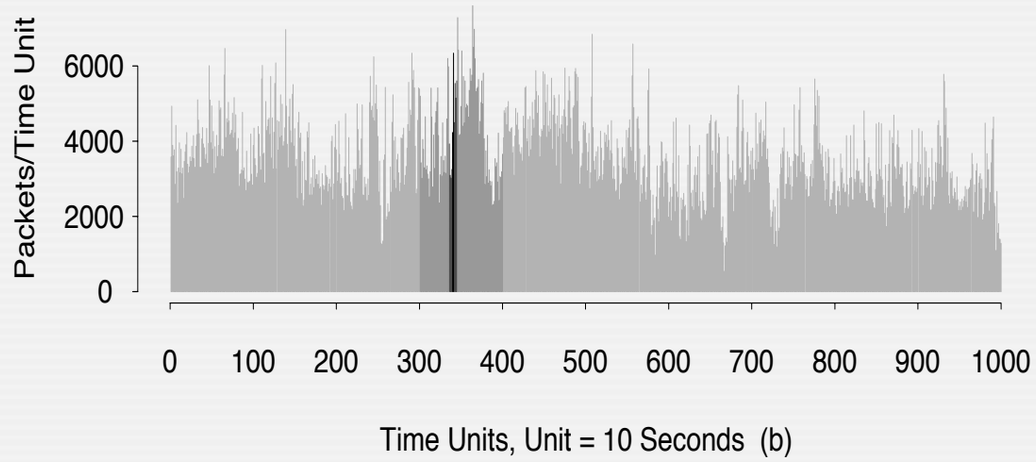
Self-Similarity of Ethernet Traffic



Source: LeLand et al. 1995

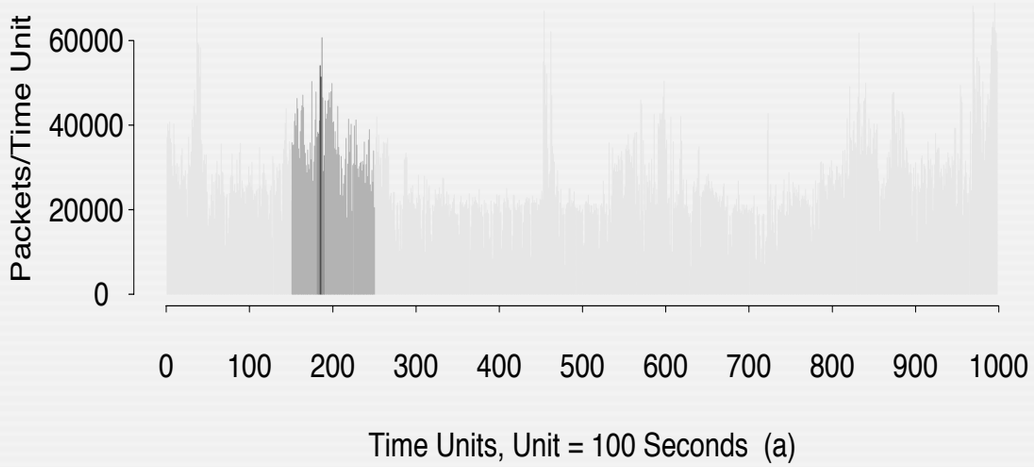
Self-Similarity of Ethernet Traffic

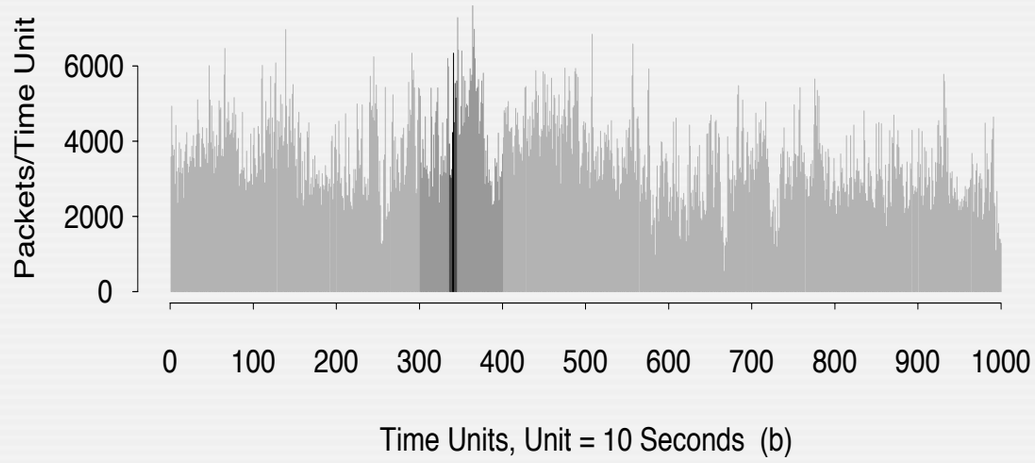


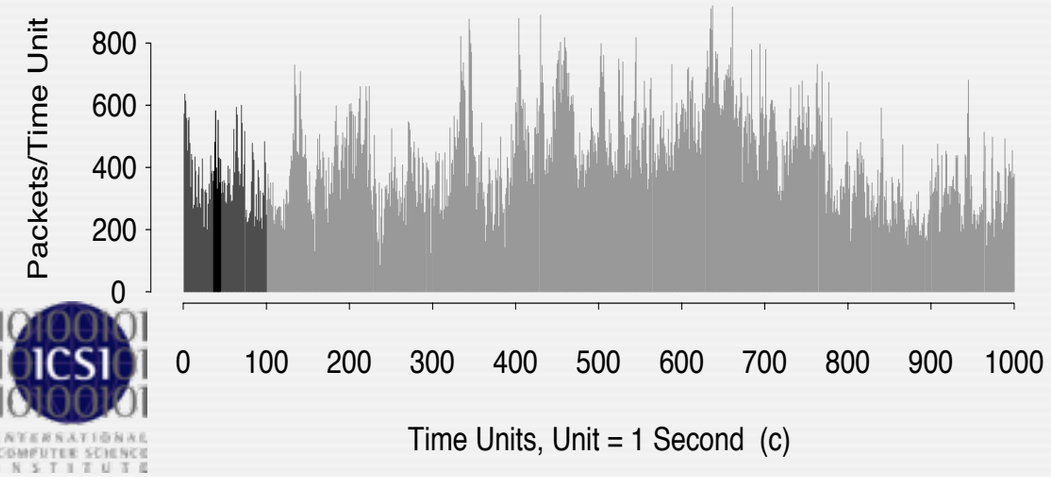


Source: LeLand et al. 1995

Self-Similarity of Ethernet Traffic

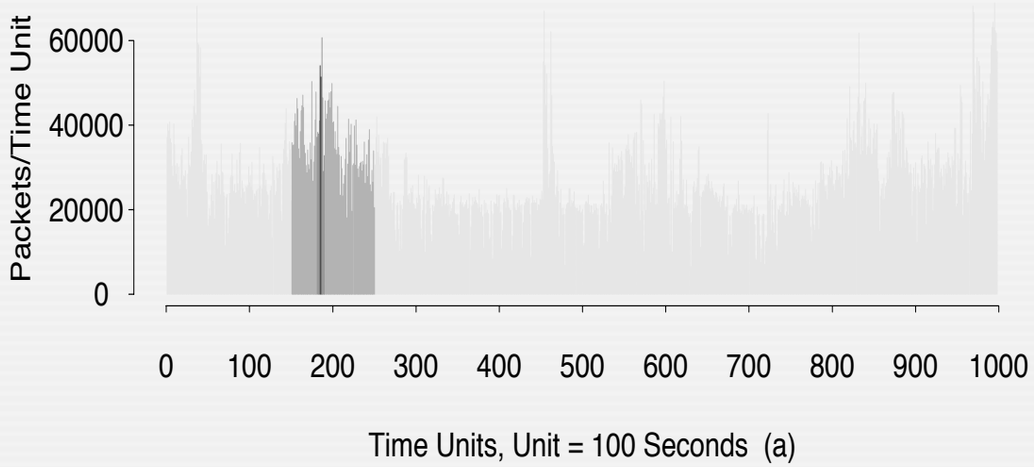


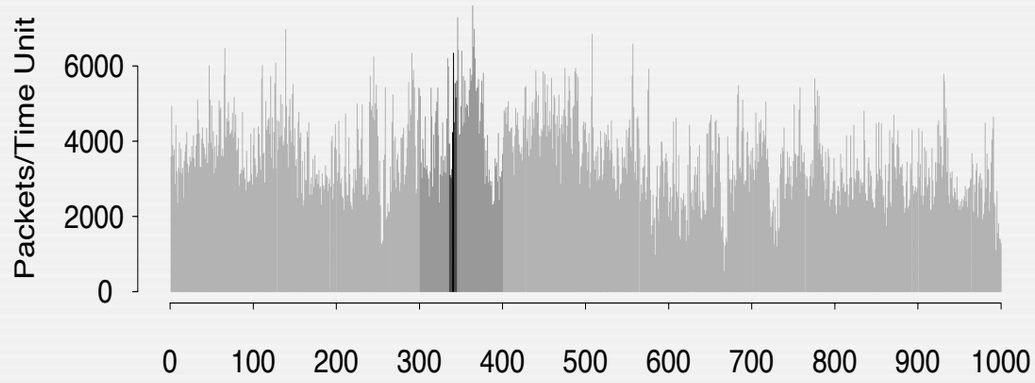


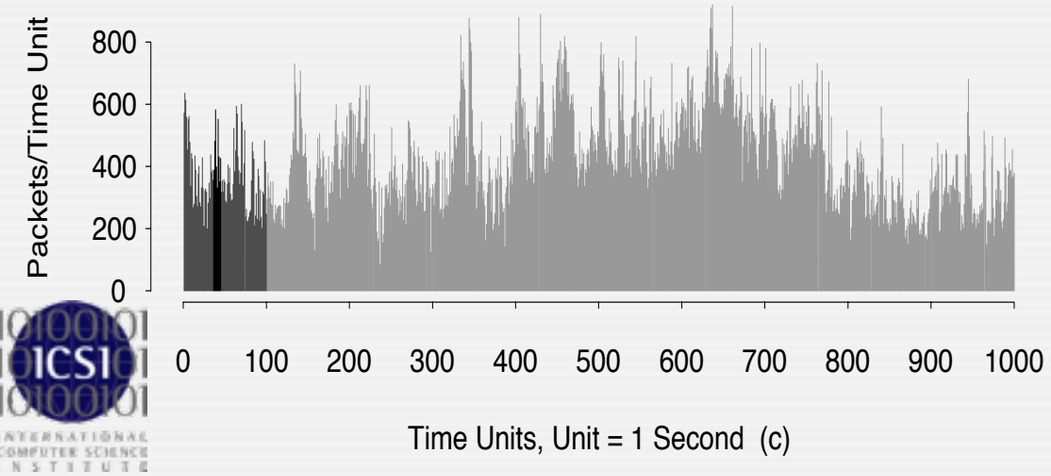


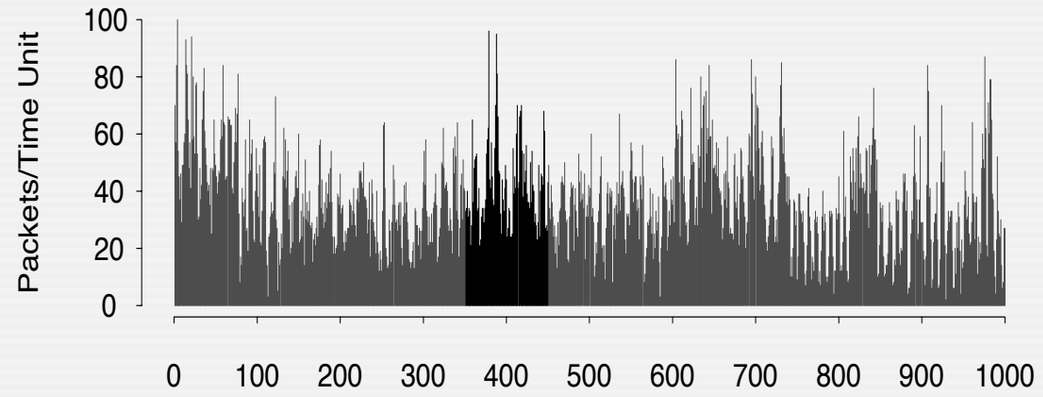
Source: LeLand et al. 1995

Self-Similarity of Ethernet Traffic





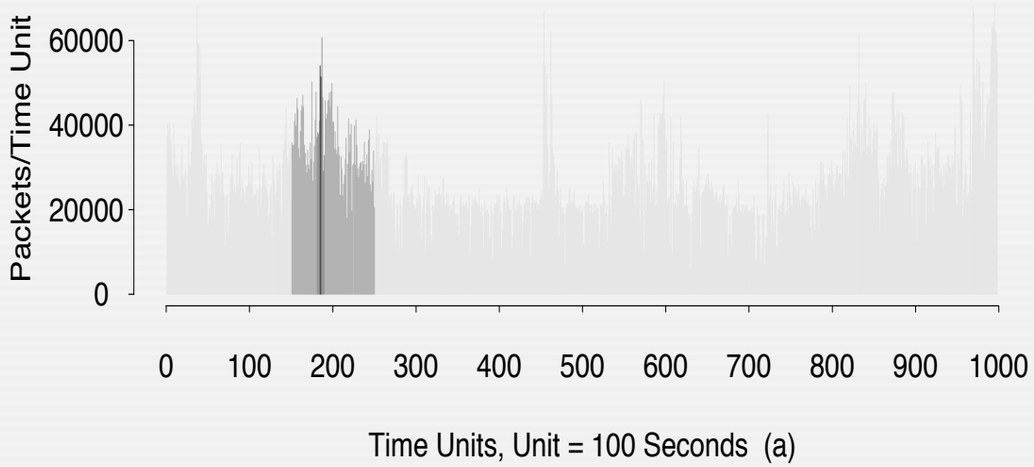


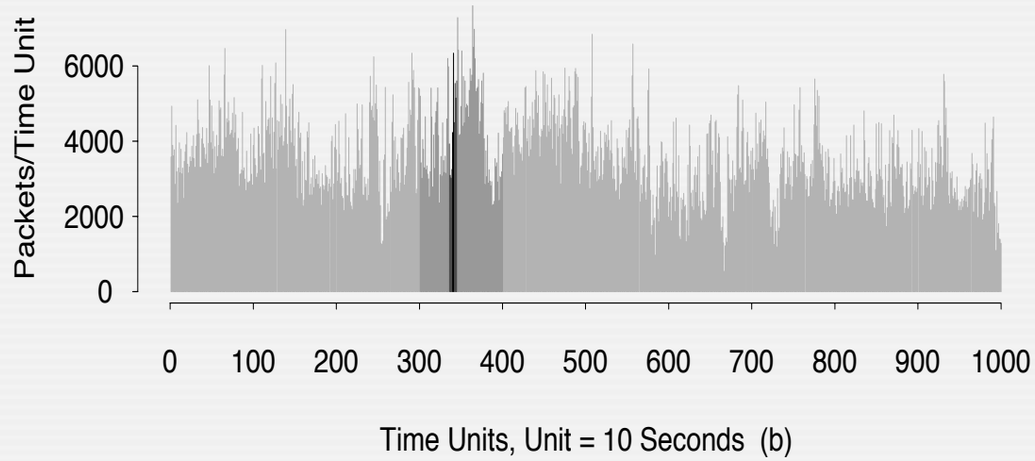


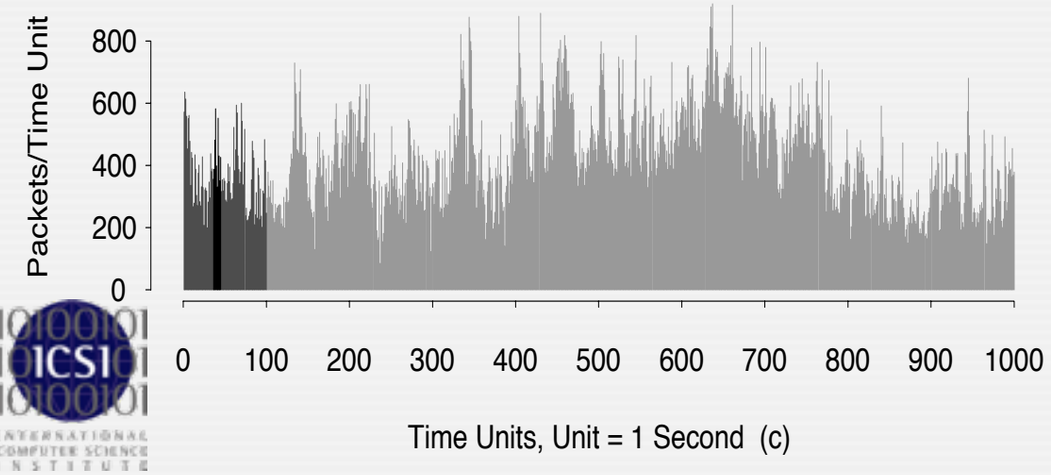
Time Units, Unit = 0.1 Second (d)

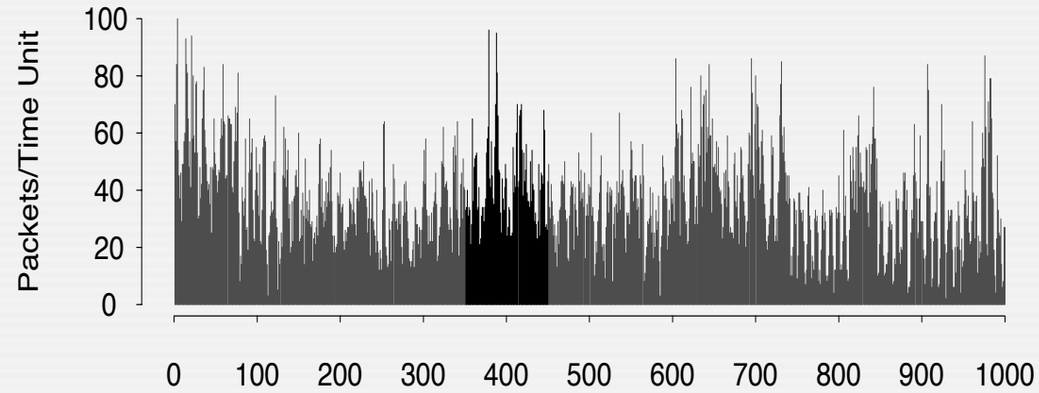
Source: LeLand et al. 1995

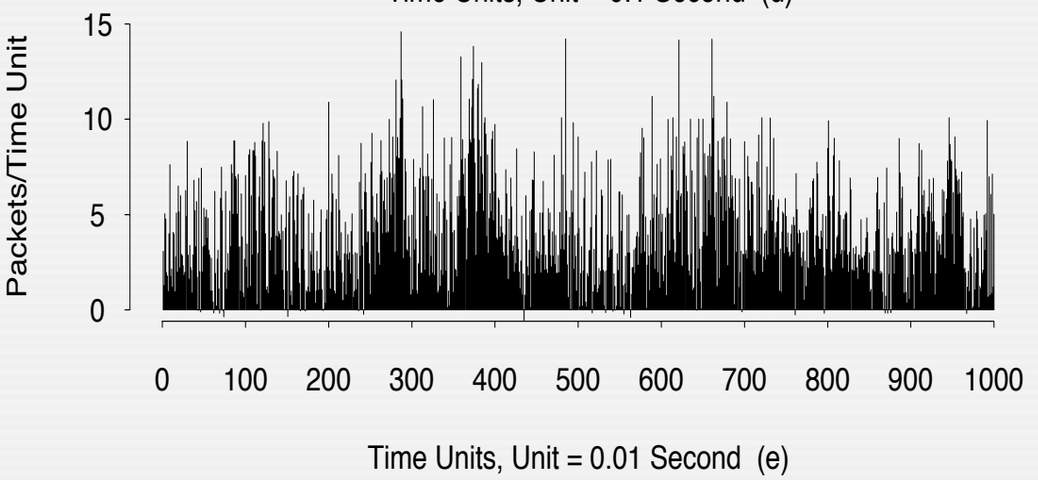
Self-Similarity of Ethernet Traffic





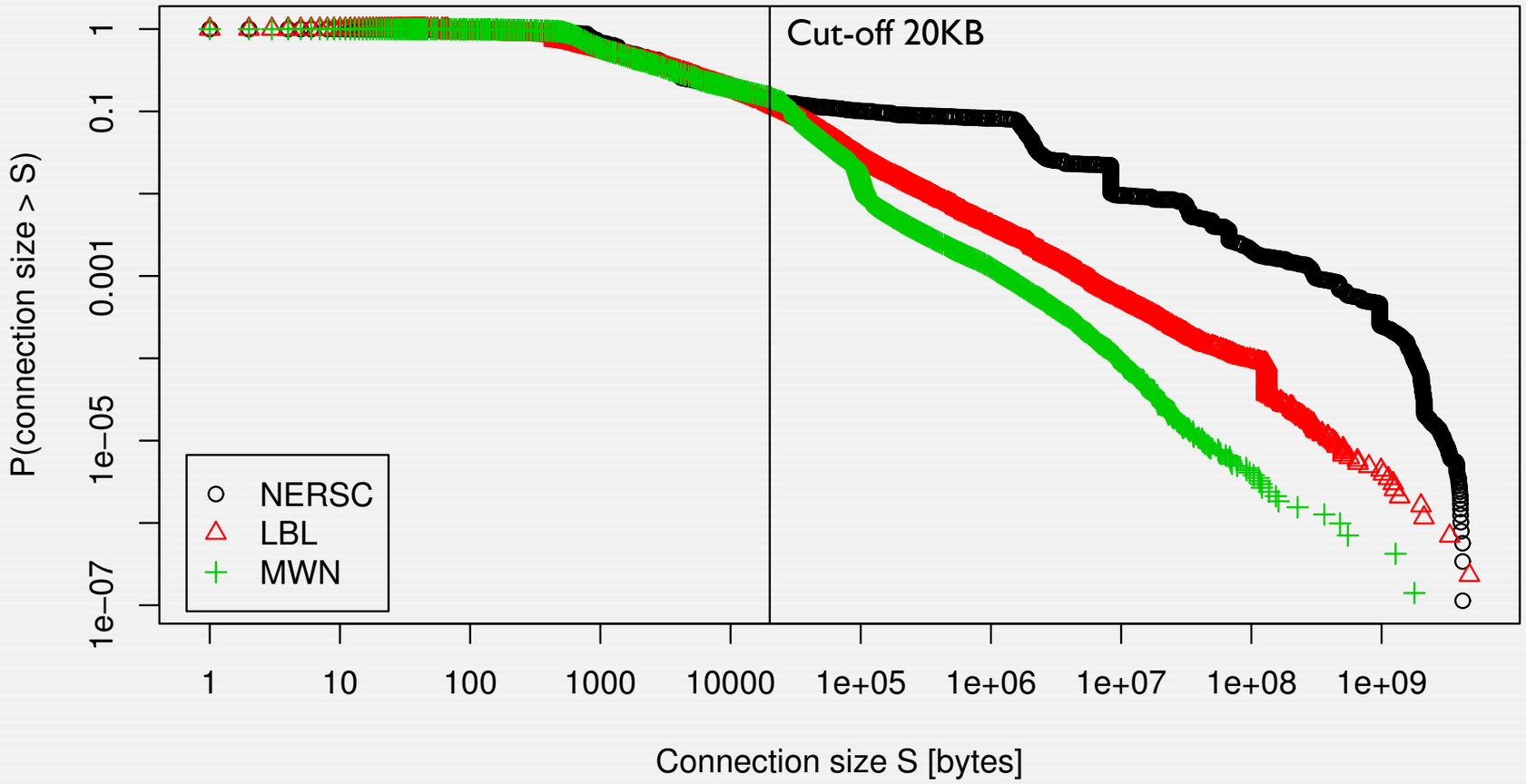






Source: LeLand et al. 1995

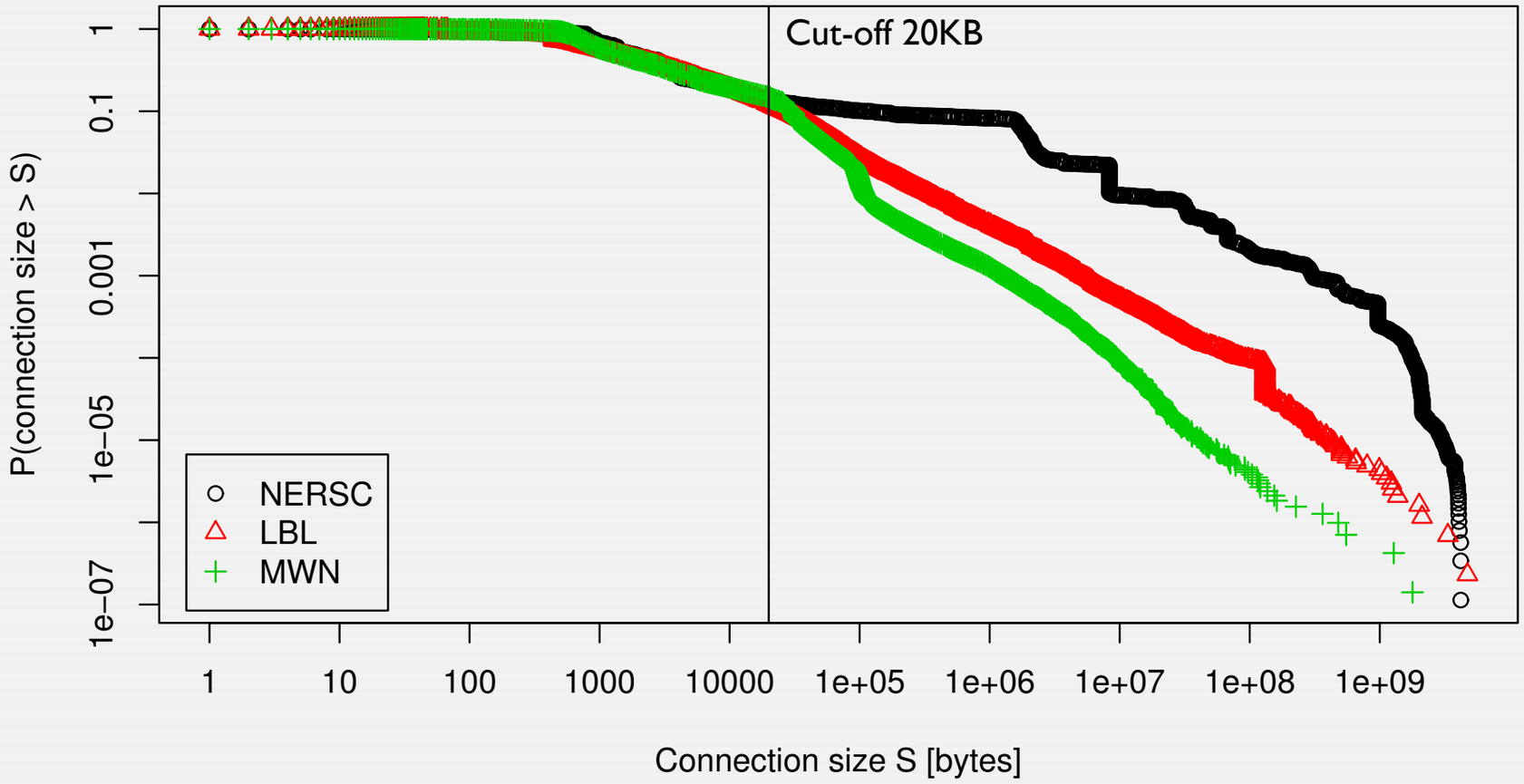
Internet Traffic: Heavy Tails



Site	Conns > 20KB	%Bytes
MWN	15%	87%
LBL	12%	96%
NERSC	14%	99.86%

Source: Kornexl 2005

Internet Traffic: Heavy Tails

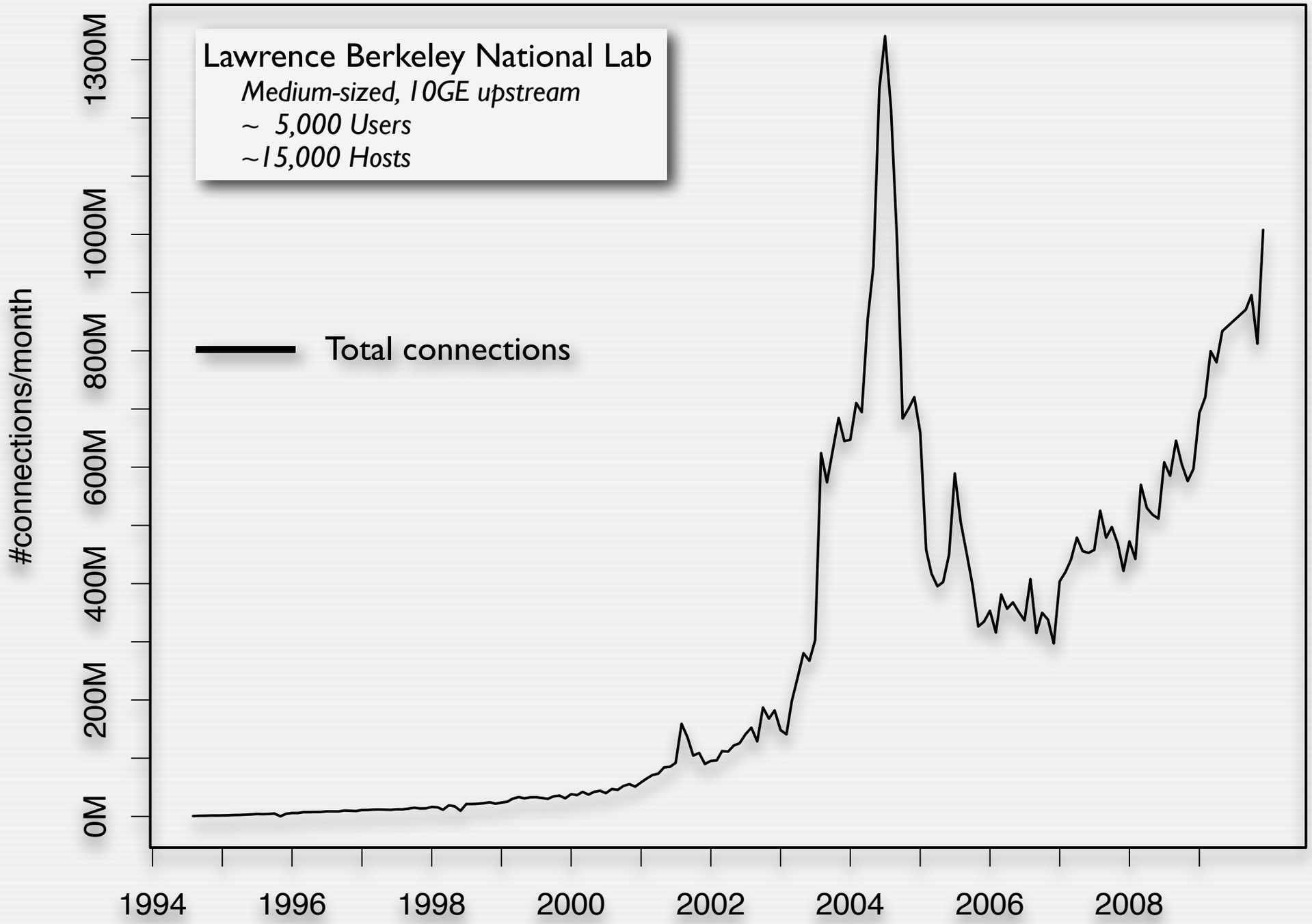


Site	Conns > 20KB	%Bytes
MWN	15%	87%
LBL	12%	96%
NERSC	14%	99.86%

Self-similarity/heavy-tails lead to extreme bursts.

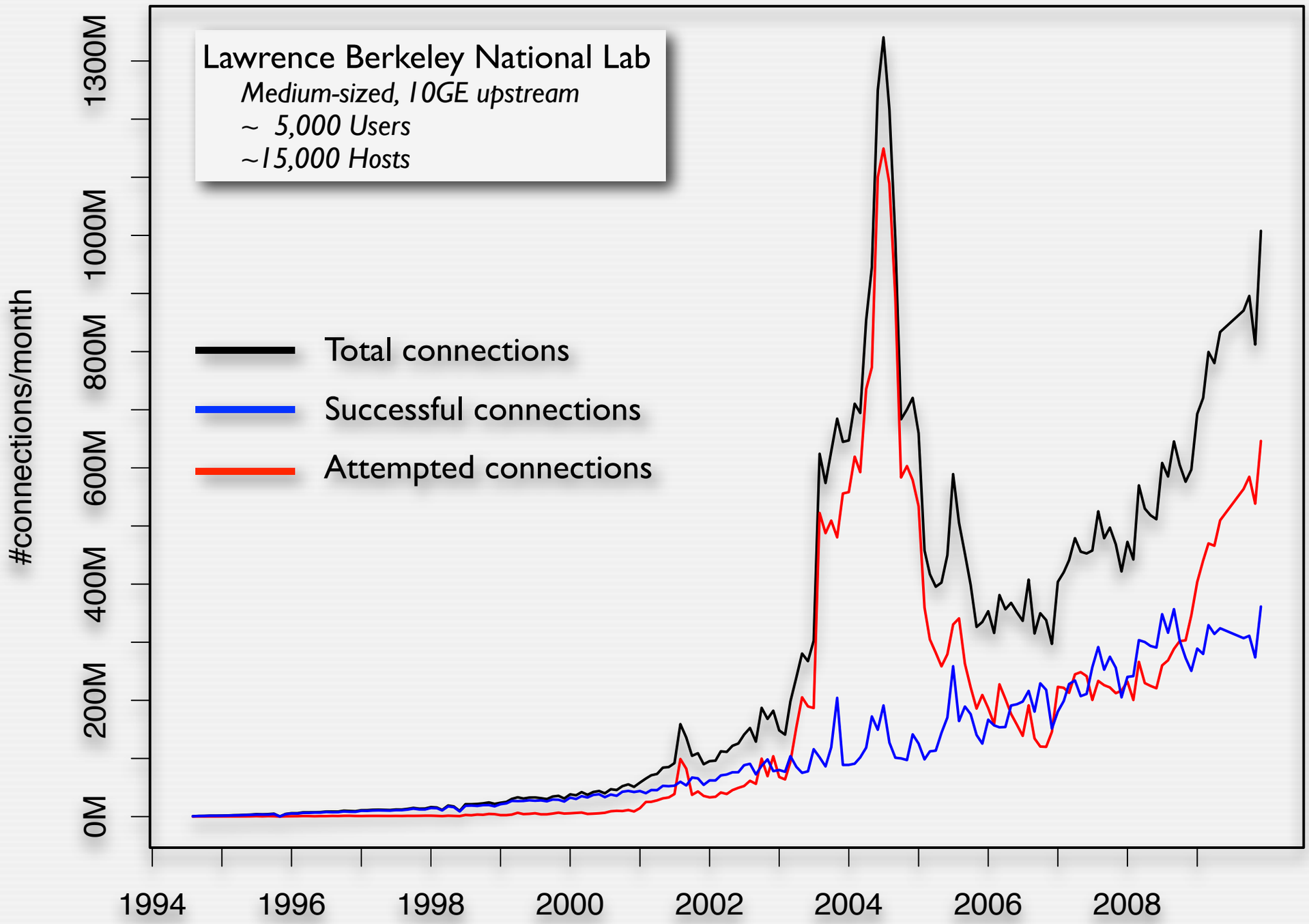
Source: Kornexl 2005

A Moving Target



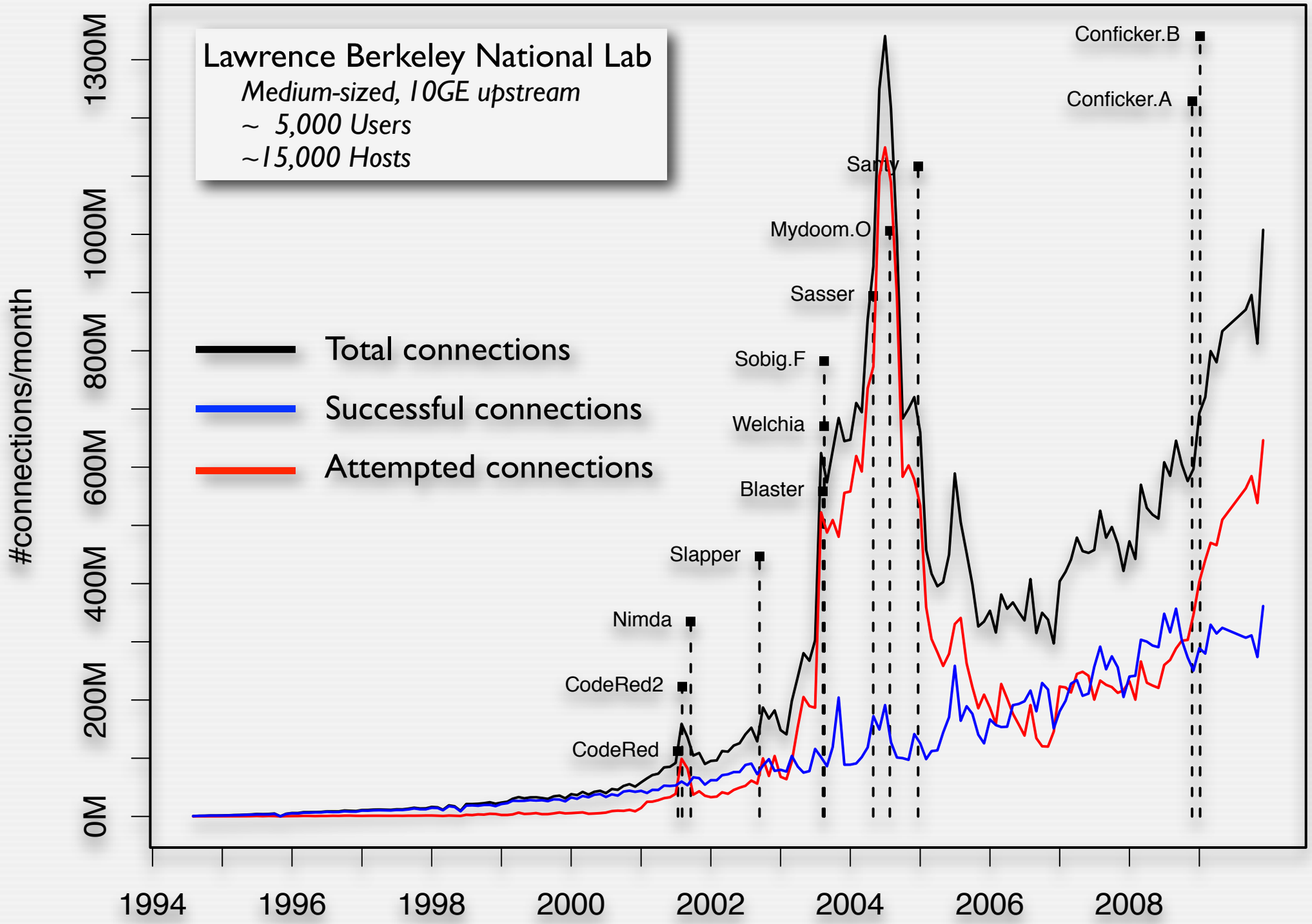
Data: Lawrence Berkeley National Lab

A Moving Target



Data: Lawrence Berkeley National Lab

A Moving Target



Data: Lawrence Berkeley National Lab

One Day of Crud at ICSI

Postel's Law: *Be strict in what you send and liberal in what you accept ...*

One Day of Crud at ICSI

Postel's Law: *Be strict in what you send and liberal in what you accept ...*

active-connection-reuse	DNS-label-len-gt-pkt	HTTP-chunked-multipart	possible-split-routing
bad-Ident-reply	DNS-label-too-long	HTTP-version-mismatch	SYN-after-close
bad-RPC	DNS-RR-length-mismatch	illegal-%-at-end-of-URI	SYN-after-reset
bad-SYN-ack	DNS-RR-unknown-type	inappropriate-FIN	SYN-inside-connection
bad-TCP-header-len	DNS-truncated-answer	IRC-invalid-line	SYN-seq-jump
base64-illegal-encoding	DNS-len-lt-hdr-len	line-terminated-with-single-CR	truncated-NTP
connection-originator-SYN-ack	DNS-truncated-RR-rdlength	malformed-SSH-identification	unescaped-%-in-URI
data-after-reset	double-%-in-URI	no-login-prompt	unescaped-special-URI-char
data-before-established	excess-RPC	NUL-in-line	unmatched-HTTP-reply
too-many-DNS-queries	FIN-advanced-last-seq	POP3-server-sending-client-commands	window-recision
DNS-label-forward-compress-	fragment-with-DF		

One Day of Crud at ICSI

Postel's Law: *Be strict in what you send and liberal in what you accept ...*

active-connection-reuse	DNS-label-len-gt-pkt	HTTP-chunked-multipart	possible-split-routing
bad-Ident-reply	DNS-label-too-long	HTTP-version-mismatch	SYN-after-close
bad-RPC	DNS-RR-length-mismatch	illegal-%-at-end-of-URI	SYN-after-reset
bad-SYN-ack	DNS-RR-unknown-type	inappropriate-FIN	SYN-inside-connection
bad-TCP-header-len	DNS-truncated-answer	IRC-invalid-line	SYN-seq-jump
base64-illegal-encoding	DNS-len-lt-hdr-len	line-terminated-with-single-CR	truncated-NTP
connection-originator-SYN-ack	DNS-truncated-RR-rdlength	malformed-SSH-identification	unescaped-%-in-URI
data-after-reset	double-%-in-URI	no-login-prompt	unescaped-special-URI-char
data-before-established	excess-RPC	NUL-in-line	unmatched-HTTP-reply
too-many-DNS-queries	FIN-advanced-last-seq	POP3-server-sending-client-commands	window-recision
DNS-label-forward-compress-	fragment-with-DF		**155K in total!**

Is There a Stable Notion of *Normal?*

- Internet traffic is composed of *many* individual sessions.
  - Leads to enormous variety and unpredictable behavior.
- Complex distributions of features:
  - Self-similarity, heavy tails, long-range dependance.
  - Constantly changing.
  - Incessant background noise and tons of crud.
- Observable on all layers of the protocol stack.
- In general, it's pretty much impossible to define "normal".
  - Huge fluctuations are *normal* and *expected* short-term.
  - No attacker needed for that!

Are Outliers Attacks?

- Implicit assumption that anomaly detectors make:

Outliers are malicious!

Are Outliers Attacks?

- Implicit assumption that anomaly detectors make:

Outliers are malicious!

- With such diversity, that can be hard to justify.
- Typical reply: "*I know, anomaly detection doesn't report attacks.*"

Are Outliers Attacks?

- Implicit assumption that anomaly detectors make:

Outliers are malicious!

- With such diversity, that can be hard to justify.
  - Typical reply: "*I know, anomaly detection doesn't report attacks.*"

- That leads to a *semantic gap*.
  - Disconnect between what the system reports and what the operator wants.
  - Root cause for the common complaint of "too many false positives".

Every Mistake Is Expensive

Every Mistake Is Expensive

- Each false alert costs scarce analyst time.
  - Go through log files, inspect system, talk to users, etc.
  - "Trains" the operator to mistrust future alarms.

Every Mistake Is Expensive

- Each false alert costs scarce analyst time.
  - Go through log files, inspect system, talk to users, etc.
  - "Trains" the operator to mistrust future alarms.
- In other domains, errors tend to be cheap.
  - Wrong recommendation from Amazon? Not a big deal.
    Greg Linden: " ... *guess work. Our error rate will always be high.*"
  - Letter misclassified by an OCR system? Spell-checker.
  - Machine translation? Have you tried it?
  - Spam detection? Lopsided tuning.

Every Mistake Is Expensive

- Each false alert costs scarce analyst time.
  - Go through log files, inspect system, talk to users, etc.
  - "Trains" the operator to mistrust future alarms.
- In other domains, errors tend to be cheap.
  - Wrong recommendation from Amazon? Not a big deal.
    Greg Linden: " ... *guess work. Our error rate will always be high.*"
  - Letter misclassified by an OCR system? Spell-checker.
  - Machine translation? Have you tried it?
  - Spam detection? Lopsided tuning.
- What error rate can we afford with an IDS?
  - Base-rate fallacy: false alarm rate must be extremely low.

Relating Features To Semantics

- Key question: What *can* our features tell us?
  - Do packet arrival times tell us something about SQL injection?
  - Do NetFlow records allow us to find inappropriate content?
  - What are the right features to learn how SSNs look like?

Relating Features To Semantics

- Key question: What *can* our features tell us?
  - Do packet arrival times tell us something about SQL injection?
  - Do NetFlow records allow us to find inappropriate content?
  - What are the right features to learn how SSNs look like?
- Need to consider a site's security policy as well.
  - What *is* appropriate content?
  - What *is* tolerable usage of P2P systems?

Relating Features To Semantics

- Key question: What *can* our features tell us?
  - Do packet arrival times tell us something about SQL injection?
  - Do NetFlow records allow us to find inappropriate content?
  - What are the right features to learn how SSNs look like?
- Need to consider a site's security policy as well.
  - What *is* appropriate content?
  - What *is* tolerable usage of P2P systems?
- There are striking examples of how much more information a data set might contain than expected.

Relating Features To Semantics

- Key question: What *can* our features tell us?
  - Do packet arrival times tell us something about SQL injection?
  - Do NetFlow records allow us to find inappropriate content?
  - What are the right features to learn how SSNs look like?
- Need to consider a site's security policy as well.
  - What *is* appropriate content?
  - What *is* tolerable usage of P2P systems?
- There are striking examples of how much more information a data set might contain than expected.
- These exploit *structural* knowledge.
  - Hard to see a classifier just "learning" peculiar activity.

Why is Anomaly Detection Hard?

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Outlier detection and the high costs of errors
Interpretation of results
Evaluation
Training data
Evasion risk

Why is Anomaly Detection Hard?

The intrusion detection domain faces challenges that make it fundamentally different from other fields.

Can we still make it work? Yes, by:

- Limiting the scope
- Gaining insight into the detector's capabilities

Can We Still Make it Work?

Limiting the Scope

- What attacks is the system to find?
  - The more crisply this can be defined, the better the detector can work.
  - Must include a consideration of threat model (environment; costs; exposure).

Limiting the Scope

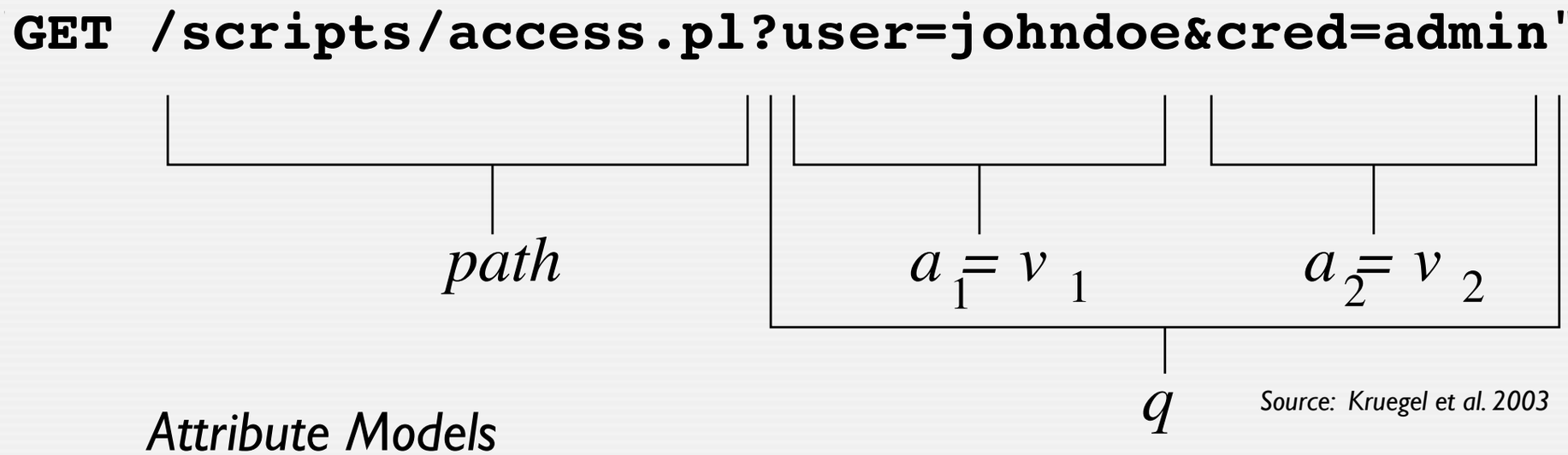
- What attacks is the system to find?
  - The more crisply this can be defined, the better the detector can work.
  - Must include a consideration of threat model (environment; costs; exposure).
- Define a concrete task upfront, e.g.:
  - CGI exploits.
  - Botnet communication.
  - Service usage patterns.
  - Don’t go for the obvious ones ...

Limiting the Scope

- What attacks is the system to find?
  - The more crisply this can be defined, the better the detector can work.
  - Must include a consideration of threat model (environment; costs; exposure).
- Define a concrete task upfront, e.g.:
  - CGI exploits.
  - Botnet communication.
  - Service usage patterns.
  - Don't go for the obvious ones ...
- Define the problem so that ML makes less mistakes:
  - Build a real classification problem.
  - Reduce variability in what's normal.
  - Look for variations of *known* attacks.
  - Use machine-learning as one tool among others.

Focusing On A Specific Problem

Anomaly Detection of Web-based Attacks



Source: Kruegel et al. 2003

Attribute Models

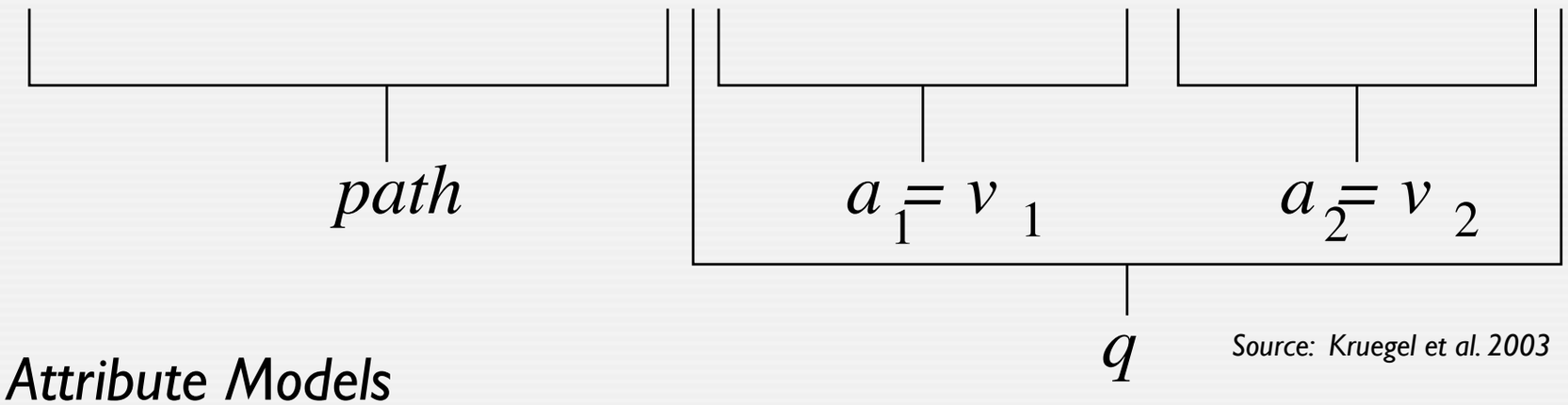
- Length
- Character distribution
- Grammatical structure
- Tokens
- Presence/Absence
- Order of attributes

Focusing On A Specific Problem

Anomaly Detection of Web-based Attacks

```
GET /scripts/access.pl?user=johndoe;SELECT+passwd+FROM+credentials&...
```

GET /scripts/access.pl?user=johndoe&cred=admin"



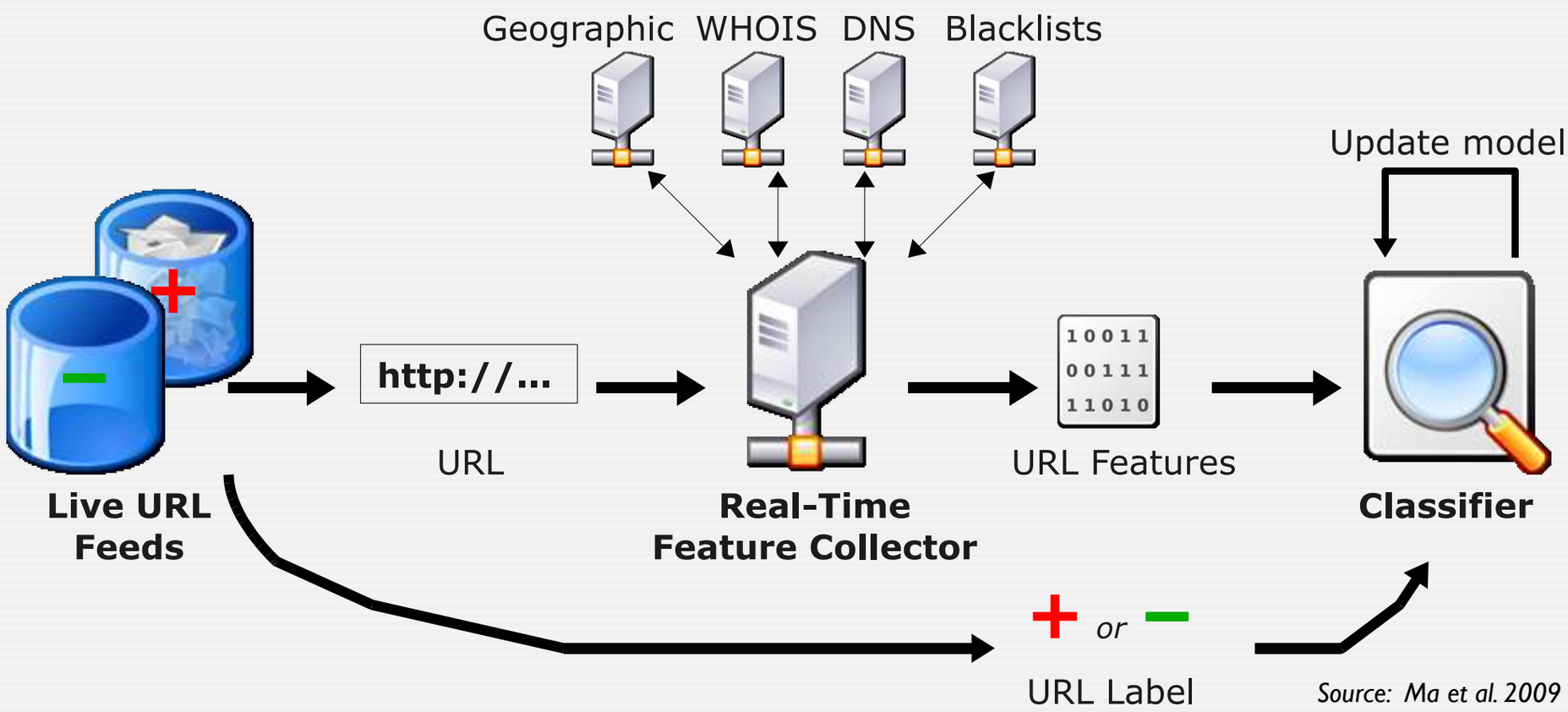
Source: Kruegel et al. 2003

Attribute Models

- Length
- Character distribution
- Grammatical structure
- Tokens
- Presence/Absence
- Order of attributes

Classification

Identifying Suspicious URIs



Source: Ma et al. 2009

Gaining Insight

- A thorough evaluation requires more than ROC curves.
  - It's not a contribution to be slightly better than anybody else on a specific data set.

Gaining Insight

- A thorough evaluation requires more than ROC curves.
  - It's not a contribution to be slightly better than anybody else on a specific data set.
- Questions to answer:
  - What exactly does it detect *and why?*
  - What exactly does it not detect *and why not?*
  - When exactly does it break? (Evasion, performance). ("Why 6?" Tan/Maxion 2001)

Gaining Insight

- A thorough evaluation requires more than ROC curves.
  - It's not a contribution to be slightly better than anybody else on a specific data set.
- Questions to answer:
  - What exactly does it detect *and why?*
  - What exactly does it not detect *and why not?*
  - When exactly does it break? (Evasion, performance). ("Why 6?" Tan/Maxion 2001)
- Acknowledge shortcomings.
  - We are using heuristics, that's ok. But understand the impact.

Gaining Insight

- A thorough evaluation requires more than ROC curves.
  - It's not a contribution to be slightly better than anybody else on a specific data set.
- Questions to answer:
  - What exactly does it detect *and why?*
  - What exactly does it not detect *and why not?*
  - When exactly does it break? (Evasion, performance). ("Why 6?" Tan/Maxion 2001)
- Acknowledge shortcomings.
  - We are using heuristics, that's ok. But understand the impact.
- Examine false positives and negatives carefully.
  - Needs ground-truth, and we should think about that early on.

Gaining Insight

- A thorough evaluation requires more than ROC curves.
  - It's not a contribution to be slightly better than anybody else on a specific data set.
- Questions to answer:
  - What exactly does it detect *and why?*
  - What exactly does it not detect *and why not?*
  - When exactly does it break? (Evasion, performance). ("Why 6?" Tan/Maxion 2001)
- Acknowledge shortcomings.
  - We are using heuristics, that's ok. But understand the impact.
- Examine false positives and negatives carefully.
  - Needs ground-truth, and we should think about that early on.
- Examine true positives and negatives as well.
  - They tell us *how* the detector is working.

Image Analysis with Neural Networks

Tank

Image Analysis with Neural Networks

Tank

No Tank

Image Analysis with Neural Networks

Tank

No Tank

Image Analysis with Neural Networks

Tank

No Tank

Image Analysis with Neural Networks

Ta ank

Bridge the Gap

- Assume the perspective of a network operator
  - How does the detector help with *operations*?
  - With an anomaly reported, what should the operator do?
  - How can local policy specifics be included?

Bridge the Gap

- Assume the perspective of a network operator
  - How does the detector help with *operations*?
  - With an anomaly reported, what should the operator do?
  - How can local policy specifics be included?

- Gold standard: work *with* the operators
  - If they deem the detector useful in daily operations, you got it right.
  - Costs time and effort on both sides however.

Once You Have Done All This ....

... you might notice that you now know enough about the activity you're looking for that you *don't need any machine-learning*.

- ML can be a tool for illuminating the problem space.
- Identify which features contribute most to outcome.
- ... to then perhaps build a non-machine learning detector.

Conclusion

Summary

- Approaches to network intrusion detection.
  - Host-based vs. network-based detection.
  - Misused detection and anomaly detection.
- Why is anomaly detection so hard?
  - Outlier detection and the high costs of errors.
  - Interpretation of results.
  - Evaluation.
  - Training data.
  - Evasion risk.
- Use care with machine-learning:
  - *Limit the scope* of the problem.
  - *Gain insight* into what the system does.

Summary

- Approaches to network intrusion detection.
  - Host-based vs. network-based detection.
  - Misused detection and anomaly detection.
- Why is anomaly detection so hard?

Open questions

"Soundness of Approach: *Does the approach actually detect intrusions? Is it possible to distinguish anomalies related to intrusions from those related to other factors?"*

-Denning, 1987

Thanks for your attention.

Robin Sommer

International Computer Science Institute, &

Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu

http://www.icir.org