Probabilistic Graph Models:
Factor Graphs

ECE/CS 498 DS U/G

Lecture 20: Course Feedback, MP3, Factor Graphs

Ravi K. Iyer

Dept. of Electrical and Computer Engineering

University of Illinois at Urbana Champaign

Announcements

- Course Timeline:
  - Mon 4/6:
    - Review of course feedback
    - Intro to MP 3
    - Introduction to factor graphs
  - Wed 4/8: Factor graphs and belief propagation
- Discussion section on Friday 4/10
  - Talk through MP 3 for ~15 minutes
  - Office hours with the TA for remaining ~45 min
- Final Project
  - Progress report 2 due **Friday April 17 @ 11:59 PM** on Compass2G
    - There should be *substantial* progress with projects by this point (i.e. meaningful results, ML/AI models)

Course Feedback

Course Feedback Results - MPs

Category	MP 1 Average Score	MP 2 Average Score
Difficulty (1: too easy, 5: too hard)	3.17	3.65
Length (1: too short, 5: too long)	3.32	3.61
Task Specification (1: too vague, 5: clearly specified)	3.65	2.92
Time Given (1: too little, 5: ample time)	4.08	3.73
Effort for Coding vs Concepts (1: balanced time with code and concepts, 5: too much time with code)	2.37	2.51
Learning various concepts (1: did not learn much, 5: learned a lot)	Basic Probability: 4.23 Naïve Bayes: 4.28	GMM Clustering: 4.08 K-Means Clustering: 4.08 PCA: 4.13

Course Feedback Results – Exam, Final Project, and Logistics

- Midterm Exam:
  - Difficulty (1: too easy, 5: too difficult) - **3.75**
  - Length (1: too long, 5: too short) - **2.07**
  - Content (1: not fair, 5: fair) - **3.48**
- Final Project:
  - Individualized Guidance (1: not helpful, 5: helpful) – **3.98**
  - Pace (1: too slow, 5: too fast) - **3.29**
- Piazza response times were good: **3.87 / 5**
- Office hours were helpful: **3.86 / 5**
- Effectiveness of lecture delivery with Zoom: **3.82 / 5**

Course Feedback Results – Addressing Suggestions

- As a reminder, all lecture videos are posted online: https://mediaspace.illinois.edu/channel/ECE+498+DSG+-+Spring+2020
- To help aid student understanding, more practice problems/examples will be covered in lecture
- Continue to ask questions on Zoom if you have confusion/doubts
  - Please "Raise your Hand" until a TA unmutes you to ask your question
  - Make use of office hours to clarify any additional doubts on lecture or MP material
- If movement to online course (due to COVID-19 protection) causes unmanageable difficulty for you, please contact course staff about potentially taking this course via credit/no-credit option

Introduction to MP 3

Problem Statement

- **Goal**: Understand the progression of a multi-stage attack that aims to leak secret keys from target system
  - Perform analysis of a multi-stage attack recreated from publicly available information on the Equifax breach
  - Explore the use of signature-based, anomaly-based, and factor graph-based techniques

Task Description

- **Task 0:** Parse the raw data into a more analysis-friendly format and identify attack related files (PyShark, pandas)
- **Task 1:** Compare the attacker’s behavior vs. legitimate users’ behavior (PyShark)
- **Task 2:** Build a simple factor graph for a single event and a single attack stage (pgmpy)
- **Task 3:** Extend the factor graph to capture the evolution of a series of events to infer the attack state (hidden) and decide the action to take (pgmpy)
- **Task 4:** Discuss about another similar attack and see if the factor graph in Task 3 can be used in this different attack

MP 3 Timeline/Schedule

- **Checkpoint 1: Monday 4/13 @ 11:59 PM via Compass**
  - .ipynb file with Tasks 0-1 completed
  - PDF of slides with answers to Tasks 0-1
- Checkpoint 1.5: Monday 4/20 @ 11:59 PM via Google Form
  - Informal progress update
  - Include image of factor graph from Task 3.0
- Checkpoint 2 (Final): Wed 4/29 @ 11:59 PM via Compass
  - .ipynb file with Tasks 0-4 completed
  - PDF of slides with answers to Tasks 0-4

Objective: Use real incident data to pre-empt attacks

- Mine patterns of alerts prior to the attack onset in real incidents

- Measure the reliability of these patterns using randomness tests.

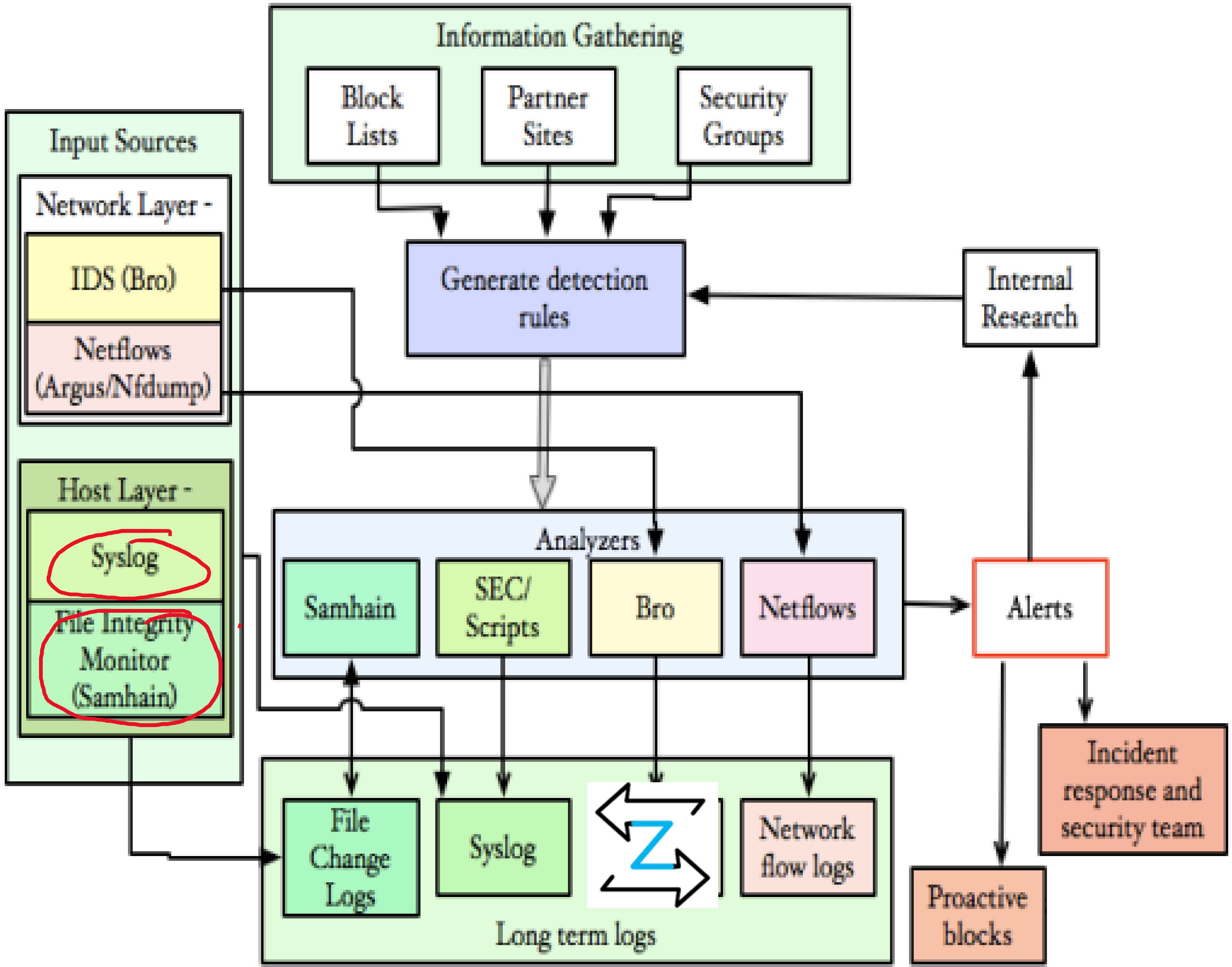
- Design pre-emption techniques, *to provide attack warning sufficiently in advance to system misuses*, to reduce missed incidents and false positives

- Develop a testbed to measure the efficacy of preemptiveness on new attacks that intermingle with legitimate traffic in production network

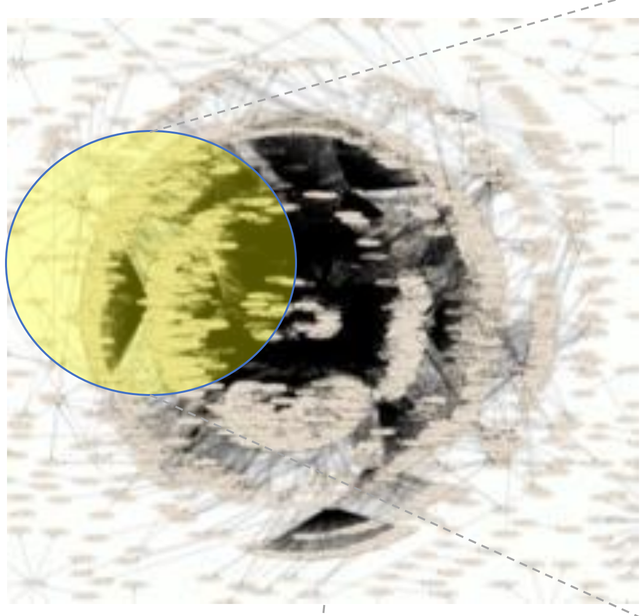
NCSA: Target system and security monitors

- Large-scale interconnected systems
- Robust monitoring tools (Zeek) as used in national labs
- Extensive operational and development knowledge w/ Zeek

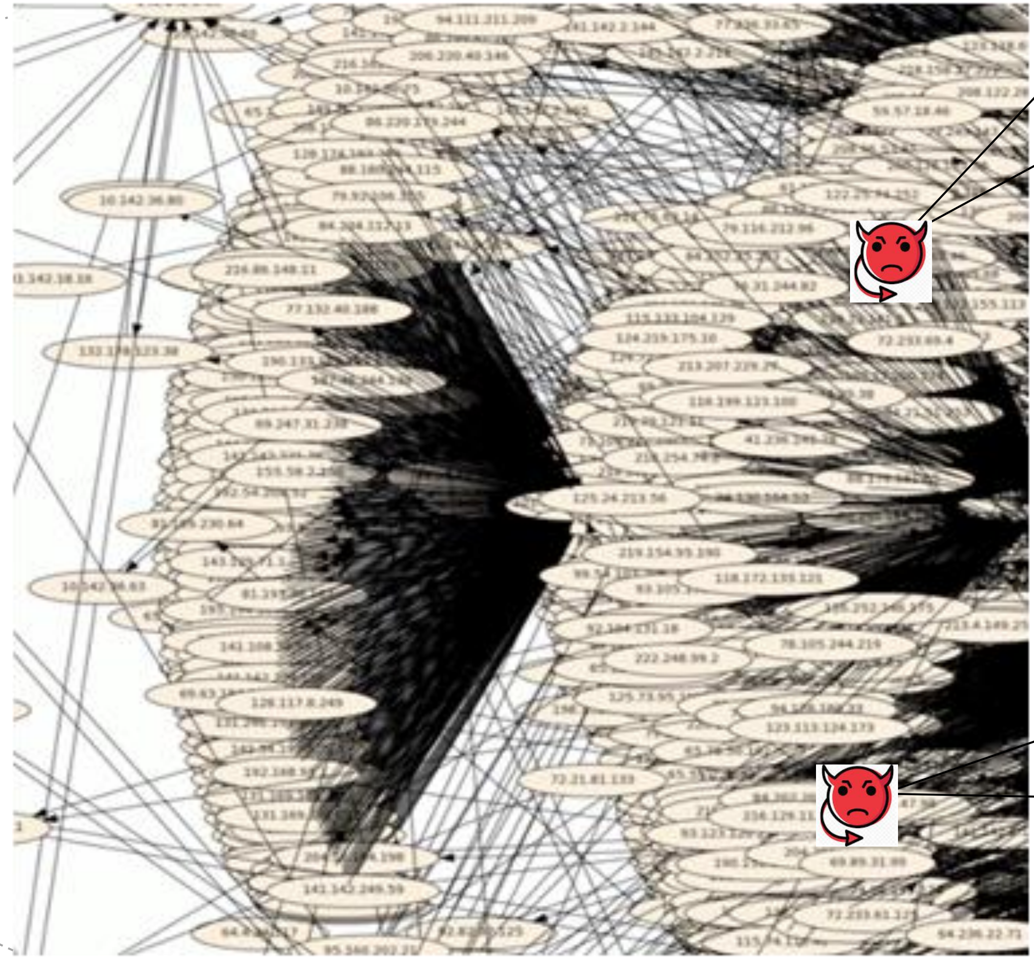
Number of hosts	5000+ (clusters, workstations, laptops)
Number of Active Users:	6000+
Network	Class B (/16)
Monitoring Links	40Gb pipes
Monitoring Tools	- IDS (4.5GB daily logs) - Network Flow (2.0G) - File integrity check - Central Syslog (1.5G)
OS Types	Linux, AIX, Solaris, OS-X, Windows



Measurements from NCSA@Illinois:
Five-minute snapshot shows dense network traffic



Dense network connections



ATTACK 1 CREDENTIAL STEALING

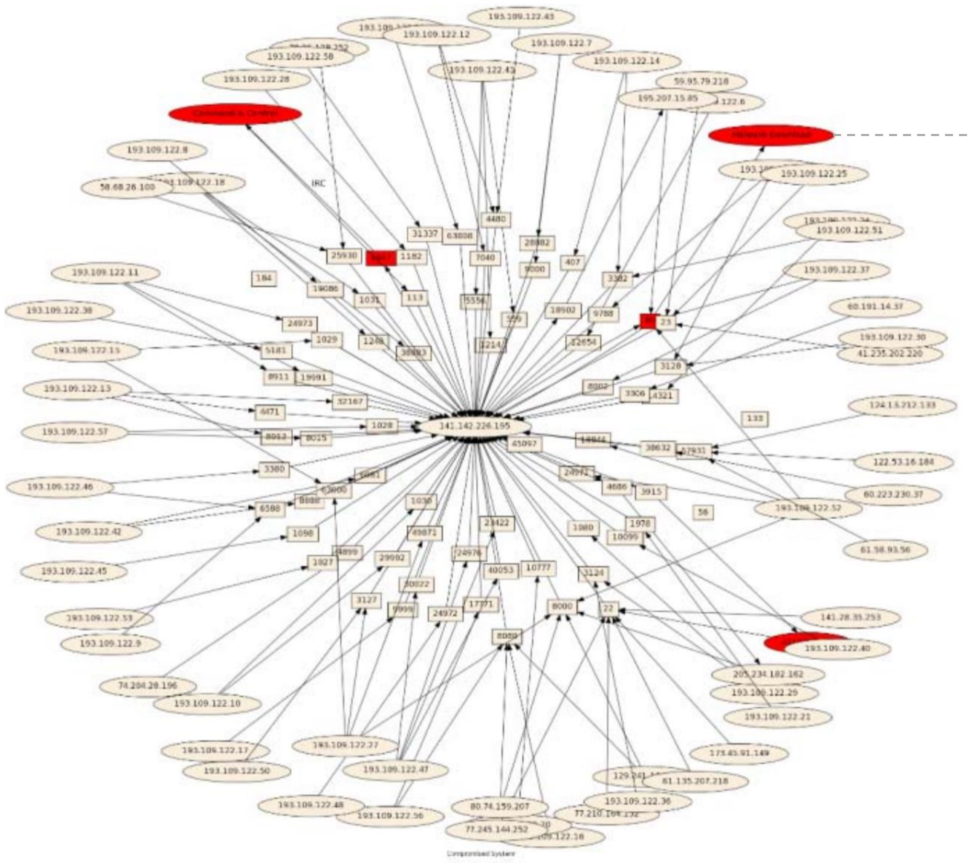
Five-Minute Snapshot of In-and-Out Traffic at NCSA

ATTACK 2 DATA EXTRACTION

THIS TRAFFIC PRESENTS A BIG DATA PROBLEM!

Challenges: partial view of maturing attacks

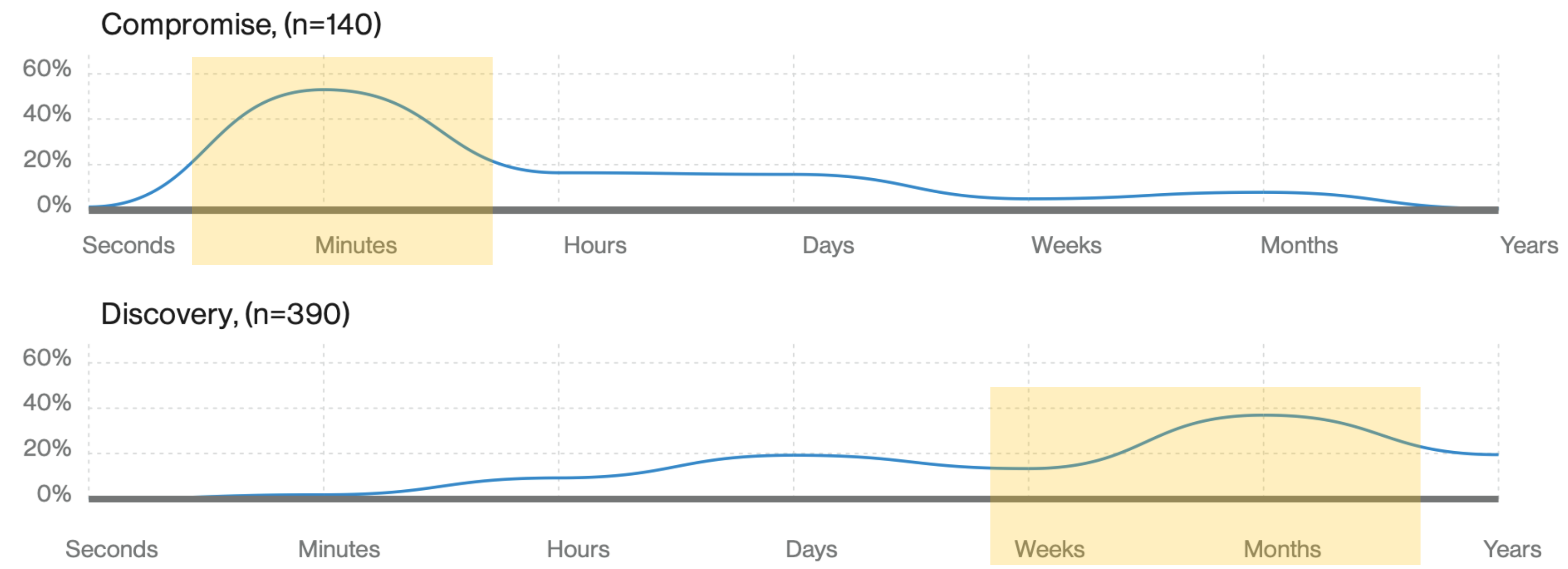
- Challenges
  - Big data
  - **Partial view of maturing attacks from host/network alerts**



- Attacks constitute a tiny fraction of alerts and traffics.
- These attacks are only visible after-the-fact (forensic investigation)
- Determining attacks (red nodes) in advance is difficult.

Challenges: Fast attacks, slow detection

- Challenges:
  - Big data
  - Partial view of attacks
  - **Fast attacks, slow detection**



NCSA

62% (23/37) OF HIGH-SEVERITY INCIDENTS WERE CAUGHT IN THE BREACH-PHASE, HAVING ALREADY RESULTED IN SIGNIFICANT DAMAGE – STOLEN CRED.

verizon

THE ATTACK MATURE IN FEW MINUTES, WHILE FORENSIC DIAGNOSIS TAKES HOURS OR DAY

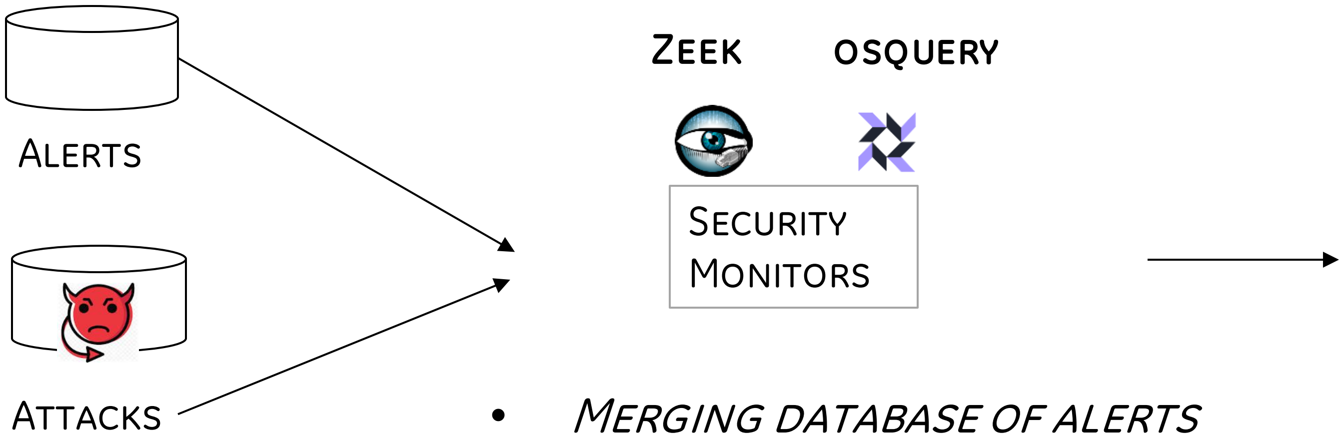
MY GOAL: PREEMPT THE ATTACK IN ADVANCE BEFORE SYSTEM MISUSE.

Challenges: abundant alerts

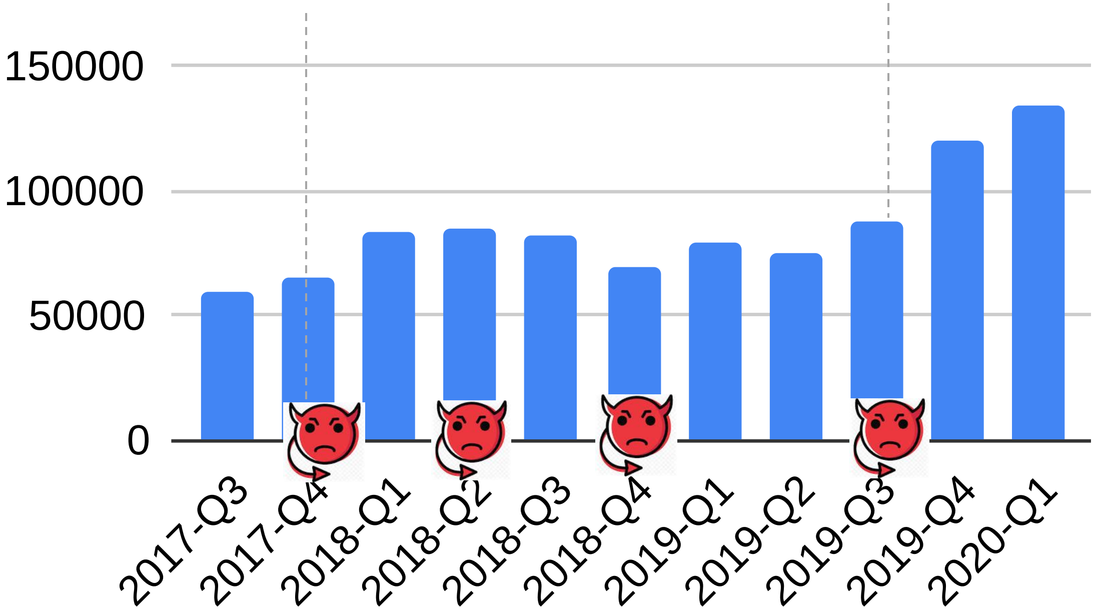
- Challenges:
  - Big data
  - Partial view of attacks
  - Fast attacks, slow detection
  - **Many alerts, but few actual attacks**

- *Avg. 80,000 alerts/day, but a few successful attacks/year.*

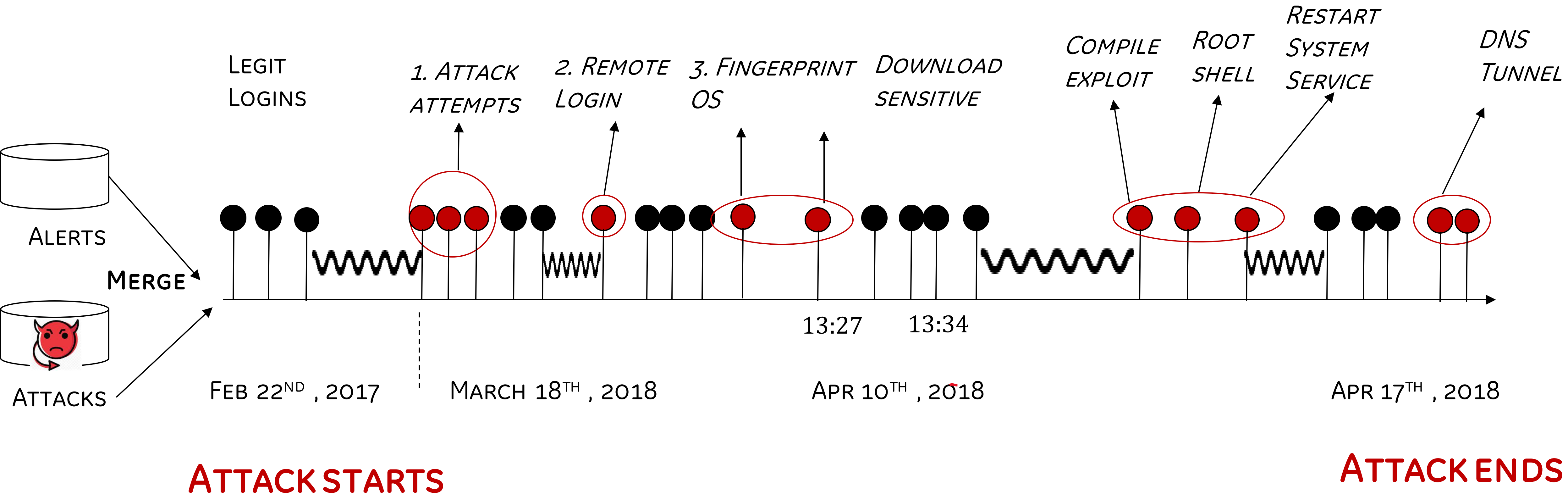
- *Very few (< 10 alerts) precede successful attacks*



- *Merging database of alerts and attacks to find alerts preceding attacks*



Attack 1. stolen credential attack that has not been discovered in a month



- Black circles are regular alerts in the system
- Red circles are actual attack correlated based on a user's IP address and/or user identifier.

We focus on alerts preceding attacks

- *How often patterns of alerts occur in the data?*
- *Are the patterns random or they have a causal effect?*
- *Given any attack, how likely we see a particular pattern to occur? (conditional probability)*

Reasoning about patterns

- Alerts occur as clusters, grouped by time proximity and IP address
- Alerts are repeated among attacks
- Some alerts are found in both legitimate users and attackers, thus are not always reliable

(Example: Logging in from a remote site appears in both users traveling and attackers)

1. Show statistical evidence of patterns (conditional probabilities)
2. Use randomness testing to validate the predictive power of a pattern
3. Encode patterns and probabilities into a detection model

(which is a Probabilistic Graphical Model.)

- **An IDS alert shows suspicious download** on a production system (victim: *xx.yy.ww.zz*) using HTTP protocol from remote host *aa.bb.cc.dd*.

```
May 16 03:32:36 %187538 start xx.yy.ww.zz:44619 > aa.bb.cc.dd:80
May 16 03:32:36 %187538 GET /.0/ptrat.c (200 "OK" [2286] server5.bad-host.com)
```

- The file is suspect because
  - This particular system is not expected to download any code apart from patches and system updates, and then only from authorized sources
  - The downloaded file is a C language source code
- The server the source was downloaded from not a formal software distribution repository.

Time

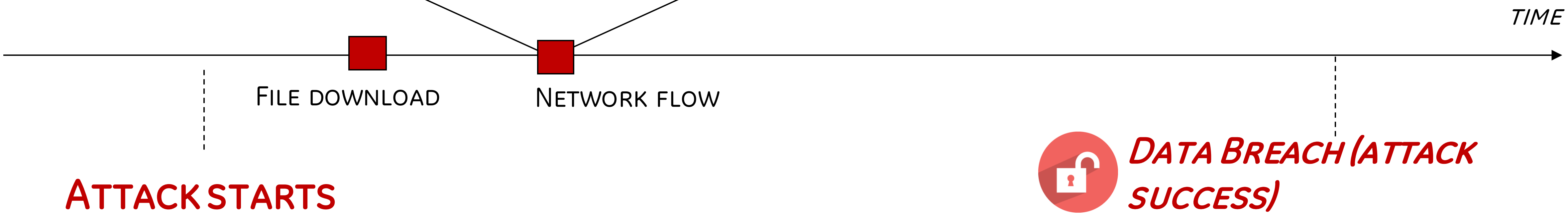
File download

Attack starts

Data Breach (attack success)

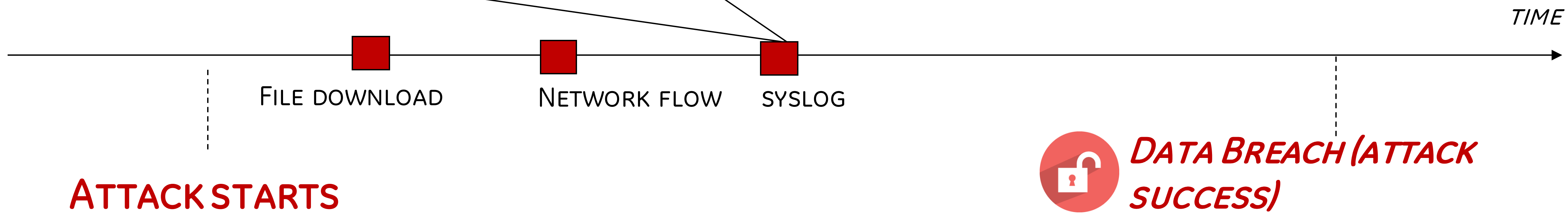
- **Network flows reveal further connections with other hosts** in close time proximity to the occurrence of the download:
  - SSH connection from IP address 195.aa.bb.cc
  - Multiple FTP connections to ee.ff.gg.hh, pp.qq.rr.ss.

```
09-05-16 03:32:27 v tcp 195.aa.bb.cc.35213 -> xx.yy.ww.zz.22 80  96  8698 14159   FIN
09-05-16 03:33:36 v tcp xx.yy.ww.zz.44619 -> aa.bb.cc.dd.http 8  6  698 4159   FIN
09-05-16 03:34:37 v tcp xx.yy.ww.zz.53205 -> ee.ff.gg.hh.ftp 1699 2527 108920 359566
FIN
09-05-16 03:35:39 v tcp xx.yy.ww.zz.39837 -> pp.qq.rr.ss.ftp 236  364  15247  546947 FIN
```



- *SYSLOG* CONFIRMS A USER LOGIN FROM *195.AA.BB.CC*, WHICH IS UNUSUAL, BASED ON THE KNOWN USER PROFILE AND BEHAVIOR PATTERNS

```
May 16 03:32:27 host sshd[7419]: Accepted password for user from 195.aa.bb.cc port 35794 ssh2
```



- Search of all files owned or created by this user found a footprint left behind by a credential-stealing exploit.

```
-rwxrwxr-x 1 user user 3945 May 16 03:37 /tmp/libno_ex.so.1.0
```

- *The additional analysis showed*
  - *The library file libno_ex.so.1.0 is known to be created when an exploit code for a known vulnerability (CVE-2009-1185) is successfully executed*
  - *File is owned by the user whose account was stolen and used to login to the system*
  - *The attacker obtained root privileges in the system and replaced the SSHD daemon with a trojaned version*
    - *Harvesting more user credentials*

Time

File download

Network flow

syslog

File system analysis

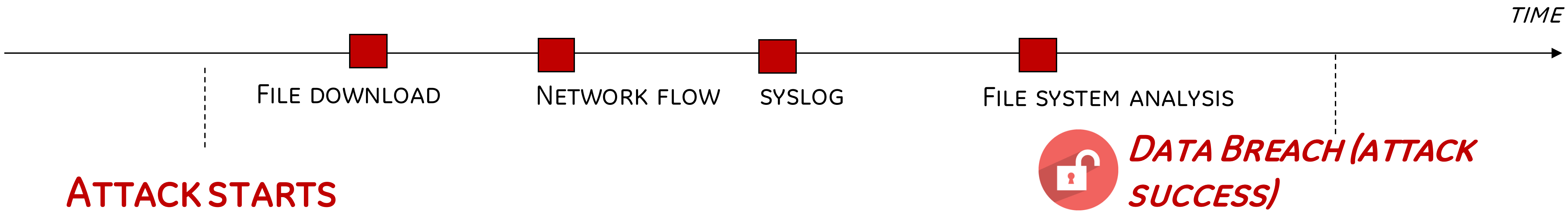
Attack starts

Data Breach (attack success)

How does a security expert analyze the attack?

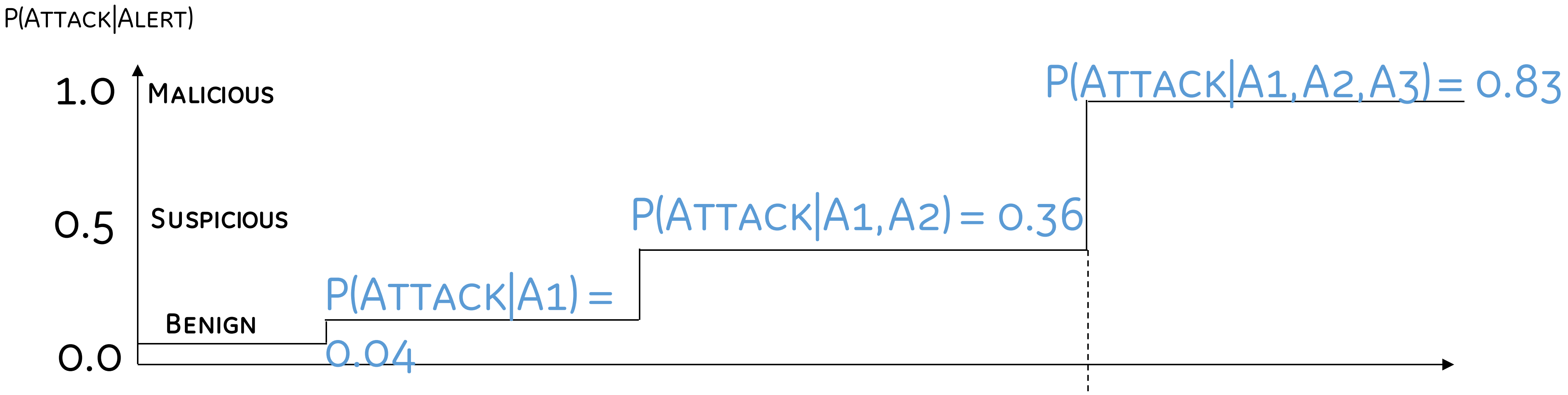
Four data points established from the analysis

1. ***A suspicious source code was downloaded,***
2. ***The user login occurred at nearly the same time as the download,***
3. ***First time login from IP address 195.aa.bb.cc,***
4. ***Additional communication on other ports (FTP)***



How do we automate this reasoning ?

THE SUSPICION LEVEL, P(ATTACK|ALERT), INCREASES AS ALERTS ARE OBSERVED.

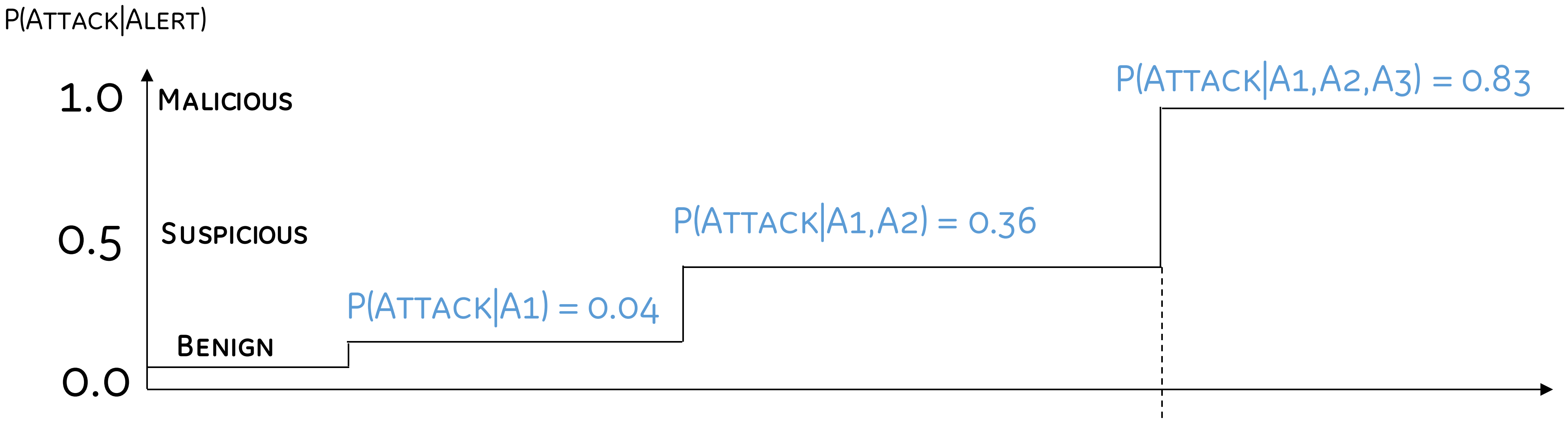


A1. REMOTE LOGIN

A2. OS FINGERPRINTING

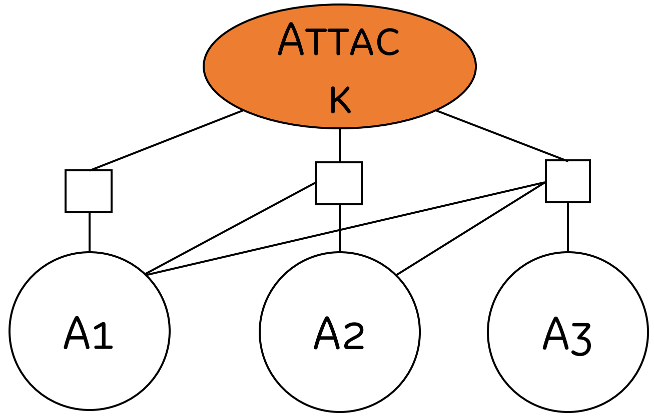
A3. DOWNLOAD SENSITIVE FILES

The suspicion level, P(Attack|Alert), increases as alerts are observed.



Query: Attack = 0 or 1 ?

The question is, given this pattern,
how likely is an attack is progressing?



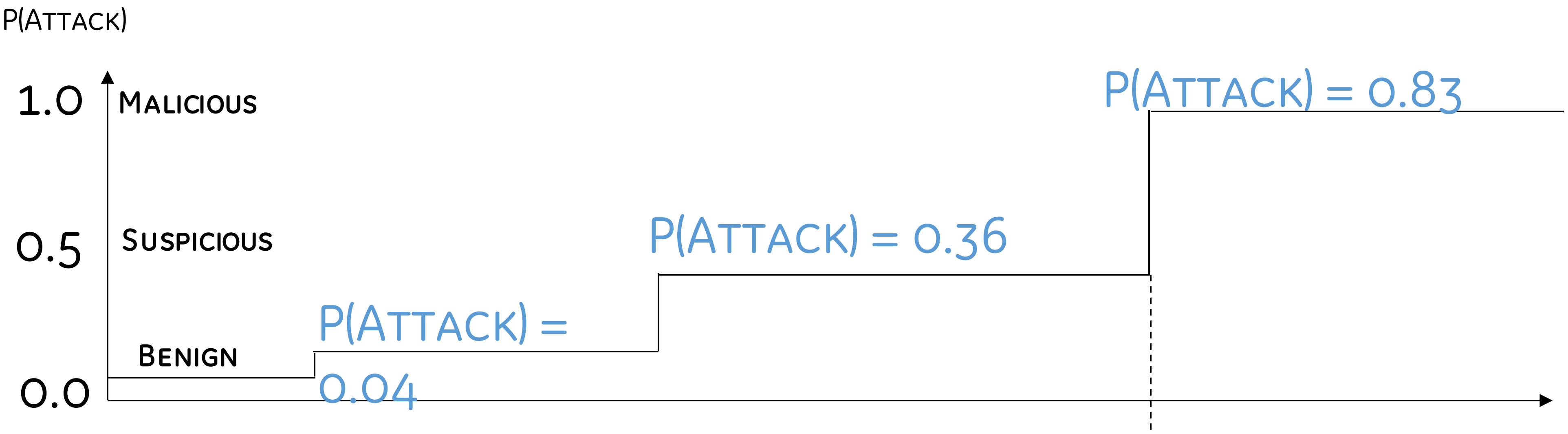
P(Attack, A1, A2, A3)

P(Attack = 0, A1, A2, A3)

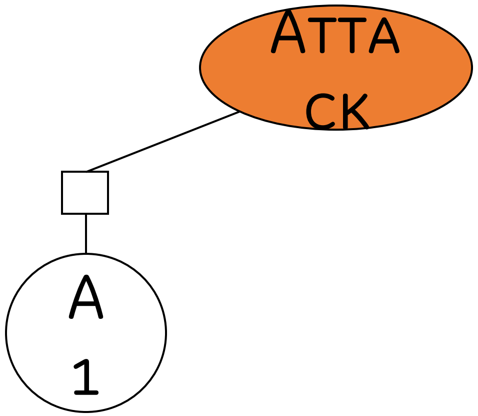
P(Attack = 1, A1, A2, A3)

P(Attack = 1, A1, A2, A3) = F1(Attack=1,A1) F2 (Attack=1, A1, A2) F3(Attack = 1, A1,A2,A3)

The suspicion level, P(Attack|Alert), increases as alerts are observed.



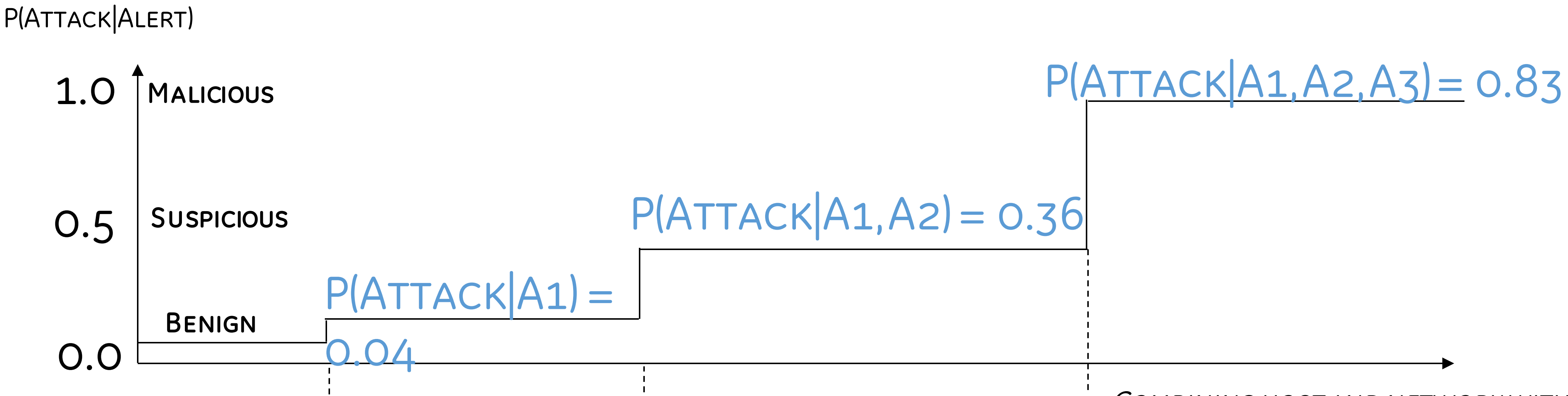
P(Attack=1) = 1/Z * f(Attack=1,A1) = 0.04



Based on past data, we count how many successful attacks does this pattern by itself indicate and how many times the pattern appears in the entire data.

Count (Attack=1, A1)
Count (Attack=0, A1)
f(Attack,A1) = Count(Attack=1, A1) / (Count(Attack=0, A1) + Count(Attack=1,A1))

The suspicion level, P(Attack|Alert), increases as alerts are observed.



At network level, remote logins are common as people travel off-campus.

Cannot conclude an attack at this point.

At host level, OS fingerprinting correlates with maintenance activities, which typically performed on-site instead of off-campus.

Performing both OS fingerprinting off-site is suspicious.

Combining host and network with the act of download a source code file (.c) raises the suspicion level, exceeding threshold of 0.5.

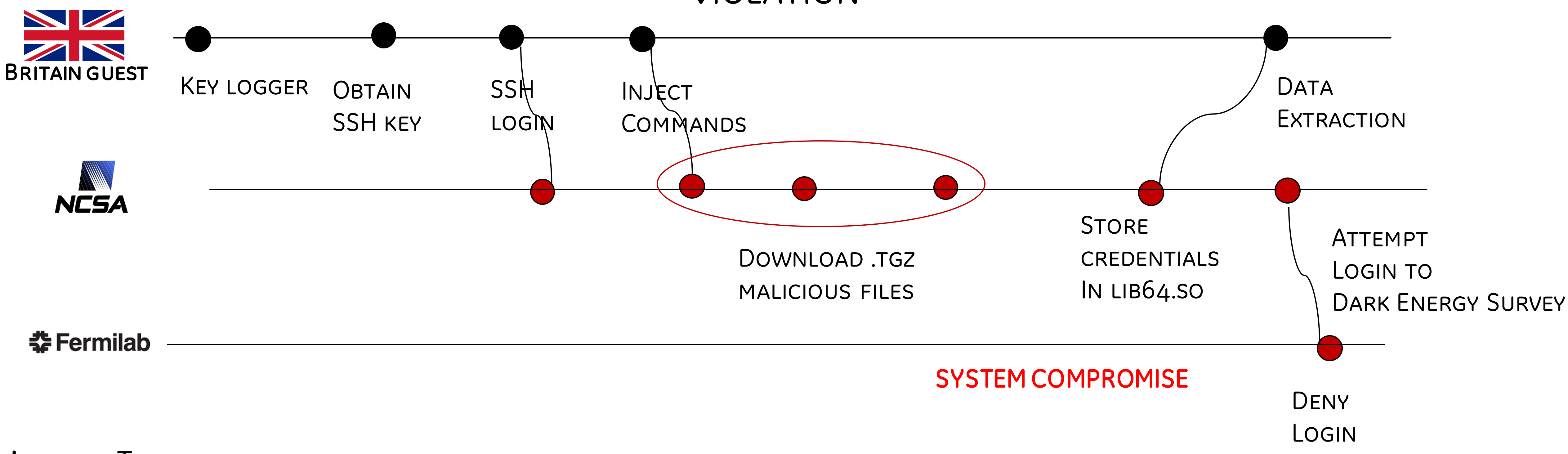
The attack is concluded before the attacker misuses the system.

A1. Remote login

A2. OS fingerprinting

A3. Download sensitive files

Attack 1: A credential stealing attack that was detected after system integrity violation



Impact. The attacker
- Stayed in the system for a month
- Collected credentials of three subsequent logins
- Attempted (but failed) to compromise the computing nodes at Fermi lab.

Why did the attack happen? The attack was not preempted because of insufficient evidence (commands were not recorded on the host).

The security team only had network traces.

Analysis of an Example Incident

(Credentials Stealing Category)

- **An IDS alert shows successful remote login to** a production system, Dark Energy Survey, (*141.142.ww.zz*) using ssh protocol from many different remote hosts

Date/Time	IP Address
2018-04-10 13:27	113.108.
2018-04-10 13:29	113.108.
2018-04-10 13:33	113.108.
2018-04-10 13:36	113.108.
2018-04-11 05:08	159.226.
2018-04-11 13:59	159.226.
2018-04-12 04:14	62.210.1
2018-04-13 07:02	159.226.
2018-04-13 14:43	159.226.
2018-04-15 05:56	159.226.
2018-04-16 05:05	159.226.
2018-04-16 05:06	159.226.

- The activity is suspect beca
  - The user was not traveling to those countries corresponding to the hosts
  - The user's credentials has been modified, rendering the user unable to login.
- ***The alerts do not reveal what attacker did on the compromised production host system.***

Correlation with network logs

- **Network flows reveal further download of sensitive files in close time proximity**

•	2018-04-10T13:27	181.215.zz.xx:24221/op3.tgz
•	2018-04-10T13:34	181.215.zz.xx:24221/sp.tgz

- These flows are suspect because
  - The downloads are for direct IP address, skipping legitimate domain name resolution (DNS) protocols.
  - The files are downloaded via HTTP protocol (usually port 80), but the server IP addresses are non-standard (24221)
- The server the source was downloaded from not a formal software distribution repository.
- ***The alert does not reveal what caused the potentially illegal download request***

Correlations with host logs

- **Further analysis of the host reveal that the OpenSSH server /usr/bin/ssh has been modified.**

```
The file, op3.tgz, is the source code for OpenSSH v5.3.p1

A key logger injected into OpenSSH to redirect ssh login
credentials to a file, '/usr/lib64/.lib/lib64.so'.
```

These activities are suspect because

The OpenSSH servers never are compiled manually, rather the OpenSSH server must be obtained from official software distribution package during maintenance.

The ".lib" directory is hidden when running standard UNIX list directory (ls) command

The lib64.so file is a text file of stolen credential, but its name masquerades as binary system file.

- ***Historical commands on the host reveal that the attacker attempted to connect to another iForge computing cluster, but was not successful.***

Preempting the above incident

- ***Four data points established from the analysis***
  - ***Multiple login attempts from remote countries affecting legitimate user logins***
  - ***The user login occurred at nearly the same time as the download of suspicious files from remote servers.***
  - ***SSH binary was compiled manually outside of maintenance window.***
  - ***Failed connection attempts to internal hosts (iForge)***

THE INCIDENT COULD HAVE BEEN PREEMPTED BEFORE DATA EXFILTRATION OF STOLEN CREDENTIA

My Research Focus

Fast compromise

My goal: **Preempt intrusion before system misuse**,
while leveraging a rich dataset of real attacks in an operational network.

Slow detection

Introduction to Factor Graphs

Hidden Markov Models

Model

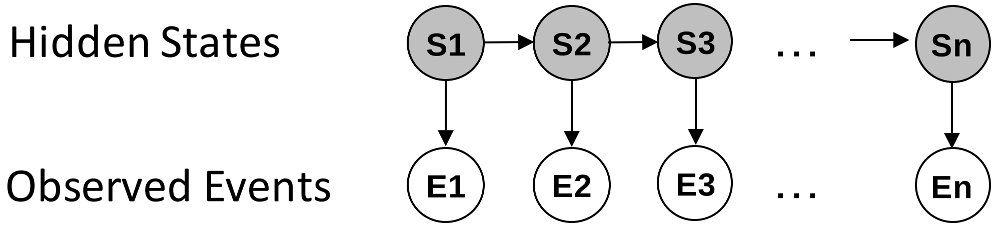
- Set of hidden states $\boldsymbol{S} = \{\boldsymbol{\sigma_1}, \dots, \boldsymbol{\sigma_N}\}$
- Set of observable events $\boldsymbol{E} = \{\boldsymbol{\epsilon_1}, \dots, \boldsymbol{\epsilon_M}\}$
- Transition probability matrix $\boldsymbol{A}$
- Observation matrix $\boldsymbol{B}$
- Initial distribution of hidden states $\boldsymbol{\pi}$

Model assumptions

- An observation depends on its hidden state
- A state variable only depends on the immediate previous state (Markov assumption)
- The future observations and the past observations are conditionally independent given the current hidden state

Advantages:

- HMM can model sequential nature of input data (future depends on the past)
- HMM has a linear-chain structure that clearly separates system state and observed events.

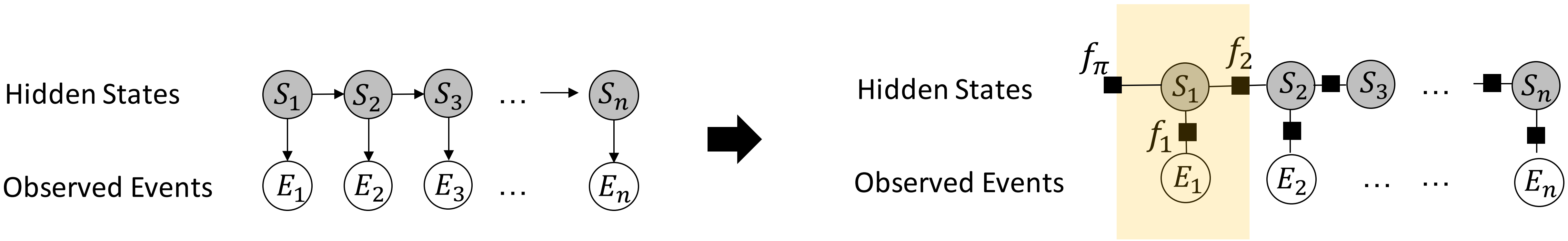


A Hidden Markov model on observed events and system states

$$P(S_1, \dots, S_n, E_1, \dots, E_n) = P(S_1)P(E_1|S_1)\prod_{i=2}^{n} P(S_i|S_{i-1})P(E_i|S_i)$$

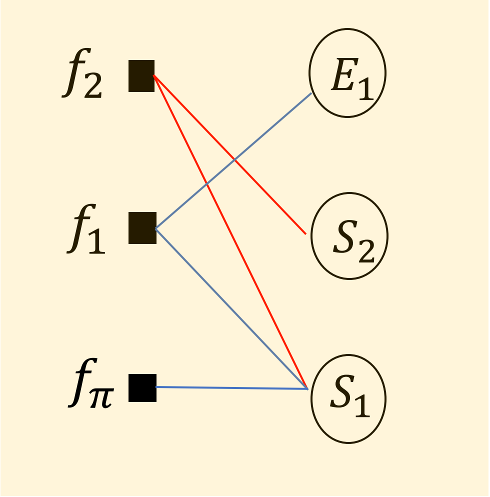
Conversion of a Hidden Markov Model to a Factor Graph

Hidden Markov Model **Factor Graph of the HMM**



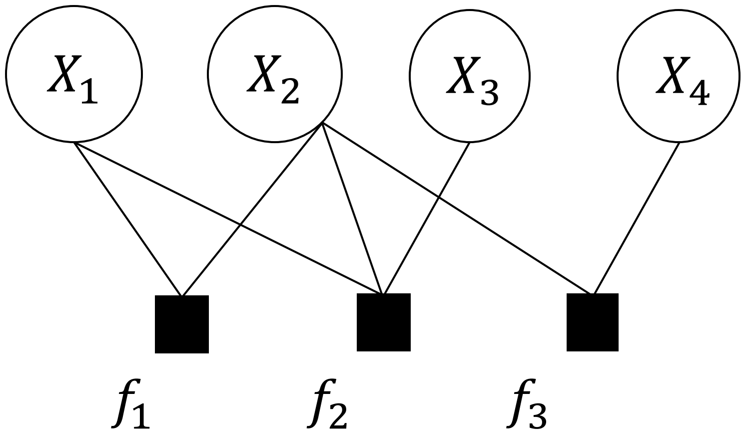
The above *Factor Graph* (FG) is a generalization of the Hidden Markov Model

- Boxes (f_π, f_1, f_2) represents factor function
- In the above case, it maintains the Markov assumption between states



Bipartite graph representation of the FG

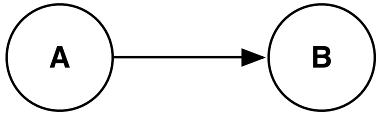
Definition of a Factor Graph



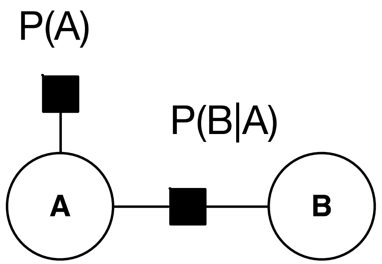
A factor graph is a **bipartite, undirected graph** of **random variables and factor functions. [Frey et. al. 01].**

G (graph) = (X,f,E); E denotes the edges

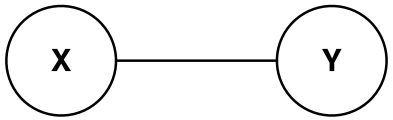
*FG can represent both **causal and non-causal relations**.*



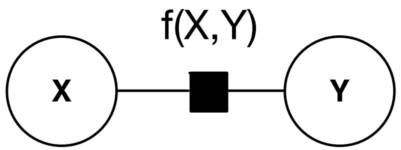
Bayesian Network (BN)



Factor Graph equivalent of BN



Undirected Graph



Factor Graph equivalent of UG

Example Factor function for HMMs

Assume that the state space and observation space are $S = \{\sigma_0, \sigma_1\}$, $E = \{\epsilon_1, \epsilon_2\}$. An example of factor functions is shown.

S	$f_\pi(S)$
σ_0	40
σ_1	25

S_t	E_t	$f_1(S_t, E_t)$
σ_0	ϵ_1	20
σ_0	ϵ_2	15
σ_1	ϵ_1	40
σ_1	ϵ_2	3

S_t	S_{t+1}	$f_2(S_t, S_{t+1})$
σ_0	σ_0	5
σ_0	σ_1	1
σ_1	σ_0	10
σ_1	σ_1	15

- Factor values represents the *affinities* between the related variables
  - E.g., $f_1(\sigma_1, \epsilon_1) > f_1(\sigma_0, \epsilon_1)$ implies that σ_1 and ϵ_1 are more compatible than σ_0 and ϵ_1
- Factor functions don't necessarily represent PDs or joint probability distributions
- How are these values found?
  1. Given by expert or from domain knowledge
  2. Derived from the data (priors)

Definition of Factor functions

Definition:

- Let $\boldsymbol{D}$ be a set of random variables. We define a factor f to be a function from $Val(D)$ to $\mathbb{R}$. A factor is non-negative if all its values are non-negative.
- The set of variables $\boldsymbol{D}$ is called the scope of the factor f and is denoted as $Scope(f)$.
- $Val(\boldsymbol{D})$ represents the set of values $\boldsymbol{D}$ can take.

Example:

A	B	$f(A,B)$
a_0	b_0	30
a_0	b_1	5
a_1	b_0	1
a_1	b_1	10

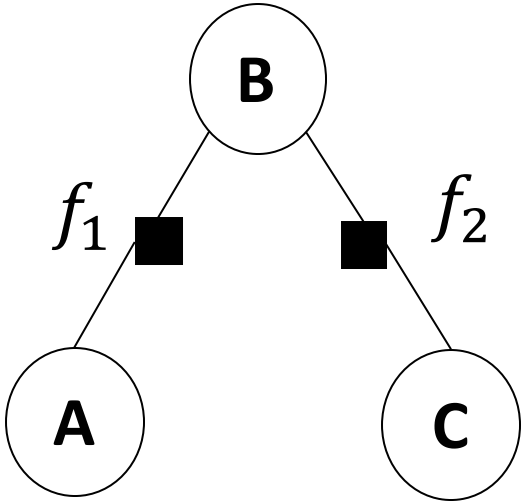
$\boldsymbol{D} = \{A, B\}$
$Val(\boldsymbol{D}) = \{(a_0, b_0), (a_0, b_1), (a_1, b_0), (a_1, b_1)\}$

$A \in \{a_0, a_1\}$

$B \in \{b_0, b_1\}$

Product of Factor Functions in a Factor Graph

- In HMMs, we derived the joint distribution from the graph representation: $P(S_1, \ldots, S_n, E_1, \ldots, E_n) = P(S_1)P(E_1|S_1)\prod P(S_i|S_{i-1})P(E_i|S_i)$
- For a Factor Graph, the joint distribution can be derived from the product of factor functions (given that all factor functions are non-negative)



Example Factor Graph over variables A, B, C.

$$P(A, B, C) = \frac{1}{Z} f_1(A, B) f_2(B, C)$$

where, the normalization Z is given as

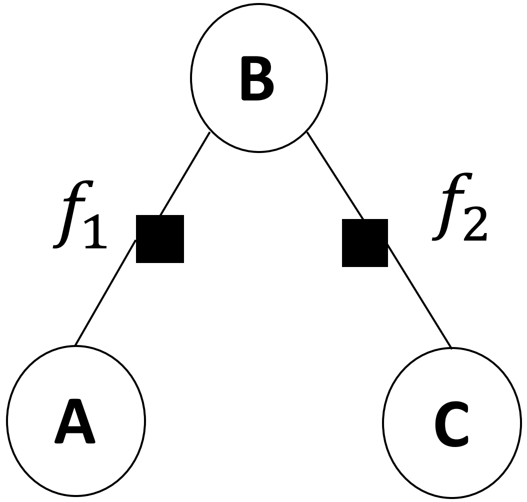
$$Z = \sum_{A,B,C} f(A, B, C) = \sum_{A,B,C} f_1(A, B) f_2(B, C)$$

Z is also referred to as the *partition function.*

Example of product of factor functions

Two factors f_1 and f_2 are multiplied in a way that "matches up" the common variables

$$f(A, B, C) = f_1(A, B) f_2(B, C)$$



f_1

a^1	b^1	0.5
a^1	b^2	0.8
a^2	b^1	0.1
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.9

f_2

b^1	c^1	0.5
b^1	c^2	0.7
b^2	c^1	0.1
b^2	c^2	0.2

f

a^1	b^1	c^1	$0.5 \cdot 0.5 = 0.25$
a^1	b^1	c^2	$0.5 \cdot 0.7 = 0.35$
a^1	b^2	c^1	$0.8 \cdot 0.1 = 0.08$
a^1	b^2	c^2	$0.8 \cdot 0.2 = 0.16$
a^2	b^1	c^1	$0.1 \cdot 0.5 = 0.05$
a^2	b^1	c^2	$0.1 \cdot 0.7 = 0.07$
a^2	b^2	c^1	$0 \cdot 0.1 = 0$
a^2	b^2	c^2	$0 \cdot 0.2 = 0$
a^3	b^1	c^1	$0.3 \cdot 0.5 = 0.15$
a^3	b^1	c^2	$0.3 \cdot 0.7 = 0.21$
a^3	b^2	c^1	$0.9 \cdot 0.1 = 0.09$
a^3	b^2	c^2	$0.9 \cdot 0.2 = 0.18$

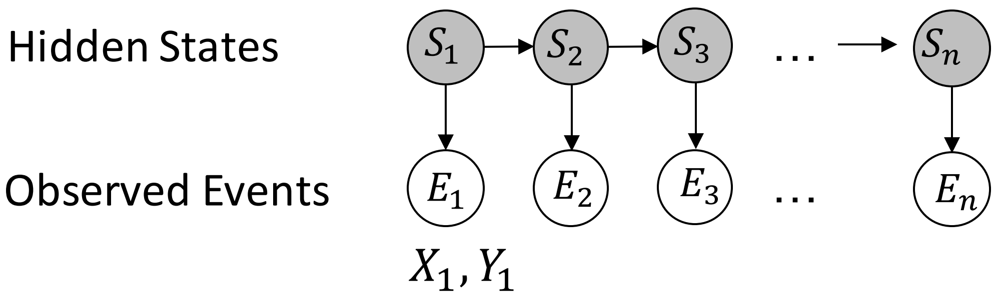
For example, $f(a^2, b^1, c^1) = f_1(a^2, b^1) f_2(b^1, c^1)$

Conversion of a Hidden Markov Model to a Factor Graph– Two dimension

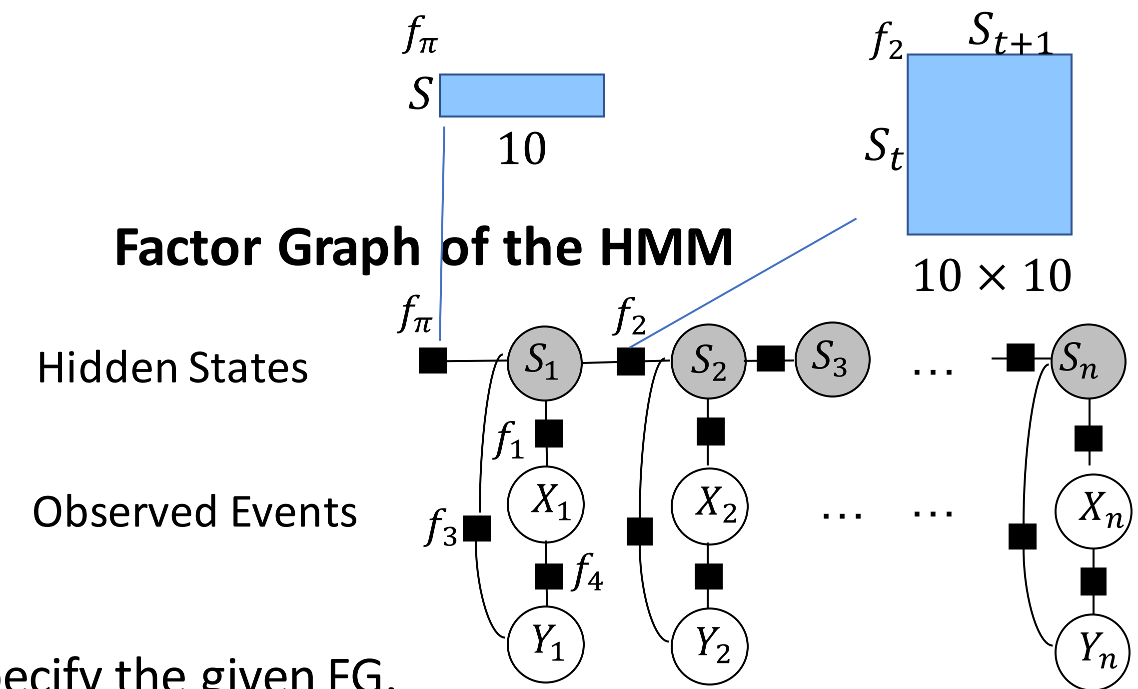
Assume that at each time point, two observations are made corresponding to random variables X and Y.

Example: Let $|\boldsymbol{S}| = \mathbf{10}, |\boldsymbol{X}| = \mathbf{10}, |\boldsymbol{Y}| = \mathbf{10}$

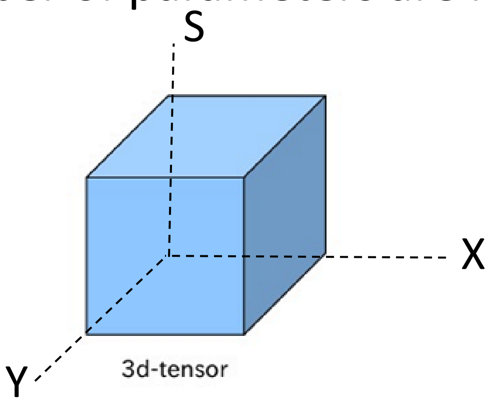
Hidden Markov Model



Factor Graph of the HMM

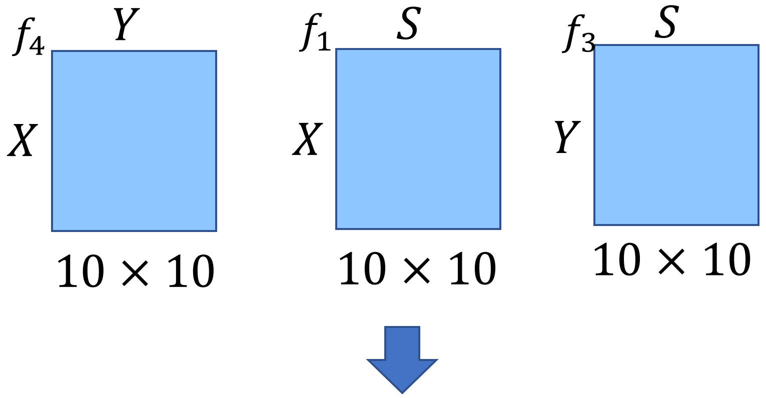


Fewer number of parameters are required are required to specify the given FG.



size of tensor is exponential

$10 \times 10 \times 10 = \mathbf{1000}$



size of five matrices

$10 + 10 \times 10 + 10 \times 10 + 10 \times 10 + 10 \times 10 = \mathbf{410}$

Modeling the credential stealing attack using Factor Graphs - Data

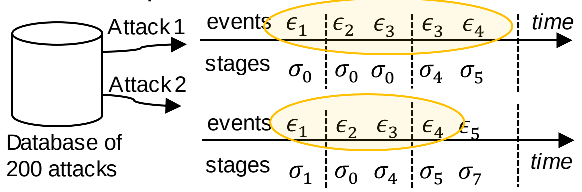
State space of variables

Attack stage: $X = \{\sigma_0, \sigma_1, \dots, \sigma_7\}$

(Observed) Events: $E = \{\epsilon_1, \dots, \epsilon_5\}$

OFFLINE ANNOTATION ON PAST ATTACKS

a) Annotated events and attack stages in a pair of attacks



b) Event-stage annotation table for the attack pair (Attack 1 and Attack 2)

Event	Attack stage
$\{\epsilon_1\}$	$\{\sigma_0 \vert \sigma_1\}$
$\{\epsilon_2\}$	$\{\sigma_0\}$
$\{\epsilon_3\}$	$\{\sigma_4\}$
$\{\epsilon_4\}$	$\{\sigma_5\}$
$\{\epsilon_5\}$	$\{\sigma_7\}$

ϵ_1	vulnerability scan	σ_0	benign
ϵ_2	login	σ_1	discovery
ϵ_3	sensitive_uri	σ_4	privilege escalation
ϵ_4	new_library	σ_5	persistence

- **Attack Information**
  - Multi-stage credential stealing attack
  - Attack stage $\sigma \in X$ is not observed; however an attack happens in a chain of exploits, thus we have a sequence of events
  - Each security event is a known variable ϵ, each takes value from a discrete set of events E
- **Problem statement.** Given a set of security events, infer whether an attack is in progress?
  - Goal is to detect and pre-empt the attack
- **Model assumptions**
  - There are multivariate relationships among the events
  - There is no restriction on order of the relationships (can be non-causal or correlation based)
- Markov Model and Bayesian Networks cannot be used in this scenarios
- Factor graphs can be used for modeling highly complex attacks, where the causal relations among the events are not immediately clear.

Modeling the credential stealing attack using Factor Graphs

OFFLINE LEARNING OF FACTOR FUNCTIONS

Example patterns, stages, probabilities, and significance learned from the attack pair

Pattern	Attack stages	Probability in past attacks	Significance (p-value)
$[\epsilon_1, \epsilon_3, \epsilon_4]$	$[\sigma_1, \sigma_4, \sigma_5]$	q_a	p_a
$[\epsilon_1]$	$[\sigma_0 \| \sigma_1]$	q_b	p_b

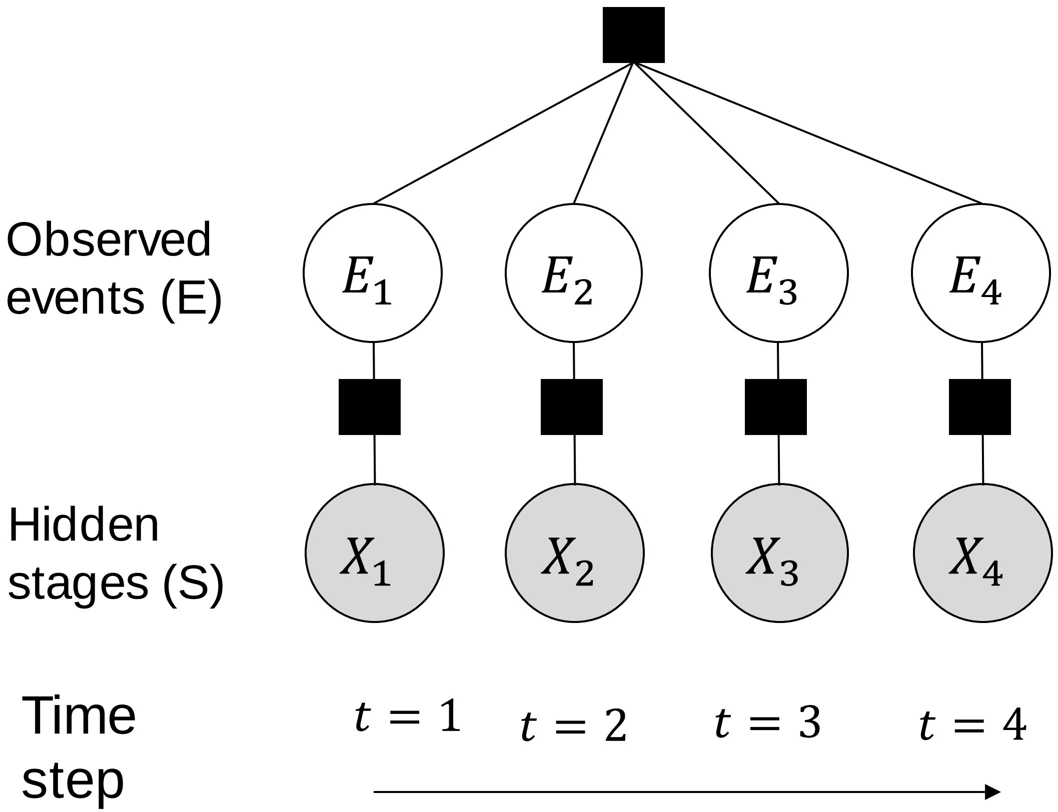
...

$$f(E) = \exp\{q_E(1 - p_E)\}$$

A factor function defined on the learned pattern, stages, and its significance

DETECTION OF UNSEEN ATTACKS

Factor Graph



Advantages and Disadvantages of Factor Graph

Advantage

- Factor graph subsumes HMMs, Markov Random Fields, Bayesian Networks etc.

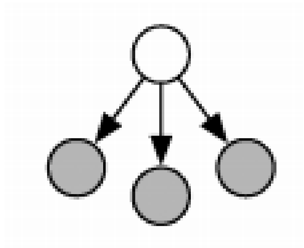
Disadvantage

- Limitations of probabilistic graphs in general

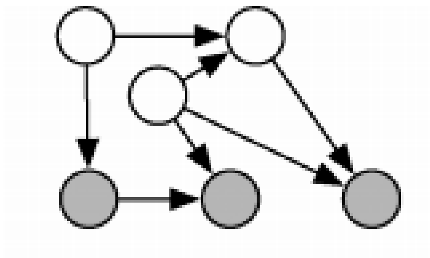
Comment

- If the problem is well represented by specific models such as Bayesian Networks, HMMs, Naïve Bayes or other graphical models then there is no need to generalize your problem as a factor graphs

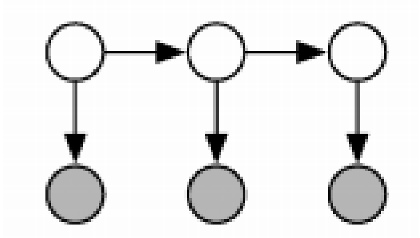
Taxonomy of graphical models



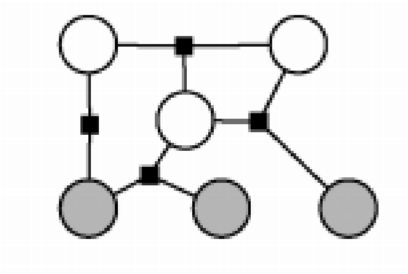
Naïve Bayes



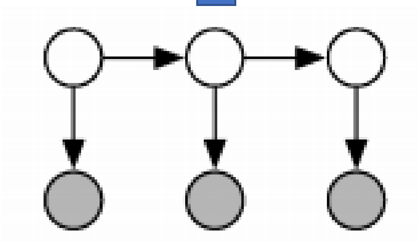
Bayesian Network



Dynamic Bayesian Network



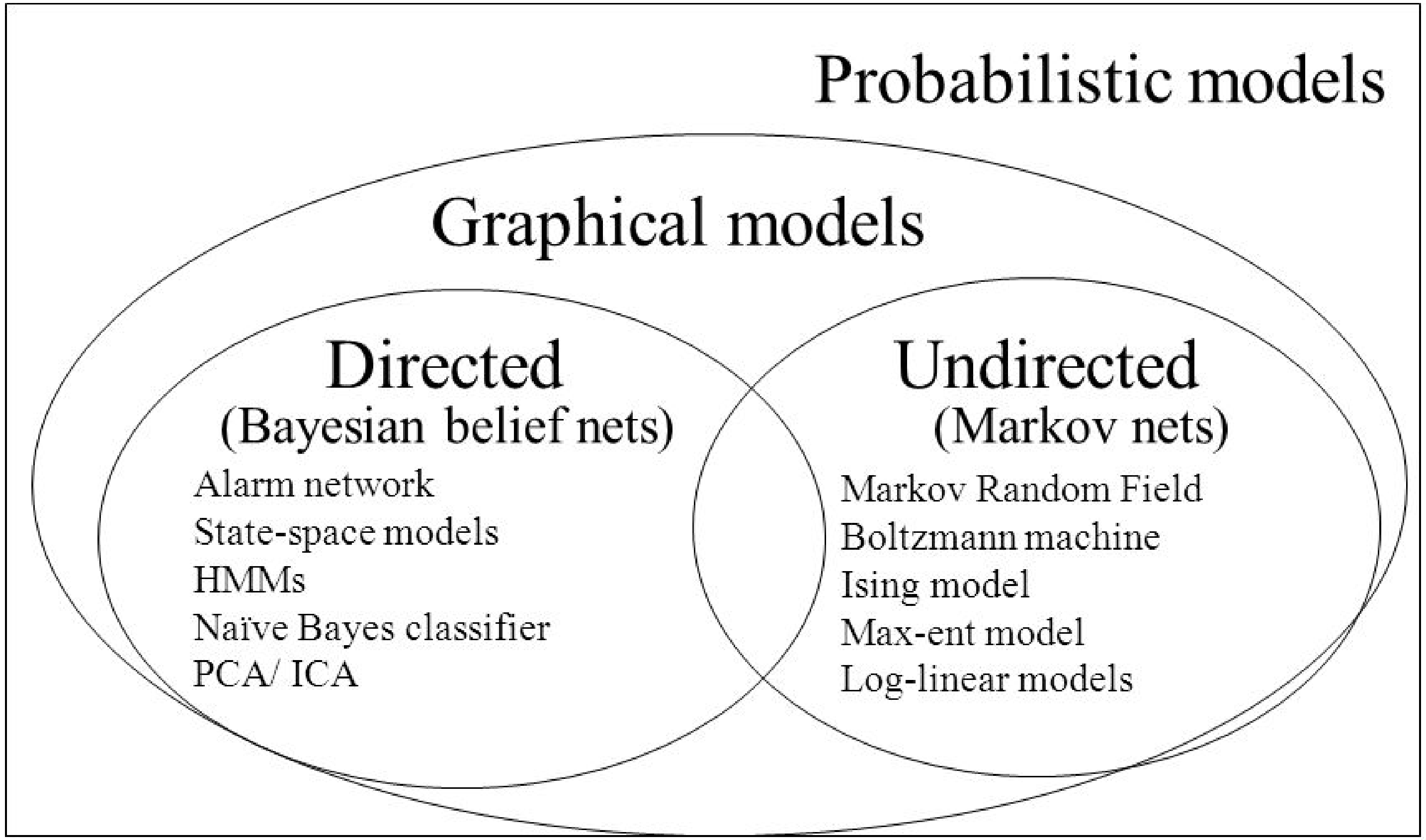
Factor Graph



Hidden Markov Model

Conditional probabilities and statistical dependencies can be represented by a general type of graph: Factor Graph

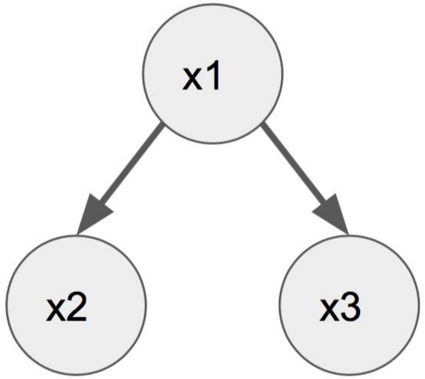
Taxonomy of Graphical Models



Machine Learning, A Probabilistic Perspective, Kevin Murphy, MIT Press

Bayesian Networks vs. Hidden Markov Models vs. Factor Graphs

Bayesian Network

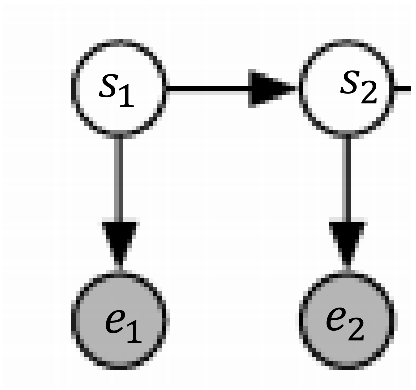


$$p(x_1)p(x_2|x_1)p(x_3|x_1)$$

Product of conditional probabilities

Causal relationships

Hidden Markov Model

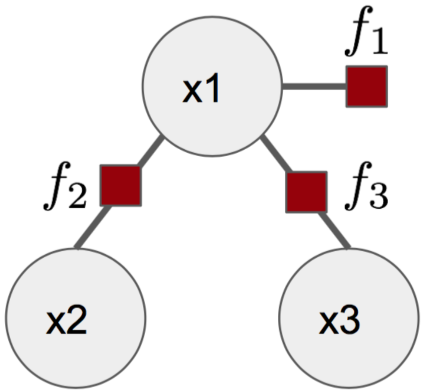


$$p(s_1)p(e_1|s_1)p(s_2|s_1)p(e_2|s_2)$$

Product of Temporal dependencies among variable

Temporal and statistical dependencies

Factor Graph



$$\frac{1}{Z}f_1(x_1)f_2(x_2,x_1)f_3(x_1,x_3)$$

Product of dependencies using univariate, bivariate, or multivariate functions

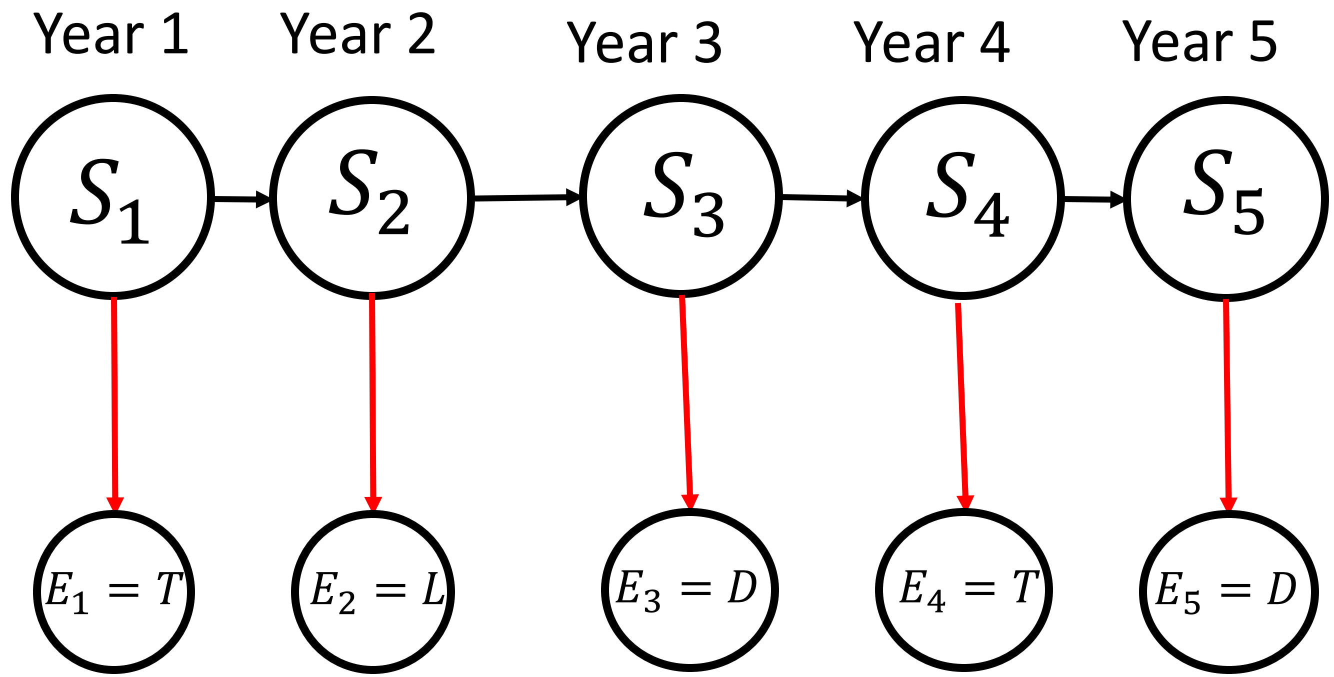
Both types of relations (including prior on a variable)

Practice with Factor Graph

HMM Example - Paleontological Temperature Model

- State space of hidden states: $S = \{H, C\}$
- State space of observations: $E = \{T, D, L\}$
- Transition probability matrix: A
- Observation Matrix: B
- Initial distribution for the hidden states: π

Given by an oracle



$$\begin{array}{c c} & \begin{array}{cc} \mathrm{H} & \mathrm{C} \end{array} \\ \begin{array}{c} \mathrm{H} \\ \mathrm{C} \end{array} & \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

A

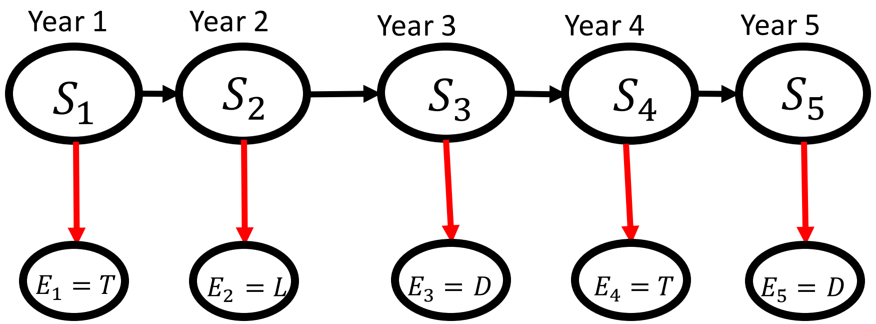
$$\begin{array}{c c} & \begin{array}{ccc} \mathrm{T} & \mathrm{D} & \mathrm{L} \end{array} \\ \begin{array}{c} \mathrm{H} \\ \mathrm{C} \end{array} & \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \end{array}$$

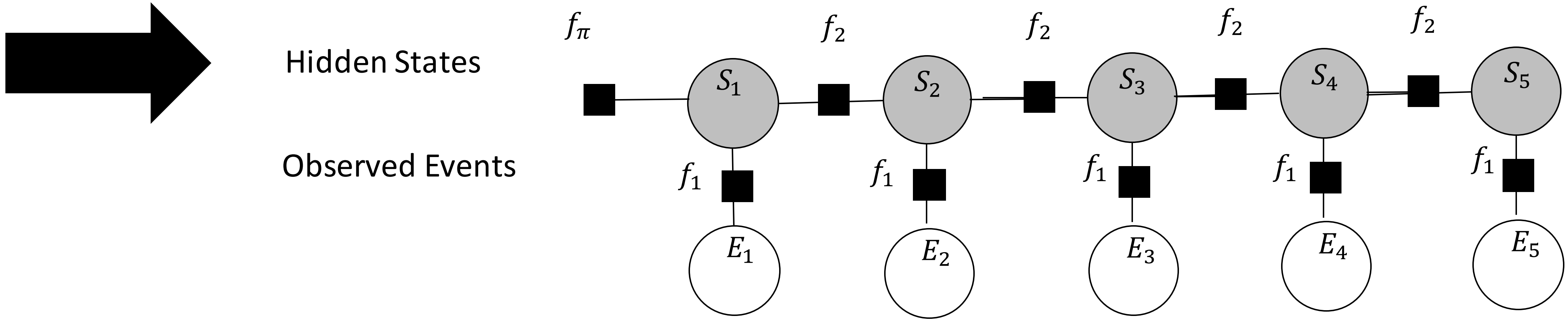
B

$$\begin{array}{cc} \mathrm{H} & \mathrm{C} \end{array}$$
$$[0.5 \; 0.5]$$

π

First Step – Drawing Factor Graph from HMM

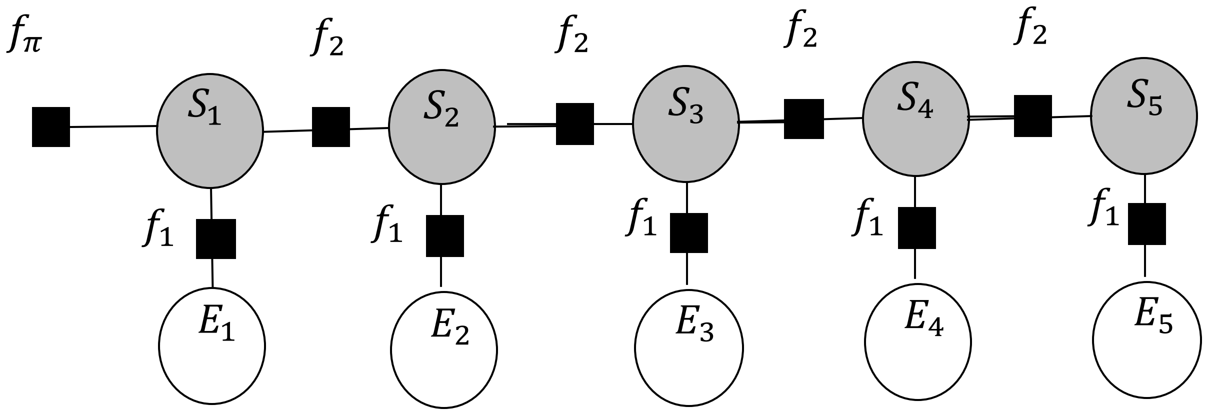




Why are the factor functions…

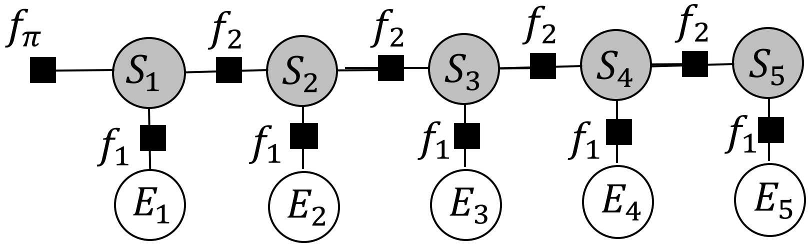
- **Between every pair of states the same?**
- **Between every pair of state and observation the same?**

Next – Figuring out the Factor Functions



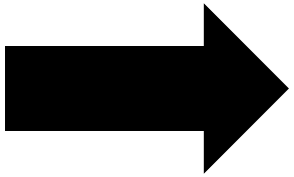
- f_π needs to capture the prior probabilities for the states
- f_1 needs to capture the affinity between observations and states
- f_2 needs to capture the affinity between consecutive states

Next – Figuring out the Factor Functions



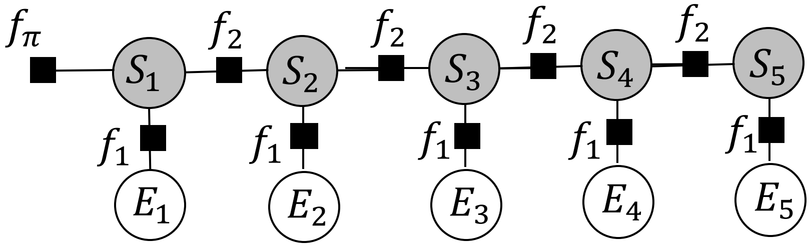
- f_π needs to capture the prior probabilities for the states

$$\begin{array}{cc} \text{H} & \text{C} \\ \end{array}$$
$$[0.5\ 0.5]$$
$$\pi$$



S_1	$f_\pi(S_1)$
H	0.5
C	0.5

Next – Figuring out the Factor Functions



- f_1 needs to capture the affinity between observations and states. (i.e., $P(E_i|S_i)$)

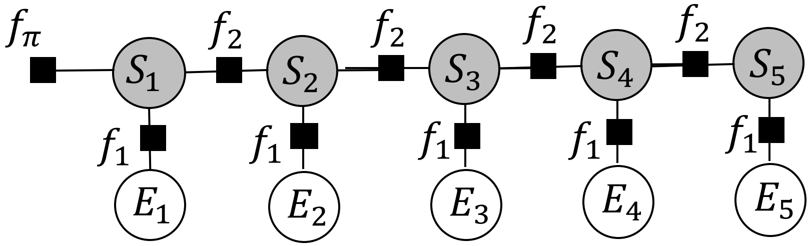
$$\begin{array}{c} \\ H \\ C \end{array} \begin{array}{ccc} T & D & L \\ \multicolumn{3}{c}{\begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{bmatrix}} \end{array}$$

B

S_i	E_i	$f_1(S_i, E_i)$
H	T	0.1
	D	0.4
	L	0.5
C	T	0.7
	D	0.2
	L	0.1

Note that the values in this table don't sum to 1
=> f_1 is not a joint probability but a conditional probability!

Next – Figuring out the Factor Functions



- f_2 needs to capture the affinity between consecutive states. (i.e., $P(S_{i+1}|S_i)$)

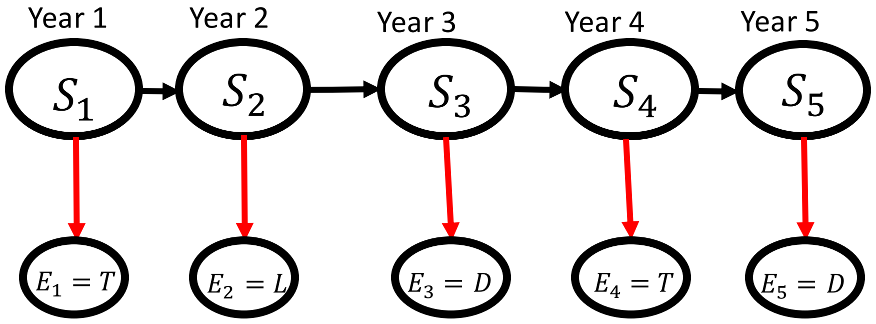
$$\begin{array}{c c} & \begin{array}{cc} \text{H} & \text{C} \end{array} \\ \begin{array}{c} \text{H} \\ \text{C} \end{array} & \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

A

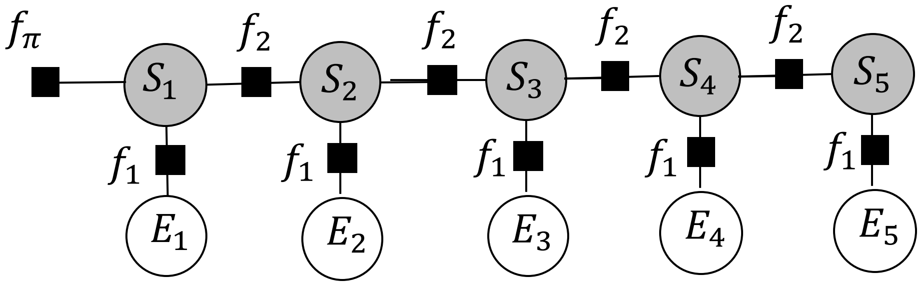
S_i	S_{i+1}	$f_2(S_i, S_{i+1})$
H	H	0.7
	C	0.3
C	H	0.4
	C	0.6

Note that the values in this table don't sum to 1 => f_2 not a joint probability but a conditional probability!

After That – Calculating the Joint



$$P(S_1, \dots, S_5, E_1, \dots, E_5) =$$
$$P(S_1)P(E_1|S_1) \prod_{i=2}^{5} P(S_i|S_{i-1})P(E_i|S_i)$$



$$P(S_1, \dots, S_5, E_1, \dots, E_5) =$$
$$\frac{1}{Z} f_\pi(S_1) f_1(S_1, E_1) \prod_{i=2}^{5} f_2(S_{i-1}, S_i) f_1(S_i, E_i)$$

$$Z = \sum_{S_i \in \{H,C\}, E_i \in \{T,D,L\}} f_\pi(S_1) f_1(S_1, E_1) \prod_{i=2}^{5} f_2(S_{i-1}, S_i) f_1(S_i, E_i)$$