Validation of IT Risk Assessment with Markov Logic Networks

Janno von Stülpnagel
and Willy Chen

June 14, 2015

Motivation

- Risk assessment is often a collaborative work of subjectiv evaluations
- This provides multiple challenges:
  - Domain experts with different backgrounds may have different understandings of risk
  - Connected infrastructure, evaluated by different persons produces incomplete knowledge
  - Fast changing threat and risk landscape
  - Few tools to validate or debug a risk assessments

Our Approach

Supporting risk assessment with result validation and debugging by:

- Formalizing the risk assessment results
- Linking information of the IT infrastructure (such as enterprise architecture or system documentations)
- Specify rules for validation (i.e. defining anomalies)
- Using Markov logic network inference to check the validity and providing a suggested correction

Agenda

- Background Risk Management and Semantic Web
- Markov Logic Networks
- Validation
- Scenario
- Discussion

Background Risk Management

- **Risk**: is a set of triplets, each triplet consisting of:
  - a scenario
  - probability of the scenario
  - impact of the scenario
- **Threat**: The cause of an scenario
- **IT Risk Management**: find, analyze and reduce unacceptable risks in the IT infrastructure
- **The Risk Assessment**:
  - The **Risk Identification** generates a list of possible threats
  - **Risk Analysis** determines the level of risk for threats by combining the likelihood and impact.
  - The **Risk Evaluation** decides if a risk is acceptable or is treatment priority

Background Semantic Web

- The idea of Semantic Web is that data should be independent of it presentation and related to one another.
- This would allow it to share and reuse data across applications and organization boarders
- A sound logical basis would make it possible to easily process the data.
- While the Semantic Web was initially thought of as extension of the World Wide Web, it is well-suited for integrating heterogeneous data in a single organization.

Background MLN

- Markov logic networks (MLN) is a combination of probability and first-order logic by adding weights to formulas.
- Together with a set of evidcense, it is possible to calculate probability distributions.

⇒ The undirected graph model of the MLN allows us to use the risk assessment of domain experts, in combination with the hard and soft validation formulas, to calculate a new corrected assessment.

Markov Logic Networks Inference

Marginal inference:

- Calculating the posteriori probability distribution over all variable assignments
- The probability of a variable assignment corresponds to the sum of the weighs of all the fulfilled soft formulas in all possible worlds

Definitions:

- Grounding: substitutes each occurrence of every variable in a formula with a constant
- Possible world: Assignment of truth values to all possible ground predicates

Markov Logic Networks as Template

Markov logic networks are templates for constructing Markov networks:

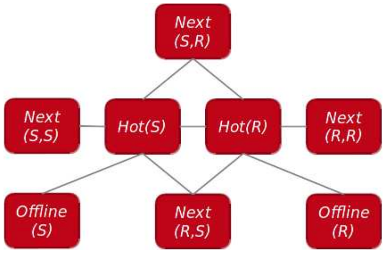
- Nodes: The grounded atoms
- Edge: There is an edge between two nodes iff the corresponding ground atoms appear together in at least one grounding of a formula

Definitions:

- Atom: A formula that consists of a single predicate
- Grounding: substitutes each occurrence of every variable in a formula with a constant

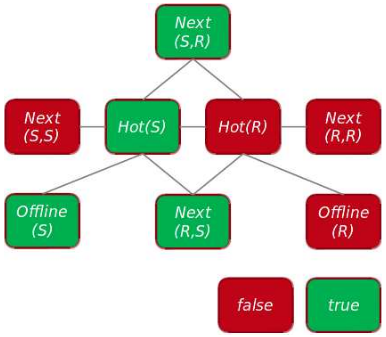
MLN Example - Part 1

- Formulas:
  - Hot devices do not work: $\forall$ x Hot(x) $\Rightarrow$ Offline(x)
  - If two devices are next to each other (in the same rag), either both are hot or neither is: $\forall x \forall y$ Next(x,y) $\Rightarrow$ (Hot(x) $\Leftrightarrow$ Hot(y)
- Constants:
  - Server (S)
  - Router (R)



MLN Example - Part 2

- A randomized algorithm to approximate the marginal distribution
- Each iteration, sample a variable by swapping it according to its probability given only by its neighbors:
    - P(x | neighbors(x))
    - We do not swap it if it violates any hard formulas
- The estimated probability of a ground atom is the ratio of samples in which it is true and total number of samples



Validation with MLN

- We define anomalies in a set of risk assessments as dissenting assertions of threats for an IT component considering its various dependencies (technical, logical, geographical ...).
- For example two servers within the same server rack have different assessments for a given physical threat.

Validation Rules for the Scenario

- The threat "Fire" makes the threat "Water damage", through the extinguishing, more probable
- "Unauthorized Access to IT Systems" will often result in "Loss of stored data"

1. `hasProbability(infra1, thread1, prob1)` $\wedge$
   `hasLocalInfluence(thread1, thread2, prob2)` $\wedge$
   `inLocation(infra1, loc)` $\wedge$ `inLocation(infra2, loc)` $\Rightarrow$
   `hasProbability(infra2, thread2, prob2)`
2. `hasProbability(infra1, thread1, prob1)` $\wedge$
   `hasNetworkInfluence (thread1, thread2, prob2)` $\wedge$
   `inNetwork (infra1, net)` $\wedge$ `inNetwork (infra2, net)` $\Rightarrow$
   `hasProbability(infra2, thread2, prob2)`
3. `hasInputProbability(infra, thread, prob)` $\Rightarrow$
   `hasProbability(infra, thread, prob)`

Scenario

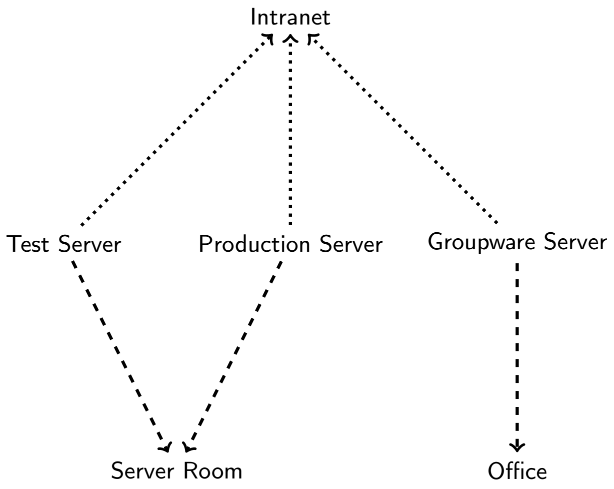


Figure: The IT infrastructure for our case study. Dotted lines represent a connection to a network and dashed lines represent that a component stands in a location.

Predicates

The formulas have 5 types of variables:

- `infra` as the type of all infrastructure components
- `threat` as the type of all threats
- `net` as the type of all networks
- `loc` as the type of all locations
- `prob` as the type of all qualitative probability

and 5 predicates:

- texttthasProbability(infra1, thread1, prob1) states that `infra1` is endangers by `threat1` with the probability of `prob1`.
- `inLocation(infra1, loc)` expressing that `infra1` are stands in the location `loc`
- `inNetwork (infra1, net)` respectively that it is connected with the network `net`.
- `hasLocalInfluence(thread1, thread2, prob1)` and `hasNetworkInfluence (thread1, thread2, prob1)` are two predicates for expressing the local and network influence of a threat.

Evidence Part 1: The IT infrastructure as MLN evidence

1. `inLocation("Test Server", "Server Room")`
2. `inLocation("Production Server", "Server Room")`
3. `inLocation("Groupware Server", "Office")`
4. `inNetwork("Test Server", "Intranet")`
5. `inNetwork("Production Server", "Intranet")`
6. `inNetwork("Groupware Server", "Intranet")`

Evidence Part 2: The influence of the threats on each other

1. `hasLocalInfluence("Fire", "Fire", "Possible")`
2. `hasLocalInfluence("Fire", "Water", "Improbable")`
3. `hasNetworkInfluence("Unauthorized Access to IT Systems", "Unauthorized Access to IT Systems", "Probable")`

Evidence Part 3: The result of the risk assessment, which we want to check for invalidity

1. `hasInputProbability("Test Server", "Fire", "Possible")`
2. `hasInputProbability("Test Server", "Unauthorized Access to IT Systems", "Very Probable")`
3. `hasInputProbability("Production Server", "Fire", "Improbable")`
4. `hasInputProbability("Production Server", "Water", "Probable")`
5. `hasInputProbability("Production Server", "Unauthorized Access to IT Systems", "Improbable")`
6. `hasInputProbability("Groupware Server", "Fire", "Improbable")`
7. `hasInputProbability("Groupware Server", "Unauthorized Access to IT Systems", "Possible")`

The Example of the result of our calculation for the infrastructure Test Server and the threat Fire

Component	Threat	Occurrence Probability	Marginal Inference Result
Test Server	Fire	Possible	0.999730
Test Server	Fire	Very Probable	0.492144
Test Server	Fire	Probable	0.491708
Test Server	Fire	Improbable	0.490258

The results of our calculation

Component	Threat	Input Probability	Highest Marginal Inference Probability
Test Server	Unauthorized Access	Very Probable	Probable
Test Server	Fire	Possible	Possible
Test Server	Water	-	Improbable
Production Server	Unauthorized Access	Improbable	Probable
Production Server	Fire	Improbable	Improbable
Production Server	Water	Probable	Probable
Groupware Server	Unauthorized Access	Possible	Probable
Groupware Server	Fire	Improbable	Improbable
Groupware Server	Water	-	Improbable

Only the most probable threat occurrence probability is given for each infrastructure threat combination

Discussion

- The validity of a risk assessment result can be checked over some given hard and soft rules.
- These soft rules allow us to capture domain experts knowledge which cannot easily be translated into hard formulas and would be otherwise be lost.
- Our approach doesn't only find the existence of possible anomalies, but also recommends correction.
- The undirected MLN model also allows reciprocal and circular influence from components and threats, which exists in real world scenarios.

Discussion

- The relationship between weights and the probabilities of the marginal inference result can be counterintuitive.
- While the OWL2-QL profile has a very good scalability, we need to test the MLN inference with bigger data sets.
- To our knowledge no MLN solver supports full first-order logic and because of the undecidability of first-order logic there will most likely never be one.

Questions

?