Applying combinatorial group testing to trust evaluation in a distributed computing model

Chris Mitchell
www.chrismitchell.net
me@chrismitchell.net

Acknowledgements

- This talk is based on research conducted by my (ex) PhD student Georgios Kalogridis.
- George recently completed a part-time PhD whilst employed at Toshiba Research Labs in Bristol.

Agenda

1. Agent systems and spy agents
2. The spy agent routing problem
3. A simple approach
4. Problems
5. Building in resilience
6. Multi-stage testing
7. Concluding remarks

Agenda

1. Agent systems and spy agents
2. The spy agent routing problem
3. A simple approach
4. Problems
5. Building in resilience
6. Multi-stage testing
7. Concluding remarks

Mobile agent systems

- A **mobile agent** is an aggregation of software and data able to:
  - migrate (move) from one computer to another autonomously;
  - continue execution on destination computer.
- Motivation is to reduce need to communicate – autonomous agent could visit many sites before returning to originator with results.

Context

- Just one example of a distributed computing model.
- Of course, each host machine must be able to receive, execute, and forward mobile agents.
- Model has attracted considerable attention from researchers over last 10-15 years.
- Interesting problems relating to game theory, artificial intelligence, etc.

Example application

- One widely discussed possible application relates to e-commerce.
- A ‘shopping agent’ could be programmed with user requirements and then sent out to find the best deal on offer.
- It might:
  - return to originator and provide summary of deals on offer;
  - actually conclude the best deal autonomously, and then simply return details of deal to originator.

Trust issues

- Two major security/trust issues associated with mobile agents.
  1. **Malicious agents**: a malicious agent might try to subvert a visited host and/or learn about other agents.
  2. **Malicious hosts**: a malicious host might seek to learn originator secrets from agent code, or simply unfairly influence outcome of agent computations (e.g. by changing competitor offers in e-commerce example).

Malicious agents

- This threat arises in any mobile code scenario.
- Many possible solutions, including:
  - sandboxing (as in Java);
  - proof carrying code (code carries proof of its properties which can be verified before execution);
  - code signing;
  - ...

Malicious hosts

- As many authors have observed, fixing this problem is very difficult.
- Host has complete control over code.
- Possible solutions include:
  - code obfuscation;
  - homomorphic encryption (allowing computing on encrypted data);
  - use of trusted computing to provide guarantees over host behaviour.
- These measures are designed to prevent bad things occurring ...

Remote host assessment

- In practice, it is often impossible to completely prevent bad outcomes.
- One approach is to try to minimise risk by using ‘more trusted’ hosts.
- Idea underlying this talk is a possible method for remote host trust evaluation.
- Results from this evaluation could be used by a reputation management system.

Spy agents

- Idea of spy agents is to send out agents which look genuine but which are purely present to test hosts.
- Originator tests a set of hosts by sending out a number of spy agents and awaiting results.
- Spy agents must contain information which can be misused (incentive to misbehave).
- Misbehaviour must be detectable by originator.

Analysing results

- Assumption is that agent mishandling will be detected; not who did it, but which agents have been abused.
- That is, after sending out agents, each to a predetermined set of hosts, the originator will (eventually) receive a positive or negative indication for each agent, i.e. of whether or not it has been abused.
- Need to analyse these results to identify bad hosts.

E.g. – decoy email addresses

- Could equip each agent with a decoy email address which looks genuine (and has high entropy).
- Agent policy could require non-dissemination of email address.
- If email address receives spam, then this is evidence of agent abuse.

Agenda

1. Agent systems and spy agents
2. <u>The spy agent routing problem</u>
3. A simple approach
4. Problems
5. Building in resilience
6. Multi-stage testing
7. Concluding remarks

Assumptions

- All hosts are bad or good.
- An approach to agent design and use has been chosen that guarantees:
  - if an agent **route** (i.e. set of hosts it visits) includes at least one bad host then it will yield a positive result;
  - if an agent route include no had hosts then it will yield a negative result.
- The order in which an agent visits hosts is immaterial.
- Malicious hosts do not collude.

Discussion

- These are very strong assumptions.
- Consider later in talk how they might be weakened.
- We are concerned with choosing a set of agent routes so that the malicious hosts can be uniquely identified, no matter how they are distributed.
- Clearly a combinatorial problem ...

Constraints

- Wish to minimise number of spy agents and also number of agents received by each host.
- However, also assume that agents with large route sets are better, as malicious hosts are more likely to misbehave.
- Sending a unique agent to each host is not acceptable (unacceptable risk of detection to host, and perhaps no incentive to misbehave).

Agenda

1. Agent systems and spy agents
2. The spy agent routing problem
3. A simple approach
4. Problems
5. Building in resilience
6. Multi-stage testing
7. Concluding remarks

Group testing

- The underlying combinatorial problem has been well-studied under many guises.
- In **group testing** a population of items containing a small set of defectives is tested in order to identify the defectives.
- Items are pooled for testing; a group test reports ‘positive’ if the tested pool contains one or more defective elements, and reports ‘negative’ otherwise.

Sequential & non-adaptive testing

- Two main types of group testing (GT) schemes: sequential and non-adaptive.
  - **Sequential** schemes allow the selection of later tests to be based upon the outcomes of previous tests. (Fewer tests in general).
  - In a **non-adaptive** scheme, the set of tests is predetermined. (Allows parallelism).
- Sequential GT goes back almost 70 years (Dorfman ,1943).

Non-adaptive GT

- Range of non-adaptive GT constructions have been proposed based on block designs, superimposed codes, transversal designs, cover-free families, and other combinatorial designs.
- Recent survey of non-adaptive GT provided by Du and Hwang (2006).

Application to spy agents

- In most cases non-adaptive approach likely to be more fruitful, because:
  - possibility of parallelism (sending out multiple agents at same time);
  - need results in shortest possible time.
- Look at a simple example.
- Note need for decoding algorithm (given agent results, need a means to determine bad host).

Formalisation

- **Route design** is a triple (R, S, I), where:
  - R is set of agents,
  - S is a set of n hosts, and
  - I is an incidence relation between R and S, corresponding to an agent visiting a host).
- Identify R with points and S with blocks of a block design (rows and columns of an incidence matrix).

Notational abuse

- Will often think of as a row or column of an incidence matrix as a set, and will refer to ‘membership’ of a row or column.
- Will also take this further and refer to the **union** of columns or rows, with the ‘obvious’ meaning.
- My excuse? Well, I’m just a Computer Scientist, so I don’t know any better ...

Classifiers

- Call a route design a *d*-classifier if, given **exactly** *d* defective hosts, the outcome of the design can be used to identify all the defective (and honest) hosts.
- A route design is a *d*-classifier if, given **at most** *d* defective hosts, the outcome of the design can be used to identify all the defective (and honest) hosts.

Separable matrices

- Incidence matrix is *d*-separable if 'unions' of subsets of **exactly** *d* columns are all distinct.
- Incidence matrix is *d̲*-separable if 'unions' of subsets of **at most** *d* columns are all distinct.
- Route design is *d*/*d̲*-classifier if and only if incidence matrix is *d*/*d̲*-separable (Kautz and Singleton, 1964).

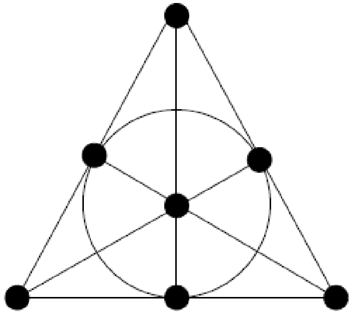
Decoding problem

- This solves the problem ...
- However, there is no efficient general decoding algorithm for separable schemes.
- (Decoding algorithm takes as input the outcome vector for all agents, and outputs the set of defective hosts).
- Hence look for restricted class, as follows.

Disjunct matrices

- Incidence matrix is *d*-disjunct if 'union' of any subset of **exactly** *d* columns does not contain any other column (as a 'subset').
- If matrix is *d*-disjunct, then it is:
  - *d′*-disjunct for all $d' \leq d$;
  - *d*-separable.
- Moreover, there is a simple decoding algorithm:
  - union of all negative rows (agents with negative outcomes) = set of all non-defective hosts.

Example



- Fano plane - a 2-(7,3,1) design.
  - 7 lines, 7 points;
  - 3 lines/point, 3 points/line;
  - 2 points on 1 line,
  - 2 lines intersect in 1 point

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

- Incidence matrix:
  - the incidence matrix for Fano plane is 2-disjunct but not 3-disjunct.

Example (continued)

- Decoding:
  - say the set of defectives is {2,5};
  - outcome vector is {1,1,0,1,1,0,1};
  - only columns that do not appear in negative routes are 2 and 5.

Agenda

1. Agent systems and spy agents
2. The spy agent routing problem
3. A simple approach
4. <u>Problems</u>
5. Building in resilience
6. Multi-stage testing
7. Concluding remarks

Overly strong assumptions

- The assumptions we made are clearly very strong.
- Malicious hosts may not always misbehave.
- We need to develop techniques which work even when malicious hosts only selectively misbehave.

Reformulations

- We could assume that malicious hosts will behave in a more random manner.
- Group Testing techniques exist which can cope with errors, and such techniques might be appropriate in such an environment.
- We next consider a slightly different model, in which malicious hosts behave in ways to try to conceal their behaviour.

Agenda

1. Agent systems and spy agents
2. The spy agent routing problem
3. A simple approach
4. Problems
5. <u>Building in resilience</u>
6. Multi-stage testing
7. Concluding remarks

A selective misbehaviour model

- We suppose a malicious host will only mishandle a visiting agent if it is scheduled to visit at least e-1 other malicious hosts, for some e>0;
  - such a malicious host is said to be of **type** e.
- We further suppose that type e (e>1) malicious hosts are aware of other hosts which are malicious.
- Clearly a type 1 malicious host will always mishandle a visiting agent.

Complex defectives

- To design sets of routes capable of dealing with scenarios where malicious hosts may be of varying types, we use the theory of **group testing for complexes** (GTC).
- GTC deals with identifying sets of objects that collectively (and minimally) yield a positive result.
- Such sets we call **defective complexes**.

Defective complexes – examples

- If there is a single malicious host of type 2, then set of defective complexes is empty.
- If there are two defective hosts of type 2, then the single defective complex will contain them both.
- If there are only two defective hosts, one of type 2 and one of type e (>2), then the single defective complex will contain them both.

Definition

- Given a set of hosts, a set of defective complexes is a collection *D* of subsets of hosts satisfying:
  1. an agent will give a positive result if and only if it contains a member of *D*;
  2. the previous property does not hold for a proper subset of *D*.
- Can show that the set of defective complexes is unique.
- Need to identify it ...

Separable matrices revisited

- The **rank** of a set of complex defectives is the size of the largest element.
- An incidence matrix is (d,e)-separable if, when applied to distinct sets of defective complexes of size at most d and rank e, distinct outcomes result.

Disjunct matrices revisited

- A route design is (d,e)-disjunct if, give any set of d+1 mutually non-inclusive complexes (sets of hosts) of rank e, then:
  - the set of agent routes containing all the hosts in the first complex, contains at least one agent route not containing any of the other complexes.
- (d,e)-disjunct implies (d,e)-separable (Du and Hwang, 2006).

Example

- The following (simple) route design is (2,2)-disjunct:

	c_1	c_2	c_3	c_4
r_1	1	1	0	0
r_2	1	0	1	0
r_3	1	0	0	1
r_4	0	1	1	0
r_5	0	1	0	1
r_6	0	0	1	1

Decoding

- Suppose a (d,e)-disjunct route design is applied to a set D of defective complexes with cardinality at most d and rank e.
- Determine D as follows:
  - Let E = set of all f-subsets of hosts, $f \leq e$.
  - Let G be elements of E (i.e. f-subsets of hosts) with property that every agent containing every host in the subset gives a positive result.
  - D = ‘minimal’ elements of G (i.e. those not containing another element of G as a subset).

Finding (*d*,*e*)-disjunct matrices

- A (*d*,*e*)-disjunct route design is equivalent to (see Chen, Du & Hwang, 2007):
  - a (*d*,*e*)-superimposed code;
  - a (*d*,*e*)-cover-free family, and
  - a (*d*,*e*)-key distribution pattern.
- Can construct such objects in many ways, e.g. using *t*-designs.

Identifying individual hosts

- So far we have considered identification of set of defective complexes.
- This must then be analysed to try to determine set of defective hosts.
- In general this will not be possible.
- For example, if there are *d* malicious hosts all with types greater than *d*, then no malicious host will ever misbehave.
- However, some special cases can be addressed.

Agenda

1. Agent systems and spy agents
2. The spy agent routing problem
3. A simple approach
4. Problems
5. Building in resilience
6. Multi-stage testing
7. Concluding remarks

Adaptive group testing

- So far we have considered non-adaptive group testing.
- However, may be cases where adaptive (SGT) approach is more efficient.
- Unfortunately, ‘standard’ adaptive techniques don’t really work in our setting,
- This is because they typically involve doing tests for very small sets of hosts.

Definition – weak

- We wish to design schemes which never send agents to a very small set of hosts.
- Note that an adaptive scheme does not contain a single set of routes – the route set will vary depending on the results of earlier tests.
- In any scheme, a **weak** route is one with the smallest possible number of hosts for that scheme.

Optimality

- Suppose a SGT scheme *A* is capable of identifying all malicious hosts, regardless of their number.
- Let r_A be the length of a weak route in *A*.
- *A* is said to be **route-length-optimal** if, for any other scheme *B* which can identify all malicious hosts, $r_A \geq r_B$.

A result

- Suppose a set of n hosts is known to contain at most d malicious hosts.
- Then the length r of a weak route in an sequential scheme capable of detecting all malicious hosts satisfies $r \leq n-d$.

Meeting the bound

- A simple construction shows that the bound is tight.
- Essentially, conduct a series of rounds, and in round $i \geq 0$ send agents to every subset of $n-i$ hosts.
- Unfortunately, the route-length-optimal schemes involve sending large numbers of agents.
- Some sort of compromise is required ...

Agenda

1. Agent systems and spy agents
2. The spy agent routing problem
3. A simple approach
4. Problems
5. Building in resilience
6. Multi-stage testing
7. Concluding remarks

Further work

- I have presented a combinatorial problem arising from a possible security mechanism for mobile agent systems.
- The solutions presented are all based on rather restrictive models of malicious host behaviour.
- Clearly, there is ample scope to develop schemes which correspond to less restricted models of behaviour.

Further information

- Much more information about this work is available in George's PhD thesis, which is available online:
  - G. Kalogridis, *Preemptive mobile code protection using spy agents*, Mathematics Department Technical Report **2012-04** (http://www.ma.rhul.ac.uk/static/techrep/2012/MA-2012-04.pdf).

Questions

- Questions ...

- Contact details:
  - Chris Mitchell: me@chrismitchell.net
  - Georgios Kalogridis: george@toshiba-trel.com