

Towards Learning Risk Estimation Functions for Access Control*

Luke Dickens and Alessandra Russo, Pau-Chen Cheng and Jorge Lobo,
Imperial College, IBM Watson Research Center,
{lwd03,ar3}@doc.ic.ac.uk {pau,jlobo}@us.ibm.com

Thursday 11th February, 2010

Summary The security of access and information flow carries with it the risk that resources will be misused – intentionally or accidentally. Static Access Control (AC) policies based on qualitative judgements are insufficient for scenarios where roles and access requirements are subtle and change frequently. We propose using quantified risk and benefit estimates in the design and management of AC policies. The first step is to build models for estimating risk and benefit. Many factors may affect risk and benefit, and the relationship among them and their impacts are usually complex and hard to determine analytically. Therefore we decided to use machine learning techniques to learn the models from AC data. Due to the sensitive and evolving nature of real AC data, we must generate synthetic data to begin demonstrating the efficacy of our approach. Given that no sets of AC data could cover all possibilities, we must also show that the models learned can be applied to data outside the training sets.

We start examining these problems by first creating a parametrised simulation model, designed to capture certain properties related to the risk and benefit of information transfer, including how risk aggregates over time. We then explore how different choices of the model parameters affect our ability to predict this risk, and describe some preliminary results of transferring learning between different instantiations of the model.

Motivation Access control has traditionally focused on static sets of roles, and this implies a fixed mapping from users to permissions. The JASON report [7] indicated that this approach is too rigid for modern organisations, and leads to a particularly risk averse approach to resource sharing, while improved yet secure resource sharing is a balance between risk and benefit. Current AC models and policies, such as MLS/Bell-Lapadulla, RBAC, ACL, etc, provide certain structures that allow some control over how resources are managed, but with no quantitative measures by which they can be evaluated [2]. These approaches fix accesses and do not adapt to changing circumstances. Other more recent approaches [2, 9] have tried to address this by showing how assessments of trust or risk might be incorporated into AC policies, allowing them to represent more justifiable behaviour, but these assessments and how they contribute to decisions are explicitly coded into the system. [5] and [6] show how policies might be inferred automatically by mining datasets of contextualised AC decisions and AC settings. These results depend on two factors, that good policies (those that we would like to replicate) are used in practice, and that detailed logs of these decisions and associated context are available for training.

Here, instead of mining decisions made to infer policy, we wish to predict the likelihood of subsequent damage or benefit associated with AC decisions. If we can predict the risk of damage and future benefit associated with some AC decision in some context, then we can either develop new AC models and base access decisions on those predictions, or provide a human decision maker with the tools to make an informed choice.

Models The problem is how to verify our hypothesis; we cannot know the future and are unlikely to be given access to current and relevant data. Instead we are creating an experimental environment in which we run simulations which are intended to be abstract representations of real situations. The intention being that we learn on these abstractions, and somehow map this learning into a form that can be used to support or make decisions in the related real world scenario. We have started using a relatively simple simulation environment and picked a learning methodology to predict risk. Our aim is to expand the simulation environment, experiment with different learning methodologies and transfer results from simpler to more complex scenarios.

A simulation is not perfect to be suitably rich to capture many of the details that can influence a real life situation, there are too many factors, and we can't be sure that these models are sufficiently accurate and suitably representative of the situation they abstract. However, humans have shown themselves to be very good at abstractly representing real life scenarios. Trainees are regularly tested in abstracted and simplified games before allowing them to practice in the real world – for example, surgeons are trained on software [3], as are pilots [1], soldiers regularly play war-games and even may use abstracted models for mission planning [4]. Our natural intuition when building such abstractions is that we are discarding unessessary information, while retaining important features of the underlying structure of the problem.

*This research is continuing through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and U.K. Ministry of Defence.

Simulation In our scenario, an air vehicle has crashed in a region controlled by a hostile force, and we must send a rescue group to search for survivors while avoiding (where possible) scouting parties. We abstract this to a grid world with discrete time-steps. At each point in time, our rescue agent is located somewhere on the grid, and has certain knowledge about the potential crash sites and the distribution of enemy scouts. There is also a centralised control agent, that has a richer, more accurate understanding of these features, and the control agent makes a decision at each time-step whether to communicate its richer view of the world with the rescue agent.

There are two kinds of risk associated with transfer of this information. If the information is not transferred, then the rescue group may blunder into enemy scouts or fail to find the crash site in time, whereas if they had the information this would not have happened. However, the sending of information carries with it the risk of interception, which causes the enemy scouts to raise their alertness level, increasing the likelihood that they will find the crash site at each time-step.

In the simulation, the base state consists of a grid of locations and a position for the rescue agent, each location has a probability of containing the crash site and a probability that scout is currently occupying it. The rescue agent’s state knowledge is a noisy representation of the base state, while the control agent *observes* the rescue agent’s view and its own more accurate view. Its own view is updated by satellite imaging which has the effect of allowing it to *see* the contents of a small proportion of the base state at every time-step. If the control agent sends this information at time t , then the rescue agent gets this improved view at time $t + 1$.

We can simulate multiple runs of this environment and determine for each time-step whether the observation and action choice (input) for the control agent led to success, failure or partial failure of the mission, each with an associated value of damage (output). These input-output pairs can then be used for regression analysis, allowing us to infer a model of the relationship between input and output. These predictions make up our risk estimates for the model, but how can we know how much of this relationship has a bearing on the real problem we have tried to model. By exploring a number of learning approaches we aim to explore this relationship.

For example, we use a support vector regression (SVR) approach to predict the damage for unseen examples given our training sets, as this makes efficient use of data, allows us to explore linear and non-linear decision functions, and does not suffer from the *false structure* problem [10]. We explore a variety of kernels and hyper-parameters to see which perform most effectively for different abstractions. We test our SVR by finding the root mean squared error (RMSE) between the predicted and actual damage for some unseen dataset. An example of the RMSEs for 5 fold cross-validation on a selection of hyper-parameters is shown in Figure 1; here, data harvested from each Monte-Carlo trace is shared across the cross-validation folds.

As suggested earlier, for this exercise to ultimately bear fruit, we will need to transfer learning at one abstraction, and apply it at an entirely different abstraction. In the most difficult case the latter will be the zero abstraction, i.e. the real world. Hence, we need to show that the transfer is valid and can effectively capture useful knowledge. We first focus our efforts into showing scalability and for that we take advantage of locality.

Preserving Locality One property we felt to be important about our simulation model, was that it had some representation of locality. To put this formally, each abstraction of the scenario model represents some mathematical relationship between observation-action and subsequent risk. Our intuition suggests that one property of this relationship is locality, meaning that values close to one another on the grid interact differently than those that are distant. When we move between abstractions, we assume something about how to preserve information relating to these relative positions and magnitudes.

Training our regression at one abstraction and testing on data from another is one way of exploring how much information in this relationship is preserved, and how much is destroyed. To achieve this, we define a mapping between input data at different abstractions. This allows us to train and test data from different abstraction on the same learning structure. Put another way, we cannot train on input vectors length 32 and directly test on vectors of length 128. Instead, we have to map the vectors from one space to that of the other space. This training in one domain then moving or mapping to another domain to apply the learning, has been called transductive transfer learning [8], and is an active area of research.

When we compare how well our regression predicts values for mapped versus unmapped data, this can be seen as providing a metric by which we can measure the difference between abstractions. The more information that is shared by different levels of abstraction, the more likely it is that either: locality has no effect on the aggregation of risk, or that the locality property is preserved as we move between abstractions.

With these cross dataset tests, we are exploring how to extract useful knowledge about how risk aggregates within our scenario model, and that there is some structure to this knowledge – independent of the granularity of the abstraction. Table 1 shows some preliminary results for training on one abstraction and testing on a potentially different abstraction (an abstraction is identified by its grid-size, $n \times n$). Where the grid sizes of the training and testing set matches, the two datasets are from separate Monte-Carlo traces of the same model; this explains the slightly lower accuracy when compared with the cross-validation results. If we were to replace our scenario model with an alternative, the amenity to abstraction for each model could be compared. In this and other ways, we hope to develop a series of benchmarks for the representational quality of such abstract models.

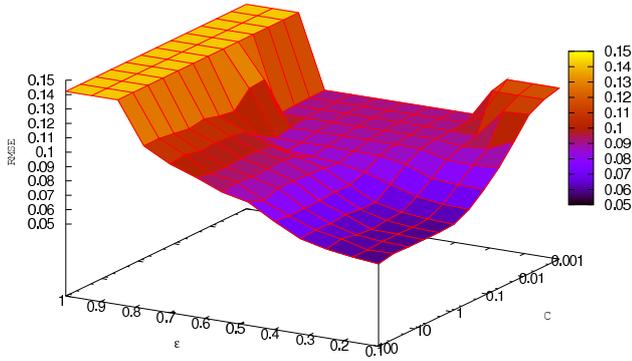


Figure 1: RMSEs for SVR cross-validation on data from 8×8 abstraction of the rescue scenario. Data harvested from each Monte-Carlo trace is shared across folds.

Train	Test	4×4	8×8	16×16	32×32
4×4		0.115	0.100	0.109	0.132
8×8		0.097	0.093	0.109	0.133
16×16		0.100	0.092	0.107	0.130
32×32		0.107	0.107	0.129	0.108

Table 1: RMSEs for SVR predictions on data trained on one abstraction and tested on another with the testing data mapped appropriately. Uses the gaussian kernel with $\gamma = 0.5$, $C = 10$ and $\epsilon = 0.3$.

Bibliography

- [1] Loz Blain. Advanced video games for us navy pilot training, March 2007. [Online: Accessed Feb 11, 2010].
- [2] P. C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. Technical Report RC24190, IBM Research, 2007.
- [3] J. C. Rosser Jr, P. J. Lynch, Cuddihy L, D. A. Gentile, J. Klonsky, and R. Merrell. The impact of video games on training surgeons in the 21st century. *Arch Surg.*, 142(2):181 – 186, February 2007.
- [4] Alexander Kott, Larry Ground, Ray Budd, Lakshmi Rebapragada, and John Langston. Toward practical knowledge-based tools for battle planning and scheduling. In *Eighteenth national conference on Artificial intelligence*, pages 894–899, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.
- [5] Yow Tzu Lim, Pau Chen Cheng, Pankaj Rohatgi, and John Andrew Clark. Mls security policy evolution with genetic programming. In *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1571–1578, New York, NY, USA, 2008. ACM.
- [6] Qun Ni, Jorge Lobo, Seraphin B. Calo, Pankaj Rohatgi, and Elisa Bertino. Automating role-based provisioning by learning from examples. In Barbara Carminati and James Joshi, editors, *SACMAT*, pages 75–84. ACM, 2009.
- [7] JASON Program Office. HORIZONTAL INTEGRATION: Broader Access Models for Realizing Information Dominance. Technical Report JSR-04-132, The Mitre Corporation, Mclean, Virginia, December 2004.
- [8] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 99(1), 5555.
- [9] Mudhakar Srivatsa, Shane Balfe, Kenneth G. Paterson, and Pankaj Rohatgi. Trust management for secure information flows. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM Conference on Computer and Communications Security*, pages 175–188. ACM, 2008.
- [10] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, November 1995.