POSTER CATALOGUE
WASP Winter Conference 2020

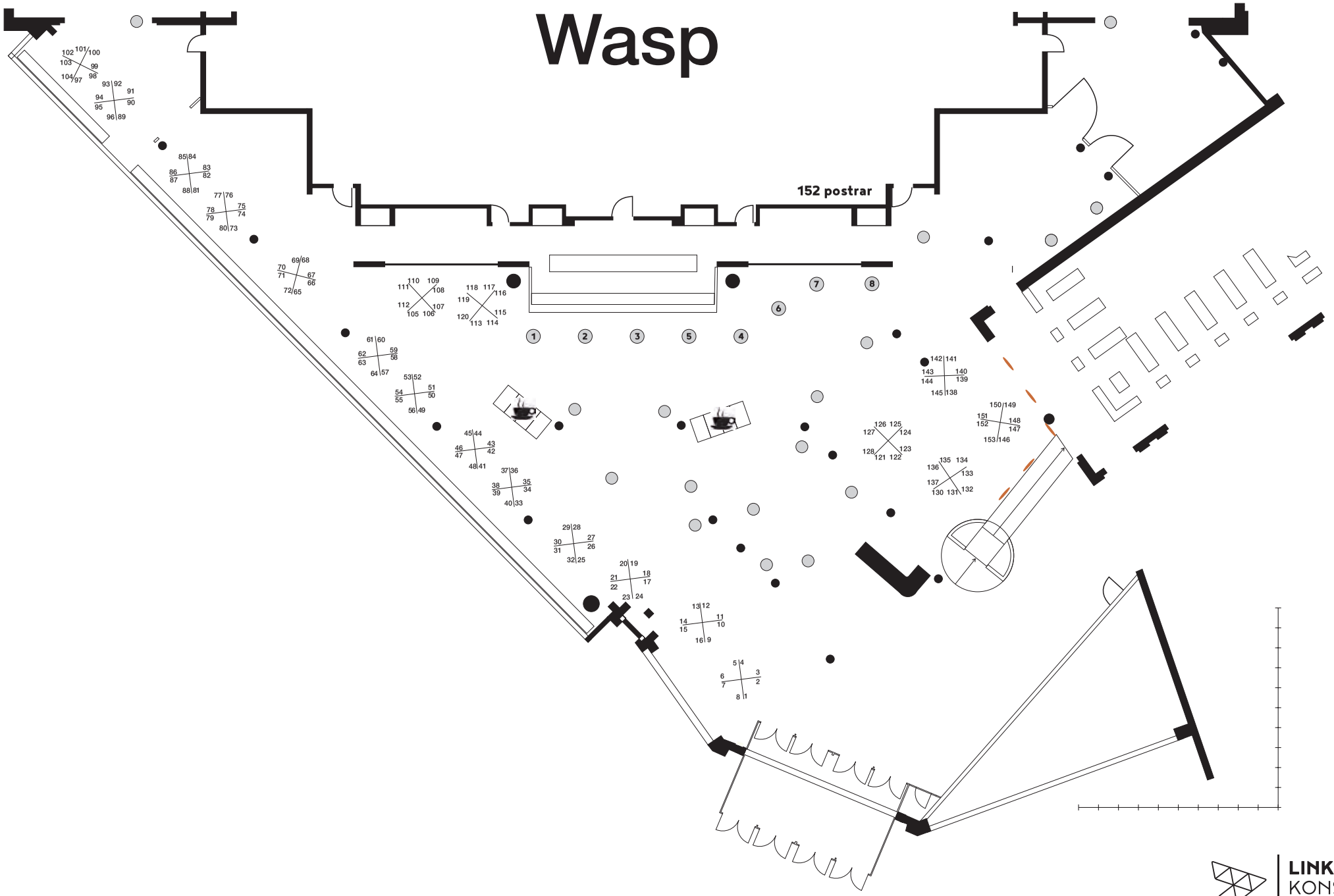
Poster Session 4
Wednesday 15 January 11.00-12.00

Poster no.	First name	Last name	Title of poster
4	Adam	Dahlgren Lindström	Generating Scene Graphs Conditioned on Images and Text *
8	Adam	Lindhe	A Large Deviation Approach for a Goodnes-of-Fit test for Variational Autoencoders
12	Albin	Heimerson	Reinforcement learning using an online datacenter model
16	Alexander	Karlsson	Automatic Target Recognition with Noncoherent Radar and Deep Learning
20	Alexander	Dürr	Robot Skill Learning based on Interactively Acquired Knowledge-based Models *
24	Alexandre	Martins	Autonomous learning camera systems in resource constrained environments
28	Alexandru	Dura	MetaDL: Declarative Program Analysis for the Masses
32	Amanda	Olmin	Probabilistic models and deep learning - bridging the gap
36	Amber	Zelvelder	The Interpolating, Normalising and Kernel Allocating (INKA) Network
40	Amir	Roozbeh	Make the Most Out of Last Level Cache in Intel Processors
44	Anton	Johansson	Model Selection for Neural Networks
48	Arman	Rahbar	Analysis of Optimization and Generalization in Neural Networks *
52	Axel	Berg	Deep Networks for Ordinal Regression
56	Bram	Willemsen	Developing a GWAP to collect visually-grounded collaborative dialogue *
60	Carl-Johan	Hoel	Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving
64	Caroline	Svahn	Inter-frequency radio signal quality prediction for handover, evaluated in 3GPP LTE
68	Chanh	Nguyen	Toward Elasticity Control For Edge Data Centers *
72	Christian	Nelson	Ultra-Reliable and Low-Latency Wireless Communication for Control
76	Christopher	Blöcker	Mapping flows in bipartite networks *
80	Christos	Matsoukas	AI-supported endpoints in chronic kidney disease
84	Claudio	Mandrioli	Modeling of Energy Consumption in GPS Receivers for Power Aware Localization Systems
88	Daniel	Ahlsén	Verification of Quantitative Multi-Agent Systems
92	Daniel	Gedon	Deep Learning for Sequential Data
96	-	-	-
100	Fredrik	Präntare	Simultaneous Coalition Structure Generation and Assignment
104	-	-	-
108	Héctor	Rodríguez Déniz	A Bayesian Dynamic Multilayered Block Network Model
112	Johan	Källström	Adaptive Air Combat Training Systems for Competency Based Training
116	Karl	Norrman	Cryptographic Protocol Verification
120	Kristin	Nielsen	Underground Positioning
124	Martin	Pallin	-
128	Martin	Isaksson	Secure Federated Learning in 5G Mobile Networks
132	Masoud	Bahraini	Uncertain constrained networked control systems
136	Mohammad Ali	Nazari	-
140	Nikita	Korzhitskii	Characterizing the Trust Landscape of Certificate Transparency Logs
144	Noric	Couderc	Smart modules: Automatic selection of collection classes
148	Timotheus	Kampik	Models of Bounded Rationality for Autonomous Agents

*) Poster not available in this catalogue

WASP | WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Wasp



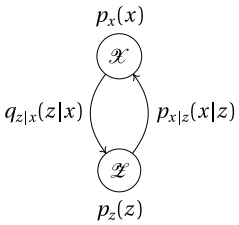
A Large Deviation Approach for a Goodnes-of-Fit test for Variational Autoencoders

A. Lindhe, H. Hult,

Department of Mathematics, Mathematical statistics, KTH, Stockholm, Sweden

① Variational Autoencoders

- The goal is to be able to sample from a complex distribution $p_x(x)$ from which we have drawn data $x_1, x_2, \dots, x_n$
- Do this by draw a z from a latent space $\mathscr{Z}$ with prior p_z, and the draw x from the likelihood or decoder $p_{x|z}(x|z)$.
- The decoder $p_{x|z}(x|z)$ is in a parametric family of distributions and is paramterized by a neural network.
- The posterior $p_{z|x}(z|x)$ is often intractable. Instead it is approximated by variational methods by the encoder $q_{z|x}(z|x)$ that again is parametrized by a neural network.



- The target is then to maximize the Evidence lower bound (ELBO) given by

$$\text{ELBO}(p_{x|z}, q_{z|x}) = \sum_{i=1}^{n} E_{q_{z|x}(z|x_i)}\left[\log(p_{x|z}(z|x_i))\right] + D_{KL}\left(q_{z|x}(z|x_i)|p_z(z)\right).$$

- D_{KL} is the Kullback-Leibler divergence.
- The first term is the reconstruction term, making sure that the generated pictures look similar to the samples.
- The second term is a tractable way of forcing the encoder to be close to the true posterior.

.

② Large Deviation Theory

- With large deviation theory the goal is to characterize the asymptotic behaviour of a series of measures μ_n.
- This is done by finding the rate function $I(\cdot)$ fulfilling

$$\lim_{n\to\infty} \frac{-1}{n} \log\left(P(\mu_n \in \Gamma)\right) = \inf_{\nu\in\Gamma} I(\nu).$$

- $I(\cdot)$ gives the exponential decay rate of the probability that the measures is in some extreme set.
- We say that μ_n fulfils the large deviation property with rate function $I(\cdot)$
- The following theorem gives the rate function for the empirical measure.

Theorem (Sanov)

Let $X_1, X_2, \dots, X_n$ be IID random variables with law μ and define the empirical measure L_x^n

$$L_x^n = \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i},$$

where δ_x is the point mass at x. Then L_x^n fulfils the large deviation property with rate function

$$I(\nu) = D_{KL}(\nu|\mu).$$

- Let μ_n fulfil the large deviation property with rate function $I(\cdot)$. Then for large values of n we have that

$$P(\mu_n \in \Gamma) \approx \exp\left(-n \inf_{\nu\in\Gamma} I(\nu)\right).$$

③ Rate function for the encoder

- If the encoder $q_{z|x}$ is close to the true posterior $p_{z|x}$ then the following holds

$$p_z(z) = \int_{\mathscr{X}} p_{z|x}(z|x)p_x(dx) \approx \int_{\mathscr{X}} q_{z|x}(z|x)p_x(dx) \approx \frac{1}{n}\sum_{i=1}^{n} q_{z|x}(z|x_i) = L_z^n(z).$$

- Using the contraction principle from large deviation theory we can prove the following theorem

Theorem (Large deviation principle for encoder)

Let the encoder be equal to the true posterior $q_{z|x} = p_{z|x}$. Then the empirical measure L_z^n defined above satisfies the large deviation property with rate function

$$I(\nu) = \inf_{\mu}\left\{ D_{KL}(\mu(x)|p_x(x)) \;:\; \nu(z) = \int_{\mathscr{X}} p_{z|x}(z|x)\mu(dx)\right\}.$$

The rate function above is often intractable but there exists lower bound that can be calculate with standard Monte-Carlo methods

$$I(\nu) \geq D_{KL}(\nu(z)|p_z(z)).$$

④ Goodness-of-Fit

A Goodness-of-Fit test can now be constructed as follows

- Define a "interesting" set that L_z^n belongs to. For example

$$A_\epsilon = \{\nu \in \mathscr{M}(\mathscr{Z}) \; s.t \; d(\nu, p_z) \geq \epsilon\}.$$

  For some metric $d(\cdot,\cdot)$ such that $d(L_z^n, p_z) = \epsilon$. Useful metrics could be D_{KL} or the Wasserstein metric.

- Compute a large sample p-value

$$p = P(L_z^n \in A_\epsilon) \approx \exp\left(-n \inf_{\nu\in A_\epsilon} I(\nu)\right) \leq \exp\left(-n \inf_{\nu\in A_\epsilon} D_{KL}(\nu|p_z)\right).$$

- A model with higher p-value is regarded, at least in latent space, to be better.

⑤ MNIST example

We train two VAEs on the MNIST dataset with different structures. In Figure 1 and 2 the empirical measure is plotted against the true prior and some generated images from this VAE.

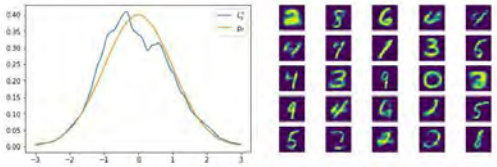


Figure 1: One dimension of the latent space and some generated numbers

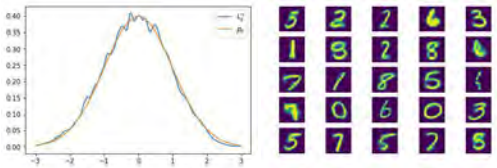


Figure 2: One dimension of the latent space and some generated numbers

There seem to be a correlation between how close L_z^n and p_z are and the quality of the generated image.

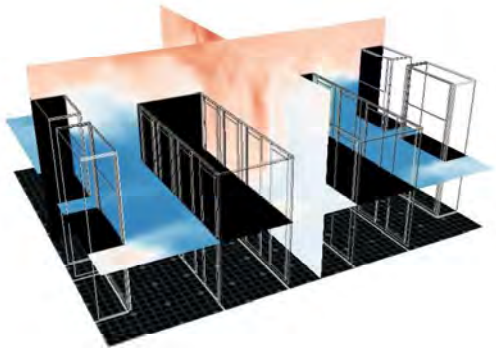
WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Reinforcement learning using an online datacenter model

Lund University, RISE SICS North

Introduction

With an electricity consumption that amounts to more than 1% of the world's total the data center industry can greatly benefit from any small improvement in consumption.
We look at distributing jobs with the help of reinforcement learning so that hotspots are kept away and the cooling energy minimized.



Model

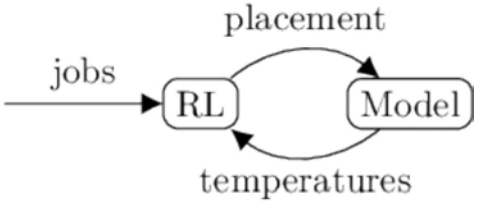
We want to use CFD models based on lattice Boltzmann methods to be able to:

- Run fast simulations of the temperature flow.
- Train RL algorithms to keep hotspots away.

CFD is normally a very compute intensive and slow process, but the lattice boltzmann method allows for both changing the granularity of the simulation and parallelizing the computation. This can speed up the simulation to allow it to run faster than real time.

Reinforcement learning

Starting with a very simple setup the reinforcement learning algorithm choose where to place incoming jobs based on the jobs and current temperatures.
With this we can run on-policy algorithms and learn in an online fashion with respect to the model.



Conclusion

We want to show that we can create efficient pre-training for the RL agent with the help of CFD simulations and in the end create a temperature aware job-scheduling algorithm that can be transferred to real systems.

Automatic Target Recognition with Noncoherent Radar and Deep Learning

Alexander Karlsson

Description:

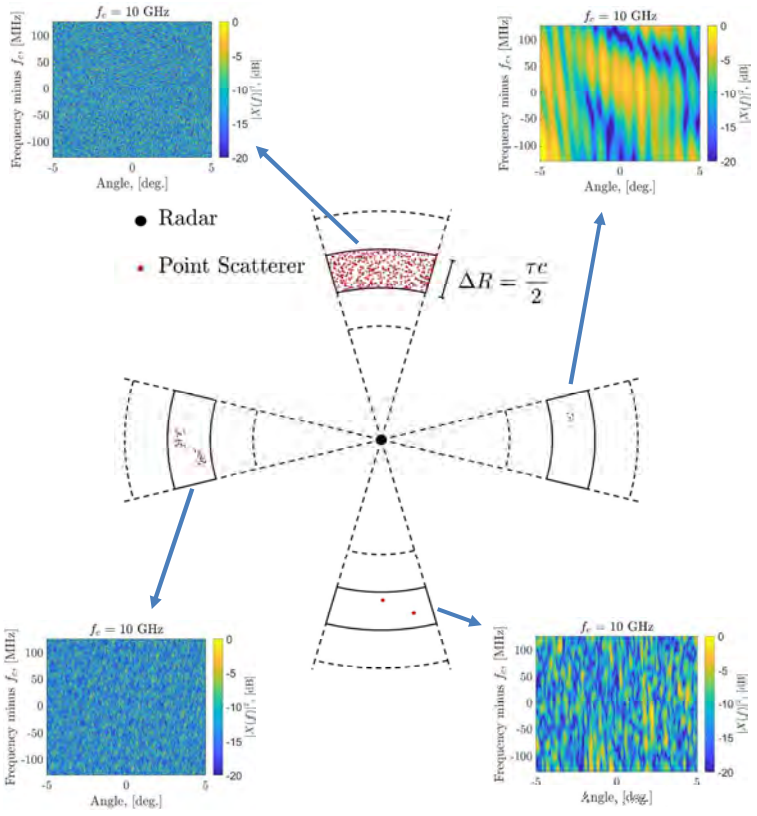
Classifying radar echoes has been a hot topic since the dawn of radar. As a result many methods exists for Automatic Target Recognition (ATR) in radar systems. A common method is to analyze the High Range Resolution Profile (HRRP) of the target. Recent approaches for this involves machine learning and Deep Learning techniques. In this study we will look at the possibility of obtaining the HRRP with noncoherent radar as a means for ATR. This is also done using Deep Neural networks (DNN).

Background

Complex objects such as airplanes, cars, etc. are often modeled as a set of point scatterers. The received echo from such a target with K scatteres can, after complex down conversion, be approximated as

$$\mathrm{X}(f_i) \propto \lambda_{c+i} e^{j\phi_i} \sum_{k=0}^{K-1} \sqrt{\sigma_k}\, G(\alpha_k - \varphi, \delta_k - R) e^{-j2\pi(f_c - f_i)\frac{2\delta_k}{c}} + w_i$$

where $\mathrm{X}(f_i)$ is the received signal voltage, ϕ_i is the phase of the transmitted pulse, $\lambda_{c+i} = c/(f_c + f_i)$ is the wavelength of the transmitted pulse, $\sigma^2$is the RCS of the $k\,'$th scatterer, G is a gain and depends on the difference between the angle to the scatterer α_k and the antenna look direction φ (azimuth antenna gain) as well as the difference between the current range bin R and the range to each scatterer δ_k (pulse gain), and w_i is complex noise. Below are some illustrations of how $|\,\mathrm{X}(f)\,|$ may look as a function of f and aspect angle for different scatter models.



Methods

If the pulse phase ϕ_i is known and f_i is chosen as

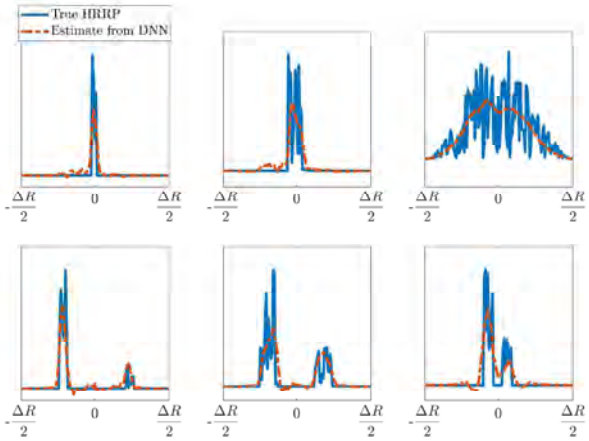
$$f_i = \left(i - \frac{N-1}{2}\right)\frac{c}{2\Delta R}$$

for integer $i = 0,1,\ldots,\mathrm{N}-1$, one could estimate the HRRP as the absolute value of the inverse Fourier transform of $\mathrm{X}(f_i)$. This would yield a high range resolution of $\Delta\rho = \frac{\Delta R}{N} = \frac{c}{2BW}$ and is referred to as a stepped-frequency waveform. In this case the radar is noncoherent i.e. the phase is not known. We thus have to look at methods of obtaining the HRRP $x[n]$ from $|X(f)|$ instead. This is a phase retrieval problem where one aims to recover the phase of a spectrum in order to calculate the inverse transform. This is a hopless problem without some prior knowledge of the original signal $x[n]$, e.g. that it is real and non-negative. In such cases iterative algorithms exist that can recover $x[n]$ from $|X(f)|$. In this case we instead use DNN to recover $x[n]$ directly from $|X(f)|$.

$$|X(f)| \rightarrow \boxed{\text{DNN}} \rightarrow \hat{x}[n]$$

Examples

In the following example a fully connected DNN has been trained on simulated data with $|X(f)|$ as input and $x[n]$ as label. The labels $x[n]$ were always centered in the same manor.



Autonomous learning camera systems in resource constrained environments

Alexandre Martins

Description

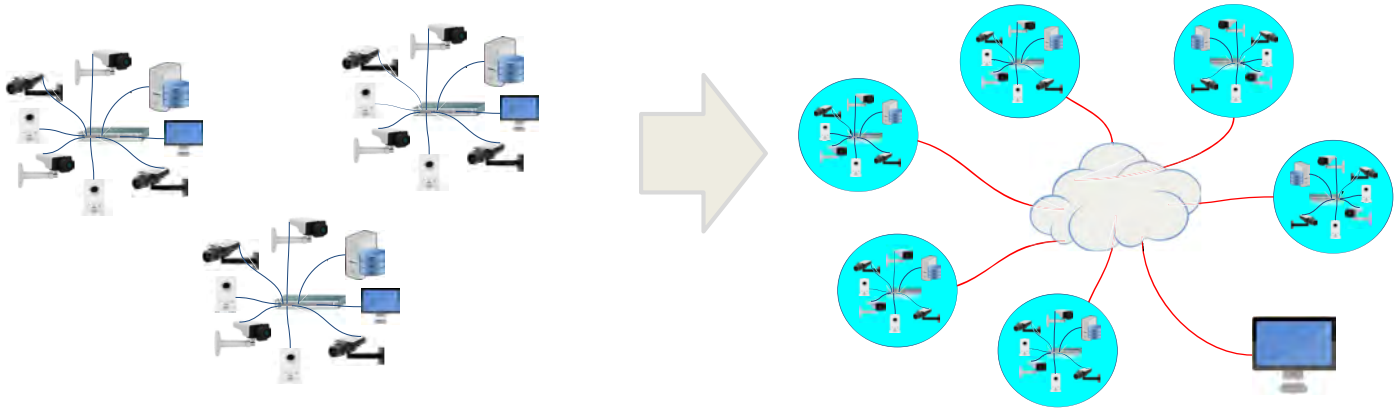
This project aims to turn camera systems into a swarm of autonomous scene-learning devices that share the same resources, turning today's central server (if any is available) as a viewing-only client. The systems will make sure that available resources are dynamically and optimally allocated at all time. The swarm will be completely flexible allowing devices to be added or removed from it and re-allocating resources accordingly. Each of these devices will be communicating with its surroundings, and, will in the process learn situation specific parameters, such as resources availability and expenditure, scene properties etc, in order to predict future resource needs and allow for superior system wide resource management.

Background & Motivation

The future networked society will contain a huge number of devices, many of them processing a very large amount of sensor data. One example of this is distributed video cameras in surveillance and supervision applications. Due to efficiency and price constraints the communication and computing platforms are often limited, hence dynamic resource management is required.

Research Goal & Questions

The focus of this project is on dynamic resource management of large-scale autonomous self-expressive and self-learning devices, in our case mostly video camera systems. The project has strong connections to the WASP Autonomous Cloud project which also focuses on resource management of distributed systems and aims at bringing a smarter world of interconnected devices in tomorrow's infrastructures.



Publications

[2018] Camera networks dimensioning and scheduling with quasi worst-case transmission time
30th Euromicro Conference on Real-Time Systems, ECRTS 2018.

[2018] H.264 video frame size estimation - Technical report
Department of Automatic Control, Lund Institute of Technology, Lund University.

[2020] Control-Based Resource Management for Storage of Video Streams
International Federation of Automatic Control, IFAC 2020

Current work

MPC (model predictive controller) using the model published in 2018. (standby)

Game theory reward assignment for storage allocation

MetaDL

Declarative Program Analysis for the Masses

Alexandru Dura | Hampus Balldin | Christoph Reichenbach

Department of Computer Science | Lund University

MetaDL program - type hierarchy

```
IMPORT('P, "ch.java", "java4").

analyze ('P, "java4") {
  InterfaceExtendsInterface(i, j) :-
       <: interface `i extends `j { .. } :>,
       ID(`i, i), ID(`j, j).
  ClassImplementsInterface(c, i), ClassExtendsClass(c, d) :-
       <: class `c extends `d implements .., `i, .. { .. } :>,
       ID(`c, c), ID(`d, d), ID(`i, i).
  ClassExtendsClass(c, d) :-
       <: class `c extends `d { .. } :>,
       ID(`c, c), ID(`d, d).
  ClassImplementsInterface(c, i) :-
       <: class `c implements .., `i, .. { .. } :>,
       ID(`c, c), ID(`i, i).
}
SuperClass(c, s) :- ClassExtendsClass(c, s).
SuperClass(c, s) :- ClassExtendsClass(c, d),
                    SuperClass(d, s).
SuperInterface(i, s) :- InterfaceExtendsInterface(i, s).
SuperInterface(i, s) :- InterfaceExtendsInterface(i, j),
                        SuperInterface(j, s).
Interface(c, i) :- ClassImplementsInterface(c, i).
Interface(c, i) :- SuperClass(c, d), Interface(d, i).
Interface(c, i) :- Interface(c, j), SuperInterface(j, i).

OUTPUT('Interface).
OUTPUT('SuperClass).
OUTPUT('SuperInterface).
```

Analyzed Java program

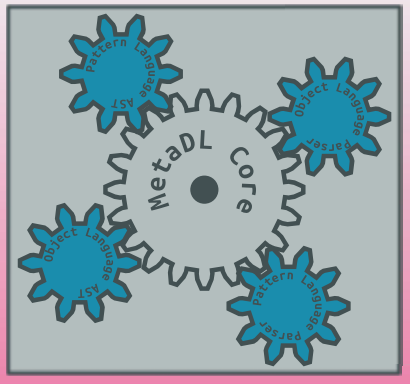
```
interface  I  {  }
class  A  {  }


interface  J  extends  I  {
       int  f();
}
interface  H  extends  J  {
       int  g();
}


class  B  extends  A  implements  H  {
       public  int  f()  {
              return  1;
       }
       public  int  g()  {
              return  0;
       }
}


class  C  extends  B  {
  public  void  h()  {  }
}


class  D  implements  I  {  }
```



Relational representation of patterns

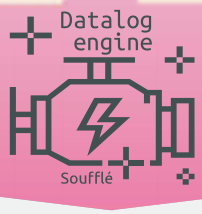
```
// The clause:
// ClassImplementsInterface(c, i), ClassExtendsClass(c, d) :-
//      <: class `c extends `d implements .., `i, .. { .. } :>,
//      ID(`c, c), ID(`d, d), ID(`i, i).
// is transformed to the following relational representation:

ClassImplementsInterface(c, i),
ClassExtendsClass(c, d) :-
              P("ClassDecl", t1, 0, t2, _),
              P("ClassDecl", t1, 1, ?c, _),
              P("ClassDecl", t1, 2, ?d, _),
              P("ClassDecl", t1, 3, t3, _),
              P("ClassDecl", t1, 4, _, _),
              P("Modifiers", t2, 0, _, _),
              P("List", t3, _, ?i, _),
              ID(?c, c), ID(?d, d), ID(?i, i).

ID(n, v) :- P("IdWrapper", n, 0, t, _),
            P("Terminal", t, _, _, v).
ID(n, v) :- P("ParseName", n, 0, t, _),
            P("Terminal", t, _, _, v).
```

Relational representation of programs

Node kind	Node ID	Child Index	Child ID	Optional
ClassDecl	4	0	5	
ClassDecl	4	1	6	
ClassDecl	4	2	7	
ClassDecl	4	3	8	
ClassDecl	4	4	9	
Modifiers	5	0	10	
IdWrapper	6	0	2	
ParseName	7	0	3	
List	8	0	11	
List	9	0	12	
List	9	1	13	
ParseName	11	0	4	
MethodDecl	12	0	14	
…	…	…	…	
MethodDecl	13	0	27	
…	…	…	…	
Modifiers	14	0	20	
List	20	0	21	
Modifier	21	0	5	
Modifiers	27	0	33	
List	33	0	34	
Modifier	34	0	5	
Terminal	2	0	0	B
Terminal	3	0	0	A
Terminal	4	0	0	H
Terminal	5	0	0	public



Results

Class	SuperClass
B	A
C	B
C	A

Class	Interface
B	H
C	H
B	I
B	J
C	I
C	J
D	I

Interface	SuperInterface
J	I
H	J
H	I

Runtime

At runtime, an instance of the MetaDL system takes as input a MetaDL file and one or multiple files containing the programs under analysis. MetaDL invokes the object language parser which produces the ASTs of the analyzed programs. The resulting ASTs are then transformed to program representation relations. In the same stage, MetaDL generates ASTs for the syntactic patterns and transforms these ASTs to plain Datalog clauses (cf. [1]). The execution continues with the translation of the resulting program into the Datalog dialect used by the Soufflé engine [3]. The program representation relations are also serialized. Finally, MetaDL invokes the Soufflé engine to compute the results.

The MetaDL system

The MetaDL system consists of the **MetaDL core language** and a set of **grammar extension tools**. The **MetaDL core language** is an extension of Datalog with **syntactic patterns** [1]. The syntactic patterns allow the user to match subtrees in the AST of the analyzed program and to refer to the matched subtrees in other plain Datalog atoms. To illustrate, the pattern

```
<: class `c implements .., `i, .. { .. } :>
```

expresses the "class c implements interface i" relation from a Java program. The language of the syntactic patterns extends the object language with **metavariables** (e.g. `c and `i) and **gaps** (..). **Metavariables** bind to any AST subtree that is allowed by the object language in their position. In our framework, list nodes represent sequences (e.g. implemented interfaces, statements, method parameters), while all other node kinds have a fixed number of children. **Gaps** represent ignored elements inside these list nodes.

By using a description of the concrete and abstract grammars, the grammar extension tools generate the language of the syntactic patterns. While we currently support only Java 1.4 and MetaDL itself, the grammar extension tools can be used to add new languages with minimal effort.

System configuration

To instantiate the MetaDL system for a particular language, the user must provide an EBNF description of the concrete grammar and a description of the abstract grammar. To extend the abstract grammar with metavariables and gaps, we use a modified version of JastAdd [2], a metacompilation framework. A related tool, JastAddParser, is used to generate the pattern grammar. The pattern grammar is a super-set of the object grammar, where every non-terminal has two supplementary productions: a metavariable and a gap. While an LR(1) parser is used for the object language and for the main MetaDL program, the pattern language is possibly ambiguous and thus requires the use of an Earley parser that generates all the possible parse trees. The parsers and the classes describing the abstract grammars are packaged together with the MetaDL core language.

Object language description

Concrete grammar

```
ClassDecl class_declaration = modifiers? 'class' wrapped_name super? interfaces? class_body
Access super = 'extends' class_type
List interfaces = 'implements' interface_type_list
List interface_type_list = interface_type
                         | interface_type_list ',' interface_type
```

Abstract grammar

```
ClassDecl : ReferenceType ::= Modifiers IdWrapper [Extends:Access] Implements:Access* BodyDecl*
IdWrapper ::= <ID:String>;
ParseName : Access;
Modifiers ::= Modifier*;
Modifier ::= <ID:String>;
```

Configure once for each analyzed language

The pattern grammar - generated from the object grammar

```
ClassDecl class_declaration = modifiers? 'class' wrapped_name_mv super? interfaces? class_body

Access wrapped_name_mv = wrapped_name
                       | '`ID'

Access super = 'extends' class_type_mv

Access class_type_mv = class_type
                     | '`ID'

List interfaces = 'implements' interface_type_list

List interface_type_list = interface_type_mv
                         | interface_type_list ',' interface_type_mv

Access interface_type_mv = interface_type
                         | '`ID'
                         | '..'

List modifiers = modifier_mv
               | modifiers modifier_mv

Access modifier_mv = interface_type
                   | '`ID'
                   | '..'
```

Legend

`ID - metavariable
.. - gap
class_type_mv - metavariable production derived from class_type

References

[1] Alexandru Dura, Hampus Balldin, and Christoph Reichenbach. 2019.MetaDL: Analysing Datalog in Datalog. In Proceedings of the 8th ACM SIGPLAN International Workshop on State Of the Art in Program Analysis (SOAP 2019). ACM, New York, NY, USA, 38–43. https://doi.org/10.1145/3315568.3329970

[2] Görel Hedin and Eva Magnusson. 2003. JastAdd: An Aspect-oriented Compiler Construction System. Sci. Comput. Program. 47, 1 (April 2003),37–58. https://doi.org/10.1016/S0167-6423(02)00109-0

[3] Bernhard Scholz, Herbert Jordan, Pavle Subotić, and Till Westmann.2016. On Fast Large-scale Program Analysis in Datalog. In Proceedings of the 25th Int. Conf. on Compiler Construction (CC 2016). ACM, NewYork, NY, USA, 196–206. https://doi.org/10.1145/2892208.2892226

github.com/lu-cs-sde/metadl
alexandru.dura@cs.lth.se
christoph.reichenbach@cs.lth.se

This work was partially supported by WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Probabilistic models and deep learning - bridging the gap

Amanda Olmin, Jakob Lindqvist

Project background

Probabilistic models come with many desirable properties in that they:

- enable reasoning about uncertainties inherent to most data
- can be constructed hierarchically to build complex models from simple parts
- provide a natural safeguard against overfitting
- allow for fully coherent inferences over complex structures from data

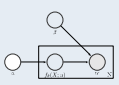
Deep learning is another branch of machine learning which has had great impact in several fields, but is still lacking many of these desirable properties. The goal of probabilistic deep learning is to retain the performance while gaining the properties of probabilistic models.

The project is a cooperation between Amanda Olmin at LiU and Jakob Lindqvist at Chalmers, supervised by Fredrik Lindsten (LiU) and Lennart Svensson (Chalmers).

Probabilistic deep learning

Probabilistic deep learning is the combination of deep learning and probabilistic models. The goal is to use DL algorithms as integral parts of probabilistic models. Both for more accurate observations but also to improve training of DL models, by making use of

- weak annotations
- partially annotated sequences of data



This requires good uncertainty characterization for the DL models. We note that:

- the model weights can be viewed as random variables $\theta \sim p(\theta|\mathcal{D})$
- there could be inherent noise in the environment that our model do not account for

Inference gives a probability distribution over the network output $y \sim p(y|\mathcal{D}, x)$ and enables us to capture both model and environment uncertainty.

Uncertainty characterization

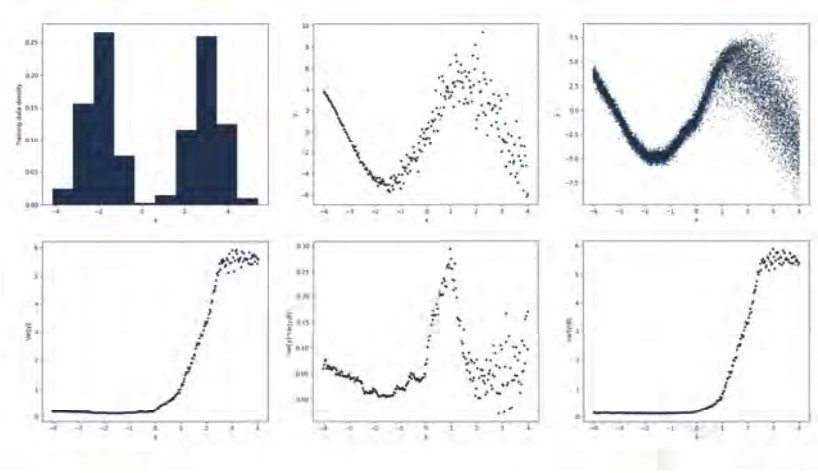
The total uncertainty is split into model (*epistemic*) and environment (*aleatoric*) uncertainty.

Variance measure:

$$\underbrace{\mathrm{Var}[y|x]}_{Total} = \underbrace{\mathrm{Var}_\theta[\mathbb{E}[y|x,\theta]]}_{Epistemic} + \underbrace{\mathbb{E}_\theta[\mathrm{Var}[y|x,\theta]]}_{Aleatoric}$$

Entropy measure:

$$\underbrace{\mathbb{H}[p|x,\mathcal{D}]}_{Total} = \underbrace{\mathbb{I}[p;\theta|x,\mathcal{D}]}_{Epistemic} + \underbrace{\mathbb{E}_\theta[\mathbb{H}[p|x,\theta]]}_{Aleatoric}$$



Uncertainty separation obtained with method by [1].

Distilling probabilistic ensembles

Ensembling methods have been succesfully used to represent epistemic uncertainty.

Using some ensembling method where ensemble members predict a parametric distribution of the output rather than the output itself, allows us to quantify both epistemic and aleatoric uncertainty, [3].

Network distillation aims to slim the ensemble model while retaining its benefits. We propose a distillation procedure which learns a higher-level task than the ensemble members (similar but more general approach than [4]):

An ensemble with members propose distributions

$$y \sim p(y; f_{\theta_i}(X)).$$

A distilled model learns to resemble the ensemble:

$$f_{\theta_i}(X) \sim p(f_{\theta_i}(X); g(X)).$$

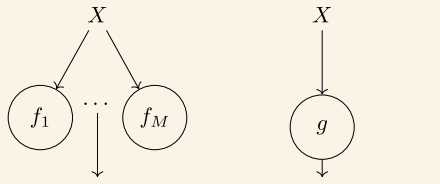
The method works for both classification and regression; it differs only in the probability distr. chosen to model the output.

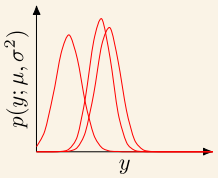
For classification, fit Dirichlet distribution:

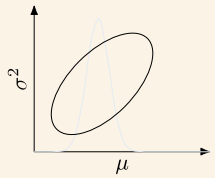
$$f_{\theta_i}(X) \sim Dir(f_{\theta_i}(X); \alpha).$$

For regression, fit Normal-inverse-Wishart:

$$f_{\theta_i}(X) \sim \mathcal{NIW}(f_{\theta_i}(X); \mu, \lambda, \Psi, \nu).$$







Active learning

Active learning can be used to decrease annotation costs by selective labeling of data points [5]. The assumption is that data is abundant while annotation is costly.

Objective:

- minimize the error rate, ϵ, using a fixed annotation budget, N, indicating how many samples we can afford to annotate *or*
- minimize the number of samples, N, we need to annotate in order to get a model with an error rate of ϵ or smaller

Idea: exploit uncertainty separation; use epistemic uncertainty as acquisition function (as in e.g. [2]):

$$x^* = \underset{x}{\mathrm{argmax}}\ \mathrm{Var}_\theta[\mathbb{E}[y|x,\theta]]$$

or

$$x^* = \underset{x}{\mathrm{argmax}}\ \mathbb{I}[p;\theta|x,\mathcal{D}]$$

References

[1] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Uncertainty Decomposition in Bayesian Neural Networks with Latent Variables. *arXiv e-prints*, page arXiv:1112.5745, Nov 2017.

[2] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning. *arXiv e-prints*, page arXiv:1112.5745, Dec 2011.

[3] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, pages 5574–5584, 2017.

[4] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble Distribution Distillation. 2019.

[5] Burr Settles. Active Learning Literature Survey. *Sciences-New York*, 2009.

The Interpolating, Normalising and Kernel Allocating (INKA) Network

UMEÅ UNIVERSITY
Amber Zelvelder, Umeå university
Department of Computer Science

Abstract

This poster describes the Interpolating, Normalising and Kernel Allocating (INKA) neural network that is based on the principles of radial-basis-function (RBF) networks. Modifications to the basic RBF network concern the training algorithm, the use of a non-Gaussian output function for the hidden neurons and the use of normalised inputs to the neurons of the output layer instead of absolute ones. Several tests on known functions show that these modifications give INKA learning performances which are superior to those of both multilayer backpropagation networks and usual RBF networks. These improvements concern learning speed, sensibility to local minima, precision, definition of training parameters and generalization capabilities.

Background

This poster presents a neural network (NN) that is based on the principles of the so-called radial-basis-function (RBF) networks. This NN approximates functions by interpolation between the peaks of the radial basis functions instead of trying to make them fit directly to the approximated function, as is usually done in RBF networks. This is the main reason why the network is called the Interpolating, Normalising and Kernel Allocating (INKA) network. Kernel Allocating means that the number of hidden neurons is adjusted in order to improve the function approximation.

Localized neural networks are mainly used in classification tasks, but work has also been done on continuous function approximation [1,2,3,4,5]. Since one RBF neuron usually covers only a limited input subspace, there are often big holes which are not covered at all unless the shape of the RBF functions is perfectly tuned to the approximated function. In particular, multidimensional input vector spaces can easily require an impractically large number of neurons. However, the local nature of the neurons means that the RBF networks can learn relatively fast [6,7,8] Even one-shot algorithms exist for RBF networks [9].

Backpropagation (BP) networks have been successfully applied to both function approximation and classification problems [10]. As the hidden neurons have a global output function, there are usually no holes in the input vector space covered by the BP network. Unfortunately, the global nature of the BP network makes learning very slow. This is especially true for the high-dimensional input vector spaces. These advantages and inconveniences have been verified by the tests performed in section 5 with both RBF and BP NNs. The INKA network presented in this article combines the best aspects of the BP network (continuous and global function approximation).

References

1) Klopfenstein, R. W., S.R.: Approximation by uniformly spaced gaussian functions. Approximation Theory 4, 575–580 (1983)
2) Hartman, E.J., Keeler, J.D., Kowalski, J.M.: Layered neural networks with gaussian hidden units as universal approximations. Neural Computation 2(2), 210–215 (jun 1990)
3) Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation 3(2), 246–257 (jun 1991)
4) Park, J., Sandberg, I.W.: Approximation and radial-basis-function networks. Neural Computation 5(2), 305–316 (mar 1993)
5) Benaim, M.: On functional approximation with normalized gaussian units. Neural Computation 6(2), 319–333 (mar 1994)
6) Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Computation 1(2), 281–294 (jun 1989)
7) Debenham, R.M., Garth, S.C.J.: The detection of eyes in facial images using radial basis functions. In: Neural Networks for Vision, Speech and Natural Language, pp. 50–64. Springer Netherlands (1992)
8) Platt, J.: A resource-allocating network for function interpolation. Neural Computation 3(2), 213–225 (jun 1991)
9) Jokinen, P.A.: DYNAMICALLY CAPACITY ALLOCATING NETWORK MODELS FOR CONTINUOUS LEARNING. In: Artificial Neural Networks, pp. 1153–1156. Elsevier (1991)
10) McClelland, J.L., Rumelhart, D.E., Hinton, G.E.: The appeal of parallel distributed processing. Morgan Kaufmann (1988)…

Algorithm

The main principle of the training algorithm is to allocate new hidden neurons where the output error is the greatest, thus trying to reduce the global error as much as possible at each step. However, the distance between the centroids of two hidden neurons is not allowed to be smaller than the constant c. The Algorithm as presented below is applicable for one output, but can be applied to multiples as well.

ALGORITHM 1 (Training)
1. Choose the values for d_2 and c parameters.
2. Start off with an empty hidden layer, which means the network output is always zero.
3. Initialize the set of resting examples R to the whole training set.
4. Select the training example in R for which the output error is the greatest, add it to the set of used examples U and remove it from the set of resting examples R .
5. If there is no hidden neuron closer than c to the example, allocate a new hidden neuron i with $\vec{w}_i$ equal to $\vec{x}$.
6. Recalculate the B matrix for the examples in U .
7. Test if the global error measure is small enough. If not, go back to step 4

A simple optimisation algorithm (Algorithm 2) may be executed after training to remove redundant neurons that might be created during training.

ALGORITHM 2 (Optimisation)
1. Temporarily remove one hidden neuron at a time, recalculate the B matrix on the whole training set and register the resulting
global error for each case.
2. Identify the neuron removal that caused the smallest error increase (the global error may even decrease with the removal of some neurons).
3. If the maximum limit for global error is not violated by the removal of the identified neuron, the neuron is removed permanently (this means that neurons for which the global error decreases are always removed).
4. Go back to step one until no more neurons can be removed with out increasing the global error too much.

Tests and comparisons

For testing our algorithm, we will be comparing it to a plain RBF function and a normalized RBF function written in the same programming language. If we can we will also include a comparison to a basic backpropagation network. We will be doing our testing on two different datasets that vary in the amount of parameters provided. The parameters we will be inspecting to show the performance of the algorithm will be learning speed and complexity. The learning speed will be measured separately for the optimisation and learning algorithms where possible to provide a more detailed comparisons. For the complexity the main data measured will be the number of neurons generated.

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Make the Most Out of Last Level Cache in Intel Processors

Contact: Amir Roozbeh <amir.roozbeh@ericsson.com>

Email me

1 Problem

Faster links exposes processing elements to packets at a higher rate, but the performance of the processors is no longer doubling at the earlier rate, making it harder to keep up with the growth in link speeds. For instance, a server receiving 64 B packets at a rate of 100 Gbps has only 5.12ns to process a packet before the next packet arrives.

Approach:

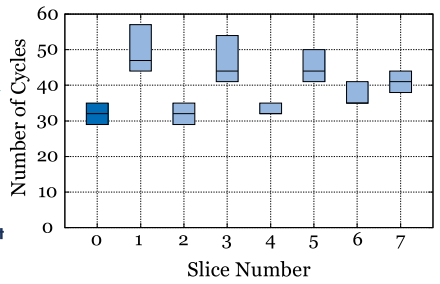
It is essential to exploit every oppurtunity to optimize current computer systems. We focus on better management of LLC.

2 Slice-aware Memory Management

In modern (Intel) processors, Last Level Cache (LLC) is divided into multiple slices.

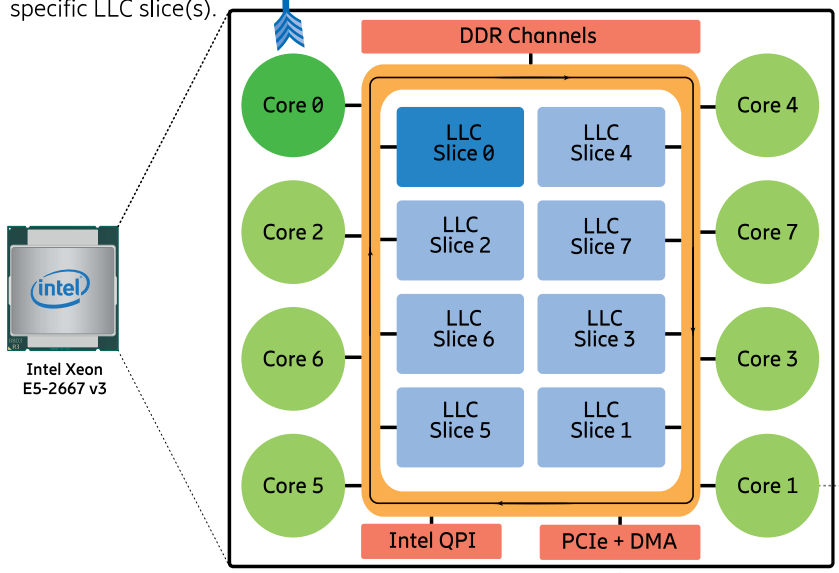
For each core, accessing data stored in a closer slice is faster than accessing data stored in other slices.

We introduce a slice-aware memory management scheme, wherein data can be mapped to specific LLC slice(s).

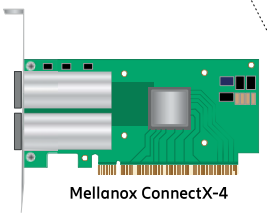


Measuring read access time from Core 0 to different LLC slices.

Accessing the closer LLC Slice can save up to 20 cycles, i.e., 6.25 ns.



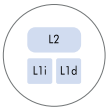
Intel Xeon E5-2667 v3



Mellanox ConnectX-4

Sending/Receiving Packets via DDIO

Data Direct I/O (DDIO) technology sends packets to LLC rather than DRAM.



3 CacheDirector

CacheDirector is a network I/O solution that extends DDIO by implementating slice-aware memory management as an extension to Data Plane Development Kit (DPDK).

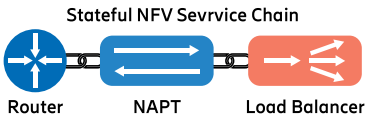
It sends the first 64 B of a packet (containing the packet's header) directly to the appropriate LLC slice.

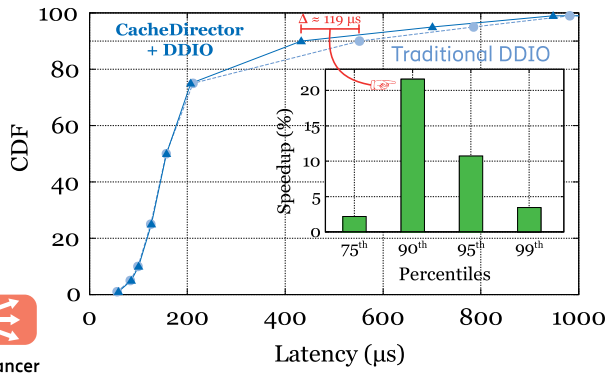
The CPU core that is responsible for processing a packet can access the packet header in fewer CPU cycles.

4 Impact on Tail Latency for NFV Service Chains

We evaluate the performance of NFV service chains in the presence of CacheDirector.

CacheDirector reduces tail latencies by up to 21.5% for optimized NFV service chains that are running at 100 Gbps.





5 Conclusions

With slice-aware memory management scheme, which exploits the latency differences in accessing different LLC slices, it is possible to boost application performance and realize cache isolation.

We proposed CacheDirector, a network I/O solution, which utilizes slice-aware memory management.

CacheDirector increases efficiency and performance, while reducing tail latencies over the state-of-the-art.

Our Paper

Alvireza Farshin+, Amir Roozbeh+*, Gerald Q. Maguire Jr.+, Dejan Kostic+

Ericsson Research * KTH Royal Institute of Technology (EECS/COM) +

CHALMERS

Model Selection for Neural Networks

Anton Johansson, Petter Mostad

Department of Mathematical Sciences

Research interests

Our research is mainly focused on investigating model selection for neural networks. Model selection for neural networks is a broad area in general and encompasses directions such as

- Neural Architecture Search where we find the structure of the network during optimization.
- Understanding which neural network structures are easier to optimize.
- Investigating what set of functions specific neural network structures can model.

These areas are broad, **we specialize in classical statistical model selection techniques applied to neural networks** where we investigate directions such as

- How can we apply Information Criterions to the training and evaluation of neural networks?
- Applying Bayesian model selection techniques.
- Developing specialized cross-validation procedures.

Our research is a mixture of theoretical and applied, thus not solely on algorithmic development that can beat the state of the art.

Bayesian model selection

Find the model with the highest posterior probability. Given a set of candidate models $\mathcal{M} = \{M_1, ..., M_n\}$, choose the model M_k that maximizes

$$P(M_k|\text{data}) = \frac{P(\text{data}|M_k)P(M_k)}{P(\text{data})}.$$

Issues:

- Posterior distribution is not available.
- Have to use Variational Approximations.
- How does this affect the selection of a given model?

Approaches similar to Neural Architecture Search also exists

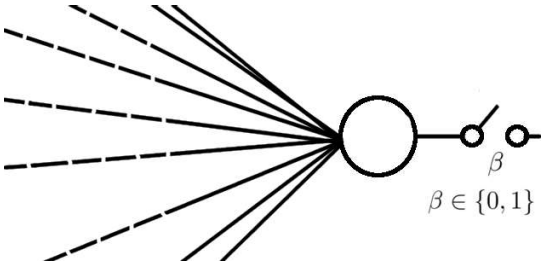


Figure 1: Learn the posterior distribution over β to see if the neuron is important.

Information Criterions

Information Criterions is a way to assess how well a model generalizes without access to an untouched set of data for testing. We choose the model that minimizes some Information Criterion of interest.

The Information Criterion known as **Takeuchi Information Criterion (TIC)** was recently shown to be well-correlated with the generalization of neural network models [1]. It is given as

$$\text{TIC} = -2\log\widehat{L} + \text{tr}\left[J(\widehat{\theta})^{-1}I(\widehat{\theta})\right].$$

Issues:

- Matrix inversion is costly and unstable.
- Only valid close to local optima.

Our research into applying TIC for neural networks:

- Applying it in transfer learning situations.
- More efficient approximations.
- Invert in a subspace.

Issues we have encountered:

- How do we choose the subspace of parameters to update?
- How valid is the criteria when we move around during optimization?
- Good data set to try the method on, each data point should be informative enough.

References

[1]: Valentin, Thomas, Pedregosa, Fabian, Merriënboer, Bart, Mangasol, Pierre-Antoine, Bengio, Yoshua, and Le Roux, Nicolas. Information matrices and generalization. arXiv preprint arXiv:1906.07774, 2019.

Contact information

If you are interested in working with us or want to know more information, do not hesitate in contacting me at

- **Name: Anton Johansson**
- **Email: johaant@chalmers.se**.

Deep Networks for Ordinal Regression

Axel Berg, Mark O'Connor, Magnus Oskarsson
axel.berg@arm.com, mark.oconnor@arm.com, magnus.oskarsson@math.lth.se

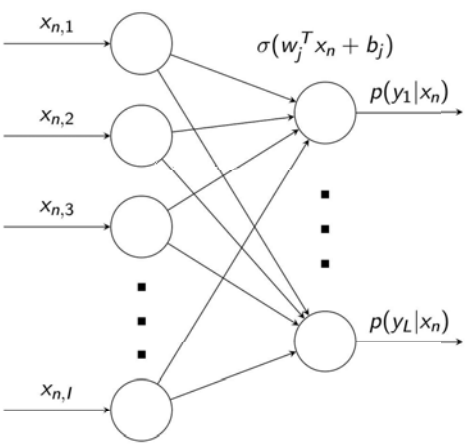
Description

We present a simple technique for turning a standard regression problem into the framework of ordinal regression. This allows us to exploit the advantages of ordinal classifiers, which can be implemented as deep neural networks. Instead of solving a difficult K-class problem, we divide it into $K/2$ binary classification problems by creating overlapping intervals of the ordinal classes. We demonstrate our method on horizon-line estimation and show promising results.

Background & Motivation

Ordinal regression is used for classification problems where classes can be naturally ordered. A classifier h maps each input $x_n \in \mathcal{X}$ into an ordered set $h : \mathcal{X} \to \mathcal{C}$, where $\mathcal{C} = \{c_1 \prec c_2 \prec ... \prec c_K\}$. Recent development has shown that many problems where the predictor is a continuous variable can be successfully treated as an classification problem by discretizing the predictor into disjoint classes. However, since standard classification techniques do not exploit the inherent ordinal structure of such a discretization, a better option is to use some form of ordinal regression technique. Recent work, such as in [1, 2] has shown that this method works well for several computer vision problems. In our research, we try to apply similar methods for the task of horizon line estimation.

Horizon Line Estimation using Deep Ordinal Regression



Deep Ordinal Regression

Given a problem where we want to classify data into ordinal classes $\{c_k\}_{k=1}^K$, we instead create a new set of overlapping intervals $\{y_l\}_{l=1}^L$ by forming the union of nearby classes:

$$y_i = c_{i-M} \cup c_{i-M-1} \cup ... \cup c_{i+M} \quad (1)$$

By such a grouping we exploit the ordinal structure of the classes. The classifier is then trained to maximise the likelihood $p(y_l|x_n)$. At inference time, the likelihood of each original class label can be calculated by marginalising over y_i:

$$p(c_k|x_n) = \sum_{i=1}^{L} p(c_k|y_i, x_n)p(y_i|x_n) \quad (2)$$

In general, we don't have access to $p(c_k|y_l, x_n)$, but we can approximate it by assuming that it does not depend on the input sample. It then becomes natural to assume a uniform distribution over classes:

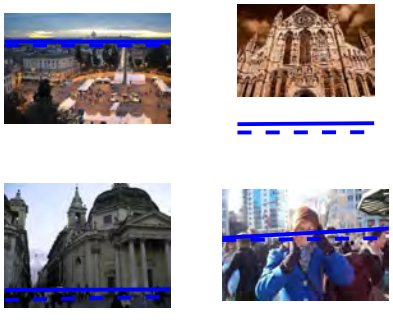
$$p(c_k|y_i, x_n) \approx p(c_k|y_i) = \begin{cases} \frac{1}{2M}, & y_i \cap c_k \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Horizon Line Estimation

We demonstrate our method on horizon line estimation using the dataset from [3]. Each image is labeled with a horizon paramatrised by the perpendicular distance ρ from the image center in pixels and the angle θ. In homogeneous coordinates, the horizon line can then be written as

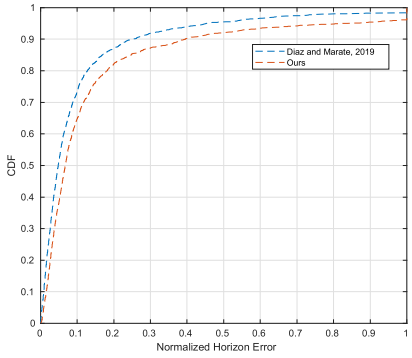
$$\mathbf{h} = \left[\sin\theta, \quad \cos\theta, \quad -\tfrac{W}{2}\sin\theta - \rho\cos\theta\right]^T \quad (4)$$

where W is the image width. Each parameter is then discretized into $K = 100$ ordinal classes $\{c_k\}_{k=1}^K$, such that e.g. $c_k < c_{k+1}$, by interpolating the empirical cumulative distribution such that each class contains equally many training samples. We then group the ordinal classes into binary classes y_i according to equation (1), where we let $M = K/2$, such that each y_i contains 50 % of the training data.
Similarly to the methods described in [1, 3, 4], we train a ResNet50 architecture by replacing the last fully connected layer with two disjoint fully connected layers, one for each parameter. We then use $L = 100$ sigmoid activations and train the classifiers using binary cross-entropy. Evaluation of our method yields near state of the art result on the "Horizon Lines in the Wild" dataset. The figure below shows examples of images with ground truth horizons and estimates in dashed lines.



Further Research

Our method can easily be extended to different number of class combinations M and we plan to do an ablation study to see if one can find a best choice for a given dataset. Furthermore, it is not clear that M should be the same for all y_i, i.e. the class interval width need not be the same for all classes. For regression problems with a continuous predictor, there is a large degree of freedom in how to divide the data into classes, which we will try to exploit in order to improve our results.



We will also look at horizon estimation for sequences of images, e.g. video data, which has previously been done in [5]. If the camera calibration is known, then the camera pitch and roll can be recovered. A robust algorithm for this could potentially be used in localisation and mapping algorithms that normally rely on IMU data. We hope to be able to incorporate our method into a larger SLAM framework and be able to do inference in real time using Arm hardware.

Bibliography

[1] Diaz, Raul and Marathe, Amit. "Soft Labels for Ordinal Regression" 2019
[2] Niu, Zhenxing et al. "Ordinal Regression with Multiple Output CNN for Age Estimation" 2017
[3] Workman, Scott et al. "Horizon Lines in the Wild" 2016
[4] Abbas, Ammar and Zisserman, Andrew. "A Geometric Approach to Obtain a Birds Eye View from an Image" 2019
[5] Kluger, Florian et al. "Temporally Consistent Horizon Lines" 2019

Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving

Carl-Johan Hoel, Katherine Driggs-Campbell, Krister Wolff, Leo Laine, and Mykel J. Kochenderfer

CHALMERS UNIVERSITY OF TECHNOLOGY

Volvo Group Trucks Technology

Abstract

Tactical decision making for autonomous driving is challenging due to the diversity of environments, the uncertainty in the sensor information, and the complex interaction with other road users. This paper introduces a general framework for tactical decision making, which combines the concepts of planning and learning, in the form of Monte Carlo tree search and deep reinforcement learning. The method is based on the AlphaGo Zero algorithm [1], which is extended to a domain with a continuous state space where self-play cannot be used. The framework is applied to two different highway driving cases in a simulated environment and it is shown to perform better than a commonly used baseline method. The strength of combining planning and learning is also illustrated by a comparison to using the Monte Carlo tree search or the neural network policy separately.

Contributions

- The extension of the AlphaGo Zero algorithm, which allows it to be used in the autonomous driving domain.
- The introduction of a general tactical decision making framework for autonomous driving, based on this extended algorithm.
- The performance analysis of the introduced tactical decision making framework, applied to two different test cases.

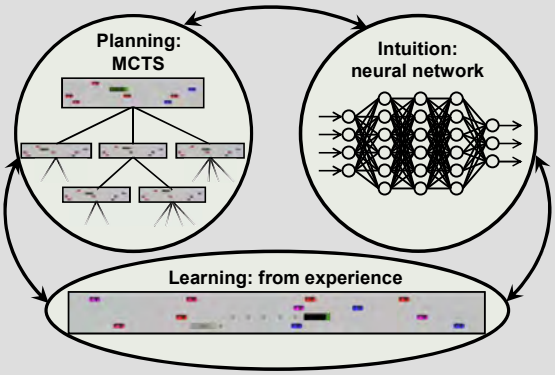


Fig. 1. Overview of the proposed decision making framework, which combines the concepts of planning, intuition, and learning. A Monte Carlo tree search is guided by a neural network, which helps to focus the tree search to the relevant parts of the tree. The network is trained by a reinforcement learning algorithm that learns from experiences in a simulated environment.

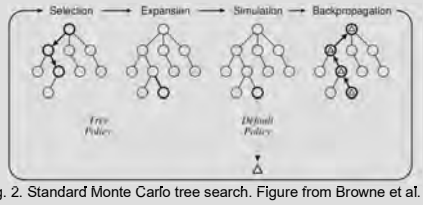


Fig. 2. Standard Monte Carlo tree search. Figure from Browne et al. [2].

Method

A neural network f_θ, with parameters θ, estimates the value $V(s,\theta)$ of a specific state s and prior probabilities $\boldsymbol{p}(s,\theta)$ of taking different actions a,

$$\big(\boldsymbol{p}(s,\theta), V(s,\theta)\big) = f_\theta(s). \tag{1}$$

The value estimates replace rollouts, which are used in standard MCTS, and the prior probabilities guide the MCTS to the relevant regions of the search tree, by using the modified UCB condition

$$UCB(s,a,\theta) = Q(s,a) + c_{\text{puct}} P(s,a,\theta) \frac{\sqrt{\sum_b N(s,b)+1}}{N(s,a)+1}. \tag{2}$$

Furthermore, the MCTS improves the training process of the neural network. An overview of the algorithm is shown in Fig. 1, and for reference, standard MCTS is shown in Fig. 2. A special neural network architecture is used, which makes the ordering and the number of surrounding vehicles irrelevant, see Fig. 3.

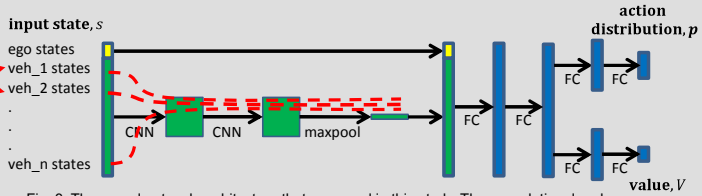


Fig. 3. The neural network architecture that was used in this study. The convolutional and max pooling layers create a translational invariance between the input from different surrounding vehicles, which makes the ordering and the number of vehicles irrelevant.

Test cases

The properties of the presented decision making framework were investigated for two conceptually different highway driving cases, which are illustrated in Fig. 4. The first case involves navigating in traffic as efficiently as possible, whereas the second case involves exiting on an off-ramp.

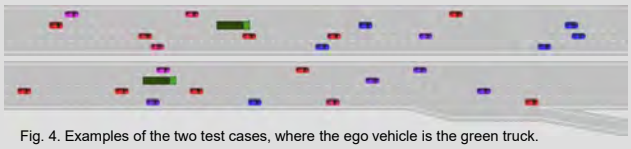


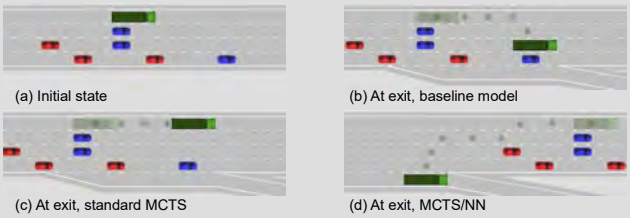
Fig. 4. Examples of the two test cases, where the ego vehicle is the green truck.

Results

The results show that the agent that was obtained by using the proposed framework, referred to as the MCTS/NN agent, outperformed the baseline methods in the two test cases, see Table 1. The performance difference is larger for the more complex highway exit case, since it requires a longer planning horizon, which is exemplified in Fig. 5. The learned values for different states on an empty highway are shown in Fig. 6.

Table. 1. Summary of the results for the different agents.

	Continuous highway case	Highway exit case
	Normalized speed	Success rate
Baseline model	1.00	54%
Standard MCTS	1.03	70%
MCTS/NN	1.05	100%



(a) Initial state (b) At exit, baseline model

(c) At exit, standard MCTS (d) At exit, MCTS/NN

Fig. 5. Example of when it is necessary to plan relatively far into the future to solve a specific situation. The initial state is shown in (a) and the state at the exit is shown for the three agents in (b), (c), and (d). The dots show the position of the ego vehicle relative to the other vehicles during the maneuver, i.e., in (b) and (c) the ego vehicle accelerates and overtakes the slower vehicles, whereas in (d), the ego vehicle slows down and ends up behind the same vehicles.

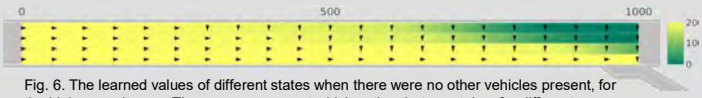


Fig. 6. The learned values of different states when there were no other vehicles present, for the highway exit case. The arrows represent which action that was taken for different states. Note that the axes do not have the same scale.

Conclusions

The results of this paper show that the presented framework, which combines planning and learning, can be used to create a tactical decision making agent for autonomous driving. For two conceptually different highway driving cases, the resulting agent performed better than individually using planning, in the form of MCTS, or learning, in the form of the trained neural network. The agent also outperformed a baseline method, based on the IDM and MOBIL model. The presented framework is flexible and can easily be adapted to other driving environments. It is also sample efficient and required one order of magnitude less training samples than a DQN agent that was applied to a similar case [3].

Acknowledgement

This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems, and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation, and partially by Vinnova FFI.

Contact

carl-johan.hoel@chalmers.se
+46 703 273628
Chalmers University of Technology, 412 96 Göteborg, Sweden

References

[1] D. Silver et al., "Mastering the game of Go without human knowledge," Nature, 2017.

[2] C. Browne et al., "A survey of Monte Carlo tree search methods.," IEEE Comput. Intellig. and AI in Games, 2012.

[3] C. J. Hoel, et al., "Automated speed and lane change decision making using deep reinforcement learning," in IEEE Intelligent Transportation Systems (ITSC), 2018.

Inter-frequency radio signal quality prediction for handover, evaluated in 3GPP LTE

Published in: Proceedings of the 89th IEEE Vehicular Technology Conference, 2019

Caroline Svahn[†,‡], Oleg Sysoev[†], Mirsad Čirkić[‡], Fredrik Gunnarsson[‡], Joel Berglund[‡]
[†]IDA, Linköping University: {firstname.lastname}@liu.se
[‡]Ericsson Research, Linköping: {firstname.lastname}@ericsson.com

Description

In cellular networks, frequent measuring of signal quality on available frequencies would allow for highly reliable networks and optimal connection at all times, however, these measurements are associated with performance costs. To reduce the costs, we consider predictions of inter-frequency radio quality measurements to assess potential inter-frequency handover decisions. We demonstrate that straightforward applications of the most commonly used machine learning models are unable to provide high accuracy predictions. Instead, we propose a novel approach with a duo-threshold for high accuracy decision recommendations. Our approach leads to class specific prediction accuracies as high as 92% and 95%, still drastically reducing the need for inter-frequency measurements.

Background & Motivation

The booming interest in wireless services means that cellular networks are upgraded with service over multiple frequency carriers. This poses a resource management problem of how to assign user connections to different frequency carriers, and the assignments need to be re-evaluated over time as they may be subject to handover [1] from one frequency carrier to another. To maintain relevant frequency assignments, frequent measurements would need to be performed, which would lead to unnecessary load of the network and extensive battery consumption. Signaling overhead can be greatly reduced if an automatic prediction of the inter-frequency signal quality is provided.

Methods & Results

We define observations for which the inter-frequency RSRQ is higher than the serving RSRQ as positive. Consider a three-class output with two thresholds, δ_1 and δ_2:

$$C_d = \begin{cases} 1, & \text{if } p(C=2|\mathbf{x}) \leq \delta_1 \\ 2, & \text{if } p(C=2|\mathbf{x}) \geq \delta_2 \\ 3, & \text{otherwise.} \end{cases}$$

We define the evaluation measures as

$$TNR_d = \frac{TN + N_{mn}}{TN + FP + N_{mn}}$$

$$TPR_d = \frac{TP + N_{mp}}{TP + FN + N_{mp}}.$$

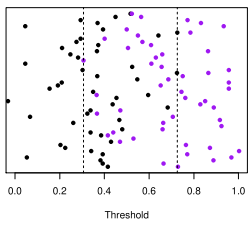


Figure 1: Example of the proposed threshold bonds.

N_{mn} and N_{mp} are the number of observations from class 1 and class 2 which were predicted as class 3, respectively. We also use *measure share* to evaluate our models, which we define as $\frac{N_{mn}+N_{mp}}{N_{tot}}$, where N_{tot} is the total number of predicted observations. We set lower acceptance thresholds as $TPR_d = 0.90$ and $TNR_d = 0.95$, to ensure high quality predictions while considering that a faulty handover decision yield greater loss than to stay on a non-optimal frequency. Results for the two best models in terms of the set lower limits and measure share is presented in Table 1.
Random Forest appears to be superior to the Neural Network as it allows for a higher δ_1 and a lower δ_2, still maintaining a lower measure share.

Table 1: Best thresholds.

Model	δ_1	δ_2	$\mathbf{TPR_d}$	$\mathbf{TNR_d}$	M. share
RF	0.50	0.65	0.919	0.952	0.0563
NN	0.50	0.95	0.985	0.952	0.138

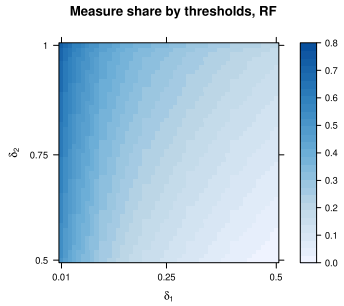


Figure 2: Measure share for Random Forest.

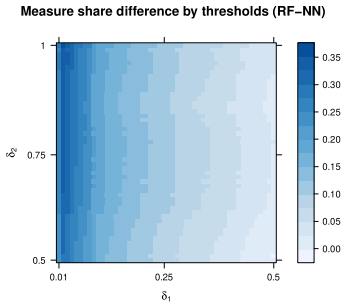


Figure 4: Measure share difference.

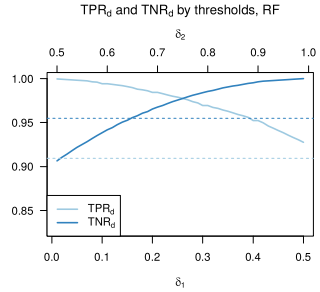


Figure 3: TPR_d and TNR_d for Random Forest.

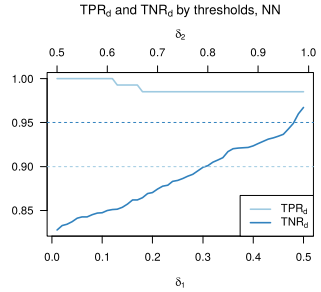


Figure 5: TPR_d and TNR_d for Neural Nets.

Research Goal & Question

Predictions of the inter-frequency radio signal strength (RSRP) has been discussed in [2]. In this paper, we instead consider Reference Signal Receiver Quality (RSRQ) as this is a better measure of the actual network performance. RSRQ depends on the total received radio signal energy from all signals, Radio Signal Strength Indicator (RSSI), according to

$$RSRQ = \frac{R_b \cdot RSRP}{RSSI},$$

where R_b is the number of resource blocks [3]. In this paper, we predict inter-frequency RSRQ using the most commonly used machine learning methods: Random Forest, Neural Networks, Gaussian Processes and Logistic regression. We pose inter-frequency prediction as a classification problem.

Bibliography

[1] Dimou, K., Wang, M., Yang, Y., Kazmi, M., Larmo, A., Pettersson, J., Muller, W. and Timner, Y., Handover within 3GPP LTE: Design Principles and Performance, IEEE 70th Vehicular Technology Conference Fall, 2009.

[2] Rydén, H., Berglund, J., Isaksson, M., Cöster, R. and Gunnarsson, F., Predicting Strongest Cell on Secondary Carrier using Primary Carrier Data, IEEE Wireless Communications and Networking Conference Workshops (WCNCW): 7th International Workshop on Self-Organizing Networks (IWSON), pp. 137–142, 2018.

[3] Evolved Universal Terrestrial Radio Access (E-UTRA),"Physical layer; Measurements" in Technical Specification (TS) 36.214, 3rd Generation Partnership Project (3GPP), 2018. URL: `http://www.3gpp.org/ftp//Specs/archive/36_series/36.214/`.

[4] C.M. Bishop, "Pattern recognition and machine learning", Springer, 12, pp. 138–139, 205–206, 227-228, 304–305, 666, 2006.

[5] R Core Team, "R: A language and environment for statistical computing". R Foundation for Statistical Computing, Vienna, Austria, 2017. URL: `https://www.R-project.org/`https://www.R-project.org/.

[6] Liaw, A. and Wiener, M., "Classification and Regression by randomForest", R News 2(3), 2002, pp. 18–22.

[7] Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. "kernlab - An S4 Package for Kernel Methods in R", Journal of Statistical Software 11(9), 2004, pp. 1–20.

[8] Fritsch, S. and Guenther, F., "neuralnet: Training of Neural Networks", 2016. URL: `https://CRAN.R-project.org/package=neuralnet`

Ultra-Reliable and Low-Latency Wireless Communication for Control

Designing a new wireless protocols

Introduction

Industry 4.0 and autonomous vehicles are two fields which are rapidly advancing. Both areas have in common that they will rely on low-latency and ultra-reliable wireless communication to optimize performance. To put the controller on the cloud, or for safety-critical systems, the round-trip latency must be less than 2 ms, and the bit error rate (BER) must not exceed 10^{-9}. The wireless communication protocols used today cannot meet both these requirements at the same time [1]. They are designed for other purposes than control, e.g., high throughput or low energy consumption.

Method

There exist work to tweak some of these standards, but to really solve the issue at its core a new communications protocol is needed [1-5]. The network is assumed to be fixed and connected in a star or in a daisy chain topology with the controller at the center.

The requirements can be achieved by greatly simplifying or combining the network layer with the MAC layer and eliminating higher levels of the protocol. With smart network coding and using channel prediction, to use the best relays, will reduce the latency and improve the BER.

Preliminary results

At a nominal SNR of 5 dB, simulations achieve a latency < 2 ms with a BER of 10^{-9} using 13 relay. Only 3 relays are needed if channel prediction is applied. In the current work, channel measurements have been performed to verify the theory, and the results looks promising, Figure 1. Next, a multi-link measurement will be performed to test the protocol on real channels.

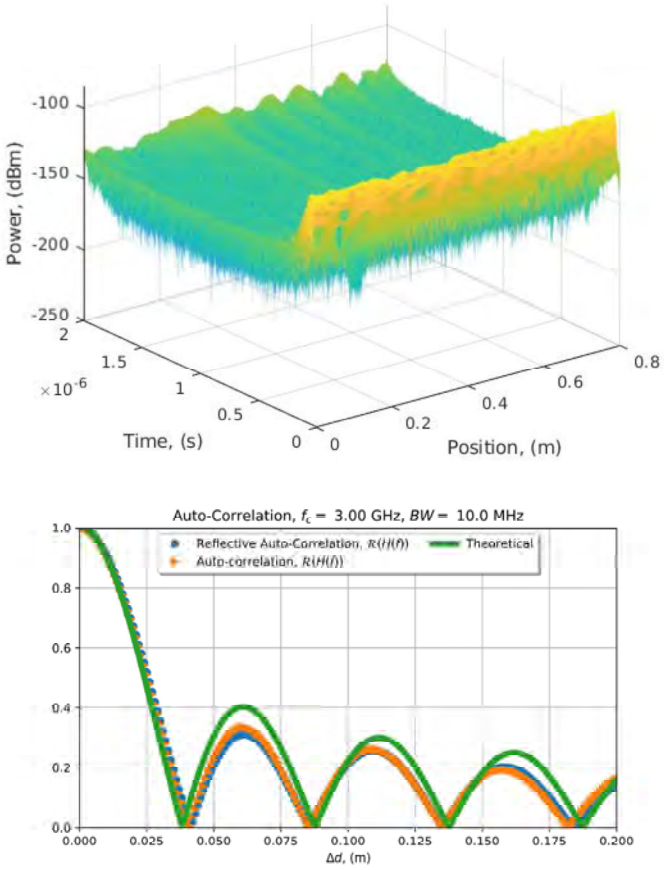


Figure 1. Top: Channel response from an obstructed LOS measurement. Bottom: Corresponding correlation.

[1] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolic, "Design of a low-latency, high-reliability wireless communication system for control applications," in 2014 IEEE International Conference on Communications, ICC 2014, 2014, pp. 3829–3835.

[2] V. N. Swamy, P. Rigge, G. Ranade, B. Nikolic, and A. Sahai, "Predicting Wireless Channels for Ultra-Reliable Low-Latency Communications," in IEEE International Symposium on Information Theory - Proceedings, 2018, vol. 2018-June, pp. 2609–2613.

[3] V. N. Swamy, P. Rigge, G. Ranade, A. Sahai, and B. Nikolic, "Network coding for high-reliability low-latency wireless control," in 2016 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2016, 2016, pp. 138–144.

[4] V. N. Swamy et al., "Real-Time Cooperative Communication for Automation Over Wireless," Ieee Trans. Wirel. Commun., vol. 16, no. 11, pp. 7168–7183, 2017.

[5] V. N. Swamy et al., "Cooperative communication for high-reliability low-latency wireless control," in IEEE International Conference on Communications, 2015, vol. 2015-Septe, pp. 4380–4386.

Christian Nelson | EIT, Faculty of Engineering | Lund University | Sweden

Paul Rigge | Department of Electrical Engineering and Computer Sciences | University of California, Berkeley | USA

AI-supported endpoints in chronic kidney disease

Christos Matsoukas

Description:

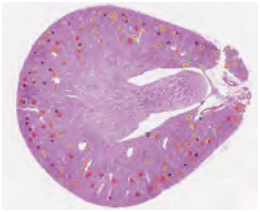
Chronic kidney disease (CKD) is a major global health problem, affecting almost 10% of the population. Two major obstacles in developing new drugs for CKD are the limitations in translatability from animal disease models to patients and the variability between pathologists diagnoses on kidney biopsies, which causes diagnostic uncertainties and makes stratification for clinical trials difficult. Our first aim is to develop AI-assisted scoring algorithms for two commonly used mouse models for CKD. The next step of the project will be translation from animals to human patients.

Background & Motivation

Clinical example:
Human biopsy with annotations

Current AI methods fail to grade the degree of damage that is present in the kidney's filtering units. The main reasons are the limited annotated samples, the complexity of the problem and the extreme variability between pathologists which leads to noisy labels.

Research Goal & Questions



Preclinical example:
Animal's kidney full cross section with annotations

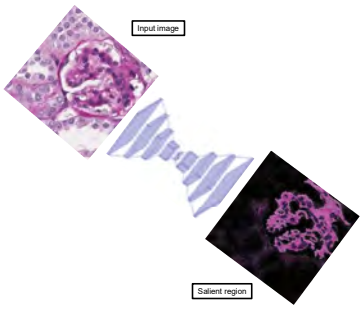
- Is it possible to achieve expert-level performance for glomeruli grading?
- Is translation from animals to human patients feasible?
- Can we use the model's learned features to create new possible biomarkers?

Methods & Preliminary Results

Our first aim is to develop AI-assisted scoring algorithms for two commonly used mouse models for CKD; the BTBR ob/ob diabetic mouse and the ischemia-reperfusion injury model. The networks we develop will be applied for decision making in drug discovery to reduce turnaround time and data variability.

The next step of the project will be translation from animals to human patients. The goal will be to train a network to grade human kidney biopsies with similar types of injury as observed in the animal models, primarily in diabetic nephropathy (DN) but also in glomerulopathies with similar morphological features that are of interest from a drug development perspective. The challenge of incomplete and limited data becomes pronounced here, as a typical human tissue sample contains approximately 10-20 glomeruli and limited number of samples are available

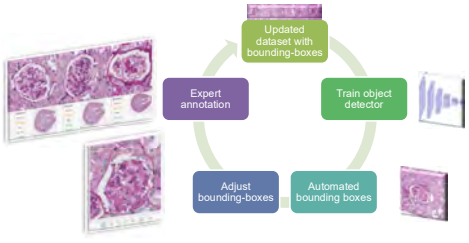
Roadmap & Milestones



- AI-assisted scoring algorithms for animal models.
- Going from the animal models to human biopsies using transfer learning.
- Deploy deep saliency prediction and explainability methods to identify the important regions and features of the kidneys.

Current State

Annotation is a bottleneck to the project due to the currently manual process of glomerular scoring. For the annotation process, a pathologist needs to manually search through the high-resolution images to identify, set a bounding box, and label each individual glomeruli. We overcome this time-consuming procedure by creating a tool that suggests areas of interest to the annotator and a bag of classes.



WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Modeling of Energy Consumption in GPS Receivers for Power Aware Localization Systems

Claudio Mandrioli[1], Alberto Leva[2], Bo Bernhardsson[1], Martina Maggio[1]
Department of Automatic Control, Lund University
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano

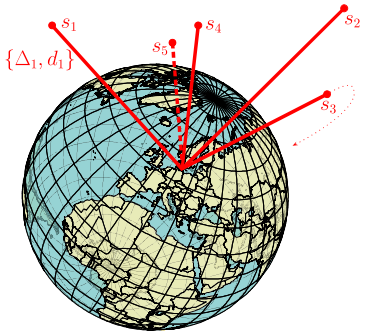
Abstract

This work proposes a first-principle model of GPS receivers, that allows us to exploit the trade-off between battery consumption and positioning accuracy. We present the model and propose a GPS sampling strategy that uses both the current positioning confidence, and information about the GPS status. We complement the GPS sensor with internal measurement units and show how the given model exposes the battery-accuracy trade-off in the context of sensor fusion. We demonstrate the usefulness of the proposed sampling strategy using both simulation and real data.

Motivation

GPS sensors are one of the most battery draining components of small devices. Is important to evaluate of how the receiver affects the needed power in relation to the provided measurements so that energy-accuracy trade-offs are optimized.

GPS Principles



Power consumption model: the sensor will consume power when the GPS antenna is turned on and this power can be reasonably assumed constant in time.

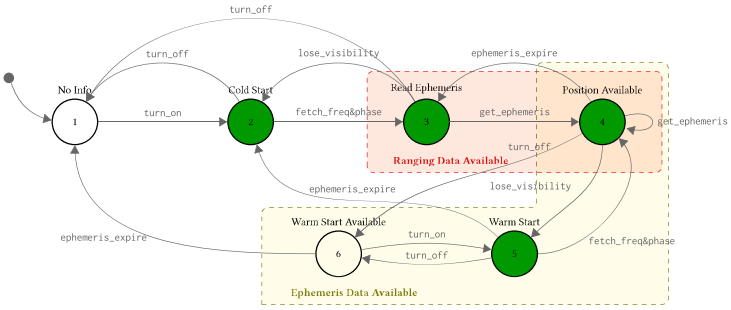
Position availability: two pieces of information required for trilateration

- Ephemeris data - position of the satellites
- Rangind data - distance from the

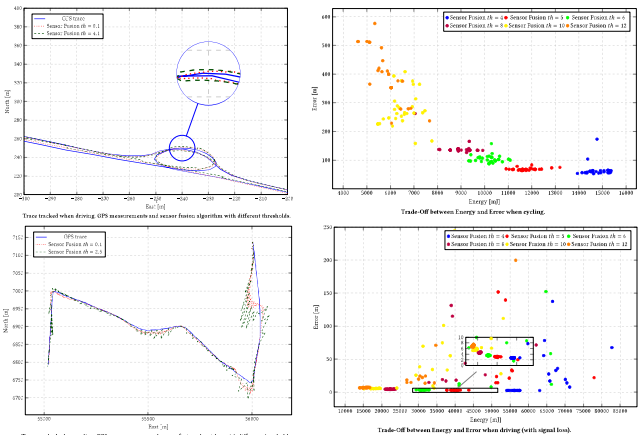
Energy Consumption Model

Need to capture:

- Ephemeris data availability
- Ranging data availability
- Antenna on/off



Evaluation and Conclusions



The proposed model guarantees evaluation of power consumption of GPS receives by construction. It successfully allows positioning accuracy-power consumption trade-offs.

References

Mandrioli, Claudio & Leva, Alberto & Bernhardsson, Bo & Maggio, Martina. (2019). Modeling of energy consumption in GPS receivers for power aware localization systems. 217-226. 10.1145/3302509.3311043.

Verification of Quantitative Multi-Agent Systems

Daniel Ahlsén

Stockholm University
Department of Mathematics

Verification as Model Checking

Formal verification is the act of formally proving that an algorithm satisfies some desired specification. This can be modeled as a logical problem: specifications correspond to formulas in some logical language, while programs corresponds to models of said logic. The problem may be formalized as

Given a formula ϕ and a model $\mathcal{M}$,
determine if $\mathcal{M} \models \phi$?

This decision problem is called **model checking**, and the aim of this project is to study model checking of multi-agent systems with payoffs.

The Mutual Exclusion Problem

Imagine two computers connected to a shared printer. If both tries to use them at the same point, the system become deadlocked. Finding an algorithm which guarantees that the system does not deadlock is called the **mutual exclusion problem**, and solutions can be verified using model checking.

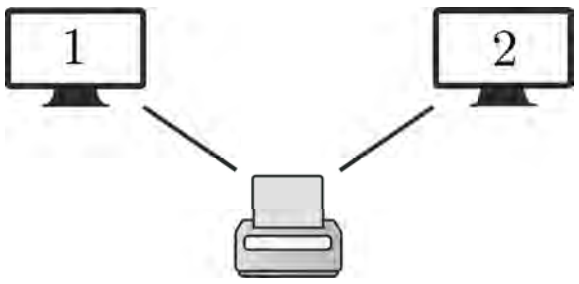


Figure 1: Two computers attached to a printer.

Suppose one adds the specification that the printer should not print a document unless it has enough papers to print the entire document. This specification is quantitative: it is determined by the value of a numerical variable.

Traditional approaches to verification have difficulty handling specifications such as these, since the number of possible configurations is unbounded.

Quantitative ATL*

The first use of temporal logics in verification was by Pnueli, who introduced **LTL**[1]. Clarke and Emerson [2] extended Pnueli's ideas to branching computations and these were unified by Emerson and Halpern[3] in the logic **CTL***.

Alur, Henzinger, and Kupferman extended **CTL*** to multi-agent systems[4], resulting in **ATL***. All these logics have decidable model checking problem and have found some commercial use.

However, they are unable to verify quantitative specifications. Bulling and Goranko [5] introduced **QATL*** to allow for quantitative reasoning in multi-agent system. This project aims to develop procedures for model checking of this logic.

Current Progress

Model checking **QATL*** is undecidable in general, by reduction to the halting problem, but decidability can be restored to subclasses of models or formulas[5].

In **QATL*** numerical variables can take infinitely many values. Since the termination of most algorithms in earlier work relies on the fact that certain sets are finite, the current focus is on finding suitable representations.

References

[1] A. Pnueli, "The temporal logic of programs.," *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, pp. 46–57, 1977.

[2] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching time temporal logic.," in *Workshop on Logics of Programs*, Springer, 1981, pp. 52–71.

[3] E. A. Emerson and J. Y. Halpern, ""sometimes" and "not never" revisited: On branching versus linear time temporal logic," *J. ACM*, vol. 33, no. 1, pp. 151–178, 1986.

[4] R. Alur, T. Henzinger, and O. Kupferman, "Alternating-time temporal logic," *Journal of the ACM*, vol. 49, no. 5, pp. 672–713, Sep. 2002.

[5] N. Bulling and V. Goranko, "Combining quantitative and qualitative reasoning in multi-stage games," *Unpublished manuscript*, 2018.

Deep Learning for Sequential Data

Daniel Gedon
daniel.gedon@it.uu.se

UPPSALA UNIVERSITET

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Overview

Project Team:

- Main Advisor: Thomas Schön (UU)
  → SMC Methods, Deep Learning
- Co-Advisor: Niklas Wahlström (UU)
  → Deep Learning
- Co-Advisor: Lawrence Murray (Uber AI)
  → Probabilistic Programming (Birch, Pyro)
- Project partner: Gizem Çaylak (KTH)
  Title: Deep Probabilistic Programming
  Supervised by David Broman (KTH)

Topic overview:

- Start of PhD studies in August 2019 after MSc in System & Control (TU Delft)
- Leverage the background in system identification to construct new deep learning models for sequence learning.
- We will consider applications in medical area, e.g. in cardiology.
- Connections to probabilistic programming will be explored in combination with Lawrence and Gizem.

Research Questions

A. How to construct deep learning architectures for sequential data

B. How to use deep learning for decision making where uncertainty quantification is crucial

Motivation / Application

- Barely access to medical specialists in remote areas (e.g. amazonian rainforest).
- Automated identification of heart diseases from ECG data.
- Fast and reliable decision making for treatment of patients and life saving.
- Necessity for quantification of uncertainty.
- Availability of large dataset of 2.5 million labeled ECG recordings from Minas Geiras, Brazil. [1]

First Directions

- Bayesian Deep Learning for uncertainty quantification in deep learning models. [2]
- Deep State Space Models as combination of (1) well understood and interpretable model in control theory with (2) flexible state of the are deep learning models. This can be combined in RNNs. [3, 4]
- Approximate inference algorithms such as SMC and variational inference as general purpose approximate inference mechanisms for intractable problems.

Deep State Space Models

Combinations of Deep Learning techniques with classical models:

- Gaussian state transition distribution [5]

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t|\mu_t, \sigma_t)$$

  with μ_t, σ_t as deep neural networks

- Deep RNN with depth in temporal direction. New approach: additional depth in neural operators for state transition. [6]

Goal: Exploit combination of the these probabilistic models with classic system identification and control theory approaches.

Bibliography

[1] Ribeiro, Antonio et al. (2019). "Automatic Diagnosis of the Short-Duration 12-Lead ECG using a Deep Neural Network: the CODE Study".
[2] Gal, Yarin (2016), "Uncertainty in Deep Learning", PhD Thesis, University of Cambridge.
[3] Chung, Junyoung et al. (2015). "A Recurrent Latent Variable Model for Sequential Data", in Advances in Neural Information Processing Systems 28.
[4] M. Fraccaro (2018). "Deep Latent Variable Models for Sequential Data" PhD Thesis, DTU.
[5] Rangapuram, Syama S. et al. (2018). "Deep State Space Models for Time Series Forecasting", in Advances in Neural Information Processing Systems 31.
[6] Pascanu, Razvan (2015). "On Recurrent and Deep Neural Networks", PhD Thesis, Universite de Montreal.

Simultaneous Coalition Structure Generation and Assignment

Fredrik Präntare (fredrik.prantare@liu.se)

Department of Computer Science (AIICS), Linköping University

Background & Motivation

A major challenge in artificial intelligence is to organize and coordinate multiple entities to improve their performance, behavior, and/or capabilities. We develop algorithms for the *simultaneous coalition structure generation and assignment* problem (Fig. 1) to generate better teams and coordination-schemes in complex goal-oriented domains.

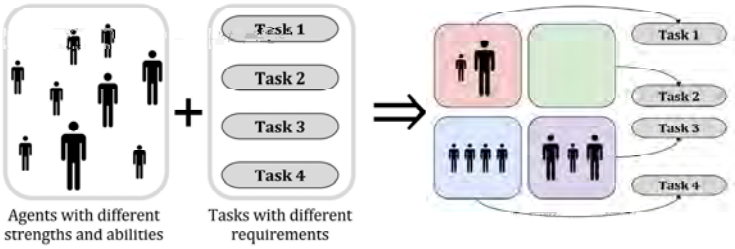


Fig. 1: In the *simultaneous coalition structure generation and assignment* problem, we consider the problem of simultaneously forming coalitions and assigning them to alternatives (e.g., tasks).

Fig. 2: Our algorithms are used in the commercial strategy game *Europa Universalis 4* (a game with more than 1 million players) to coordinate and deploy armies to different regions.

Contributions

Our contributions include developing better algorithms for coalition structure generation, as well as theoretical and practical/empirical advances in game theory and combinatorial optimization. We are also the first to apply a coalition structure generation algorithm in a commercial real-world application (Fig. 2).

Algorithms

- Best-first branch-and-bound [2,3]
  (+ search space representation based on multiset permutations of integer partitions)
- Dynamic programming [4]
- Hybrid (current state-of-the-art) [4]
- Monte Carlo tree search variant [5]
  (+ deep neural networks to guide search)

Publications

[1] Präntare, F., Ragnemalm, I., & Heintz, F. (2017).
Lilla Polhemspriset
An Algorithm for Simultaneous Coalition Structure Generation and Task Assignment. In *International Conference on Principles and Practice of Multi-Agent Systems*.

[2] Präntare, F., & Heintz, F. (2018).
Best Student Paper Award
An Anytime Algorithm for Simultaneous Coalition Structure Generation and Assignment. In *International Conference on Principles and Practice of Multi-Agent Systems*.

[3] *(pending publication)* Präntare, F., & Heintz, F. (2019).
An Anytime Algorithm for Optimal Simultaneous Coalition Structure Generation and Assignment. In *Journal of Autonomous Agents and Multi-agent Systems*.

[4] *(submitted)* Präntare, F., & Heintz, F. (2020).
Dynamic Programming for Optimal Simultaneous Coalition Structure Generation and Assignment. In *International Conference on Autonomous Agents and Multiagent Systems*.

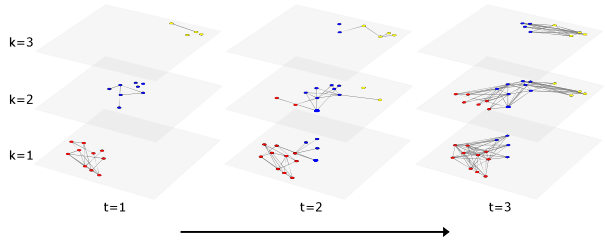
[5] *(work in progress)* Präntare, F., Tiger, M., Bergström, D., Appelgren, H., & Heintz, F. (2020).
Solving Large-Scale Simultaneous Coalition Structure Generation and Assignment Problems with Deep Neural Networks and Monte Carlo Tree Search.

A Bayesian Dynamic Multilayered Block Network Model

Hector Rodriguez-Deniz[1], Mattias Villani[1,2], Augusto Voltes-Dorta[3]
[1]Linköping University - Statistics and Machine Learning (STIMA)
[2]Stockholm University - Statistics
[3]Edinburgh University - Business School

Description

In this paper we introduce a dynamic multilayer block network model with a latent space represention for blocks rather than nodes. A block structure is natural for many real networks, such as social or transportation networks, where community structure naturally arises. A Gibbs sampler based on Pólya-Gamma augmentation is presented for the proposed model. Results from extensive simulations on synthetic data show that the inference algorithm scales well with the size of the network. We present a case study using real data from an airline system, an example of hub-and-spoke network.



Methods & Simulations

We propose the following extension of the dynamic multilayer network model from [1, 2] to impose a latent community structure z to the dynamic multilayer graph $A^k_{ij}(t)$

$$z_i \sim \text{Categorical}(\eta_1, ..., \eta_B)$$

$$A^k_{ij}(t) | (z_i = p, z_j = q) \sim \text{Bernoulli}\left(\pi^k_{pq}(t)\right)$$

$$\text{Logit}\left(\pi^k_{pq}(t)\right) = \begin{cases} \mu(t) + \sum_{r=1}^{R} \bar{x}_{pr}(t)\bar{x}_{qr}(t) + \sum_{h=1}^{H} x^k_{ph}(t)x^k_{qh}(t) & \text{if } p \neq q \\ \mu^k_p(t) + \sum_{r=1}^{R} \bar{x}_{pr}(t) & \text{if } p = q \end{cases}$$

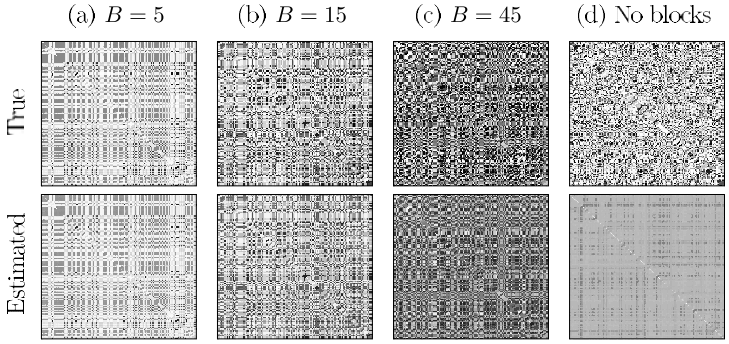
where the latent processes, $\mu(t)$, $\mu^k_p(t)$, $\bar{x}_{pr}(t)$ and $x^k_{ph}(t)$, are assumed to be smoothly evolving Gaussian processes with RBF kernel functions

$$\mu(t) \sim \mathcal{GP}(0, k_\mu) \qquad \mu^k_p(t) \sim \mathcal{GP}(0, k_{\mu_p})$$

$$\bar{x}_{pr}(t) \sim \mathcal{GP}(0, \tau_r^{-1} k_{\bar{x}}) \qquad x^k_{ph}(t) \sim \mathcal{GP}(0, \tau_h^{k^{-1}} k_x)$$

The blockmodeling [4] structures the original likelihood into a reduced set of more informative Binomial components. This likehood allows for exact Bayesian inference using Gibbs sampling with the Pólya-Gamma data augmentation for Binomial logistic regression from [3], while automatically reducing the model size for large network problems as the estimation will be over network communities instead of individual vertices.

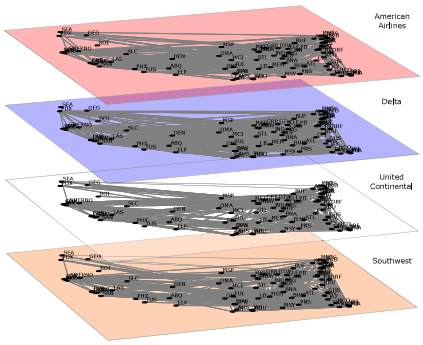
Below be present the true and estimated probabilities on four dynamic multilayer networks with $N = 128$ nodes and different block structure. The four images in the top row show the true probabilities, which appear clearly structured in (a-c) compared to the full-granularity graph in (d). In the bottom row we see how our model is able to almost perfectly recover all probabilities when $B = 5$ and $B = 15$, is doing a decent job when $B = 45$ and, as expected, struggles to fit the data without any block structure.

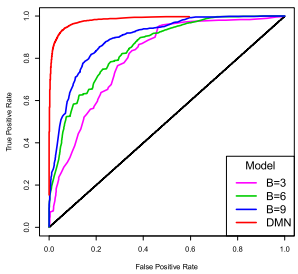


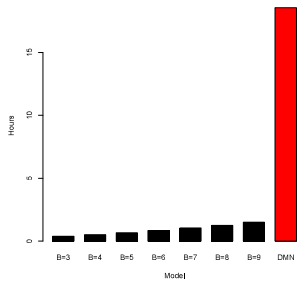
Case Study

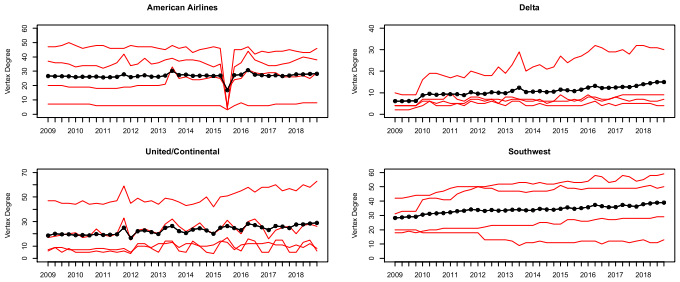
We test our model on real airline network data using airports as nodes and airlines as layers. We collected publically available data to create a multilayer graph with 80 US airports and four airlines over a period of ten years (2009-2018). The dynamic and multilayer dimensions of the network are modeled jointly in a probabilistic fashion, and the stochastic block structure allows for interesting model-based clustering of airports.

We show how the estimated probabilities captured the hub-and-spoke nature of the air transport network, and how our model can project the entire multilayer graph into the future for out-of-sample full network forecasts, which differs from the current practice of visual analysis of static topological indicators. The stochastic blockmodeling allows for a time-series clustering of the airports' connectivity dynamics, and the identification of relevant communitites, while keeping estimation times within reasonable limits.









References

[1] Durante, D., Mukherjee, N., and Steorts, R. (2017). Bayesian learning of dynamic multilayer networks. JMLR 18(43):1-29.
[2] Durante, D., and Dunson, D. (2014). Nonparametric bayes dynamic modelling of relational data. Biometrika 101(4):883-898.
[3] Polson, N., Scott, J., and Windle, J. (2013). Bayesian inference for logistic models using polya-gamma latent variables. JASA 108(504):1339-1349.
[4] Nowicki, K., and Snijders, T. (2001). Estimation and prediction for stochastic block structures. JASA 96(455):1077-1087.

Adaptive Air Combat Training Systems for Competency Based Training

Johan Källström, Saab and Linköping University
Johan.Kallstrom@liu.se

Description:

This project will address the growing need for efficient and effective pilot training solutions for fighter aircraft. The project will study applications of reinforcement learning techniques for generation of intelligent behavior for Computer Generated Forces (CGF) intended to support Live, Virtual and Constructive (LVC) simulation. LVC simulation means linking real aircraft and ground systems (Live), manned simulators (Virtual) and computer controlled entities (Constructive). LVC simulation has the potential to substantially decrease the cost of live training by minimizing the need for real assets and personnel to participate in complex training scenarios.

Background & Motivation

Agent-based simulation is a valuable asset for analysis as well as training. However, building behavior models for intelligent agents is challenging, time-consuming and expensive. Manually defined behavior may also be perceived as unsophisticated and lacking of variation. It would therefore be valuable if agents could learn and adapt to their environment.

Research Goal & Questions

In this project we aim to develop techniques that allow agents to learn how to achieve multiple, possibly conflicting objectives in complex, partially observable environments. Cooperative as well as competitive scenarios will be studied. The goal of the project is to develop diverse and adaptive intelligent agents that can stimulate humans in training.

We will try to answer the following research questions:

1. How can agents learn cooperative and competitive strategies in complex environments?
2. How can scenario contents and the behavior of agents be adapted to the training needs of a specific student?
3. How can the behavior of agents be explained to humans?

Methods & Preliminary Results

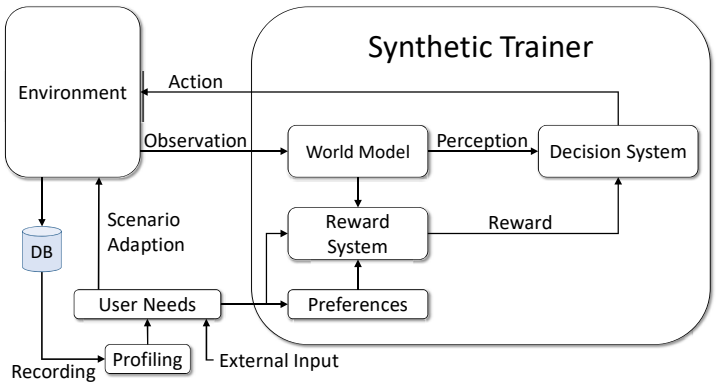
Research methods:

1. Interviews to better understand user needs
2. Initial concept evaluation and algorithm development in simple and fast simulation environments
3. Further development of interesting concepts in high fidelity simulators
4. Evaluation of human-agent interaction in networked simulators

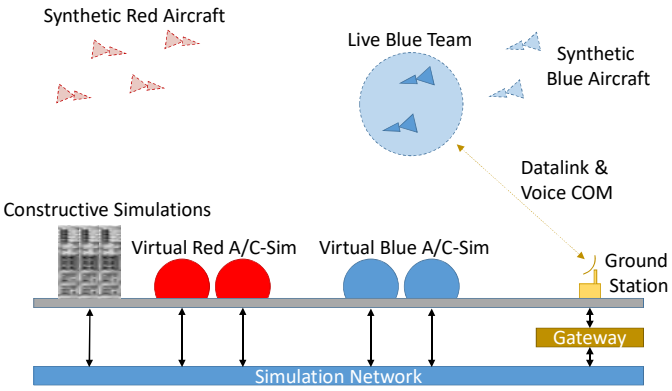
Preliminary results:

1. Survey of related work in reinforcement learning and simulation-based training
2. Development of experiment framework and evaluation of state of the art algorithms in the target system [1]
3. Initial concept development for adaptive agents [2]
4. Development and initial evaluation of an architecture for an adaptive training system [3]
5. Initial user interviews and training goal identification [4]

System Architecture



Distributed simulation system



References

[1] Källström, J., & Heintz, F. (2019). Reinforcement Learning for Computer Generated Forces using Open-Source Software. In Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC 2019).

[2] Källström, J., & Heintz, F. (2019). Tunable Dynamics in Agent-Based Simulation using Multi-Objective Reinforcement Learning. In Proceedings of the Adaptive and Learning Agents Workshop (ALA-19) at AAMAS.

[3] Källström, J., & Heintz, F. (2019). Multi-Agent Multi-Objective Deep Reinforcement Learning for Efficient and Effective Pilot Training. In Proceedings of the Aerospace Congress 2019 (FT2019).

[4] Källström, J., Granlund R., & Heintz, F. (2020). Design of Simulation-Based Pilot Training Systems using Machine Learning Agents. Accepted for Presentation at the 32nd Congress of the International Council of the Aeronautical Sciences (ICAS 2020).

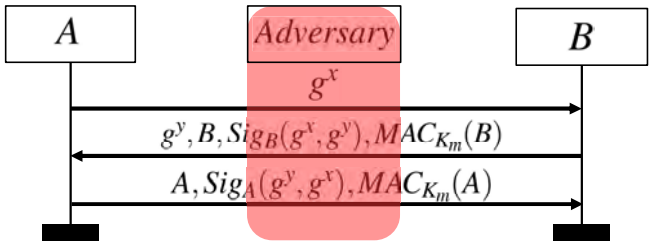
LINKÖPING UNIVERSITY · SAAB · VINNOVA Sweden's Innovation Agency · WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Cryptographic Protocol Verification

Karl Norrman
Ericsson Research and KTH Royal Institute of Technology
Theoretical Computer Science Department

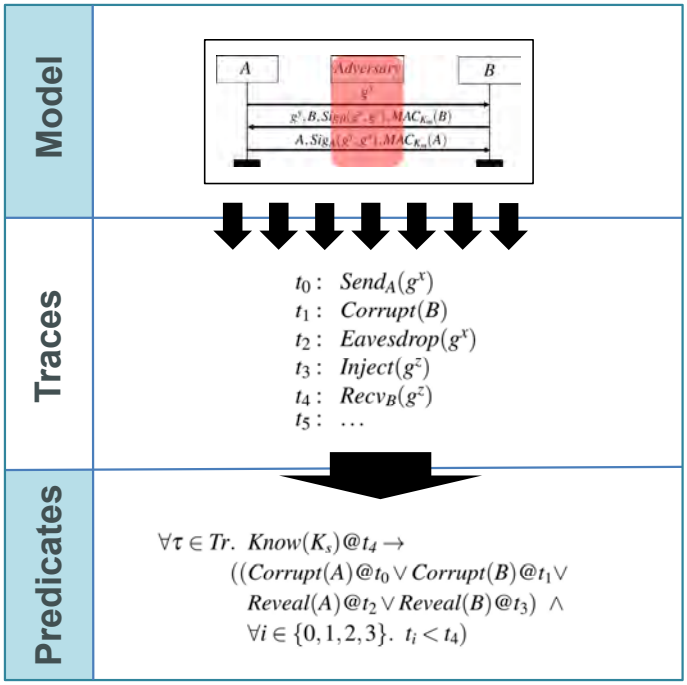
Motivation and Research goals

The research project will advance the methods and tools for modeling and analyzing security of distributed systems to bring the techniques closer to industrial use in order to increase safety and security of cloud, distributed control system and IoT products and services developed by the industry and used by society at large.



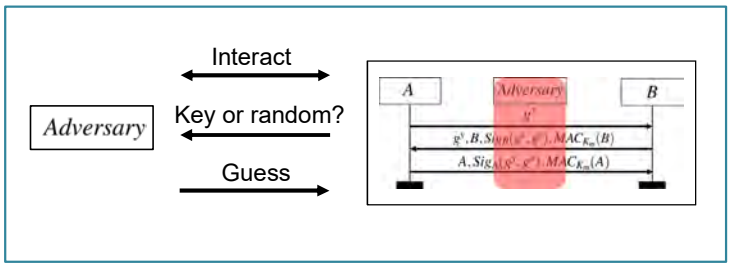
Formal methods
(symbolic model)

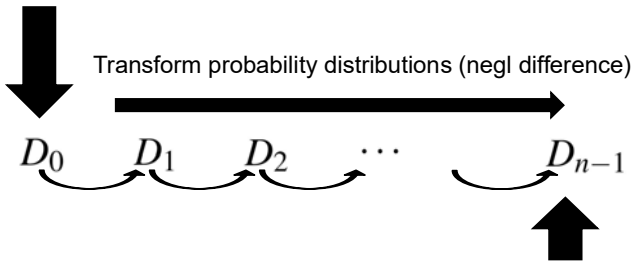
- Modelling:
  - Term-rewriting systems, Π-calculus
  - Equational reasoning
  - First order temporal logics for properties
  - Cryptographic primitives are black-box functions
- "Easy" to mechanize proof search: Tamarin, ProVerif, …
  - Induction and model checking



Reduction based proofs
(Computational model)

- Modelling:
  - Game based definitions as experiments
    - E.g., can adversary distinguish a key from a random string?
  - Show that adversary has 1/2 + negl probability of winning
- Typically pen-and-paper proofs, much more informal
- Typically more expressive than symbolic models
- Proofs involve reductions
- Hard to mechanize proof construction support: EasyCrypt, CryptHOL, …
- Harder to mechanize proof search: CryptoVerif





Obviously Pr[Adversary wins] = 1/2

Current activities

- **Case study: Pen-and-paper proofs in computational model for IETF draft key establishment protocol standard (EDHOC)**
- **Case study: Formal verification of EDHOC in Tamarin**

Method

1. Apply state-of-the-art methods and tools to industrial use cases
2. Identify problems with existing methods and tools
3. Extend and expand as needed and as possible

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Positioning in an Underground Mine

Kristin Nielsen
Gustaf Hendeby, Robert Lundh, Fredrik Gustafsson
kristin.nielsen@epiroc.com

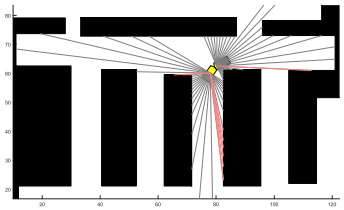
Abstract

A mid-articulated underground loader from Epiroc. (Asset: Epiroc)

Human-Robot Cohabitation Allowing autonomous vehicles interact with unprotected humans requires a real-time positioning system with high accuracy.

Positioning with Lidar Lidar data is compared to a predefined map of the operation area of the vehicle.

Sensor Selection Strategies are proposed allowing for using only a subset of the available measurements with maintained accuracy at lower computational complexity



Lidar measurements are compared to a map to localize an underground loader.

Background & Motivation

New Higher Demands Underground mines are forces to go deeper. Automation is a key enabler to keep productivity and to guarantee personnel safety.

Competing Techniques Dead reckoning, WiFi network, RFID and *ultra wideband* (UWB) based methods are often put forward as possible technologies for indoor positioning.

An autonomous Epiroc loader equipped with two lidars, an odometer and an IMU. (Asset: Epiroc)

Lidar mounted to be protected from dust and falling rocks. (Asset: Epiroc)

Existing Solutions Semi-autonomous loaders equipped with odometers, *inertial measurement units* (IMUs) and lidars, are commercially available from Sandvik, Caterpillar, and Epiroc.

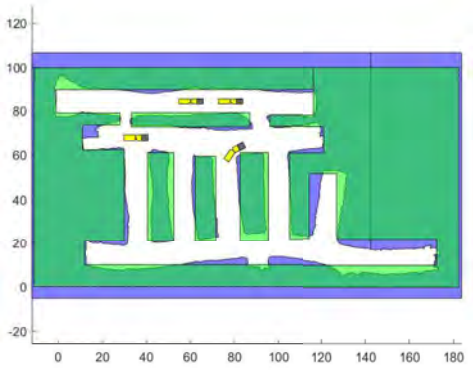
Research Goal & Question

An important aspect of the research is to find algorithms for localization that can be run in real-time on rugged hardware limited in its computational power.

1. What limits the accuracy of position estimation in an underground mine?

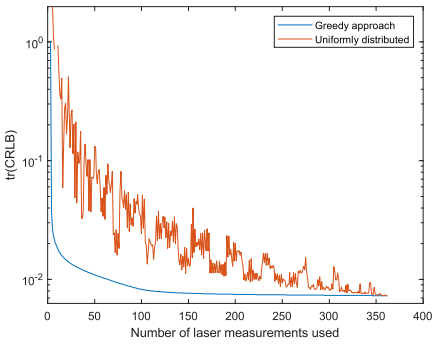
2. Is it possible to improve the position estimate without adding new external infrastructure or new sensors?

3. How are errors in a provided map affecting the resulting position estimate?

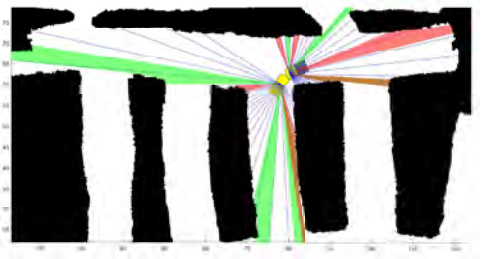


Different accuracies of the map will affect the performance of the position estimate.

Methods & Preliminary Results



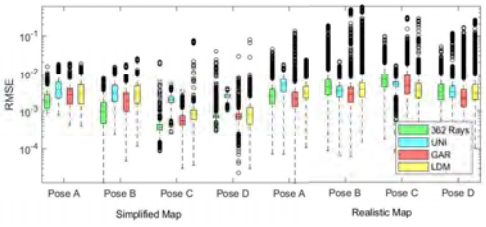
The CRLB decays faster when adding laser measurements for processing, if a greedy optimal selecting strategy is used compared to uniformly distribute the processed laser rays.



Three different measurement selection strategies that are proposed and evaluated.

Sensor Selection in Lidar Data

- Laser ray measurements are cleverly selected to maximize the Fisher information, a measure on information density.
- Simulations show the benefit of suggested selection strategies compared to commonly used methods



The error in the position estimate varies when different selection strategies are applied.

Roadmap & Milestones

- Conference paper covering the sensor selection strategies is in the final preparation stage for submission to *23rd International Conference on Information Fusion*, Sun City, South Africa.

- Developed a baseline solution and investigated properties of the approach of using laser ray measurements one-by-one.

Future work

- Feature extraction in lidar data
- Outlier rejection
- Auto-tuning of hyperparameters in the filter
- Handle dynamic maps

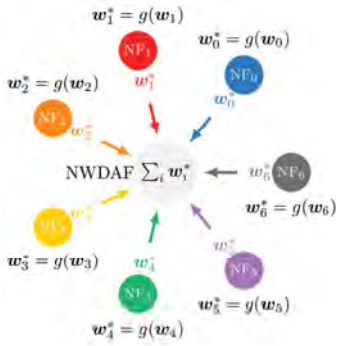
Secure Federated Learning in 5G Mobile Networks

Martin Isaksson[1,2], Karl Norrman[1,2]

[1]Ericsson Research [2]KTH Royal Institute of Technology

Abstract

Machine Learning is an important enabler for optimizing, securing and managing mobile networks. This leads to increased collection and processing of data from network functions, which in turn may leak sensitive information about end-users. Consequently, mechanisms to protect the privacy of end-users are needed. We seamlessly integrate Federated Learning into the 3GPP 5G Network Data Analytics framework, and add a Multi-party Computation protocol for protecting the confidentiality of local updates. We also outline how our work can be fitted into the WARA-PS research arena.



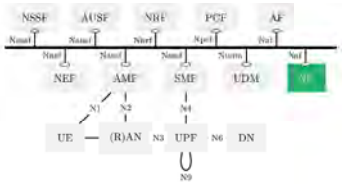
Introduction

Privacy and security challenges

In traditional Machine Learning, data is collected and sent to a central location where a Machine Learning model is trained. This data may contain sensitive information. Collecting it in a single point increases the risk of that point being attacked.

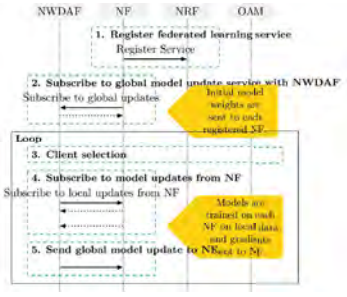
In Federated Learning [1] we don't send the raw data to a central location, and we get some privacy benefits from this. However, since we share trained models we are giving up some information.

Service-based architecture



5G Network architecture. The two top-rows show functions of the core network, which are defined in the SBA framework [2]. Reference point names are shown on the lines between NFs.

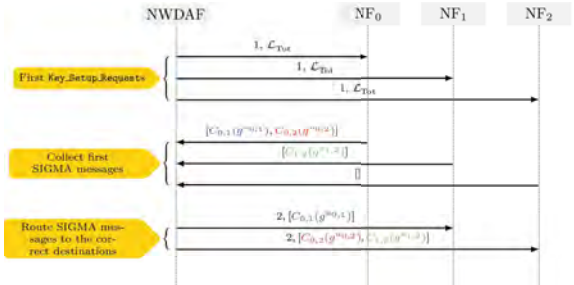
Federated Learning



5G Network Data Analytics

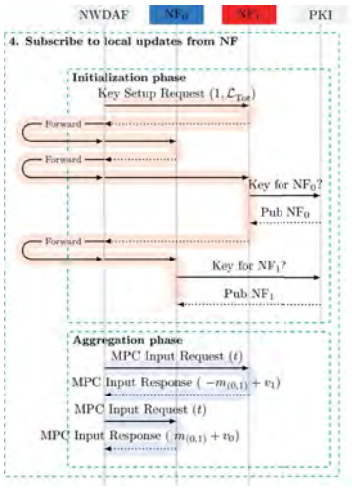
The 5G Network Data Analytics framework is built on SBA principles. It enables service consumers to request, or subscribe to, events occurring at service producing functions. The framework also contains a Network Data Analytics Function. This function collects data and offers predictions and analytics services to service consumers in SBA. Service consumers then use this information to optimize network efficiency and reduce resource consumption.

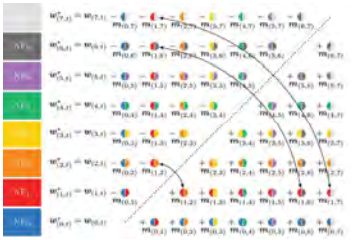
Session Initialization



Tunneling of containers carrying SIGMA messages via the NWDAF. The figure illustrates the first session establishment steps, where the initial SIGMA messages are tunneled to the correct destination in batch via the NWDAF.

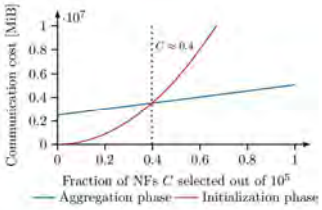
Multi-party Computation

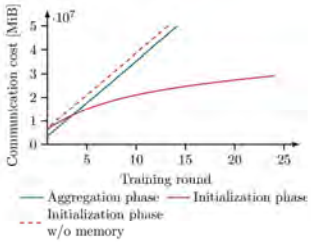




The 8 selected NFs each add masks for every other NF. The cancellation of masks are indicated by the arrows. Each mask above the diagonal is canceled out by a mask below the diagonal. We note that these additions and cancellations of masks do not affect the aggregated sum.

Results





Our contributions

1. An integration of collaborative learning (Federated learning) into the 5G NWDA framework;
2. A privacy-enhancing, efficient, protocol for collaborative learning algorithms in the 5G NWDA based on [3];
3. An evaluation of said protocol with respect to communication cost, storage cost and computational cost.

WARA-PS

Even though we designed our protection scheme in the context of 5G, the general structure can be re-used in other systems. We outline how our work can be fitted into use cases in the WARA-PS [4] research arena. For example, where object-detection models are trained locally in UAVs.

Acknowledgment

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th international conference on artificial intelligence and statistics*, 2017, pp. 1273–1282.

[2] 3GPP, "System architecture for the 5G System (5GS)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, Apr. 2019.

[3] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 acm sigsac conference on computer and communications security*, 2017, pp. 1175–1191.

[4] "WARA-PS (Public Safety)," *Wallenberg AI, Autonomous Systems and Software Program*. WASP, 2018.

Uncertain Constrained Networked Control Systems

Masoud Bahraini
masoudb@chalmers.se

Description

Networked control systems have attracted a lot of attention due to their merits such as cost reduction by removing the wiring and ease of installation, fault detection, and future expansions. However, networked systems introduce new control challenges stemming from network imperfections such as **bandwidth limitation** and **packet losses**. In order to address these issues, we formulate a communication **scheduling** problem for the network. We draw equivalence of the formulated scheduling problem with a well-known scheduling problem, i.e., the **Pinwheel Problem**. The designed schedule assures **robust constraints satisfaction** for the systems of the network. Furthermore, we extend the designed schedule to **cope with packet losses**.

Problem Formulation

Consider a networked control system with disjoint systems where system i is described as

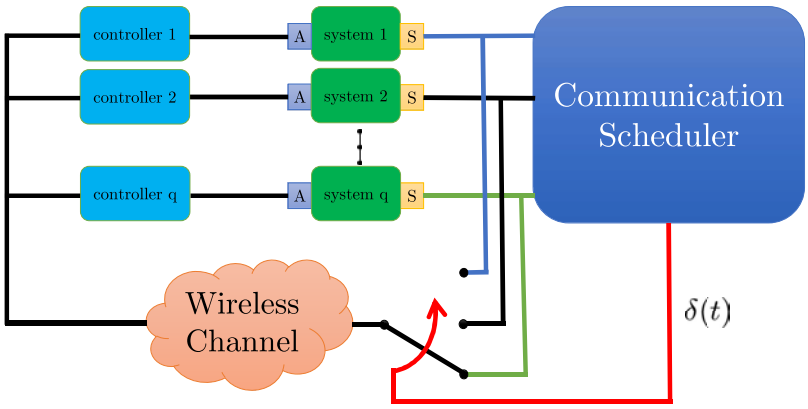
$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + F_i v_i(t),$$
$$\text{s.t. } x_i(t) \in \mathcal{X}_i,\ u_i(t) \in \mathcal{U}_i,\ v_i(t) \in \mathcal{V}_i,$$

and the estimated state $\hat{x}_i(t)$ and control input $u_i(t)$ are considered as

$$\hat{x}_i(t) := \begin{cases} x_i(t), & \text{if } \delta(t) = i \\ A_i \hat{x}_i(t-1) + B_i u_i(t-1), & \text{if } \delta(t) \neq i \end{cases},$$

$$\hat{x}_i(0) = x_i(0),\ u_i(t) = -K_i \hat{x}_i(t).$$

Consider the following network in which only one system can receive measurement updates at a time. The scheduling problem is to **design** $\delta(t)$ **such that constraints satisfaction for all systems of the network is guaranteed** despite uncertainties.



Design Steps

Case 1: No packet loss

- We define the **safe time interval** α_i that is the number of time instants during which a system remains in its **maximal robust invariant set** $\mathcal{S}_{i,\infty}$ without receiving any measurement updates.
- A schedule which guarantees **robust invariance** for all systems can be designed based on the instance $\{\alpha_1, \ldots, \alpha_q\}$ of the Pinwheel Problem.

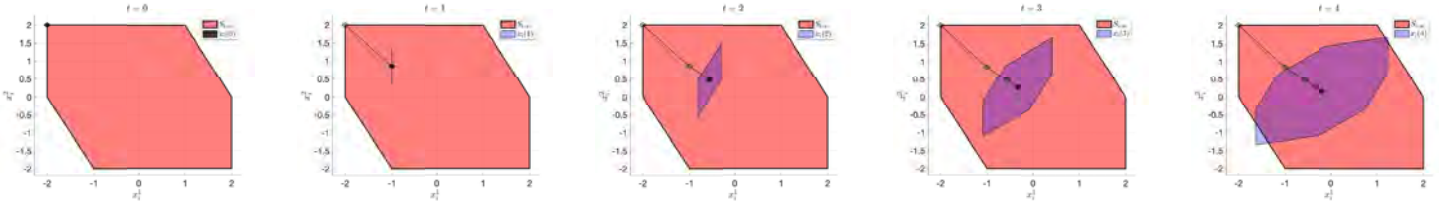
Case 2: Lossy Communication

- We assume only n_i packets can be lost during α_i consecutive time instants.
- In order to guarantee robust invariance in a lossy network, we design a schedule based on instance $\{\alpha_1 - n_1, \ldots, \alpha_q - n_q\}$ of the Pinwheel Problem.
- When a packet is lost, we **reconnect** the same system at the next time instant by shifting the schedule for one step.

One can also designed an online schedule that maximizes a safety index, i.e., the **measurement update deadlines**.

An Example of the Safe Time Interval Calculation

Robust invariant set $\mathcal{S}_{i,\infty}$ and the reachable sets for the state. In this case, the safe time interval is $\alpha_i = 3$!



References

[1] Colombo, Alessandro, Masoud Bahraini, and Paolo Falcone. "Measurement Scheduling for Control Invariance in Networked Control Systems." 2018 IEEE Conference on Decision and Control (CDC). IEEE, 2018.

[2] Bahraini, Masoud, et al. "Receding-horizon robust online communication scheduling for constrained networked control systems." 2019 18th European Control Conference (ECC). IEEE, 2019.

Characterizing the Trust Landscape of Certificate Transparency Logs

Nikita Korzhitskii, Niklas Carlsson (supervisor), Linköping University

Department of Computer and Information Science

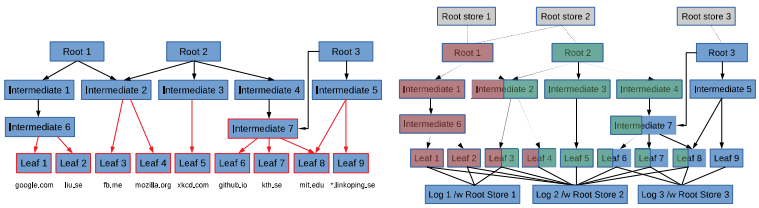
LINKÖPING UNIVERSITY

Abstract

We present the first characterization of the emerging Certificate Transparency (CT) root store landscape [1], as well as the tool [2] that we developed for data collection, visualization, and analysis of the root stores of the available CT logs. We compare the root stores and quantify their changes with respect to both each other and the root stores of major software vendors, look at evolving vendor CT policies, and conclude that root store mismanagement may be linked to log misbehavior.

Introduction

Internet security and privacy stand on the trustworthiness of public certificates signed by Certificate Authorities (CAs). However, software products do not trust the same CAs and therefore maintain different root stores, each typically containing hundreds of trusted roots capable of issuing "trusted" certificates for any domain. Incidents with misissued certificates motivated Google to implement and enforce Certificate Transparency (CT). CT logs archive certificate chains in a public, auditable and append-only manner. The adoption of CT considerably changed the trust landscape, as the logs started to maintain their own distinct lists of trusted CAs and only accept certificates that chain back to one of their trusted roots.

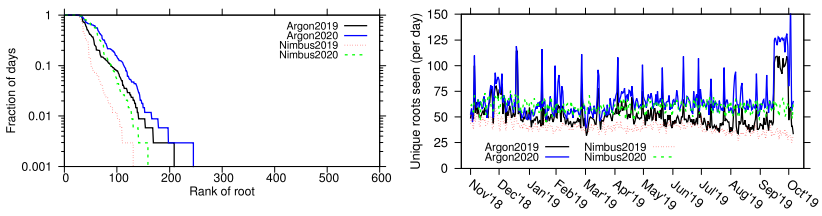


(a) WebPKI (b) WebPKI and CT

Example of a WebPKI hierarchy with and without Certificate Transparency

Methodology

We analyzed root stores of CT logs relative to the stores of Apple, Microsoft and Mozilla; we monitored and summarized the properties of 57 available logs from Google's and Apple's log programs, obtained daily number of roots included in logged certificate chains and measured how the root stores change over time. To directly probe the status of the available logs, we have performed a number of test submissions, and by calculating frequencies of acceptable root certificates in logs, we have found a number of roots that are trusted by just one or two log operators. Finally, we surveyed all log operators trusted by Apple and Google (Cloudflare, DigiCert, Google, Let's Encrypt and Sectigo) on their log management policies and our findings.



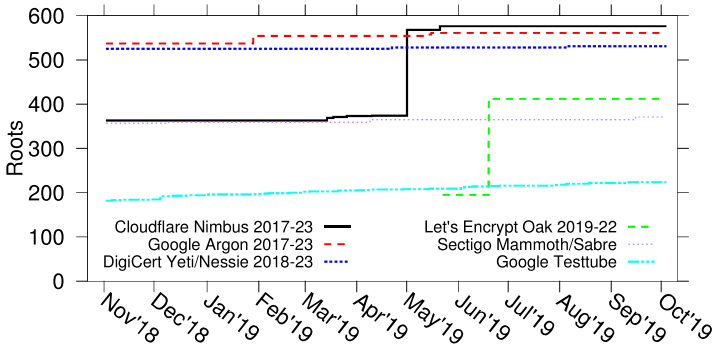
Root usage at CT logs Google Argon and Cloudflare Nimbus 2019/2020

References

1. Korzhitskii, N. and Carlsson, N., Characterizing the Trust Landscape of Certificate Transparency Logs, 2020 (submitted)
2. https://nikita-kun.github.io/certificate-transparency-root-explorer/

Results

The landscape of CT is evolving: software vendors introduce and change their CT policies, new logs are established regularly, while the recent introduction of Cloudflare Cirrus extended the use of CT to ResourcePKI. Over our measurement campaign root stores have became larger and increased their coverage of the roots used by the major vendors.

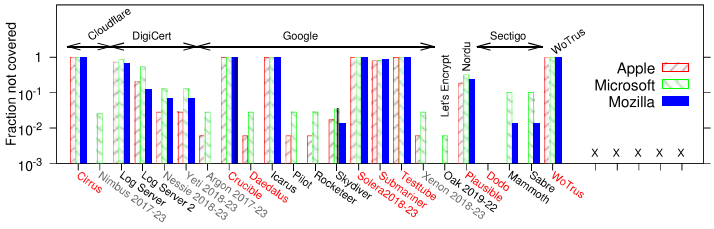


Root store sizes of Certificate Transparency logs over time

Surprisingly, the majority of Cloudflare logs announce trust in a compromised DigiNotar root certificate. Multiple logs are considered Apple/Google-trusted despite violating corresponding policies. We have discovered that all CT logs (except Google's and Let's Encrypt's) do not specify cross-origin headers in their HTTP responses, which obstructs access to the logs using JavaScript in modern browsers.

Overall, we have observed that some WebPKI roots trusted by major software vendors are not sufficiently covered by the CT logs; Apple and Google rely on the CT backbone that is comprised of just five actively logging operators: Cloudflare, DigiCert, Google, Let's Encrypt and Sectigo. Moreover, we have found logs with duplicates in their root lists, and that an accident with GDCA Log 1 and outages of Sectigo Mammoth coincide with anomalous presentation of their root stores.

Finally, the paper presents the results of the survey of the operators and a number of examples and arguments to suggest that management of CT policies and logs' root stores must be performed in a more careful, timely and transparent manner.



Fraction of roots from vendor root stores that is not covered by each log

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Smart modules: automatic selection of collection classes

Noric Couderc, Lund University, Sweden

Which collections should my program use?

Consider the following code snippet:

```
public List<Integer> insertMany(List<Integer> l) {
    for (int i = 0; i < 10000; i++) {
        l.add(0, i);
    }
    return l;
}
```

This code snippet works for any Lists, however, the performance of this code snippet will change depending of the List used: ArrayList, LinkedList and Vector will return the same result, but the performance will change.

A more generic approach to choosing collections

There are some existing solutions for this problem:

Method	Language	Type	Data used	Model
CoCo	Java	Dynamic	Counters on method calls	User-defined thresholds
CollectionSwitch	Java	Hybrid	Counters on method calls	Learned thresholds
Brainy	C++	Static	Hardware and software performance counters	Neural networks

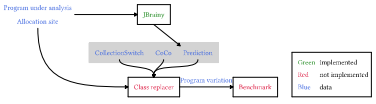
More abstractly:

Method	Advantages	Drawbacks
Static	No run-time overhead	Dynamic information is not available
	Model can be as complex as needed	Maybe you need to change gears
Dynamic	Run-time information is available	Monitoring and decision-making consume resources

We need to **compare** these options.

Experimental design

We need to describe the experimental design of the project. The program under analysis and the allocation site are provided to JBrainy. JBrainy outputs a prediction, which is then compared with other solutions, like CollectionSwitch and CoCo.

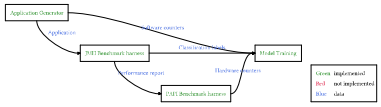


JBrainy

JBrainy is a Java re-implementation of the Brainy project, in Java.

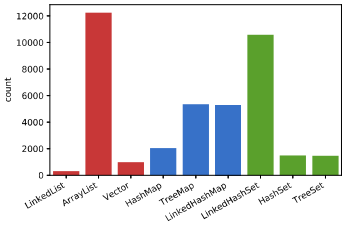
It works in four steps:

1. Generate synthetic benchmarks to use as a training set
2. Benchmark them to get record labels (fastest data structure)
3. Re-run synthetic benchmarks to get training features (hardware performance counters)
4. Train a classifier (in this case a random forest)



Distribution of winners for generated data structures

After benchmarking, we can check which data structures are the fastest in most cases.

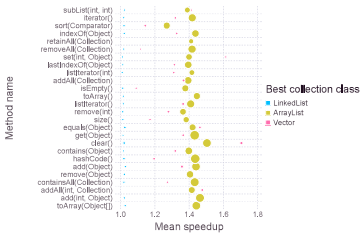


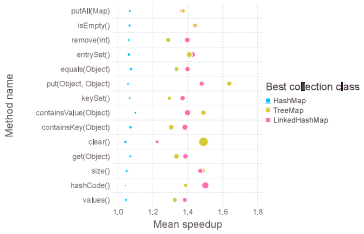
This data shows that when in doubt, if you need a List, you should probably use an ArrayList! If you need a Set, using a LinkedHashSet is probably the right choice. For Maps, the situation is more complicated.

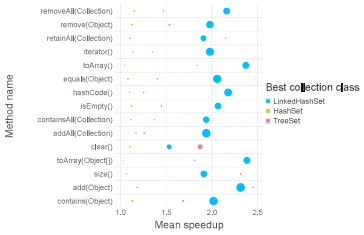
How does this relates to method calls?

- Seed of the random number generator: between 0 and 500
- Number of method calls during the run: 10, 100, and 1000 calls.
- Size of collection before starting the benchmark: 0, 1000, and 10000 elements.
- Collection class to use
  - For lists: ArrayList, LinkedList, Vector.
  - For maps: HashMap, TreeMap, LinkedHashMap.
  - For sets: HashSet, TreeSet, LinkedHashSet.

The total number of samples is 13500.







How do I read these plots?

1. Pick the most common method used on your collection, pick the line with this method.
2. A bigger bubble means that the collection is *often* the best.
3. A bubble to the right means that it is the best *by a large margin* compared to other options.

Anything surprising?

- LinkedHashMap seems to be a really good choice, while 82% of maps used in the wild are HashMaps [1].
- LinkedHashSet seems to be a really good choice too, while 66% of sets used in the wild are HashSets [1].

Fantastic! How can I help?

Remaining questions:

1. Is a really smart model really worth it?
2. Is it better to switch dynamically or have static suggestions?
3. How to make real-world benchmarking easier?
4. How to handle concurrency?

References

[1] Diego Costa, Artur Andrzejak, Janos Seboek, and David Lo. 2017. *Empirical Study of Usage and Performance of Java Collections*. In Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering - ICPE '17. ACM Press, L'Aquila, Italy, 389–400. https://doi.org/10.1145/3030207.3030221

[2] Changhee Jung, Silvius Rus, Brian P. Railing, Nathan Clark, and Santosh Pande. 2011. *Brainy: Effective Selection of Data Structures*. SIGPLAN Not. 46, 6 (June 2011), 86–97. https://doi.org/10.1145/1993316.1993509

[3] G. Xu, *CoCo: Sound and Adaptive Replacement of Java Collections*, in ECOOP 2013 – Object-Oriented Programming, Berlin, Heidelberg, 2013, pp. 1–26.

[4] D. Costa and A. Andrzejak, *CollectionSwitch: a framework for efficient and dynamic collection selection*, in Proceedings of the 2018 International Symposium on Code Generation and Optimization - CGO 2018, Vienna, Austria, 2018, pp. 16–26.

Models of Bounded Rationality for Autonomous Agents

Timotheus Kampik (Supervisors: Helena Lindgren, Juan Carlos Nieves)

Department of Computing Science, Umeå University

Summary

Problem:
Autonomous agents are typically designed to align their actions with the goal of maximizing simple metrics, *e.g.*:

- One-dimensional utility functions
- Recommender systems optimized for clicks and views

$\Longrightarrow$ Many complex autonomous (information) systems apply a simplistic approach to decision-making–***their measure of performance becomes the target***.

Solution:
Design and apply algorithms that enable agents to compromise between different measures, stakeholder interests, or qualitative considerations, while also considering properties of ideal, *rational* economic decision-making.

Example scenarios:

- User-centric recommender systems that compromise between user and system provider goals
- Autonomous vehicles that dynamically resolve conflicts of interest in the interest of the party who can benefit most
- Decision support systems that allow for strategic decision-making under uncertainty

Research question:

How can we apply and design formal modals of boundedly rational economic choice that enable better decision-making of autonomous agents?

Contact Information

- Web: https://people.cs.umu.se/tkampik
- Email: tkampik@cs.umu.se

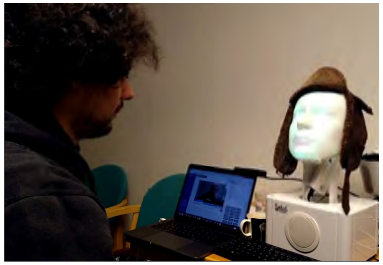


Figure: Human-agent economic game: do agent explanation facilitate fair sharing?

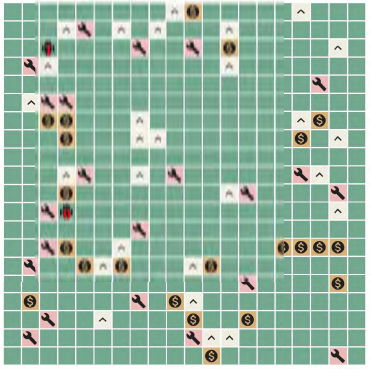


Figure: Reinforcement learning agents: rewards are given based on distance to fair equilibrium

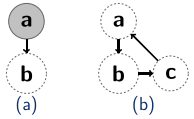


Figure: Inconsistent preferences: using many *abstract argumentation* reasoning methods, the left graph implies $\{a\}$ is preferred over $\{\}$, while the right graph implies $\{\}$ is preferred over $\{a\}$.

Formal Models of Bounded Rationality

- Given any set of choice items A, a rational agent consistently chooses the same items $A^* \subseteq A$, which implies that $\forall A_c \subseteq A, A_c \neq A^*$: A^* is preferred over A_c.
- Given a set of options $A' \supseteq A$, the agent must choose A'^* so that $A'^* = A^*$ or $A'^* \not\subseteq A$.
- This model is too simple to guide real-life decision-making (as has been shown by a range of behavioral economics research).
- However, the model can be used as a sanity check for decision and reasoning algorithms. *E.g.*, the figure above shows that many abstract argumentation methods are not compliant with properties of economic rationality.

Preliminary Key Results

- **Conceptual/*Blue Sky***
  1) Analysis of coercion and deception in persuasive technology [1]
  2) Conceptual proposal for technology-facilitated societal consensus [2]
- **Formal Theory**
  1) Preliminary game-theoretical foundations of boundedly altruistic agents [3, 4].
  2) Formal proofs: abstract argumentation methods are not economically rational [5]
- **Engineering**
  1) Proof-of-concept implementation of boundedly utilitarian agents [6].
  2) JavaScript library for implementing light-weight autonomous agents with different reasoning loops [7]
- **Empirical Research**
  1) Preliminary RL experiment: rewarding agents based on *fair equilibria* teaches agents fairness [8]
  2) Preliminary HCI study: explanations might facilitate sharing in human-agent economic games [9]

Bibliography (selection)

[1] Timotheus Kampik, Juan Carlos Nieves, and Helena Lindgren. Coercion and deception in persuasive technologies. In *20th International Trust Workshop (co-located with AAMAS/IJCAI/ECAI/ICML 2018), Stockholm, Sweden, 14 July, 2018*, pages 38–49. CEUR-WS. 2018.

[2] Timotheus Kampik and Amro Najjar. Technology-facilitated societal consensus. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, UMAP'19 Adjunct, page 3–7, New York, NY, USA, 2019. Association for Computing Machinery.

[3] Timotheus Kampik, Juan Carlos Nieves, and Helena Lindgren. Empathic autonomous agents. In Danny Weyns, Viviana Mascardi, and Alessandro Ricci, editors, *Engineering Multi-Agent Systems*, pages 181–201, Cham, 2019. Springer International Publishing.

[4] Timotheus Kampik. Empathic agents: A hybrid normative/consequentialistic approach. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 2423–2425, Richland. SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems.

[5] Timotheus Kampik and Juan Carlos Nieves. Abstract argumentation and the rational man (arxiv pre-print), 2019.

[6] Timotheus Kampik, Juan Carlos Nieves, and Helena Lindgren. Implementing argumentation-enabled empathic agents. In Marija Slavkovik, editor, *Multi-Agent Systems*, pages 140–155, Cham, 2019. Springer International Publishing.

[7] Timotheus Kampik and Juan Carlos Nieves. Js-son-a minimal javascript bdi agent library. In *7th International Workshop on Engineering Multi-Agent Systems (EMAS 2019)*, 2019.

[8] Timotheus Kampik and Helge Spieker. Learning agents of bounded rationality: Rewards based on fair equilibria. 2019.

[9] Timotheus Kampik, Juan Carlos Nieves, and Helena Lindgren. Explaining sympathetic actions of rational agents. In Davide Calvaresi, Amro Najjar, Michael Schumacher, and Kary Främling, editors, *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 59–76, Cham, 2019. Springer International Publishing.

Acknowledgements

This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

WASP | UMEÅ UNIVERSITY