

Bachelor Class 2015-2016

Siegfried Nijssen

17 November 2015



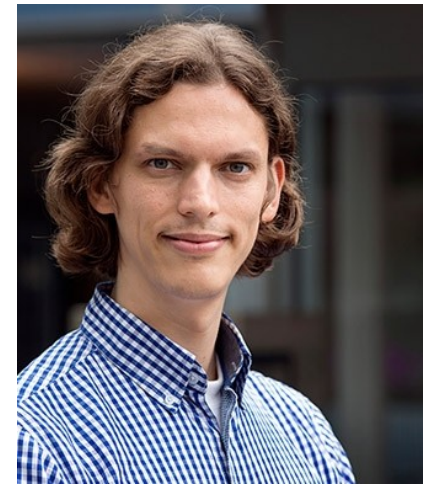
Universiteit
Leiden

Coordinators

Siegfried Nijssen

s.nijssen@liacs.leidenuniv.nl

<http://www.liacs.leidenuniv.nl/~nijssensgr>



Amr Ali-Eldin

a.m.t.ali-eldin@liacs.leidenuniv.nl

The Bachelor Class in a Nutshell

- 1) Overview of research groups and bachelor project topics
- 2) Hints & tips: how to write a thesis, how to manage your project, ...
- 3) Progress reports:
 - Poster presentation about the problem you are studying
 - Intermediate presentation about how you are solving the problem
 - Final presentation about your results

Overview of the Bachelor Class

| week | Datum | Ma | Di | Wo | Do | Vr | |
|------|--------|--------------------|-----------------|-----------------|-----------------|-----------------|--|
| nr | Ma | 1 2 3 4 5 6 7 8 | 1 2 3 4 5 6 7 8 | 1 2 3 4 5 6 7 8 | 1 2 3 4 5 6 7 8 | 1 2 3 4 5 6 7 8 | |
| 47 | 16 nov | | HCI | DaMi | | | |
| 48 | 23 nov | | HCI | DaMi | | | |
| 49 | 30 nov | | HCI | DaMi | | | |
| 50 | 7 dec | | HCI | DaMi | | | |
| 51 | 14 dec | | T CoCo | | | | |
| 52 | 21 dec | | T HCI | | | | |
| 53 | 28 dec | Gesloten | | Gesloten | | Gesloten | |
| 1 | 4 jan | | T DaMi | | | | |
| 2 | 11 jan | | | | | T TvC | |
| 3 | 18 jan | | | Bklas | | | |
| 4 | 25 jan | | | Bklas | | | |
| 5 | 1 feb | spb | FI3 | Netw | wFI3 | | |
| 6 | 8 feb | FI3 | Diesviering | Netw | wFI3 | | |
| 7 | 15 feb | | FI3 | Netw | wFI3 | | |
| 8 | 22 feb | | FI3 | Netw | wFI3 | | |
| 9 | 29 feb | | FI3 | Netw | wFI3 | | |
| 10 | 7 mrt | | H HCI | | H CoCo | H DaMi | |
| 11 | 14 mrt | | FI3 | Netw | wFI3 | beta barometer | |
| 12 | 21 mrt | | FI3 | Netw | wFI3 | | |
| 13 | 28 mrt | Tweede Paasdag | | Netw | wFI3 | | |
| 14 | 4 apr | | FI3 | Netw | wFI3 | | |
| 15 | 11 apr | | FI3 | Netw | wFI3 | | |
| 16 | 18 apr | | FI3 | Netw | wFI3 | | |
| 17 | 25 apr | | FI3 | Netw | wFI3 | | |
| 18 | 2 mei | | FI3 | Netw | wFI3 | | |
| 19 | 9 mei | | FI3 | Netw | wFI3 | | |
| 20 | 16 mei | Tweede Pinksterdag | | Netw | wFI3 | | |
| 21 | 23 mei | | | T Netw | | | |
| 22 | 30 mei | | T NC | | Bklas | T FI3 | |



Presentation of Topics

Allocation of Topics

Poster Presentations

Howtos

Presentations

Detailed Information



Bachelorklas 2015-2016 - Chromium <2>

Bachelorklas 2015-2016 x

liacs.leidenuniv.nl/~nijssensgr/bachelorklas-2015-2016/

Siegfried

SCHEDULE INFORMATION FORMS

Bachelorklas 2015-2016

| Date | More information |
|------------------|---|
| 17 November 2015 | Introduction to bachelor projects (1) |
| 1 December 2015 | Introduction to bachelor projects (2) |
| 9 December 2015 | Introduction to bachelor projects (3) |

Rules for Participation

- Everybody is welcome to attend...
- Active participation is only allowed if:
 - You have finished your propedeuse
 - At the start of second semester, at most 2 courses from the second year and the first semester are missing

Contact Jeannette De Graaf or Ronniy Joseph for exceptions

- You have to participate in order to graduate, including attending all classes and giving all presentations
 - If your 2 missing courses are in the spring semester, you may get permission to finish your project in the autumn and skip the intermediate presentation. Contact Jeannette or Ronniy!

Rules for Participation

- If you cannot be present for a class, you should send a mail **in advance** providing a good motivation
- After 2 missed classes for no good announced reason, you need to make an appointment with the study advisor
- Every class, you need to put your name on an attendance form

Bachelor Dossier

- Includes:
 - Spring seminar
 - Autumn seminar
 - Bachelor class
 - Bachelor thesis, presentation
- 18EC for informatica
16EC for bachelor thesis and bachelorclass
- 16EC for informatica & economie
14EC for bachelor thesis and bachelorclass

Choosing a Topic

- All research groups of LIACS will present topics till **December 9**
- The topics will be on the website; by **December 20** you will need to **rank** at least **6 topics**, from at least **3** different supervisors (→ link will be on the website)
- By **January 20** we will finalize the allocation of students to supervisors
- By **February 3** you have to hand in (on paper and online) a **contract** (→ document will be on the website)
<http://liacs.leidenuniv.nl/~nijssensgr/bachelorklas-2014-2015/contract.html>

Research at LIACS

- Algorithms and Software Technology (AST)
 - Games
 - Formal methods
 - Optimization
 - Data science
- Computer systems and Imagery & Media (CSI)
 - Imaging, information retrieval
 - Bioinformatics
 - High performance computing
 - Embedded systems

AST: Games

- *Walter Kusters, Hendrik Jan Hoogeboom, Aske Plaat, Jaap van den Herik*
- Related to artificial intelligence, complexity
- Example projects:
 - Hanabi: A co-operative game of fireworks
 - Compact Decision Trees for Dou Shou Qi Tablebases
 - Predicting the Outcome of the Game Othello
 - Using Outcome Weights in Monte-Carlo Tree Search for Multiplayer 3D Hex
 - A Difficulty Measure for Light Up Puzzles
 - Strategies for Klondike Solitaire
 - Solving Jungle Checkers
 - An Analysis of Dominion
- Popularity warning: typically, many students interested

AST: Formal Methods

- *Marcello Bonsangue, Jetty Kleijn, Farhad Arbab, Frank de Boer, Rudy van Vliet, Luuk Groenenwegen*
- Related to concepts of logic, programming languages, fundamentals of computer science, programming and correctness, software engineering, theory of concurrency
- Example projects:
 - Context Free Guarded Languages: A system for determining Guarded Strings
 - The Constraint-Relation Modelling Language and its relation to Petri Nets
 - Equivalence checking of regular expressions using non-deterministic finite automata
 - An On-Line Parsing Algorithm for conjunctive grammars
 - Reducing copying and network traffic in Reo circuits
 - Testing of Channel Based Service Connectors

AST: Optimization

- *Thomas Bäck, Michael Emmerich*
- Related to artificial intelligence, computational intelligence, natural computing
- Example projects:
 - Mining Bitcoins with Natural Computing Algorithms
 - An Evolutionary Algorithm for Finding Diverse Sets of Molecules with User-Defined Properties
 - Multi-objective Generation of Bicycle Routes
 - A Genetic Algorithm for the Travelling Salesman Problem with Area Constraints

CSI: High Performance Computing

- *Harry Wijshoff, Kristian Rietveld*
- Related to operating systems, networks, digital techniques, computer architecture, compiler construction
- Example projects:
 - Deploying Phenotype Analysis On LLSC
 - Deploying Single Particle Analysis on the LLSC
 - A Framework for Cross-Platform Dynamically Loaded Libraries
 - Implementing I/O Infrastructure Improvements for S.M.A.C.K.

CSI: Bioinformatics

- *Fons Verbeek, Kathy Wolstencroft, Sacha Goultiaev*
- Related to human computer interaction, data mining, topics from high computing, software engineering
- Example projects:
 - Deploying Phenotype Analysis On LLSC
 - Deploying Single Particle Analysis on the LLSC
 - Integrating data modeling with data analysis in Taverna workflows
 - Ontology viewer: from proof-of-concept to layered software

CSI: Embedded Systems

- *Todor Stefanov*
- Related to digital techniques, computer architecture, compiler construction
- Example projects:
 - Exploring scheduling alternatives for a Computer Vision application on embedded MPSoCs
 - Auto-vectorization using polyhedral compilation for an embedded ARM platform

CSI: Imaging & Media

- *Michael Lew, Erwin Bakker*
- Related to data mining, computer graphics, artificial intelligence
- Example projects:
 - An algorithm for morphing audio
 - Combined Neural Networks for Movie Recommendation
 - Image Similarity Using Color Histograms
 - Video rating and sorting with a genuine approach
 - A comparison of search engine user interfaces
 - Finding correspondence in stereo image pairs using an adaptive window comparison algorithm
 - Detailed crowd simulation and spatial hashing for large-scale collision detection
 - Video Recommendation, A comparison between collaborative filtering algorithms
- Popularity warning: typically, many students interested

AST: Data Science

- *Joost Kok, Aske Plaat, Jaap van den Herik, Siegfried Nijssen, Peter Lucas, Stefan Manegold, Thomas Bäck, Michael Emmerich, Matthijs van Leeuwen, Frank Takes, Cor Veenman, Wojtek Kowalczyk, Arno Knobbe*
- Related to data mining, databases, statistics, artificial intelligence
- Example projects:
 - Combining graph mining and deep learning in molecular activity prediction
 - Mining a scientific conference
 - Data mining the Peptide Sequenome
 - Analysis and Visualisation of Data of an Outdoor Sports Mobile Application
 - Inference in Markov Networks
 - Data triangulation: combining Ebola datasets for gaining retrospective insights

AST: Data Science Applications

- Industry (steel factories, car manufacturers, aircraft manufacturers)
Michael Emmerich, Thomas Bäck, Matthijs van Leeuwen
- Banking, insurance
Wojtek Kowalczyk, Arno Knobbe
- Sports
Arno Knobbe, Joost Kok
- Biology, chemistry
Siegfried Nijssen, Michael Emmerich
- Hospital
Aske Plaat, Peter Lucas, Joost Kok, Siegfried Nijssen
- Law enforcement
Aske Plaat, Cor Veenman
- Traffic
Aske Plaat, Arno Knobbe, Siegfried Nijssen
- Social media
Frank Takes, Aske Plaat, Siegfried Nijssen

AST: Data Science Fundamentals

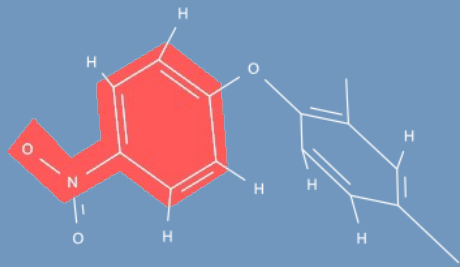
- Database systems
Stefan Manegold, Michael Emmerich
- Neural networks
Wojtek Kowalczyk, Siegfried Nijssen, Walter Kusters
- Pattern mining algorithms (itemset mining, subgroup discovery)
Siegfried Nijssen, Matthijs van Leeuwen, Arno Knobbe
- Supervised machine learning algorithms
Siegfried Nijssen
- Gaussian processes
Michael Emmerich
- Bayesian networks
Peter Lucas
- Graph algorithms
Frank Takes, Siegfried Nijssen

Siegfried Nijssen

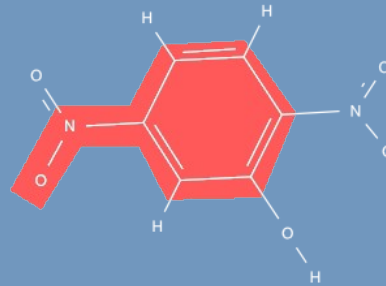
- Master in computer science (Leiden, 2000)
- PhD in computer science (Leiden, 2006)
- Post doc in Leuven (KU Leuven)
- Docent (Leiden)

- Machine learning
- Data mining
- Artificial intelligence

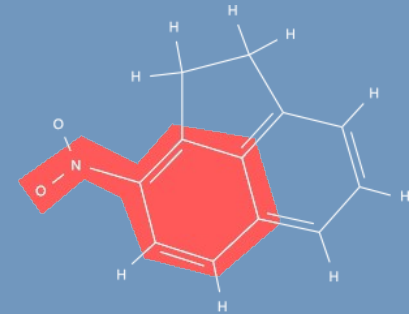
Graph Mining



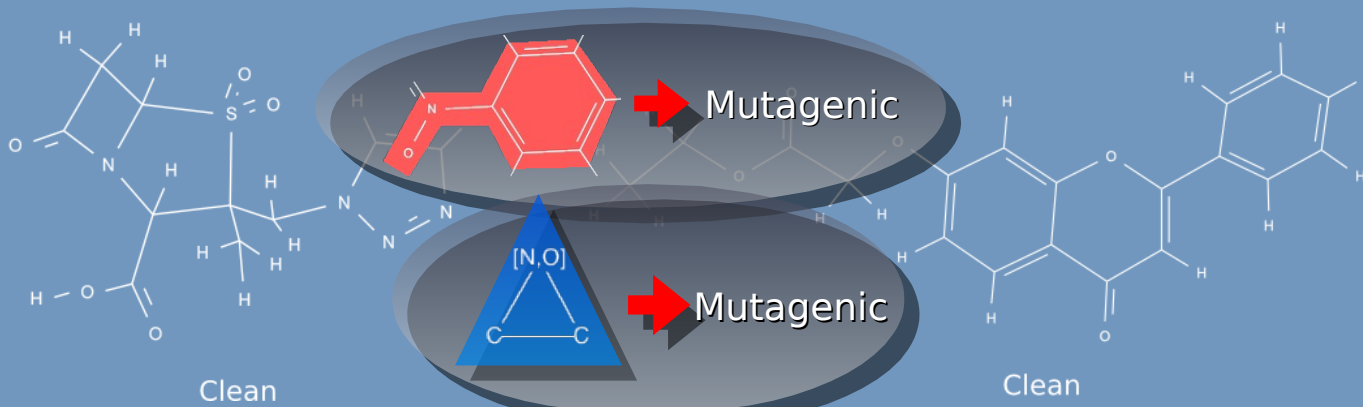
Mutagenic



Mutagenic



Mutagenic



Graph Mining

- Can we efficiently predict whether a molecule is active?
- Questions:
 - How to represent molecules?
 - How to search for features?
 - Which classifier to use?
- Requirements:
 - An interest in efficient programming in C++
 - An interest in graph theory
 - An interest in data mining

Pattern Mining

Market basket data

$$\text{support}(\text{Pampers, Beer}) = 3$$

| | | | | | |
|---|---|--|---|---|---|
|  | |  |  |  | |
|  |  |  |  | | |
|  |  | | |  |  |
|  | |  |  | | |
|  | | |  | |  |

Pattern Mining

- Situation comparable to having a specialized system for each possible database query

- Apriori
- FP-Growth
- Eclat
- SD-Apriori
- DDPMine
- Gaston
- gSpan
- FFSM

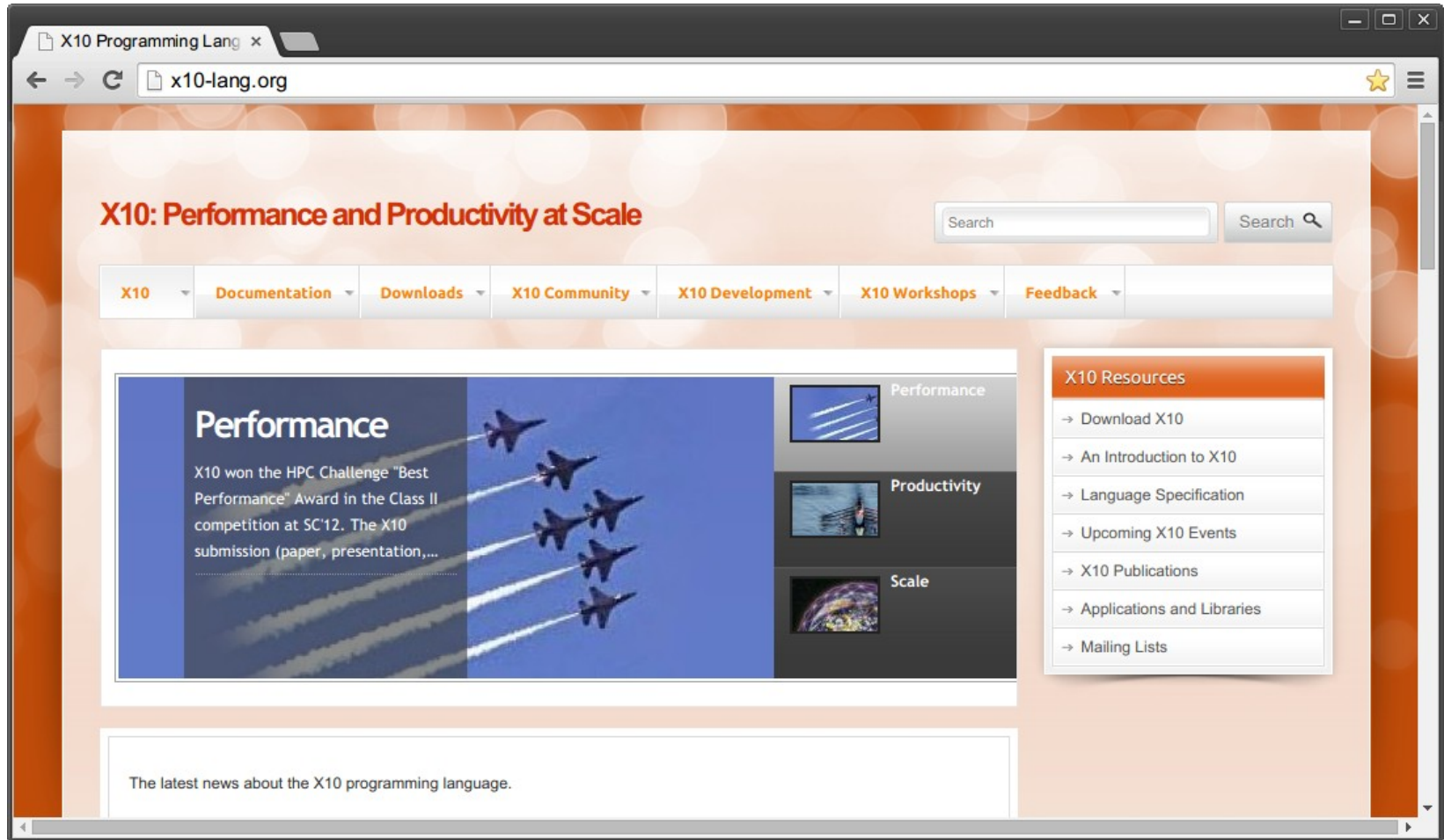
- TreeMiner
- LCM
- MaxMiner
- DualMiner
- Molfea
- CorrMine
- EclatV
- Mafia

- kDCI
- ARMOR
- AIM
- COFI-tree
- DCI closed
- WinePI
- MinePI
-

A 4th Generation Language for Data Mining

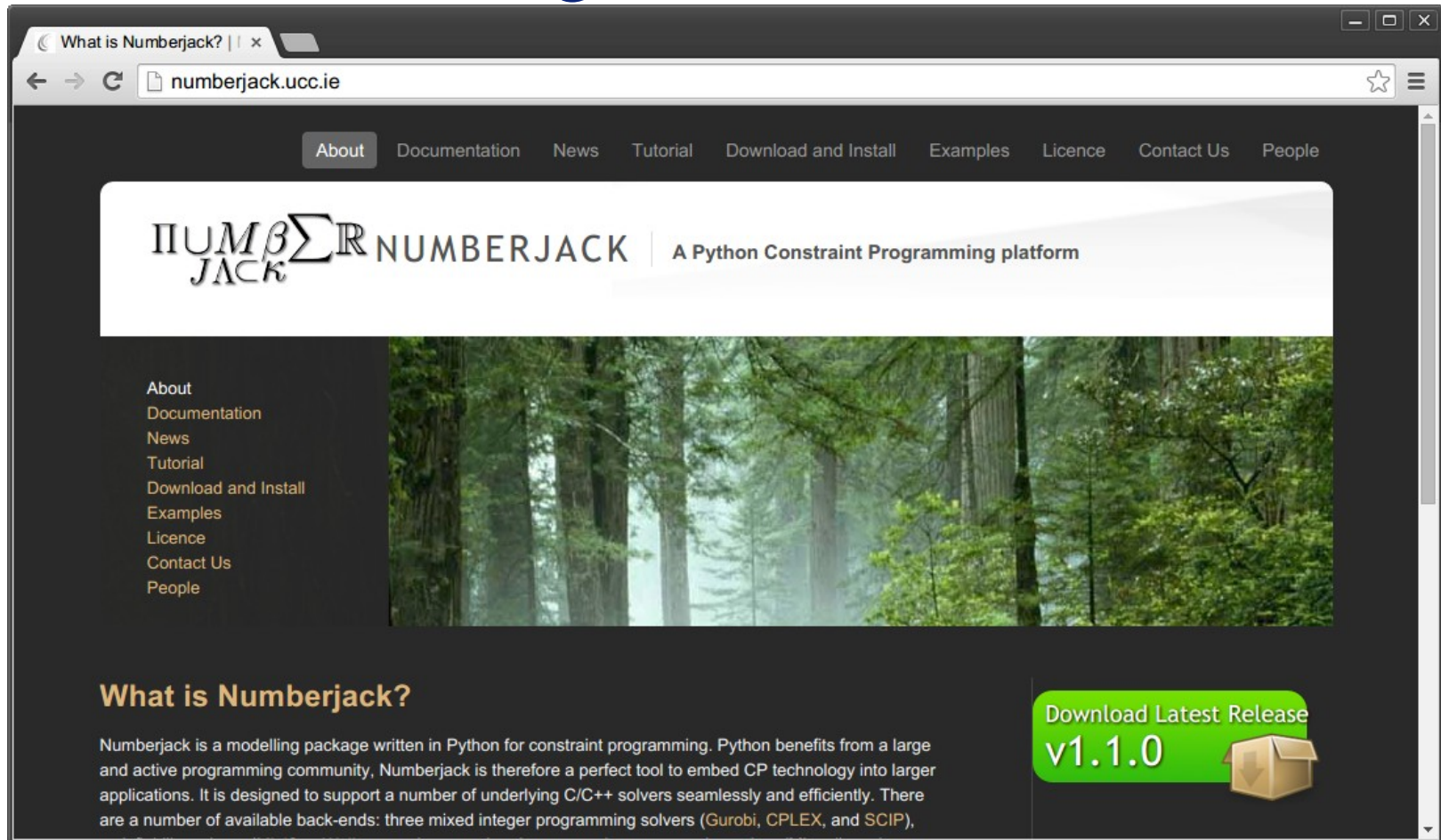
- “An SQL for data mining”

A 4th Generation Language for Data Mining



(IBM's X10 language)

A 4th Generation Language for Data Mining



The screenshot shows the homepage of the Numberjack website. The browser address bar displays 'numberjack.ucc.ie'. The navigation menu includes 'About', 'Documentation', 'News', 'Tutorial', 'Download and Install', 'Examples', 'Licence', 'Contact Us', and 'People'. The main header features the logo 'NUMBERJACK' with mathematical symbols Π , β , Σ , and R integrated into the letters, and the tagline 'A Python Constraint Programming platform'. A large image of a forest is visible in the background. On the left, a sidebar lists the navigation menu items. The main content area has a section titled 'What is Numberjack?' with a paragraph of text. To the right, there is a green button that says 'Download Latest Release v1.1.0' with a cardboard box icon.

What is Numberjack?

Numberjack is a modelling package written in Python for constraint programming. Python benefits from a large and active programming community, Numberjack is therefore a perfect tool to embed CP technology into larger applications. It is designed to support a number of underlying C/C++ solvers seamlessly and efficiently. There are a number of available back-ends: three mixed integer programming solvers (Gurobi, CPLEX, and SCIP),

Download Latest Release
v1.1.0

(Numberjack CP system)

A 4th Generation Language for Data Mining

The screenshot shows the Apache Spark website homepage. The browser title is "Apache Spark™ - Lightning-Fast Cluster Computing - Chromium". The address bar shows "spark.apache.org". The page features the Spark logo and the tagline "Lightning-fast cluster computing". A navigation bar includes links for "Download", "Libraries", "Documentation", "Examples", "Community", and "FAQ". A central text box states: "Apache Spark™ is a fast and general engine for large-scale data processing." Below this, a "Speed" section highlights that Spark runs programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk. A bar chart compares the running time for logistic regression: Hadoop takes 110 seconds, while Spark takes 0.9 seconds. A "Latest News" section lists recent releases: Spark 1.5.2 (Nov 09, 2015), Spark 1.5.1 (Oct 02, 2015), and Spark 1.5.0 (Sep 09, 2015). A "Download Spark" button is visible at the bottom right.

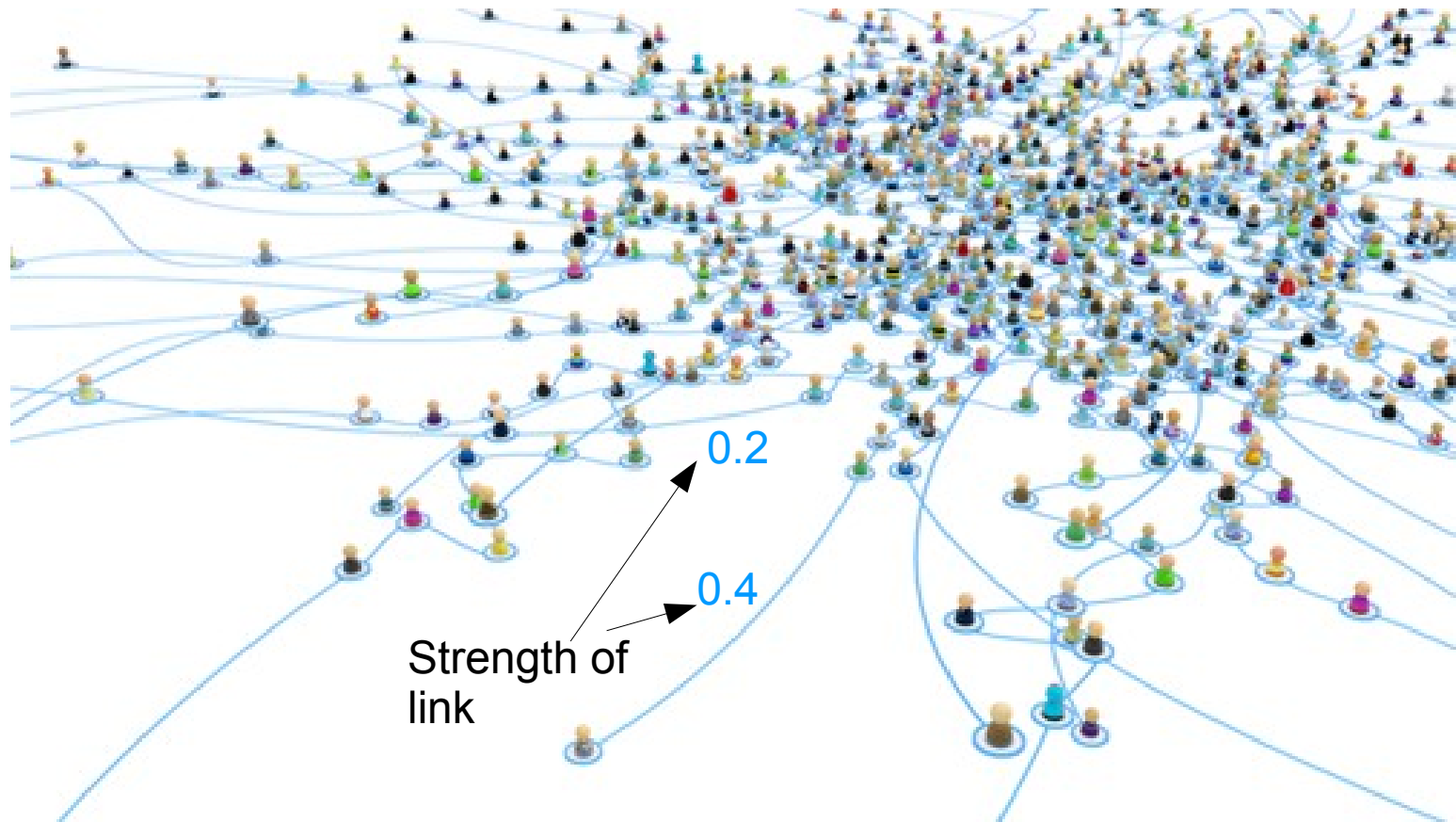
| Technology | Running time (s) |
|------------|------------------|
| Hadoop | 110 |
| Spark | 0.9 |

- Apache Spark

A 4th Generation Language for Data Mining

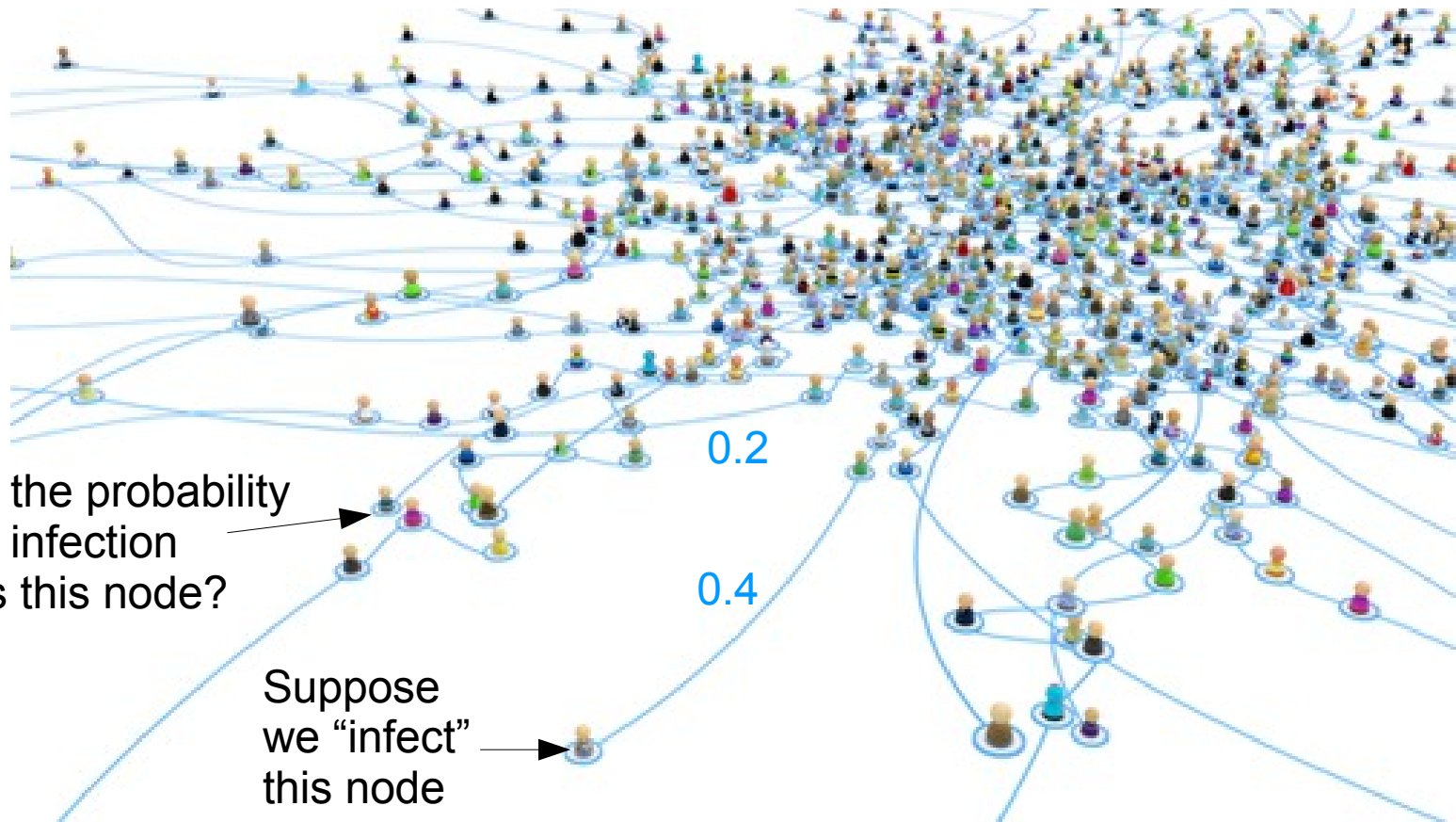
- Can an effective declarative data mining system be built in Python, based on *Numberjack*, *X10*, or *Apache Spark*?
- Requirements:
 - An interest in programming in developing and learning new languages
 - An interest in algorithms
 - An interest in artificial intelligence
 - An interest in data mining

Inference in Probabilistic Networks



Social network, protein interaction network, ... with uncertain or unreliable links

Inference in Probabilistic Networks



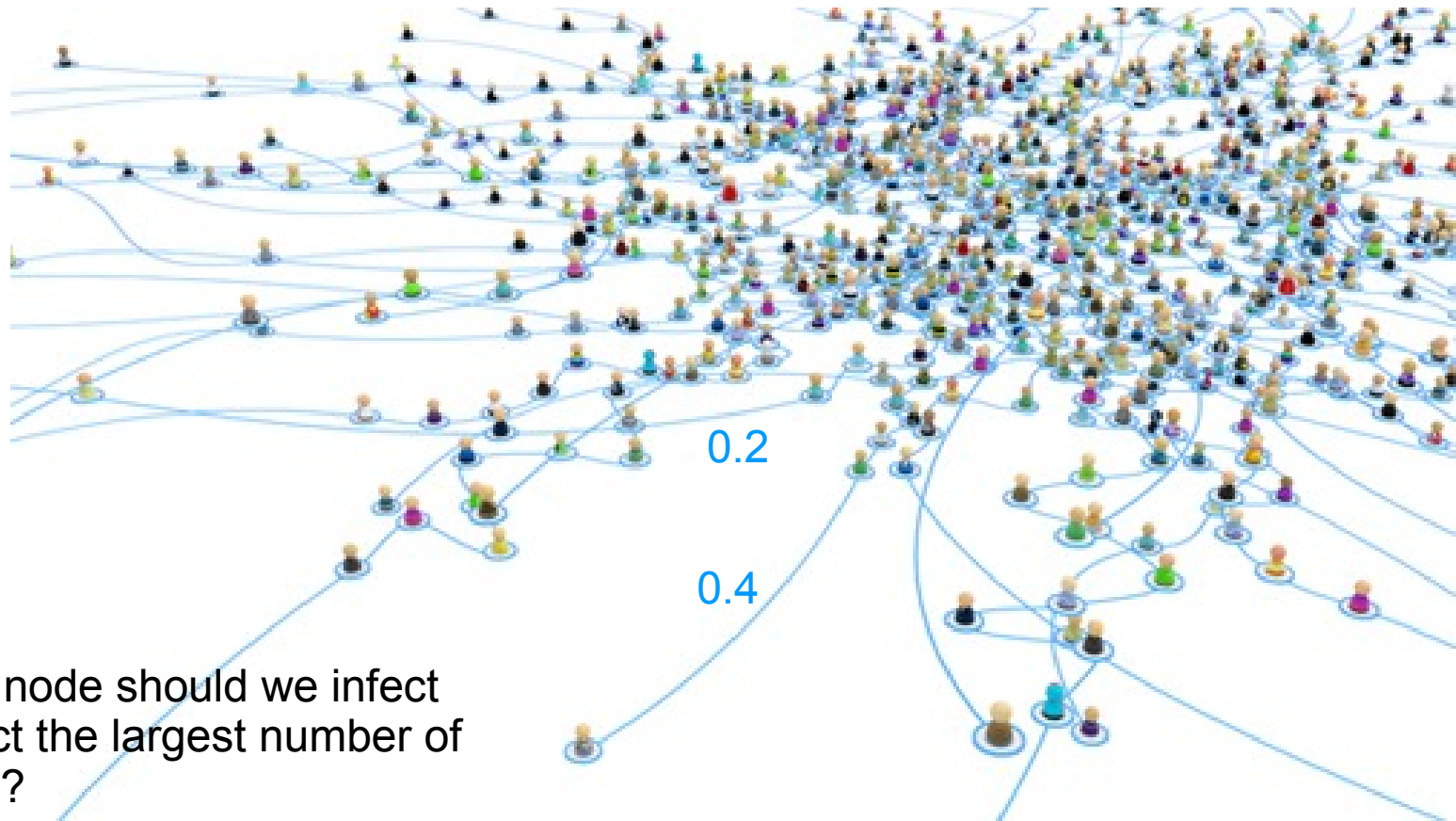
What is the probability that the infection reaches this node?

Suppose we "infect" this node

0.2
0.4

Social network, protein interaction network, ... with uncertain or unreliable links

Inference in Probabilistic Networks



Which node should we infect to infect the largest number of people?

Social network, protein interaction network, ... with uncertain or unreliable links

Inference in Probabilistic Networks

- Questions:
 - How to efficiently calculate probabilities?
 - How to efficiently find the state with the largest probability?
 - How to represent problems in networks?
- Requirements:
 - An interest in algorithms
 - An interest in probabilistic reasoning, artificial intelligence