Preface

Benelearn, as the Annual Belgian Dutch Conference on Machine learning and the major forum for machine learning researchers in Belgium and the Netherlands, is up to its twenty-first edition this year. For the past twenty editions, it has served as one of the main occasions for regional machine learning scientists to present their work and foster new collaborations. This year will be no different. In fact, we expanded the scope of the conference somewhat, by including a satellite workshop day on applications of Predictive Modeling in the Life Sciences: PMLS. The twenty-first BeneLearn will be held May 24th 2012 in Ghent, Belgium, at the Faculty of Bioscience Engineering. The PMLS workshop will take place at the same location on the 25th of May. Organizers are Bernard De Baets, Michael Rademaker, Jan Verwaeren and Willem Waegeman from the KERMIT research unit, Ghent University, and Bernard Manderick, VUB.

It has become traditional to invite a plenary speaker at BeneLearn. In fact, because of the satellite workshop with a focus on applications, this year, we were able to invite three speakers without sacrificing possibilities for regional starting researchers to present their work. On the BeneLearn day, we have Ricardo Silva (Department of Statistical Science at the Centre for Computational Statistics and Machine Learning, University College London (UK)) as invited speaker, who will give an overview of his work on copulas: “Structured Copula Models in Supervised and Unsupervised Learning”. The remainder of the day will mainly consist of presentations based on abstract and full paper submissions. We have a distinguished paper presentation, too, where Joachim De Beule will present his submission “Overcoming the Tragedy of the Commune in the Hawk-Dove Game through Conventional Coding”. This year, we also organise a poster spotlight session, preceding the lunch and contiguous poster session, composed of submissions to both BeneLearn and PMLS. We are sure this will be another occasion to bridge the gap between theoretical and application-focused research. On the PMLS day, we are happy to present Peter Challenor (National Oceanography Centre (UK)), discussing “Climate, Models and Uncertainty”, and Tijl De Bie (University of Bristol (UK)), talking about “Structured Output Prediction: from Biology to Music”. We also include ten oral presentations of the more application-oriented submissions.

Time for some numbers. In total, we have 21 oral presentations (11 for BeneLearn, 10 for PMLS) and 19 poster presentations. The proceedings are composed of 9 short papers and 31 one-page abstracts. As of the time of finalising these proceedings, we have a total number of 80 registered participants.

Thanks are in order to a number of people: the program committee (especially Peter van der Putten, Maarten Van Someren and Hendrik Blockeel) for feedback and reviews, secretary Timpe Vogelaars for administrative duties, and PhD-student Jan Verwaeren, as web-site developer, proceedings cover designer and all-round behind-the-scenes organiser. We also would like to mention our sponsors, without whom this conference would have been impossible: We thank FWO and SIKS for their contributions. We are also grateful to our alma mater, Ghent University, and the Faculty of Bioscience Engineering for the conference location. Finally, we also thank you, the attendees, without whom this conference would have been quite pointless.

BeneLearn 2012 and PMLS Scientific Committee:
Bernard De Baets   Bernard Manderick   Michael Rademaker   Willem Waegeman
BeneLearn 2012 and PMLS: Table of Contents
### BeneLearn 2012 and PMLS: Front Matter

- List of Program Committee Members
- List of Organizing Committee Members
- List of Authors

### BeneLearn 2012 and PMLS: Invited Speakers

- **Ricardo Silva**  
  Structured Copula Models in Supervised and Unsupervised Learning

- **Peter Challenor**  
  Climate, Models and Uncertainty

- **Tijl De Bie**  
  Structured Output Prediction: from Biology to Music

### BeneLearn 2012 and PMLS: Short Papers

<table>
<thead>
<tr>
<th>Authors</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Catteeuw and Bernard Manderick</td>
<td>Honest Signaling: Learning Dynamics versus Evolutionary Stability</td>
<td>01</td>
</tr>
<tr>
<td>Joachim De Beule</td>
<td>Overcoming the Tragedy of the Commune in the Hawk-Dove Game through Conventional Coding</td>
<td>07</td>
</tr>
<tr>
<td>Gert-Jan Den Heijer, Peter van der Putten, Erica Benard, Annemarie Meijer and Fons Verbeek</td>
<td>Explorations in Texture Based Classification for Bacterial Infection in Zebrafish</td>
<td>13</td>
</tr>
<tr>
<td>Miguel Lopes, Patrick Meyer and Gianluca Bontempi</td>
<td>Estimation of Temporal Lags for the Inference of Gene Regulatory Networks from Time Series</td>
<td>19</td>
</tr>
<tr>
<td>Alexander Nezhinsky, Esther Stoop, Anastasiya Vasylevska, Astrid van der Sar and Fons Verbeek</td>
<td>Spatial Analysis of Bacterial Infection Patterns in Zebrafish</td>
<td>27</td>
</tr>
<tr>
<td>Kevin Stadler, Pieter Wellens and Joachim De Beule</td>
<td>The Combinatorial Naming Game</td>
<td>33</td>
</tr>
<tr>
<td>Gitte Vanwinckelen and Hendrik Blockeel</td>
<td>Estimating Model Accuracy with Repeated Cross-Validation</td>
<td>39</td>
</tr>
<tr>
<td>Jonas Vlasselaer and Wannes Meert</td>
<td>Statistical Relational Learning for Prognostics</td>
<td>45</td>
</tr>
<tr>
<td>Willem Wybo, Camille Colle, Pieter-Jan Kindermans and Benjamin Schrauwen</td>
<td>Distance Dependent Extensions of the Chinese Restaurant Process</td>
<td>51</td>
</tr>
</tbody>
</table>
Pieter Boets, Koen Lock and Peter Goethals
Modelling Habitat Preference, Abundance and Species Richness of Alien Macrocrustaceans in Surface Waters in Flanders (Belgium) using Decision Trees

Veronika Cheplygina, David Tax, Robert Duin, Wan-Jui Lee and Marco Loog
Bag Dissimilarities for Multiple Instance Learning

Ellen Colman, Willem Waegeman, Bernard De Baets and Veerle Fievez
Can Acidosis in Dairy Cows be Diagnosed from Milk Fatty Acids?

Eduardo P. Costa, Sicco Verwer and Hendrik Blockeel
New procedure for estimating prediction certainty in decision trees

Iulian M. Covlescu and Zoltán Szlávik
Exploring Item-User-Rating Datasets as Images

Jeff Daelman, Jeanne Marie Membré, Liesbeth Jacxsens, Frank Devlieghere and Mieke Uyttendaele
A Bayesian Model for Bacillus cereus Contamination in Raw Materials Used for REPFED Production

Jonas Degrave, Francis Wyffels, Tim Waegeman, Pieter-Jan Kindermans and Benjamin Schrauwen
Applying Morphological Changes during the Evolution of Quadruped Robots Results in Robust Gaits

Rob D’Hondt, Dries Landuyt and Katrien Van Der Biest
Determination of Trade-Offs in Ecosystem Service Delivery Using Bayesian Belief Networks

Shinji Fukuda, Bernard De Baets, Willem Waegeman, Jan Verwaeren and Ans Mouton
Comparison of Predictive Accuracy and Habitat Preference Information Retrieved from Seven Species Distribution Models

Shinji Fukuda, Norio Onikura, Takahiko Mukai and Jun Nakajima
Multilabel Classification for Assessing the Genetic Disturbance by Freshwater Fish Invasions in Northern Kyushu, Japan

Javier Holguin Gonzalez, Gert Everaert and Peter L.M. Goethals

Arnaud Joly, François Schnitzler, Pierre Geurts and Louis Wehenkel
L1-Based Compression of Random Forest Models

Ik-Joon Kang, Shinji Fukuda, Junya Moroishi and Trinh Quang Huy
Rule-based Classification of Toxic Response of a Freshwater Fish to Contaminated River Water

Pieter-Jan Kindermans, Francis Wyffels, Ken Caluwaerts, Bert Guns and Benjamin Schrauwen
Towards Incorporation of Hierarchical Bayesian Models into Evolution Strategies for Quadrupeded Gait Generation

Parisa Kordjamshidi and Marie-Francine Moens
Spatial Role Labeling using Structured Support Vector Machines
Sofie Landschoot, Willem Waegeman, Kris Audenaert, Geert Haesaert and Bernard De Baets
A Machine Learning Approach to Predict Fusarium Head Blight

Wessel Luijben and Zoltán Szlávik
Stage Detection in Runs of Evolutionary Algorithms

Mihail Mihaylov, Karl Tuyls and Ann Nowé
Simple Decentralized Algorithm for Coordination Games

Jens Nyman, Ken Caluwaerts, Tim Waegeman and Benjamin Schrauwen
System Modeling for Active Noise Control with Reservoir Computing

Hafeez Osman, Michel Chaudron and Peter van der Putten
Classifying Presence of Classes in UML Design using Software Metrics

Valentijn Pauwels, Gabrielle De Lannoy, Harrie-Jan Hendricks Franssen and Harry Vereecken
Simultaneous Estimation of Model State Variables and Observation and Forecast Biases using a Two-Stage Hybrid Kalman Filter

Douglas Antonio Plaza Guingla and Valentijn Pauwels
Improved Particle Filter by using the Ensemble Kalman Filter as the Importance Density Function in Rainfall-Runoff Models

Joeri Ruyssinck, Tom Dhaene and Yvan Saeys
Inferring Gene Regulatory Network Topologies using Ensembles of Feature Selection Techniques

Sean Stijven, Tom Dhaene and Wouter Minnebo
Separating the Wheat from the Chaff: On Feature Selection in Regression Random Forests and Symbolic Regression

Michiel Stock, Willem Waegeman and Bernard De Baets
Relational Learning and Ranking Algorithms for Biological Applications

Jafar Tanha, Maarten van Someren and Hamideh Afsarmanesh
A Multiclass Semi-Supervised Boosting Algorithm

Gustavo Torres, Kathleen Marchal, Yves Van De Peer and Martine De Cock
Predicting Long Term Behavior of Genetic Regulatory Networks with Answer Set Programming

Jasper Van doninck, Niko E. C. Verhoest, Jan Peters, Bernard De Baets and Els Ducheyne
Identifying Relevant Remote Sensing Products for the Spatial Modelling of a Bluetongue Vector using Random Forests

Jan Van Haaren and Guy Van Den Broeck
Relational Learning for Football-Related Predictions

Mathias Verbeke, Roser Morante, Paolo Frasconi and Luc De Raedt
A Statistical Relational Learning Approach to Identify Sections in Scientific Abstracts using Sentence and Document Structure

Jan Verwaeren, Willem Waegeman, Tapio Pahikkala, Antti Airola and Bernard De Baets
Incorporating Prior Knowledge in Multiple Output Regression with Kernel-Based Vector Functions
BeneLearn 2012 and PMLS: Lists of Committee Members and Authors
BeneLearn 2012 and PMLS Program Committee

Hendrik Blockeel, KU Leuven
Gianluca Bontempi, ULB
Walter Daelemans, UA
Jesse Davis, KU Leuven
Bernard De Baets, KERMIT, UGent
Tom Dhaene, UGent
Kurt Driessens, Maastricht University
Pierre Dupont, UC Louvain
Damien Ernst, Université de Liège
Ad Feelders, Utrecht University
Pierre Geurts, Université de Liège
Tom Heskes, Radboud University Nijmegen
Wojtek Kowalczyk, Vrije Universiteit Amsterdam
Jan Lemeire, VUB
Steven Maenhout, HoGent
Michael Mampaey, UA
Ann Nowé, VUB
Mykola Pechenizkiy, TU Eindhoven
Jan Peters, VITO
Eric Postma, Universiteit van Tilburg
Michaël Rademaker, KERMIT, UGent
Jan Ramon, KU Leuven
Yvan Saeys, UGent
Benjamin Schrauwen, UGent
Rodolphe Sepulchre, Université de Liège
Evgueni Smirnov, Maastricht University
Johan Suykens, KU Leuven
Evgeni Tsvitsivadze, Radboud University Nijmegen
Stefan Van Aelst, UGent
Dirk Van Den Poel, UGent
Peter Van Der Putten, Pegasystems & Leiden University
Marieke Van Erp, Vrije Universiteit Amsterdam
Menno van Zaanten, Universiteit van Tilburg
Koen Vanhoof, Universiteit Hasselt
Joaquin Vanschoren, Leiden University
Michel Verleysen, UC Louvain
Sicco Verwer, Technische Universiteit Delft
Willem Waegeman, KERMIT, UGent

BeneLearn 2012 and PMLS Organizing Committee

Bernard De Baets, KERMIT, UGent
Michaël Rademaker, KERMIT, UGent
Jan Verwaeren, KERMIT, UGent
Willem Waegeman, KERMIT, UGent
Bernard Manderick, COMO, VUB
<table>
<thead>
<tr>
<th>NAME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afsarmanesh, Hamideh</td>
<td>82</td>
</tr>
<tr>
<td>Airola, Antti</td>
<td>87</td>
</tr>
<tr>
<td>Audenaert, Kris</td>
<td>72</td>
</tr>
<tr>
<td>Benard, Erica</td>
<td>13</td>
</tr>
<tr>
<td>Blockeel, Hendrik</td>
<td>60, 39</td>
</tr>
<tr>
<td>Boets, Pieter</td>
<td>57</td>
</tr>
<tr>
<td>Bontempi, Gianluca</td>
<td>19</td>
</tr>
<tr>
<td>Caluwaerts, Ken</td>
<td>70, 75</td>
</tr>
<tr>
<td>Catteeuw, David</td>
<td>01</td>
</tr>
<tr>
<td>Chaudron, Michel</td>
<td>76</td>
</tr>
<tr>
<td>Cheplygina, Veronika</td>
<td>58</td>
</tr>
<tr>
<td>Colle, Camille</td>
<td>51</td>
</tr>
<tr>
<td>Colman, Ellen</td>
<td>59</td>
</tr>
<tr>
<td>Costa, Eduardo P</td>
<td>60</td>
</tr>
<tr>
<td>Covlescu, Iulian M</td>
<td>61</td>
</tr>
<tr>
<td>Daelman, Jeff</td>
<td>62</td>
</tr>
<tr>
<td>De Baets, Bernard</td>
<td>59, 65, 72, 81, 84, 87</td>
</tr>
<tr>
<td>De Beule, Joachim</td>
<td>07, 33</td>
</tr>
<tr>
<td>De Cock, Martine</td>
<td>83</td>
</tr>
<tr>
<td>De Lannoy, Gabrielle</td>
<td>77</td>
</tr>
<tr>
<td>De Raedt, Luc</td>
<td>86</td>
</tr>
<tr>
<td>Degrave, Jonas</td>
<td>63</td>
</tr>
<tr>
<td>Den Heijer, Gert-Jan</td>
<td>13</td>
</tr>
<tr>
<td>Devlieghere, Frank</td>
<td>62</td>
</tr>
<tr>
<td>Dhaene, Tom</td>
<td>79, 80</td>
</tr>
<tr>
<td>D'Hondt, Rob</td>
<td>64</td>
</tr>
<tr>
<td>Ducheyne, Els</td>
<td>84</td>
</tr>
<tr>
<td>Duin, Robert</td>
<td>58</td>
</tr>
<tr>
<td>Everaert, Gert</td>
<td>67</td>
</tr>
<tr>
<td>Fievez, Veerle</td>
<td>59</td>
</tr>
<tr>
<td>Frasconi, Paolo</td>
<td>86</td>
</tr>
<tr>
<td>Fukuda, Shinji</td>
<td>65,66,69</td>
</tr>
<tr>
<td>Geurts, Pierre</td>
<td>68</td>
</tr>
<tr>
<td>Goethals, Peter L. M.</td>
<td>57, 67</td>
</tr>
<tr>
<td>Guns, Bert</td>
<td>70</td>
</tr>
<tr>
<td>Haesaert, Geert</td>
<td>72</td>
</tr>
<tr>
<td>Hendricks Franssen, Harrie-Jan</td>
<td>77</td>
</tr>
<tr>
<td>Holguin Gonzalez, Javier</td>
<td>67</td>
</tr>
<tr>
<td>Jacksens, Liesbeth</td>
<td>62</td>
</tr>
<tr>
<td>Joly, Arnaud</td>
<td>68</td>
</tr>
<tr>
<td>Kang, Ik-Joon</td>
<td>69</td>
</tr>
<tr>
<td>Kindermans, Pieter-Jan</td>
<td>51, 63, 70</td>
</tr>
<tr>
<td>Kordjamshidi, Parisa</td>
<td>71</td>
</tr>
<tr>
<td>Landschoot, Sofie</td>
<td>72</td>
</tr>
<tr>
<td>Landuyt, Dries</td>
<td>64</td>
</tr>
<tr>
<td>Lee, Wan-Jui</td>
<td>58</td>
</tr>
<tr>
<td>Lock, Koen</td>
<td>57</td>
</tr>
<tr>
<td>Loog, Marco</td>
<td>58</td>
</tr>
<tr>
<td>Lopes, Miguel</td>
<td>19</td>
</tr>
<tr>
<td>Luijben, Wessel</td>
<td>73</td>
</tr>
<tr>
<td>Manderick, Bernard</td>
<td>01</td>
</tr>
<tr>
<td>Marchal, Kathleen</td>
<td>83</td>
</tr>
<tr>
<td>Meert, Wannes</td>
<td>45</td>
</tr>
<tr>
<td>Meijer, Annemarie</td>
<td>13</td>
</tr>
<tr>
<td>Membré, Jeanne Marie</td>
<td>62</td>
</tr>
<tr>
<td>Meyer, Patrick</td>
<td>19</td>
</tr>
<tr>
<td>Mihaylov, Mihail</td>
<td>74</td>
</tr>
<tr>
<td>Minnebo, Wouter</td>
<td>80</td>
</tr>
<tr>
<td>Moens, Marie-Francine</td>
<td>71</td>
</tr>
<tr>
<td>Morante, Roser</td>
<td>86</td>
</tr>
<tr>
<td>Moroishi, Junya</td>
<td>69</td>
</tr>
<tr>
<td>Mouton, Ans</td>
<td>65</td>
</tr>
<tr>
<td>Mukai, Takahiko</td>
<td>66</td>
</tr>
<tr>
<td>Nakajima, Jun</td>
<td>66</td>
</tr>
<tr>
<td>Nezhinsky, Alexander</td>
<td>27</td>
</tr>
<tr>
<td>Nowé, Ann</td>
<td>74</td>
</tr>
<tr>
<td>Nyman, Jens</td>
<td>75</td>
</tr>
<tr>
<td>Onikura, Norio</td>
<td>66</td>
</tr>
<tr>
<td>Osman, Hafeez</td>
<td>76</td>
</tr>
<tr>
<td>Pahikkala, Tapio</td>
<td>87</td>
</tr>
<tr>
<td>Pauwels, Valentijn</td>
<td>77, 78</td>
</tr>
<tr>
<td>Peters, Jan</td>
<td>84</td>
</tr>
<tr>
<td>Plaza Guingla, Douglas Antonio</td>
<td>78</td>
</tr>
<tr>
<td>Quang Huy, Trinh</td>
<td>69</td>
</tr>
<tr>
<td>Ruyssinck, Joeri</td>
<td>79</td>
</tr>
<tr>
<td>Saesys, Yvan</td>
<td>79</td>
</tr>
<tr>
<td>Schnitzler, François</td>
<td>68</td>
</tr>
<tr>
<td>Schrauwen, Benjamin</td>
<td>51, 63, 70, 75</td>
</tr>
<tr>
<td>Stadler, Kevin</td>
<td>33</td>
</tr>
<tr>
<td>Stijven, Sean</td>
<td>80</td>
</tr>
<tr>
<td>Stock, Michiel</td>
<td>81</td>
</tr>
<tr>
<td>Stoop, Esther</td>
<td>27</td>
</tr>
<tr>
<td>Szlávik, Zoltán</td>
<td>61, 73</td>
</tr>
<tr>
<td>Tanha, Jafar</td>
<td>82</td>
</tr>
<tr>
<td>Tax, David</td>
<td>82</td>
</tr>
<tr>
<td>Torres, Gustavo</td>
<td>83</td>
</tr>
<tr>
<td>Tuyls, Karl</td>
<td>74</td>
</tr>
<tr>
<td>Uyttendaele, Mieke</td>
<td>62</td>
</tr>
<tr>
<td>Van De Peer, Yves</td>
<td>83</td>
</tr>
<tr>
<td>Van Den Broeck, Guy</td>
<td>85</td>
</tr>
<tr>
<td>Name</td>
<td>Page Numbers</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Van Der Biest, Katrien</td>
<td>64</td>
</tr>
<tr>
<td>van der Putten, Peter</td>
<td>13, 76</td>
</tr>
<tr>
<td>van der Sar, Astrid</td>
<td>27</td>
</tr>
<tr>
<td>Van doninck, Jasper</td>
<td>84</td>
</tr>
<tr>
<td>Van Haaren, Jan</td>
<td>85</td>
</tr>
<tr>
<td>van Someren, Maarten</td>
<td>82</td>
</tr>
<tr>
<td>Vanwinckelen, Gitte</td>
<td>39</td>
</tr>
<tr>
<td>Vasylevska, Anastasiya</td>
<td>27</td>
</tr>
<tr>
<td>Verbeek, Fons</td>
<td>13, 27</td>
</tr>
<tr>
<td>Verbeke, Mathias</td>
<td>86</td>
</tr>
<tr>
<td>Vereecken, Harry</td>
<td>77</td>
</tr>
</tbody>
</table>
BeneLearn 2012 and PMLS: Invited Speakers
It is not hard to understand what a copula function is: a cumulative distribution function with uniformly distributed marginals. But this seemingly simple concept has interesting consequences. Copulas provide an alternative way of representing multivariate dependencies: it allows one to encode the marginal and the dependence structure of a joint distribution separately, a type of representational modularity that complements other popular tools in machine learning such as graphical models. For instance, one could build flexible nonparametric representations for univariate marginals while using parametric forms for the potentially much more daunting task of representing associations in high-dimensional spaces - and such a parametric structure will impose no constraints on how marginals are represented. In this talk, we will first provide a brief introduction to copulas. We then proceed to exploit this concept in the context of graphical models. First, we show how to build high-dimensional copulas using a mixture of trees representation, which we then learn with a Bayesian approach. Second, we will discuss how copulas can be exploited in structured prediction models, where there is a residual association structure among a set of labels or other outcome variables of interest that remain dependent even after observing the predictive features (joint work with Robert Gramacy (Chicago), Charles Blundell and Yee Whye The).

Anthropogenic climate change is one of the most important challenges facing society. The associated scientific problems are difficult, urgent and fascinating. Predicting what future climates might look like, under different greenhouse gas emission scenarios, involves the use of very large computer codes to solve complex numerical models of the Earth system. We would like to have a measure of the uncertainty on such predictions. In this talk I will concentrate on the problem of how we might estimate the uncertainty on predictions of the climate from complex numerical models. I will discuss both the single and multi-model case.

In this talk I will survey a number of recent approaches to the task of learning to predict structured output labels, and I will discuss recent developments such as the adaptation of these methods for semi-supervised learning scenarios. The talk will be motivated and illustrated by practical use-cases from biological sequence analysis, music annotation, and more.
BeneLearn 2012 and PMLS: Short Papers
Honest Signaling: Learning Dynamics versus Evolutionary Stability

David Catteeuw  
Bernard Manderick  
Artificial Intelligence Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Keywords: honest signaling, handicap principle, reinforcement learning, evolutionary stable strategy

Abstract

Until now, biologist have mostly studied under what circumstances honest signaling is stable. Stability, however, is not sufficient to explain the emergence of honest signaling. We observe that honest signaling can emerge through learning. The settings where honest signaling evolves, however, do not exactly match those where honest signaling is evolutionarily stable. They do, however, match the set where honest signaling is a Pareto-optimal Nash equilibrium. As such, we provide an alternative explanation for the emergence and existence of honest signaling.

1. Introduction

We study the emergence of honest signaling in the Philip Sidney game. It was introduced by Maynard Smith (Maynard Smith, 1991) and has become the standard game theoretic model of the handicap principle. The handicap principle (Zahavi, 1975) states that under conflict of interest honest signaling can only be an equilibrium if signals are costly. The handicap principle may explain, among other phenomena, the male characteristics used for sexual selection, such as a peacock’s tail. The peacock’s tail is a handicap for its survival, but, as such, it is an honest signals of the male’s fitness. Females can reliably infer from the size of those tails which males would make better mates, since only the fittest can afford the largest tails.

To demonstrate or contest the significance of honest signaling (Maynard Smith & Harper, 2003; Searcy & Nowicki, 2005), biologists are almost exclusively relying on an finding the evolutionary stable strategies of different game theoretic models such as the Philip Sidney game (Grafen, 1990; Maynard Smith, 1991). Such explanations of honest signaling, however, do not tell us whether honest signaling will emerge, it merely states that if it would emerge (for example through mutation) it can persist. The same critique has been formulated by Huttegger and Zollman (Huttegger & Zollman, 2010). We discuss this related work in Section 5.

Here, we apply learning dynamics (Section 3) to different classes of the Philip Sidney game (Section 2) and we observe that:

- a simple adaptive process, such as reinforcement learning can lead to honest signaling,
- the settings where honest signaling is evolutionary stable do not fully match the settings where honest signaling emerges through learning,
- the learning process always converged to a Pareto-optimal Nash equilibrium.

A Nash equilibrium is Pareto-optimal if there is no other Nash equilibrium where no players are worse off and at least one player is strictly better off. We discuss the experimental results in Section 4.

2. The Philip Sidney Game

The Philip Sidney game, see Figure 1, is a signaling game, introduced by Maynard Smith (Maynard Smith, 1991) as an example of the handicap principle. The Philip Sidney game is a two-player extensive form game of incomplete information. The first player (Sender) can be in one of two states: healthy or needy, with probability \( p \) and \( 1 - p \) respectively. In both cases he can either send a signal at some cost \( c \) or he can remain silent. Player 2 (Receiver) does not know the true state of Sender, but he can observe whether or not Sender signals. Furthermore, Receiver has a resource and must decide whether or not to donate his resource to Sender.
The players’ chances of survival depend on the state $t \in \{\text{healthy, needy}\}$, the signal $s \in \{\text{signal, silent}\}$, and the action $a \in \{\text{donate, keep}\}$ as follows. Sender is sure to survive if he receives the resource. If Sender does not receive the resource and he is needy, then his chance of survival is 0. If he does not receive the resource and he is healthy, then his chance of survival is $V < 1$. Thus, he benefits more from receiving the resource when he is needy, than when he is healthy. Sender’s survival probability $v_S(t, s, a)$ is summarized by Equations 1 and 2.

$$v_S(t, s, a) = \begin{cases} (1 - c)w_{t,a} & \text{if } s = \text{signal} \\ w_{t,a} & \text{if } s = \text{silent} \end{cases}$$  

(1)

$$w = \begin{pmatrix} \text{healthy} \\ \text{needy} \end{pmatrix} \begin{pmatrix} 1 & V \\ 1 & 0 \end{pmatrix}$$  

(2)

On the other hand, if Receiver keeps the resource to himself, he is sure to survive. If he donates the resource to Sender his chances of survival are reduced to $S < 1$, see Equation 3.

$$v_R(t, s, a) = \begin{cases} S & \text{if } a = \text{donate} \\ 1 & \text{if } a = \text{keep} \end{cases}$$  

(3)

Clearly there is no reason for Receiver to donate his resource, except that the players may be related by some factor $r$, such that the utility $w$ of each player is his survival probability plus $r$ times the other player’s survival probability. See Equations 4 and 5 for Sender’s and Receiver’s utility $u_S$ and $u_R$.

$$u_S(t, s, a) = v_S(t, s, a) + rv_R(t, s, a)$$  

(4)

$$u_R(t, s, a) = v_R(t, s, a) + rv_S(t, s, a)$$  

(5)

Increasing the relatedness factor $r$ reduces the conflict between the players. For $r = 1$, both players have the same utilities. This is known as Hamilton’s principle of inclusive fitness (Hamilton, 1964). The game tree and the utilities for each possible outcome of the Philip Sidney game are shown in Figure 1 and the parameters are summarized in Table 1.

The standard way to represent the behavior of players of an extensive form game is by a behavioral strategy profile. A behavioral strategy profile is a list of strategies, one for each information state – this is a set of player nodes between which the player cannot distinguish. Sender’s information states are healthy and needy. Receiver’s information states are signal and silent. The behavior corresponding to honest signaling: “Sender signals only when needy and Receiver donates only when signal” is denoted by $(l, s, d, k)$, where the letters $l$, $s$, $d$, $k$ respectively stand for the pure strategies: silent, signal, donate, and keep.

We selected a number of Philip Sidney games that allow us to illustrate the most important effects we observed during simulations of learning dynamics. The exact parameters and the classification of those games are given in Table 2. Table 3 shows an overview of the types of equilibria that apply to honest signaling. We come back to these games in Section 4 where we discuss the experimental results.

### 3. Selfish Adaptive Behavior

Reinforcement learning algorithms are ideal as models of simple adaptive behavior of selfish individuals. A reinforcement learner repeatedly observes the state of the environment, selects an action and receives a payoff indicating the quality of his action. An individual can maintain statistics of the payoffs for all combinations of observations and actions. This way, he can learn which are the most rewarding actions in each state of the environment and hence change his behavior by selecting better actions more often.

![Figure 1. The Philip Sidney game.](image-url)
Table 2. Some Philip Sidney games and their classification. For all games Sender is healthy with probability $p = 0.5$.

<table>
<thead>
<tr>
<th>$c$</th>
<th>$r$</th>
<th>$S$</th>
<th>$V$</th>
<th>Costly?</th>
<th>Conflict?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.19</td>
<td>0.5</td>
<td>0.6</td>
<td>0.65</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.5</td>
<td>0.7</td>
<td>0.8</td>
<td>YES</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
<td>0.5</td>
<td>0.8</td>
<td>0.8</td>
<td>YES</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
<td>0.5</td>
<td>0.8</td>
<td>0.8</td>
<td>YES</td>
</tr>
<tr>
<td>5</td>
<td>0.2</td>
<td>0.5</td>
<td>0.7</td>
<td>0.5</td>
<td>YES</td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>0.5</td>
<td>0.8</td>
<td>0.95</td>
<td>NO</td>
</tr>
<tr>
<td>7</td>
<td>0.0</td>
<td>0.5</td>
<td>0.8</td>
<td>0.8</td>
<td>NO</td>
</tr>
<tr>
<td>8</td>
<td>0.2</td>
<td>0.25</td>
<td>0.7</td>
<td>0.5</td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 3. For which Philip Sidney games honest signaling is an evolutionary stable strategy (ESS), a Nash equilibrium (NE), a Pareto-optimal Nash equilibrium (PO), and when honest signaling emerged during experiments with learning dynamics. In the column of Pareto-optimal (PO) we write "yes*" when honest signaling is not the only Pareto-optimal Nash equilibrium.

<table>
<thead>
<tr>
<th>ESS?</th>
<th>NE?</th>
<th>PO?</th>
<th>Learned?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>YES</td>
<td>YES</td>
<td>YES*</td>
</tr>
<tr>
<td>3</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>NO</td>
<td>NO</td>
<td>YES*</td>
</tr>
<tr>
<td>5</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>6</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>7</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>8</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

We use Q-learning (Watkins & Dayan, 1992) with $\epsilon$-greedy action selection, because of its ease of applicability. The parameters are easy to choose and the algorithm poses no constraints on the payoffs. Sender has a Q-value $Q_{t,s}$ for each possible state $t \in \{\text{healthy, needy}\}$ and each possible signal he can send, $s \in \{\text{signal, silent}\}$. Receiver has a Q-value $Q_{s,a}$ for each possible signal he can observe, $s \in \{\text{signal, silent}\}$, and each possible action he can take, $a \in \{\text{donate, keep}\}$.

We initialize Q-values optimistically by setting them to a value which is greater than or equal to the highest payoff of the game. This makes sure that there is sufficient exploration in the beginning of the game (Sutton & Barto, 1998, p.40). In the experiments we set initial Q-values to 2.

Now, for each game with outcome $(t, s, a)$, both players update the Q-value that corresponds to the observation they made and the action they took during that game, according to the update rule 6. The other Q-values remain unchanged.

$$Q_{t,s} \leftarrow Q_{t,s} + \alpha (u_t(t,s,a) - Q_{t,s})$$

$$Q_{s,a} \leftarrow Q_{s,a} + \alpha (u_s(t,s,a) - Q_{s,a})$$

(6)

In this update rule, $\alpha \in [0,1]$ represents the learning rate. In the extreme case of $\alpha = 0$, nothing is ever learned. In the other extreme case where $\alpha = 1$, Q-values only reflect the last utility for their corresponding outcome. In fact, the Q-values represent the exponentially weighted moving average of the utilities (Sutton & Barto, 1998, p.38). Both players used the same learning rate $\alpha = 0.1$ in all experiments.

To balance exploration and exploitation, we used $\epsilon$-greedy action selection: with probability $1 - \epsilon$ individuals select the action with the highest Q-value (braking ties randomly) and with probability $\epsilon$ they select an action at random according to a uniform distribution. For all experiments we used a constant exploration rate $\epsilon = 0.01$.

4. Experiments and Results

In all experiments reported here, initial Q-values were 2, the learning rate $\alpha = 0.1$ and the exploration rate $\epsilon = 0.01$ for both players. For each game setting, we did 100 runs. Per run, the game was repeated 2000 times. This is long enough, since most of the time, the behavior was already stable after a few hundred iterations.

In Table 4 we present the learned behavior for each of the game settings given in Table 2. Per game we give a list of behaviors, each of them preceded by the percentage of runs that converged to that behavior. The basic notation for behavior is described at the end of Section 2. The question mark represents a wild card and $(\frac{1}{2}a + \frac{1}{2})$ represents a mixed strategy where Sender signals with probability 1/2 and is silent with probability 1/2.

4.1. Partial Conflict and Costly Signals

Games 1 to 5 all have a partial conflict of interest between players but costly signals, thus the handicap principle says that honest signaling can be evolutionary stable. This is the case for games 1 to 3, but not for games 4 and 5.

For the first setting, game 1, as expected, honest signaling always emerged through learning. A typical run is shown in Figure 2.

In the second setting, game 2, some runs (59%) converged to honest signaling: $(1,s,d,k)$. Some
runs converged to another equilibrium: “Sender remains always silent and Receiver donates when silent”, (1, 1, ?, d). The question mark in (1, 1, ?, d) denotes that the receiver can either donate or keep in case Sender signals. Note, that Sender will still signal at a rate of $\epsilon/2 = 0.005$ due to the constant exploration rate (see Section 3). The difference with game setting 1 may be explained by the fact that game 2 has two Pareto-optimal Nash equilibria: (1, s, d, k) and (1, 1, ?, d). These correspond to the learned behavior. In contrast, in game setting 1 honest signaling is the unique Pareto-optimal Nash equilibrium and the only behavior learned.

In game 3, honest signaling is evolutionary stable (and by definition also a Nash equilibrium). The learning dynamics, however, never converged to the honest signaling equilibrium, instead it converged to “Sender remains always silent and Receiver always donates”, (1, 1, d, d), which is a Pareto-optimal Nash equilibrium for this game. So, as before, the learning process converged to a Pareto-optimal Nash equilibrium.

Game 4 shows the opposite effect of game 3. Here, honest signaling is not evolutionary stable although it is a Pareto-optimal Nash equilibrium. There is another Pareto-optimal Nash equilibrium: “Sender is always silent and Receiver donates when Sender is silent”, (1, 1, ?, d). The learning dynamics evolved to this equilibrium in 88% of the runs. In a smaller number of runs, 12 out of 100, we observed periods of honest signaling (1, s, d, k) interleaved by periods where Sender, when healthy, is bluffing half of the time: ($1/3s + 1/3d, s, d, k$). This last behavior is not Pareto-optimal in itself but the combination of (1, s, d, k) and ($1/2s + 1/2d, s, d, k$) is, as long as Sender signals at most $1/3$ of the time when he is healthy.

In game 5 honest signaling is not even a Nash equilibrium. Again, the learning process converged to a Pareto-optimal Nash equilibrium: “Sender is always silent and Receiver donates when Sender is silent”, (1, 1, ?, d).

4.2. Cost-free Signals

Game 6 has cost-free signals and no conflict of interest. Honest signaling is evolutionary stable and thus also a Nash equilibrium. Since there is no cost for signaling, both (1, s, d, k) and (s, 1, k, d) come down to the same thing. These behaviors simply have the meaning of silent and signal switched. The learning dynamics reached both equilibria in a more or less equal amount of runs. In fact, this example shows how arbitrary signals can acquire meaning through learning processes. This was already studied extensively (Skyrms, 2010), see the related work in Section 5.

Game 7 also has cost-free signals but with partial conflicting interest. According to the handicap principle honest signaling cannot be evolutionary stable. Honest signaling is not a Nash equilibrium either. There are, however, many equilibria were partial information is transferred. Some of them were identified by Hüttegger and Zollman (Hüttegger & Zollman, 2010), see Section 5. In our case the learning dynamics always converged to any of the pure strategy equilibria: (s, s, d, ?) and (1, 1, ?, d), both in about half of

---

Table 4. The behavioral strategy profiles to which learning converged and for which percentage of the runs for each of the Philip Sidney games in Table 2. The notation is explained in the text.

<table>
<thead>
<tr>
<th>LEARNED BEHAVIOR</th>
<th>P(healthy, silent, keep)</th>
<th>P(needy, signal, donate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 100% (1, s, d, k)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2 59% (1, s, d, k), 41% (1, 1, ?, d)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3 100% (1, 1, d, d)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4 88% (1, 1, ?, d), 12% (1, s, d, k), ($1/3s + 1/3d, s, d, k$)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5 100% (1, 1, ?, d)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6 56% (1, s, d, k), 44% (s, 1, k, d)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7 51% (s, s, d, ?), 49% (1, 1, ?, d)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8 100% (1, ?, k, k)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

---

Figure 2. A typical run for game 1 showing the emergence of honest signaling. The solid red line represents the probability of seeing outcome (healthy, silent, keep) and the dotted green line the probability of seeing outcome (needy, signal, donate). Other than equilibrium outcomes quickly converged towards zero probability and are left out for clarity.
the runs. All Nash equilibria have the same payoffs (1,4,1.3), so both strategies profiles are also Pareto-optimal Nash equilibria.

4.3. Full Conflict

When there is a full conflict, Receiver prefers to keep his resource whether Sender is healthy or not, and Sender would prefer Receiver to donate at all times. In such settings, there are no Nash equilibria (and hence no evolutionary stable strategies) where Receiver donates. As a consequence, it does not really matter whether Sender is signaling honestly or not. Moreover, when signals are costly, Sender is better off by remaining silent when he is healthy. In game 8, for example, we observed Sender learned to signal honestly. Receiver, however, never bothered about the signal, (1, s, k, k).

In summary, the learning process always converged to a Pareto-optimal Nash equilibrium. So, for this learning process, honest signaling can emerge, but only when it is a Pareto-optimal Nash equilibrium. We were able to identify settings in which honest signaling is evolutionary stable but not Pareto-optimal, for example game 3. We also identified settings where honest signaling is not evolutionary stable but could emerge through learning, for example game 4.

5. Related Work

Huttegger and Zollman (Huttegger & Zollman, 2010) study evolutionary dynamics in the Philip Sidney game, and contrast their results to those obtained by calculating the evolutionary stable strategies. They find that in many cases honest signaling has far smaller basins of attraction than other equilibria. This may mean that honest signaling is far less significant as is suggested by the existence of evolutionary stable strategies. Indeed, there is an ongoing debate about how widespread honest signaling really is (Maynard Smith & Harper, 2003; Searcy & Nowicki, 2005). Our work is exploring the same question and we argue that analyzing dynamics of adaptive processes leading to honest signaling can provide better insights.

Closely related to the emergence of honest signaling is the emergence of signaling itself. The emergence of signaling in a game theoretic setting was first studied by Lewis (Lewis, 1969). His signaling games represent a communication problem where there is neither cost for signaling, nor conflict. The problem is fully cooperative and the goal for the players is to come up with a convention for the meaning of the signals without any pre-existent means of communication. Recently, both evolutionary dynamics (Hobauer & Huttegger, 2008; Huttegger et al., 2009) and learning dynamics (Barrett & Zollman, 2009; Argiento et al., 2009; Skyrms, 2010; Catteeuw et al., 2011) received more attention and it was shown that perfect communication can evolve from initially random behavior through simple adaptive processes. While we concentrate on honest signaling in this text, we could indeed observe that initially arbitrary signals can acquire meaning through repeated interaction between learning processes. Games 6 and 7 are signaling games where signals are cost-free. We observed that the meaning acquired by the signals was swapped in about half of the runs, see Table 4 and Section 4.2.

6. Conclusion

In biology honest signaling has mostly been studied by doing a static analysis of signaling games, that is verifying whether honest signaling is an evolutionary stable strategy. In the process, they have been ignoring other equilibria which may be equally or even more important. Both the work of Huttegger and Zollman (Huttegger & Zollman, 2010), using evolutionary dynamics, and our work, using learning dynamics, show that honest signaling can emerge from initially random behavior through adaptive processes and more importantly it shows the existence of settings where honest signaling is an equilibrium, but where it is not (necessarily) the outcome of the dynamic process. We also observed the opposite, honest signaling can emerge through learning in settings where it is not evolutionary stable.

Note that there is an important difference between evolutionary stability and evolutionary dynamics on the one hand and learning dynamics on the other hand. Evolutionary stability and dynamics considers how and whether characteristics due to genetic mutations can spread in a population. Learning dynamics considers whether individuals can acquire certain characteristics. Learned behavior can spread through the population through imitation processes (individuals imitate the behavior of fitter individuals), but it can also influence genetic evolution since natural selection may favor individuals that can acquire the characteristic through their lifetime (Hinton & Nowlan, 1987). As such the genetic evolution is guided by the learning. In the limit the characteristic can even be "canalized" and does not need to be learned at all (Waddington, 1942).

This allows us to question whether analysis of evolutionary stability is a good or the only good methodology to study the emergence of honest signaling. Other
equilibrium concepts exist, but it is not clear which one should apply. Learning dynamics provides a simple and intuitive alternative.

One may wonder, in how far the choice of learning model (Section 3) influenced our results. We argue that the choice is a reasonable one. Similar models of reinforcement learning have been applied to signaling games (Barrett & Zollman, 2009; Catteeuw et al., 2011). All of them are very limited in terms of computational and informational requirements as mentioned before. They keep track of statistics summarizing the quality of their actions and keep a balance between exploiting the best actions and exploring the others. Despite their simplicity, these models have characteristics also seen in human and animal behavior.

References


Overcoming the Tragedy of the Commune in the Hawk-Dove game through Conventional Coding

Joachim De Beule  
Artificial Intelligence Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium  
Keywords: mechanisms of evolution, language, learning, biosemiotics, multi-agent modeling

Abstract
In the Hawk-Dove game, where two individuals compete over a resource, fully cooperating or “dove”-like behavior is vulnerable to invasion by defecting or “hawk”-like behavior, a fact also known as the “tragedy of the commune”. This tragedy can be overcome by so-called “bourgeois” or “anarchist” players which conventionally base their behavior on an external sign. However, it is not a-priori clear how such behavior could evolve through natural selection alone. In this paper it is shown, through simulations, that it can be the result of a more general strategy by which adaptive agents learn and establish a globally shared conventional code.

1. Introduction
The Hawk-Dove game (Maynard Smith & Price, 1973), also known as the snowdrift game or the chicken game, is used to study a variety of topics, from the evolution of cooperation (Doebeli et al., 2004) to nuclear brinkmanship (Russell, 1959). In the game, two players compete over a resource. They can choose between two actions named ‘hawk’ and ‘dove’. If both players play dove then they share the resource. If both play hawk then they share the resource minus a fighting cost. Hawks receive the complete resource when playing against doves. Doves can only thrive if the fighting cost exceeds the reward. Hawks can also take advantage of them. In sum, naively cooperative doves are destined to be exploited by aggressive hawks.

This ‘paradox of cooperation’ was termed the ‘tragedy of the commune’ by Doebeli.

The game can be extended with an uncorrelated asymmetry by informing players about whether they are the first or the second player. If players have equal chances of being first, then they may decide to play hawk when first and dove when second (or vice versa). This strategy was called the ‘bourgeois’ (or ‘anarchist’) strategy by Maynard Smith, and is an evolutionary stable strategy (it outperforms hawks and doves). This shows that full cooperation (in the sense of the Hawk-Dove game) can evolve if players can rely on a fair (but otherwise arbitrary) external sign.

It is another question how the bourgeois strategy could evolve. This crucially depends on what it means to be the first or the second player. An obvious choice is arriving first at a resource or territory. For example, territorial disputes between male speckled wood butterflies (Pararge aegeria) in England are resolved according to the bourgeois strategy (Davies, 1978). In this case, it is conceivable that the behavior is encoded in the hereditary material and the result of a mutation. This is much less so for other arbitrary signs. Consider for instance a label displaying the letters ‘private property’. The meaning of such a sign cannot be genetically encoded, it is cultural. This means that it needs to be learned by new (offspring) players from other (parent) players. Since learning can be costly, it is not a-priori clear if this is possible in an evolutionary context. Furthermore, learning presupposes that a stable cultural convention is established and available for learning, which requires further explanation.

In this paper it is shown that ‘bourgeois-like’ behavior can evolve as a result of a more general ‘coding’ strategy which allows agents to couple meanings (actions) to signals through learning. This strategy is

---

1 Although the same butterflies were reported to behave differently at other locations.
more general in the sense that it also supports ‘anar-
chistic’ behavior, or any of the other (pure) strategies.
It is also more general because it works for any signal,
not only for those designating ownership or being first.
Therefore it does not require the signals to be encoded
in the hereditary material, but only the capacity to
couple arbitrary signals to meanings, together with an
appropriate learning strategy. As is discussed in this
paper, under certain conditions such a capacity allows
coding players to beat doves and hawks by establish-
ing a cultural convention by which they can coordinate
their actions as a species.

2. The Hawk-Dove game

The payoff matrix for the Hawk-Dove game as consid-
ered in this paper is given in table 1. The symbols
r and c stand for the resource value and fighting cost
respectively, with $r \leq c$. These are the same in every
game. Each row (column) in the table corresponds
to a possible action of the first (second) player. For
example, the first entry in the first row and column
indicates that if both players play ‘hawk’, then each of
them receives a payoff of $b + (r - c)/2$. If however
the second player plays ‘dove’, then the first player receives
the payoff $b + r$ (first row, second column) whereas the
second player only receives a payoff $b$ (second row, first
column) etc. In the remainder of the paper, the off-
set value $b$ is set to $-(r - c)/2$ so that all payoffs are
non-negative.

<table>
<thead>
<tr>
<th></th>
<th>hawk</th>
<th>dove</th>
</tr>
</thead>
<tbody>
<tr>
<td>hawk</td>
<td>$b + (r - c)/2$</td>
<td>$b + r$</td>
</tr>
<tr>
<td>dove</td>
<td>$b$</td>
<td>$b + r/2$</td>
</tr>
</tbody>
</table>

Table 1. Payoff matrix for the Hawk-Dove game. The pos-
sible actions of the first player are listed in the first column
of the table. This player will receive a payoff as specified
in the table for each possible action of the second player.

In this section, three types of players (strategies) are
considered: hawks (always playing the action hawk),
doves (always playing the action dove) and bourgeois
(play hawk when first, dove when second). The ex-
pected payoff for a player of a particular type depends
on the strategies used by his opponents in games. Let
$p_h$, $p_d$ and $p_b$ be the fraction of hawks, doves and bour-
geois in the population respectively. It is assumed
that these fractions correspond to the probabilities to
meet an opponent of each type. If all players have
equal chances of being first then the expected payoffs
per game for each strategy are:

$$
\langle \mu_d \rangle = b + (1 - p_h + p_d)r/4
$$
$$
\langle \mu_h \rangle = b + (3 - p_h + p_d)r/4 - (1 + p_h - p_d)c/4
$$
$$
\langle \mu_b \rangle = b + (2 - p_h + p_d)r/4 - p_h c/2
$$

By setting $p_d = 1 - p_h$ in these expressions, one ob-
tains the standard result that neither the hawk nor
the dove strategy are evolutionary stable strategies
(ESS’s) since both of them can be invaded by the other
until the equilibrium ratio $p_h = r/c$ is reached. Both
strategies can also be invaded by the bourgeois strat-
agy, which is an ESS with respect to the others. This
can also be seen from the phase plot of the replicator
system that follows from the expected payoffs as shown
in Figure 1 for $r = 2, c = 3$ (the offset parameter $b$
only affects the time scale of the dynamics and hence
does not influence the plot). The state in which there
are only bourgeois type players is marked as ‘$P_b = 1$’.
It is the only asymptotically stable state.

![Figure 1. Phase plot of the Hawk-Dove-Bourgeois game](image)

3. The Coding Strategy and The Effect
of Population Turnover

From the previous it follows that if natural selection
acts upon a mixture of hawk, dove and bourgeois players
then the bourgeois strategy prevails. This result
has been known for a long time. However, it is based
on the assumption that offspring of bourgeois players
automatically use the same convention, that is,
play hawk when first and dove when second. In other
words, it assumes that the bourgeois strategy is en-
coded in the hereditary material of players. It is not
obvious how this could be established for arbitrary signals. Investigating the consequences of relaxing this assumption also is interesting in its own right. This is the topic of the remainder of the paper.

If the convention is not passed on to offspring through the hereditary material, then it must be learned. We therefore introduce a new strategy, the ‘coding strategy’. Players endowed with this strategy have the capacity to code the strategy. Players endowed with this strategy have the hereditary material, then it must be learned. If the convention is not passed on to offspring through the topic of the remainder of the paper.

Assumption also is interesting in its own right. This is shown to perform well in similar tasks (Catteeuw et al., 2011). First and Second) and the available responses or actions are \{F,S\} (for Hawk and Dove). For each possible combination \((s,a)\) of a signal \(s\) and an action \(a\), coding agents keep a score \(\phi(s,a)\), initially set to a fixed (genetically encoded) value \(\phi_i\). When an agent receives signal \(s^*\) in a game, it chooses action \(a^*\) from all available actions with probability proportional to \(\phi(s^*,a^*)\). If after the game the agent receives payoff \(\mu\), scores are updated as follows:

\[
\phi(s^*,a^*) \leftarrow \lambda \phi(s^*,a^*) + \mu,
\]

\[
\phi(s^*,a \neq a^*) \leftarrow \lambda \phi(s^*,a),
\]

\[
\phi(s \neq s^*,a) \leftarrow \phi(s,a).
\]

The parameter \(\lambda > 0\) is a discounting factor and is also encoded in the hereditary material of coding agents. Initially, coding agents explore all different actions with equal probability. As they play more games and apply the above update rules, the score of one action per signal will approach the real expected payoff for that action, while all other scores approach zero. Note that the action upon which the agent converges is not necessarily the one yielding the highest expected payoff, although the probability that it is approaches one when the initial score \(\phi_i\) approaches infinity and/or the discounting factor \(\lambda\) approaches zero. However, the time to converge upon a deterministic behavior then approaches infinity as well, so that there is a trade-off between ‘exploration’ and ‘exploitation’ (see also Figure 2). This trade-off puts a limit on the capacity of individual agents to learn the optimal behavior. In consequence, just by playing games, a population consisting solely of coding agents will not necessarily be able to establish a fully shared convention such as all agents deterministically playing the bourgeois strategy. This is because, as agents cool down or “grow old”, there is always a probability that they settle on different strategies.

It is known that children play a crucial role in the establishment of new natural languages (see e.g. (Senghas & Coppola, 2001; Verhooft & de Boer, 2011)). If a population turnover is added to the model, that is if “old” coding agents are replaced by new ones, exploration continues. Furthermore, new agents will tend to pick up and hence reinforce emerging conventions. This might allow a population of coding agents to establish a shared convention after all. The effectiveness of this mechanism was confirmed through simulations of which the results are shown in Figures 3 and 4. Note that these results depend on the rate at which agents are replaced in the population. This is investigated in more detail in the following.

4. Selection

Selection is brought into the model by replacing agents based on their success in games instead of randomly. We consider a finite population of \(N = 100\) agents undergoing natural selection according to a Moran process (Moran, 1962; Nowak, 2006). Each simulation run, the population is initialized to contain a variable fraction \(R \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}\) of coding agents. The rest of the population consists of 100 \((1-R)\) hawks and 100 \((1-R)\) \((1-r/c)\) doves, with \(r = 2\) and \(c = 3\). Between each replacement step, agents are randomly selected to play a total of \(\tau\) games, with \(\tau \in \{500, 1000, 2000, 4000\}\) (that is, on average, each agent plays \(2\tau/N\) games before a replacement occurs). Simulations were run for \(\lambda \in \{0.99, 0.95, 0.9, 0.8\}\) and \(\phi_i \in \{1, 10, 100\}\). For each combination of parameter values, 100 simulation runs were performed, and the fraction of runs that leads to a population consisting exclusively of coders after a maximum of \(2\times10^4\) games was recorded. Results are shown in Figure 5. From the figure, it can be seen that coding agents approach the optimal behavior as defined by the behavior exhibited by bourgeois (or anarchist) agents under a wide range of conditions, provided that agents have enough time to learn between successive replacement or selection steps. Only when learning is too greedy (corresponding to high values of \(\lambda\) and low values of \(\phi_i\) towards the top right of the figure) do they not have a selective advantage over hawks and doves, but also no disadvantage, as in these cases the probability that coders take over the population simply approaches their initial relative abundance \(R\).
5. Discussion and Conclusion

Although it has long been known that the tragedy of the commune can be overcome by players adopting a conventional strategy such as ‘bourgeois’ and ‘anarchist’ players, to our knowledge it was not investigated before how such strategies could evolve. The standard explanation is differential reproduction (mutation and selection). This implies that the convention is encoded in the hereditary material. However, because a convention, like language and culture, is arbitrary, it is not a-priori clear how this is possible. It was shown that the tragedy of the commune can also be overcome by “coding players”. Coding players learn association strengths between signals and actions that reflect expected payoffs and act accordingly. Interaction between such players induces a positive feedback loop between their preferences. This results in the amplification of otherwise arbitrary preferences at the population level, until a globally shared convention emerges with which coding players can coordinate their actions in a nearly-optimal way. In line with what is proposed in (Barbieri, 2008b), I propose to refer to this mechanism of evolution as evolution by natural conventionalization.

Crucially, it is the interplay between evolution by natural selection and evolution by natural conventionalization that determines the outcome of evolution in total. Without differential reproduction, that is without even population turnover, evolution stops and no conventionalization takes place. On the other hand, without conventionalization, the Major Transitions in macro-evolution would never have occurred. These transitions are characterized by increased degrees of coordination, for instance between cellular agents that became organized into multi-celled organisms (Maynard-Smith & Szathmáry, 1995). As in the Hawk-Dove game, coordination requires that a conventional code is available, and hence that conventionalization mechanisms are at work. It should therefore come as no surprise that the Major Transitions are all accompanied by the appearance of new, arbitrary codes, an example of which is the genetic code (Maynard-Smith & Szathmáry, 1995; Barbieri, 1998).

Like Barbieri, I conclude that natural selection and natural conventionalization are complementary mechanisms of evolution, the first accounting for the gradual transformation of existing species through differential reproduction and the second for the origin and fixation of absolute novelties at higher levels of organization. These mechanisms are not independent however, and their interplay must be taken into account in order to obtain a full understanding of evolution.

References


Verhoef, T., & de Boer, B. (2011). Language acquisition age effects and their role in the preservation and change of communication systems. Linguistics in Amsterdam, 4.
Conventional Coding in the Hawk-Dove game

Figure 2. Sampled behavior of coding agents learning by playing games with dove agents. The point at $(x = 1, y = 0)$ corresponds to pure hawk-like behavior. The opposite point at $(x = -1, y = 0)$ to pure dove-like behavior. The points at $(x = 0, y = 1)$ and $(x = 0, y = -1)$ correspond to pure bourgeois- and anarchist-like behavior respectively. Mixed behaviors are also possible since coder agents are not necessarily deterministic. The different point types correspond to the amount of games played. The more games played, and depending on the learning parameters $\lambda$ and $\phi_i$, the more deterministic the behavior becomes. The optimal behavior in this case is hawk-like, which is eventually reached when learning parameters are “favorable”, that is towards the left and bottom of the Figure.

Figure 3. Same as in Figure 2 but instead of playing against doves, the coding agents now learn by playing against other (learning) coding agents. Due to the exploration/exploitation trade-off, a convention is not always established, that is the agents do not always adopt the bourgeois or anarchist strategies, not even when learning parameters are “favorable” (see fig. 2)
Conventional Coding in the Hawk-Dove game

Figure 4. Same as in Figure 3, but now agents are randomly replaced in the population at some fixed rate. If learning parameters are favorable, a convention emerges and agents eventually adopt the bourgeois or anarchist strategy. The reason why convergence never appears to be complete is because new agents are constantly entering in the population and because these still need to adopt the convention.

Figure 5. Fraction of simulation runs that lead to an end state of only coders for different values of the learning parameters $\lambda$ and $\phi_i$. The X-axis indicates the initial fraction of coding agents $R$. Values that are above the bisecting line ‘$y=x$’ indicate a selective advantage for coding agents over hawks and doves. Each curve is labeled with the number of games $\tau$ in between two successive (Moran) replacement steps. The curves labeled ‘$\infty$’ were obtained with bourgeois agents instead of coders, and indicate the best attainable behavior. Coding agents approach this optimal behavior if learning parameters are favorable and if there is enough time to learn between successive replacement events ($\tau$ large). The learning parameters determine the speed of learning which in turn interferes with how much time is available for learning, which explains why the optimal set of parameters now is no longer found simply towards the bottom left of the Figure.
Explorations in Texture Based Classification for Bacterial Infection in Zebrafish

G.A. den Heijer¹
P. van der Putten¹
A.E. Nezhinsky²
E.L. Benard²
A.H. Meijer²
P.J. Verbeek²

¹Imaging & Bioinformatics, Leiden Institute of Advanced Computer Science, Leiden University,
Niels Bohrweg 1, 2333 CA, Leiden, the Netherlands
²Molecular Cell Biology, Institute Biology Leiden, Leiden University,
Einsteinweg 55, 2333 CC, Leiden, the Netherlands

Keywords: tuberculosis, texture, zebrafish, classification

Abstract

Tuberculosis is a serious infectious disease caused by Mycobacterium tuberculosis bacteria that infect cells of the human immune system. These infected cells form clusters, called granulomas, which are the hallmark of this disease. Animal model systems have been used to understand the complex molecular interactions involved in tuberculosis infection. The zebrafish embryo model system is useful to study the early stages of tuberculosis infection and granuloma formation. In these transparent embryos, the clustering of infected immune cells into granulomas can be studied through fluorescence imaging and image analysis can generate parameters describing the progression of infection. Such analysis supports experiments that are conducted to reveal the underlying mechanisms of granuloma formation. For image analysis a number of different features need be probed. Next, the bias of these features and the subsequent data analysis need be understood. Texture can potentially be an important descriptor of infection patterns. Prior to large scale application we explore the analysis of groups of infected embryos with a number of standard texture features. In this manner we hope to obtain insight for the analysis of future experiments. Here we compare six groups of infected zebrafish, for which the start conditions were slightly different, using a two-sample t-test, clustering and classification methods. The results from our analysis provide further insight in balanced analysis of experimental data.

1. Introduction

Tuberculosis (TB) affects about a third of the world population and is caused by infection with the bacterial species Mycobacterium tuberculosis (Dye et al., 1999). The exact mechanisms of the progression of this disease are, to date, subject of investigation (Lesley and Ramakrishnan, 2008; Russell, 2011). The progress of the infection is usually halted before any serious damage is done. In fact, roughly 90% of all persons infected with TB never experience any symptoms but develop a latent infection where the bacteria become dormant. However, under certain conditions that weaken the immune system, such as malnutrition, immunosuppressive therapies, or HIV co-infection, the bacteria can be reactivated from dormancy and cause active disease (Gengenbacher and Kaufmann, 2012). M. tuberculosis has a unique method of surviving and replicating in its host: it infects the macrophages that try to kill it. From there it attracts and infects other macrophages. This eventually results in structures called granulomas, which consist of tight aggregates of infected and uninfected immune cells (Saunders and Britton, 2007; Russell, 2011; Gengenbacher and Kaufmann, 2012). These granulomas are considered as the hallmark of TB.

There is a range of features that can be used to describe granulomas in microscopy images. Considering the texture of the granuloma might be an effective approach to reveal differences between groups/organisms. We therefore wish to study these parameters with real data. The zebrafish is a good model system for the study of Tuberculosis; they can be obtained in large numbers, they can be
manipulated easily and they are transparent in their embryonic phase, so that the disease progression can be monitored (Prouty et al., 2003; Nezhinsky et al., 2010). Moreover, their immune system is very similar to that of humans, i.e. genes found to be involved in TB infection and progression are likely to be involved in human TB infection and progression as well. The system can also be explored from the perspective of the bacteria. For example, Stoop et al. (2011) have recently used zebrafish to identify mycobacterial genes involved in the initial stages of granuloma formation.

To study the granuloma infection, zebrafish have been infected with *Mycobacterium marinum*, a close relative to *Mycobacterium tuberculosis*. After infection with the bacteria, images are acquired with bright field and fluorescence microscopy. The images from fluorescence microscopy are used for the texture analysis.

The data that are evaluated for this study are from six infection experiments. In the experiments, the conditions differ (injection needles, bacteria cultures). All considered the experiments provide a good scope of what one can be expected from the injection experiments.

The images need to be analyzed so that the parameters can be extracted. To that end an automated framework is used (Nezhinsky et al., 2012). The analysis if this framework is focused on the spread and size of the granuloma. For this study, we consider the texture of the granuloma. This additional information about the granuloma may provide additional features for describing granuloma development and along that line contribute to better analysis of experimental data.

The aim of our exploration is to gain insight in the quality of the texture measures for the analysis of granulomas in zebrafish that arise from TB infection. Therefore, a framework and a statistical analysis strategy will be developed for simulating the tracking of differences in texture between groups (experiments). This setup can then be used for follow-up research. In addition the testing framework may prove useful for the assessment of additional types of texture descriptors.

The remainder of this paper is organized as follows. First the background of the research is explained. Next the experiments used will be explained and their results will be presented after that. Next, these results will be discussed, along with possible future work that might improve our insights in TB infection and progression. Finally, this paper will be summarized with a conclusion.

2. Related work

Zebrafish is used to gain insight in the infection progression and to obtain a better understanding of the exact mechanisms behind the disease. Until recently the analysis of the images from the infection was done manually; no suitable tools were available to conduct an analysis in a high-throughput fashion. An automated analysis will contribute to the objectivity of the results that are obtained.

A framework for automatic shape retrieval and granuloma detection was developed (Nezhinsky & Verbeek, 2012) so as to obtain objective, reproducible and robust analysis. This framework has been successfully applied to analyze the spatial distribution and size of the granulomas in zebrafish larvae over time. This work was focused on mutant bacteria strains and their effect on infection. Amongst other things it was observed that size and spread of the granulomas increases over time. To that end other features should be considered. In this study, therefore, the changes in granuloma texture in zebrafish infected with *Mycobacterium marinum*, are measured and analyzed.

3. Experiments and results

In this section, the experiments, as well as the observed results will be described. The focus of the exploration will be on texture descriptors and their value for classification of the groups. At the same time the groups are analyzed so that we can better understand the variation that is natural to these kinds of experiments.

3.1 Overall approach

In the study by Nezhinsky et al. (2012), the changes between mutant bacteria were considered and analyzed. In this study control groups are used to test and experiment on (cf. 3.2).

Zebrafish groups are regularly used to study the disease progression. One zebrafish group will be the reference group, used to observe the normal texture development in zebrafish infection. By classifying the texture of the granulomas we hope to learn more about the mechanism behind tuberculosis, as well as to establish whether texture is an appropriate feature for classification in this context. In a traditional classification approach a classifier would be built on known positive and negative examples, and then applied to the experiment data sets to identify the ones that are the least similar to the control group. However, in this study we use classification in a different manner. We build models for each reference-experiment combination, and a strong predictive power is seen as an indication of significant differences in texture. This is supplemented with basic univariate and cluster analysis for more explorative data mining.

3.2 Imaging

In each experiment, bright field and fluorescence images are taken from a batch of zebrafish infected with bacteria over consecutive days. For each experiment, a separate batch of zebrafish left uninfected. While no granulomas are present in these zebrafish, in the fluorescence images some background noise is observed. These images are used as an estimate of the level of background noise in the rest of the experiment.
The bright field images depict up to three zebrafish, aligned as in Figure 1. The software developed by Nezhinsky et al. (2012) detects the shape and location of the zebrafish. These are then used to distinguish actual spots in the corresponding fluorescence image from background noise.

![Figure 1. Bright field image corresponding to Figure 2.](image)

![Figure 2. A fluorescence image contrast and brightness have been adjusted for visualization purposes.](image)

### 3.3 Framework

The analysis framework developed by Nezhinsky et al. (2012) was used to detect the zebrafish in the bright field images and the granulomas in their corresponding fluorescent images. This framework was integrated in a new framework in which the shape and location of the granulomas are not considered. Instead, six basic texture descriptors (cf. 3.4) are calculated. The output of the new framework is a matrix of values of the six texture descriptors for every granuloma that is detected. Table 1 shows the amount of granulomas detected for each experiment. As can be seen, the distribution of the granulomas is very imbalanced. This will be discussed later.

<table>
<thead>
<tr>
<th>EXPERIMENT</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRANULOMAS</td>
<td>516</td>
<td>1398</td>
<td>5712</td>
<td>90</td>
<td>2164</td>
<td>2479</td>
</tr>
</tbody>
</table>

### 3.4 Texture descriptors

Six basic texture descriptors have been used to study changes in granuloma textures in experimental zebrafish groups (Gonzalez & Woods, 2008). These six descriptors will be explained briefly.

\[
\max(p_{ij})
\]

The maximum probability is the probability of the most frequent pixel intensity transition.

\[
\sum_{i=1}^{K} \sum_{j=1}^{K} (i-m_{i})(j-m_{j})p_{ij}; \sigma_{r} \neq 0; \sigma_{c} \neq 0
\]

where \( K \) is both the row and column length of \( G \), \( m_{i} \) and \( m_{c} \) are the mean row and column values, and \( \sigma_{r}^{2} \) and \( \sigma_{c}^{2} \) are their variances, denoted as

\[
m_{i} = \frac{\sum_{j=1}^{K} p_{ij}}{p_{ij}}; \quad m_{c} = \frac{\sum_{i=1}^{K} p_{ij}}{p_{ij}}
\]

\[
\sigma_{r}^{2} = \frac{\sum_{i=1}^{K} (i-m_{i})^{2} p_{ij}}{p_{ij}}; \quad \sigma_{c}^{2} = \frac{\sum_{j=1}^{K} (j-m_{j})^{2} p_{ij}}{p_{ij}}
\]

Correlation is a descriptor for the correlation of each pixel to its neighbor. Note that the standard deviations may not be zero. However, this is only the case if the contrast in the image is zero.

\[
\sum_{i=1}^{K} \sum_{j=1}^{K} (i-j)^{2} p_{ij}
\]

Contrast is a descriptor for the contrast for each pixel to its neighboring pixel. The value of this descriptor tends to be larger for larger images.

\[
\frac{\sum_{i=1}^{K} \sum_{j=1}^{K} p_{ij}^{2}}{\frac{1}{2} \sum_{i=1}^{K} \sum_{j=1}^{K} |i-j|}
\]

This is the measure for uniformity in the image.

Homogeneity measures the spatial closeness of the pixels to the diagonal.

\[
-\sum_{i=1}^{K} \sum_{j=1}^{K} p_{ij} \log p_{ij}
\]

Entropy is a measurement for the variability of the image. In this equation, \( \log p_{ij} \) is defined as 0 if \( p_{ij} = 0.0 \). As with contrast, the value of this descriptor tends to be larger for larger images, albeit not nearly as dramatically due to its logarithmic nature.

In order to obtain these measurements easily, a co-occurrence matrix of the images of each individual spot is constructed. This co-occurrence matrix, referred to as \( G \), will be filled with counts of pixel intensity transition from pixels and their neighbor to the right. Using \( G \), one can now calculate the probability for a specific transition for two given pixel intensities denoted as \( p_{ij} = g_{ij}/n \), where \( g_{ij} \) is the value in \( G \) at the \( i \)th row and the \( j \)th column and \( n \) is the amount of transitions (or the amount of pixels that have a neighboring pixel to the right) in the image. For example, if in an image of 5 by 5 pixels...
the transition of a pixel intensity from 2 to 1 occurs twice, the probability of this transition is $p_{21} = g_{21}/(5*(5-1)) = 0.1$.

### 3.5 Univariate analysis

All six texture descriptors have been calculated for the granulomas detected in the images. The results have been statistically analyzed.

The Shapiro-Wilk test for normality has been performed for every texture descriptor in all experiments in order to assess whether the data found is normally distributed. None of the groups passed this test for any of the descriptors. However, when calculating contrast, the resulting value tends to be higher for larger images. Since this value has a quadratic growth with respect to size, the contrast values are not likely to be normally distributed. The same applies to entropy, albeit to a much lesser extent, because the values have a logarithmic growth with respect to size. It would be desirable to correct both these data before doing a normality test. This is not yet completed for this dataset.

To get an impression of what we can expect in the multivariate analysis, univariate analysis is performed.

After the normality test, a two-sample t-test was done between the reference group and each individual experimental group. If the t-test indicates that the reference group and an experimental group have different means, then this is an indication that the experimental group is different from the reference group.

Table 2 shows that the reference group has a significantly different mean than D and F for all attributes. C differs significantly from A for all descriptors but correlation. B and E are only significantly different with respect to A for correlation, contrast and entropy. It is, however, also possible that the contrast and entropy alone are sufficient for good clustering and classification results. These observations suggest that the data can be clustered and classified with good results.

<table>
<thead>
<tr>
<th>DESCRIPTOR</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX. PROB.</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>CORRELATION</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>CONTRAST</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UNIFORMITY</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>HOMOGENEITY</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>ENTROPY</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HYPOTHESIS FOR EXPERIMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
</tr>
</tbody>
</table>

As can be seen in Table 3, the resulting clusters for the two cluster algorithms are remarkably similar. The clusters in Table 4 are also very similar, albeit to a lesser extent.

<table>
<thead>
<tr>
<th>EXP.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>49</td>
<td>51</td>
<td>0</td>
<td>48</td>
<td>51</td>
</tr>
<tr>
<td>B</td>
<td>44</td>
<td>55</td>
<td>1</td>
<td>43</td>
<td>54</td>
</tr>
<tr>
<td>C</td>
<td>24</td>
<td>48</td>
<td>28</td>
<td>23</td>
<td>40</td>
</tr>
<tr>
<td>D</td>
<td>29</td>
<td>28</td>
<td>43</td>
<td>30</td>
<td>24</td>
</tr>
<tr>
<td>E</td>
<td>51</td>
<td>45</td>
<td>4</td>
<td>51</td>
<td>42</td>
</tr>
<tr>
<td>F</td>
<td>50</td>
<td>38</td>
<td>3</td>
<td>58</td>
<td>36</td>
</tr>
</tbody>
</table>

### 3.6 Clustering experiments

Next, we attempted to cluster the values of the texture descriptors for the reference group and each experimental group in two groups. As with the two-sample t-test, sufficient separation between the reference and experimental groups would indicate an effect on granuloma development in the experiment.

We first used cluster algorithms to divide all data into two and three clusters. Clustering to two clusters can tell us which groups are similar to other groups. Clustering to three groups can tell us how the data is spread further. The results of this can be seen in Table 3 and 4.

Table 3. Clustering all data simultaneously to two clusters using Expectation Maximization (E.M.) and simple K-means.

Table 4. Clustering all data simultaneously to three clusters using Expectation Maximization (E.M.) and simple K-means.
Table 5. The percentages correctly clustered instances after clustering the control group and each individual experimental group to two clusters using E.M. and simple K-means.

<table>
<thead>
<tr>
<th></th>
<th>E.M.</th>
<th>SIMPLE K-MEANS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP.</td>
<td>% CORRECTLY CLUSTERED INSTANCES</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>53</td>
<td>53</td>
</tr>
<tr>
<td>C</td>
<td>70</td>
<td>68</td>
</tr>
<tr>
<td>D</td>
<td>68</td>
<td>69</td>
</tr>
<tr>
<td>E</td>
<td>50</td>
<td>51</td>
</tr>
<tr>
<td>F</td>
<td>56</td>
<td>56</td>
</tr>
</tbody>
</table>

Table 6. The percentages of instances clustered to the control group and to each individual experimental group to two clusters after clustering using E.M. and simple K-means.

<table>
<thead>
<tr>
<th></th>
<th>E.M.</th>
<th>SIMPLE K-MEANS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP.</td>
<td>% INSTANCES ASSIGNED TO CLUSTER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>22.52</td>
<td>26.02</td>
</tr>
<tr>
<td>B</td>
<td>26.54</td>
<td>24.92</td>
</tr>
<tr>
<td>A</td>
<td>11.61</td>
<td>38.54</td>
</tr>
<tr>
<td>C</td>
<td>31.84</td>
<td>18.02</td>
</tr>
<tr>
<td>A</td>
<td>47.52</td>
<td>0.17</td>
</tr>
<tr>
<td>D</td>
<td>31.35</td>
<td>20.96</td>
</tr>
<tr>
<td>A</td>
<td>22.76</td>
<td>25.82</td>
</tr>
<tr>
<td>E</td>
<td>24.40</td>
<td>27.01</td>
</tr>
<tr>
<td>A</td>
<td>25.21</td>
<td>23.87</td>
</tr>
<tr>
<td>F</td>
<td>20.27</td>
<td>30.65</td>
</tr>
</tbody>
</table>

Table 5 suggests that clustering the control group with experimental groups C and D results in a relatively low error rate. However, due to the imbalanced amount of data points, these numbers may not be reliable. In order to observe the within-group spread of the clusters, we also clustered the reference group with each individual experiment. The results of this can be seen in Table 6, which suggests a clear distinction in cluster composition for clustering the control group with the experimental groups C and D.

Taken together, the results of the clustering experiments suggest that experimental groups C and D are similar and different from reference group A. Also, the experimental groups B, E and F do not seem to be clearly distinguishable from the reference group. These results do not correspond fully to that of the univariate analysis, where the reference group has a significantly different mean compared to D and F for all attributes, and C for all attributes except correlation. According to the results from clustering, F is not different from the control group.

3.7 Classification experiments

Lastly, an attempt was made to classify the values of the texture descriptors for the control group and each experimental group. Once again, successful classification on experimental groups would indicate that the experiment had effect on the granuloma development. The number of correctly classified instances is taken as the measure of how successful the classification was.

Directly applying classification on the data negatively influenced the results, because the number of granulomas than in experimental groups differed. Therefore, before classifying the reference and the other groups, we downsampled the larger of the two.

We repeated this experiment, but this time we upsampled the smaller group instead.

As can be seen in Figure 3 and 4, the classification results using downsampled and upsampled data differ considerably. While classification on upsampled information uses all available information, it might give results that are too optimistic due to repeated data points. On the other hand, downsampled information does not use all available information, but the results are reliable. We therefore chose to focus on the results for the downsampled information.

Table 7. The mean and standard deviations of the percentages correctly classified instances for classification of the control group with each experimental group using Id3, J48, Random Forest, Naïve Bayes and a logarithmic classifier.

<table>
<thead>
<tr>
<th></th>
<th>% CORRECTLY CLASSIFIED INSTANCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP.</td>
<td>MEAN</td>
</tr>
<tr>
<td>B</td>
<td>63.44</td>
</tr>
<tr>
<td>C</td>
<td>70.08</td>
</tr>
<tr>
<td>D</td>
<td>68.03</td>
</tr>
<tr>
<td>E</td>
<td>68.56</td>
</tr>
<tr>
<td>F</td>
<td>69.55</td>
</tr>
</tbody>
</table>

Table 7 shows that all experimental groups are classified better than at random. Group C is classified slightly better than the rest, except for group B,
which is classified clearly less well. This, again, does not correspond fully to previous results; most notably, we see that group F is clearly different from the control group, and the univariate analysis showed this as well, but clustering did not.

4. Future work

For this study we have analyzed the texture changes between several groups of infected zebrafish. The statistical analyses suggest that groups C, D and F are more different from the reference group. The results between these analyses are not corresponding and conclusive in all cases. To that end more research is required as well and analysis of the composition of the groups so as to determine the optimal combination of analyses for the most reliable results.

The next step is to further test this framework with a larger body more experimental sets; ones that have known differences with the reference set. The results of the texture analysis have been developed independent from that of analysis based on spread and size; a comparison might reveal interesting differences and correspondences. The texture analysis can also be expanded with advanced texture measurements.

A next step is to study the texture changes over time. To that end, new data will be made available that enable this study. Repeating the experiments and analyses with this new data will be beneficial for the outcomes yet obtained.

5. Conclusion

In support of a better understanding of the mechanisms behind TB, we have developed a new framework to obtain granuloma texture measurements from infected zebrafish. The textures of the granulomas of the reference group were compared to other groups in order to test the experimental analysis.

The comparison was done using a univariate analysis, clustering and classification. The results of these comparisons do not correspond in all cases. Experiments with induced differences, e.g. gene knock-out experiments will help to gain insight in using texture and texture change to study/detect changes in granuloma formation. The research can also be expanded with additional texture descriptors.

Acknowledgements

This research is partially supported by the SmartMix Program and by the Cytron Program.

References


Estimation of temporal lags for the inference of gene regulatory networks from time series

Miguel Lopes
mlopes@ulb.ac.be

Patrick Meyer
pmeyer@ulb.ac.be

Gianluca Bontempi
gbonte@ulb.ac.be

Machine Learning Group, Computer Science Department, Faculty of Sciences, ULB, Brussels

Keywords: gene regulatory network inference, time series data, mutual information, partial correlation

Abstract

Temporal precedence is normally assumed to be essential for defining causation and it is undoubtedly one of the most important clues that can be used to distinguish causal from other types of association. However, most of the existing approaches to network inference from gene expression data do not take advantage of the temporal nature of data. We propose here an intuitive manner of introducing the temporal dependency within three existing methods of network inference. The added value of taking into account temporal dependency is validated by a number of experiments on simulated datasets.

1. Introduction

Microarray technology and more recently high throughput sequencing allow us to measure various types of biological processes that occur inside the cell. These measures shed light on the interaction and the regulation activities of networks of genes. However an accurate inference of these networks from gene expression data is still an open challenge for the bioinformatics community (Marbach et al., 2010). The largest majority of existing approaches to network inference assume stationary data and does not take advantage of the temporal information. For instance well-known approaches like CLR (Faith et al., 2007) and MRNET (Meyer et al., 2007) adopt measures like mutual information to infer undirected dependencies between genes on the basis of steady-state data.

Though some work addressed the analysis of the temporal characteristics of gene expression time series (Sima et al., 2009; Zoppoli et al., 2010) there is no standard way to extend state-of-the-art static algorithms to the temporal dimension.

This paper proposes a lag estimation procedure as an easy and intuitive manner of reusing conventional static algorithm for time series expression data. A regulatory interaction between genes is in fact not instantaneous, and a temporal lag occurs between the time a regulatory gene is expressed and the time its target is expressed.

The contribution of this paper is twofold. First we propose an algorithm, based on the maximization of the temporal correlation, which estimates the lag from observed data and takes advantage of this information by improving the accuracy of three static methods in inferring the directionality of the dependence. Second, we assess the quality of the algorithm by performing a series of experiments on a number of artificially generated datasets. In particular we used datasets proposed by the DREAM4 network inference challenge, made available by the GeneNetWeaver framework (Schaffter et al., 2011) and datasets generated by the software Netsim (Di Camillo B., 2009).

The idea of extending a steady-state algorithm by taking advance of temporal information is not new and was already presented in the TimeDelay-ARACNE approach (Zoppoli et al., 2010). What is specific to our approach is that we adopt a much simpler and intuitive way to detect time-delay dependency and that we apply it to other methods than ARACNE, like MRNET, CLR and partial correlation networks.

Preliminary work. Under review for BeneLearn and PMLS 2012. Do not distribute.
2. Approaches to gene regulatory network inference

Several approaches have been proposed to infer networks of gene regulatory models (Bansal et al., 2007). These include Bayesian networks, approaches based on ordinary differential equations and methods based on measures of dependency between pairs of genes, like Pearson correlation measure, mutual information or partial correlation. In this section we will review two methods based on the mutual information, MRNET and CLR, and one method based on partial correlation, presented in (Whittaker, 1990).

2.1. Network inference based on partial correlation

The partial correlation coefficient describes the dependency between two variables after the effects of all other variables are removed. The partial correlation between two variables \( X_i \) and \( X_j \), given a set of controlling variables \( Z \) is defined as the correlation between the residuals \( r_i \) and \( r_j \) resulting from the linear regression on \( X_i \) and \( X_j \) using \( Z \), respectively:

\[
\rho(X_i; X_j | Z) = \rho(r_i, r_j) \rho(z, z) \]

The partial correlation coefficient can be computed recursively, or by inverting the covariance matrix, both methods requiring a time complexity of \( O(n^3) \) where \( n \) is the number of elements in \( Z \). Mathematical details concerning these properties can be found in the literature, eg. in (Kontos, 2009).

The use of partial correlations to assess the dependency between two pair of genes has been proposed in the literature in (de la Fuente et al., 2004) or in (Schäfer & Strimmer, 2005), (Opgen-Rhein & Strimmer, 2007). The undirected graph that results from the partial correlation network is commonly known as a Graphical Gaussian Model (GGM). A missing link between two vertices (genes) in a GGM indicates that they are conditionally independent. Problems regarding the computation of the partial correlation coefficients arise when we are confronted with a very large number of variables (genes) and a relative small number of observations. This has lead to regularized strategies for estimating the covariance matrix and its inverse (Kontos, 2009). In the experimental assessment presented in this paper the computation of the partial correlations is made by the R package GeneNet (Schäfer et al., 2007).

2.2. Network inference based on mutual information

The mutual information between two discrete random variables \( X_i \in X \) and \( X_j \in X \) is defined as:

\[
I(X_i; X_j) = \sum_{x_i \in X, x_j \in X} p(x_i, x_j) \log \left( \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \right)
\]

where \( p(x_i, x_j) \) is the joint probability distribution function of \( X_i \) and \( X_j \) and \( p(x_i) \) and \( p(x_j) \) are the marginal probability distribution functions of \( X_i \) and \( X_j \), respectively. The concept of mutual information between two variables measures the amount of information these variables share. If two variables \( X_i \) and \( X_j \) follow a normal distribution, the mutual information between them is given by \( I(X_i; X_j) = -\frac{1}{2} \log(1 - \rho^2) \), where \( \rho \) is the Pearson’s correlation coefficient between \( X_i \) and \( X_j \), for details see (Meyer, 2008).

We consider here two network inference algorithms based on information theory, MRNET and CLR. These algorithms use the value of the mutual information \( I(X_i; X_j) \) between two variables \( X_i \) and \( X_j \) as the scoring metric to calculate their statistical dependency.

The CLR (Faith et al., 2007) algorithm works the following way: for each pair of variables \( X_i \) and \( X_j \) it computes their mutual information \( I(X_i; X_j) \) and a score \( z_{ij} = \sqrt{z_i^2 + z_j^2} \), where \( z_i = \max \left( 0, \frac{I(X_i; X_j) - u_i}{\sigma_i} \right) \). \( u_i \) is the mean of the distribution of the mutual information \( I(X_i; X_k) \), \( k = 1, ..., n \) and \( \sigma_i \) is the respective standard deviation.

The MRNET algorithm (Meyer et al., 2007) is based on the minimum redundancy/maximum relevance (mRMR) feature selection method (Peng et al., 2005). For a given target \( Y \), this method ranks sequentially the set \( V \) of inputs as follows: the first selected variable \( X_1 \) is the one that has the highest mutual information towards \( Y \). This variable is added to the set \( S \) of selected variables. The next variable \( X_j^{mRMR} \) is:

\[
X_j^{mRMR} = \arg \max_{X_j \in V \setminus S} (u_j - r_j)
\]

which maximizes a score \( s_j = u_j - r_j \), where \( u_j \) and \( r_j \) are defined as:

\[
u_j = I(X_j; Y) \\
r_j = \frac{1}{|S|} \sum_{X_k \in S} I(X_j; X_k)
\]
The first term $u_j$ represents the relevance of $X_j$ towards $Y$ and the second term $r_j$ represents the redundancy of $X_j$ with the previously selected variables in $S$. The selected variable is added to the set $S$ and the procedure is repeated for all the remaining variables in $V \setminus S$. The goal of this algorithm is to rank, for a target variable $Y$, sets of variables that taken together function as good predictors of $Y$ and that are information theoretically different between them (not redundant). What is common to CLR and MRNET is that they both return a score for each pair of genes and use such score to infer a network of dependencies. In other words, the higher the score between two genes, the higher the confidence in the prediction that a regulatory link exists between them.

2.3. Analysis of gene expression time series data

Unlike the majority of gene regulatory network inference methods, which do not consider the time series nature of the gene expression data, some approaches tried to take it into account (Sima et al., 2009).

The pairwise Pearson correlation coefficients are used to infer a regulatory network, using time series data coming from rat fibroblast cell lines in (Remondini et al., 2005). In (Gupta et al., 2006), time series allow to infer the directionality of gene interactions: a slope metric is proposed to represent the interaction between two genes. It is assumed that the expression of gene $g_i$ is linked to the expression of gene $g_j$ through a linear relationship $g_i = a_{ij} + b_{ij}g_j$. If $|b_{ij}| > |b_{ji}|$ the directionality of the interaction is interpreted as being from $g_i$ to $g_j$, as it is believed that a small change in the regulatory gene results in a larger variation on the regulated gene. In (Kwon et al., 2003) the gene expression time series are converted into strings of three possible events: 'rising', 'constant' and 'falling'. For each pair of genes their respective strings are aligned in a way that maximizes their event sequence similarity. Another approach is found in (Bickel, 2005): for each gene a matrix of expression values is created where the rows are the values of the expression levels translated left and right, representing a negative and positive lag in time. The missing values of the line are padded with the value next to them, the first or last of the series, if the translation is made to the right or the left, respectively. Then, all the possible pairs of all rows of all the matrices are tested for a coregulation criteria, which is the correlation coefficient. In (H Ma & C Chan, 2006) there is a discretization of gene expression values into three possible status, 'high', 'average' and 'low', and a search for dependencies between $g_i$ being 'high' at one time point and $g_j$ being 'high' at the next time point. A Bayesian approach was used to discover, for each gene $g_i$, a set of parents that are good predictors of it, using threshold values for the probability of the activation or repression of $g_i$, when a parent is expressed (Barker et al., 2006). Dynamic Bayesian networks, which extend the concept of Bayesian Networks by introducing an additional set of variables - the original ones with a time lag of 1 point relative to the reference -, have been proposed to infer gene regulatory networks, eg. in (Dojer et al., 2006). First order vector autogressive (VAR) models are applied to network inference in (Shimamura et al., 2009) and (Fujita et al., 2007), where the expression of gene $g_i$ at a given time point $t$ is assumed to be linearly dependent of all other genes at a time $t - 1$. In (Zoppoli et al., 2010) the ARACNE algorithm is adapted to consider time series data. Here, an optimal time lag between pair of genes is found by using empirical copula. Time series of whole-genome microarrays of the cyanobacterium *Synechocystis* sp. strain PCC 6803, after it is exposed to light, are used in (Schmitt et al., 2004), where a group of significantly correlated genes is identified - each pair of genes is lagged in time and its optimal lag and correlation value are found. These last two approaches follow a similar strategy to the one proposed in this paper.

3. The lag discovery

We propose an algorithm to extend existing gene regulatory network inference algorithms by estimating the temporal lags that characterize regulatory interactions. We assume that gene regulations are not instantaneous, and that a gene $g_i$ is regulated by a gene $g_j$, with a temporal lag $l$, possibly different from zero. For this reason it makes sense to predict the existence of this interaction by associating the values of $g_i$ at the time $t$ with the values of $g_j$ at the time $t - l$.

In order to discover the lags that characterize gene interactions, we followed an approach that consists in finding, for each pair of genes $g_i$ and $g_j$, the lag that returns the highest score between their lagged expression values. This lag value is given by:

$$\text{lag}_{i,j} = \arg \max_{l \in L} S(g_i, g_j, l)$$

where

$$S(g_i, g_j, l) = \begin{cases} S(g_i[1 + l : N]; g_j[1 : N - l]) & \text{if } l \geq 0 \\ S(g_i[1 : N + l]; g_j[1 - l : N]) & \text{if } l < 0 \end{cases}$$

$S$ is the scoring metric - which is the Pearson correlation coefficient -, and the lag $l$ takes values within the
Estimation of temporal lags for the inference of gene regulatory networks

interval \( L = \{ -l_{\text{max}}, \ldots, l_{\text{max}} \} \) where \( l_{\text{max}} \) is set by the user.

After the estimation of a lag for each pair of genes, we incorporate this information within a given network inference methods to improve its performance. In the following we will denote the network inference using the lag estimation as lag discovery network inference, in opposition to the conventional static network inference.

The goal of this paper is to assess the impact of this transformation on the performance of the network inference methods, in comparison with the alternative static approach to time series data. One advantage of the lag discovery procedure is the fact that it provides information about the directionality of gene links and may complement undirected network inference algorithms. For instance if for a pair of genes \( g_i \) and \( g_j \), the \( lag_{i,j} \) is positive we will set the direction of the interaction from \( g_j \) to \( g_i \).

3.1. Extending MRNET and CLR with the lag estimation

The mutual information algorithms MRNET and CLR take as input a symmetric square matrix of size \( n \times n \), where \( n \) is the number of genes and the \([i,j]\) element of the matrix is \( I(g_i; g_j) \), the mutual information between genes \( g_i \) and \( g_j \).

The application of the lag discovery procedure with the mutual information based algorithms MRNET and CLR is straightforward. Here, the mutual information values that are given as input are the ones that result from the lagging of each pair of genes, by their respective lag: \( I(g_i; g_j) = I(g_i(t); g_j(t - lag_{i,j})) \)

The mutual information was estimated using the Pearson’s correlation, as it was demonstrated to be a good estimator in (Olsen et al., 2009), and to provide a good comparative basis with the partial correlation method.

3.2. Extending the partial correlation approach with the lag estimation

The extension of the partial correlation (henceforth denoted PC) approach with lag discovery is described in what follows. For each gene \( g_i \), a gene expression matrix is created with the expression of all the genes lagged with respect to gene \( g_i \).

If, for time series of \( N \) points, the \( lag_{i,j} \) between genes \( g_i \) and \( g_j \) is negative, the expression vector corresponding to gene \( g_j \) will start at time \( t = 1 - lag_{i,j} \) and the last \( -lag_{i,j} \) values of the vector of length \( N \) will be attributed the value corresponding to time \( t = N \). If the \( lag_{i,k} \) between genes \( g_i \) and \( g_k \) is positive, the first \( lag_{i,k} \) values of the gene \( g_k \) vector will be attributed the value corresponding to time \( t = 1 \) and the remaining positions of the vector will correspond to the first \( N-lag_{i,k} \) values of the original expression vector. Figure 1 illustrates this procedure, where the coloured areas correspond to the values that are padded.

A partial correlation matrix for this expression data is computed and kept. All the resulting matrices, one for each gene, are then summed up. The values of this final matrix indicate a degree of association between pairs of genes that integrates the information relative to their lags and arguably improving the simple partial correlation measure.

4. Results

4.1. Generation of in silico data

Two in silico data generators are used: Netsim (Di Camillo B., 2009) and GeneNetWeaver (Schaffter et al., 2011) and five networks are considered for each generator. The simulated time series data consists of 100 time points and 100 genes. The GeneNetWeaver datasets are obtained by reusing the structures of the 5 datasets made available during the DREAM4 challenge. For validation purposes, we also generated data with different settings: 200 time points and 100 genes; 100 time points and 200 genes; 200 time points and 200 genes. Regarding the 200 genes setting, only the Netsim datasets were used, as the DREAM4 challenge is composed of 100 genes datasets.

The goal of the experimental session is to assess the impact of the lag discovery procedure on the accuracy of the considered network inference methods.

The accuracy of the lag discovery version of MRNET, CLR and PC for different maximum lags (set as a percentage of the time series length) is reported in the tables 1, 2 and 3. The accuracy is presented in terms of precision values for the following recall threshold values 0.1, 0.2, 0.3 and 0.4. For CLR, we see that the optimal maximum allowed lag corresponds to 10%
Estimation of temporal lags for the inference of gene regulatory networks

Table 1. Precision as a function of recall and maximum allowed time lags, for the MRNET algorithm, bold means the best score for a given recall.

<table>
<thead>
<tr>
<th>MAX ALLOWED LAG</th>
<th>RECALL</th>
<th>2%</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GNW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>.062</td>
<td>.078</td>
<td>.07</td>
<td>.056</td>
<td>.05</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>.057</td>
<td>.064</td>
<td>.063</td>
<td>.054</td>
<td>.053</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>.053</td>
<td>.056</td>
<td>.054</td>
<td>.045</td>
<td>.044</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>.048</td>
<td>.052</td>
<td>.053</td>
<td>.048</td>
<td>.046</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NETSIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>.075</td>
<td>.075</td>
<td>.073</td>
<td>.069</td>
<td>.068</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>.079</td>
<td>.089</td>
<td>.087</td>
<td>.088</td>
<td>.083</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>.07</td>
<td>.078</td>
<td>.074</td>
<td>.089</td>
<td>.053</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>.064</td>
<td>.066</td>
<td>.068</td>
<td>.052</td>
<td>.049</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Precision as a function of recall and maximum allowed time lags, for the CLR algorithm. The bold notation is used to denote the best score for a given recall and algorithm.

<table>
<thead>
<tr>
<th>MAX ALLOWED LAG</th>
<th>RECALL</th>
<th>2%</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GNW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>.064</td>
<td>.074</td>
<td>.079</td>
<td>.057</td>
<td>.056</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>.051</td>
<td>.067</td>
<td>.065</td>
<td>.052</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>.048</td>
<td>.054</td>
<td>.06</td>
<td>.053</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>.047</td>
<td>.054</td>
<td>.053</td>
<td>.05</td>
<td>.047</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NETSIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>.096</td>
<td>.19</td>
<td>.249</td>
<td>.187</td>
<td>.168</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>.081</td>
<td>.11</td>
<td>.105</td>
<td>.099</td>
<td>.078</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>.072</td>
<td>.082</td>
<td>.08</td>
<td>.062</td>
<td>.06</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>.06</td>
<td>.066</td>
<td>.067</td>
<td>.056</td>
<td>.052</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Precision as a function of recall and maximum allowed time lags, for the PC algorithm. The bold notation is used to denote the best score for a given recall and algorithm.

<table>
<thead>
<tr>
<th>MAX ALLOWED LAG</th>
<th>RECALL</th>
<th>2%</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GNW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>.058</td>
<td>.078</td>
<td>.059</td>
<td>.079</td>
<td>.054</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>.056</td>
<td>.064</td>
<td>.052</td>
<td>.067</td>
<td>.044</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>.052</td>
<td>.056</td>
<td>.049</td>
<td>.06</td>
<td>.042</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>.048</td>
<td>.052</td>
<td>.047</td>
<td>.054</td>
<td>.043</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NETSIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>.06</td>
<td>.105</td>
<td>.076</td>
<td>.249</td>
<td>.051</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>.066</td>
<td>.066</td>
<td>.105</td>
<td>.044</td>
<td>.039</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>.057</td>
<td>.078</td>
<td>.056</td>
<td>.08</td>
<td>.042</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>.066</td>
<td>.055</td>
<td>.067</td>
<td>.043</td>
<td>.045</td>
<td></td>
</tr>
</tbody>
</table>

time points, in both of the dataset types. For MRNET, the maximum allowed values corresponding to 5% and 10% both stand out. When it comes to PC, the maximum allowed lags that return the best results are the ones corresponding to 10% and 20%.

The occurrence that the optimal maximum allowed lags belong in a certain range (5% - 10%) can be explained by the fact that high correlations can occur between variables of small number of samples, and as the maximum allowed lag increases, so does the number of pair of genes that are found to have a high lag. However, as the number of considered points decreases, the correlation loses its discriminative power and a high correlation score will be assigned to otherwise relatively uncorrelated genes.

4.2. Lag discovery analysis and static analysis

The accuracy of the the lag discovery procedure is compared to the static one in table 4 for both the GNW and Netsim datasets. The results shown in this table refer to an optimal allowed lag value set according the results of the previous section (i.e. 5% for MRNET, 10% for CLR, and 20% for PC). Again, the table contains precision values for given recall threshold values. S corresponds to the static analysis and L to the lag discovery analysis, L5 indicates that the value of the maximum allowed lag value is 5% of the time series length. L10 and L20 correspond to 10% and 20%.

It can be seen that there is a clear improvement in the performance of all the algorithms when they are subject to the lag discovery procedure, compared to the static analysis.

We performed the McNemar's test in the two groups of datasets to assess the statistical difference between the predictions using the static and the lag discovery analysis. For each dataset and algorithm the top 500 predictions (which returned the highest algorithm score) were retrieved (approximately 1/10 of all possible interactions), and the number of True Positives and False Negatives for the static and lag discovery analysis were used in the McNemar's test. The returned p-values are in table 6. A high p-value means...
Table 5. Precision for a recall of 0.2, for static analysis and lag discovery analysis, for all algorithms and different experimental settings. The bold notation is used to denote the best score for a given recall and algorithm.

<table>
<thead>
<tr>
<th>N.GENES x N.TIME POINTS</th>
<th>ALG</th>
<th>S</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRNET</td>
<td>.054</td>
<td>.079</td>
</tr>
<tr>
<td></td>
<td>CLR</td>
<td>.049</td>
<td>.093</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>.041</td>
<td>.092</td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100X200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRNET</td>
<td>.051</td>
<td>.052</td>
<td>.031</td>
</tr>
<tr>
<td>CLR</td>
<td>.056</td>
<td>.074</td>
<td>.032</td>
</tr>
<tr>
<td>PC</td>
<td>.041</td>
<td>.045</td>
<td>.019</td>
</tr>
<tr>
<td>200X100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRNET</td>
<td>.048</td>
<td>.048</td>
<td>.031</td>
</tr>
<tr>
<td>CLR</td>
<td>.066</td>
<td>.066</td>
<td>.037</td>
</tr>
<tr>
<td>PC</td>
<td>.024</td>
<td>.024</td>
<td>.016</td>
</tr>
<tr>
<td>200X200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRNET</td>
<td>.038</td>
<td>.038</td>
<td>.023</td>
</tr>
<tr>
<td>CLR</td>
<td>.075</td>
<td>.075</td>
<td>.075</td>
</tr>
<tr>
<td>PC</td>
<td>.02</td>
<td>.02</td>
<td>.02</td>
</tr>
</tbody>
</table>

Table 6. McNemar’s test using the number of True Positives and False Negatives of the static and lag discovery analysis.

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>MRNET</th>
<th>CLR</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNW</td>
<td>0.01</td>
<td></td>
<td>0.085</td>
</tr>
<tr>
<td>NETSIM</td>
<td>2.6e-10</td>
<td>2.6e-10</td>
<td>0.21</td>
</tr>
</tbody>
</table>

6. Acknowledgements

This work was supported by grants from the Communauté Française de Belgique Actions de Recherche Concertées (ARC).

References


Estimation of temporal lags for the inference of gene regulatory networks


Spatial Analysis of Bacterial Infection Patterns in Zebrafish

A.E. Nezhinsky\textsuperscript{1} \ A. Vasylevska\textsuperscript{3} \ A.VanderSar\textsuperscript{2} \ F.J. Verbeek\textsuperscript{1}

\textsuperscript{1}Leiden Institute of Advanced Computer Science, Niels Bohrweg 1, Leiden, The Netherlands
\textsuperscript{2}Dept Medical Microbiology and Infection Control, VUMC, vd Boechorststraat 7, Amsterdam, The Netherlands
\textsuperscript{3}LUMC, Albinusdreef 2, Leiden, The Netherlands

Keywords: image analysis, datamining, pattern recognition, deformable template matching

Abstract

About 1/3 of the world population is infected by tuberculosis. Effective treatment is limited due to bacteria resistance. In order to elucidate which genes are responsible for infection the behavior of the tuberculosis bacteria needs to be analyzed. As a model organism we consider zebrafish larvae that are infected with a variant of the tuberculosis bacteria, i.e. \textit{Mycobacterium marinum}, to study the progression of the infection. We have developed an image analysis technique based on deformable template matching that is able to automatically recognize the zebrafish shape and analyze the infection spread in a high throughput fashion. We have chosen to compare the infection pattern both quantitative as spatial of the \textit{Mycobacterium marinum} and its mutant. From statistical analysis we gained more insight of the infection spread and use the results as a proof of concept for further in depth analysis.

1. Introduction

Tuberculosis is still a widely spread disease annually causing a high number of victims. In order to get more insight on possible treatment it is important to better understand the behavior of the tuberculosis bacteria within an infected organism. The progression of a tuberculosis infection can be characterized by the presence of granulomas. Granulomas are clusters of immune cell and bacteria that indicate an infection location.

In order to model the infection we use the \textit{Mycobacterium marinum} (Mm), a close relative of the \textit{Mycobacterium tuberculosis} Mm can be hosted by zebrafish and is not directly dangerous to humans. Zebrafish is chosen as a model, as its innate immune system is in many ways comparable to human and can be studied by fluorescent imaging. Granuloma presence is visualized by fluorescent agents and in this way a visual inspection of bacteria spread can be obtained. The zebrafish infection model was used in (Stoop et al., 2011) in a high throughput fashion in order to find Mm mutants that were not able to efficiently induce granuloma formation. After screening 1000 randomly created mutants we have identified 30 mutant bacteria unable to infect the larvae.

Previously we (Nezhinsky et al., 2012) have analyzed the infection spread throughout the zebrafish larvae. As a test case, we have chosen mutant 714 [714M], as it is one of the 30 mutants which does not make the fish ill. Therefore we have used this to find different characteristic patterns in granuloma cluster spread and size for both the Mm and 714M.

The analysis is accomplished though imaging. For each zebrafish, a bright field and fluorescence microscopy image is acquired (Figure 1). Until recently, these images were analyzed manually. The analysis included localization of the zebrafish shape in the bright field images and qualitative estimation of the granuloma cluster size and spread from the fluorescent images. The fluorescent signal is often of low intensity and noisy and therefore manual analysis is difficult and will not result in objective evaluation.

Figure 1. A bright field image and a matching fluorescent image. Images were taken with Leica DC500 microscope.
We have designed and implemented a framework for automated shape retrieval and cluster analysis (Nezhinsky and Verbeek, 2010). The recognition algorithm was based on a deformable template matching (Jain et al., 1998; Felzenswalb, 2003) approach that distributes the zebrafish shape into vertical sub regions (slices). Besides our own work this framework has also been applied in large scale applications (Stoop et al., 2011). However, this algorithm used for zebrafish detection had certain drawbacks:

- The performance of the algorithm was slow due to high complexity of template matching (Jain et al., 1998).
- The slices were fixed-width vertical regions that did not discriminate between the top and the bottom of the shape.

Recently, we have developed a new template matching algorithm that distributes the zebrafish shapes into certain characteristic (Figure 4, cf. 2.3.2) but not size fixed regions. This algorithm is more true to nature, due to high biological variation in shape and size of the zebrafish larvae (Figure 2).

![Figure 2. Bright field images under different conditions.](image)

With this new approach more in depth analysis can be realized so that our previous analysis can be extended. The research question is to investigate if there are different infection patterns between the wildtype (Mm) and the 714 mutant.

2. Data & Tools

2.1 Zebrafish Larvae

For this study, we used 189 zebrafish larvae. Of these larvae 5 were not infected, 67 were infected with the MM and 117 were infected with the 714M. (infected with approximately the same amount of bacteria at the age of 6 days). The experiment contained multiple batches of about 30 wells. Each batch contained sibling zebrafish larvae of the three groups in the same imaging settings.

2.2 Microscopy

For this experiment we have made images of zebrafish larvae contained in wells. Up to 3 specimen were present in each well. The larvae were imaged 5 days past infection. Example images are shown in Figure 2. In addition to each bright field image a fluorescent image (Figure 1) at each specific location was taken. The fluorescent image contains the signal at granuloma clusters location.

2.3 Software

The images are the input for an analysis framework that is separated into two parts. First the bright field image is analyzed and the zebrafish shapes are recognized (Nezhinsky et al., 2012). The fluorescent image is segmented and combined with the zebrafish shapes found, subsequently spatial analysis is performed. The result is written to an output file.

2.3.1 BRIGHT FIELD IMAGING; SHAPE LOCALIZATION AND ANNOTATION

For our previous analysis on infection spread (Nezhinsky et al., 2011) we used the deformable template matching algorithm as described in (Nezhinsky and Verbeek, 2010) to localize the zebrafish shape and divide it in 11 parts counting from head to tail. This allowed for annotating the shape as well as doing spatial analysis.

For this study we have abandoned this approach and developed a new template matching algorithm that localizes the zebrafish shapes without the computationally hard sub-region search approach.

![Figure 3. Output for a single image as created by the ZIF software. The shapes are denoted with a red line. The red circle denotes the found head area, the green circles denote the body area and the blue circles the tail area. The purple regions indicate the location of granuloma formation. This image is created in an automated fashion with a processing time of 1s.](image)
and Verbeek, 2012 submitted, Nezhinsky and Verbeek, 2010) can be meaningfully retrieved. Due to the rotation independence of the algorithm the input images of the zebrafish do not need to be aligned in a certain way. In Figure 3 we show the output of the algorithm.

2.3.2 FLUORESCENT IMAGING; ANALYSIS
A mask for each zebrafish is obtained from the shape retrieval in the previous step. No granuloma formation is present at NI, therefore this group was only used to obtain a color value that represents the noise level threshold (n). All signal below n is considered noise, while signal above n represents granuloma presence. Depending on its location in the zebrafish shape the granulomas are assigned to the regions, as shown in Table 1 and in the graphical representation in Figure 4.

<table>
<thead>
<tr>
<th>AREA NAME</th>
<th>NOTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEAD</td>
<td>H</td>
</tr>
<tr>
<td>BODY TOP</td>
<td>BT</td>
</tr>
<tr>
<td>BODY BOTTOM</td>
<td>BB</td>
</tr>
<tr>
<td>TAIL TOP</td>
<td>TT</td>
</tr>
<tr>
<td>TAIL BOTTOM</td>
<td>TB</td>
</tr>
</tbody>
</table>

Table 1. Automatic region assignment.

Figure 4. Graphical representation of the separation of a zebrafish into regions.

This division is estimated from the midline that results from the segmentation procedure (cf. Figure 3). The estimated regions correspond to those proposed by (Volman et al., 2004; Gollavelli and Ling, 2011).

2.3.3 OUTPUT AND DATASET GENERATION
The result of the image analysis are twofold. For usual feedback the segmentation results are overlaid on the specimen (cf. Figure 3). The numerical results are written to a csv file that is used in the further analysis. The granuloma spread for each region is described by the amount of Clusters in a region (CC[region]) and the total amount of Infection I[region] in each region. Hence, the Average Cluster Size per region is calculated by ACS[region]=I[region]/CC[region].

3. Analysis and Results
In our study we set out to analyze the relationship between mutant and wild type in the amount of clusters, spatial distribution and the cluster size. Please note, that it is impossible to directly compare the amount of infection between different regions, since the regions are of different size. However it is possible to compare each region between the MM and the 714 mutant.

3.1 Amount of infection
First we compare the average amount of granulomas in each region between MM and 714 (cf. Figure 5).

Figure 5. Spatial comparison of the mean amount of infection for MM and 714M in relation to the zebrafish regions.

At this point this distribution is not conclusive. This is due to the fact that the MM makes the fish more ill and thus overall produces more granulomas then the 714M. The mean amount of infection throughout the entire fish is 4558.24 ( \sigma M = 569.51) for MM and 825.69 ( \sigma M = 102.74) for the 714M. Roughly the amount of infection in MM is thus 5 times higher than in the 714M.

In order to analyze the different batches in the same way we normalize the I[region] over the total I. The normalization is done for each individual case and subsequently the mean is calculated. In Figure 6 the results are depicted.

Figure 6. Spatial comparison of the normalized mean amount of infection for MM and 714M in relation to the zebrafish regions with a 95% confidence interval.
From the graph we can observe that the behavior of MM and 714M is approximately the same. The mean and the 95% confidence interval suggest that the two distributions in H, BT and TT can be considered similar. In BB (body bottom) and TB (tail bottom) the distribution behaves differently. Let us consider this in more detail.

Our null hypothesis states that, under assumption that the two groups are independent, their variances are equal. We therefore, apply the Levene’s Test for Equality of Variances to the $|i|$, $i$ in range [H,BT,BB,TT,TT]. The results are shown in Table 2.

Table 2: Levene’s test per region for equality of variances for normalized $|i|$, in this table equal variance is assumed.

<table>
<thead>
<tr>
<th>Region</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>0.043*</td>
</tr>
<tr>
<td>Body Top</td>
<td>0.758</td>
</tr>
<tr>
<td>Body Bottom</td>
<td>0.029*</td>
</tr>
<tr>
<td>Tail Top</td>
<td>0.804</td>
</tr>
<tr>
<td>Tail Bottom</td>
<td>0.004*</td>
</tr>
</tbody>
</table>

For zebrafish regions H, BB and TB the significance is < 0.05 and thus the hypothesis is accepted, the corresponding variances are equal (marked with * in Table 2).

For other regions the variances significantly differ. Finally, we performed the independent samples t-test. Based on the results from Levene’s test we know which variances significantly differ; in Table 3 only the correct assumptions are listed.

Table 3: t-test for Equality of Means.

<table>
<thead>
<tr>
<th>REGION</th>
<th>ASSUMPTION</th>
<th>SIG. 2T</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>Equal var.</td>
<td>0.617</td>
</tr>
<tr>
<td>BB</td>
<td>Equal var.</td>
<td>0.015**</td>
</tr>
<tr>
<td>TB</td>
<td>Equal var.</td>
<td>0.028**</td>
</tr>
</tbody>
</table>

We observe that there is a significant difference (this means significance < 0.05) in the mean value for regions BB and TB (marked with ** in Table 3).

3.2 Amount of granuloma clusters

We compare the average number of clusters (CC variable) between MM and 714 (cf. Figure 7).

Figure 7: Spatial comparison of the mean amount of clusters for MM and 714M in relation to the zebrafish regions.

Throughout the entire fish MM gives more clusters than 714M. MM has a mean area of 161.94 ($\sigma(M) = 18.60$) throughout the entire fish and the 714M has an area of 46.40 ($\sigma(M) = 4.13$). The amount of clusters in MM is about 3 times higher than in the 714M. This makes it difficult to compare the relative behavior of clustering. Therefore let us consider the normalized cluster distribution pattern in Figure 8.

Figure 8: Spatial comparison of the normalized mean amount of clusters for MM and 714M in relation to the zebrafish regions.

From the graph we observe that the behaviors of MM and 714M are approximately the same. We analyze the graph in more detail by testing variance and mean equality of each region, in the same way as described in 3.1.

Table 4: Levene’s test for equality of variances for normalized CC[i], in this table equal variance is assumed.

<table>
<thead>
<tr>
<th>REGION</th>
<th>SIG.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEAD</td>
<td>0.931</td>
</tr>
<tr>
<td>BODY TOP</td>
<td>0.924</td>
</tr>
<tr>
<td>BODY BOTTOM</td>
<td>0.015*</td>
</tr>
<tr>
<td>TAIL TOP</td>
<td>0.162</td>
</tr>
<tr>
<td>TAIL BOTTOM</td>
<td>0.004*</td>
</tr>
</tbody>
</table>
Equal variance is found for BB and TB.

**Table 5. t-test for Equality of Means.**

<table>
<thead>
<tr>
<th>REGION</th>
<th>ASSUMPTION</th>
<th>SIG.2T</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>Equal var.</td>
<td>0.406</td>
</tr>
<tr>
<td>TB</td>
<td>Equal var.</td>
<td>0.072</td>
</tr>
</tbody>
</table>

From the results of Table 4, 5 observe that there is no significant difference in the mean value for the regions with equal variances.

### 3.3 Average size of granuloma clusters

We have looked also at the average size of the granuloma clusters (ACS). Figure 9 shows the average cluster sizes for different regions of the zebrafish.

![Figure 9. Spatial comparison of the average cluster sizes for MM and 714M in relation to the zebrafish regions.](image)

In order to find significant differences in the behavior we repeat the analysis as done in 3.1 and 3.2. the results are given in Table 6 and 7.

**Table 6. Levene’s test for equality of variances for the average cluster size. In this table equal valiance is assumed.**

<table>
<thead>
<tr>
<th>REGION</th>
<th>SIG.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEAD</td>
<td>0.245</td>
</tr>
<tr>
<td>BODY TOP</td>
<td>0.202</td>
</tr>
<tr>
<td>BODY BOTTOM</td>
<td>0.005*</td>
</tr>
<tr>
<td>TAIL TOP</td>
<td>0.311</td>
</tr>
<tr>
<td>TAIL BOTTOM</td>
<td>0.004*</td>
</tr>
</tbody>
</table>

**Table 7. t-test for Equality of Means.**

<table>
<thead>
<tr>
<th>REGION</th>
<th>ASSUMPTION</th>
<th>SIG.2T</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>Equal var.</td>
<td>0.000**</td>
</tr>
<tr>
<td>TB</td>
<td>Equal var.</td>
<td>0.003**</td>
</tr>
</tbody>
</table>

From the results of Table 5 and 6 we observe that there is a significant difference in the mean value for BB and TB regions.

### 3.4 Results

In all regions of the zebrafish the MM gives a higher area of infection then the 714 (overall the infection area is 5 times larger).

The same percentage of the granulomas in both MM and 714 are located in the head region.

A higher percentage of the granulomas of the MM then 714 are located in the body bottom region.

A higher percentage of the granulomas of the 714 then MM are located in the tail bottom region.

A graphical representation of these results is shown in Figure 10.

![Figure 10. Graphical representation of the amount of infection. While there is always more Mm infection then 714M, the percentages of their presence in certain regions are different.](image)

### 3.4.1 Amount of Granuloma Clusters

In all regions of the zebrafish the MM gives a higher amount of clusters then the 714 (overall the amount of clusters is 3 times higher).

Cluster spread pattern is not significantly different for MM and 714M.

In most regions the cluster sizes are not significantly different (though MM tends to have larger clusters), except the body bottom and the tail bottom area, where the MM gives significantly bigger clusters than 714M. A graphical representation of these results is shown in Figure 11.

![Figure 11. Graphical representation of the average cluster size. Spot location within each region does not provide locational information.](image)
4. Conclusion and Future work

In this paper we have used the framework we developed for automatic zebrafish larvae recognition in depth analysis of granuloma cluster spread. From statistical analysis of the output data we gained some insights on the spread pattern of the MM as opposed to the 714M.

The results indicate that this technique is suitable for the analysis of mutants and the phenotype of the mutant can be characterized by the infection pattern.

In the future this will therefore be used for a larger set of mutants and larger datasets. We want to improve the analysis by taking the region size into account, more precise region definition and a larger set of features i.e. texture.

Acknowledgements

This research presented in this paper was partially supported by the Smartmix Program.

References


The Combinatorial Naming Game

Kevin Stadler
Pieter Wellens
Joachim De Beule

Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Keywords: learning in multi-agent systems, lexicon formation, naming game, source coding, language change

Abstract

In this article we introduce combinatorial form into the well-known Naming Game paradigm in which autonomous agents have to establish a globally shared communication system through strictly local interactions. While virtually all investigations of the Naming Game so far have been carried out using atomic names, we highlight some interesting aspects that arise only when naming is done using combinatorial forms. We present an analysis which relates Naming Games to information theory and discuss first results from multi-agent simulations in light of this analysis.

1. Introduction

The Naming Game paradigm (Steels, 1995; Baronchelli et al., 2006; Wellens, 2012) has been used extensively to investigate the emergence of shared communication systems. The goal of a Naming Game is for a set of agents to agree on a name for one or more objects without any central control. The agents engage in local interactions where one of the agents acts as a speaker who tries to draw the attention of another agent (the listener) to a specific object in the communicative context by uttering a name. The game fails if the listener fails to identify the object or does not agree with the name, but both agents can use their experience of the game outcome to improve their internal lexicon so as to communicate more successfully in the next game.

So far, Naming Games have only been investigated with atomic names that do not exhibit any internal structure. Human language on the other hand exhibits the interesting property of “duality of patterning” (Hockett, 1960): the smallest meaningful units in a given language (morphemes, such as the names for objects in the Naming Game) are again combinations of a basic inventory of meaningless units (typically called phonemes). This property is peculiar because compositional communication systems as expressive as human language do not in principle require this combinatorial nature of morphemes. There do however exist reasons for combinatoriality: the free combination of a basic inventory of meaningless sounds allows the creation of an unlimited inventory of basic words which is not constrained by the limits of perceptual distinctiveness (Zuidema & de Boer, 2009).

While a wealth of multi-agent simulations in the wake of the Naming Game have yielded interesting results regarding both lexicon as well as ontology formation (Steels, 2011), there are still many open questions regarding the role that structured form spaces play in the emergence of communication systems, where a form space is said to be structured if some forms are more easily confused with one another than others (de Boer & Verhoef, 2012). While the Naming Game has already been investigated using structured form spaces (Steels & Kaplan, 1998; Lipowski & Lipowska, 2009), these investigations have only dealt with atomic names but not with combinatorial form, where the confusability stems from the fact that the names are themselves made up of smaller units which are re-used across different names.

The most interesting aspect of such a sub-semiotic treatment of names is that part of the agents’ communication strategies do not directly impact on the level of meaningful names, but only on the basic sound inventory. Of particular interest is the well known “principle of least effort” (Zipf, 1949) which states that speakers minimise the effort involved in utter-
ing their linguistic expressions. This principle is not limited to the selection of names in the case of synonyms, it also implies that articulatory laziness leads to a constant erosion (shortening) of phonetic forms. When language is viewed as being static, such variation might appear dangerous in terms of threatening a “successful” communication system. But a recent trend in linguistics is to characterise language as a complex adaptive system (Steels, 2000) that adapts to the cognitive and communicative requirements of the language users. Much like in biological evolution, it is variation which lies at the core of languages’ capability to change and adapt to different scenarios.

In this article we extend the Naming Game paradigm by investigating its dynamics in a combinatorial form space. The model will be formally introduced in Section 2, followed by an information-theoretic analysis and the results of numerical simulations in Section 3. Section 4 points out directions for future work.

2. The Combinatorial Naming Game

A basic Naming Game requires a set of objects and a population of agents who communicate about these objects in local interactions. For every interaction a context (i.e. a subset of all the objects that can be communicated about) is selected and two agents are chosen from the population. One agent is assigned the role of speaker, the other of listener. Both agents are confronted with the same context of objects. The interaction then proceeds according to the following interaction script:

1. The speaker mentally picks one object from the context (called the topic) and utters a name for it.
2. The listener interprets the name and points to the object he believes the speaker intended.
3. The speaker either agrees or disagrees with the listener.
4. Both agents have the opportunity to adapt their inventory of names.

The extensions of the Combinatorial Naming Game to the original Naming Game are twofold: firstly, the atomic names are replaced with names which are sequences of meaningless characters drawn from a fixed alphabet. Secondly, in order for this change to have more impact on the conventionalisation dynamics, this combinatorial language model is enhanced by a model of phonetic erosion (see Section 2.2).

Based on the interaction script of the Naming Game, the dynamics of the simplest version of a Combinatorial Naming Game are dependent on the following parameters:

- $n \geq 2$: the number of agents
- $o \geq 1$: the total number of objects
- $c \leq o$: the number of objects which are presented at the same time in individual contexts
- $p \geq 2$: the size of the phoneme inventory, i.e. the number of distinct phonemes that can be combined to form names
- $l \geq 1$: the initial length of newly invented names
- $S$: the agents’ strategy for storing names and selecting which names to produce (see Section 2.1)
- $U$: the utterance production model which determines how a word might erode (see Section 2.2)
- $0 \leq e \leq 1$: the probability of phonetic erosion occurring while uttering a name

Before we go into the details of the different strategies, it is insightful to discuss the kinds of conventionalisation problems the agents face in this game. At the core of all Naming Game scenarios lies synonymy or word form competition, which means that there are multiple competing names for the same object in a population instead of one shared convention. We also find this basic property in the Combinatorial Naming Game, but here not alone due to distributed invention of multiple names by different agents, but also due to phonetic erosion which introduces competitors in the form of shorter variants of the same name.

A second, even more interesting effect of the combinatorial form space is that it also introduces homonymy, i.e. it is possible that the same form is used to refer to multiple different objects. This problem does not occur in the traditional Naming Game, and it is indeed a property normally associated with language games involving referential uncertainty of the intended meaning, such as in the Guessing Game (De Beule et al., 2006). Next to establishing one shared name for every object, the agents’ strategies thus have to solve the additional problem of making sure that no two objects share the same name.

2.1. Name Selection Strategies

The strategies according to which the agents select the names they use are the most interesting aspect of language games, because they determine if and how well the communicative tasks at hand are solved. Due to the homonymy introduced by using structured names, it is not possible to employ the simple name lookup
like in traditional Naming Games. A name for an object might also be used to refer to some other object present in the current communicative context, which would result in referential ambiguity of the name.

To counteract this, the strategies used in the Combinatorial Naming Game make use of a feature called re-entrance (Steels, 2003). Re-entrance is analogous to the symmetry principle found in human language: speakers only use utterances for a communicative purpose if they think that these utterances would have the same effect when used by someone else on themselves. Before uttering any name, an agent thus first attempts to interpret the name in the given context to see if it does indeed unambiguously pick out the desired topic.

Based on this general condition, the agent which is selected as the speaker proceeds by selecting an unambiguous name from its lexicon according to one of the following strategies:

**winner-take-all strategy:** of all the names the agent knows for the intended topic, it deterministically selects the one with the highest frequency, i.e. the one that the agent has most often seen being used for the topic by other speakers in previous interactions. The same bias is used in interpretation, so if there are multiple possible interpretations of the name in the context, they deterministically pick the object which they have most often heard being referred to with that name.

**economical strategy:** the agent selects the shortest name known to them which unambiguously refers to the topic in the given context. This strategy explicitly follows the “principle of least effort” on the side of the speaker. Interpretation is in this case stricter: the listener only points to an object if it is the only possible interpretation of the name in the context.

**minimal strategy:** this strategy is based on the eponymous minimal strategy known from the original Naming Game (Baronchelli et al., 2006). As speakers, the agents sample a random name from all the names they know for the chosen topic. This set of synonyms is constantly expanded by new forms that are encountered, but when a listener hears the same name for an object for the second time it discards all other synonyms, thereby settling on a single name for the object.

If the agent does not yet know a name for the object or none of the names would unambiguously refer to the intended topic in the given context, the agent invents a new name, which is a random, not necessarily unique sequence of $l$ characters from the phonetic inventory.

### 2.2. Name Production

To get variation on top of the invented names, we introduce a very simple speech production model which might shorten the utterance intended by the agent. If the agent wants to produce a certain name of length $i > 1$ which is the sequence of characters $[c_1 \ldots c_i]$, then with a certain probability $e$ the utterance will erode at the end, i.e. the name will instead be produced as $[c_1 \ldots c_{i-1}]$.

The utterance erosion model introduces stochastic variation into the population, and the goal of our model is exactly to investigate how different strategies can handle the homonymy it introduces as well as exploit this variation to arrive at more economical communication systems.

### 2.3. Naming as Source Coding

Apart from the conventionalisation dynamics, the combinatorial form space adds a new characteristic to the naming game, namely that of source coding. Source coding is a mapping from symbols from a finite source alphabet $\Sigma_1$ (in our case the atomic objects) to a sequence of symbols from a finite target alphabet $\Sigma_2$. The Combinatorial Naming Game introduced here can be interpreted as the negotiation of a source code among distributed agents, with alphabet sizes $|\Sigma_1| = o$ and $|\Sigma_2| = p$.

Most importantly, Shannon’s source coding theorem (Shannon, 1948) provides us with bounds on the average name length that an optimal communication system can use while maintaining the capability to reconstruct the exact original message (lossless source coding). Assuming that all $o$ objects occur with equal probability $\frac{1}{o}$, this means that the entropy of each name is $\log_2(o)$ bit. Since in all Naming Game experiments to date the successive symbols (or contexts) are randomly drawn in an independent fashion, the overall entropy $H$ of the communicative contexts as an information source is also $\log_2(o)$ bit.

The shortest possible representation to encode the messages (the objects) in a given alphabet is their entropy divided by the logarithm of the number of symbols in the target alphabet, i.e. $\frac{\log_2(o)}{\log_2(p)}$. We will use these values as a baseline against which the results of our simulations in the following section can be related.

### 3. Experiments

We implemented the Combinatorial Naming Game laid out in the previous section as a multi-agent system to study its dynamics through numerical simulation.
Due to space limitations, the discussion in this paper is necessarily limited to a subset of the huge parameter space. Particularly, because we are actually more interested in the languages’ characteristics rather than the spreading of conventions across the population of agents, we set the size of the population to its minimal possible value of $n = 2$ agents, as has previously been done in other investigations of structured form spaces (Lipowski & Lipowska, 2009).

To facilitate the information-theoretic analysis of the resulting languages, we set the size of the phoneme inventory to $p = 2$. With three different name selection strategies, erosion probabilities and initial name lengths, the remaining parameter space is still large. To further simplify later analysis, we stick to powers of two for the number of objects ($o = \{4, 8, 16, 32, 64\}$) while always using an identical context size of $c = o$.

A crucial choice for the model is the initial length $l$ of new words that the agents can invent. Given the fact that names can only get shorter through erosion, it is also an boundary for the maximum possible word length the agents can use. If we interpret the set of objects as a source alphabet with $o$ characters, the Shannon information of every single object will be $\log_2(o)$.

A minimally optimal language for unambiguously naming $o$ objects uses all different names of length $\log_2(o)$ characters, which therefore provides an absolute lower bound for the length of initial words which might feasibly work together to form a fully successful language (i.e. a losslessly reconstructable source code). Because we are interested in seeing how agents might converge towards such an optimal language themselves, we combine every combination of the other parameters with initial word length settings of $l = \{\log_2(o), \log_2(o) + 1, \log_2(o) + 2, \log_2(o) + 3\}$.

### 3.1. Evaluation Measures

In order to evaluate and understand the game dynamics the following population-level measures are used:

- **Communicative success** captures how successful agents are at solving the communicative task of identifying the topic intended by the speaker. While individual interactions result in either success or failure, averaging over many trials with the same parameter settings can be used to analyse the overall development of communicative success as a percentage of successful interactions.

- **Alignment success** is based on communicative success, but with the extra condition that the name used by the speaker is the same name that the listener would have used to name the object if she or he would have been chosen as the speaker. This measure is thus more indicative of whether the agents actually converge on a shared set of conventions, rather than simply remembering every form they encountered in their input.

**Average name length** is the number of phonetic characters used by the agents to name objects in the communicative contexts. While this integer measure will again vary a lot between individual interactions, averaging over many trials can be used to determine overall trends.

### 3.2. Results

We ran 200 simulation trials in 180 different conditions, for 4000 communicative interactions each. All plots in this section display the average behaviour of the simulation across these 200 trials.

Of prime interest is of course how the different strategies fare in terms of establishing a shared communication system. A comparison of the three strategies using a representative parameter setting can be seen in Figure 1. The winner-takes-all strategy (left) is the only one to reach full alignment success, disrupted only by occasional cases of ambiguity caused by random erosions. The frequency-biased production and interpretation leads to a strict 1:1 mapping bias between objects and forms which completely ignores variation of the form, and the agents converge on the set of (full-length) names which were invented first. While this strategy is thus most successful at establishing a working communication system, it exhibits no dynamics whatsoever in terms of form.

The economical strategy (Figure 1 middle) behaves quite differently. It implements an explicit bias towards shorter forms, and convergence towards the shortest possible expected name length for a losslessly reconstructable source code occurs rapidly. While the resulting code is more economical, the unconditional acceptance of shorter conventions comes at the expense of a robust communication system, as is reflected in significantly lower communicative success rates.

The minimal strategy (Figure 1 right) exhibits less extreme behaviour than the other two. Rather than being geared towards some absolute external property (absolute frequency or economy), the name selection behaviour of this strategy is based on the agent’s local interaction history. The language also converges towards the minimal possible name length, but with higher communicative success than the economical strategy. Neither of the two latter strategies achieve the same level of communicative success as the winner-
take-all strategy, but this is a consequence of the resulting set of conventions: because the most economical codes are not redundant in any way, any occurrence of erosion is almost guaranteed to result in ambiguity and thus in communicative failure.

Let us now turn to the role of the initial word length \( l \). Having a small inventory of possible initial names means that a lot of the randomly invented names for different objects will be homonymous from the start, resulting in ambiguity which cannot be resolved because many of the few other possible names will already be taken by yet other objects. This problem gets worse with increasingly small initial name lengths, as can be seen in the left-most panels of Figures 2 and 3. Because the form space is too small, a completely successful communication system is simply never able to get off the ground. Different strategies do however exhibit different sensitivity to the initial word lengths, confirming previously identified characteristics: the winner-take-all strategy (not shown), while inevitably suffering from a small initial inventory, still manages to establish a working set of conventions with limited communicative success. The economic strategy on the other hand struggles tremendously with small name lengths, as can be seen in Figure 2. The minimal strategy (Figure 3), while also exhibiting convergence on an economic code, does seem to do so in a more adept way, as it is able to recover as quickly as the winner-take-all strategy.

4. Conclusions & Future Work

In this article we have presented a model which extends the Naming Game paradigm to structured, particularly combinatorial, form spaces. The numerical simulations laid out here are of course only a first step, and we are already investigating how the strategies scale with larger population sizes, as well as establishing convergence proofs for the different strategies.

Our model has used the simplest possible phonetic space with only one distinction, a next step will be to introduce more structure by increasing the size of the phoneme inventory and allowing mutation of characters by modelling truly continuous phonetic spaces where the distinction of phonemes is based on their contrastive nature. This most simple simulation presented here is thus only a first endeavour into the Combinatorial Naming Game, a general model for the systematic investigation of structured form spaces.

Acknowledgments

This research was funded by a PhD grant of the Agency for Innovation by Science and Technology (IWT).

References


The Combinatorial Naming Game

**Figure 1.** The impact of the three different name selection strategies on the dynamics of the Combinatorial Naming Game with 8 objects, a context size of 8, two agents using an initial word length of $l = 5$ and an erosion probability $e = 0.05$. The graph shows the average values over 200 separate trial runs. With the economical as well as minimal strategy, the average utterance length converges towards the shortest possible expected name length for a losslessly reconstructible source code, indicated by the dotted line.

**Figure 2.** The impact of the initial word length $l$ on the dynamics of the Combinatorial Naming Game with 8 objects, a context size of 8, two agents using the economical strategy and an erosion probability of $e = 0.05$.

**Figure 3.** The impact of the initial word length $l$ on the dynamics of the Combinatorial Naming Game with 8 objects, a context size of 8, two agents using the minimal strategy and an erosion probability of $e = 0.05$. 
On Estimating Model Accuracy with Repeated Cross-Validation.

Gitte Vanwinckelen
Hendrik Blockeel
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, Box 2402, B-3001 Heverlee, Belgium

Keywords: repeated cross-validation, predictive model evaluation, conditional prediction error

Abstract
Evaluation of predictive models is a ubiquitous task in machine learning and data mining. Cross-validation is often used as a means for evaluating models. There appears to be some confusion among researchers, however, about best practices for cross-validation, and about the interpretation of cross-validation results. In particular, repeated cross-validation is often advocated, and so is the reporting of standard deviations, confidence intervals, or an indication of "significance". In this paper, we argue that, under many practical circumstances, when the goal of the experiments is to see how well the model returned by a learner will perform in practice in a particular domain, repeated cross-validation is not useful, and the reporting of confidence intervals or significance is misleading. Our arguments are supported by experimental results.

1. Introduction
Evaluation of predictive models is a ubiquitous task in machine learning and data mining. The task is not as trivial as it may seem. It is generally known that, to get an unbiased estimate of the accuracy of a model learned via machine learning, one should test the model on unseen data, not on the training set. In some cases, the population accuracy can be estimated from the training error using mathematical formulas. For instance, in linear regression, no separate test set is needed to estimate the error of the model: This error can be estimated accurately from the training data, using the concept of “degrees of freedom” to transform a training error into an unbiased estimate for the population error. For many advanced data analysis methods, however, one can not mathematically derive an unbiased estimate of population accuracy from training set accuracy, and more empirical methods are needed.

A basic approach is to use hold-out sampling which splits the available data set into a training set to learn a model, and a test set to estimate the accuracy of this model on. This requires that the training and test sets are disjoint, and that the training set is no more representative for this test set than for the population as a whole.

Often, when a limited set of data is available, one wants to learn a model from the whole data set, in order to maximally exploit the available information. Unfortunately, that leaves no unseen data to evaluate the accuracy of the model. In this case, an often used procedure is to learn a model \( \hat{f} \) from the whole data set, and estimate the population accuracy of \( \hat{f} \) by using a resampling technique, such as cross-validation. Like any estimator, cross-validation has some bias and variance. The non-zero bias has been pointed out by, for example, Hastie et al. (2011). Because its variance is known to be relatively high, it is often advocated to repeat the cross-validation a number of times and average out the results, or to add confidence intervals that indicate how accurate the estimates are.

There are obvious statistical problems with estimates based on repeated subsampling of one data set, and for this reason, one may doubt whether repeated cross-validation is all that useful. In this paper, we investigate this question. We start with clearly defining some concepts and terminology, showing that several types of experimental questions need to be distinguished, and results of cross-validation need to be interpreted carefully. Next, we show experimentally that for the questions that are most important in practice, it is not useful to conduct repeated cross-validation.
2. Cross-validation based estimates

Consider the following problem. We have a data set $S$ from some domain $D$. $S$ is typically assumed to be a random sample drawn from a population $P$. The data consists of a set of predictor variables $X$ which are in relation to a target variable $Y$ as $Y = f(X)$. We also have a learner $L$, which, given a data set $S$, returns a model $\hat{f}(X)$. The loss function $l(Y, \hat{f}(X))$ measures how well $\hat{f}$ approximates $f$ and is a measure of the accuracy of $L$. We use the one-zero loss function which equals 0 if for a given $x$, $\hat{f}(x)$ equals the real value $f(x)$, and which equals 1 otherwise. We can now consider several questions about $L$ or $\hat{f}(X)$. Focusing on accuracy as the most important performance measure, we may be interested in estimating the following population parameters:

- $\alpha_1 = E[l(Y, \hat{f}(X))]:$ the mean accuracy of $\hat{f}(X)$ on $P$, taken over all data sets $S'$ of the same size as $S$
- $\alpha_2 = E[l(Y, \hat{f}(X)|S]):$ the accuracy of $\hat{f}(X)$ on $P$ for a fixed sample $S$

$\alpha_1$ is computed by computing the mean accuracy over all models $\hat{f}(X)$ that can be learned from data sets $S'$ of the same size as $S$. For $\alpha_2$ on the other hand, $\hat{f}(X)$ is a fixed model determined by the chosen $S$. $\alpha_1$ is known as the unconditional prediction error, and indicates to some extent how well learner $L$ is suited for this problem domain. $\alpha_2$ is known as the conditional prediction error, and it indicates how well the specific model obtained by running $L$ on the available data can be expected to perform. When it is the intention to deploy the model learned from $S$ in practice, $\alpha_2$ is the most relevant parameter. Therefore, we focus on estimating $\alpha_2$.

As said, cross-validation is often used to estimate the performance of learners or models. $k$-fold cross-validation works as follows. The available data set $S$ is divided into $k$ equally sized subsections $S_i$, also called folds. For each fold, a training set $T_i$ is defined as $S\setminus S_i$, from which a model $M_i$ is learned. Next, the accuracy of this model is computed on $S_i$, and finally the mean of all these accuracies is returned as an estimate $\hat{A}$.

$\hat{A}$ is usually interpreted as an estimate of the predictive accuracy of the model $f(X)$ learned from the whole data set $S$. This estimate is pessimistially biased, because it really estimates the average accuracy of models learned from a subset of $(k-1)/k \cdot 100\%$ of the data, which is likely to be slightly less good than the accuracy of the more informed model that is learned from the whole data set. This type of bias can be minimized by performing leave-one-out cross-validation, which sets $k$ to the number of instances in the data set.

In addition to bias, the results of a $k$-fold cross-validation also have high variance. If we run two different tenfold cross-validations for the same learner on the same data set $S$, but with a different random partitioning of $S$ into subsets $S_i$, these two cross-validations can give quite different results. An estimate with smaller variance can be obtained by repeating the cross-validation several times, with different partitionings, and taking the average of the results obtained during each cross-validation.

Repeated cross-validation is often advocated, using as an argument the high variance of the result of a single cross-validation. However, while this procedure indeed reduces the variance of the estimates, it does not remove the bias. We now try to make this more precise.

We introduce the following notation:

- $\hat{A}$: the result returned by a single $k$-fold cross-validation
- $\hat{C}_k$: the population of all possible $k$-fold cross-validations over this particular data set $S$
- $\mu_k$: the mean of $\hat{A}$ taken over all possible $k$-fold cross-validations over $S$ (i.e., taken over $\hat{C}_k$)
- $\alpha_3$: the mean accuracy of $L(S')$ on $P$, taken over all $S'$ of size $(k-1)/k|S|

Repeated cross-validation boils down to repeatedly drawing an element from $\hat{C}_k$, say $n$ times, and computing the average of all these results. It is clear that this average, $\hat{A} = \sum_{i=1}^{n} \hat{A}_i/n$, approximates $\mu_k$ as $n$ goes to infinity: $E(\hat{A}) = \mu_k$ and $Var(\hat{A}) = \sigma_k^2/n$ with $\sigma_k^2$ the variance of $\hat{A}$ taken over $\hat{C}_k$.

Consequently, repeated cross-validation allows us to accurately estimate $\mu_k$, the mean of all possible $k$-fold cross-validations over the given data set $S$. However, the parameter we are really interested in is $\alpha_2$. It is unclear whether $\mu_k$ is a good estimator for $\alpha_2$ and if it is not, whether this is because of bias or large variance.

One could argue that the estimator is biased due to the fact that $\hat{A}$ reflects the accuracy of models learned from only a proportion $(k-1)/k$ of the data. It estimates the accuracy of models learned using slightly less data than available in $S$. But this is perhaps only part of the truth: $\alpha_3$ is the mean accuracy of models learned from equally few data, and it would be interesting to investigate whether, for a particular $S$, $\mu_k$ is
equal to \( \alpha_3 \) (though there is no prior reason to believe it is higher or lower).

Thus, given a data set \( S \), \( \hat{A} \) asymptotically approximates \( \mu_k \), but not necessarily \( \alpha_1 \), \( \alpha_2 \) or \( \alpha_3 \). It is uncertain whether it approximates any of the parameters we may be interested in, even the \( \alpha_3 \) parameter that explicitly takes into account the differences in size of the training sets for the individual models.

Confidence intervals around \( \bar{\alpha}_2 \) are sometimes constructed. When using the standard formula for confidence intervals, they are constructed so that they contain \( \mu_k \) with a certain confidence, not any other parameter. It would be erroneous to interpret confidence intervals, based on repeated cross-validation, as "almost certainly containing \( \alpha_2 \)" (or any of the other parameters one might be interested in).

The conclusion is that statistical inference can easily be done for \( \mu_k \), but \( \mu_k \) itself is of little interest; on the other hand, for the parameters we are interested in, the \( \alpha_1 \), there is no guarantee that they will be estimated with higher precision as the number of cross-validation repetitions increases.

3. Related work

Several authors have discussed the experimental evaluation of learners, comparative or otherwise. We focus on those contributions that are most relevant for this work.

The fact that cross-validation based estimators have high variance and non-zero bias has been pointed out several times. Kohavi (1995) considers the goal of selecting the best learner among a set of possible learners, and, in this context, experimentally compares bootstrapping and cross-validation for varying numbers of folds. He studies the bias and variance of these methods, shows that \( k \)-fold cross-validation has smaller bias but higher variance as \( k \) increases, and concludes that, from the point of view of selecting the most suitable learner, stratified tenfold cross-validation is overall the best method, even when it is computationally possible to use more folds. He suggests that repeated stratified tenfold cross-validation may work even better, as it is likely to reduce variance, but he does not experiment with this.

Braga-Neto and Dougherty (2004) investigate cross-validation for estimating \( \alpha_2 \) in the context of small-sample microarray classification. They provide a formal definition of the bias and variance of cross-validation estimates by looking at the deviation distribution of \( \alpha_2 - \bar{A} \) for a certain data distribution. A \( k \)-fold cross-validation estimator is unbiased if \( \mu_k, = E[\alpha_2 - \bar{A}] = 0 \). A large spread of the variance \( \text{Var}[\alpha_2 - \bar{A}] \) of the deviation distribution indicates a large variance of the estimator. Bias and variance are combined in the root-mean-square error \( \sqrt{E[\alpha_2 - \bar{A}]^2} \) of the distribution. The focus of the paper lies on an investigation of the variance of cross-validation. The conclusion is that cross-validation estimators typically have high variance for small samples, which makes their use problematic for analysis on small microarray samples.

Hastie et al. (2011) also discuss the bias and variance of cross-validation, and the fallacies when using it for estimating model accuracy. They draw attention to the fact that dependencies are often unknowingly introduced between the training and the test set by first using test points to design the learner, and performing cross-validation afterwards. This leads to an overly optimistic accuracy estimate. The authors also empirically investigate that cross-validation typically results in a good estimate for \( \alpha_1 \), but it does not for \( \alpha_2 \).

Schaffer (1993) specifically views cross-validation as a meta-learning technique that allows us to choose which among a given set of learners is likely to give the best predictive model. He concludes that cross-validation selects the best learner in most cases. However, similar to any other learning technique, performance depends on the setting the learner is used in.

Dietterich (1998), in a very influential paper, showed that comparing learners on the basis of repeated resampling of the same data set can lead to very high Type-I errors. This result is quite generally known, and the paired \( t \)-test methods discussed in that paper are generally considered discredited. Still, based on current practice in cross-validation, it would seem that the underlying reasons for this result are less generally understood, since the construction of confidence intervals on the basis of repeated cross-validation can be expected to suffer from similar problems.

Repeated cross-validation is used quite frequently in the literature. Also, in the experience of the authors, reviewers sometimes insist that cross-validation experiments be based on repeated cross-validation, that confidence intervals are shown, or that it is indicated which of the cross-validation results are "significantly better" than previously published results. In the light of the above work, it would seem obvious that such information, at best, is not very informative, needs careful interpretation by the reader, and is prone to misinterpretation.

Since Dietterich’s paper, there has been a series of re-
Table 1: Overview of the data sets and their properties.

<table>
<thead>
<tr>
<th>data set</th>
<th>instances</th>
<th>attributes</th>
<th>classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>adult</td>
<td>48842</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>kropt</td>
<td>28056</td>
<td>7</td>
<td>18</td>
</tr>
<tr>
<td>letter</td>
<td>20000</td>
<td>17</td>
<td>26</td>
</tr>
<tr>
<td>krvskp</td>
<td>3196</td>
<td>37</td>
<td>2</td>
</tr>
<tr>
<td>mushroom</td>
<td>8124</td>
<td>23</td>
<td>2</td>
</tr>
<tr>
<td>nursery</td>
<td>12960</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>optdigits</td>
<td>5620</td>
<td>65</td>
<td>10</td>
</tr>
<tr>
<td>pageblocks</td>
<td>5473</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>pendigits</td>
<td>9737</td>
<td>17</td>
<td>10</td>
</tr>
</tbody>
</table>

Tables 2 and 3 show the results of these experiments. The symbol + indicates that \( \alpha_2 \) is larger than the upper bound of \( CI \), while − indicates it is smaller than the lower bound of \( CI \).

As can be seen from both tables, the length of the confidence interval decreases with the number of repetitions of cross-validation. However, this does not imply \( \hat{A} \) converges to \( \alpha_2 \). On the contrary, while most of the confidence intervals for \( \hat{A} \) contain \( \alpha_2 \) when using a single cross-validation, most of them do not when using repeated cross-validation. In fact, the number of intervals containing \( \alpha_2 \) decreases with the number of repetitions. As mentioned before, repeated cross-validation improves the estimate of \( \mu_k \) and this result demonstrates that \( \mu_k \) is not necessarily close to \( \alpha_2 \).

Another observation is that in most cases where \( \alpha_2 \notin CI \), \( \alpha_2 \) lies to the right of \( CI \). This shows that there is a pessimistic bias, which is consistent with our expectations (as cross-validation models are learned from a subset of the data, they tend to be less accurate). It also shows that, for repeated cross-validation, this bias is often larger than half the width of the confidence interval; for a single cross-validation this is typically not the case.

One might argue that this problem can be avoided by giving the confidence intervals the same width as those constructed from a single cross-validation. In this case, one would expect the confidence intervals to contain \( \alpha_2 \) about as frequently as when a single cross-validation is used, or even slightly more frequently, if the point estimates obtained are closer to \( \alpha_2 \). However, when inspecting the tables, we see that this is certainly not always, and often only marginally, the case.

Lastly, we look at the influence of the sample size on the estimates. The slope of a learning curve is typically high around small training set sizes, and decreases with an increasing training set size. As a result, the pessimistic bias caused by not using all the available data should be large for small sample sizes and decreases with increasing sample size. Table 4 confirms this by showing that the difference between \( \hat{A} \) and \( \alpha_2 \) is in most cases smaller for a sample size of 1000. However, a comparison of Table 2 and Table 3 shows that increasing the size of \( S \) from 200 instances to 1000 instances does not substantially increase the number of correct confidence intervals.

5. Conclusions

Repeated cross-validation is often advocated for the evaluation of models in machine learning, the argument being that cross-validation estimates have high variance, which can be reduced by using the mean of multiple cross-validations as an estimate. In this paper, we have argued that, due to the fact that the same data set is continuously resampled in cross-validation, this mean converges to another value than any of the
Table 2: The accuracy results for C4.5 and Naive Bayes (N.B.) with the sample size of $S$ equal to 200 instances, computed on different data sets by tenfold cross-validation, 10× repeated tenfold cross-validation, and 30× repeated tenfold cross-validation. The last column shows the population accuracy $\alpha_2$ computed on $D \setminus S$.

<table>
<thead>
<tr>
<th>C4.5 data set</th>
<th>cross-validation</th>
<th>10× cross-validation</th>
<th>30× cross-validation</th>
<th>Pop. $\alpha_2$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mushroom</td>
<td>97.0 (94.64, 99.36)</td>
<td>96.9 (96.14, 97.66)</td>
<td>96.78 (96.34, 97.23)</td>
<td>98.56</td>
</tr>
<tr>
<td>nursery</td>
<td>97.0 (94.64, 99.36)</td>
<td>97.0 (96.25, 97.75)</td>
<td>96.78 (96.34, 97.23)</td>
<td>98.56</td>
</tr>
<tr>
<td>optdigits</td>
<td>85.0 (80.05, 89.95)</td>
<td>84.8 (83.23, 86.37)</td>
<td>84.75 (83.84, 85.66)</td>
<td>83.06</td>
</tr>
<tr>
<td>pageblocks</td>
<td>92.0 (88.24, 95.76)</td>
<td>92.25 (91.08, 93.42)</td>
<td>92.18 (91.5, 92.86)</td>
<td>92.89</td>
</tr>
<tr>
<td>pendigits</td>
<td>73.5 (67.38, 79.62)</td>
<td>74.75 (72.85, 76.65)</td>
<td>74.58 (73.48, 75.69)</td>
<td>76.87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N.B. data set</th>
<th>cross-validation</th>
<th>10× cross-validation</th>
<th>30× cross-validation</th>
<th>Pop. $\alpha_2$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mushroom</td>
<td>98.7 (98.0, 99.4)</td>
<td>99.01 (98.82, 99.2)</td>
<td>98.96 (98.85, 99.07)</td>
<td>99.07</td>
</tr>
<tr>
<td>nursery</td>
<td>88.09 (86.08, 89.09)</td>
<td>87.54 (86.89, 88.19)</td>
<td>87.51 (87.13, 87.88)</td>
<td>91.21</td>
</tr>
<tr>
<td>optdigits</td>
<td>83.6 (81.31, 85.89)</td>
<td>83.9 (82.57, 84.03)</td>
<td>83.53 (83.11, 83.95)</td>
<td>82.42</td>
</tr>
<tr>
<td>pageblocks</td>
<td>95.9 (94.67, 97.13)</td>
<td>96.08 (95.7, 96.46)</td>
<td>95.95 (95.68, 96.12)</td>
<td>95.98</td>
</tr>
<tr>
<td>pendigits</td>
<td>86.6 (84.49, 88.71)</td>
<td>87.6 (86.95, 88.25)</td>
<td>87.65 (87.28, 88.03)</td>
<td>87.41</td>
</tr>
</tbody>
</table>

Table 3: The accuracy results with the sample size of $S$ equal to 1000 instances.

<table>
<thead>
<tr>
<th>C4.5 data set</th>
<th>cross-validation</th>
<th>10× cross-validation</th>
<th>30× cross-validation</th>
<th>Pop. $\alpha_2$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mushroom</td>
<td>98.7 (98.0, 99.4)</td>
<td>99.01 (98.82, 99.2)</td>
<td>98.96 (98.85, 99.07)</td>
<td>99.07</td>
</tr>
<tr>
<td>nursery</td>
<td>88.09 (86.08, 89.09)</td>
<td>87.54 (86.89, 88.19)</td>
<td>87.51 (87.13, 87.88)</td>
<td>91.21</td>
</tr>
<tr>
<td>optdigits</td>
<td>83.6 (81.31, 85.89)</td>
<td>83.9 (82.57, 84.03)</td>
<td>83.53 (83.11, 83.95)</td>
<td>82.42</td>
</tr>
<tr>
<td>pageblocks</td>
<td>95.9 (94.67, 97.13)</td>
<td>96.08 (95.7, 96.46)</td>
<td>95.95 (95.68, 96.12)</td>
<td>95.98</td>
</tr>
<tr>
<td>pendigits</td>
<td>86.6 (84.49, 88.71)</td>
<td>87.6 (86.95, 88.25)</td>
<td>87.65 (87.28, 88.03)</td>
<td>87.41</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N.B. data set</th>
<th>cross-validation</th>
<th>10× cross-validation</th>
<th>30× cross-validation</th>
<th>Pop. $\alpha_2$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mushroom</td>
<td>98.7 (98.0, 99.4)</td>
<td>99.01 (98.82, 99.2)</td>
<td>98.96 (98.85, 99.07)</td>
<td>99.07</td>
</tr>
<tr>
<td>nursery</td>
<td>88.09 (86.08, 89.09)</td>
<td>87.54 (86.89, 88.19)</td>
<td>87.51 (87.13, 87.88)</td>
<td>91.21</td>
</tr>
<tr>
<td>optdigits</td>
<td>83.6 (81.31, 85.89)</td>
<td>83.9 (82.57, 84.03)</td>
<td>83.53 (83.11, 83.95)</td>
<td>82.42</td>
</tr>
<tr>
<td>pageblocks</td>
<td>95.9 (94.67, 97.13)</td>
<td>96.08 (95.7, 96.46)</td>
<td>95.95 (95.68, 96.12)</td>
<td>95.98</td>
</tr>
<tr>
<td>pendigits</td>
<td>86.6 (84.49, 88.71)</td>
<td>87.6 (86.95, 88.25)</td>
<td>87.65 (87.28, 88.03)</td>
<td>87.41</td>
</tr>
</tbody>
</table>
On Estimating Model Accuracy with Repeated Cross-Validation

Table 4: A comparison of the differences between $\hat{A}$ and $\alpha_2$ for sample sizes 200 and 1000. The symbol ‘*’ indicates a case where the difference is smallest for sample size 200.

<table>
<thead>
<tr>
<th></th>
<th>C4.5</th>
<th>Naive Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cross-validation</td>
<td>10×cross-validation</td>
</tr>
<tr>
<td>$</td>
<td>S</td>
<td>$</td>
</tr>
<tr>
<td>adult</td>
<td>200 1000</td>
<td>200 1000</td>
</tr>
<tr>
<td>kropt</td>
<td>4.08 1.08</td>
<td>4.23 0.41</td>
</tr>
<tr>
<td>letter</td>
<td>1.32 3.02</td>
<td>0.72 3.81</td>
</tr>
<tr>
<td>krskp</td>
<td>3.07 1.18</td>
<td>2.97 0.9</td>
</tr>
<tr>
<td>mushroom</td>
<td>1.56 0.16</td>
<td>1.56 0.15</td>
</tr>
<tr>
<td>nursery</td>
<td>1.94 0.92</td>
<td>1.74 0.43</td>
</tr>
<tr>
<td>optdigits</td>
<td>12.09 0.42</td>
<td>7.29 0.55</td>
</tr>
<tr>
<td>pageblocks</td>
<td>0.89 0.58</td>
<td>0.64 0.85</td>
</tr>
<tr>
<td>pendigits</td>
<td>3.37 1.29</td>
<td>2.12 0.76</td>
</tr>
<tr>
<td>adult</td>
<td>6.06 0.84</td>
<td>5.06 0.99</td>
</tr>
<tr>
<td>kropt</td>
<td>0.17 1.25</td>
<td>5.06 1.21</td>
</tr>
<tr>
<td>letter</td>
<td>4.78 0.99</td>
<td>0.13 0.61</td>
</tr>
<tr>
<td>krskp</td>
<td>6.92 0.80</td>
<td>3.28 0.42</td>
</tr>
<tr>
<td>mushroom</td>
<td>0.62 1.06</td>
<td>5.32 1.02</td>
</tr>
<tr>
<td>nursery</td>
<td>3.58 0.18</td>
<td>0.97 0.32</td>
</tr>
<tr>
<td>optdigits</td>
<td>2.53 0.67</td>
<td>3.73 0.06</td>
</tr>
<tr>
<td>pageblocks</td>
<td>2.48 2.27</td>
<td>1.03 2.02</td>
</tr>
<tr>
<td>pendigits</td>
<td>2.30 0.23</td>
<td>3.88 0.42</td>
</tr>
</tbody>
</table>

values one might really be interested in estimating. Repeated cross-validation should not be assumed to give much more precise estimates of a model’s predictive accuracy. The pessimistic bias due to the fact that cross-validation models are learned from smaller data sets (in the paper’s notation, $\alpha_3 - \alpha_2$), together with the bias introduced by using a single data set ($\mu_k - \alpha_3$), can easily dominate the estimation error, which means reducing the variance is, in many cases, not very useful, and essentially a waste of computational resources.

Acknowledgments

This research was supported by the Research Foundation-Flanders (FWO Vlaanderen); Project G.0179.10 Multi-objective compiler optimization space exploration.

References


Statistical Relational Learning for Prognostics

Jonas Vlasselaer
Wannes Meert
KU Leuven, Department of Computer Science, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Keywords: Prognostics, Temporal models, Statistical Relational Learning

Abstract

The field of prognostics aims to predict the remaining useful life of a component or machine by means of probabilistic models. These models typically need to satisfy different requirements imposed by the available data, the expert knowledge and the prediction task at hand. Graphical probabilistic models are popular tools in prognostics but satisfy these requirements only partially. We introduce some examples to illustrate these requirements, analyze the most suitable graphical model and propose to use Statistical Relational Learning (SRL) for prognostics to satisfy more requirements. The expressiveness offered by this framework, however, has a computational cost and in this paper we investigate how current state-of-the-art SRL systems perform on prognostics tasks.

1. Introduction

The affordance of computing power, data storage and sensors make automated monitoring increasingly appealing for the mechatronics industry. This has lead to machines that are monitored more closely and detect signal deviations from normal behavior at a much earlier stage. But not only deviations can be observed, more complex probabilistic models also allow for prediction of future states of the machine. Where previously an expert was needed to inspect a machine and estimate how the machine would evolve over time (e.g., degradation), this can now be (partially) automated.

In this paper we focus on models for prognostics and more specifically on the prediction of the Remaining Useful Life (RUL) of a machine and its components. Determining the RUL accurately has a positive impact on the efficiency of a plant. Popular methods for prognostics are probabilistic (graphical) models like (hidden) Markov models, hidden semi-Markov models, support vector machines and Bayesian networks. Although these models are able to efficiently deal with particular temporal processes, the constraints they impose are often too limiting for a real-world scenario. Often, a hierarchical model, or a model with functional and relational dependencies between classes of objects, or a combination of different models is necessary. Furthermore, most probabilistic models require in-depth knowledge to build such models. Ideally, a domain expert would be able to express his knowledge in a declarative, modular manner without the need to understand the underlying algorithms.

Statistical Relational Learning (SRL) techniques (Getoor & Taskar, 2007; De Raedt & Kersting, 2003) extend the probabilistic model framework with relational and functional elements, allows us to express expert or background knowledge about the system, and can express structural information such as time. The more logic based techniques in the field of SRL also allow a declarative and modular way of expressing knowledge. As we are mainly interested in formalisms that are flexible enough to fulfill all requirements, we will focus on logic based SRL techniques.

In this paper we investigate whether SRL could offer a solution to some of the (artificial) restrictions we encounter in the field of prognostics when using traditional probabilistic models. As a first contribution we identify a set of requirements that are often encountered in a real-world prognostics setup. The second contribution is a comparison of how current state-of-the-art SRL systems perform on a set of models adhering to the requirements defined in this paper. We are mainly interested in the trade-off between efficiency and the gained expressiveness.

Appearing in Proceedings of BeneLearn and PMLS 2012. Copyright 2011 by the author(s)/owner(s).
2. Prognostics

In this section, we introduce the field of prognostics and enumerate some specific requirements inherent to this field. These requirements are distilled from the real-life data sets and problems we encountered in the context of the Prognostics for Optimal Maintenance project.\(^1\) This project aims to automate and optimize the maintenance strategy for industrial machinery.

2.1. Background

Maintenance is typically one of the biggest costs for large manufactures and therefore the choice of a good maintenance strategy is crucial. In the search for an optimal strategy, companies have to deal with a trade-off: if one waits too long to do maintenance, an unexpected breakdown of a component or machine can result in a loss of productiveness; a more progressive strategy, on the other hand, ensures a continuous productivity process but results often in an unnecessary replacement of components.

To find the most optimal moment for maintenance, particular information about the machine or its components is needed. This information is mostly obtained by a set of sensors which measure specific characteristics of the component at a constant rate. An expert typically searches irregularities or trends in this temporal data and decides which machine needs more attention because of the risk of a breakdown. In this way, the expert tries to estimate the Remaining Useful Life (RUL) of the system, given the sensor information up until that moment and his past experience or knowledge about the machine.

Although an expert can often make an accurate estimate of the RUL, there are several reasons why companies are interested to use machine learning techniques to (partially) automate this task. Information management: knowledge needs to be managed such that it can be reused once an expert leaves the company. Scalability: an expert is limited in time and space and therefore multiple similar experts could be needed to monitor similar machines, especially if they are spread over different locations. Complexity: machine learning techniques can typically handle more information or explore more complex relations between machines and components and as such improve the accuracy of the predictions. Prognostics as a service: besides prognostics for the manufacturing process one could think about prognostics for the products a company fabricates. In this case it is clear only automated prognostics has a real benefit.

To estimate the RUL of a system we first need to model its degradation and next predict the moment where this model will cross a certain threshold. Unfortunately, most components do not allow us to observe the degradation directly without shutting down the machine. For example, the real degradation of ball bearings can often only be observed if we open them up and inspect the condition of the balls and the inside of the cage. Because the real degradation is hidden, it is necessary to measure other characteristics and use them to infer knowledge about the condition of the component. This relation is mostly probabilistic because sensors occasionally make some measurement errors or they can be influenced by external, and probably unknown, events.

Prognostics is not only relevant for components and machines, similar models and techniques are useful for other applications. One of them is health care for elderly persons in service flats or retirement homes. Different sensors could be used to infer some information about the hidden health condition of the residents. The goal would be to predict a dramatic change in the health condition of a person such that medical personnel can be notified in time.

2.2. Requirements

Based on the data and tasks involved with a prognostics problem, we can formulate a set of requirements which a probabilistic model, ideally, fulfills to help solve a prognostics problem:

1. **Handle probabilistic and partial observable data.** The real degradation is typically unobservable and its relation with the measured data is often probabilistic.

2. **Efficient inference and learning for temporal data.** Multiple sensor measurements per day over a period of several weeks or months accumulates into a large temporal dataset. More observations lead in general to a better estimation of the RUL and therefore efficient inference is necessary. Because the parameters in the model are ideally learnable from observation sequences, efficient learning is needed.

3. **Continuous distributions to model duration.** To plan maintenance in the most optimal way, an estimation of the time until failure is needed. This can be, for example, a (mixture of) Gaussian or Gamma distribution.

4. **Multiple observations for every time instance, possibly related.** Different sensors can be used to measure several characteristics of a

\(^1\)POM2 IWT-SBO project (http://www.pom2sbo.org)
component and this results in multiple observations per time instance. The existing relations between these observations are ideally included in the model to improve accuracy.

5. Structural and modular representation of how a system evolves over time. A structural and modular representation allows the expert to represent his knowledge in a more abstract way, independent from all the details of the system. Additionally, it allows to easily reuse or transfer parts of the model to other similar systems.

6. A unifying framework to represent knowledge about different types of interactions. A domain expert should be able to write down his knowledge about a system without the need to be an expert in probabilistic modeling or algorithms.

3. Temporal model approaches

In this section, we introduce some examples of increasing complexity to illustrate the problems encountered in prognostics. We relate these problems to the requirements defined above and briefly analyze the most suitable temporal model.

3.1. Hidden Markov models

The first problem is to model the degradation of a ball bearing given a single sensor measurement for each time point. The real degradation is hidden but is assumed to have an influence on the vibration amplitude measured at the outside of the bearing. This can be modeled with a Hidden Markov Model (HMM) (Rabiner, 1989) which represents the degradation as a finite set of hidden states related according to a certain probability distribution. As shown in figure 1(a), the degradation (D) is a cause for the sensor to measure a certain vibration amplitude (V). A HMM assumes the underlying process to be memoryless and therefore allows efficient inference and parameter learning for long sequences of data.

3.2. Hidden-semi Markov models

A HMM can be used to model the ball bearing in our first example, and satisfies as such the first two requirements defined in section 2.2, but is too limited to model the RUL of a component. Assume the degradation can be modeled with two hidden states, the RUL is equivalent to the time the model stays in the G(ood) state and makes a transition towards the B(ad) state. This is by definition an exponential distribution (Rabiner, 1989) and therefore limits the capability of an HMM to model the real distribution over the RUL.

A Hidden semi-Markov Model (HsMM) allows the underlying process to be a semi-Markov chain and explicitly defines the duration or sejourn time for every state (Yu, 2010). This duration is related to the number of observations produced while in the state, as shown in figure 1(b), and therefore the self-transition probabilities are equal to zero. A HsMM allows the state duration to be modeled by, for example, a Gaussian or Gamma distribution and satisfies as such the third requirement. The duration specified for the G(ood) state can be the lifetime of the ball bearing, specified by the producer with a certain mean and variance.

3.3. Dynamic Bayesian networks

Until now, we assumed to have only one observation for every time instance but in general also more observations could be available. For example, the vibration caused by a ball bearing typically consist of two orthogonal components which can be measured separately with two different sensors. Because it is cheaper to use only one sensor and measure the composed signal, the separate components are only measured if the composed signal reaches a threshold and the expert decides to inspect the machine with two separate sensors. This results in a partial observed dataset where the relations between the signals, as shown in 1(c), are incorporated in the model.

A Dynamic Bayesian Network (DBN) (Mihajlovic & Petkovic, 2001) is a generalization of a HMM used to model more complex temporal data. Besides the relations between multiple observations, a DBN allows to model more complex relations between the hidden states and satisfies as such the fourth requirement defined in section 2.2. Note that the term “dynamic” implies that a dynamic system is modeled and does not mean the structure of the model changes over time.

3.4. Logical hidden Markov models

The models introduced in the previous examples make use of observations, caused by the hidden process, to estimate the degradation of the component. On the other hand, experts might have knowledge on how the environment influences the degradation. For example, a ball bearing tends to degrade faster when it is used in a humid environment and therefore the transition probabilities of the hidden process depend on the observed humidity. In this way, this observation (C) is rather an input for the model as shown in figure 1(d).

This type of relations can be modeled with a logical HMM (LOHMM) (Kersting et al., 2006), a inductive logic programming setting that upgrades HMMs to deal with structure. A LOHMM extends the un-
structured representation of transitions in an HMM by utilizing modular rules and relational information to define the transitions between states. This allows us to condition the transition probabilities on the context, for example humidity, and as such a LOHMM satisfies the fifth condition.

3.5. Probabilistic logic programming

Analogous to more complex relations between multiple observations, in some cases, more expressibility is needed to represent the structure of the context. For example, it is typically not possible to measure whether water reaches the inner side of a bearing but one can measure whether water is present at other parts of the machine. If the expert knows how the water possibly flows through the parts of the machine, he can represent this knowledge as a probabilistic graph as shown in figure 1(e). In this case the degradation of the component, and therefore the transition probabilities, is influenced by the probability of different paths in this graph.

ProbLog (De Raedt et al., 2007) is a probabilistic logic programming language which can be used to represent a probabilistic graph, compute the probability of paths in this graph and to model the hidden temporal process. Furthermore, it can be used to represent all problems introduced before and therefore satisfies our sixth requirement. Additionally, ProbLog allows us to describe knowledge in a declarative way. This is an important advantage since there is no direct mapping between a graph, represented as a set of probabilistic edges, and the nodes in a Bayesian network.

We can situate ProbLog in the more general setting of Statistical Relational Learning (SRL) (Getoor & Taskar, 2007; De Raedt & Kersting, 2003). This framework aims to integrate probabilistic reasoning mechanism with machine learning and relational or logical representations and as such allows us to incorporate different types of background or expert knowledge in a more structured and modular way. Years of research has lead to different SRL systems, each with their own strengths and limits, and therefore it is useful to compare some of these systems.

4. Experiments

As experiment, we modeled the five examples discussed in section 3, and shown in figure 1, in four state-of-the-art SRL systems: PRISM, ProbLog, Primula and Alchemy. We discuss and compare these systems and are especially interested in the trade-off between expressiveness and efficiency.

4.1. PRISM

PRISM (PRogramming In Statistical Modeling) (Sato & Kameya, 1997) is a logic-based probabilistic language implemented on top of Prolog. It combines SLD-resolution with probability calculations to compute the success probability of a query. A combination of linear tabling and restrictions on the programs makes inference and learning for a HMM as efficient as the specialized algorithms used in graphical representations.

A limiting restriction for PRISM is the exclusiveness condition which states that all disjunctive paths in a proof tree are required to be probabilistic exclusive. Because of this restriction, it is not possible to model, for example, a graph and define a path predicate. Another issue is the exponential blow-up for inference in (dynamic) Bayesian networks. In these networks inference is exponential in the number of nodes because the success probability is computed by exhaustively computing the probability of every possible world. PRISM allows the use of a junction tree such that inference is only exponential in the size of the largest clique but this requires the BN to be represented in an unnatural form and typically requires an expert.

4.2. ProbLog

The inference mechanism of ProbLog also uses SLD-resolution to find all the proofs of a query but then uses a Binary Decision Diagram (BDD), a more compact and efficient representation of the proofs.
Table 1. The results of inference (I) and learning (L) for the different examples defined in section 3 and the systems introduced in section 4. HsMM1 makes use of a random (non exponential) duration and HsMM2 of a Gaussian duration.

<table>
<thead>
<tr>
<th></th>
<th>HMM</th>
<th>HsMM1</th>
<th>HsMM2</th>
<th>DBN</th>
<th>LOHMM</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRISM</td>
<td>I</td>
<td>L</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>ProbLog</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primula</td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alchemy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

cause a BDD deals with the disjoint sum problem, no restrictions need to be imposed on the program and therefore ProbLog can be used to model a graph and a path predicate. Compared with PRISM, inference in a BN is more efficient because in a BDD representation the required success probability can be computed in polynomial time.

The drawback of the expressiveness offered by ProbLog is the loss of efficiency for inference and parameter learning if it is used for temporal models like a HMM. The proofs for these models are by definition mutually exclusive and there is no need to compile them in a BDD to tackle the disjoint sum problem. In this case, compiling the proofs is unnecessary and potentially intractable for large time sequences.

4.3. Primula
Primula is an SRL system for modeling and inference with random relational structure models, more specific relational Bayesian networks (RBNs) (Jaeger, 1997). RBNs allow to define relations between different random events and combine as such a relational representation with Bayesian networks. Different from the two aforementioned systems, Primula compiles the complete network off-line based on the domains defined in a structure file and therefore allows fast inference for multiple queries. As a drawback, the domain for the network needs to be specified in the model itself which is not desired for temporal models.

4.4. Alchemy
The Alchemy system is based on the Markov logic representation (Richardson & Domingos, 2006), a combination of first-order logic and Markov networks. Markov logic can be used to represent a HMM but requires to convert probabilities into weights which is in practice very hard. Although the transition probabilities in a HMM are independent to the length of the sequence, the weights in a Markov network depend on the sequence length (Bruynooghe et al., 2009). This is undesirably because it restricts the expert to represent his knowledge of the system.

4.5. Other related systems
In the previous sections we introduced four different SRL systems but more systems exist and some of them were already compared for a (partial) different set of problems (Bruynooghe et al., 2009). Although this paper compares a larger set of systems, we deliberately chose not to test these other systems because for some of them it was already shown they cannot be used to model a HMM. Other systems are not supported anymore and are therefore also left out.

Some work specifically aims to improve the efficiency of inference and learning for temporal models with a structural or relational representation. One of them is Causal Probabilistic Time-Logic (CPT-L) (Thon et al., 2011) which defines a probability distribution over a sequence of interpretations. This technique is restricted to fully observable data and can therefore not be used to model systems where part of the data is hidden or unobserved for multiple time instances.

4.6. Comparison
The results for the four SRL systems are shown in Table 1 for inference (I) and learning (L). A white box indicates the task can be executed efficiently, a gray box indicates the task can be executed but is suboptimal and a black box indicates the task cannot be executed by the system. Because of the drawbacks of Primula and Alchemy for the intended models, we do not further include them in our comparison.

Inference and parameter learning for a simple HMM is, as expected, the most efficient in PRISM. This system computes the probability distribution over the hidden states for the 50000th time instance within seconds while ProbLog fails to compute the same distribution for the 25th time instance within a minute. The efficiency of ProbLog is somewhat improved if tabling is enabled (Mantadelis & Janssens, 2010) but it still fails
to compute the distribution for the the 1000th time instance. ProbLog supports as well learning from entailment as learning from interpretations but both of them are not able to learn the parameters of the model if the observed data sequences are too long. Learning from entailment uses a k-best approach, without tabling, as underlying inference algorithm which is intractable for observation sequences larger than 25 time instances. These sequences can also not be used for the learning from interpretations setting because building the BDD is exponential and runs out of memory.

Both PRISM and ProbLog can be used to model a HsMM with a random, non-exponential, distribution over the duration (HsMM1). There is however a difference between both systems if we use them to model a HsMM with a continuous distribution over the duration (HsMM2). ProbLog supports continuous facts and accompanying predicates to model a Gaussian distribution and these are not available for PRISM. Recent work introduced a multivariate Gaussian switch for PRISM (Islam et al., 2011), but no interval checks are supported and therefore these distributions cannot be used to model a continuous duration.

In general we observe a trade-off between efficiency and expressiveness. ProbLog can be used to model the five different examples but fails to offer the same efficiency as PRISM for the temporal process. As a drawback to the efficiency offered by PRISM, it cannot be used to model a graph.

5. Conclusion

We proposed a set of requirements to model the Remaining Useful Lifetime of machines in the field of prognostics. These requirements allow us to identify the restriction enforced by probabilistic models to be able to perform efficient inference and learning. Furthermore, we experimentally compared four state-of-the-art implementations of Statistical Relational Learning approaches.

We argue that expressive, declarative models offer a language closer to the domain expert. On the other hand, efficiency is not to be ignored. The experiments have shown that SRL languages potentially allow for a trade-off within the same unifying language. Therefore, in future work, we want to develop further methods to automatically detect parts of the model where we can utilize efficient methods, thus allowing full expressiveness but with efficient inference where possible. Furthermore, more work on continuous distributions is needed because most of these distributions offered in SRL systems are still limited and experimental.
Distance Dependent extensions of the Chinese Restaurant Process

Willem Wybo
Camille Colle
Pieter-Jan Kindermans
Benjamin Schrauwen
Ghent University, Electronics and Information Systems department, Sint Pietersnieuwstraat 41, 9000 Ghent, Belgium

Keywords: Chinese Restaurant Process, Distance Dependent Chinese Restaurant Process, Averaged Distance Dependent Chinese Restaurant Process

Abstract
In this paper we consider the clustering of text documents using the Chinese Restaurant Process (CRP) and extensions that take time-correlations into account. To this purpose, we implement and test the Distance Dependent Chinese Restaurant Process (DD-CRP) for mixture models on both generated and real-world data. We also propose and implement a novel clustering algorithm, the Averaged Distance Dependent Chinese Restaurant Process (ADDCRP), to model time-correlations, that is faster per iteration and attains similar performance as the fully distance dependent CRP.

1. Introduction
Non-parametric clustering algorithms have been used often in the classification of text documents. These algorithms exist in plenty of variations, that are generally referred to through some metaphor with a restaurant where exotic cuisine is served. The simplest of them is the Chinese Restaurant Process, and it will provide a starting point for the discussions in this paper.

However, these algorithms generally assume that the data is independent and identically distributed (iid). Such an assumption is questionable in many cases. Today’s newspaper articles show a high correlation with what was in the news yesterday, and scientific papers tend to be generated in areas were researchers see interest, generally through what was written earlier by their colleagues.

Recently a new clustering algorithm, the Distance Dependent Chinese Restaurant Process, was proposed (Blei & Frazier, 2011), that drops this assumption of iid-draws by including a dependence on the distance between data points. Note that the DD-CRP can be used for many applications. In this work we restrict ourselves to document clustering with documents that have a time stamp and are assumed to exhibit time-correlations. We test and compare this algorithm with the normal CRP and also propose a novel variation to it, the Averaged Distance Dependent CRP, where the distance is defined between data points and clusters. We also test this new algorithm and compare it with the previous ones.

This paper is structured as follows: first we provide a short introduction to the algorithms, where the focus is not on completeness but on summarizing the necessary concepts. Next we show what draws from all three processes look like and finally we discuss our testing method and apply it to a real-world dataset.

2. Algorithms
In document clustering, documents are often modelled as a bag of words, in which the order of the words is ignored. Through this assumption, the documents can be modelled as if they are generated by a latent topic, where the topic governs the parameters of the multinomial distribution from which their words are drawn. These parameters are assumed to be drawn from a Dirichlet prior, that is unobserved, but can be learned from the data.
More precisely, a list of all unique words \( w = (w_1, \ldots, w_N) \) that occur in the documents can be constructed. A document is then represented by a vector \( x = (n_1, \ldots, n_N) \), where \( n_i \) denotes the number of times the word \( w_i \) appears in the document. If a document belongs to a cluster, it is assumed to have been generated from a multinomial distribution with parameters determined by that cluster. Denoting the parameters of this distribution as \( \theta = (\theta_1, \ldots, \theta_N) \), with \( \sum_n \theta_n = 1 \), the probability to find a certain document \( x \) is given by

\[
p(x \mid \theta) \sim \theta_1^{n_1} \cdots \theta_N^{n_N}
\]

In the algorithms discussed in this paper, the probability of the parameters \( \theta \) is assumed to follow a Dirichlet distribution. We denote this distribution as \( G_0 \) and its parameters as \( g_1, \ldots, g_N \). Thus the likelihood of a given vector \( \theta \) is

\[
p(\theta \mid G_0) \sim \theta_1^{g_1-1} \cdots \theta_N^{g_N-1}
\]

In mixture models the parameters \( g_1, \ldots, g_N \) are determined by the documents already present in the cluster.

### 2.1. The Chinese Restaurant Process

The Chinese Restaurant process is a process that generates a distribution over partitions (topics) from which sampling is possible (Neal, 2000). The simplest of the sampling methods is based on Gibbs sampling (Bishop, 2007) and through this sampling the CRP becomes a powerful clustering algorithm, based on probabilistic cluster assignments. It has many interesting mathematical properties that are reviewed in (Teh et al., 2006). We repeat only the very basics.

In the CRP, a data point can be assigned to previously formed clusters with a probability that is proportional to the amount of points already in such a cluster. A data point can also be assigned to a new cluster with a certain probability. This can be expressed intuitively through a metaphor, from which the CRP borrows its name, where customers successively enter a Chinese restaurant and decide to sit at tables (the clusters that will be formed) with a probability proportional to the amount of customers already sitting at the table. They can also decide to sit at a new table with a certain probability proportional to a fixed constant. If we denote the data point that is to be assigned as \( i \), the cluster assignment of \( i \) as \( c_i \), the partition of the previous data points defined by the clustering as \( z_{i-1} = (z_1, \ldots, z_i) \) and the constant to which the probability to start a new cluster is proportional as \( \alpha \), the previous considerations can be written as:

\[
p(c_i = z_k \mid z_{i-1}, \alpha) \propto \begin{cases} n_k & \text{if } z_k \in z_{i-1} \\ \alpha & \text{if } z_k = \text{new table} \end{cases}
\]

where \( n_k \) denotes the number of data points in cluster \( k \). As described previously, documents are considered to be generated from a multinomial distribution, with parameters determined by the cluster assignment. The probability that a document \( x_i \) is assigned to a cluster \( z_k \), containing a set of documents \( \{x_j \in z_k\} \), is then also proportional to the predictive probability that this document could have been generated by this cluster. This predictive probability is obtained by integrating out the parameters \( \theta \) weighted by their Dirichlet likelihood:

\[
p(x_i \mid \{x_j \in z_k\}) \propto \int p(x_i \mid \theta)p(\theta \mid \{x_j \in z_k\}, G_0)d\theta
\]

The Gibbs sampling algorithm for the CRP then successively removes documents from their cluster and reassigns them to one of the other clusters according to the appropriate probabilities. These probabilities follow from combining equation (3) and (4) and are given by

\[
p(c_i = z_k \mid z_{i-1}, x, G_0) \propto \begin{cases} n_k \int p(x_i \mid \theta)p(\theta \mid \{x_j \in z_k\}, G_0)d\theta & \text{if } z_k \in z_{i-1} \\ \alpha & \text{if } z_k = \text{new table} \end{cases}
\]

Since the multinomial and Dirichlet are conjugate distributions, a closed form of integral 4 exists, allowing for computationally effective sampling.

### 2.2. The Distance Dependent Chinese Restaurant Process

The distance dependent Chinese restaurant process can be seen as an extension of the normal CRP. However, now the customers don’t sit at tables, but are linked to each other, and the tables arise merely as clusters of connected customers. The customers are assumed to have a sequential variable (e.g., a time stamp) through which we can calculate a distance \( d_{ij} \). The probability for a customer \( i \) to sit with customer \( j \) is then

\[
p(c_i = j \mid D, \alpha) \propto \begin{cases} f(d_{ij}) & \text{if } j \neq i \\ \alpha & \text{if } j = i \end{cases}
\]
where $f$ is some kind of decaying function, $D$ the distance matrix and $c_i$ represents the customer to which $i$ links. To model time correlations one usually takes
\[ d_{ij} = \infty \] if $j$ has a larger time stamp then $i$, so that no customer can be assigned to a future customer. This decay function can be chosen so as to suit the modellers needs. Good choices for $f$ to model time-correlation include an exponential function or a logistic decay function. If $f(d) = 1$ for $d < \infty$, the distance dependent CRP reduces to the normal CRP.

Since in the DDCRP tables arise as clusters of connected customers, the Gibbs sampler will have a slightly different form than the Gibbs sampler for the normal CRP. Instead of doing probabilistic cluster assignments, the Gibbs sampler will now do probabilistic customer assignments. These assignments probabilities are proportional to (6) and to the appropriate mixture probabilities. If $\mathbf{z}(c_{-i})$ is a partition with clusters $(z^1, \ldots, z^n)$, formed by the links $c_j$ ($j \neq i$), then the probability that the customer $i$ will have a link $c_i$ is given by
\[
p(c_i = j | c_{-i}, \mathbf{x}, \alpha, D, G_0) \propto \begin{cases} 
\alpha & \text{if } c_i = i \\
f(d_{ij}) & \text{if } c_i \text{ does not join two clusters} \\
f(d_{ij}) \frac{p(\mathbf{x} | \mathbf{z}(c_{-i}) \cup \{c_i\})}{p(\mathbf{x} | \mathbf{z}(c_{-i}))} & \text{if } c_i \text{ joins clusters } k \& l
\end{cases}
\] (7)

Again, as we are modelling text documents, we have words that are drawn from a multinomial distribution and clusters from a Dirichlet distribution. Hence the posterior probability
\[
p(\mathbf{x} | \mathbf{z}(c) \cup \{c_i\}) = \int \prod_{x_j \in z_i(c)} p(x_j | \theta)p(\theta | G_0)d\theta
\] (8)
can again be written in closed form through their respective conjugacy.

2.3. The averaged distance dependent CRP

In this section we introduce a novel clustering algorithm, the averaged distance dependent CRP (ADDCRP), that is a hybrid between the DDCRP and the normal CRP. In this algorithm, the distance is no longer defined between individual data points, but between data points and clusters, through an averaging procedure. If $\mathbf{z}(c_{-i})$ is a partition $(z^1, \ldots, z^n)$ formed by the cluster-assignments $(c_1, \ldots, c_{i-1})$, and $t_i$ denotes the time stamp of the $i$'th document, the distance between a data point $i$ and a cluster $z^k$ is defined as
\[
d_{ik} = t_i - \frac{1}{|I_{ik}|} \sum_{j \in I_{ik}} t_j
\] (9)
where $I_{ik} = \{j | j \in z^k \land t_j < t_i\}$ is the set of documents in cluster $k$ with time stamps smaller than the time stamp of document $i$. Other definitions of this distance are also possible, for instance by taking a weighted mean
\[
d_{ik} = t_i - \sum_{j \in I_{ik}} \frac{f(t_i - t_j)}{\sum_{l \in I_{ik}} f(t_i - t_l)} t_j
\] (10)
and hence attributing more importance to closer data points, or by simply taking the closest point
\[
d_{ik} = \min_{j \in I_{ik}} (t_i - t_j)
\] (11)
to which we will respectively refer as the weighted ADDCRP and the minimal ADDCRP. Note that if $|I_{ik}| = 0$, we set $d_{ik} = \infty$, so that no data point can be assigned to clusters that contains only later data points.

Cluster assignments are then drawn according to
\[
p(c_i = z_k | D, \mathbf{z}_{i-1}, \alpha) \propto \begin{cases} 
\alpha & \text{for } z_k \in \mathbf{z}_{i-1} \\
f(d_{ik}) n_k & \text{for } z_k = \text{new table}
\end{cases}
\] (12)

It can be readily seen that for $f(d) = 1$ if $d < \infty$ this reduces again to the normal CRP. The probabilities of assignment during a Gibbs-sampling run are given by an analogous formula as equation (5) for the normal CRP, but now they are weighted with a factor $f(d)$.

\[
p(c_i = z_k | \mathbf{z}_{i-1}, \mathbf{x}, G_0) \propto \begin{cases} 
f(d_{ik}) n_k & \int p(x_i | \theta)p(\theta | \{x_j \in z_k\}, G_0)d\theta \\
\alpha & \text{if } z_k \in \mathbf{z}_{i-1}
\end{cases}
\] (13)

3. Drawing from the CRP and its variations

A useful way to gain intuitive understanding in what draws from all three processes look like, is to run them in a generative way. Successive data points are assigned to the previous data points according the formulas (3), (6) and (12) for respectively the normal
Distance Dependent extensions of the Chinese Restaurant Process

CRP, the fully distance dependent CRP and the averaged distance dependent CRP. These assignments are visualized in figure (1) for an exponential decay function \( f(d) = e^{-\beta d} \) and for a few different values of the parameters \( \alpha \) and \( \beta \). As one can see, draws from the DDCRP and the ADDCRP look reasonably similar, as the amount of formed clusters and their average length in time are approximately equal, whereas draws from the CRP are very different. This leads us to believe that the DDCRP and the ADDCRP have similar performance.

For testing purposes, we also generated full ‘documents’ of words drawn from a number of topics. That way we could test and compare models on datasets that are more manageable than real world datasets. We could for instance choose the size of the vocabulary. We now describe the procedure that we used to generate those datasets.

Beside drawing assignments \( c_i \) for data points \( x_i \) by using formulas (3), (6) and (12), words should, in a fully generative context, be drawn from the predictive probabilities

\[
p(x_i \mid \{x_j \in z_k\}) = \int p(x_i \mid \theta) p(\theta \mid \{x_j \in z_k\}, G_0) d\theta
\]

where \( z_k \) is the mixture component to which \( c_i \) points. For practical reasons we followed a simpler approach, where our vocabulary, containing \( W \) words, was split up in a number of \( C \) different classes. For each mixture component, we selected one or two of those classes as main ‘topics’, and generated documents largely from those main topics. On average we selected about two thirds of the words in a document from those main topics in a random fashion. The other third of the words was selected from random other topics in the vocabulary. For all our tests on generated data we used \( W \approx 800 - 900, C = 16 \) and on average 600 words per document.

4. Runtime

In this paragraph we consider the scaling properties of the runtime of all three algorithms as a function of the number of documents \( N \). In the CRP and the ADDCRP a document is assigned to one of the cluster according to equations (5) (CRP) and (12) (DDCRP). If there are \( K \) clusters, one has to calculate this likelihood \( K \) times. Since the samplers successively remove and reassign all elements once every iteration, an iteration scales as \( O(KN) \). We show this effect in figure 2 for data generated by the procedure describe in paragraph 3. Note that these considerations only apply to the runtime per iteration. As Blei and Frazier note, the number of iterations until convergence can be much smaller in the DDCRP, as that sampler can move whole chunks of clusters at once, whereas the samplers for the CRP and ADDCRP only move one element at a time. This depends however heavily on the data at hand.

5. Prediction

The goal of analysing data with time-correlations is often to estimate the likelihood of future data
Figure 2. Average runtime per iteration as a function of the number of documents

points. Suppose that $N$ documents, represented by the vector $x$, are clustered by an algorithm in a partition $(z^1, \ldots, z^n)$ by the cluster assignments $c = (c_1, \ldots, c_N)$ (in the case of the distance dependent CRP, $c$ denotes the element assignments). The total predictive likelihood for a given later document $x_{new}$ is then given by

$$p(x_{new} \mid x, D, G_0, \alpha) = \sum_{c_{new}} \sum_c p(c_{new} \mid c, D, \alpha)p(x_{new} \mid c_{new}, c, x, G_0)p(c \mid x, D, \alpha, G_0)$$

(15)

where $D$ is the distance matrix. The last factor on the right hand side denotes the probability of a given partition, given the model and the data. Since only sequential data is considered, the assignment of a new data point does not change the probability of the previous assignments. Hence the sum over $c$ can be estimated by averaging over different clustering runs. The second factor is the probability of the new data point under the assignment $c_{new}$. This can be computed with the standard inference methods. The first factor is the probability of that assignment. As can be seen from (3), (6) and (12), this probability is in the most general case only dependent on the previous assignments and the distance matrix. Under the distance dependent CRP this probability becomes independent of the previous assignments and (15) reduces to

$$p(x_{new} \mid x, D, G_0, \alpha) = \sum_{c_{new}} p(c_{new} \mid D, \alpha)$$

$$\sum_c p(x_{new} \mid c_{new}, c, x, G_0)p(c \mid x, D, \alpha, G_0)$$

(16)

Under the normal CRP it follows that these assignment probabilities are all equal, and therefore equation (16) simplifies further to

$$p(x_{new} \mid x, D, G_0, \alpha) = \frac{1}{N} \sum_{c_{new}} \sum_c p(x_{new} \mid c_{new}, c, x, G_0)p(c \mid x, D, \alpha, G_0)$$

(17)

This can be seen from the interpretation of a normal CRP as a distance dependent CRP with $f(d) = 1$ if $d < \infty$.

6. Held-out likelihood as a comparative test

To test the performance of our model we follow the approach of Blei and Frazier (Blei & Frazier, 2011) and compute the so-called held-out likelihood. Suppose a dataset contains $M$ documents and the clustering is performed on a smaller number $N$ of earlier documents. The held-out likelihood of a later document is then defined as the predictive likelihood for this document, given the $N$ earlier documents and the clustering. This is nothing else than equation (15). The held-out likelihood is a measure of how well the held-out data can be predicted by the mixture components created by the clustering algorithm. Hence algorithms that achieve a higher likelihood can be seen as better performing in this context.

Blei and Frazier computed this held-out likelihood on real-world datasets of newspaper articles and scientific papers. They concluded that in most cases the DDCRP is a better model than the normal CRP. We replicate those tests on another dataset and also perform them for the ADDCRP. We used abstracts from award winning papers from the National Science Foundation\footnote{http://kdd.ics.uci.edu/databases/nsfabs/}, with submission dates ranging from 1989 to 1995, where we removed a standard list of stop-words and words that appeared only once in an abstract.

If we look at the held-out likelihood in function of the training set size, figure (3a), we see that the likelihood increases as the training set size is increased, the expected behaviour.

In figure (3b) we compare the different clustering models. Both the DDCRP and the ADDCRP perform better than the CRP. Their relative performance is however dependent on the choice of the decay parameter, which has to be adjusted to the data.
7. Conclusion

We have implemented and tested two methods, a new one (ADDCRP) and an already existing one (DDCRP), for clustering time-correlated data in the context of mixture models applied to document modelling. We introduced the ADDCRP as a method to model time-correlations that scales in a similar way as the CRP if the document number is increased. Implementation-wise, the ADDCRP has the same complexity as the CRP, whereas the DDCRP is more complex because the clusters have to be tracked through a list of links. We found that the ADDCRP also achieved better performance than the DDCRP on our data, according to the held-out likelihood measure. The performance is however dependent on the choice of the decay parameter. We have shown that for their main goal, the prediction of future data, all the distance dependent variants reach better performance than the original CRP. Hence they form an interesting subject of future research. For instance, a question that arises naturally is whether an algorithm can be found that can learn the decay parameter $\beta$ from the data. If some probabilistic framework could be found for this, algorithms as discussed here could become very powerful. Another question that relates to our newly proposed algorithm, is whether it can also be applied successfully to other domains where non-parametric clustering algorithms are used. Applications where the DDCRP can outperform the CRP are numerous (image compression, language modelling,...). Hence an interesting topic for future research would be to investigate whether the ADDCRP can match the performance of the DDCRP in those applications. If performance would turn out to be equivalent, the modellers choice will ultimately be determined by the trade-off between the convergence time and the runtime per iteration, both can vary heavily dependent on the application. Modelling time- and other correlations through a distance dependence is not yet explored very thoroughly in the context of non-parametric clustering algorithms, and as such these questions are very interesting to address in future research.

References


Modelling habitat preference, abundance and species richness of alien macrocrustaceans in surface waters in Flanders (Belgium) using decision trees

Pieter Boets
Koen Lock
Peter L.M. Goethals
Laboratory of Environmental Toxicology and Aquatic Ecology, J. Plateaustraat 22, 9000 Ghent, Belgium

Keywords: biological invasions, classification trees, habitat suitability modelling, integrated modelling, regression trees

The introduction and spread of alien invasive species is a worldwide phenomenon causing global ecological and economic damage. Among the invaders, alien macrocrustaceans are known to be very successful invertebrates that colonise new habitats rapidly. Data from different fresh and brackish waters gathered by the Flemish Environment Agency (VMM) were used to build data-driven models predicting habitat preference, abundance and species richness of alien macro-Crustacea present in surface waters in Flanders. In total, 882 samples of the year 2004 from different sampling locations scattered over surface waters in Flanders and comprising biological as well as physical-chemical and shipping data were used to build the models. Different machine-learning techniques such as regression and classification trees in combination with several optimisation methods (e.g. pruning) were used to construct the models.

The performance of the models was moderate, because a balance between performance, ecological relevance and complexity was strived for and because of the variability inherent to alien species. When using a three-fold cross validation, it was found that the variation between the folds was limited, which is an indication of the robustness and the good reliability of the constructed models. Based on a sensitivity analysis, the importance of conductivity, Kjeldahl nitrogen and shipping were stressed as well as graphically illustrated.

The ecological evaluation pointed out that conductivity and shipping in combination with chemical water quality were the major factors determining presence or absence of alien macrocrustaceans. Alien macrocrustaceans were predicted as present under brackish water conditions as well as in fresh waters with intensive ship traffic and low levels of organic pollution. Also the alien species richness was higher in rivers with intensive ship traffic and increased with increasing conductivity. Especially in brackish waters, alien macro-crustaceans reached high abundances. In freshwater, the abundance of alien species was generally lower. Brackish water conditions in combination with high levels of ship traffic seemed to be favoured by alien macrocrustaceans.

An integrated model that combined our habitat suitability model with a water quality model (PEGASE) was used to predict the future distribution of alien macrocrustaceans for the year 2015 and the year 2027. The predictions indicated that the prevalence and species richness of alien macrocrustaceans is likely to increase with improving chemical water quality, whereas their abundance will probably decrease slightly. From our analysis, it is clear that these models are a useful tool and that decision makers should focus on vulnerable areas such as brackish water areas and areas with intensive ship traffic in order to prevent the further introduction and spread of alien species.

Acknowledgments
We would like to thank the Flemish Environment Agency (VMM) for the opportunity to study their samples and nv De Scheepvaart and the River Information System for providing us with data regarding shipping in Flanders. We also would like to thank the VMM and Tom D’heygere in particular for providing the PEGASE water quality data.
Bag Dissimilarities for Multiple Instance Learning

Veronika Cheplygina
David M.J. Tax
Bob Duin
Wan-Jui Lee
Marco Loog

Pattern Recognition Lab, Delft University of Technology, The Netherlands

Keywords: multiple instance learning, dissimilarity space

Multiple-instance learning (MIL) (Dietterich et al., 1997) extends supervised learning in order to learn from objects described by a set (bag) of feature vectors (instances) \( B_i = \{ x_{ik} \mid k = 1, \ldots, n_i \} , x_{ik} \in \mathbb{R}^d \). During training, the bag labels (positive or negative) are available, but the labels of the individual instances are not. It is often assumed that a bag is positive if at least one instance is positive or inside a concept; a bag is negative if all instances are negative. This way, a single instance may be responsible for the bag label. Many MIL algorithms therefore focus on finding the “most positive” instances, modelling the concept and classifying bags according to whether they contain instances from this concept.

Many real-life problems do not fit into this strict formulation. For example, when classifying emails based on the presence of “spam-related” words such as “buy”, “offer” or “bonus”, we might only want to classify the emails as spam when a certain percentage of such words is present. In this case, several instances are responsible for the bag label. In such situations it might be more beneficial to compare bags at a more global level. By defining kernels or dissimilarities between bags, we can convert the MIL representation to a vector \( d(B_i, T) = [d(B_i, B_1), \ldots d(B_i, B_{|T|})] \) where \( T \) is the training set. Here, standard supervised learning methods can be used.

In our previous work (Tax et al., 2011) we have examined two main ways of defining MIL bag dissimilarities \( d(B_i, B_j) \). One possibility is to use the pairwise instance dissimilarities \( D = [d(x_{ik}, x_{jl})]_{N_i \times N_j} \). An example is the overall minimum instance distance \( d_{\text{minmin}}(B_i, B_j) = \min_{k,l} d(x_{ik}, x_{jl}) \). Many other formulations are possible, such as the Hausdorff distance \( \text{maxmin} \). Dissimilarities that compare the instance distributions directly are the other possibility.

We studied, for instance, the Mahalanobis distance \( \text{maha} \) and the Earth Mover’s Distance \( \text{emd} \).

We compared dissimilarities in combination with a nearest neighbor (NN) and Parzen (P) classifiers to several traditional MIL methods. Results on standard benchmark datasets are shown in Table 1.

<table>
<thead>
<tr>
<th>Data</th>
<th>Best dissimilarity</th>
<th>Best other</th>
</tr>
</thead>
<tbody>
<tr>
<td>musk1</td>
<td>94.7 (minmin+P)</td>
<td>92.9 (miSVM)</td>
</tr>
<tr>
<td>musk2</td>
<td>92.3 (minmin+P)</td>
<td>95.3 (MILES)</td>
</tr>
<tr>
<td>alt.ath</td>
<td>90.0 (emd+NN)</td>
<td>69.8 (miSVM)</td>
</tr>
<tr>
<td>rec.mot</td>
<td>89.2 (maxmin+NN)</td>
<td>76.4 (miSVM)</td>
</tr>
<tr>
<td>pol.mid</td>
<td>87.8 (emd+P)</td>
<td>79.8 (miSVM)</td>
</tr>
<tr>
<td>fox</td>
<td>75.0 (maha+NN)</td>
<td>69.8 (MILES)</td>
</tr>
<tr>
<td>tiger</td>
<td>78.6 (minmin+NN)</td>
<td>87.2 (MILES)</td>
</tr>
<tr>
<td>elephant</td>
<td>87.8 (minmin+NN)</td>
<td>91.1 (miSVM)</td>
</tr>
</tbody>
</table>

Table 1. AUC performances

The results show that there is no overall “best” dissimilarity; the choice should depend on the problem. The results are comparable to other MIL algorithms. The most notable exception concerns the Newsgroups datasets. The bag-of-words representation these rely on renders the single concept invalid. As in our spam example, a combination of words is necessary to define the class of the bag.

References


Can acidosis in dairy cows be diagnosed from milk fatty acids?

E. Colman
Laboratory for Animal Nutrition and Animal Product Quality, Ghent University, Proefhoevestraat 10, 9090, Melle, Belgium
ELLEN.COLMAN@UGENT.BE

W. Waegeman
Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure Links 653, 9000, Gent, Belgium
WILLEM.WAEGEMAN@UGENT.BE

B. De Baets
BERNARD.DEBAETS@UGENT.BE

V. Fievez
VEERLE.FIEVEZ@UGENT.BE

Keywords: subacute ruminal acidosis, classification techniques, milk fatty acids, rumen pH

Subacute ruminal acidosis (SARA) is a nutritional disorder occurring when dairy cattle are fed diets rich in quickly fermentable carbohydrates (QFCH), particularly when animals are not adapted to these feeds. This results in an increased production of volatile fatty acids, lowering ruminal pH to nonphysiological levels (Bramley et al., 2008). An excess of QFCH in the rumen causes a change in the rumen microbial flora (Nagaraja and Titgemeyer, 2007), which are the major source of odd- and branched-chain fatty acids (OBCFA) in milk fat (Vlaeminck et al., 2006). Hence, changes in the ruminal population are reflected in the milk fatty acid profile. As milk samples are a non-invasive alternative of pH measurements, research is ongoing to investigate their potential as a diagnostic tool for SARA.

A dataset of six different acidosis induction experiments is available (Colman et al., 2010, Colman et al., submitted, Khafipour et al., 2009a, Khafipour et al., 2009b, unpublished data). In each experiment, rumen pH was recorded every minute during 24h and milk samples were analyzed by gas chromatography in order to retrieve the milk fatty acid profile.

Colman et al. (2010, submitted) observed relationships between milk fatty acid and rumen pH changes. However, these relationships were not unequivocal across experiments and classification by discriminant analysis in acidotic vs. non-acidotic cases included both false positive and false negative classifications. Several machine learning techniques such as support vector classification and non-linear regression models are further explored for this purpose.

References


New procedure for estimating prediction certainty in decision trees

Eduardo P. Costa
Sicco Verwer
Hendrik Blockeel

Dept. of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, Leuven, Belgium
Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands
Leiden Inst. of Advanced Computer Science, Universiteit Leiden, Niels Bohrweg 1, Leiden, The Netherlands

Keywords: decision trees, prediction certainty

In classification, we often need classifiers that do not only have high accuracy, but can also tell how certain they are about a prediction. This is one reason for evaluating classifiers based on their area under the ROC curve (AUC), instead of accuracy: AUC penalizes incorrect predictions more strongly when they are made with higher certainty.

In decision trees (DTs), a prediction’s certainty is usually estimated using the class distribution in the leaf responsible for the prediction. We introduce an alternative method that yields better estimates. It is based on the following questions: (1) “If we add the test instance to the training set but with a different label than the correct one, will the DT learner find a tree that explains this instance according to the wrong label?”; (2) “How certain will the tree be about this prediction?”; (3) “How does this situation compare to the one where the instance is added to the training set with the correct label?”. Investigating these questions allows us to identify instances that might present some difficulties to be correctly classified, and to attribute some uncertainty to the prediction for these instances.

However, as the correct label of the test instance is not known during training, the method tries all possibilities for the target attribute. More specifically, to classify an instance $x$, the method builds $k$ trees, where $k$ is the number of possible labels for the target attribute. For each tree, $x$ is inserted in the training set with one of the possible labels. At the end, the final prediction and its certainty for $x$ are obtained by combining the prediction of all the trees. To combine the probability for each class we average the probability for that class for the $k$ trees. In this way, the final prediction will have a lower certainty when the trees give different predictions than when all trees give the same prediction, for example. Additionally, to avoid that small leaves have too much effect in the final prediction, we apply the Laplace smoothing to add some weight in the way we combine the class probabilities.

We evaluated our method on 55 randomly selected UCI datasets, using leave-one-out validation. First, we used 6 datasets for fine-tuning and validation - we tested different alternatives to combine the class probabilities of the trees, leading to the method configuration just described. We then evaluated the method on the other 49 datasets.

We implemented our method as an extension of the DT learner Clus\(^1\); we call it Clus-Mod. We compare Clus-Mod to the original Clus (Clus-Orig). To ensure that an improvement of Clus-Mod is not simply due to “ensemble effects” (since our prediction averages that of $k$ trees), we also compare to standard bagging (Clus-Ens) with the same number of trees, averaging class probabilities in the same way.

In a comparison with Clus-Orig, Clus-Mod obtained better AUC for 38 out of the 49 test datasets; 4 ties and 7 losses were observed. When compared to Clus-Ens, we observed 29 wins, 2 ties and 18 losses for Clus-Mod. Accuracy-wise, Clus-Mod does not improve over the others. We conclude that the new method does not yield more accurate predictions, but estimates its prediction certainty more reliably than the other methods.

Acknowledgments

This research was supported by projects G.0413.09 “Learning from data originating from evolution” and G.0682.11 “Declarative experimentation”, funded by the Research Foundation - Flanders (FWO-Vlaanderen).

\(^1\)http://dtai.cs.kuleuven.be/clus/
Exploring Item-User-Rating Datasets as Images

Iulian M. Covlescu
Zoltán Szlávik
VU University Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Keywords: data mining, visualisation, Fast Fourier transform, image processing

We explore the possibility of using image processing techniques, Fast Fourier Transform (FFT) in particular, to gain insights into a user-item rating type dataset often used in the context of recommender systems.

We look at a dataset as a set of signals describing user-item interaction. This set of signals can be displayed as an image: rows correspond to users, columns to items, while a pixel's intensity is a user's rating of a particular item. If we consider human behaviour a sort of a signal—a voice of the user—then it makes sense to look for a periodic pattern in the item rating behaviour of the user. We consider that the user is tuned in to some types of items, which is equivalent to the user emitting a signal on certain frequencies in the item rating space (note that the signal is not a signal in time here). A pattern in the image dataset is the result of having similar users' ratings. This image pattern is hard to observe with the naked eye because of the sparse data and overlapping patterns. However if patterns exist, they should be visible in the frequency domain of the image. Finding these frequencies was the rationale of choosing FFT as an image transformation.

In our experiments we used the MovieLens dataset, and represented it as a 1682x943 image where each rating (1-5) was pictured as a pixel and its value represented the intensity of the pixel.

By applying FFT on this image we obtained two sets of data: magnitude and phase, the real part and imaginary part of the image transformation. For the purpose of detecting pattern frequencies, the magnitude spectrum is more important because it holds the real part of the dataset image, corresponding to information about signal power.

If the initial image of the dataset presents patterns of ratings distribution, then in the spectrum image we can expect to find frequencies distributed on a line perpendicular to the direction of the pattern. The intensity of each pixel in the spectrum image represents the frequency coefficient in the Fast Fourier sum. This means that the more intense a pixel, the higher the value of the coefficient, thus the higher importance of the frequency in the pattern of the initial image. In the spectrum image we found that there were two clear directions (horizontal and vertical) where the frequency coefficients were higher (more intense pixels), which was expected since there was no reordering of the lines and columns, so the only directions that presented patterns were vertical (users) and horizontal (movies). Also, as expected, we observed that the lowest frequency, point (0,0) was the most intense. This point corresponds to the level of energy of the average pixel in the initial picture and because the dataset is sparse and we replaced unrated movies with intensity value 0, the initial image had most of its information encoded in the average value pixel.

We further experimented how new images can be created from the same dataset. Based on linear combinations of popularity and average ratings for both movies (columns) and users (lines) we reordered the IDs in the dataset, thus creating new dataset images for each combination. We observed that each image presented different directions of dispersion of data and different levels of intensity of frequencies in the FFT spectrum. This supports the idea that the way a picture is created is strongly connected to the information present in the spectrum domain.

Rebuilding an image from FFT needs both components, i.e. spectrum and phase data. Selecting only the most intense frequencies in the spectrum as a basis for rebuilding an image and considering the rest of frequencies as noise, it will lead to a smoothed initial image. Because the smoothed image has values of pixels that the initial image had not, this can be considered as a prediction based on the dominant frequencies of the initial image. We consider the evaluation of such prediction methods as future work.

To sum up, in our experiments we tested if an image processing technique could be used to gain insights in a user-item dataset patterns. We observed that the patterns in the images we created are shown in the spectrum domain and are strongly connected with how the initial image was constructed from the dataset. For future, we intend to use information from the spectrum domain to generate predictions based on the principal coefficients of the Fourier transform and the pattern information observed in the spectrum.
Cooked chilled foods, or REPFEDs (refrigerated and processed foods of extended durability), are a heterogeneous group of food products. One of the pathogens of concern is *Bacillus cereus*. To assess the risk related to *B. cereus* in these products, a Quantitative Microbial Risk Assessment (QMRA) has been developed. The first input required in QMRA, is the prevalence/level of hazard in the raw materials of which the product is composed. The probability distributions fitted per ingredient will enable to estimate the variability of the *B. cereus* contamination in a crude product composition (e.g. 5% starch, 1% herbs, etc.).

Microbiological analysis results (n=541) were collected from multiple REPFED producing companies. The data, which contained many censored values (below detection limit), was divided in 5 groups: dry herbs, spices and powders (n=223, 12.1% positive samples), starch components (n=64, 6.25%), meat, fish and dairy products (n=137, 2.9%), fruits and vegetables (n=89, 1.1%) and ambient stable products (n=28, 0%).

The hierarchical statistical model was set up as follows. It was assumed that (i) the *B. cereus* contamination (in log CFU/g) followed a normal distribution characterized by a mean (µ) and a standard deviation (σ), (ii) the mean of these distributions was product dependent (i.e. one mean for each product group), (iii) while the standard deviation was microorganism dependent (i.e. constant standard deviation whatever the ingredient). This hierarchical model, including censored data, was solved using a Bayesian Inference technique implemented in Winbugs. The Monte Carlo Markov Chain algorithm was run. The model predictions were validated using the actual data by comparing the percentage of samples exceeding the detection limit.

The model resulted in five normal distributions (µ, σ) for the contamination of *B. cereus* (log CFU/g), one for each group. (i) Herbs and spices (-2.7, 3.3), (ii) starch components (4.1, 3.3), (iii) meat, fish and dairy products (-5.2, 3.3), (iv) fruits and vegetables (6.4, 3.3) and (v) ambient stable products (-7.2, 3.3). Results were satisfactory even if the probability of exceeding the detection limit was slightly higher than the actual “positive” sample percentage.

The resulting distribution of *B. cereus* concentration in a composite food product at the beginning of production allow a product specific risk assessment. The Bayesian inference is a flexible and easy-to-implement technique to assess prevalence/level of hazard in QMRA.

Acknowledgments

This study was funded by the Belgian Federal Public Service of Health, Food Chain Safety and Environment (Contract RT 09/01 MICRORISK).
Applying Morphological Changes During the Evolution of Quadruped Robots Results in Robust Gaits

Jonas Degrave, Francis wyffels, Tim Waegeman, Pieter-Jan Kindermans, Benjamin Schrauwen
JONAS.DEGRAVE@UGENT.BE
Ghent University, ELIS Department, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium

Keywords: evolutionary algorithm, gait optimization, quadruped robot, morphology

In nature, land animals started off with stable morphologies. A good example is the tortoise: its legs are spread out and it has a low center of mass. This made it very stable and therefore suitable for learning robust gaits. Over the course of evolution, land animals slowly grew bigger and their legs grew longer. Consequently, they adjusted their gaits to deal robustly with the locomotion problem on this more complex morphology.

In this work, we verify whether these morphological changes play an important role in learning robust gaits. Using CMA-ES as an evolution strategy (Hansen et al., 2003), it is possible to slowly learn controllers and gaits for quadruped robots. We want to investigate whether the transfer of learned gaits through morphological changes significantly increases the speed and robustness of the resulting gaits, as was reported by Bongard (2011).

Methodology

We build a simulation model of a quadruped robot in Open Dynamics Engine. It has been developed in such a way that small parametric changes to the body can be made during the CMA-ES optimization. So earlier generations in the CMA-ES can start with robust morphologies while later generations are similar to the final robot morphology. These parametric changes include changing the weight of the robot, the leg length and position of the legs as shown in Figure 1. To verify whether this indeed leads to more robust gaits, we will compare gaits which emerged from both evolved and static morphologies.

Discussion

The focus of our comparison is the robustness of the optimized gaits against morphological changes in the robot structure, changes in the environment and errors in the actuators. We expect that gaits of robots with experience on various morphologies are more robust to unexpected changes than gaits learned based on a single morphology.

Acknowledgment

The research leading to the results presented here has received funding from the European Community’s Seventh Framework Programme (EU FP7) under grant agreement n.248311 Adaptive Modular Architecture for Rich Motor Skills (AMARSi).

References


Determination of trade-offs in ecosystem service delivery using Bayesian belief networks

Rob D’hondt
Dries Landuyt
Laboratory of Environmental Toxicology and Aquatic Ecology, Ghent University, Jozef Plateaustraat 22, 9000 Ghent, Belgium
Katrien Van Der Biest
Sander Jacobs
Department of Biology, University of Antwerp, Universiteitsplein 1, 2610 Wilrijk, Belgium

Keywords: ecosystem services, Bayesian belief networks, trade-off analysis

Ecosystem services (ES) are the benefits people derive from ecosystems. During the past decades, the term ES was introduced to the general public and has since led to broad-scale recognition of ecosystems and their value for human well-being. If possible, economic valuation of these services enhances compatibility with current economical-oriented policy development. Therefore, ES assessment, analysis and optimization will play a vital role in future nature conservation or restoration decision-making.

Currently, both quantitative and qualitative assessment of ES is a rapidly emerging topic in scientific research. This has led to a broad range of developed ES models varying from basic qualitative models to complex mechanistic models that enable ES quantification. Interesting model applications could enhance the evaluation of different management scenarios on ES provision and facilitate implementation of the results in decision-support. This outlook highlights the need for tools that can identify and visualize trade-offs or synergies between ES and are easily communicable to policy-makers.

We investigated the applicability of Bayesian belief networks (BBNs) in modelling ES provision and trade-offs. BBNs are graphical modelling tools that use a causal network of parent and child nodes as their basis. These nodes or variables are interrelated by conditional probabilities, expressing the likelihood for a child node to be in a certain state given the states of its parent nodes. Bayesian modelling has been widely applied in various disciplines and recently also in ecological modelling (Haines-Young, 2011). In our research, we qualitatively analyse three ES and combine them into an integrated BBN model, based on existing data and process equations. The services are interrelated through the model structure so that, by varying the states of input management nodes, trade-offs and synergies between the services are identified and can be visualized.

Our research focuses on the ‘Vennen’, a large valley bottom wetland landscape located in the upstream part of the Grote Nete basin. Over the last century the landscape has been modified strongly for the purpose of agricultural intensification and mechanization. However, since the early nineties attempts are being made to restore parts of the area into wetlands. Most of the land management plans have been directed either towards optimization of provisioning services (mainly agricultural and timber production) or towards increase of biodiversity. However, wetlands fulfill a non-negligible regulating role in storage of carbon. Hence, the ES described in the model are agricultural and timber production and carbon sequestration as soil organic carbon (SOC).

The network structure and knowledge rules are based on Bayesian learning with research data from the area. The carbon sequestration submodel is learned with an existing regression model based on historical data of SOC (Meersman et al., 2008). Input nodes include characteristics of the soil-water system and land use. Output nodes qualitatively describe provision of the three services with states ranging from very low to very high. Theoretic scores were assigned to the services’ states to enable mapping of service provision in the area and visualization of the relationships between services through plotting.

References


Comparison of predictive accuracy and habitat preference information retrieved from seven species distribution models

Shinji Fukuda
Institute of Tropical Agriculture, Kyushu University, 6-10-1 Hakozaki, Fukuoka 812-8581, Japan

Bernard De Baets
KERMIT, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Willem Waegeman
KERMIT, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Jan Verwaeren
KERMIT, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Ans Mouton
Institute for Nature and Forest (INBO), Kliniekstraat 25, 1070 Brussels, Belgium

Keywords: data mining, machine learning, data-driven model, spawning habitat, European grayling

The study on species-environment relationships provides a greater understanding of the structure of target ecosystems, and thus it is actively studied in ecology and biogeography with the aid of a variety of species distribution models. For conservation or restoration of the habitat of target species, it is also essential to understand at which habitat conditions the target species prefers.

In this study, we used seven data-driven models of artificial neural networks (ANN), classification and regression trees (CART), fuzzy habitat preference model (FHPM) optimized by a genetic algorithm, generalized additive model (GAM), generalized linear model (GLM), random forests (RF) and support vector machines (SVM), for assessing spawning habitat of European grayling (Thymallus thymallus L.) in the Aare river (Bern, Switzerland). All models except FHPM were implemented using R software (R development Core Team, 2011). Field observation data consisted of four input variables of water depth (m), flow velocity (m/s), percentage of small-sized gravel and percentage of medium-sized gravel, and an output of presence/absence of the spawning ground. A 10-fold cross validation was applied to evaluate the predictive accuracy of the models. The predictive accuracy was evaluated using multiple criteria such as mean squared error (MSE), area under the receiver operating characteristics curve (AUC), correctly classified instances (CCI), and kappa, because different performance measures evaluate the same model differently. Habitat preference information was retrieved from the models was used for comparison.

As a result, the best performance was found for RF, followed by SVM, CART, ANN, GAM, FHPM and GLM, which are slightly different according to the performance measure used. FHPM, GAM and GLM provided habitat preference curves for spawning, whilst ANN and RF evaluated variable importance of model input. The curves showed the nonlinear relationship between the habitat conditions and the occurrence of spawning habitat. It was observed from the curves produced by the FHPM that the fish prefers to spawn at the habitat conditions such as the velocity around 0.3 m/s, the depth of 1 m, smaller percentage of small-sized gravel and larger percentage of medium-sized gravel. The variable importance suggested water depth to be important for the fish. Further study is necessary to assess the capability of machine learning methods such as RF and SVM to produce habitat preference curves (or surfaces) for the target species, because the habitat preference curves provide useful information in planning and management for conservation or restoration of target species.

Acknowledgments
The spawning habitat data were developed by Schneider & Jorde Ecological Engineering and the Swiss Federal Institute for Environmental Science and Technology (EAWAG). SF is a recipient of JSPS Institutional Program for Young Researcher Overseas Visits (Kyushu University).

References
Multilabel classification for assessing the genetic disturbance by freshwater fish invasions in northern Kyushu, Japan

Shinji Fukuda  
Institute of Tropical Agriculture, Kyushu University, 6-10-1 Hakozaki, Fukuoka 812-8581, Japan  
SHINJI-FKD@AGR.KYUSHU-U.AC.JP

Norio Onikura  
Fishery Research Laboratory, Kyushu University, 2506 Tsuyazaki, Fukutsu, Fukuoka 811-3304, Japan  
ONIKURA@AGR.KYUSHU-U.AC.JP

Takahiko Mukai  
Faculty of Regional Studies, Gifu University, 1-1 Yanagido, Gifu 501-1193, Japan  
TMUKAI@GIFU-U.AC.JP

Jun Nakajima  
Fukuoka Institute of Health and Environmental Sciences, 39 Mukaizano, Dazaifu, Fukuoka 818-0135, Japan  
CYPRIN@KYUDAI.JP

Keywords: random forests, data-driven model, co-occurrence, topmouth gudgeon, Japanese crucian carp

Topmouth gudgeon (PP; Pseudorasbora parva) is widely known as a highly invasive freshwater fish species that has expanded from East Asia (native range) to central Asia, northern Africa and Europe (introduced range) (Gozlan et al., 2010). Considering its flexible habitat selection and high invasiveness, it is necessary to understand the invasion ecology and its impact on ecosystems at local and global scales.

In this study, we developed a multilable species distribution model using random forests (RF; Breiman, 2001) in order to evaluate the potential of genetic disturbance on the basis of the likelihood of co-occurrence of PP and Japanese crucian carp (CC; Carassius cuvieri) which, we believe, is an indicator fish for the genetic disturbance of PP in Japan. RF-based species distribution model and genetic disturbance model were trained using 3-fold cross validation with 10 replications (10×3 CV), in which the entire data were used. Species distribution data (1064 data points) consist of twelve landscape attributes and a 4-label species distribution: (a) both absent, (b) PP present, (c) CC present and (d) both present. The model output, the likelihood of each of four labels, was then used for estimating the potential of genetic disturbance. The genetic disturbance was evaluated based on the population genetic analyses of mtDNA sequences for the PP in the northern Kyushu. Totaled 318 individuals were collected from 46 sampling points and used for the analysis, from which genetic disturbance ratio was evaluated as the ratio of the number of individuals of Kyushu-type population to the total number of individuals in a sampling point.

As a result, the species distribution model perfectly classified the four distribution patterns of PP and CC. The variable importance measure of the RF model could explain the factors affecting the species distributions in the region. The genetic disturbance model estimated the genetic disturbance ratio observed, which resulted in high correlation values ($R^2 = 0.850 \pm 0.006$: mean ± standard deviation). This suggests that the patterns of co-occurrence of two fish species can be used for assessing the potential of genetic disturbance of PP in the target region.

Further study using larger ecological data together with predictive models is needed to clarify the relationship between the patterns of co-occurrence of various fish species and the genetic disturbance of target species, from which key species can be identified for conservation and improved control of target species in the region.

Acknowledgments

This work was supported in part by the Global Environment Research Fund (RF-0910) of the Ministry of the Environment, Japan. SF is a recipient of JSPS Institutional Program for Young Researcher Overseas Visits (Kyushu University).

References


Use of multivariate statistics and machine learning techniques for integrated ecological modelling and decision support in river management

Javier E. Holguin
Gert Everaert
Peter L.M. Goethals

1,2,3 Ghent University, Laboratory of Environmental Toxicology and Aquatic Ecology, J. Plateaustraat 22, B-9000 Ghent, Belgium

Keywords: Integrated ecological modelling, habitat suitability modelling, GLM, decision trees

Environmental managers are constantly driven by politics searching for an optimal balance between preserving habitats and economics. Thus, there is a need for the development of practical tools providing accurate ecological assessment of rivers and species conditions. This should allow preserving habitats and species, stop degradation and restore water quality. However, the assessment of investments in sanitation infrastructure of urban wastewater systems has traditionally been done considering the fulfilling of legal physical-chemical emission limits, but omitting receiving water’s ecological conditions (Devesa et al., 2009). In this context, ecological assessment models can show the limitations of the self-cleaning capacity of surface waters. The proposed research aims to introduce an integrated ecological modelling approach for decision support in river management. We look for the integration of hydro-morphological, physical-chemical, and biological components in sub-models. Ecological models base on multivariate statistics and machine learning techniques allow predicting the biological water quality in terms of the presence and abundance of macroinvertebrates and the river status according to ecological water quality indices. Holguin et al. (2010) previously have shown that there is high potential of integrating hydrodynamic and physical-chemical water quality models with ecological models based on macroinvertebrates to evaluate the impact of wastewater discharges.

To date, we have applied this integrated ecological approach in three different case studies with rivers of different size, altitude and geographical location. The first case is a lowland river in a tropical region (Colombia), the second one is a mountain river in a tropical region (Ecuador); and the third is a lowland river in a temperate zone (Croatia). We did the integration of different hydraulic and physical-chemical water quality models (i.e. QUAL2Kw, MIKE11 and WEST) with habitat suitability and ecological assessment models. For the ecological modelling, multivariate statistics (i.e. logistic and quasi-Poisson regression) and machine learning techniques (i.e. classification and regression trees) were applied in these case studies. Two different model selection techniques were applied in the logistic and quasi-Poisson regression. These techniques were: 1) step-wise based on AIC/QAIC with a fine tuning procedure using hypothesis testing and; 2) the Multi-model inference based on the information-theoretic approach. Additionally, to test the robustness of the models, different internal validation techniques such as three-fold or leave-one-out cross validation (or ‘jackknife’ procedure) were applied.

Acknowledgments

Javier E. Holguin is currently supported by a doctoral fellowship from the Special Research Fund of Ghent University (BOF) in Belgium.

References


The goal of this one page abstract is to present the following article (Joly et al., 2012).

High-dimensional supervised learning problems, e.g. in image exploitation and bioinformatics, are more frequent than ever. Tree-based ensemble methods, such as random forests (Breiman, 2001) and extremely randomized trees (Geurts et al., 2006), are effective variance reduction techniques offering in this context a good trade-off between accuracy, computational complexity, and interpretability.

The number of nodes of a tree ensemble grows as $nM$ ($n$ being the size of the learning sample and $M$ the number of trees in the ensemble). Empirical observations show that the variance of individual trees increases with the dimension $p$ of the original feature space used to represent the inputs of the learning problem. Hence, the number $M(p)$ of ensemble terms yielding near-optimal accuracy, which is proportional to this variance, also increases with $p$. The net result is that the space complexity of these tree-based ensemble methods will grow as $nM(p)$, which may jeopardize their practicality in large scale problems, or when memory is limited.

While pruning of single tree models is a standard approach, less work has been devoted to pruning ensembles of trees. To further investigate the feasibility of reducing the space complexity of tree-based ensemble models, we consider the following experiment: (i) build an ensemble of trees; (ii) apply to this ensemble a ‘compression step’ by reformulating the tree-ensemble based model as a linear model in terms of node indicator functions and by using an $L_1$-norm regularization approach - à la Lasso (Tibshirani, 1996) - to select a minimal subset of these indicator functions while maintaining predictive accuracy.

We propose an algorithmic framework and an empirical investigation of this idea, based on three complementary datasets, and we show that indeed it is possible to so compress significantly tree-based ensemble models, both in regression and in classification problems. We also observe that the compression rate and the accuracy of the compressed models further increase with the ensemble size $M$, even beyond the number $M(p)$ of terms required to ensure convergence of the variance reduction effect.

**Acknowledgments**

F. Schnitzler is supported by a F.R.I.A. scholarship. This work was funded by the Biomagnet IUAP network of the Belgian Science Policy Office and the Pascal2 network of excellence of the EC. The scientific responsibility is the authors’.

**References**


Rule-based Classification of Toxic Response of a Freshwater Fish to Contaminated River Water

Ik-Joon Kang  
Faculty of Agriculture, Kyushu University, 6-10-1 Hakozaki, Fukuoka 812-8581, Japan.  
KANGNEW@AGR.KYUSHU-U.AC.JP

Shinji Fukuda  
Institute of Tropical Agriculture, Kyushu University, 6-10-1 Hakozaki, Fukuoka 812-8581, Japan.  
SHINJI-FKD@AGR.KYUSHU-U.AC.JP

Junya Moroishi  
Seiko Electric Co., Ltd., Fukuoka, Japan.  
J-MOROI@SEIKO-DENKI.CO.JP

Trinh Quang Huy  
Hanoi University of Agriculture, Trauquy, Gialam, Hanoi, Vietnam  
TQHUY@HUA.EDU.VN

Keywords: fish swimming behavior, rule-based assessment, machine learning, water quality, monitoring

As it is well known, fish shows abnormal swimming behavior when they are exposed to contaminants in water. In previous studies we already reported that dramatic changes in swimming pattern of Japanese medaka (Oryzias latipes) exposed to several kinds of toxic chemicals and developed monitoring method for water quality using swimming behaviors of medaka (Kang et al., 2009a, b).

In this study, we investigated the applicability of a monitoring system using swimming behavior of fish to assess water quality of polluted river water. River water was collected in To Lich river located in Hanoi, Vietnam, which was reported to be polluted by heavy metals, organic compounds, and so on. Three conditions of the river water were prepared: 100% river water, 50% river water and 0% river water (control). Zebrafish (Danio rerio) was used as a test fish. The test fish was exposed to either of the three conditions and fish movement was tracked by three-dimensional (3D) biomonitoring under a static condition. Using the 3D tracking data, 10-min mean values of the three behavioral parameters, namely swimming speed, vertical position in the water and swimming angle, were calculated.

In this study, we applied a rule-based classification system to discriminate toxic response of the fish, and two machine leaning methods, classification and regression trees (CART) and random forests (RF) for learning the relationship between the toxic response and behavioural parameters. The rule-based system was formulated based on previously studies (Kang et al., 2009a, b), in which eight rules were constructed using a combination of the three behavioral parameters. The input to CART and RF was the mean proportion of three behavioral parameters and the model output was three classes of 0%, 50% or 100%. The model accuracy was evaluated using correctly classified instances (CCI).

As a result, the test fish showed different responses to the different conditions of river water. For instance, a constant response over test periods was observed between individuals in the control. The test fish stayed the upper layer of water in the case of 100% river water, of which all fish died within 90 minutes after the exposure. In the case of 50% river water, the behavioral response appeared slowly as shown in a decreased 10-min mean value of swimming speed, whereas vertical position and swimming angle differed at 30-40 minutes after the exposure, but recovered around 80 minutes after the exposure.

The model result showed that the rule-based system is useful to distinguish the fish behaviour under 0% and 100% river water. However, the system could not detect toxic response to 50% river water. The CCI of CART was 0.910, while that of RF was 0.958. The rules of CART may suggest the importance of vertical position in the water, which can be strongly related to hypoxia. The variable importance of RF indicated that the both swimming speed and vertical position in the water are useful for the classification of fish responses to the different conditions of water.

References


Towards Incorporation of Hierarchical Bayesian Models into Evolution Strategies for Quadruped Gait Generation

Pieter-Jan Kindermans, Francis wyffels, Ken Caluwaerts, Bert Guns, Benjamin Schrauwen
ELIS Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium

Keywords: Transfer Learning, Hierarchical Bayesian Modeling, Quadruped, Gait, CMA-ES

Learning how to control a compliant quadruped robot is a challenging task. Hand tuning and evolutionary strategies are the two most often used approaches for developing quadruped gaits. These approaches are time consuming and the end result is only usable in a specific environment. Furthermore the stability of the gait is often very sensitive to minute parameter changes. This limits the usability of such a robot. We want to circumvent these problems by using transfer learning. The goal in transfer learning is to improve learning by using knowledge from a relating task. This can speed up the learning process and improve the end result. Hierarchical Bayesian Modeling (Kemp et al., 2007) is a method which can be used for transfer learning which is able to learn from just a few examples. We investigated the similarity between gaits on flat terrain and slopes. We simulated the robot using an in-house built simulator, based on Open Dynamics Engine (ODE). The gaits were optimized using CMA-ES(Hansen, 2006). In the first experiment we optimized gaits for specific slopes. Inspection of the final population has shown that distributions over gait parameters are very similar for the different slopes. Thus indicating that transfer learning is applicable to this task. As a proof of concept we ran a second experiment where we have changed the starting conditions and/or the search boundaries based on the data obtained by optimization on a single slope. The results in Figure 1 show that the optimization process converges faster and that there is less variance on resulting fitness. In Figure 2 we see that the number of gaits which are stable has increased with the adapted search.

Future work will focus on combining ideas from Hierarchical Bayesian Models with Evolution Strategies.

Acknowledgments

The research leading to the results presented here has received funding from the European Community’s Seventh Framework Programme (EU FP7) under grant agreement n.248311 (AMARSi) and a PhD fellowship of the Research Foundation Flanders (FWO).

References


Spatial Role Labeling using Structured Support Vector Machines

Parisa Kordjamshidi
Marie-Francine Moens

Department of Computer Science, Katholieke Universiteit Leuven, Belgium

Extraction of spatial information from natural language is a challenging problem in applications such as robotics, geographical information systems and human-machine interaction. The task of spatial role labeling (SpRL) is a newly introduced task (Kordjamshidi et al., 2011) that formalizes the spatial concepts and relations in the language to be mapped to formal spatial calculus using machine learning. The first level of this task is to extract spatial roles called spatial indicators (ind), trajectories (tr) and landmarks (lm) and their link. Spatial indicators indicate the existence of spatial information in a sentence. Trajectory is an entity whose location is described and landmark is a reference object for describing the location of a trajectory. For example in the sentence: The book on AI in on the table behind you., the second on is a spatial indicator. book is a trajectory and its landmark is table. A spatial relation is a triplet of spatial roles. Hence, there are two spatial relations in this sentence: < onind, booktr, tablelm > and < behindind, tabletr, youlm >.

This abstract reports a work in progress for extraction of spatial relations using structured SVMs (Tsochantaridis et al., 2006). This framework provides the flexibility to learn from structured data considering the complex correlations in both input and output spaces.

SpRL is formulated as a supervised learning to learn a mapping \( f : \mathcal{X} \rightarrow \mathcal{Y} \) between the input space \( \mathcal{X} \) and discrete output space \( \mathcal{Y} \) given a set of examples, \( E = \{(X^{(i)}, Y^{(i)}) \in \mathcal{X} \times \mathcal{Y} : i = 1 \ldots N\} \). In the joint learning setting, we tend to learn an \( F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \) over input-output pairs. Then the prediction is performed by maximizing \( F \) over \( \mathcal{Y} \) for a given input \( \mathcal{X} \). Therefore the general form of \( f \) is: \( f(X; W) = \text{arg max}_{Y \in \mathcal{Y}} F(X, Y; W) \). \( F \) is assumed to be a linear function over a combination of input and output features \( \Psi(X, Y) \), i.e., \( F(X, Y, W) = \langle W, \Psi(X, Y) \rangle \).

The \( Xs \) in our model are the natural language sentences that are represented as a sequence of words: \( X = w_1, w_2, \ldots w_n \). The \( Ys \) are corresponding spatial triplets: \( < w_{ind}, w_{tr}, w_{lm} > \), represented as \( n \times n \times n \) matrices. Each element \( y_{ijk} \) in the matrix, is one if there is a spatial relation between words \( \langle w_i, w_j, w_k \rangle \).

To reduce the size of the output space a set of candidate words are selected according to the background knowledge about part of speech tags. For example the indicators can only be the prepositions in our model.

For a structured SVM model, we need to design three main components (Tsochantaridis et al., 2006). The first component is the joint feature mapping \( \Psi(X, Y) \). The input space features are defined based on the linguistically motivated features of the words in addition to the relations between adjacent and long distance words. The linguistic features contain, the word-form, part of speech tag, semantic role, dependency relation and subcategorization of each word (Kordjamshidi et al., 2011). These features are joined with the spatial roles that each word plays in the output space. For example whether a specific word-form is a spatial indicator will be one binary joint feature and whether it plays a trajectory role with respect to a specific indicator is another joint binary feature and so on. The second component is the inference algorithm. Since the number of possible \( Ys \) for each input \( X \) is very large in structured output prediction, we use a cutting plane algorithm (Tsochantaridis et al., 2006) in which only the most violated \( Y \) is considered in formulating the constraints for the max-margin optimization problem. However an efficient inference algorithm is required for maximization over \( Y \) for a given \( X \) and finding the most violated constraints. To this aim we use a greedy inference algorithm. The third component, is the loss function. We define a flexible F1-based loss function for this task and evaluate the prediction of the model for the spatial roles and relations.

References


A machine learning approach to predict Fusarium head blight

Sofie Landschoot, Kris Audenaert, Geert Haeasaert
University College Ghent, Valentin Vaerwyckweg 1, 9000 Gent, Belgium

Willem Waegeman, Bernard De Baets
Ghent University, Coupure links 643, 9000 Gent, Belgium

Keywords: Fusarium head blight, regression, web tool

Fusarium head blight is a fungal disease of wheat. Besides reduced quality and yield loss, most Fusarium species produce mycotoxins that can be hazardous to the health of humans and animals. One of the most common toxins is deoxynivalenol (DON). Given the severe consequences, there is an urgent need to know the level of contamination in advance. In this research, various machine learning techniques were used to predict the disease index (DI) and DON content of wheat. This work led to the development of a web tool to deliver field-specific predictions. Model development was based on data from wheat fields on 15 Belgian locations from 2002 till 2011, which results in 3,100 entries in the database.

To predict the continuous variables DON content and the DI a regression approach seemed in place. All constructed models were based on agronomic and weather variables. We compared linear and ridge regression, regression trees, boosting and support vector regression. Additionally, we showed that the commonly applied K-fold cross-validation (CV) strategy cannot be adopted for evaluating disease prediction models if year and location effects are present since this strategy suffers from a substantial optimistic bias. We presented the cross-year cross-location validation (CYCLV) strategy that enables the evaluation of the predictive performance for future years and new locations.

The linear regression model provided the best illustration of potential pitfalls in evaluation. For the K-fold CV, the performance of the techniques was comparable. When comparing the results for the CYCLV strategy, we saw that linear regression performed bad, by comparison with the more advanced techniques. This was due to over-fitting because linear regression contains no complexity control mechanisms (Landschoot et al., 2011a). Since in general the performance of the metric regression techniques was unsatisfactory to implement in practice, we shifted to ordinal regression. The thresholds used were based on the European legislation for mycotoxins. We compared proportional odds models and support vector ordinal regression (SVOR). The performance was evaluated with K-fold CV and CYCLV and different performance measures were compared. Ordinal regression models clearly outperformed the metric regression approaches. The performance of the proportional odds model and SVOR was comparable. However, the SVOR can fit non-linear models with a large number of features that are correlated, while in proportional odds models only linear models are considered. As a preprocessing step, we selected the features that were sufficiently uncorrelated. This might be the explanation for the comparable performance (Landschoot et al., 2011b).

As a result of the comparison, we embedded a four class proportional odds model in the web tool. With this tool, developed in collaboration with the Soil Service of Belgium, growers can obtain a field-specific prediction for DON content and the DI. The web tool provides a graphical representation of the predictions together with recommendations. The tool is developed for Belgium, so the predictions will only be accurate in regions similar as Belgium. However, the techniques and the framework used can be inspiring to generate tools for other regions and diseases.

References

Stage Detection in Runs of Evolutionary Algorithms

Wessel Luijben
Zoltán Szlávik
VU University Amsterdam, de Boelelaan 1081a, 1081HV Amsterdam, The Netherlands

Keywords: stage detection, evolutionary algorithms, time series clustering, feature extraction.

In order to tailor an evolutionary algorithm (EA) to a specific problem it is important to understand how it searches the fitness landscape and balances between bias and variance. By careful inspection of the fitness landscape it may be possible to do this analytically (i.e. without actually running the algorithm), but for real world problems this is impractical. In our work, we analyze the behavior of EAs (as quantified by runtime statistics) to provide similar insights. Examples of runtime statistics are maximum fitness within the population, average age of the population, etc.

In this research we apply several EAs and a few other numerical optimization algorithms to a well-known set of 10 benchmarking problems (Eiben & Bäck, 1997). For each algorithm (A) / benchmark (B) combination we obtain 100 runs. We then look for recurrent patterns within and between runs. Since runs are multivariate time series and stages can be of any length, finding stages in EA runs can be expressed as a multivariate different length time series clustering problem. Figure 1 shows a segmented EA run.

Figure 1. An example run of the algorithm Evolution Strategies (ES) applied to the sphere benchmark problem. The run is split into 5 segments q0 ... q4, and for each segment the features mean, standard deviation, trend and detrended variance are extracted. Segments are required to be at least 10 generations long. (This example shows a univariate run).

Our methodology is as follows: For a dataset of 100 runs originating from a single A/B combination, we split all runs into segments by finding regions of high volatility. We then extract features from all segments and apply k-means clustering. We compare clustering results with those originating from other A/B combinations. Finally, we interpret and visualize the differences. Figure 2 shows the result of clustering applied to a dataset of 100 segmented ES runs applied to the sphere benchmark problem.

Figure 2. The result of clustering 100 runs (only runs 1 to 9 are shown). Run 9 (highlighted) is identical to the run shown in Figure 1. To this end, the symbol ▲ can be interpreted as a steep drop in fitness; the symbol ■ as a modest drop in fitness and the symbol ○ as convergence. This small sample acts as a sanity check where drops in fitness always precede convergence. It is possible to observe the typical segment length, which is useful when studying multiple A/B combinations. We are aware that the sphere benchmark problem is an obvious case, but this example is intended as a proof of concept to work our way up to more complex examples.

In summary, our preliminary results indicate that, indeed, meaningful stages within runs can be found, and recurrent patterns across runs can be identified. However, to draw conclusions from a large number of EAs and problems, further research is required, which is currently being carried out. Ultimately, this research should lead to the ability to better design or control EAs in the future and help bridge the gap between theory and practice.

References


1 Stages are prototypical segments, e.g. “convergence” is a stage.

2 The choice for k-means clustering and feature extraction can be justified within the MDL framework (Rakthanmanon, 2011).
Simple Decentralized Algorithm for Coordination Games

Mihail Mihaylov
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Karl Tuyls
Maastricht University, Sint Servaasklooster 39, 6200 MD Maastricht, The Netherlands

Ann Nowé
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Keywords: decentralized coordination, learning algorithm, pure coordination games

Many biological or computer systems are comprised of intelligent, but highly constrained agents with common objectives that are beyond the capabilities of the individual. Often such multi-agent systems (MASs) are inherently decentralized and therefore agents need to coordinate their behavior in the absence of central control to achieve their design objectives. Wireless sensor networks (WSNs) are an example of a decentralized MAS where sensor nodes gather environmental data and collectively forward it towards the base station of the observer. The limited resources of such sensor nodes and the lack of global knowledge make the design of a WSN application challenging. The main question we are concerned with is the following: based only on local interactions and incomplete knowledge how can the designer of a decentralized system make agents achieve good collective performance imposing minimal system requirements and overhead?

In our work in progress we examine the decentralized coordination problem through repeated local interactions between agents arranged in different topologies. Pairs of neighboring agents are randomly assigned to play 2-by-2 pure coordination games, as done by Shoham and Tennenholtz (1997). The only information agents receive from each interaction is a payoff of 1 if they select the same action and 0 if their actions differ. Inspired by the challenges and constraints of WSNs, we would like to make memory-constrained agents learn to coordinate under limited (binary) feedback in as few interactions as possible.

We propose a simple decentralized algorithm that when adopted by individual agents leads them to global successful coordination. Our algorithm, called Win-Stay Lose-probabilistic-Shift (WSLpS) guides agents in selecting their actions in order to reach online a mutually beneficial outcome. Intuitively, if an agent receives payoff of 1 from an interaction (“win”), it means that its interaction partner has selected the same action. In that case it is reasonable that the agent will select the same action in the next time period (“stay”). A payoff of 0, on the other hand (“lose”), indicates that its neighbor has picked a different action. To avoid having both agents swap their actions upon conflict, each agent should change with a certain probability $\alpha \in (0,1)$ (“probabilistic shift”).

Our action selection algorithm is unique in that it is very light, requires no memory of previous interactions, given the current one, and quickly drives agents to full coordination. Moreover, coordination outcomes are absorbing states of the system and therefore agents will never change their actions once converged.

We carried out a Markov chain analysis of WSLpS and compared it to other algorithms applied in this domain in terms of convergence time. Initial experiments suggest that, besides being a much simpler algorithm and imposing virtually no system requirements, WSLpS outperforms Q-learning, highest cumulative reward rule (Shoham & Tennenholtz, 1997) and Win-Stay Lose-Randomize (Barrett & Zollman, 2009) in ring, scale-free and fully connected topologies.

References
Active noise control (ANC) is a way of actively reducing the amplitude of disturbing noise in audio. ANC reproduces the incoming noise signal with opposite phase so the residual signal approaches zero. While the idea is simple, implementation faces some hard problems. Problems occurring in real setups range from speaker and microphone distortion to reflection and deformation of acoustic waves. Also, noise prediction is often needed as the compensator isn’t infinitely fast. Typically, these problems are coped with by appropriate controllers. Many solutions have already been proposed. Recently, the use of genetic algorithms (GA) (Russo & Sicuranza, 2007) was considered, which we used as reference.

This work (Nyman et al., 2012) investigates the use of reservoir computing for the ANC problems of fig. 1. We showed that this problem can be solved by assessing the different subsystems (primary and secondary path in fig. 1), modelling them with reservoir computing and finally concatenating the right models to get a zero error signal. The reservoirs that model the subsystems are trained, using one shot learning. It should be noted that we did not consider predicting future samples which in fact is indispensable in the one-microphone setup. This task, however, highly depends on the signal type so we always assumed knowing the relevant part of the noise source beforehand.

The choice of the noise signal for training and evaluation was an important parameter because learning systems tend to overfit on strongly correlated inputs. Therefore, we trained our systems on uncorrelated white noise, which worked well\(^1\), even on other types of test signals. This proved a first important advantage over GA, that only performs well with strong correlations on the input samples.

A second advantage over GA, is that our approach needs less data and this data can be gathered in one experiment only. Lastly, the reservoir approach allows for a lot of further improvement. It will be interesting to investigate adaptive re-weighting of the reservoir readout, or to check if noise prediction and system modelling can be combined into one reservoir.

References


\(^1\)With 2 microphones and white noise, GA attained 6.70 dB reduction while our approach could offer 13.64 dB reduction. Applying the same methodology on the one-microphone setup, we attained 12.62 dB reduction.
Classifying Presence of Classes in UML Design using Software Metrics

Hafeez Osman  
Michel R.V. Chaudron  
Peter van der Putten  
Leiden University, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands

Keywords: software engineering, UML, reverse engineering

The Unified Modeling Language (UML) is a widely used modeling language in the software industry. UML is used to graphically represent the design of software systems. UML models which were created during the design phase are often poorly maintained during development. As a result the correspondence between design and implementation degrades as the implementation evolves considerably from its initial design (A. Nugroho, 2007). Indeed, it is well known that up-to-date design documentation is important in later stages of development and especially also in the maintenance phases. Also for legacy software, reliable designs are often no longer available. One popular technique in recovering the design is reverse engineering. Nevertheless, current reverse engineering techniques do not adequately solve this problem. In particular, too many details are presented. Hence, in order to achieve better design representations we need to learn which information from the implemented system to include and which information to leave out in reverse engineered designs.

![Source code classes & metrics](image1.png)  
![Learn design abstraction](image2.png)  
![UML design model](image3.png)

Figure 1: Design Abstraction in UML Design Model

This project specifically aims at reconstructing the class diagrams from source code in such a way that unnecessary detail that results from reverse engineering is eliminated (figure 1). This is work in progress and in this abstract we report on early results and open problems.

We have spent considerable effort to construct benchmark data sets for 10 pairs of UML design class diagrams and associated Java source code (obtained from open source projects). The number of classes (N) in the source code ranges from 59 to 903. The percentage of classes that is included in the UML design ranges between 3% to 47%.

<table>
<thead>
<tr>
<th>Project</th>
<th>k-NN (1)</th>
<th>Func. Logistic</th>
<th>Dec. Tree</th>
<th>Dec. Table</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>0.77</td>
<td>0.86</td>
<td>0.80</td>
<td>0.78</td>
<td>0.85</td>
</tr>
<tr>
<td>N=184</td>
<td>±0.07</td>
<td>±0.05</td>
<td>±0.06</td>
<td>±0.08</td>
<td>±0.06</td>
</tr>
<tr>
<td>Case 2</td>
<td>0.86</td>
<td>0.82</td>
<td>0.74</td>
<td>0.81</td>
<td>0.86</td>
</tr>
<tr>
<td>N=214</td>
<td>±0.02</td>
<td>±0.07</td>
<td>±0.07</td>
<td>±0.03</td>
<td>±0.03</td>
</tr>
<tr>
<td>Case 3</td>
<td>0.69</td>
<td>0.57</td>
<td>0.50</td>
<td>0.5</td>
<td>0.65</td>
</tr>
<tr>
<td>N=903</td>
<td>±0.04</td>
<td>±0.05</td>
<td>±0.01</td>
<td>0</td>
<td>±0.04</td>
</tr>
</tbody>
</table>

Table 1: Area under ROC Curve (AUC) from Classifiers

Our approach is to explore scoring models that, given a class in the source code of a system, provide a rank score for whether a class should be included in the UML design. A score provides flexibility for the end user to change the level of detail of the resulting reconstructed design. Initially we focus on using design characteristics of the source code through extracting software design metrics, which include: #in-dependencies, #out-dependencies, #operations, #attributes, #getters. In principle a large set of additional software metrics could be used (e.g. using SDMetrics).

Basic experiments show that mining this data is non trivial. For an initial study, we selected three case studies and experimented with five classifiers (Table 1). We randomly split the data set into 50% for training and 50% for test. To further improve reliability, we ran each experiment 10 times. The data can be highly unbalanced, so AUC is used as evaluation measure rather than accuracy.

Interesting future research directions include i) the use of semi-supervised learning or transfer learning to deal with the limited amount of labeled data for a given domain, ii) the integration of interactive user feedback, and the use of background or prior knowledge (existing designs). Data can be made available on request to parties wishing to collaborate on this problem.

References

Simultaneous Estimation of Model State Variables and Observation and Forecast Biases using a Two-Stage Hybrid Kalman Filter

Valentijn Pauwels
Laboratory of Hydrology and Water Management, Ghent University, Coupure links 653, B-9000 Ghent, Belgium

Gabriëlle J. M. De Lannoy
Laboratory of Hydrology and Water Management, Ghent University, Coupure links 653, B-9000 Ghent, Belgium

Harrie-Jan Hendricks Franssen
Forschungszentrum Jülich GmbH, Agrosphäre (IBG-3), Leo Brandtstrasse, 52425 Jülich, Germany

Harry Vereecken
Forschungszentrum Jülich GmbH, Agrosphäre (IBG-3), Leo Brandtstrasse, 52425 Jülich, Germany

The objective of this paper is to develop a methodology to estimate observation and forecast biases, in addition to state variables, using a two-stage hybrid Kalman Filter. The biases are estimated using the Discrete Kalman Filter, and the state variables using the Ensemble Kalman Filter. The error covariances of the forecast bias and the unbiased states are calculated as constant fractions of the biased forecast error covariance, and the observation bias error covariance is calculated as a multiplication of the observation predictions. In a series of synthetic experiments, it is shown that both static and dynamic observation and forecast biases can be successfully estimated. The results suggest that a better performance of data assimilation methods should be possible if both forecast and observation biases are taken into account.

Acknowledgments

The lead author wishes to express his gratitude to the Forschungszentrum Jülich for its hospitality during his stay there. He also wishes to thank Pramod Kumbhar for his help with the code development. Gabriëlle De Lannoy is a postdoctoral researcher funded by the Foundation for Scientific Research of the Flemish Community (FWO-Vlaanderen). We also want to thank the Department Operational Water Management of the Flemish Environmental Agency for the meteorological and discharge data.
Improved particle filter by using the ensemble Kalman filter as the importance density function in rainfall-runoff models

Douglas Plaza Guingla
Laboratory of Hydrology and Water Management, Ghent University, Ghent, Belgium

Valentijn Pauwels
Laboratory of Hydrology and Water Management, Ghent University, Ghent, Belgium

Keywords: particle filtering, ensemble Kalman filter, data assimilation, hydrologic modelling

The choice of the proposal is one of the critical design issues in particles filters. A proper performance of the PF is expected when the following key assumptions are valid: first, the point-mass approximation should represent the posterior distribution adequately, second, the particles drawn from the proposal distribution should approximate as accurate as possible the posterior of interest.

In case the first assumption is not completely valid, Markov Chain Monte Carlo move step has been proposed as a methodology to introduce variety to the particles improving the point-mass representation of the posterior.

For the second assumption, some approaches has been reported in the literature, e.g. the Auxiliary Particle Filter (APF), Regularized Particle Filter (RPF) and the Unscented Particle Filter (UPF) (Van Der Merwe et al., 2000) among others which are the derived from these techniques.

In the APF, approximated samples from the optimal importance density are obtained by using an auxiliary variable whilst in the RPF, samples are obtained from a continuous approximation of the posterior rather than from a discrete density improving the performance of the resampling step.

The UPF belongs to a set of techniques which approximate the optimal importance density by incorporating the current observation with the optimal Gaussian approximation of the state. In this context, the extended Kalman filter and the unscented Kalman filter are valid approximations to the optimal proposal. In the UPF, the optimal proposal is approximated as follows.

\[
q(x_t|x_{0:t-1}, y_{1:t})_{opt} = N(x_t^i; \bar{x}_t^i, \bar{P}_t^i) \quad (1)
\]

The samples \(\{x_t^i : i = 1, \ldots, N\}\) are drawn from a Gaussian distribution with the mean \(\bar{x}_t^i\) and covariance \(\bar{P}_t^i\) given by the unscented Kalman filter and computed for every particle.

In the same line of optimal proposals, the Ensemble Kalman Filter (EnKF) (Evensen, 1994) in combination with the Gaussian Particle Filter (GPF) (Kotecha & Djuric, 2003) is proposed as a nonlinear estimator and compared with the standard particle filter and the EnKF when discharge is assimilated into a conceptual Rainfall-Runoff model.

Acknowledgments

The work in this paper has been funded by the Belgian Science Policy for the HYDRASENS project in the frame of the STEREO II programme.

References


Inferring Gene Regulatory Network Topologies using Ensembles of Feature Selection Techniques

Joeri Ruysseink
Tom Dhaene
Yvan Saeys

Department of Information Technology, Ghent Uni.-IBBT, Gaston Crommenlaan 8, B201, 9050 Ghent, Belgium
Bioinformatics and Systems Biology Group, Ghent Uni.-VIB, Technologiepark 927, 9052 Ghent, Belgium

Keywords: feature selection, gene regulatory network, network inference, ensemble methods

In the field of computational systems biology, an important open issue is how the network topology of a gene regulatory network (GRN) can be inferred using high throughput genomic data. Unraveling an organism’s GRN plays a crucial part in gaining insight of the normal cell physiology and pathology.

Research on this subject has recently been stimulated and evaluated by two influential challenges in the field organized by the DREAM (Dialogue for Reverse Engineering Assessments And Methods) consortium. In both the DREAM4 In Silico Network Challenge (2009) and the DREAM5 Network Inference Challenge (2010) participants had to develop methods to reverse engineer GRN from microarray gene expression datasets.

The overall top performer in both challenges was the GENIE3 algorithm (Huynh-Thu et al., 2010). This algorithm solves the network inference problem for p genes by first decomposing it into p different regression problems. In each of these regression problems, a different gene expression pattern is considered as the target and all other gene’s expression patterns are considered to be possible predictors. Next, using tree-based ensemble methods such as Random Forests, an importance measure for each predictor is calculated and a high feature importance is used as an indication that a link is present between the predictor and the target gene in the GRN. The GENIE3 approach to solving the network inference is not only very successful because it achieves more accurate results than its competitors but also because it is computationally attractive as each of the independent regression sub-problems can easily be solved in parallel.

In our ongoing work we are looking at extensions and modifications of the GENIE3 approach to further improve the accuracy of the network inference process.

One of the ideas we are currently pursuing is using an ensemble of GENIE3-variants to further increase the accuracy. This is inspired by the fact that the last DREAM challenge clearly indicates that an ensemble of inference techniques achieves better results than a single technique (Marbach et al., 2012). Yet little is known about how these ensembles can be formed to achieve the best results. These GENIE3-variants are created by replacing the tree-based ensemble feature selection stage by different feature selection techniques.

To gain further insight into the performance and possible complementary properties of these GENIE3-variants an extensive comparative study is being conducted where the variable importance scores are derived using methods such as RFES-SVM, Elastic Net, Symbolic Regression, Relief-based algorithms and others. As not all of these techniques readily provide an obvious global feature score, several ways to achieve a global rank are examined.

The first results of this study show that some feature selection techniques have the potential to achieve similar or slightly better performance than the original GENIE3 algorithm. In particular, Symbolic Regression variants achieve similar performance on several examples. A first simple ensemble method created by aggregating the outputs of the original GENIE3 algorithm and the Symbolic Regression variant already showed an increase in prediction accuracy.

References


Feature selection in high-dimensional data sets is an open problem. We compare regression random forests (RF) (Breiman, 2001) and symbolic regression (SR) (Smits & Kotanchek, 2004), for obtaining variable importances from high dimensional datasets.

We study ‘conventional’ feature selection with both methods, compare the results, and identify the conceptual differences in generated feature importances. We introduce several general desired properties for quantifying contributions of individual variables to the response in an input-output relationship, whereby techniques can be compared:

- **Interpretability**— Importances should reflect the importances of the true input variables, without transformation.
- **Strictness**— Importances should only be allocated to variables relevant to describing the response.
- **Conservativeness**— Importances should be allocated to all potentially relevant variables.
- **Reproducibility**— Importances can only be considered correct if they are reproducible.
- **Universality**— Importances should be mutually comparable, and should be problem independent.

We apply both RF and SR to artificial datasets which exhibit common undesired data properties, such as imbalanced data and correlated variables, as to estimate the impact on both techniques. From these experiments we conclude that results from SR are more robust in presence of data defects, given that models of sufficient quality can be found. In general, meeting this condition is computationally intensive for large high dimensional datasets. For datasets where SR would be too time consuming, RF can relatively quickly estimate importances. However, we observe that RF models discriminate against variables with small contributions to the response. When correlated variables are present, importances obtained with RF yield unfavorable results. In addition, the RF importance technique handles data imbalance poorly.

For two real life case studies, we conclude that SR is capable of delivering more interpretable results than RF.

Since the experiments show that importances from SR and RF differ semantically, we advise against building models with SR using a variable subset selected with RF. Selecting features and estimating variable importances is a complex problem, and a thorough understanding of both modeling and importance scoring techniques is required to assess its applicability and interpret its results.

### References


Relational Learning and Ranking Algorithms for Biological Applications

Michiel Stock, Willem Waegeman, Bernard De Baets
KERMIT, Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Keywords: learning relations, ranking, kernel methods, bioinformatics, microbiology

In many domains one wants to infer properties of pairs of objects. This is particularly true for biological problems which will be our main sphere of interest. We present a broad framework that is suited for dealing with these kinds of problems. Furthermore we investigate how to incorporate some ideas from information retrieval to produce results that are more directly useful for biologists.

Our methods are based on generating a joint feature representation of pairs of objects by using the Kronecker product pairwise kernel (KPPK). Regularized least squares can then be used to make predictions on pairs of objects. It can be shown that the KPPK can learn any arbitrary relation (Waegeman et al., 2012), though it is not guaranteed to be the most optimal kernel. By using kernels we can deal with complex data structures such as sequences, graphs and trees which are frequently encountered in bioinformatics.

When the ranking error is optimized, the above framework can be used for conditional ranking. Here a set of objects are ranked conditioned on another object, the query. This means we are now treating the problem as a kind of information retrieval setting where we want to find the most relevant objects. We consider several performance measures such as the ranking error, mean average precision, ROC and CROC curves to find the most meaningful evaluation criterium for a particular problem.

We test our ideas on some real-world datasets to prove their relevance. As a first application we consider the problem of ranking a database of proteins according to their catalytic similarity to a query protein. We used five state-of-the-art similarity measures as features for the catalytic site of enzymes. We have obtained very promising results for this problem, strongly outperforming the baseline predictor. We showed that our model can give a significant boost compared to using an inferior but less computational demanding similarity measure.

In our second similar, but more advanced, application we also use a protein query, not against a database of proteins but for finding ligands that can bind the protein with high affinity. When using relevant features for these two types of molecules, this framework could serve as a post-processing method for docking algorithms, greatly aiding drug design. From a technical standpoint, this problem differs from the previous one as this is a dyadic setting since we consider two different types of objects.

A third application originates from the field of microbial ecology, where it is known that bacteria form complex ecological networks, exchanging metabolites, nutrients and information. More specifically, the interaction between methanotrophic bacteria and heterotrophic bacteria forms a key interest for biologists. The question we are interested in is how different kinds of each group influence each others growth. In wet lab experiments different combinations of both groups are incubated and their mutual growth density is measured. We want to construct a model that ranks the bacteria of one group for their predicted cooperation with our reference bacteria from the other functional group. The features that can be used for this model would be a metabolic fingerprint of the organism. This model could give environmental technologists a powerful tool for synthetic ecology.

References
A Multiclass Semi-Supervised Boosting Algorithm

Jafar Tanha  
Maarten van Someren  
Hamideh Afsarmanesh  
Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands  

Keywords: Semi-Supervised Learning, Boosting, Ensemble Learning

Most of the Semi-Supervised Learning (SSL) algorithms were designed for binary classification problems. However, many practical domains, for example recognition of speech, objects, and characters, involve more than two classes. A Multiclass classification problem can be decomposed into a number of independent binary classification problems by utilizing methods like one-versus-the-rest, one-versus-one or error-correcting output coding. However, these schemes have their problems. One-versus-all results in unbalanced distributions. Since each classifier is trained independently, the weights of their outputs may be on different scales, so that combining them is non-trivial. There is thus a need for direct multiclass algorithms for SSL.

In this paper we propose a new algorithm for Multiclass Semi-Supervised learning that follows the boosting approach and is a direct generalization of the binary SemiBoost algorithm (Mallapragada et al., 2009), which uses both the similarity between the points and the classifier prediction to sample and assign “pseudo-labels” to the unlabeled examples, to the multiclass setting, named as Multiclass SemiBoost. The key advantage of Multiclass SemiBoost is to exploits both the manifold and the cluster assumption to train the classifiers using boosting. We derive the algorithm from an objective function that combines empirical loss on the labeled data and inconsistency of the labels over all data.

Multiclass SemiBoost uses an exponential loss function on the inconsistency between labeled and unlabeled data as well as between unlabeled data. Based on this loss function we derive a criterion to sample and assign “pseudo-labels” to the unlabeled data and estimate confidence for them. At each iteration of the training process in Multiclass SemiBoost, a set of high-confidence unlabeled data along with labeled data is used to train a new classifier. The final hypothesis is formed by linear combination of the generated classifiers. The outline of the Multiclass SemiBoost algorithm is given in 1. Two ideas of the algorithm are to maximize the consistency between data and the margin. For that we design an optimization problem with the cost of both the inconsistency and the margin. Our assumptions are: (1) the labeled and unlabeled examples with high similarity must share the same label and (2) the unlabeled examples with high similarity must have the same label.

Our proposed method is similar to SemiBoost. It employs the advantages of the graph-based methods to minimize the inconsistency between data as well as the ensemble approach to minimize the margin cost to solve directly the Multiclass classification problem.

Algorithm 1 An outline of the Multiclass SemiBoost algorithm

- Initialize: L,U, S,H(x)  
- L: Labeled data; U: Unlabeled data  
- S: Similarity Matrix; H(x): Ensemble of Classifiers  
- At each iteration i:  
  - Assign “pseudo-labels” to the unlabeled examples based on the pairwise similarity and classifier prediction  
  - Sample the high-confidence examples for a component classifier  
  - Build a new component classifier based on both newly-labeled and original labeled examples  
  - Update ensemble H  
- Generate final hypothesis

References

Predicting Long Term Behavior of Genetic Regulatory Networks with Answer Set Programming

Gustavo Torres
Department of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281 (S9), 9000 Ghent, Belgium
GUSTAVO.TORRES@UGENT.BE

Kathleen Marchal
Department of Plant System Biology, VIB, Technologiepark 927, 9052 Ghent, Belgium
Kamar@psb.UGENT.BE

Yves Van de Peer
Department of Plant System Biology, VIB, Technologiepark 927, 9052 Ghent, Belgium
YVES.VANDEPEER@PSB.VIB-UGENT.BE

Martine De Cock
Department of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281 (S9), 9000 Ghent, Belgium
MARTINE.DECOCK@UGENT.BE

Keywords: ASP, GRNs, logic programming, Boolean Networks

Genetic Regulatory Networks (GRNs) hold the key for understanding the complex behavior of living organisms. This is the reason why it is very important to have flexible tools that can simulate and handle accurately different situations in GRNs. One particularly interesting aspect of study when modeling and simulating GRNs is their steady state behavior. In GRNs steady states are points where the GRN stabilizes i.e. states where the GRN falls into an oscillatory behavior. In biology this can refer to stable cell-types, e.g. skin cells, liver cells, etc. In computational models of GRNs we are interested in the sets of steady states that represent different system oscillations. Such sets of states are called attractors. The focus of this contribution is on finding all attractors of a given GRN.

Our approach for finding all attractors in GRNs, is based on a logical formalism called Answer Set Programming (ASP) (Lifschitz, 2008). ASP is a form of non-monotonic reasoning based on the stable model semantics for logic programming. Intuitively, in answer set programming one writes a set of rules (the program) such that certain minimal models (the answer sets) of this program correspond to solutions of the problem of interest. Solving an ASP program comes down to finding its answer sets. Specifically in answer set programming one writes a set of rules (the program) such that certain minimal models (the answer sets) of this program correspond to solutions of the problem of interest. Solving an ASP program comes down to finding its answer sets. Specifically in answer set programming one writes a set of rules (the program) such that certain minimal models (the answer sets) of this program correspond to solutions of the problem of interest. Solving an ASP program comes down to finding its answer sets.

A major advantage of using a declarative paradigm such as ASP for finding attractors in GRNs is that it lets us focus our efforts on describing the problem rather than on solving it, while providing a solid foundation on which further extensions can be built onto. Indeed complex behavior of GRNs can be expressed and adapted by expressing the desired system properties in a way that is both succinct and intuitively easy to understand for biologists.

We validated our method against the very well known networks of Yeast (Li et al, 2004) and Fission Yeast (Davidich et al, 2008) for both synchronous and asynchronous dynamics and, to showcase the scalability of our method, against larger networks used to explore the dynamics of Th1 and Th2 cell regulation (Pedicini et al, 2010).

Acknowledgments

This work was supported by the Ghent University Multidisciplinary Research Partnership “BioMaGNet” and the Interuniversity Attraction Poles Programme (IUAP P6/25), initiated by the Belgian State, Science Policy Office (BioMaGNet).

References


Identifying relevant remote sensing products for the spatial modelling of a bluetongue vector using random forests

Jasper Van doninck, Niko E. C. Verhoest
Ghent University, Laboratory of Hydrology and Water Management, Coupure links 653, 9000 Gent, Belgium

Jan Peters, Bernard De Baets
Ghent University, Department of Mathematical Modelling, Statistics and Bioinformatics, Coupure links 653, 9000 Gent, Belgium

Els Ducheyne
Avia-GIS, Risschotlei 33, 2980 Zoersel, Belgium

Keywords: Bluetongue, Culicoides, remote sensing, variable importance, epidemiology

Bluetongue is a viral disease affecting cattle and small ruminants, mainly sheep. The bluetongue virus is transmitted between hosts by some species of *Culicoides* midges. In the Mediterranean basin, bluetongue was historically confined to northern Africa, with sporadic outbreaks in southern Europe. Since 1998, however, regular outbreaks of bluetongue were observed in the Iberian Peninsula, Corsica, Italy, the Balkan, Greece, Turkey and Israel. These invasions are linked to the northward spread of *C. imicola*, the main vector of the bluetongue virus in Africa and Mediterranean Europe. An accurate modelling of the distribution of *C. imicola* is essential for the prediction of zones at risk of bluetongue outbreaks and the implementation of mitigating measures or vaccination campaigns.

The environmental factors constraining the distribution of *C. imicola* are as yet not fully understood. On a continental scale, air temperature is considered to be the major limiting factor of *C. imicola* distribution. On finer scales, however, *C. imicola* presences and absences have been observed in locations with similar temperature regimes. Additional environmental factors, including landcover, topography and soil properties have been assumed to further restrict *C. imicola* distribution (Conte et al., 2007).

This study aims at identifying the remote sensing products, used as proxies for different environmental factors, that are the most relevant in the spatial modelling of *C. imicola*. The research is conducted on a 3-year (2004-2006) dataset of *Culicoides* spp. absence/presence data from the Spanish Bluetongue National Surveillance Programme. Spatial modelling is performed by a supervised learning model based on Random Forests. In a previous study (Peters et al., 2011), Random Forests was identified as a suitable technique for *C. imicola* distribution modelling. Random Forests furthermore allow an assessment of the importance of the different remote sensing variables, and their associated environmental variables. The developed models are applied for a pixel-based prediction over the Iberian peninsula using the selected remote sensing products.

Acknowledgments

This research was funded by the Belgian Federal Science Policy as part of the EPIDEMOIST and HYDRASENS projects and by the Research Foundation Flanders. The Spanish Ministerio de Agricultura, Pesca y Alimentación provided data from the Spanish Bluetongue National Surveillance Programme.

References


Relational Learning for Football-Related Predictions

Jan Van Haaren  
Guy Van den Broeck  
KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Keywords: statistical relational learning, application

Association football is becoming increasingly competitive and the financial stakes involved are causing football clubs and football leagues to become more professional. Over the past 25 years, club budgets have grown enormously due to ticket sale revenues, broadcasting revenues, merchandising, and prize money. Recently, player tracking systems were introduced and are producing overwhelming amounts of data which are being used by experts to analyze matches.

State-of-the-art approaches for predicting football match results fail to leverage the full range of rich data that is currently available. These approaches are mostly extensions of well-known statistical methods that learn models with limited expressivity. Two important reasons lead to these limitations. First, until recently match statistics were usually not publicly available. Second, it is not obvious how to derive meaningful statistics from football matches.

We propose using machine learning and, more specifically, relational learning to address the shortcomings of current prediction techniques (Van Haaren & Van den Broeck, 2011). Relational learning is particularly well suited and overcomes the two most important challenges. First, it considers a variety of aspects that influence a match result. Second, it reasons about time-dependent and positional information.

A relational model offers a lot of flexibility to represent the data since its parameter set is not fixed but varies according to the events that happen during a football match. Consequently, a relational model is able to stress rare, but important, events such as a red card. Furthermore, it can represent complexly structured data such as team lineups using relations among objects (e.g., football players). All of this is much harder or impossible in a propositional representation.

Due to their expressivity, relational models can tackle many interesting learning tasks such as regression and classification. However, more complex learning tasks such as collective regression (e.g., jointly predicting player statistics), collective classification (e.g., predicting a team’s starting lineup) and link prediction (e.g., predicting who passes the ball to whom) take full advantage of a relational model’s capabilities to handle rich structured data.

Experiments with kLog (Frasconi et al., 2011) show that our relational approach yields competitive results when compared to propositional models. We used match statistics from the 2010-2011 English Premier League season to predict match results and goal differences (see Figure 1), and classify matches according to their outcome. The experimental results for relational and propositional approaches are very similar, but we expect relational learners to outperform their propositional counterparts once we manage to exploit the relational structure of the data better.

![Figure 1](image.png)  
 Figure 1. Our relational approach yields mean absolute error values for goal difference predictions that are similar to these of a propositional approach. Lower is better.

Acknowledgments

Guy Van den Broeck is supported by the Research Foundation-Flanders (FWO-Vlaanderen).

References


A Statistical Relational Learning Approach to Identify Sections in Scientific Abstracts Using Sentence and Document Structure

Mathias Verbeke, Luc De Raedt
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Roser Morante
Department of Linguistics, Universiteit Antwerpen, Lange Winkelstraat 40-42, 2000 Antwerpen, Belgium

Paolo Frasconi
Departamento di Sistemi e Informatica, Università degli Studi di Firenze, Via di Santa Marta 3, 50139 Firenze, Italy

Keywords: (biomedical) natural language processing, kernel methods, statistical relational learning

Evidence-based medicine (EBM) or evidence-based practice (EBP) is an approach whereby clinical decisions are supported by the best available evidence gained from scientific research. This is done by systematically reviewing, appraising and using clinical research findings to improve the patient care, which requires efficient access to such evidence. Therefore abstracts in EBM can be labeled using a set of predefined medical categories, the PICO criteria. This paper presents an approach to automatically annotate sentences in medical abstracts with these labels. Since each sentence can have multiple labels, the task is a multiclass multilabel classification problem at the sentence level.

As indicated by (Kim et al., 2011), both the structure of the words in the sentence and that of the sentences in the document are important features for this task. Furthermore, the sequential information can leverage the dependencies between different sentences in the text. Therefore we propose an approach using kLog (Frasconi et al., 2011) to tackle this problem. kLog is a new language for logical and relational learning with kernels, that is embedded in Prolog, and builds upon and links together concepts from database theory, logic programming and learning from interpretations. It is able to transform relational into graph-based representations and apply kernel methods on these.

The choice for kLog was motivated by the results of (Verbeke et al., 2011), where we showed that a statistical relational learning approach using kLog is able to process the contextual aspects of language. The current task adds two levels of complexity. First, next to the relations between the words in a sentence, now also the relations between the sentences in the documents, i.e. biomedical scientific abstracts, become important. To cope with this, we use the (high-dimensional) features that are generated by kLog as input for a structural support vector machine, in order to handle the sequence tagging. Second, since each sentence can have multiple labels, we use a stacking approach in which the the classifiers for the individual labels are combined in a pipeline, such that predictions made in the previous step in the process can be taken into account as features in the next, due to which we can simulate collective classification of the sentence labels.

We are currently investigating which features are most influential using feature selection techniques such as group lasso. In this way we want to select features simultaneously according to the groupings induced by the graph structure. The goal is to gain more insight into the importance of the relational features generated by the graph kernel, which could be of general interest for computational linguistics.

Acknowledgments
This research is funded by the Research Foundation Flanders (FWO-project G.0478.10 - Statistical Relational Learning of Natural Language).

References


Incorporating Prior Knowledge in Multiple Output Regression with Kernel-Based Vector Functions

Jan Verwaeren
Willem Waegeman
KERMIT, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Tapio Pahikkala
Antti Airola
University of Turku, Department of Information Technology, Turku Centre for Computer Science (TUCS) Finland

Bernard De Baets
KERMIT, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Keywords: vector valued functions, multi task learning, multiple output regression, prior knowledge

In single output learning, prior knowledge providing a rough relation between the inputs and the output can be incorporated in a learning procedure to reduce the hypothesis space and improve the predictive performance of the learned model. For kernel-based approaches, prior knowledge is often incorporated by adding constraints to the mathematical program that is used for learning. Alternatively, prior knowledge can be included by using a prior knowledge-based kernel. As the latter approach translates the problem of prior knowledge incorporation into one of designing an appropriate kernel function, it provides a flexible way of incorporating different forms of prior knowledge. As multiple output learning extends single output learning, it should be possible to incorporate such prior knowledge as well. Moreover, the multiple output nature of the learning problem allows additional types of prior knowledge to be included. More precisely, we distinguish the following three types of prior knowledge that can arise in multiple output regression:

Firstly, we distinguish output-specific input-output relations as prior knowledge of dependencies between (a subset of) the inputs and one (or several) outputs. A typical example is found in image reconstruction, when an image is to be reconstructed from several blurred versions of that image, a particular pixel in the target image (one output) is probably most influenced by pixels at the same location. As such, each target pixel has its own subset of input pixels to which it is most likely related.

Secondly, we distinguish input-input relations as a raw description of dependencies between inputs. This knowledge is not output specific. Such dependencies can for instance be used to reduce the dimensionality of the input space or derive an application-specific regularizer.

Thirdly, we distinguish output-output relations that describe knowledge about dependencies between outputs. A typical example here is clustering between outputs. When raw information about such a clustering is given, it can be incorporated in the learning algorithm.

We will see that these three types of prior knowledge can be incorporated when learning vector-valued functions in the framework introduced in (Micchelli & Pontil, 2005), by using a specific multiple output kernel. Moreover, we show that conjugate gradient methods can be used to solve the resulting mathematical program in an efficient manner.

References
Proceedings of the 21st
Belgian–Dutch Conference
on Machine Learning

May 24–25, 2012
Ghent University, Belgium

Organised by:
www.kermit.ugent.be

Academic Partners:

Editors:
Bernard De Baets
Bernard Manderick
Michael Rademaker
Willem Waegeman