Benelearn 08

The Annual Belgian-Dutch Machine Learning Conference

19-20 May 2008

Spa, Belgium

Editors
Louis Wehenkel
Pierre Geurts
Raphaël Marée

Benelearn is the annual machine learning conference of Belgium and The Netherlands. It serves as a forum for researchers to exchange ideas, present recent work, and foster collaboration in the broad field of Machine Learning and its applications. Benelearn 2008 is organised by the Systems and Modeling and Bioinformatics and Modeling research units of the Department of Electrical Engineering and Computer Science and GIGA-Research of the University of Liège. The conference takes place in the Solcress seminar center, at walking distance from the center of the city of Spa located in the Belgian Ardennes.

Conference Chair

Louis Wehenkel, Pierre Geurts, Raphaël Marée
Université of Liège, Belgium

Organizing Support

Michèle Delville, Céline Dizier
Association des Ingénieurs de Montefiore (A.I.M.)

Programme Committee

H. Blockeel, K.U. Leuven
G. Bontempi, U.L. Bruxelles
W. Daelemans, Universiteit Antwerpen
P. Dupont, U.C. Louvain
D. Ernst, Université de Liège
A. Feelders, Universiteit Utrecht
P. Geurts, Université de Liège
B. Goethals, Univ. Antwerpen
T. Heskes, Radboud Univ. Nijmegen
B. Kappen, Radboud Univ. Nijmegen
J. Kok, Universiteit Leiden
B. Manderick, Vrije Universiteit Brussel
R. Marée, Université de Liège
B. de Moor, K.U. Leuven
M. van Otterlo, Universiteit Twente
J. Piater, Université de Liège
L. de Raedt, K.U. Leuven
J. Ramon, K.U. Leuven
Y. Saeys, Universiteit Gent
R. Sepulchre, Université de Liège
M. van Someren, University of Amsterdam
A. Siebes, Universiteit Utrecht
Y. Smirnov, Universiteit Maastricht
K. Van Steen, Université de Liège
J. Suykens, K.U. Leuven
A. van den Bosch, Universiteit Tilburg
K. VanHoof, Universiteit Hasselt
M. Verleysen, U.C. Louvain
P. Vitanyi, Centrum voor Wiskunde en Informatica
L. Wehenkel, Université de Liège

Schedule

Monday, May 19th

9h15-9h30 Welcome

9h30-10h30 Invited Talk *(Chair: Damien Ernst)*

Susan Murphy, *Machine Learning and Reinforcement Learning in Clinical Research* p11

10h30-11h00 Coffee break

11h00-12h40 Session 1

Session 1.1 Reinforcement learning, planning, and games *(Chair: Susan Murphy)*

11h00-11h20 Tom Croonenborghs, Kurt Driessens, and Maurice Bruynooghe, *Learning a transfer function for reinforcement learning problems* p15

11h20-11h40 Boris Defourny, Damien Ernst, and Louis Wehenkel, *Perturb and combine in sequential decision making under uncertainty* p17

11h40-12h00 Raphael Fonteneau, Louis Wehenkel, and Damien Ernst, *Variable selection for dynamic treatment regimes: a reinforcement learning approach* p19

12h00-12h20 Jan Lemeire, *An alternative approach for playing complex games like chess* p21

Session 1.2 Graphical and relational models *(Chair: Kristel Van Steen)*

11h00-11h20 Luc De Raedt, *ProbLog* p23

11h20-11h40 Bernd Gutmann, Angelika Kimmig, Luc De Raedt, and Kristian Kersting, *Parameter learning in probabilistic databases: a least squares approach* p25

11h40-12h00 Ingo Thon, Niels Landwehr, and Luc De Raedt, *CPT-L: an efficient model for relational stochastic processes* p27

12h00-12h20 Vincent Auvray, and Louis Wehenkel, *Learning inclusion-optimal chordal graphs* p29

12h20-12h40 Sourour Ammar, Philippe Leray, Boris Defourny, and Louis Wehenkel, *Density estimation with ensembles of randomized poly-trees* p31

12h40-14h00 Lunch break

14h00-15h00 Invited Talk *(Chair: Raphaël Marée)*

Bill Triggs, *Scene segmentation with latent topic markov field models - and - classification and dimensionality reduction using convex class models* p11

15h00-15h30 Coffee break

15h30-17h30 Session 2

Session 2.1 Vision and speech *(Chair: Bill Triggs)*

15h30-15h50 Fabien Scalzo, Georgios Bebis, Mircea Nicolescu, and Leandro Loss, *Evolutionary learning of feature fusion hierarchies* p33

15h50-16h10 Cedric Simon, Jerome Meessen, and Christophe De Vleeschouwer, *Using decision trees to build an event recognition framework for automated visual surveillance* p35

16h10-16h30 Raphaël Marée, Pierre Geurts, and Louis Wehenkel, *Content-based image retrieval by indexing random subwindows with randomized trees* p37

16h30-16h50 Herman Stehouwer, *IGForest: From tree to forest* p39

16h50-17h10 Marie Dumont, Raphaël Marée, Pierre Geurts, and Louis Wehenkel, *Fast image annotation with random subwindows* p41

17h10-17h30 Thomas Drugman, Alexis Moinet, and Thierry Dutoit, *On the use of machine learning in statistical parametric speech synthesis* p43

Session 2.2 Feature selection and active learning *(Chair: Yvan Saeys)*

15h30-15h50 Yvan Saeys, Thomas Abeel, and Yves Van de Peer, *Towards robust feature selection techniques* p45

15h50-16h10 Marieke van Erp, Antal van den Bosch, Piroska Lendvai, and Steve Hunt, *Feature selection techniques for database cleansing: knowledge-driven vs greedy search* p47

16h10-16h30 Vân Anh Huynh-Thu, Louis Wehenkel, and Pierre Geurts, *Deriving p-values for tree-based variable importance measures* p49

16h30-16h50 Robby Goetschalckx, Scott Sanner, and Kurt Driessens, *Linear regression using costly features* p51

16h50-17h10 Dirk Gorissen, Tom Dhaene, and Eric Laermans, *Automatic regression modeling with active learning* p53

17h10-17h30 Kurt De Grave, Jan Ramon, and Luc De Raedt, *Active learning for primary drug screening* p55

Evening Conference dinner

Tuesday, May 20th

9h30-10h30 Invited Talk *(Chair: Pierre Geurts)*

Johannes Fürnkranz, *Preference learning* *p12*

10h30-11h00 Coffee break

11h00-12h40 Session 3

Session 3.1 Ranking and complex outputs *(Chair: Johannes Fürnkranz)*

11h00-11h20 Willem Waegeman, Bernard De Baets, and Luc Boullart, *When can we simplify a one-versus-one multi-class classifier to a single ranking?* *p57*

11h20-11h40 Michaël Rademaker, Bernard De Baets, and Hans De Meyer, *Monotone Relabeling of Partially Non-Monotone Data: Restoring Regular or Stochastic Monotonicity* *p59*

11h40-12h00 Pierre Geurts, Louis Wehenkel, and Florence d'Alché-Buc, *Learning in kernelized output spaces with tree-based methods* *p61*

12h00-12h20 Beau Piccart, Jan Struyf, and Hendrik Blockeel, *Selective Inductive Transfer* *p63*

12h20-12h40 Justus Piater, Fabien Scalzo, and Renaud Detry, *Vision as inference in a hierarchical markov network* *p65*

Session 3.2 Semi-supervised learning, missing data, and automata *(Chair: Pierre Dupont)*

11h00-11h20 Jerôme Callut, Kevin Françoisse, Marco Saerens, and Pierre Dupont, *Semi-supervised Classification in Graphs using Bounded Random Walks* *p67*

11h20-11h40 Amin Mantrach, Marco Saerens, and Luh Yen, *The Sum-Over-Paths Covariance: A novel covariance measure* *p69*

11h40-12h00 Jort Gemmeke, *Classification on incomplete data using sparse representations: Imputation is optional* *p71*

12h00-12h20 Yann-Michaël De Hauwere, Peter Vrancx, and Ann Nowé, *Multi-Agent State Space Aggregation using Generalized Learning Automata* *p73*

12h20-12h40 Sicco Verwer, Mathijs de Weerdt, and Cees Witteveen, *Efficiently learning timed models from observations* *p75*

12h40-14h00 Lunch break

14h00-15h00 Invited Talk *(Chair: Louis Wehenkel)*

Gunnar Rätsch, *Boosting, margins, and beyond* *p12*

15h00-15h30 Coffee break

15h30-17h30 Session 4

Session 4.1 Bioinformatics *(Chair: Gunnar Rätsch)*

15h30-15h50 Thomas Abeel, Yvan Saeys, and Yves Van de Peer, *ProSOM: Core promoter identification in the human genome* *p77*

15h50-16h10 Sofie Van Landeghem, Yvan Saeys, Bernard De Baets, and Yves Van de Peer, *Benchmarking machine learning techniques for the extraction of protein-protein interactions from text* *p79*

16h10-16h30 Aalt-Jan van Dijk, Dirk Bosch, Cajo ter Braak, Sander van der Krol, and Roeland van Ham, *Predicting sub-Golgi localization of glycosyltransferases* *p81*

16h30-16h50 Vincent Botta, Pierre Geurts, Sarah Hansoul, and Louis Wehenkel, *Prediction of genetic risk of complex diseases by supervised learning* *p83*

16h50-17h10 Gilles Meyer, and Rodolphe Sepulchre, *Component analysis for genome-wide association studies* p85

17h10-17h30 Fabien Scalzo, Peng Xu, Marvin Bergsneider, Xiao Hu, *Morphological Feature Extraction of Intracranial Pressure Signals via Nonlinear Regression* p87

Session 4.2 Applications *(Chair: Bernard Manderick)*

15h30-15h50 Tim Van de Cruys, *An extended NMF algorithm for word sense discrimination* p89

15h50-16h10 Koen Smets, Bart Goethals, and Brigitte Verdonk, *Automatic vandalism detection in wikipedia: towards a machine learning approach* p91

16h10-16h30 Bertrand Cornélusse, Louis Wehenkel, and Gérald Vignal, *Supervised learning of short-term strategies for generation planning* p93

16h30-16h50 Jean-Michel Dricot, Mathieu Van der Haegen, Yann-Ael Le Borgne, and Gianluca Bontempi, *Performance evaluation of machine learning techniques for the localization of users in wireless sensor networks* p95

16h50-17h10 Marc Ponsen, Jan Ramon, Kurt Driessens, Tom Croonenborghs, and Karl Tuyls, *Bayes-relational learning of opponent models from incomplete information in no-limit poker* p99

17h10-17h30 Francis wyffels, Benjamin Schrauwen, and Dirk Stroobandt, *System modeling with Reservoir Computing* p103

17:30-17:45 Closing

Invited talks

Machine Learning and Reinforcement Learning in Clinical Research

Susan A. Murphy
Institute for Social Research & Professor in Psychiatry, University of Michigan, USA

Abstract
This talk will survey some of the possible roles that machine learning researchers can play in informing and improving clinical practice. Clinical decision making, particularly when the patient has a chronic disorder, is adaptive. That is the clinician must adapt and then readapt treatment type, combinations and dose to the waxing and waning of the patient's chronic disorder. This adaption naturally occurs via clinical measurements of symptom severity, side effects, response to treatment, co-occuring disorders, etc. Currently most policies for guiding clinical decision making are informed primarily by expert opinion with an informal use of clinical trial data and clinical databases.
Some challenges in using trial data and databases are (1) there are usually many unknown causes of the patient observations; as a result high quality mechanistic models for the "system dynamics" are found only in very special cases. And (2) clinical databases often include many associations that are not causal; hence a simplistic application of learning methods can lead to gross biases. In addition to the causal issues, measures of confidence are crucial in gaining acceptance of policies constructed from data. Some advances in these areas will be discussed; however all of these are areas in which machine learning scientists could make a great impact.

Scene segmentation with latent topic markov field models and Classification and dimensionality reduction using convex class models

Bill Triggs
Laboratoire Jean Kuntzmann (LJK) and CNRS, Grenoble, France

Abstract
The talk will be in two parts. In the first part I will present work with Jakob Verbeek on semantic-level scene segmentation by combining spatial coherence models such as Markov and Conditional Random Fields with latent topic based local image content models such as Probabilisitic Latent Semantic Analysis over bag-of-words representations. In the second part I will present some recent work with Hakan Cevikalp, Frederic Jurie and Robi Polikar on using simple convex approximations to high-dimensional classes for multi-class classification and discriminant dimensionality reduction.

Preference learning

Johannes Fürnkranz
Knowledge Engineering Group, TU Darmstadt, Germany

Abstract
Preference Learning is a learning scenario that generalizes several conventional learning setttings, such as classification, multi-label classification, and label ranking. In this talk, we will give a brief introduction into this developing research area, and will in the following focus on our work on explicit modeling of pairwise preferences. In this approach, we learn a separate model for each possible pair of labels, which is used to decide which of the two labels is preferred. The predictions of the pairwise models are then combined into an overall ranking of all possible options. The key advantages of this approach lie in the simplicity of the pairwise models, and the possibility to combine the pairwise models in various ways, which allows to minimize different loss functions with the same set of trained classifiers. An obvious disadvantage is the complexity resulting from the need for training a quadratic number of classifiers. However, it can be shown that in many cases this problem can be efficiently solved. We will also briefly discuss extensions of the basic model for multilabel classification, for hierarchical classification, and for ordered classification.

Boosting, margins, and beyond

Gunnar Rätsch
Friedrich Miescher Laboratory of the Max Planck Society, Tbingen, Germany

Abstract
This talk will survey recent work on understanding Boosting in the context of maximizing the margin of separation. Starting with a brief introduction into Boosting in general and AdaBoost in particular, I will illustrate the connection to von Neumann's Minimax theorem and discuss AdaBoost's strategy to achieve a large margin. This will be followed by a presentation of algorithms which provably maximize the margin, are considerably quicker in maximizing the margin in practice and implement the soft-margin idea to improve the robustness against noise. In the second part I will discuss how these techniques relate to other convex optimization techniques and how they are connected to Support Vector Machines. Finally, I will talk about the effects of the the different key ingredients of Boosting and lessons learned from the application of such algorithms to real world problems.

Contributed abstracts

Learning a Transfer Function for Reinforcement Learning Problems

Tom Croonenborghs TOM.CROONENBORGHS@KHK.BE
Biosciences and Technology Department, KH Kempen University College, Geel, Belgium

Kurt Driessens KURT.DRIESSENS@CS.KULEUVEN.BE
Maurice Bruynooghe MAURICE.BRUYNOOGHE@CS.KULEUVEN.BE
Declarative Languages and Artificial Intelligence Group, Katholieke Universiteit Leuven (KUL), Belgium

Abstract

The goal of transfer learning algorithms is to utilize knowledge gained in a source task to speed up learning in a different but related target task. Recently, several transfer methods for reinforcement learning have been proposed. A lot of them require a hand-coded mapping that relates features from one task to another. This paper proposes a method to learn such a mapping automatically from interactions with the environment. Preliminary experiments show that our approach can learn a meaningful mapping that can be used to speed up learning through the execution of transferred actions during exploration.

1. Introduction

An area where transfer learning is particularly important is the domain of Reinforcement Learning (RL). In RL (Sutton & Barto, 1998), an agent can observe its world and perform actions in it. The agent's learning task is to maximize the reward he obtains. At the start of the learning task, the agent has no or little information and is forced to perform random exploration. As a consequence, learning can become infeasible or too slow in practice for complex domains and leveraging knowledge could increase the learning speed.

Recently, several approaches have been proposed to transfer knowledge between different reinforcement learning tasks. Often, a user-defined mapping is used to relate the new task to the task for which a policy was already learned, e.g. (Taylor et al., 2007; Torrey et al., 2006). There has been some work on learning such a mapping. E.g. in (Liu & Stone, 2006) a graph-matching algorithm is used to find similarities between state variables in the source and target task. This approach however needs a complete and correct transition model for both tasks.

In this paper, transfer learning is achieved by considering *transfer actions*, i.e. actions transferred from the source task to the target task during the exploration phase of the learning. To decide which action to transfer, the agent learns a function that predicts for each source action the probability that executing the transfered action is at least as good as executing the action which is best according to the agent's current utility function and selects the one with the highest probability.

2. Using Exploration to Transfer Knowledge

The standard exploration policy of the agent is altered sucht that with a probability of 10% the agent will select and execute a transfer action. To select the transfer action in a state t of the target problem, we will emplay a *transfer function* $p(s,t)$ that represents for each source state s the probability that the best action for s is at least as good for t as the best action according to the current approximation of the utility function. The transfer action executed in t is then the action performed on the state s for which $p(s,t)$ is maximal, i.e., $\pi_s(\text{argmax}_s\, p(s,t))$ with π_s the source task policy. Note that we assume for simplicity that actions in the source task are executable in the target task. In future work, we will extend our approach so that the transfer function maps (state,action)-pairs and is thereby able to incorporate the relation between actions.

3. Learning the Transfer Function

At the end of every learning episode in the target task, a number of learning examples for the transfer function can be generated. For every transfer action a_t the agent executed (in a state s) during that episode,

we compare the quality of the transfer action with the quality of his current policy using two different utility values. On the one hand, a Q-value $Q_{MC}(s, a_t)$ can be obtained by backtracking the Q-values from the end of that episode to the step where a_t is executed. This is comparable to a Monte Carlo estimate of this Q-value. On the other hand we let the agent learn a Q-value approximation $\hat{Q}$ of its current policy using a standard Q-learning type algorithm with generalization[1]. The generated learning example for each executed transfer action then consists of the states in both the source and target task and a label for the example: "transfer" if $Q_{MC}(s, a_t) \geq max_a \hat{Q}(s, a)$ and "no transfer" otherwise.

4. Preliminary Experiments

To evaluate our approach, the target task consists of a sequence of three four by four rooms. The rooms are connected with doors in the bottom right of each room. The agent can only pass a door if he possesses a key of the same color as the door. The keys are placed at random locations in a room. The primitive actions available to the agent include four movement actions (up,down,left and right) and a pickup action that picks up the key that is located at the agent's location (if applicable). The agent can execute at most 500 actions per episode and receives a reward of 1 if he exits the last room and 0 otherwise. The state representation includes the dimensions of the different rooms, the locations and colors of the doors, the location and colors of the keys, the keys the agent possesses, the agent's location and the goal location. The location consists of two coordinates, where a coordinate is determined by the relative position in the room and a certain offset for every room. In the source task, the agent has (successfully) learned how to navigate to the bottom right location in a four by four grid.

In a first experiment, instead of continuous learning the transfer function, we learned a single transfer function once with TILDE (Blockeel & De Raedt, 1998) based on learning examples created during the first 100 episodes in the target task and actions transferred from random source states. The algorithm was able to learn both the shift in coordinates between the different rooms and that successful transfer is very unlikely if the agent does not have the key needed to leave the current room.

We then restarted the agent in the environment with the learned transfer function. Figure 1 shows the average reward and number of actions per episode of a SARSA-learning agent, both with and without transfer. The numbers are obtained by freezing the current utility function and following a greedy test policy for 100 episodes every 50 episodes. We show results averaged over 10 runs.

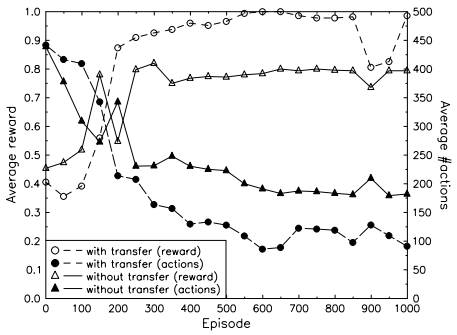


Figure 1. Results in the multi-room grid wold domain.

5. Future Work

Besides evaluating our approach in more detail, one important future direction is incorporating the quality of possible transfer in the agent's exploration policy. We would also like to substitute the batch learning of the transfer function as employed in our experiments, by a continuous, incremental approach.

Acknowledgments: Kurt Driessens is a post doctoral research fellow of the FWO.

References

Blockeel, H., & De Raedt, L. (1998). Top-down induction of first order logical decision trees. *Artificial Intelligence*, *101*, 285–297.

Liu, Y., & Stone, P. (2006). Value-function-based transfer for reinforcement learning using structure mapping. *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (pp. 415–20).

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: The MIT Press.

Taylor, M. E., Stone, P., & Liu, Y. (2007). Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, *8*, 2125–2167.

Torrey, L., Shavlik, J., Walker, T., & Maclin, R. (2006). Skill acquisition via transfer learning and advice taking. *Proceedings of the 17th European Conference on Machine Learning* (pp. 425–436).

[1]In the experiments, the SARSA-algorithm is used to learn a Q-function with the TG-algorithm, a first-order incremental decision tree learner.

Perturb and Combine in Sequential Decision Making under Uncertainty

Boris Defourny BORIS.DEFOURNY@ULG.AC.BE
Damien Ernst DERNST@ULG.AC.BE
Louis Wehenkel L.WEHENKEL@ULG.AC.BE
University of Liège, Department of Electrical Engineering and Computer Science,
Grande Traverse 10, Sart-Tilman, B-4000 Liège, Belgium

1. Overview

In the context of sequential decision making, rules that adapt the decisions to the evolving state of information are often preferable to a fixed sequence of decisions. As finding optimal such rules is complex, an approximation of the information state is often called for at the cost of some suboptimality (see Bertsekas, 2005).

In the classical discrete-time optimal control paradigm, the state of information is captured by state variables. An alternative approach (formalized in Section 2) consists in modeling the state of information by the history of a *disturbance process* that accounts for all the uncertainty over the planning horizon. Approximations over the disturbance process (in the context of convex optimization, see e.g. Rachev & Römisch, 2002) can simplify the representation of the information state without affecting the state space or the decision space.

If the disturbance space W has a finite number of elements, the disturbance process over T steps can be exactly represented by a *disturbance tree* of depth T and branching factor $|W|$. A decision rule mapping any particular outcome of a sequence of $t-1$ disturbances to a decision at time t can also be fully described by assigning decisions to the nodes of the tree. However, the size of the tree grows exponentially with T.

The celebrated Perturb and Combine principle (tracing back to Breiman, 1996) would recommend to replace the disturbance tree by an ensemble of incomplete, much smaller disturbance trees, built by a nondeterministic algorithm (described in Section 3). These partial representations of the disturbance process lead to distinct, partially specified decision rules, that can be optimized in parallel. A first-stage decision obtained from a combination of their first-stage decisions can be implemented, assuming that subsequent decisions will be subject to renewed computations.

Preliminary experiments (reported in Section 4) suggest that this approach brings formidable time savings, while being only slightly suboptimal.

2. Planning over a disturbance tree

Let $x_t \in X$, $u_t \in U$, $w_t \in W$ be the system state, the decision and the disturbance at time t. Let $x_{t+1} = f_t(x_t, u_t, w_t)$ be the state transition equation. Let $r_t(x_t, u_t, w_t)$ at $0 \leq t < T$ and $r_T(x_T)$ at $t = T$ be the reward at time t. Optimal decision rules μ_t^*, $0 \leq t < T$, maximize the expectation of a γ-discounted sum of the rewards:

$$J_{x_0}^* = \max_{\mu_t} \mathbb{E}\{\sum_{t=0}^{T-1} \gamma^t r_t(x_t, u_t, w_t) + \gamma^T r_T(x_T)\} \ . \quad (1)$$

In (1) the decision rule μ_t at time t is a mapping from histories $h_t = [w_0, w_1, \ldots, w_{t-1}]$ of the disturbance process to a decision u_t. At $t = 0$, h_0 is empty and μ_0 degenerates into a decision u_0. The state variable x_t is recovered by a chain of state transitions using the t decisions $u_\tau = \mu_\tau(h_\tau)$ and the t disturbances w_τ, $0 \leq \tau < t$, where h_τ and w_τ are extracted from h_t.

Under the convenient assumption that W is finite, there is a one-to-one correspondence between the nonterminal nodes of a disturbance tree of depth T and branching factor $|W|$, and all the possible histories h_t, $0 \leq t \leq T$. It suffices to assign a distinct element of W to the $|W|$ children of the nonterminal nodes, and view the t disturbances on the path from the root to a node n of depth t as the history associated to node n.

By assigning a decision u_n to nonterminal nodes n, a mapping from histories to decisions and hence a decision rule can be fully specified on the tree.

The expectation in (1) can also be optimized on the tree. Assuming first that the decisions u_n are set, values of state variables x_n and rewards r_n are assigned to a node n of history h_n, starting from x_0 at the

root (that has no reward associated with) and using a chain of proper state transitions along the path to n. The probability of an element of W is assigned to the nodes with that element. The node probabilities are then used in a progressive computation of a weighted sum of node rewards which ultimately gives the expectation S_μ of the γ-discounted sum of rewards under the u_n's. Such a procedure serves as an oracle for scoring the u_n's. A global solution to the maximization of S_μ over the u_n's gives (1), optimal u_n's, and hence an optimal decision rule, while suboptimal u_n's might still represent an acceptable suboptimal decision rule.

3. Decision making on incomplete trees

The complete disturbance tree is cumbersome. A simple method for building an incomplete disturbance tree consists in sampling a number m of disturbances, with m a random number in $\{1, 2, \ldots, q\}$, so as to create the children of its root. The number $m' \leq m$ of distinct disturbances defines the number of children, while the sample multiplicities induce probabilities assigned to them. The procedure is repeated for each node of depth $t < T$. The distribution of m, along with the tree depth T, determines the expected size of the tree.

The partial representation of the disturbance process by an incomplete tree leads to spurious opportunities likely to be exploited during the optimization of the decisions. Hence the great appeal of the Perturb and Combine paradigm to mitigate this effect.

An optimal first-stage decision is always well-defined for each incomplete tree. Under the assumption that the decision space U is finite, the aggregation of these decisions can be done through a majority vote.

4. Results on a navigation benchmark

The benchmark will illustrate how an ensemble of small incomplete trees advantageously replaces a single complete one.

A robot is in a corridor, at some position $x \in \{1, 2, \ldots, 19\}$. There are 2 exits at $x = 0$ and $x = 20$, with distinct rewards $r = 1$ and $r = 5$ obtained when the robot enters one of these terminal positions. The robot can move in both directions. A disturbance also affects the move. Specifically, $U = \{-1, +1\}$, $W = \{-1, 0, 1\}$ with $\mathbb{P}_w = \{0.25, 0.50, 0.25\}$, and

$$\begin{cases} x_{t+1} = x_t + u_t + w_t & \text{if } 0 < x_t + u_t + w_t < 20, \\ x_{t+1} = 0, & \text{if } x_t + u_t + w_t \leq 0, \\ x_{t+1} = 20, & \text{if } x_t + u_t + w_t \geq 20 \ . \end{cases}$$

The horizon is $T = 20$ with a discount factor $\gamma = 0.75$, so that the robot is in a hurry. (With these parameters, the optimal decisions are $u_t = -1$ if $1 \leq x_t \leq 7$, and $u_t = +1$ if $8 \leq x_t \leq 19$.)

The proposed approach is implemented as follows. The decision u_0 is chosen by a majority vote over 10 incomplete trees on which the node decisions have been optimized using the Cross-Entropy method (Rubinstein & Kroese, 2004). The sampling of m disturbances in the tree building algorithm is done with $\mathbb{P}(m = 1) = 0.6$, $\mathbb{P}(m = 2) = 0.2$, $\mathbb{P}(m = 3) = 0.2$, resulting in incomplete trees of about 1200 nodes in expectation (in constrast with the $5.2 \cdot 10^9$ nodes of the complete tree).

Simulations are carried out for the interesting initial positions $x_0 = 6, 7, 8, 9, 10$. The resulting decisions are respectively -1 (by 100% of the trees), -1(80%), -1(60%), $+1$(60%), $+1$(90%). The percentages suggest that some decisions are more reliable than others. (And in fact the decision -1 at $x_0 = 8$ is even suboptimal.) One could live with that, or carry out new simulations to ascertain the majority decision.

More technical details about the proposed approach are provided in (Defourny et al., 2008). Future work will focus on a broader set of benchmarks to evaluate the advantages of this approach on problems with large decision spaces and/or large disturbance spaces.

Acknowledgments

Damien Ernst is a Research Associate of the Belgian National Fund of Scientific Research (FNRS). This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

References

Bertsekas, D. (2005). Dynamic programming and suboptimal control: survey from ADP to MPC. *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference* (p. 50).

Breiman, L. (1996). Bagging predictors. *Machine Leaning*, *24*, 123–140.

Defourny, B., Ernst, D., & Wehenkel, L. (2008). Lazy planning under uncertainty by optimizing decisions on an ensemble of incomplete disturbance trees. Submitted for publication at EWRL08.

Rachev, S., & Römisch, W. (2002). Quantitative stability in stochastic programming: The method of probability metrics. *Mathematics of Operations Research*, *27*, 792–818.

Rubinstein, R., & Kroese, D. (2004). *The Cross-Entropy Method. A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning.* Information Science and Statistics. Springer.

Variable selection for dynamic treatment regimes: a reinforcement learning approach

Raphael Fonteneau RAPHAEL.FONTENEAU@ULG.AC.BE
Louis Wehenkel L.WEHENKEL@ULG.AC.BE
Damien Ernst[†] DERNST@ULG.AC.BE
Department of Electrical Engineering and Computer Science and GIGA-Research, University of Liège, Grande Traverse 10, 4000 Liège, Belgium. [†] Research Associate FNRS.

1. Introduction

Nowadays, many diseases as for example HIV/AIDS, cancer, inflammatory or neurological diseases are seen by the medical community as being chronic-like diseases, resulting in medical treatments that can last over very long periods. For treating such diseases, physicians often adopt explicit, operationalized series of decision rules specifying how drug types and treatment levels should be administered over time, which are referred to in the medical community as Dynamic Treatment Regimes (DTRs). Designing an appropriate DTR for a given disease is a challenging issue. Among the difficulties encountered, we can mention the complex dynamics of the human body interacting with treatments and other environmental factors, as well as the often poor compliance to treatments due to the side effects of some of the administered drugs. While typically DTRs are based on clinical judgment and medical insight, since a few years the biostatistics community is investigating a new research field addressing specifically the problem of inferring in a well principled way DTRs directly from clinical data gathered from patients under treatment. Among the results already published in this area, we mention (Murphy, 2005) which uses statistical tools for designing DTRs for psychotic patients.

2. Problem formulation

One possible approach to infer DTR from the data collected through clinical trials is to formalize this problem as an optimal control problem for which most of the information available on the 'system dynamics' (the system is here the patient and the input of the system is the treatment) is 'hidden' in the clinical data. This problem has been vastly studied in Reinforcement Learning (RL), a subfield of machine learning (see e.g., (Ernst et al., 2005)). Its application to the DTR problem would consist of processing the clinical data so as to compute a closed-loop treatment strategy which takes as inputs all the various clinical indicators which have been collected from the patients. Using policies computed in this way may however be inconvenient for the physicians who may prefer DTRs based on an as small as possible subset of *relevant* indicators rather than on the possibly very large set of variables monitored through the clinical trial. In this research, we therefore address the problem of determining a small subset of indicators among a larger set of candidate ones, in order to infer by RL convenient decision strategies. Our approach is closely inspired by work on 'variable selection' for supervised learning.

3. Learning from a sample

We assume that the information available for designing DTRs is a sample of discrete-time trajectories of treated patients, i.e. successive tuples (x_t, u_t, x_{t+1}), where x_t represents the state of a patient at some time-step t and lies in an n-dimensional space X of clinical indicators, u_t is an element of the action space (representing treatments taken by the patient in the time interval $[t, t+1]$), and x_{t+1} is the state at the subsequent time-step.

We further suppose that the responses of patients suffering from a specific type of chronic disease all obey the same discrete-time dynamics:

$$x_{t+1} = f(x_t, u_t, w_t) \quad t = 0, 1, \ldots$$

where disturbances w_t are generated by the probability distribution $P(w|x, u)$. Finally, we assume that one can associate to the state of the patient at time t and to the action at time t, a reward signal $r_t = r(x_t, u_t) \in \mathbb{R}$ which represents the 'well being' of the patient over the time interval $[t, t+1]$. Once the choice of the function $r_t = r(x_t, u_t)$ has been realized (a problem known as preference elicitation), the problem of finding a 'good' DTR may be stated as an optimal control problem for which one seeks to find a policy which leads to a sequence of actions $u_0, u_1, \ldots, u_{T-1}$, which maximizes,

over the time horizon $T \in \mathbb{N}$, and for any initial state the criterion:

$$R_T^{(u_0,u_1,\ldots,u_{T-1})}(x_0) = \mathop{\mathbb{E}}_{\substack{w_t \\ t=0,1,\ldots,T-1}} \left[\sum_{t=0}^{T-1} r(x_t, u_t) \right]$$

One can show (see e.g., (Ernst et al., 2005)) that there exists a policy $\pi_T^* : X \times [0, \ldots, T-1] \rightarrow U$ which produces such a sequence of actions for any initial state x_0. To characterize these optimal T-stage policies, let us define iteratively the sequence of *state-action value functions* $Q_N : X \times U \rightarrow \mathbb{R}$, $N = 1, \ldots, T$ as follows:

$$Q_N(x,u) = \mathop{\mathbb{E}}_{w} \left[r(x,u) + \sup_{u' \in U} Q_{N-1}(f(x,u,w), u') \right] \quad (1)$$

with $Q_0(x,u) = 0$ for all $(x,u) \in X \times U$. Dynamic programming theory implies that, for all $t \in \{1, \ldots, T-1\}$ and $x \in X$, the policy

$$\pi_T^\star(t,x) = \arg\max_{u \in U} Q_{T-t}(x,u)$$

is a T-step optimal policy.

Exploiting directly (1) for computing the Q_N-functions is not possible in our context since f is unknown and replaced here by an ensemble of one-step trajectories $\mathcal{F} = \left\{(x_t^l, u_t^l, r_t^l, x_{t+1}^l)\right\}_{l=1}^{\#\mathcal{F}}$, where $r_t^l = r(x_t^l, u_t^l)$. To address this problem, we exploit the fitted Q iteration algorithm which offers a way for computing (approximations of) the Q_N-functions ($\hat{Q}_N$) from the sole knowledge of $\mathcal{F}$ (Ernst et al., 2005). Notice that when used with tree based approximators, as it is the case in this paper, this algorithm offers good inference performances. Furthermore, we exploit the particular structure of these tree-based approximators in order to identify the most relevant clinical indicators among the n candidate ones.

4. Selection of clinical indicators

As mentioned in Section 2, we propose to find a small subset of state variables (clinical indicators), the m ($m \ll n$) most relevant ones with respect to a certain criterion, so as to create an m-dimensional subspace of X on which DTRs will be computed. The approach we propose for this exploits the tree structure of the $\hat{Q}_N$-functions computed by the fitted Q iteration algorithm. More specifically, it evaluates the relevance of each state variable x^i, by the score function:

$$S(x^i) = \frac{\sum_{N=1}^{T} \sum_{\tau \in \hat{Q}_N} \sum_{\nu \in \tau} \delta(\nu, x^i) \Delta_{var}(\nu) |\nu|}{\sum_{N=1}^{T} \sum_{\tau \in \hat{Q}_N} \sum_{\nu \in \tau} \Delta_{var}(\nu) |\nu|}$$

where ν is a nonterminal node in a tree τ (used to build the ensemble model representing one of the $\hat{Q}_N$-functions), $\delta(\nu, x^i) = 1$ if x^i is used to split at node ν and 0 otherwise, $\Delta_{var}(\nu)$ is the variance reduction when splitting node ν, and $|\nu|$ is the cardinality of the subset of tuples residing at node ν.

The approach then sorts the state variables x^i by decreasing values of their score so as to identify the m most relevant ones. A DTR defined on this subset of attributes is then computed by running the fitted Q iteration algorithm again on a 'modified $\mathcal{F}$', where the state variables of x_t^l and x_{t+1}^l that are not among these m most relevant ones are discarded.

The algorithm for computing a DTR defined on a small subset of state variables is thus as follows:
(1) compute the $\hat{Q}_N$-functions ($N = 1, \ldots, T$) using the fitted Q iteration algorithm on $\mathcal{F}$,
(2) compute the score function for each state variable, and determine the m best ones,
(3) run the fitted Q iteration algorithm on $\widetilde{\mathcal{F}} = \left\{(\widetilde{x}_t^l, u_t^l, r_t^l, \widetilde{x}_{t+1}^l)\right\}_{l=1}^{\#\widetilde{\mathcal{F}}}$ where $\widetilde{x}_t = \widetilde{M} x_t$, and $\widetilde{M}$ is a $m \times n$ boolean matrix where $\widetilde{m}_{i,j} = 1$ if the state variable x^j is the i-th most relevant one and 0 otherwise.

5. Preliminary validation

The method has been tested on the 'car on the hill' problem, a classical benchmark in RL (Ernst et al., 2005). This problem, which has a (continuous) state space of dimension two (the position p and the speed s of the car), is originally a deterministic problem. We have added to these variables some non-informative components so as to set up an experimental protocol. In our trials, the algorithm described previously was able to identify s and p as the most informative variables, which is encouraging for our future work with real-life clinical data.

Acknowledgments

This paper presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

References

Ernst, D., Geurts, P., & Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, *6*, 503–556.

Murphy, S. (2005). An experimental design for the development of adaptative treatment strategies. *Statistics in Medicine*, *24*, 1455–1481.

An Alternative Approach for Playing Complex Games like Chess

Jan Lemeire JAN.LEMEIRE@VUB.AC.BE
ETRO Dept., Vrije Universiteit Brussel, Brussels, Belgium

Abstract

Computer algorithms for game playing rely on a state evaluation which is based on a set of features and patterns. Such evaluation can, however, never fully capture the full complexity of games such as chess, since the impact of a feature or a pattern on the game outcome heavily relies on the game's context. It is a well-known problem in pattern-based learning that too many too specialized patterns are needed to capture all possible situations. We hypothesize that a pattern should be regarded as an opportunity to attain a certain state during the continuation of the game, which we call the effect of a pattern. For correct game state evaluation, one should analyze whether the desired effects of the matched patterns can be reached. Patterns indicate opportunities to reach a more advantageous situation. Testing whether this is possible in the current context is performed through a well-directed game tree exploration. We argue that this approach comes closer to the human way of game playing.

1. Why the Evaluation Fails

Besides the abundant game playing research in optimizing the brute-force minimax search much work is done on learning algorithms. They try to mimic human game playing. Explanation-based algorithms offer such an approach. In explanation-based learning (EBL), prior knowledge is used to analyze, or explain, how each observed training example satisfies the target concept (Mitchell et al., 1986). This explanation is then used to distinguish the relevant features of the training examples from the irrelevant, so that examples can be generalized based on logical rather than statistical reasoning. A *pattern* denotes an advantageous situation. The explanations must give the sufficient and necessary conditions for a pattern to be successful.

However, for a complex game like chess, patterns that have to capture all aspects of a game become too complex. Consider the task of learning to recognize chess positions - the explanations - in which "one's queen will be lost within the next few moves" - the pattern (Mitchell & Thrun, 1996). In a particular example, the queen could be lost due to a fork, in which "the white knight is attacking both the black king and queen". A fork is, however, hard to define correctly. One has to capture all situations in which the pattern leads to a successful outcome. All counter-plans that are available to the opponent for saving both its threatened pieces have to be excluded (Fürnkranz, 2001, p. 25). A quasi-unlimited number of counter moves, generated by the context in which the pattern appears, exist that can neutralize the effects. Minton (Minton, 1984) and Epstein (Epstein et al., 1996) highlight the same problem of learning too many too specialized rules with explanation-based learning. Even in simple games, such as tic-tac-toe, 45 concepts were learned with 52 exception clauses (Fawcett & Utgoff, 1991).

For adequate pattern-based evaluation functions, the patterns must contain all information on the outcome of the game. This can be written as:

$$state \perp\!\!\!\perp outcome \mid Patterns(state) \qquad (1)$$

where $Patterns(state)$ stands for the patterns that apply for $state$. All game-playing algorithms rely in one way or another on an evaluation of game states. A brute-force search tries to postpone an evaluation as much as possible, it explores all possible move sequences as far as possible into the future.

2. Alternative Approach

Our analysis is based on the observation that the outcome of a game is determined by the exact interaction of the patterns and heavily depends on the context of the game state. Trying to describe all the interactions leads, by the complexity of the game, to an enormous amount of rules or patterns. We hypothesize that the influence of a pattern on the game outcome depends on the achievement of certain states during the continuation of the game. We call these states the *effects* of the pattern. The influence of a pattern on the game outcome is completely described by these effects. The game can be analyzed by the set of existing patterns and whether their effects can be achieved. The difference with the explanation-based approach is that we do not

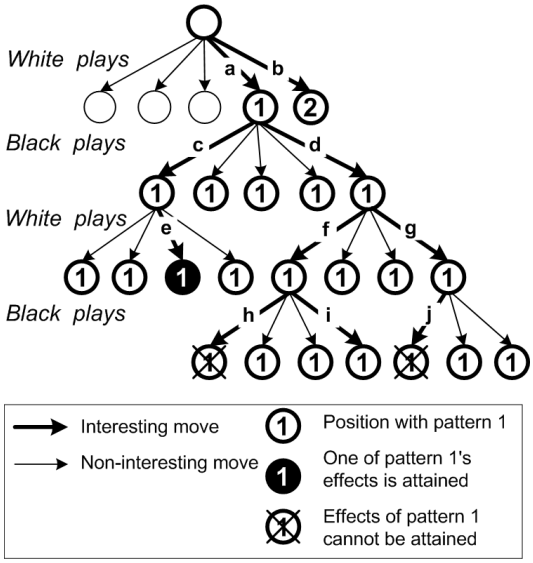


Figure 1. Game tree exploration by looking at patterns and their possible effects

expect the game always to reach the effect in the presence of the pattern.

Take the game tree of Fig. 1. Assume that the white player considers playing move a by which he arrives at a position in which pattern 1 is true. He hopes of achieving one of the advantageous effects of the pattern. The black player sees two possible counter moves. If he chooses for move c, however, white can collect the benefits of pattern 1 with move e. This is not possible if black chooses for move d. White can then play f or g, but in both cases black neutralizes the threat of pattern 1 with moves h and j respectively. Both moves bring the game in a state in which the positive effects of the pattern cannot be attained anymore. Note that not all possible moves have to be explored. The game tree can be pruned effectively. Only the moves interfering with the pattern has to be explored. The other moves can be classified as being irrelevant, since they do not approximate white to the achievement of pattern 1's effect.

We thus have defined a new kind of generic knowledge; patterns together with their effects. However, an implementation of this approach needs a yet inexistent pattern engine. We do not have a general way to describe, recognize, learn and reason with patterns.

3. Human-like Game Playing

Psychological studies have shown that the differences in playing strengths between chess experts and novices are not so much due to differences in the ability to calculate long move sequences, but to which moves they start to calculate. Cowley and Byrne showed that chess experts rely on falsification (Cowley & Byrne, 2004). The results of the research show that chess masters were readily able to falsify their plans. They generated move sequences that falsified their plans more readily than novice players, who tended to confirm their plans. Our approach confirms this; it is based on plans and on falsification.

It's well-known that humans have difficulties formally defining the knowledge they use. Our approach can explain this. A pattern only denotes an opportunity. A precise description of the states in which it is successful is not necessary, a well-directed tree search is used to confirm or falsify the hypothesis. Our approach also explains why humans can reason about a game, why we can exactly pinpoint which actions were decisive in a game and why.

References

Cowley, M., & Byrne, R. M. J. (2004). Chess masters hypothesis testing. *Proceedings of the 26th Annual Conference of the Cognitive Science Society. Mahwah, NJ* (pp. 250–255).

Epstein, S. L., Gelfand, J. J., & Lesniak, J. (1996). Pattern-based learning and spatially-oriented concept formation in a multi-agent, decision-making expert. *Computational Intelligence*, *12*, 199–221.

Fawcett, T., & Utgoff, P. E. (1991). A hybrid method for feature generation. *Machine Learning: Proceedings of the Eighth International Workshop* (pp. 137–141). Morgan Kaufmann.

Fürnkranz, J. (2001). Machine learning in games: A survey. In J. Fürnkranz and M. Kubat (Eds.), *Machines that learn to play games*, 11–59. Huntington, NY: Nova Science Publishers.

Gould, J., & Levinson, R. A. (1991). *Method integration for experience-based learning* (Technical Report UCSC-CRL-91-27). UCSC, Santa Cruz, CA.

Minton, S. (1984). Constraint-based generalization: Learning game-playing plans from single examples. *Proceedings of the 2nd National Conference on Artificial Intelligence, Austin, TX* (pp. 251–254).

Mitchell, T., Keller, R., & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, *1*, 47–80.

Mitchell, T. M., & Thrun, S. B. (1996). Learning analytically and inductively. In D. M. Steier and T. M. Mitchell (Eds.), *Mind matters: A tribute to Allen Newell*, 85–110. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc.

ProbLog and its Application to Link Mining in Biological Networks

Luc De Raedt LUC.DERAEDT@CS.KULEUVEN.BE

Dept. of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, POBox 2402, B-3001 Heverlee, Belgium

Abstract

ProbLog is a recently introduced probabilistic extension of Prolog (De Raedt et al., 2007). A ProbLog program defines a distribution over logic programs by specifying for each clause the probability that it belongs to a randomly sampled program, and these probabilities are mutually independent. The semantics of ProbLog is then defined by the success probability of a query in a randomly sampled program. It has been applied to link mining and discovery in a large biological network. In the talk, I will also discuss various learning settings for ProbLog and link mining, in particular, I shall present techniques for probabilistic local pattern mining (Kimmig & De Raedt, 2008), probabilistic explanation based learning (Kimmig et al., 2007) and theory compression from examples (De Raedt et al., 2008).

Acknowledgments

This is joint work with Angelika Kimmig, Hannu Toivonen, Bernd Gutmann, Kate Revoredo and Kristian Kersting.

References

De Raedt, L., Kersting, K., Kimmig, A., Revoredo, K., & Toivonen, H. (2008). Compressing probabilistic prolog programs. *Machine Learning*, *70*, 151–168.

De Raedt, L., Kimmig, A., & Toivonen, H. (2007). ProbLog: A probabilistic Prolog and its application in link discovery. *Proceedings of IJCAI* (pp. 2462–2467).

Kimmig, A., & De Raedt, L. (2008). Local pattern mining in probabilistic databases. submitted.

Kimmig, A., De Raedt, L., & Toivonen, H. (2007). Probabilistic explanation based learning. *Proceedings 18th European Conference on Machine Learning* (pp. 176–187).

Parameter Learning in Probabilistic Databases: A Least Squares Approach

Bernd Gutmann BERND.GUTMANN@CS.KULEUVEN.BE
Angelika Kimmig ANGELIKA.KIMMIG@CS.KULEUVEN.BE
Luc De Raedt LUC.DERAEDT@CS.KULEUVEN.BE
Dept. of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, POBox 2402, BE-3001 Heverlee, Belgium

Kristian Kersting KRISTIAN.KERSTING@IAIS.FRAUNHOFER.DE
Fraunhofer IAIS, Schloß Birlinghoven, 53754 Sankt Augustin, Germany

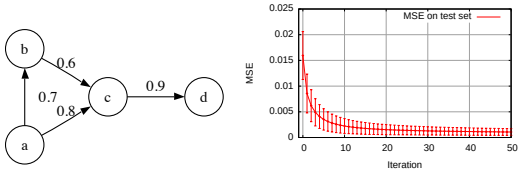
Keywords: Learning, Graphs, Probabilistic Databases, Logic

Abstract

Probabilistic databases compute the success probabilities of queries. We introduce the problem of learning the parameters of the probabilistic database ProbLog. Given the observed success probabilities of a set of queries, we use a least squares approach to compute the probabilities attached to facts that have a low approximation error on the training data as well as on unseen examples.[1]

1. Introduction

The statistical relational learning community has devoted a lot of attention to learning both the structure and parameters of probabilistic logics, cf. (Getoor & Taskar, 2007; De Raedt et al., 2008), but so far seems to have devoted little attention to the learning of probabilistic database formalisms. Probabilistic databases (Dalvi & Suciu, 2004; De Raedt et al., 2007) associate probabilities to facts, indicating the probabilities with which the facts hold. This information is then used to define and compute the success probability of queries or derived facts or tuples. Because probabilistic databases do not constitute a generative model, it has – so far – been unclear as how to learn such databases. In this paper, we introduce the problem of learning the parameters of probabilistic databases from a set of queries together with their target probabilities. The approach is incorporated in the probabilistic database ProbLog (De Raedt et al., 2007), though it can easily be integrated in other probabilistic databases as well. ProbLog has been designed to support life scientists that mine a large network of biological entities in interactive querying sessions..



(a) (b)

Figure 1. (a) Probabilistic graph. (b) Learning curve with standard deviation, on test set for graph with 88 edges.

2. ProbLog

ProbLog is a simple probabilistic extension of Prolog introduced in (De Raedt et al., 2007). A ProbLog program consists – as Prolog – of a set of definite clauses. However, in ProbLog every fact c_i is labeled with the probability p_i that it is true, and those probabilities are assumed to be mutually independent. For ease of illustration, we will consider probabilistic graphs like the one in Figure 1(a) in the following, but the entire discussion carries over to arbitrary ProbLog programs. Such a probabilistic graph can be used to sample subgraphs by tossing a biased coin for each edge. The corresponding ProbLog program $T = \{p_1 : c_1, \cdots, p_n : c_n\}$ therefore defines a probability distribution over subgraphs $L \subseteq L_T = \{c_1, \cdots, c_n\}$ in the following way:

$$P(L|T) = \prod_{c_i \in L} p_i \prod_{c_i \in L_T \setminus L} (1 - p_i).$$

[1] This is a shortened version of an extended abstract submitted to the 6th International Workshop on Mining and Learning with Graphs (MLG2008)

It is straightforward to add background knowledge in the form of Prolog clauses, say, the definition of a path by combining edges. We can then ask for the probability that there exists e.g. a path between nodes a and c in our probabilistic graph, i.e. the probability that a randomly sampled subgraph contains the edge from a to c, or the path from a to c via b (or both of them). Formally, the *success probability* $P_s(q|T)$ of a query q in a ProbLog program T is defined as

$$P_s(q|T) = \sum\nolimits_{L \subseteq L_T} P(q|L) \cdot P(L|T) \;, \qquad (1)$$

where $P(q|L) = 1$ if there exists a θ such that $L \models q\theta$, and $P(q|L) = 0$ otherwise. The success probability of query q corresponds to the probability that the query q is *provable* in a randomly sampled logic program. Due to presence of multiple paths in samples, evaluating the success probability of ProbLog queries is computationally hard, see (De Raedt et al., 2007) for an approximation algorithm.

3. Parameter Learning

ProbLog does not provide a generative model for sampling queries (e.g. paths between nodes). Thus, we cannot directly apply standard maximum likelihood techniques for parameter estimation based on the EM algorithm. We consider parameter learning for ProbLog as a function optimization problem:

Definition 1 (ProbLog Parameter Learning) ***Given*** *a set of training examples* $\{q_i, \tilde{p}_i\}_{i=1}^K$, $K > 0$, *where each* $q_i \in \mathcal{H}$ *is a logical query with success probability* $\tilde{p}_i$, ***find*** *a function* $h : \mathcal{H} \to [0, 1]$ *with low approximation error on the training data as well as on unseen examples.* $\mathcal{H}$ *comprises all parameter assigments for a given logical program* T.

We want to minimize the mean squared error (MSE):

$$MSE(T) = \frac{1}{K} \sum\nolimits_{1 \leq i \leq K} \left(P_s(q_i|T) - \tilde{p}_i\right)^2 . \qquad (2)$$

It is easy to show that minimizing the squared error in this case corresponds to finding a maximum likelihood hypothesis, provided that for each training example $(q_i, \tilde{p}_i)$, a Gaussian error is included, i.e. $\tilde{p}_i = p(q_i) + e_i$, with $p(q_i)$ the actual probability of query q_i and e_i drawn independently from a Gaussian with mean zero. We now derive the gradient of the MSE. Applying the sum and chain rule to Eq. (2) yields the partial derivative $\partial MSE(T)/\partial p_j =$

$$\frac{2}{K} \sum_{1 \leq i \leq K} \underbrace{\left(P_s(q_i|T) - \tilde{p}_i\right)}_{\text{Part 1}} \cdot \underbrace{\frac{\partial\, P_s(q_i|T)}{\partial p_j}}_{\text{Part 2}} . \qquad (3)$$

We apply standard gradient descent to minimize the MSE. (3) can be evaluated by ProbLog inference directly (Part 1) and by slightly adapting the underlying techniques (Part 2).

4. Experiments

We implemented the gradient descent algorithm in Prolog (Yap-5.1.3). Since this is ongoing work, we primarily try to answer the question: *does the gradient descent minimize the MSE?*

As our test graph G, we used a real biological graph around 3 random Alzheimer genes, with 45 nodes and 88 edges, cf. (De Raedt et al., 2007). We randomly sampled 100 node pairs and calculated the probability that there exists a path between them using approximative ProbLog inference. We performed 5-fold cross-validation, initializing the parameters randomly, with fixed seed for succeeding experiments. Figure 1(b) shows the learning curve for the test set. After 50 iterations, the MSE averaged over 5 folds is 0.00016 ± 0.00001 on the training set and 0.00107 ± 0.00065 on the test set, answering our question positively.

5. Conclusions

We introduced an approach to parameter learning for the probabilistic database ProbLog and successfully showed it at work on a real biological application. Interesting directions for future research include optimizing the learning algorithm and regularization-based cost functions. Those enable domain experts to refine probabilities of a database by stating examples.

Acknowledgments AK, BG are supported by the Research Foundation-Flanders (FWO-Vlaanderen), KK by a Fraunhofer ATTRACT fellowship.

References

Dalvi, N. N., & Suciu, D. (2004). Efficient query evaluation on probabilistic databases. *VLDB* (pp. 864–875).

De Raedt, L., Frasconi, P., Kersting, K., & Muggleton, S. (Eds.). (2008). *Probabilistic inductive logic programming - theory and applications*, vol. 4911 of *LNAI*. Springer-Verlag.

De Raedt, L., Kimmig, A., & Toivonen, H. (2007). ProbLog: A probabilistic Prolog and its application in link discovery. *IJCAI* (pp. 2462–2467).

Getoor, L., & Taskar, B. (Eds.). (2007). *Statistical relational learning*. The MIT press.

CPT-L: an Efficient Model for Relational Stochastic Processes

Ingo Thon INGO.THON@CS.KULEUVEN.BE
Niels Landwehr NIELS.LANDWEHR@CS.KULEUVEN.BE
Luc De Raedt LUC.DERAEDT@CS.KULEUVEN.BE
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Abstract

Agents that learn and act in real-world environments have to cope with both complex state descriptions and non-deterministic transition behavior of the world. Standard statistical relational learning techniques can capture this complexity, but are often inefficient. We present a simple probabilistic model for such environments based on CP-Logic. efficiency is maintained by restriction to a fully observable setting.

1. Introduction

Artificial intelligence aims at developing agents that learn and act in complex environments. Realistic environments typically feature a variable number of objects, relations amongst them, and non-deterministic transition behavior. Standard probabilistic sequence models provide efficient inference and learning techniques, but typically cannot fully capture the relational complexity. On the other hand, statistical relational learning techniques are often too inefficient. In this paper, we present a simple model that occupies an intermediate position in this expressiveness/efficiency trade-off. More specifically, we contribute a novel representation, called CPT-L (for **CPT**ime-**L**ogic), that essentially defines a probability distribution over sequences of interpretations. Interpretations are relational state descriptions that are typically used in planning and many other applications of artificial intelligence. CPT-L is a variation of CP-logic (Vennekens et al., 2006), a recent expressive logic for modeling causality. By focusing on the sequential aspect and deliberately avoiding the complications that arise when dealing with hidden variables, CPT-L is more restricted, but also more efficient to use than its predecessor and alternative formalisms within the artificial intelligence and statistical relational learning literature.

This is clear when positioning CPT-L w.r.t. to the few existing approaches that can probabilistically model sequences of relational state descriptions. First, standard SRL-approaches (Getoor & Taskar, 2007) can be used in this setting by explicitly modeling time. However, such models are often intractable for complex sequential real-world domains. Second, relational STRIPS-based techniques (Zettlemoyer et al., 2005) are able to probabilistically model relational sequences. However, they are restricted by the fact that only one rule can "fire" at a particular point in time and thus only one aspect of the world can be changed. The key contributions of our work are the introduction of 1) the CPT-L model for representing probability distributions over sequences of interpretations, 2) we report that efficient inference is possible under the assumption of fully observability and the restriction to sequential causal effects.

2. CPT-L

A relational interpretation I is a set of ground facts $a_1, ..., a_N$.A *relational stochastic process* defines a distribution $P(I_1, ..., I_T)$ over sequences of interpretations of length T.The semantics of CPT-L is based on CP-logic, a probabilistic first-order logic that defines probability distributions over interpretations (Vennekens et al., 2006). CP-logic has a strong focus on causality and constructive processes: an interpretation is incrementally constructed by a process that adds facts which are probabilistic *outcomes* of other already given facts (the *causes*). CPT-L combines the semantics of CP-logic with that of (first-order) Markov processes. Causal influences only stretch from I_t to I_{t+1} (Markov assumption), are identical for all time steps (stationarity), and all causes and outcomes are observable. Models in CPT-L are also called CP-theories, and are defined as follows:

Definition 1. *A* **CPT-theory** *is a set of rules of the form*

$$r = \underbrace{(h_1 : p_1) \vee \ldots \vee (h_n : p_n)}_{head(r)} \leftarrow \underbrace{b_1, \ldots, b_m}_{body(r)}$$

where the h_i are logical atoms, the b_i are literals (i.e., atoms or their negation) and $p_i \in [0, 1]$ probabilities s.t. $\sum_{i=1}^{n} p_i = 1$.

We shall also assume all variables appearing in the head of the rule also appear in its body. The intuition behind a rule is that whenever the (grounded) body of the rule holds in the current state I_t, one of the (grounded) heads will hold in the next state I_{t+1}. In this way, the rule models

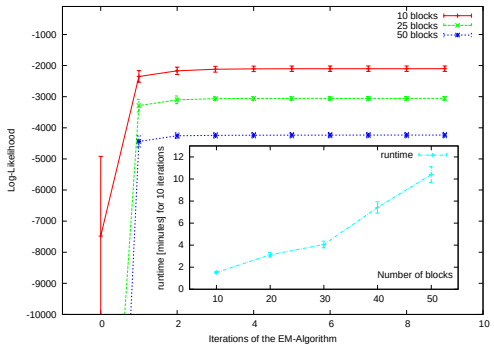


Figure 1. Large graph: per-sequence log-likelihood on training data as a function of the EM iteratio. Small graph: Running time of EM as a function of the number of blocks in the world model.

a (probabilistic) causal process as the condition specified in the body causes one (probabilistically chosen) atoms in the head to become true in the next time step. One of the main features of CPT-theories is that they are easily extended to include *background knowledge*, which can be any logic program (cf. (Bratko, 1990)). In the presence of background knowledge, we say that a ground rule is applicable in an interpretation I_t if its body $b_1\theta, \ldots, b_m\theta$ can be logically derived from I_t and the logic program B.

A CPT-theory defines a distribution over possible successor states, $P(I_{t+1} \mid I_t)$, in the following way. Let $\mathbf{R}_t = \{r_1, ..., r_k\}$ denote the set of all ground rules applicable in the current state I_t. Each ground rule applicable in I_t will cause one of its head elements to become true in I_{t+1}. More formally, a *selection* σ is a mapping from rules r_i to indices j_i denoting that head element $h_{ij_i} \in head(r_i)$ is selected. In the stochastic process to be defined, I_{t+1} is a possible successor for the state I_t if and only if there is a selection σ such that $I_{t+1} = \{h_{1\sigma(1)}, ..., h_{k\sigma(k)}\}$. We say that σ *yields* I_{t+1} from I_t, denoted $I_t \xrightarrow{\sigma} I_{t+1}$, and define

$$P(I_{t+1}|I_t) = \sum_{\sigma: I_t \xrightarrow{\sigma} I_{t+1}} P(\sigma) = \sum_{\sigma: I_t \xrightarrow{\sigma} I_{t+1}} \prod_{(r_i, j_i) \in \sigma} p_{j_i}, \quad (1)$$

where p_{j_i} is the probability associated with head element h_{ij_i} in r_i. As for propositional Markov processes, the probability of a sequence $I_1, ..., I_T$ given an initial state I_0 is defined by

$$P(I_1, ..., I_T) = P(I_1) \prod_{t=0}^{T} P(I_{t+1} \mid I_t). \quad (2)$$

Intuitively, it is clear that this defines a distribution over all sequences of interpretations of length T much as in the propositional case.

3. Experimental Evaluation

The proposed CPT-L model has been evaluated in a stochastic version of the well-known *blocks world* domain. The domain was chosen because it is truly relational and also serves as a popular artificial world model in agent-based approaches such as planning and reinforcement learning. Furthermore, it is an example for a domain in which multiple aspects of the world can change concurrently — for instance, a block can be moved from A to B while at the same time a stack collapses, spilling all of its blocks on the floor. In an experiment, we explore the convergence behavior of the EM algorithm for CPT-L. The world model together is implemented by a (gold-standard) CPT-theory $\mathcal{T}$, and a training set of 20 sequences of length 50 each is sampled from $\mathcal{T}$. From this data, the parameters are re-learned using EM. Figure 1, large graph, shows the convergence behavior of the algorithm on the training data for different numbers of blocks in domain, averaged over 15 runs. It shows rapid and reliable convergence. Figure 1, small graph, shows the running time of EM as a function of the number of blocks. The scaling behavior is roughly linear, indicating that the model scales well to reasonably large domains. Absolute running times are also low, with about 1 minute for an EM iteration in a world with 50 blocks.This is in contrast to other, more expressive modeling techniques which typically scale badly to domains with many objects. The difference between the log likelihood on an independent test set of the gold-standard model and the learned model, were by four orders of magnitudes smaller than the difference to a random model. Manual inspection of the learned model also shows that parameter values are on average very close to those in the gold-standard model.

4. Conclusions and Future Work

We have introduced CPT-L, a probabilistic model for sequences of relational state descriptions. In contrast to other approaches that address this setting, the focus in CPT-L is on computational efficiency rather than maximal expressivity. The main interesting directions for future work is to further evaluate representation power and scaling behavior of the model in challenging real-world domains.

References

Bratko, I. (1990). *Prolog programming for artificial intelligence*. Addison-Wesley. 2nd Edition.

Getoor, L., & Taskar, B. (Eds.). (2007). *Statistical relational learning*. MIT press.

Vennekens, J., Denecker, M., & Bruynooghe, M. (2006). Representing causal information about a probabilistic process. *Logics In Artificial Intelligence* (pp. 452–464).

Zettlemoyer, L. S., Pasula, H., & Kaelbling, L. P. (2005). Learning planning rules in noisy stochastic worlds. *AAAI-05* (pp. 911–918).

Learning Inclusion-Optimal Chordal Graphs

Vincent Auvray VINCENT.AUVRAY@ULG.AC.BE
Louis Wehenkel L.WEHENKEL@ULG.AC.BE
GIGA-R and Department of Electrical Engineering and Computer Science, University of Liège, Belgium

Abstract

This abstract discusses a very simple and efficient algorithm to learn the chordal structure of a probabilistic model from data. The algorithm is a greedy hill-climbing search algorithm that uses the inclusion boundary neighborhood over chordal graphs. In the limit of a large sample size and under appropriate hypotheses on the scoring criterion, the algorithm will find a structure that is inclusion-optimal when the dependency model of the data-generating distribution can be perfectly represented by an undirected graph.

1. Introduction

This abstract considers the class of graphical models whose structure is a chordal graph, known as the class of decomposable models. A chordal graph is an undirected graph (UG) where every cycle comprising more than three edges has a chord. The class of dependency models defined by chordal graphs is the intersection of the class of directed acyclic graphs (DAGs) dependency models and the class of UG dependency models.

A greedy hill-climbing search algorithm is often used to learn the DAG structure of a Bayesian Network. Different choices of search spaces and neighborhoods connecting the search space are possible. In particular, the search may proceed over the set of Markov equivalence classes of DAG structures by exploiting the inclusion boundary neighborhood (see (Chickering, 2002; Auvray & Wehenkel, 2002)). Under appropriate assumptions on the scoring criterion and on the data-generating distribution, a greedy algorithm using the inclusion boundary neighborhood returns an inclusion-optimal structure in the limit of a large sample size (see (Chickering & Meek, 2002) and (Castelo & Kočka, 2003)). Unfortunately, the size of the inclusion boundary of an equivalence class of a DAG structure is in the worst case exponential in the number of variables. The notion of inclusion boundary neighborhood can also be defined over sets of chordal graphs. In this context, its size is bounded from above by the square of the number of variables and it can be computed easily.

In this abstract, we investigate the optimality properties of the greedy hill-climbing search algorithm using the inclusion boundary neighborhood to learn a chordal structure. As mentionned above in the case of DAG structures, a desirable property of a structure learning algorithm is to return an inclusion-optimal solution. We describe a local asymptotic consistency property of scoring criteria that ensures that a greedy search will produce an inclusion-optimal chordal structure when the independence relations holding in the data-generating distribution can be represented exactly by an undirected graph. For more details and omitted proofs, see (Auvray & Wehenkel, 2008).

2. Background

Consider an undirected graph G whose vertex set X is a set of random variables. Given disjoints sets $A, B, C \subseteq X$, we say that A and B are separated by C in G if all paths between a vertex in A and a vertex in B go through at least one vertex in C. The dependency model encoded by G consists of the marginal and conditional independence relations $A \perp B|C$ such that A and B are separated by C in G. In the sequel, we sometimes identify an undirected graph and its dependency model.

Let us define the notion of inclusion-optimality for chordal graphs. Consider a particular dependency model M_0. We say that a chordal dependency model M is inclusion-optimal for M_0 if $M_0 \subseteq M$ and there is no chordal dependency model M' such that $M_0 \subseteq M' \subseteq M$. This notion has a simple graphical interpretation: a chordal graph G encodes an inclusion-optimal dependency model for M_0 if, and only if, (a) it does not encode any independence assumption that does not hold in M_0 and (b) every chordal subgraph of G encodes such an incorrect independence assumption.

Let us present the notion of inclusion boundary for chordal graphs. The inclusion boundary of a chordal graph G is the set of chordal graphs H satisfying

- $I(G) \subseteq I(H)$ and there is no chordal graph K such that $I(G) \subseteq I(K) \subseteq I(H)$, or
- $I(H) \subseteq I(G)$ and there is no chordal graph K such that $I(H) \subseteq I(K) \subseteq I(G)$,

where $I(C)$ denotes the dependency model (i.e. the set of conditional independencies) encoded by the chordal graph C. It is straightforward to describe graphically the inclusion boundary of a chordal graph G: it consists of the chordal graphs that differ from G by the addition or removal of a single edge. This is a consequence of the fact that, for any two chordal graphs G, H such that H is a subgraph of G, there exists a sequence of chordal graphs $K_0, \ldots, K_n$ such that $K_0 = H$, $K_n = G$ and K_{i+1} is obtained from K_i by adding a single edge (see (Giudici & Green, 1999)).

3. Inclusion-optimality of greedy search

Following the terminology of (Chickering & Meek, 2002), we say that a scoring criterion score$(\cdot)$ for chordal graphs is locally consistent for a dependency model I if, for each vertices a, b and chordal graphs G, H such that H is obtained from G by removing $a - b$, we have

1. $a \perp b | ne_G(a) \cap ne_G(b) \in I \Rightarrow \text{score}(H) > \text{score}(G)$,
2. $a \perp b | ne_G(a) \cap ne_G(b) \notin I \Rightarrow \text{score}(G) > \text{score}(H)$,

where $ne_K(a)$ denotes the sets of neighboring (i.e. adjacent) vertices of a in K.

Recall that a scoring criterion score$(\cdot)$ for a DAG dependency model encoded by G is decomposable if it can be written as a sum of functions that depend each on only one vertex and its parents, i.e.

$$\text{score}(G) = \sum_{v \in V} f\big(v, pa_G(v)\big).$$

The following proposition holds.

Proposition 1. *If* score$(\cdot)$ *is a scoring criterion over DAG dependency models that is decomposable and consistent for a dependency model I, then it is locally consistent for I when restricted to chordal graphs and*

$$\text{score}(G) - \text{score}(H) = f\big(b, \{a\} \cup (ne_G(a) \cap ne_G(b))\big) - f\big(b, ne_G(a) \cap ne_G(b)\big),$$

for chordal graphs G and H such that H is obtained from G by removing the edge $a - b$.

In practice, scoring criteria over DAG dependency models only satisfy the consistency property asymptotically in the limit of a large sample size. When restricted to chordal dependency models, such scoring criteria will only be locally consistent asymptotically.

The main result of this abstract can now be stated.

Proposition 2. *If* score *is a scoring criterion for chordal graphs that is consistent and locally consistent for a graph-isomorph dependency model I, then local optima of* score$(\cdot)$ *w.r.t. the inclusion boundary neighborhood are inclusion-optimal for I.*

Acknowledgments

This work presents research results of the Belgian Network BIOMAGNET funded by the Interuniversity Attraction Poles Programme. Vincent Auvray is supported by the "Action de recherche concertée" BIOMOD funded by the French Speaking Community of Belgium.

References

Auvray, V., & Wehenkel, L. (2002). On the construction of the inclusion boundary neighbourhood for Markov equivalent classes of Bayesian network structures. *Proceedings of Eighteenth Conference on Uncertainty in Artificial Intelligence* (pp. 26–35). Morgan Kaufmann.

Auvray, V., & Wehenkel, L. (2008). Learning inclusion-optimal chordal graphs. *Proceedings of Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*. (to appear).

Castelo, R., & Kočka, T. (2003). On inclusion-driven learning of Bayesian networks. *Journal of Machine Learning Research, 4*, 527–574.

Chickering, D., & Meek, C. (2002). Finding optimal bayesian networks. *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)* (pp. 94–102). San Francisco, CA: Morgan Kaufmann.

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research, 3*, 507–554.

Giudici, P., & Green, P. J. (1999). Decomposable graphical Gaussian model determination. *Biometrika, 86*, 785–801.

Density estimation with ensembles of randomized poly-trees

Sourour Ammar [1 2] SOUROUR.AMMAR@ETU.UNIV-NANTES.FR
Philippe Leray [1] PHILIPPE.LERAY@UNIV-NANTES.FR
Boris Defourny [3] BORIS.DEFOURNY@ULG.AC.BE
Louis Wehenkel [3] L.WEHENKEL@ULG.AC.BE

[1] Laboratoire d'Informatique de Nantes Atlantique (LINA) UMR 6241, École Polytechnique de l'Université de Nantes, France

[2] Laboratoire d'Informatique, Traitement de l'Information et des Systèmes (LITIS) EA 4108 - Institut National des Sciences Appliquées de Rouen, France

[3] Department of Electrical Engineering and Computer Science & GIGA-Research, University of Liège, Belgium

1. Motivation

Learning of Bayesian networks aims at modeling the joint density of a set of random variables from a random sample of joint observations of these variables (Naïm et al., 2007). Such a graphical model may be used for elucidating the conditional independences holding in the datagenerating distribution, for automatic reasoning under uncertainties, and for Monte-Carlo simulations. Unfortunately, currently available algorithms for Bayesian network structure learning are either restrictive in the kind of distributions they search for, or of too high computational complexity to be applicable in high dimensional spaces.

Ensembles of weakly fitted randomized models have been studied intensively and used successfully in the supervised learning literature during the last two decades. Among the advantages of these methods, let us quote the improved scalability of their learning algorithms thanks to randomization and the improved predictive accuracy the induced models thanks to their higher flexibility in terms of bias/variance trade-off. For example, ensembles of extremely randomized trees have been applied successfully in very complex high-dimensional tasks, such as image and sequence classification (Geurts et al., 2006).

In this work we explore the Perturb and Combine idea celebrated in supervised learning in the context of probability density estimation in high-dimensional spaces. We propose a new family of unsupervised learning methods of mixtures of large ensembles of randomly generated poly-trees. The specific feature of these methods is their scalability to very large numbers of variables and training instances. We explore various variants of these methods empirically on a set of discrete test problems of growing complexity.

2. Methods

2.1. Poly-Tree density models

Let $X = \{X_1, \ldots, X_n\}$ denote a finite set of discrete random variables.

A poly-tree model P for the density over X is defined by a directed acyclic graph which skeleton is acyclic and connected, and the set of vertices of which is in bijection with X and with a set of conditional densities $\mathbb{P}_P(X_i|pa_P(X_i))$, where $pa_P(X_i)$ denotes the set of variables in bijection with the parents of X_i in P. It represents graphically the density factorization

$$\mathbb{P}_P(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbb{P}_P(X_i|pa_P(X_i)). \qquad (1)$$

Poly-tree models can be used for probabilistic inference over $\mathbb{P}(X_1, \ldots, X_n)$ with a computational complexity linear in the number of variables n (Pearl, 1986).

One can define nested subclasses of poly-tree density models by imposing constraints on the maximum number p of parents of any node. In these subclasses, not only inference but also parameter learning is of linear complexity in the number of variables. The smallest such subclass is called the tree subspace, in which nodes have exactly one parent ($p = 1$).

2.2. Mixture models of poly-trees

A mixture model of m poly-tree models $(P_1, \ldots, P_m)$ is defined as a convex combination of the elementary poly-tree models, ie.

$$\mathbb{P}_M(X_1, \ldots, X_n) = \sum_{i=1}^{m} \mu_i \mathbb{P}_{P_i}(X_1, \ldots, X_n), \qquad (2)$$

where $\mu_i \in [0, 1]$ and $\sum_i^m \mu_i = 1$.

While single poly-tree models impose restrictions on the kind of densities they can faithfully represent, mixtures of poly-trees are universal approximators.

2.3. Learning a random mixture from data

Let X be a set of discrete random variables, and $D = (x^1, \cdots, x^d)$ be a sample of joint observations $x^i = (x^i_1, \cdots, x^i_n)$ drawn from some datagenerating distribution $\mathbb{P}_G(X)$. Let $\mathcal{P}$ be the space of all possible poly-tree graphical structures defined over X.

Our generic procedure for generating a random mixture of poly-tree models from D is described by Algorithm 1; it receives as inputs X, D, m, and three procedures *DrawPolytree*, *LearnPars*, *ComputeWeight*.

Algorithm 1 (Learning random poly-tree mixtures)

1. Repeat for $i = 1, \cdots, m$:
   (a) $P_i = \mathit{DrawPolytree}(\mathcal{P})$,
   (b) For $j = 1, \cdots, n$:
   $\mathbb{P}_{P_i}(X_j|pa_{P_i}(X_j)) = \mathit{LearnPars}(P_i, X_j, D)$
   (c) $\mu_i = \mathit{ComputeWeight}(P_i, D, m)$
2. Return $\left(\mu_i, (\mathbb{P}_{P_i}(X_j|pa_{P_i}(X_j)))^n_{j=1}\right)^m_{i=1}$.

3. Experiments and preliminary results

In (Ammar et al., 2008) we report some first results with the above algorithm applied to datasets of size $d = 1000$ generated from discrete distributions with $n = 8$, which could be faithfully represented by a chain, a single tree, or a single poly-tree model.

In these simulations we have considered two different instances of *DrawPolytree*, namely a uniform draw over the class $\mathcal{P}$ of all poly-trees, and a uniform draw over the subclass $\mathcal{P}^1$ of trees. In order to achieve this for $m \in \{1, 2, \ldots, 1000\}$, we have used efficient algorithms for sampling trees given in (Quiroz, 1989).

For parameter learning, we used maximum a posteriori values given the dataset and structure, while assuming non-informative priors on the parameters. Concerning the μ_is, we used a uniform weighting strategy, ie. $\mathit{ComputeWeight}(P_i, D, m) = 1/m$.

Overall, these results showed that the quality of the mixture-models converges rather rapidly (ie. for $m \approx 20$), and that the poly-tree mixtures were slightly superior when targeting poly-tree datagenerating distributions, while the mixtures of trees were superior in the other two cases. We also observed a slightly non-monotonic behavior of the model quality with growing values of m, which we suspect to be related to the uniform weighting scheme.

In the immediate future, we will carry out further more systematic experiments on larger problems and spanning different versions of the algorithm.

In particular, we will consider non-uniform weighting schemes, by exploiting the score obtained for a given structure and dataset so as to downweight structures that fit less well to the datagenerating distribution. We will also consider sampling from the spaces $\mathcal{P}^p$ of poly-trees of bounded number of parents.

Experiments will be made over a richer set of datagenerating distributions, in particular ones that can not be represented faithfully by a single poly-tree model. For instance, we will consider general directed acyclic graph models as datagenerating distributions.

We will compare our algorithm in terms of sample and computational efficiency with Bayesian network structure learning and algorithms targeting an *optimal* mixture of tree-models (Meila-Predoviciu, 1999).

Subsequently, we plan to extend our approach to handle continuous variables and incomplete datasets.

Acknowledgments

This work presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

References

Ammar, S., Leray, P., & Wehenkel, L. (2008). Estimation de densité par ensembles aléatoires de polyarbres. *Proceedings of JFRB*.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*, 3–42.

Meila-Predoviciu, M. (1999). *Learning with mixtures of trees*. Doctoral dissertation, MIT.

Naïm, P., Wuillemin, P.-H., Leray, P., Pourret, O., & Becker, A. (2007). *Réseaux bayésiens*. Paris: Eyrolles. 3 edition.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, *29*, 241–288.

Quiroz, A. (1989). Fast random generation of binary, t-ary and other types of trees. *Journal of Classification*, *6*, 223–231. available at http://ideas.repec.org/a/spr/jclass/v6y1989i1p223-231.html.

Evolutionary Learning of Feature Fusion Hierarchies

Fabien Scalzo FSCALZO@MEDNET.UCLA.EDU
Division of Neurosurgery, Geffen School of Medicine, University of California, Los Angeles, CA, USA

George Bebis BEBIS@CSE.UNR.EDU
Mircea Nicolescu MIRCEA@CSE.UNR.EDU
Leandro Loss LOSS@CSE.UNR.EDU
Computer Vision Lab. Department of Computer Science, University of Nevada, Reno, NV, USA

Abstract

We present a hierarchical feature fusion model for image classification that is constructed by an evolutionary learning algorithm. The model has the ability to combine local patches whose location, width and height are automatically determined during learning. The representational framework takes the form of a two-level hierarchy which combines feature fusion and decision fusion into a unified model. The structure of the hierarchy itself is constructed automatically during learning to produce optimal local feature combinations. A comparative evaluation of different classifiers is provided on a challenging gender classification image database. It demonstrates the effectiveness of these Feature Fusion Hierarchies (FFH).

1. Introduction

The generalization of new image acquisition devices and the development of new feature extractors have recently increased the interest of combining complementary modalities or features to perform automatic image classification. Hierarchical approaches (Singh et al., 2008; Podolak, 2008; Kim & Oh, 2008) to image classification are particularly interesting to solve complex problems because they are capable to decompose them into tasks that are often easier to tackle. However, these approaches often tend to manually define the structure of their hierarchy depending on the features involved (Tan & Triggs, 2007), and can only exploit a limited number of features. The current paper addresses these problems by presenting a framework that performs gender classification based on a large set of features extracted from facial images. The structure of the model as well as its parameters are estimated by a genetic learning algorithm that explores the space of possible hierarchies.

2. Feature Fusion Hierarchies

Feature Fusion Hierarchies (FFH) address the problem of fusing high-dimensional registered feature sets for image classification. The representational framework takes the form of a two-level hierarchy which combines local feature fusion and decision fusion into a unified model (Figure 1).

Given a feature set $I(x, y, f)$, where (x, y) denotes a position in the image, and f is a feature, the feature fusion level is defined as a set of compound features $\mathcal{C}$. Each compound feature $\mathcal{C}_i$ combines a subset of features $f_{\mathcal{C}_i} \in f$ over a local window $\theta_{\mathcal{C}_i}$. This fusion is done using a dimensionality reduction technique, denoted $\mathcal{R}_i(I_{f_{\mathcal{C}_i},\theta_{\mathcal{C}_i}})$, and learned in a supervised way (*e.g.* LDA). A key property of this function $\mathcal{R}_i$ is to operate locally in the sense that it exploits local adaptive windows (Scalzo & Piater, 2007) whose parameters $\theta_{\mathcal{C}_i} = \{x, y, Sx, Sy\}$ are automatically adjusted during learning (position in the image (x, y), width Sx and height Sy). The output of the function $S_i = \mathcal{R}_i(I_{f_{\mathcal{C}_i},\theta_{\mathcal{C}_i}})$ corresponds to a lower dimensionality response vector. An additional classifier is learned on the top of the first level to form the second level $\mathcal{D}$ corresponding to the decision fusion. Its input data correspond to the compound feature output $\{S_1, S_2, \ldots, S_n\}$ merged into a single vector S.

3. Learning of Fusion Hierarchies

A canonical genetic algorithm is used to explore the space of possible hierarchies (both the *structure* and the *parameters* are estimated). The optimal solution is the one that offers the best classification rate on the validation data and minimizes the number of features used as well as the size of the patches.

3.1. Genome Representation

Each evolving genome in the population is represented as a binary vector encoding the structure and the pa-

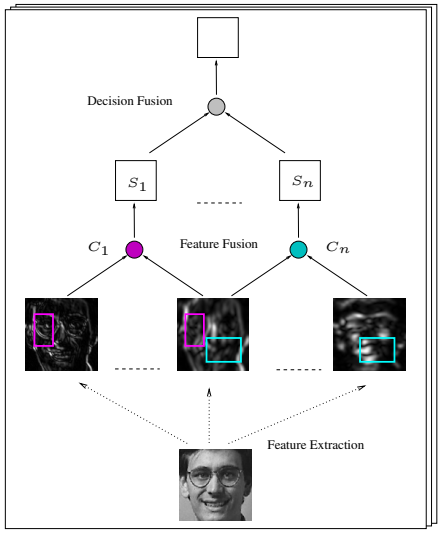


Figure 1. Overview of a Feature Fusion Hierarchy (FFH).

rameters of a specific hierarchy (Fig. 2). A genome defines the hierarchy as a set of N_c combinations $\mathcal{C}_i$. The *structure* of $\mathcal{C}_i$ corresponds to the subset of features that are combined $f_{\mathcal{C}_i} = \{f_1, \ldots, f_n\}$ whereas its *parameters* define the location (x, y) and size (Sx, Sy) of the local window in the image on which the fusion is performed.

Given n features at the first level, the structural part is represented as a n-length binary vector encoding the presence of the features in the combination. For the parameter part, variables $\{x, y, Sx, Sy\}$ are each represented as b bits vector.

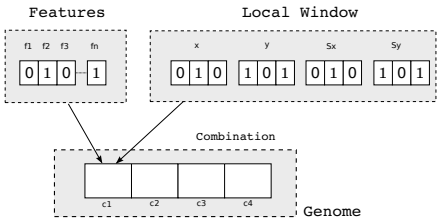


Figure 2. A Feature Fusion Hierarchy made of four compound features $(c1, c2, c3, c4)$ is encoded into a genome. The structural part and the parameters are embedded into a single binary vector.

3.2. Fitness, Crossover and Mutation

The fitness function $\text{fit}(h)$ is used to evaluate each individual in the population. It is set proportional to the classification rate r of the genome encoded hierarchy g, $\text{fit}(g) = r(g) + \alpha_1 n + \alpha_2 s^{-1}$, where n is the number of zeros in the structure part of the genome g and s is the total area covered by the patches. Parameters α_1 and α_2 are used respectively to support combinations that have a fewer number of features and are defined over a smaller window. A bi-parental random crossover and a single point mutation operator are used in our algorithm to produce new individuals.

4. Experiments

The effectiveness of the proposed framework is evaluated on a gender classification problem. Given a set of 400 facial images (Sun et al., 2002) captured under various conditions, the task is to correctly identify the gender of the subject present in the image. Each image is convolved with 35 Gabor filters and 5 Laplacian filters to produce the initial feature set on which our Feature Fusion Hierarchies (FFH) are constructed. The classification results after a three-fold cross-validation are reported in Table 1 for LDA, SVM and KSR classifiers. It can be observed that the use of the Feature Fusion Hierarchies (FFH) reduces significantly the classification error of a PCA-based framework and outperforms the results obtained by PCA-GA approach (Sun et al., 2002). This can be explained by the fact that our FFH approach exploits local features whereas PCA-GA computes the projections on the entire image.

	LDA	SVM	KSR
PCA	14.2%	8.9%	-%
GA-PCA(Sun et al., 2002)	9%	4.7%	-%
FFH (this paper)	**7.2%**	-%	**3.8**%

Table 1. Results for three different classifiers are reported for PCA, GA-PCA (Sun et al., 2002) and the Feature Fusion Hierarchies (FFH).

References

Kim, Y., & Oh, I. (2008). Classifier ensemble selection using hybrid genetic algorithms. *Pattern Recogn. Lett.*, *29*, 796–802.

Podolak, I. T. (2008). Hierarchical classifier with overlapping class groups. *Expert Syst. Appl.*, *34*, 673–682.

Scalzo, F., & Piater, J. (2007). Adaptive patch features for object class recognition with learned hierarchical models. *Beyond Patches*.

Singh, R., Vatsa, M., & Noore, A. (2008). Hierarchical fusion of multi-spectral face images for improved recognition performance. *Information Fusion*, *9*, 200–210.

Sun, Z., Bebis, G., Yuan, X., & Louis, S. J. (2002). Genetic Feature Subset Selection for Gender Classication: A Comparison Study. *IEEE International Conference on Image Processing*.

Tan, X., & Triggs, B. (2007). Fusing gabor and lbp feature sets for kernel-based face recognition. *Analysis and Modeling of Faces and Gestures* (pp. 235–249).

Using decision trees to build an event recognition framework for automated visual surveillance

Cedric Simon CEDRIC.SIMON@UCLOUVAIN.BE
Christophe De Vleeschouwer DEVLEESCHOUWER@UCLOUVAIN.BE
Communication and Remote Sensing Lab, UCL, Louvain-La-Neuve, Belgium

Jerome Meessen JEROME.MEESSEN@MULTITEL.BE
Multitel, Mons, Belgium

Abstract

This paper presents a classifier-based approach to recognize possibly sophisticated events in video surveillance. The aim of this work is to propose a flexible and generic event recognition system that can be used in a real world context. Our system uses the ensemble of randomized trees procedure to model each event as a sequence of structured activity patterns, without using any tracking method. Experimental results demonstrate the robustness of the system toward artifacts and passer-by, and the effectiveness of its framework for event recognition applications in visual surveillance.

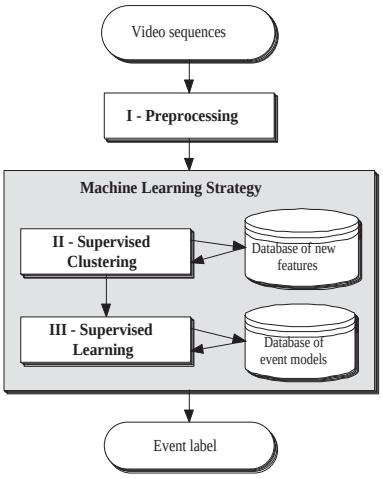


Figure 1. Overview of the system

1. Overview

The growing number of cameras in public and private areas increases the interest of the image processing community in automated visual surveillance system. Nonetheless, there is still a broad gap between what automated systems offer and the actual needs of the industry (Dee & Velastin, 2007). The system we propose aims at reducing this gap, by using a coherent framework that links local and coarse features with meaningful concepts. Those local features are attributes representing the moving objects (the blobs) in the scene, at each frame. No tracking procedure nor intermediate reasoning (eg. occlusions) is needed which leads in a greater genericity. The framework we propose (figure 1) relies on two main parts that are both based on the ensemble of randomized trees concept (Geurts et al., 2006).

In the first stage (section 2), sophisticated features are built by clustering the blobs according to their feature's values at each frame. Those clusters are tagged, and each tag is then associated to the appropriate frames in the video sequences. In the second stage (section 3), the events are classified, based on the temporal distribution of those tags in the sequences. The main idea is to discriminate event classes by investigating the temporal relationships between the tags, for example by asking if there is a blob of one type before a blob of another type. This framework is inspired from Geman works in (Amit & Geman, 1997).

2. Construction of elaborated features and definition of tags

From the blob's features, we adopt an information-theoretic approach and cluster similar blobs by using the decision tree methodology. Two sets of *clustering trees* are built according to the type of attributes used by the trees:

- In the first pool, attributes are based on blob's features (i.e its position, size and velocity)
- In the second pool, attributes are based on pair of blob's features (i.e the distance between the blobs and their relative velocity). For efficiency, from all pairs of blobs in each frame, we keep only pairs having a relatively short distance between the blobs

At each node of one tree, a question is selected by choosing one attribute, i.e by picking one feature and one threshold that reduce as much as possible the class entropy within the node. Thereby, each blob (or pair of blobs) inherit the event class of the whole sequence it belongs to.

Once the trees are built, we tag each node of the trees, except for the root node. Hence, by definition, each tag correspond to a specific combination of coarse and local features, that characterize each frame of the video sequences. Diversity of the tags can then be boosted by increasing the number of the clustering trees N_{CT}, while more specificity is obtained by increasing the depth of the trees. For instance, we observed in our experiments that the system achieved the best performance for $N_{CT} \approx 10$. This supervised clustering process allows each tag to be more discriminant regarding the event classes, while using only local information (each frame individually).

3. Modeling the spatio-temporal events with randomized trees

Once the elaborated features are computed, we use another set of randomized trees to model and classify the video sequences, based on the spatio-temporal arrangement of those advanced features. This is done by defining the two output branches of a tree node based on the presence/absence of a specific temporal arrangement A of tags in the video sequence. In order for these arrangements to be scale invariant (and better fit the notion of visual surveillance event), coarse binary relations like "before", "after" and "at the same time" are used. An example of arrangement would then be: tag x exists "before" tag y which exists "at the same time" as tag z.

At the root node N_r a tag T_x from the full set $\mathcal{T}$ of tags is chosen. The Question (Q_0) is *"Does T_x exist in the training sequences?"*. Those training samples for which $Q_{N_r} = 0$ are in the "no" child node and we search again through T. Those samples for which $Q_{N_r} = 1$ are in the "yes" child node and we then put T_x among the participating tags $\mathcal{T}_p$. Further question can be:

$$\begin{cases} \exists T_i ? \\ \exists T_i \star T_j ? \\ \exists T_{j_x} \star T_{j_y} ? \end{cases} with \begin{cases} T_i \in \mathcal{T} \\ T_j, T_{j_x}, T_{j_y} \in \mathcal{T}_p \\ \star \in [\prec, |, \succ] \end{cases} \quad (1)$$

with $\prec$ meaning 'before', | 'during' and $\succ$ 'after'.

4. Experiments and perspective

The described system was tested on two scenarios. The first one used simulated video surveillance events, while the second is a real case scenario, which occurs in the entrance lobby of a public company. More information about those datasets can be found on the web [1]. Results are encouraging our framework, even if passer-by or by-stander are significantly decreasing the accuracy.

Further improvement could then analyze how to use histograms of similarity between the blobs to enrich our elaborated features and gain some robustness, along with features characterizing each frame more globally (i.e. the number of blobs in the frame, the density of the moving objects in each frame, the average/variance of each features..). We will also focus on using an active learning procedure in order to interact with the user, and to be able to enhance the system on line.

References

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, *9*, 1545–1588.

Dee, H., & Velastin, S. (2007). How close are we to solving the problem of automated visual surveillance? a review of real-world surveillance, scientific progress and evaluative mechanisms. *Machine Vision and Applications, Special Issue on Video Surveillance Research in Industry and Academic Springer*.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*, 3–42.

[1] `http://www.tele.ucl.ac.be/~csimon/trajectories.html`.

Content-based Image Retrieval by Indexing Random Subwindows with Randomized Trees

Raphaël Marée RAPHAEL.MAREE@ULG.AC.BE
GIGA Bioinformatics, University of Liège, GIGA Tower B34 (+1), Avenue de l'Hopital, 1, 4000 Liege, Belgium

Pierre Geurts P.GEURTS@ULG.AC.BE
Louis Wehenkel L.WEHENKEL@ULG.AC.BE
Systems and Modeling, Department of Electrical Engineering and Computer Science & GIGA Research, University of Liège, Institut Montefiore B28, Grande Traverse 10, 4000 Liège, Belgium

Abstract

We propose a method for content-based image retrieval which exploits the similarity measure and indexing structure of totally randomized tree ensembles induced from a set of subwindows randomly extracted from a sample of images. The approach is quantitatively evaluated on various types of images with good results despite its conceptual simplicity and computational efficiency.

1. Introduction

In content-based image retrieval (CBIR), users want to retrieve images that share some similar visual elements with a query image, without any further text description neither for images in the reference database, nor for the query image (see Figure 1). To be practically valuable, a CBIR method should combine computer vision techniques that derive rich image descriptions, and efficient indexing structures.

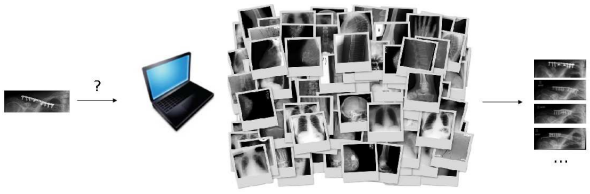


Figure 1. Illustration of the goal of a CBIR system for one query image (left) from the IRMA-2005 dataset.

Our method for CBIR is an extension of the method we proposed for image classification (Marée et al., 2005) and was published in (Marée et al., 2007) where more formal notations and experiment details are given.

2. Method

The different steps of our algorithm are now described.

2.1. Extraction of random subwindows

Square patches of random sizes are extracted at random locations in images, resized by bilinear interpolation to a fixed-size (16×16), and described by HSV values (resulting into 768 feature vectors) for color images, or gray intensities (resulting into 256 feature vectors) for graylevel images. This provides a rich representation of images corresponding to various overlapping regions, both local and global, whatever the task and content of images. Using raw pixel values as descriptors avoids discarding potentially useful information while being generic, and fast.

2.2. Indexing subwindows with totally randomized trees

We use ensembles of totally randomized trees (Geurts et al., 2006) for indexing extracted random subwindows. The method recursively partitions the training sample of subwindows by randomly generated tests. Each test is chosen by selecting a random pixel component and a random cut-point in the range of variation of the pixel component in the subset of subwindows associated to the node to split. The development of a node is stopped as soon as either all descriptors are constant in the leaf or the number of subwindows in the leaf is smaller than a predefined number $n_{\min}$. A number T of such trees are grown from the training sample. To be able to later perform image retrieval, at each leaf of each tree, we record for each image of the reference set that appears in the leaf the number of its subwindows that have reached this leaf.

2.3. Inducing image similarities from tree ensembles

We first defined a similarity measure between any two subwindows by considering that two subwindows are *very similar* if they fall in a same leaf that has a *very small* subset of training subwindows. Then, we defined the similarity induced by an *ensemble* of T trees by considering that two subwindows are similar if they are considered similar by a large proportion of the trees. Given this similarity measure between subwindows, we derived a similarity between two images that is the average similarity between all pairs of their subwindows.

2.4. Image retrieval

The similarities between the query image and all reference images could then computed by propagating into each tree all subwindows from the query image, and by incrementing, for each of its subwindow, each tree, and each reference image, the similarity between the query and reference images. It is then possible to rank the reference images according to their similarity to the query image, as illustrated by Figure 2.

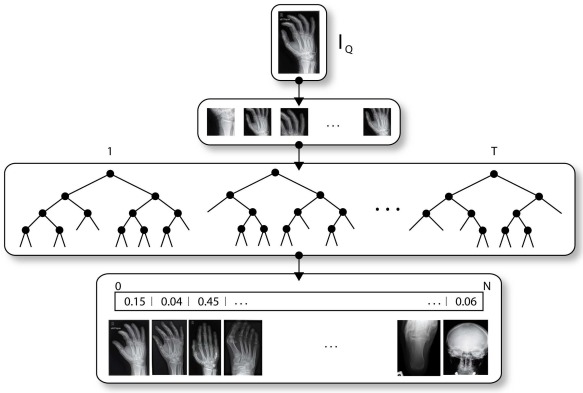


Figure 2. Ranking of the reference images according to their similarity to the query image.

3. Results

We performed a quantitative evaluation of our method in terms of its retrieval accuracy on various image datasets with ground-truth labels. These databases contains images representing various buildings (ZuBuD), objects and scenes (META and UkBench), and radiographs (IRMA). In our experiments, we consider that an image is relevant to a query if it is of the same class as the query image, and irrelevant otherwise. Note that, while using class labels to assess accuracy, this information is not used during the indexing phase. Results are shown in Table 1.

Dataset	#images ls/ts	Accuracy
ZuBuD	1005/115	96.52%
IRMA-2005	9000/1000	85.4%
UkBench	10200/10200	75.25%
META/UkBench	205763/10200	66.74%

Table 1. Image retrieval accuracy results.

4. Discussion

This method has nice properties that we will explore in future work for large-scale, distributed content-based image retrieval: The possibility of updating the model as new images come in, the capability of comparing new images using a model previously constructed from a different set of images, and its computational efficiency (due to randomization used both in image description and indexing) while keeping overall good performances. Finally, let us note that our image similarity measure actually defines a kernel and it could thus be exploited in clustering or supervised learning with kernel methods.

5. Acknowledgements

Raphaël Marée is supported by the GIGA (University of Liège) with the help of the Walloon Region and the European Regional Development Fund. PG is a research associate of the FNRS, Belgium. This work presents research results of the Belgian Network BIOMAGNET, funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The authors thank Vincent Botta for Figure 2.

References

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *36*, 3–42.

Marée, R., Geurts, P., Piater, J., & Wehenkel, L. (2005). Random subwindows for robust image classification. *Proc. IEEE CVPR* (pp. 34–40).

Marée, R., Geurts, P., & Wehenkel, L. (2007). Content-based image retrieval by indexing random subwindows with randomized trees. *Proc. 8th Asian Conference on Computer Vision (ACCV), LNCS* (pp. 611–620). Springer-Verlag.

IGForest
From tree to forest

Herman Stehouwer J.H.STEHOUWER@UVT.NL
ILK Research Group, Faculty of Humanities Tilburg University

Abstract

TiMBL is an implementation of K-nn containing several algorithms for classification. One of the implemented, and often used, classifiers is IGTree. IGTree is a fast trie-based approximation of k-nn. Because of its trie-based nature IGTree can mismatch on a feature quite fast, resulting in sub-optimal classification compared to IB1. We present an early version of IGForest that tries to lessen the impact of this problem while retaining the behavior of this set of algorithms. IGForest consists of an ensemble of IGTrees. The performance of IGForest is compared to both IGTree and IB1 on a diverse set of NLP problems.

1. Introduction

IGTree is a fast approximation of k-nearest neighbor classification (Daelemans et al., 1997). IGTree allows for fast training and testing even with millions of examples. IGTree compresses a set of labeled examples into a decision tree structure similar to the classic C4.5 algorithm (Quinlan, 1993), except that throughout one level in the IGTree decision tree, the same feature is tested. Classification in IGTree is a simple procedure in which the decision tree is traversed from the root node down, and one path is followed that matches the actual values of the new example to be classified. If an end node is met, the outcome stored at the end node is generated as classification. If the last visited node is a non-ending node, but no outgoing arcs match with the value in the new example, the most likely outcome stored at that node is produced as the resulting classification.

IGTree is typically able to compress a large example set into a lean decision tree with high compression factors, in reasonably short time, comparable to other compression algorithms. More importantly, IGTree's classification time depends only on the number of features ($O(f)$). Indeed, we observe high compression rates: trained on almost 2 million examples of a Dutch d/dt inflection task containing both part-of-speech and full-word features, IGTree builds a tree containing a mere 46,466 nodes, with which it can classify 17 thousand examples per second on a current computing server.

Well known techniques for boosting classifier performance are ensemble creation methods such as bagging and boosting. In (Banfield et al., 2007) a comparison is given of several such combination techniques on different training sets. They compare randomised C4.5, random subspaces, random forests, AdaBoost.M1W, and bagging. Most of these training sets where taken from the UCI repository (Murphy & Aha, 1995). Their main conclusion is that *"...for any given data set the statistically significantly better algorithms are likely to be more accurate, just not by a significant amount on that data set."*.

We frequently see that on large datasets, such as that the CoNLL 2000 shared task (Tjong Kim Sang & Buchholz, 2000), that IB1 easily outperforms IGTree in terms of accuracy, at the cost of classification time as can be clearly seen in Table 1. As the speed difference increases non-lineairly with the amount of training data, it would be prohibitive to run IB1 on experiments with millions of training instances.

The algorithm proposed here tries to find the middle ground between IB1 and IGTree by creating a forest of semi-random IGTrees. This is motivated by the fact that we wish to retain some of the unique characteristics of IGTree, such as its resemblance to a basic language model in smoothing (Zavrel & Daelemans, 1997), while still being a generic classifier supporting any number and type of features.

2. System Architecture

IGForest consists of a number of simple steps that generate an ensemble of IGTrees, which is than applied to

	F-score	Time
IB1	0.926	866
IGTree	0.868	1
IGForest	0.906	219

Table 1. F-score and total training and testing time in seconds on the CoNLL 2000 shared task (Tjong Kim Sang & Buchholz, 2000) by IB1, IGTree and IGForest using IB1 as an arbiter. 100.000 training instances where used.

the data which is to be classified. These steps are discussed briefly below. As should become clear IGForest is easy to parallelize by running IGTRee training and tagging in parallel. IGForest implements the following steps, that combined result in an ensemble classifier

1: Analyse Data The first step is the analysis of the incoming training instances. This includes generating a default set of feature weights (Gain Ratio) and determining the performance on the training instances of a simple most-frequent outcome baseline.

2: Generate IGTrees In the second step a fixed number of IGTree feature orders is generated semi-randomly. We use the default set of feature weights as a bias for the random order, i.e. if one feature has a default weight of half the total of the default weights it has a 50% chance of being picked first.

These feature orders are filtered to ensure uniqueness and improved performance on the training instances using cross-validation. It is ensured that each ordering improves on the most-frequent outcome baseline. The forest will always contain the default IGTree.

3: Train Combination Method IGForest contains several combination methods. Most of these combination methods need some training.

The simplest of these is linear unweighted voting, which does not need training and simply takes a vote of the different trees.

Somewhat more complex is linear weighted voting, where the weight of the vote of each tree is determined by the performance on the training instances using cross-validation.

The final combination method is that of an arbiter which trains on a holdout part of the training instances. Any classifier can be used as an arbiter. We use either IGTree or IB1.

4: Apply Finally, all the separate IGTrees are applied to the unclassified instances, after which the combination method is applied resulting in classified instances.

3. Preliminary Results

We have implemented a basic version of IGForest, which does not check for uniqueness of the trees, nor does it cull the trees that perform worse than the most-frequent outcome baseline. Nevertheless, the first results are promising. The results for the CoNLL 2000 task for IGForest using IB1 as an arbiter can be found in Table 1.

This basic version performs in between IB1 and IGTree on the CoNLL 2000 data. We observe similar numbers in other experiments, such as pos-tagging on the Wallstreet Journal. In different experiments, where IGTree already outperforms IB1, we do not see such an improvement, such as *-d/-dt* inflection in Dutch. IGTree outperforms IB1 in cases where the feature order is much more absolute than the feature weighting would suggest.

References

Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2007). A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Daelemans, W., Van den Bosch, A., & Weijters, A. (1997). IGTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, *11*, 407–423.

Murphy, P., & Aha, D. W. (1995). Uci repository of machine learning databases – a machine-readable repository. Maintained at the Department of Information and Computer Science, University of California, Irvine. Anonymous ftp from `ics.uci.edu` in the directory `pub/machine-learning/databases`.

Quinlan, J. (1993). C4.5*: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Tjong Kim Sang, E., & Buchholz, S. (2000). Introduction to the CoNLL-2000 shared task: Chunking. *Proceedings of CoNLL-2000 and LLL-2000* (pp. 127–132).

Zavrel, J., & Daelemans, W. (1997). Memory-based learning: Using similarity for smoothing. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (pp. 436–443).

Fast Image Annotation with Random Subwindows and Multiple Output Randomized Trees

Marie Dumont M.Dumont@ulg.ac.be
Systems and Modeling, Department of Electrical Engineering and Computer Science & GIGA Research, University of Liège, Institut Montefiore B28, Grande Traverse 10, 4000 Liège, Belgium

Raphaël Marée Raphael.Maree@ulg.ac.be
GIGA Bioinformatics, University of Liège, GIGA Tower B34 (+1), Avenue de l'Hopital, 1, 4000 Liege, Belgium

Pierre Geurts P.Geurts@ulg.ac.be
Louis Wehenkel L.Wehenkel@ulg.ac.be
Systems and Modeling, Department of Electrical Engineering and Computer Science & GIGA Research, University of Liège, Institut Montefiore B28, Grande Traverse 10, 4000 Liège, Belgium

Abstract

This work addresses image annotation, i.e. labelling pixels of an image with a class among a finite set of predefined classes. The proposed method extracts a sample of subwindows from a set of annotated training images to train a subwindow annotation model by tree-based ensemble methods. In one variant of the approach, the classifier is train to label only the central pixel of the subwindow, while in a second variant the classifier is extended to annotate all subwindow pixels simultaneously. This approach is evaluated on four different problems, where its overall good performance and efficiency are highlighted.

1. Image annotation

In this work, we propose a supervised learning approach for the generic problem of image annotation. Given a training set of images with pixel-wise labelling (ie. every pixel is hand-labelled with one class among a finite set of predefined classes), the goal is to build a model that will be able to predict accurately the class of every pixel of any new, unseen image.

2. Methods

To tackle this problem, our starting point is the results obtained by (Marée et al., 2005) for image classification. Their method first *randomly* extracts a large set of image subwindows (or patches) and describes those by high-dimensional feature vectors composed by *raw pixel values*. Then, the method uses an *ensemble of extremely randomized decision trees* (Geurts et al., 2006a) to build a subwindow classification model. To predict the class of a new image, the method extracts random subwindows from this image, classifies these subwindows using the decision tree ensemble and then aggregates the subwindow classifications by majority voting to get a single class prediction for the whole image.

In the context of image annotation, we follow a similar scheme based on the extraction of random subwindows and the use of ensemble of randomized trees. We propose two approaches.

The first approach builds extremely randomized trees to predict the class of the central pixel of subwindows. To classify a pixel of an unseen image, a subwindow centered on that pixel is extracted and the pixel classifications obtained by applying the trees on the subwindow are aggregated by majority voting. We denote this method SCM for Subwindow Classification Model.

The second approach extends the classifier so as to predict the class of every subwindow pixels simultaneously. We propose to replace the score function used to evaluate and select splits during tree induction by the average over all output pixel classes of the information theoretic score used for standard classification problems. Once the model is built, to annotate a new image, several (overlapping) subwindows are randomly extracted from the image, a prediction is computed by the classifier for every subwindow pixel and the final annotation of the image is obtained by taking for every

image pixel the majority class among all class predictions that were obtained for this pixel. We call this method SCMMO for Subwindow Classification Model with Multiple Outputs.

3. Experiments

To assess the performance and usefulness of the proposed methods as a foundation for image annotation, we have evaluated them on four datasets representing various types of images (microscope imaging, photographs of natural scenes, etc.) and a large variety of classes. Table 1 reports results with both methods in terms of pixel misclassification error rate on these four problems. Table 2 reports computing times. Examples of annotated images are shown in Figure 1. Detailed results and description of the datasets can be found in (Dumont et al., 2008).

SCMMO is almost always significantly better than SCM, which was expected as SCM predictions do not exploit correlation between pixels. On three problems among the four, SCMMO results are in fact comparable to those obtained by the state of the art. Moreover, our methods are very attractive in terms of computing times.

Table 1. Error rates obtained on four datasets.

Database	SCM	SCMMO
Retina	7.53%	7.56%
Bronchial	3.42%	3.13%
Corel	49.43%	36.01%
Sowerby	14.96%	10.93%

Table 2. Computing times on two datasets.

Dataset	Sowerby		Retina	
Method	Training	Prediction	Training	Prediction
SCM	76.63 s	0.12 s	41.22 s	17.64 s
SCMMO	315.68 s	0.26 s	125.5 s	26.98 s

4. Conclusion

We have introduced and compared two generic image annotation methods: SCM and SCMMO. From our experiments on four distinct databases, SCMMO appears to be the most interesting one among the two proposed methods: it is almost always significantly better than SCM and obtains state-of-the-art results on three problems among four. We deem that the main merit of our approach is its good overall performance while remaining conceptually very simple and keeping computing times very low. Moreover, it might be exploited as a first, fast step for image annotation as its output predictions might be post-processed by various techniques at the local and/or global level.

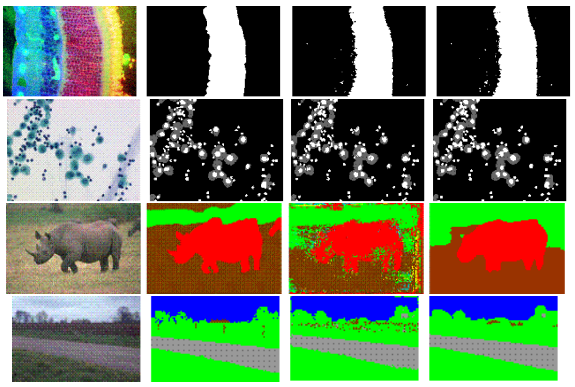


Figure 1. From left to right: original image, manual annotation, annotation obtained with SCM, annotation obtained with SCMMO. From top to bottom: images from Retina, Bronchial, Corel, and Sowerby datasets.

Acknowledgments

MD is a research fellow of the FNRS, Belgium. RM is supported by the GIGA interdisciplinary cluster of Genoproteomics of the University of Liège with the help of the Walloon Region and the European Regional Development Fund. PG is a research associate of the FNRS, Belgium.

References

Dumont, M., Marée, R., Geurts, P., & Wehenkel, L. (2008). Fast image annotation with random subwindows and multiple output randomized trees. *Submitted.*

Geurts, P., Ernst, D., & Wehenkel, L. (2006a). Extremely randomized trees. *Machine Learning*, *36*, 3–42.

Geurts, P., Wehenkel, L., & d Alché-Buc, F. (2006b). Kernelizing the output of tree-based methods. *ICML* (pp. 345–352).

Marée, R., Geurts, P., Piater, J., & Wehenkel, L. (2005). Random subwindows for robust image classification. *CVPR* (pp. 34–40).

On the use of Machine Learning in Statistical Parametric Speech Synthesis

Thomas Drugman
Alexis Moinet
Thierry Dutoit FIRSTNAME.NAME@FPMS.AC.BE
Faculté Polytechnique de Mons, TCTS Lab, 31 Boulevard Dolez, 7000 Mons, Belgium

Abstract

Statistical parametric speech synthesis has recently shown its ability to produce natural sounding speech while keeping a certain flexibility for voice transformation without requiring a huge amount of data. This abstract presents how machine learning techniques such as Hidden Markov Models in generation mode or context oriented clustering with decision trees are applied in speech synthesis. Fields that are investigated in our laboratory to improve this method are also discussed.

1. HMM-based Speech Synthesis

Before the last five years, synthetic speech was typically produced by concatenating frames of natural speech selected from a huge database, possibly applying signal processing to them so as to smooth discontinuities. In 2002, Tokuda et al. (K. Tokuda, 2002) proposed a system relying on HMM generation of speech parameters. Compared to the previous one, this approach has the advantage to allow voice transformation without requiring a large amount of data, merely by adapting its statistics through a short training (A. W. Black & Tokuda, 2007). By voice transformation we here mean voice conversion towards a given target speaker or expressive/emotive speech production from the initial trained system.

The key idea of a HMM-based synthesizer is to generate sequences of speech parameters directly from the trained HMMs. Next subsections describe the two main steps in the bloc diagram of such a synthesizer (see Figure 1).

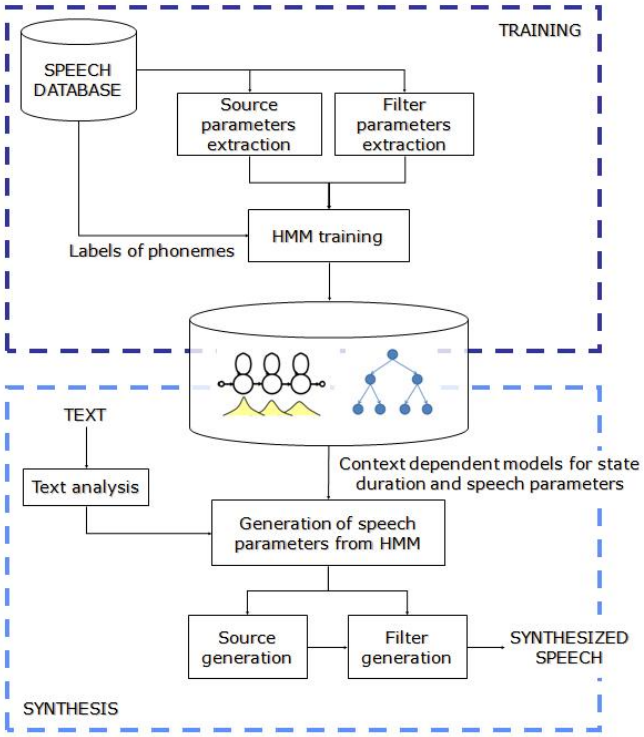


Figure 1. Bloc diagram of a HMM-based speech synthesizer

1.1. The training part

Training our system assumes that a large segmented speech database is available. Labels consist of phonetic environment description. First, speech waveforms are decomposed into their source (glottal) and filter (vocal tract) components. Representative features are then extracted from both contributions. Since source modeling is composed either of continuous values or a discrete symbol (respectively during voiced and unvoiced regions), multi-space probability density HMMs have been proposed. Indeed this approach turns out to be

able to model sequences of observations having a variable dimensionality.

Given these latter parameters and the labels, HMMs are trained using the Viterbi and Baum-Welch re-estimation algorithms. Till that point this may seem very close to building a speech recognizer. Nevertheless decision tree-based context clustering is here used to statistically model data appearing in similar contextual situations. Indeed contextual factors such as stress-related, locational, syntaxical or phone identity factors affect prosody (duration and source excitation characteristics) as well as spectrum. More precisely an exhaustive list of possible contextual questions is first drawn up. Decision trees are then built for source, spectrum and duration independently (as factors have a different impact on them) using a maximum likelihood criterion. Probability densities for each tree leaf are finally approximated by a Gaussian mixture model.

1.2. The synthesis part

The text typed by the user is first converted into a sequence of contextual labels. From them, a path through context-dependent HMMs is computed using the duration decision tree. Source and spectrum parameters are then generated by maximizing the output probability. The incorporation of dynamic features makes the coefficients evolution more realistic and smooth. Speech is finally synthesized from the generated parameters by an operation of signal processing.

2. Our ongoing research activities

Our main goal is to develop an efficient HMM-based speech synthesizer for French. For this, the ACAPELA group kindly provided us with their natural language processor. Since English and French have both their own phonological particularities, an adaptation of the questions used for the context oriented clustering was necessary.

Basically our (ongoing) research activities focus on three main issues:

- **Speech analysis**: A major disadvantage of such a synthesizer is the "buzziness" of the produced speech. This is typically due to the parametrical representation of speech. To overcome this hindrance a particular interest is devoted to speech analysis. Our approach particularly investigates a method of source-filter deconvolution based on the zeros of the Z-transform (B. Bozkurt & Dutoit, 2007). By this way an estimation of the glottal signal and the vocal tract impulse response is achieved. Different models for the source (LF and CALM models) as well as for the spectrum (MLSA, LSP or MFCC coefficients) are tested and their perceptual quality is assessed.

- **Intelligibility enhancement**: In some applications speech has to be synthesized in adverse conditions (in cars, at the station,...). Intelligibility consequently becomes of a paramount importance (Langner & Black, 2005). If we can model the modifications occurring when speech is produced in noise (possibly implying a training), a synthesizer with (adaptive) intelligibility enhancement could be carried out.

- **Voice conversion** : In voice conversion (Y. Stylianou & Moulines, 1995) it is aimed at modifying the source speaker's voice towards a particular target speaker given a limited dataset of his utterances. This approach implies the study of the statistical learning transforming representation spaces of both speakers. This could allow us to easily generate new voices, including the production of more emotions and expressivity in speech.

Acknowledgments

Thomas Drugman is supported by the "Fonds National de la Recherche Scientifique". The authors also would like to thank the ACAPELA group and the Walloon Region (grant #415911) for their support.

References

A. W. Black, H. Z., & Tokuda, K. (2007). Statistical parametric speech synthesis. *Proc. of ICASSP* (pp. 1229–1232).

B. Bozkurt, L. C., & Dutoit, T. (2007). Chirp group delay analysis of speech signals. *Speech Comm., 49, issue 3*, 159–176.

K. Tokuda, H. Zen, A. W. B. (2002). An hmm-based speech synthesis system applied to english. *Proc. of IEEE SSW.*

Langner, B., & Black, A. W. (2005). Improving the understandability of speech synthesis by modeling speech in noise. *Proc. ICASSP.*

Y. Stylianou, O. C., & Moulines, E. (1995). Statistical methods for voice quality transformation. *Proc. EUROSPEECH.*

Towards robust feature selection techniques

Yvan Saeys YVAN.SAEYS@PSB.UGENT.BE
Thomas Abeel THOMAS.ABEEL@PSB.UGENT.BE
Yves Van de Peer YVES.VANDEPEER@PSB.UGENT.BE
Department of Plant Systems Biology, VIB, Technologiepark 927, 9052 Gent, Belgium,
Department of Molecular Genetics, Ghent University, Technologiepark 927, 9052 Gent, Belgium

Abstract

Robustness of feature selection techniques is a topic of recent interest, especially in high dimensional domains with small sample sizes, where selected feature subsets are subsequently analysed by domain experts to gain more insight into the problem modelled. In this work, we investigate the robustness of various feature selection techniques, and provide a general scheme to improve robustness using ensemble feature selection. We show that ensemble feature selection techniques show great promise for small sample domains, and provide more robust feature subsets than a single feature selection technique. In addition, we also investigate the effect of ensemble feature selection techniques on classification performance, giving rise to a new model selection strategy.

1. Introduction

During the past decade, the use of feature selection for knowledge discovery has become increasingly important in many domains that are characterized by a large number of features, but a small number of samples. Typical examples of such domains include text mining, computational chemistry and the bioinformatics and biomedical field, where the number of features (problem dimensionality) often exceeds the number of samples by orders of magnitude (Saeys et al., 2007). When using feature selection in these domains, not only model performance but also robustness of the feature selection process is important, as domain experts would prefer a stable feature selection algorithm over an unstable one when only small changes are made to the dataset.

Surprisingly, the robustness (stability) of feature selection techniques is an important aspect that received only relatively little attention during the past. Recent works in this area mainly focus on the stability indices to be used for feature selection, introducing measures based on Hamming distance (Dunne et al., 2002), correlation coefficients (Kalousis et al., 2007), consistency (Kuncheva, 2007) and information theory (Krízek et al., 2007). The work of Kalousis et al. (2007) also presents an extensive comparative evaluation of feature selection stability over a number of high-dimensional datasets. However, most of these recent works only focus on the stability of single feature selection techniques.

In this work, we investigate whether the use of ensemble feature selection techniques can be used to yield more robust feature selection techniques, and whether combining multiple methods has any effect on the classification performance.

2. Methods

2.1. Quantification of robustness

Depending on the outcome of a feature selection technique, the result can be either a set of weights, a ranking, or a particular feature subset. In order to assess robustness, a subsampling scheme is used that generates k subsamples containing 90% of the original data. The robustness of a technique is then measured by the average over all pairwise similarity comparisons between the different feature selectors:

$$S_{\text{tot}} = \frac{2\sum_{i=1}^{k}\sum_{j=i+1}^{k} S(\mathbf{f}_i, \mathbf{f}_j)}{k(k-1)}$$

where $\mathbf{f}_i$ represents the outcome of the feature selection method applied to subsample i ($1 \leq i \leq k$), and $S(\mathbf{f}_i, \mathbf{f}_j)$ represents a similarity measure between $\mathbf{f}_i$ and $\mathbf{f}_j$.
Here, we focus on similarities between rankings - using the Spearman rank correlation coefficient - and subsets, using the Jaccard index (Kalousis et al., 2007) or

Table 1. Robustness of the different feature selectors across the different datasets. Spearman correlation coefficient (Sp), Jacard index (JC) and consistency index (CI) on a subset of 1% best features.

Dataset		SU		Relief		SVM_RFE		RF	
		Single	Ensemble	Single	Ensemble	Single	Ensemble	Single	Ensemble
Colon	Sp	0.61	0.76	0.62	0.85	0.7	0.81	0.91	0.99
	JC	0.3	0.55	0.45	0.56	0.44	0.5	0.01	0.64
	CI	0.45	0.7	0.61	0.71	0.6	0.65	0.01	0.77
Leukemia	Sp	0.68	0.76	0.58	0.79	0.73	0.79	0.97	0.99
	JC	0.54	0.6	0.44	0.55	0.49	0.57	0.36	0.8
	CI	0.7	0.74	0.6	0.71	0.64	0.72	0.53	0.89
Lymphoma	Sp	0.59	0.74	0.49	0.76	0.77	0.81	0.96	0.99
	JC	0.37	0.55	0.42	0.56	0.43	0.46	0.22	0.73
	CI	0.53	0.7	0.58	0.71	0.6	0.63	0.35	0.84
Average	Sp	0.63	0.75	0.56	0.8	0.73	0.80	0.95	0.99
	JC	0.40	0.57	0.44	0.56	0.45	0.51	0.2	0.72
	CI	0.56	0.71	0.6	0.71	0.61	0.67	0.3	0.83

the consistency index (Kuncheva, 2007).

2.2. Ensemble feature selection

In order to improve the robustness of feature selection, a similar idea as in ensemble learning can be used, where multiple classifiers are combined in order to improve performance. In this work, we construct an ensemble of feature selectors by bootstrapping the data, and creating a *consensus feature selector* that aggregates the results of the single feature selectors by rank summation.

3. Results

Table 1 shows the robustness of a representative sample of feature selectors, including two filter based approaches (Symmetrical Uncertainty (SU) and Relief) and two embedded approaches (recursive feature elimination using a SVM (SVM_RFE) and Random Forests (RF)). It can be observed that constructing an ensemble version of each feature selector significantly improves robustness.

However, considering only robustness of a feature selection technique is not an appropriate strategy to find good feature rankings or subsets, and also model performance should be taken into account to decide which features to select. Therefore, feature selection needs to be combined with a classification model in order to get an estimate of the performance of the feature selector-classifier combination.

Our results show that in most cases, classification performance using ensemble feature selection is comparable to the performance using conventional feature selection techniques, or performs only slightly worse (data not shown). It turns out that the best trade-off between robustness and classification performance depends on the dataset at hand, giving rise to a new model selection strategy, incorporating both classification performance as well as robustness in the evaluation strategy by taking e.g. the harmonic mean of robustness and classification performance as a combined measure.

Acknowledgments

Y.S. is funded by a post-doctoral grant from the Research Foundation Flanders (FWO-Vlaanderen). T.A. is funded by a grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

Dunne, K., Cunningham, P., & Azuaje, F. (2002). *Solutions to instability problems with sequential wrapper-based approaches to feature selection* (Technical Report TCD-2002-28). Dept. of Computer Science, Trinity College, Dublin, Ireland.

Kalousis, A., Prados, J., & Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowl. Inf. Syst.*, *12*, 95–116.

Krízek, P., Kittler, J., & Hlavác, V. (2007). Improving stability of feature selection methods. *Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns* (pp. 929–936).

Kuncheva, L. (2007). A stability index for feature selection. *Proceedings of the 25th International Multi-Conference on Artificial Intelligence and Applications* (pp. 309–395).

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*, 2507–2517.

Feature Selection Techniques for Database Cleansing: Knowledge-driven vs. Greedy Search

Marieke van Erp M.G.J.vanErp@uvt.nl
Antal van den Bosch Antal.vdnBosch@uvt.nl
Piroska Lendvai P.Lendvai@uvt.nl
Steve Hunt S.J.Hunt@uvt.nl
ILK, Tilburg University, P.O. Box 90153, NL-5000 LE Tilburg, The Netherlands

Abstract

A comparative study is presented in which a wrapper-based feature selection algorithm and a knowledge-driven feature selection approach are compared to each other on a database cleaning task. Results show that both perform better than a classifier that is trained on the full feature set. The knowledge-driven approach provides a considerable speed up of the experiments over the wrapper-based feature selection approach.

1. Introduction

In order for machine learning algorithms to achieve optimal performance on a particular task, it is important to present the machine learner with optimal training data, e.g., not containing any attributes that may distract the algorithm. Today's data sets often contain large numbers of attributes, hence it is often not possible for human experts to remove bad attributes from the data before it is presented to the machine learner, or there may not even be a human expert to do this. Automatic feature selection promises to take over this task from humans, but may come at a cost. Some algorithms, such as C4.5 (Quinlan, 1993) order features during classification, but more often feature selection takes place before the actual classification. Among these techniques is the wrapper approach (Kohavi & John, 1997) in which the algorithm that will be used in the classification task is run on different feature subsets to evaluate the 'goodness' of each set for the final classification task. The order of the feature subsets under scrutiny can be organised via various algorithms that browse the feature set search space in a smart way to avoid having to test every possible subset of features.

Another approach is to mimic the human expert who chooses features based on the world knowledge he/she has by selecting features via the introduction of knowledge in the classification experiment, for instance via an ontology. Some work has been done to steer feature selection via a domain ontology for microarray gene expression data (Qi & Tang, 2006). This work applies a similar approach to the natural history domain.

The natural history domain harbours vast amounts of data, of which some is digitised (mostly manually). In this work a flat database from the Dutch National Museum for Natural History (Naturalis) is used. It contains 16,870 records organised into 39 columns that describe findings and characteristics of reptile and amphibian specimens, as well as how they are preserved in the collection and when they entered the collection. The data was entered manually by researchers at the museum and contains typos, spelling variations and incorrect values.

In earlier work, a machine learner was already applied to this data to detect and correct errors (Sporleder et al., 2006). Error detection and correction is treated as a classification task, in which it is assumed that the majority of the database fields is correct, and can be used as training data to predict other database fields. In that work all features were used, which is regarded here as the baseline because it is known that the performance of the k nearest neighbour algorithm can deteriorate when it is trained on irrelevant features (Wettschereck et al., 1997).

2. Domain Knowledge

To express domain knowledge, an ontology was created for the reptiles and amphibians via the CIDOC-CRM scheme (international standard ISO 21127:2006). By adhering to the CIDOC-CRM structure it is easier to integrate the ontology with information sources from

Table 1. Number of times each approach achieves highest accuracy over the other approaches

	all feats	*hill climb*	*ontology*
all feats	–	14 (2)	28 (3)
hill climb	23 (2)	–	29 (1)
ontology	8 (3)	9 (1)	–

within the same institution, or even from other institutions (Crofts et al., 2007).

3. Experimental setup

For the ontology-driven feature selection (*ontology*), 3 experiments were performed per database column, where the number of experiments represents the maximum distance between the *to be predicted concept* (i.e., database column) and the *concepts in the ontology* (i.e, features). The experiments were carried out in an incremental fashion: first all features within distance 1 to the focus column in the ontology were selected, then all the features within distance 2, etc.

The ontology-driven feature selection was compared to a greedy heuristic search, namely a bi-directional hill climber (*hill climb*). The search carried out by this method starts out with an empty feature set. During each iteration it adds or deletes the one feature that, in combination with the other features already selected, improves the classifier the most (Caruana & Freitag, 1994). This heuristic feature selection algorithm does a very thorough job and can lead to very good results, but is very expensive.

All experiments were carried out in a leave-one-out cross validation setting, with the implementation of feature-weighted k-NN as implemented in the TiMBL software package (Daelemans et al., 2007). Empty database fields were not included in the values that were to be predicted because in some cases a field is supposed to be empty, whereas in some cases it is not, which is out of the scope of this work.

4. Results

In Table 1 the number of times every approach 'wins' from the other approaches is given, where the rows represent the score of the approach against the others in each column. The numbers in parentheses are draws. 'All feats' is the baseline with the complete feature set.

5. Conclusions

For most database columns, there is an increase in accuracy when feature selection is applied. As expected the heuristic feature selection approach performs best, but often the accuracy of the ontology-driven approach was very close. The added bonus of the ontology-driven approach is that it is very fast, whereas the hill climbing search took significantly more time to come up with a well performing feature set. The results vary, because there is much variety between the different database columns. Follow-up research will include investigating how these differences affect each approach, and whether it is possible to combine the knowledge-driven approach with the heuristic approach.

Acknowledgements

This research is funded by NWO, the Netherlands Organisation for Scientific Research as part of the CATCH programme.

References

Caruana, R., & Freitag, D. (1994). Greedy attribute selection. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 28–36).

Crofts, N., Doerr, M., Gill, T., Stead, S., & Stiff, M. (2007). *Definition of the cidoc conceptual reference model*. The version 4.2.2 of the reference document.

Daelemans, W., Zavrel, J., Van der Sloot, K., & Van den Bosch, A. (2007). *Timbl: Tilburg memory based learner, version 6.1, reference guide.* (Technical Report 07-07). ILK/Tilburg University.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*, 273–324. Special issue on 'Relevance'.

Qi, J., & Tang, J. (2006). Gene Ontology Driven Feature Selection from Microarray Gene Expression Data. *Proceedings of CIBCB'06. 2006* (pp. 1–7).

Quinlan, J. (1993). *C4.5: Programs for machine learning.* Morgan Kaufmann.

Sporleder, C., van Erp, M., Porcelijn, T., & van den Bosch, A. (2006). 'spotting the odd-one-out': Data-driven error detection and correction in textual databases. *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM).* Trento, Italy.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and comparative evaluation of feature-weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review, special issue on Lazy Learning*, *11*, 273–314.

Deriving p-values for Tree-based Variable Importance Measures

Vân Anh Huynh-Thu VaHuynh@ulg.ac.be
Louis Wehenkel L.Wehenkel@ulg.ac.be
Pierre Geurts P.Geurts@ulg.ac.be
Department of Electrical Engineering and Computer Science
GIGA-Research, Bioinformatics and Modeling Unit, B34, University of Liège, B-4000 Liège, Belgium

1. Motivation

In many applications and specially in bioinformatics, an important task is the selection or ranking of some variables or features according to their relevance to predict some target variable, e.g. in order to identify genes or markers involved in a disease (see Saeys et al. (2007) for a recent review). Feature selection has usually two goals: improve the model accuracy by reducing overfitting and provide a better understanding of the problem under study. In this research, we focus on the latter goal.

Univariate approaches in the form of statistical tests are widely used in this domain. These methods provide for each variable a so called p-value that represents the probability of getting a value of some statistic as unusual as the observed one under the (null) hypothesis that there is no correlation between the variable and the target. These methods are statistically well founded and provide results that are intuitive and easily interpretable by practitioners. However, they make sometimes strong parametric hypotheses and they potentially miss important features that are only relevant in interaction with some other features.

On the other hand, machine learning brings several feature selection and ranking approaches that can be applied to a large variety of prediction problems and are able to model feature dependencies. Among them, variable importance measures derived from tree-based models have been used in many application domains (see Saeys et al. (2007) for some references in bioinformatics). These methods make no strong parametric hypotheses about the problem, can handle mixtures of categorical and numerical variables, and are often computationally very efficient. However, with respect to univariate statistical tests, this kind of importance measure is not so intuitive for practitioners. One consequence is that it is not easy to indicate an importance threshold in order to distinguish truly relevant from irrelevant variables. Furthermore, despite their popularity, there has not been much empirical studies of these measures in controlled experiments (Strobl et al. (2007) is an exception but it however focuses only on the detection of a univariate effect).

In this context, our goal in this research is two-fold: (i) improve interpretability by proposing a p-value to be associated to each variable in the ranking and (ii) provide a systematic study of these measures on artificial datasets in order to assess their strengths and limitations. Below, we briefly describe tree-based variable importance measures and how we propose to exploit random permutations to evaluate them. We then report some preliminary results on artificial data.

2. Tree-based importance measures

Tree-based algorithms, single trees or ensembles of trees, allow to easily compute an attribute importance measure for a given classification problem. Among the importance measures proposed in the literature, we use the information measure from (Wehenkel, 1998). Namely, at each test node n, we compute the total reduction of the class entropy due to the split, which is defined by:

$$I(n) = \#S H_C(S) - \#S_t H_C(S_t) - \#S_f H_C(S_f), \quad (1)$$

where S denotes the set of samples that reach node n, S_t (resp. S_f) denotes its subset for which the test is true (resp. false), $H_C(\cdot)$ is the Shannon entropy of the class frequencies in a subset, and $\#$ denotes the cardinality of a set of samples. The overall importance of an attribute is then computed by summing the I values of all tree nodes (of a single tree, or of an ensemble of trees) where this attribute is used to split. The importances are usually normalized for the different variables so that they sum up to 100%.

3. Evaluation by random permutations

The false discovery rate, FDR, has been introduced to control multiple testing issues in the context of univariate tests. It is defined as the proportion of irrele-

vant variables among all variables that are predicted as relevant by the model (Storey & Tibshirani, 2003). This concept could be applied directly to the variable importance ranking provided by a tree-based method. For an importance threshold v_{imp}, the FDR is estimated as the proportion of variables that receive an importance greater than v_{imp} under the null hypothesis that all variables are irrelevant. The null hypothesis can be simulated by randomly permutating the class labels in the learning sample.

One problem of this approach in a multivariate context is that it assumes that the relevance measures are independent of each other, and it has been shown that this assumption potentially leads to an overestimation of the FDR (see Listgarten and Heckerman (2007) for a discussion of this issue in the context of Bayesian networks). In order to overcome this problem, we propose an alternative measure based on the computation of a conditional p-value, which we define, for a variable with an importance v_{imp}, as the probability that it receives an importance equal or greater than v_{imp}, under the hypothesis that this variable and all its successors in the ranking are irrelevant. This quantity can be estimated by random permutations as well, by permutating the values of those variables that have an importance equal or smaller than v_{imp} instead of permutating the class labels, so as to keep the original relationship between the first variables and the target class unchanged.

Like the FDR, this conditional p-value has an intuitive interpretation. It should better reflect the relevance of a variable as it takes into account the dependence between the importances.

4. Preliminary results

To give some intuition about the proposed measures, we report here a small experiment on some artificial data. The dataset is composed of 100 objects and 20 variables. The first three are really relevant (and of decreasing importance), while the others are pure Gaussian noise. The problem is such that the third variable is only relevant in combination with the first two. Table 1 shows variable importances obtained with ensembles of extremely randomized trees (Geurts et al., 2006) and one standard statistical test (the Mann-Whitney U test) for comparison. FDRs and conditional p-values are estimated from 1000 random permutations.

We observe that the true ranking is well retrieved by the tree-based method while the univariate Mann-Whitney test fails at finding the third feature (as expected). The FDRs and cond. p-values usefully complement the ranking and the cond. p-value is indeed less conservative than the FDR, giving more chance to the third variable to be considered relevant.

Table 1. Variable rankings

ET				Mann-Whitney		
Var.	Imp.	FDR	cond. p	Var.	p	FDR
feat1	17.7	0	0	feat1	4.5e-09	0
feat2	15.6	0	0	feat2	1.2e-07	0
feat3	6.7	0.36	0.06	feat5	4.1e-02	0.27
feat8	5.4	1	0.23	feat9	5.0e-02	0.25
feat9	4.8	1	0.60	feat7	6.2e-02	0.26
feat19	4.2	1	0.93	feat18	6.8e-02	0.23
feat5	4.1	1	0.94	feat19	7.2e-02	0.21
feat14	3.8	1	0.98	feat10	1.5e-01	0.39
feat6	3.8	1	0.98	feat3	3.1e-01	0.71
...				...		

We are currently investigating how these measures behave when parameters of the problems (number of irrelevant and relevant features, learning sample size, etc.) and of the algorithms (splitting criterion, importance measure, etc.) are varied. In the light of these new findings, the same approach will be applied to real practical bioinformatics problems.

Acknowledgments

V.A. Huynh-Thu is recipient of a F.R.I.A. fellowship. P. Geurts is a Research Associate of the F.R.S.-FNRS. This work presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

References

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*, 3–42.

Listgarten, J., & Heckerman, D. (2007). Determining the number of non-spurious arcs in a learned DAG model: Investigation of a bayesian and a frequentist approach. *Proceedings of UAI*. UAI Press.

Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*, 2507–2517.

Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proc Natl Acad Sci U S A*, *100*, 9440–9445.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Horthorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*.

Wehenkel, L. (1998). *Automatic learning techniques in power systems*. Boston: Kluwer Academic.

Linear Regression using Costly Features

Robby Goetschalckx ROBBY@CS.KULEUVEN.BE
KULeuven - Department of Computer Science, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Scott Sanner SCOTT.SANNER@NICTA.COM.AU
National ICT Australia, Tower A, 7 London Circuit, Canberra City ACT 2601, Australia

Kurt Driessens KURTD@CS.KULEUVEN.BE
KULeuven - Department of Computer Science, Celestijnenlaan 200A, 3001 Heverlee, Belgium

1. Introduction

In this paper we consider the problem of linear regression where some features might only be observable at a certain cost. We assume that this cost can be expressed in the same units and thus be compared to the approximation error cost. The learning task becomes a search for the features that contain enough information to warrant their cost. Costs of features could reflect necessary time for complicated computations, the price of a costly experiment setup, the price an expert asks for advice, ...

Sparsity in linear regression has been widely studied, cf. chapter 3 of (Hastie et al., 2001). In these methods, striving for sparsity in the parameters of the linear approximation leads to higher interpretability and less over-fitting. Our setting, however, is significantly different as sparsity is gained by only including features with a cost lower than their value.

Including costs of features for classification and regression has also been discussed in (Turney, 2000; Domingos, 1999; Goetschalckx & Driessens, 2007). In these approaches, binary tests are used in a decision tree if the expected information gain is worth more than the cost. The difference with this paper is that here we are able to deal with numerical features instead of only binary ones.

2. Problem Statement

The problem we try to solve can be stated as following.

Given:

- a set X of examples,$P(x)$ is the distribution from which examples x are drawn
- a target function $Y : X \to \mathbb{R}$ which we want to approximate
- a set F of features: $f_i : X \to \mathbb{R}$
- a cost of features: $c : R \times X \to \mathbb{R}$

Find: a subset $\mathcal{F}$ of F and weights w_i such that the accumulated cost, consisting of the approximation error and the feature cost, is minimized: find $\arg\min_{\mathcal{F},\vec{w}} \sum_{x \in X} P(x) \left[|Y(x) - \hat{y}(x)| + \sum_{f \in \mathcal{F}} c(f, x) \right]$ where $\hat{y}(x) = w_0 + \sum_{f_i \in \mathcal{F}} w_i f_i$.

In other words, we try to find a linear combination of a subset of the features, where the absolute error of the approximation is compared to the cost of the set of features.

3. Cost Sensitive Forward Stagewise Regression

Many sparse linear regression methods are based on least-angle regression methods such as *lasso* and *forward stepwise regression* (Efron et al., 2002). In these methods the cost function includes a term $\sum_{f_i \in F} |w_i|$, to encourage lower weights and sparsity. For our setting, we simply change this to $\sum_{f_i \in \mathcal{F}} c(f_i, x)$.

We adapt an existing method, forward-stepwise regression, to solve the problem in an efficient way. We briefly describe this below and refer the reader to the detailed discussion in (Efron et al., 2002) for more information on this and related methods. The modified algorithm will be called C-FSR.

We start by gathering a batch of examples, together with all their feature values and their target values. We take the average target value and assign its value to w_0 (this normalizes the residuals). All features are normalized to have 0 mean and standard deviation

equal to 1, and assigned the feature vectors $\vec{f}_i$. We assign the normalized target values to the vector of residuals $\vec{r}$.

Cost-sensitive Forward-Stagewise Regression (C-FSR)
repeat:

1. Find the feature

$$f_i = \arg\max_{f \in F} |\vec{f}_i \cdot \vec{r}| - \mathbb{I}[f_i \notin \mathcal{F}] \cdot c_i > 0$$

2. Increase its weight w_i by $\epsilon.sign(\vec{f}_i \cdot \vec{r})$
3. Recompute residuals
4. $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_i\}$

until no such feature can be found.

In step 1, the cost is only included in the score if the feature is not yet introduced in the linear function.

It should be noted that the C-FSR is a *greedy* selection approach. Because of this, the approximation obtained might not always be globally optimal. A local optimum is guaranteed, however and the increase in prediction accuracy of using this greedy feature set is guaranteed to equal or exceed the cost of its computation.

4. Experiments

The setting we used for experiments is a simple domain, where there are seven different examples with different values. We provided seven indicator features f_i for $1 \leq i \leq 7$, one for every example. f_1 and f_4 have a cost c, the others are cost-free.

We varied the value of c over a range of 0 to 0.5. With increasing c the agent can not distinguish between x_1 and x_4 without paying a cost. C-FSR showed a clear-cut phase transition near $c = 0.185$ (the theoretically correct threshold value): with lower costs, the cost for the features is always payed, with higher costs the agent prefers not to use the costly features. This can be clearly seen in figure 1.

Here the cost the agent pays is compared with the prediction error the agent is aware of of making (the norm of the final residual vector). For very low values of c, the agent actually computes both of the features while only one is needed. As the costs are indeed very low, this does not pose a real problem. For low values of c, the agent keeps paying for one of the features. For values higher than the threshold, the error the agent is aware of making is lower than the cost of extra information and the agent will not pay for the features. For larger domains similar tests have shown that the algorithm scales well, with running time linear in the number of features.

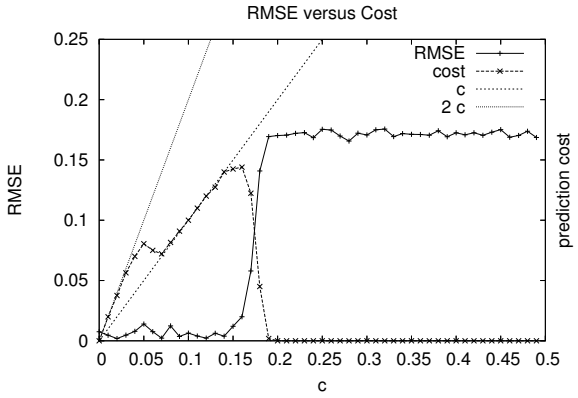


Figure 1. Error the agent is aware of making versus the amount spent on costly features

5. Conclusion

We introduced a novel sparse linear regression method, C-FSR, in the case where features have different costs. Empirically we have shown that C-FSR behaves near-perfectly for a test domain.

Acknowledgments

The research presented in this paper was funded by the Fund for Scientific Research (FWO Vlaanderen), and by National ICT Australia.

References

Domingos, P. (1999). Metacost: A general method for making classifiers cost-sensitive. *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining* (pp. 155–164).

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2002). *Least angle regression* (Technical Report). Statistics Department, Stanford University.

Goetschalckx, R., & Driessens, K. (2007). Cost sensitive reinforcement learning. *Proceedings of the workshop on AI Planning and Learning* (pp. 1–5).

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning.* Springer-Verlag.

Turney, P. (2000). Cost-sensitive learning bibliography. Institute for Information Technology, National Research Council, Ottawa, Canada.

Automatic Regression Modeling with Active Learning

Dirk Gorissen DIRK.GORISSEN@UGENT.BE
Tom Dhaene TOM.DHAENE@UGENT.BE
Eric Laermans ERIC.LAERMANS@UGENT.BE
Ghent University - IBBT, Department of Information Technology (INTEC), Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium

Abstract

Many complex, real world phenomena are difficult to study directly using controlled experiments. Instead, the use of computer simulations has become commonplace as a feasible alternative. However, due to the computational cost of these high fidelity simulations, the use of neural networks, kernel methods, and other surrogate modeling techniques have become indispensable. Surrogate models are compact and cheap to evaluate, and have proven very useful for tasks such as optimization, design space exploration, visualization, prototyping, and sensitivity analysis. Consequently, there is great interest in techniques that facilitate the construction of such regression models, while minimizing the computational cost and maximizing model accuracy. We present a fully automated machine learning toolkit for regression modeling and active learning to tackle these issues. We place a strong focus on adaptivity, self-tuning and robustness in order to maximize efficiency and make our algorithms and tools easily accessible to other scientists in computational science and engineering.

1. Introduction

For many problems from science and engineering it is impractical to perform experiments on the physical world directly (e.g., airfoil design, earthquake propagation). Instead, complex, physics-based simulation codes are used to run experiments on computer hardware. While allowing scientists more flexibility to study phenomena under controlled conditions, computer experiments require a substantial investment of computation time (one simulation may take many minutes, hours, days or even weeks) (Wang & Shan, 2007). As a result, the use of various approximation methods that mimic the behavior of the simulation model as closely as possible has become standard practice. This work concentrates on the use of data-driven, **global** approximations using compact surrogate models (also known as metamodels, or response surface models (RSM)). Popular metamodel types include: neural networks, Kriging models, and Support Vector Machines (SVM).

Global surrogate models provide a fast and efficient way for the engineer to explore the relationship between parameters (design space exploration), study the influence of various boundary conditions on different optimization runs, or enable the simulation of large scale systems where this would normally be too cumbersome. For the last case a classic example is the full-wave simulation of an electronic circuit board. Electromagnetic modeling of the whole board in one run is almost intractable. Instead the board is modeled as a collection of small, compact, accurate replacement surrogate models that represent the different functional components (capacitors, resistors, ...) on the board.

2. Motivation

However, in order to come to an acceptable approximation, numerous problems and design choices need to be overcome: what data collection strategy to use (active learning), what model type is most applicable (model selection), how should model parameters be tuned (hyperparameter optimization), how to optimize the accuracy vs. computational cost trade-off, etc. Particularly important is the data collection strategy. Since data is computationally expensive to obtain, it is impossible to use traditional, one-shot, space filling experimental designs. Data points must be selected iteratively, there where the information gain will be the greatest. An intelligent sampling function is needed that minimizes the number of sample points selected in each iteration, yet maximizes the informa-

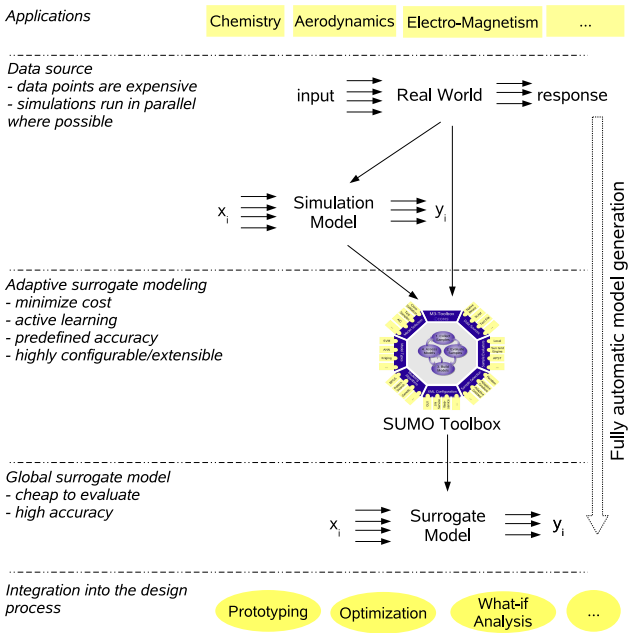


Figure 1. Automatic Adaptive Surrogate Modeling

tion gain of each iteration step. This is the process of active learning (Sugiyama & Ogawa, 2002), but it is also known as adaptive sampling or sequential design.

Together this makes that there are an overwhelming number of options available to the designer: different model types, different experimental designs, different model selection criteria, etc. However, in practice it turns out that the designer rarely tries out more than one subset of options. All too often, surrogate model construction is done in a one-shot manner. Iterative and adaptive methods, on the other hand, have the potential of producing a much more accurate surrogate at a considerably lower cost (less data points) (Busby et al., 2007). We present a state-of-the-art machine learning platform that provides an automatic, flexible and rigorous means to tackle such problems and that can easily be integrated in the engineering design process: the **SU**rrogate **MO**deling (SUMO) Toolbox.

3. SUMO Toolbox

The SUMO Toolbox (Gorissen et al., 2006) is an adaptive tool that integrates different modeling approaches and implements a fully automated, adaptive global surrogate model construction algorithm. Given a simulation engine, the toolbox automatically generates a surrogate model within the predefined accuracy and time limits set by the user (see figure 1). However, at the same time keeping in mind that there is no such thing as a 'one-size-fits-all', different problems need to be modeled differently. Therefore the toolbox was designed to be modular and extensible but not be too cumbersome to use or configure. Different plugins are supported: model types (neural networks, SVMs, splines, ...), hyperparameter optimization algorithms (Pattern Search, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), ...), active learning (density based, error based, gradient based, ...), and sample evaluation methods (local, on a cluster or grid). The behavior of each component is configurable through a central XML configuration file and components can easily be added, removed or replaced by custom, problem-specific, implementations.

The difference with existing machine learning toolkits such as Rapidminer (formerly Yale), Spider, Shogun, Weka, and Plearn is that they are heavily biased towards classification and data mining (vs. regression), they assume data is freely available and cheap (no active learning), and they lack advanced algorithms for the automatic selection of the model type and model complexity.

Our approach has been successfully applied to a very wide range of fields ranging from combustion modeling in chemistry and metallurgy, semi-conductor modeling (electromagnetism), aerodynamic modeling (aerospace), to structural mechanics modeling in the car industry. Its success is primarily due to its flexibility, self tuning implementation, and its ease of integration into the larger computational science and engineering pipeline.

References

Busby, D., Farmer, C. L., & Iske, A. (2007). Hierarchical nonlinear approximation for experimental design and statistical data fitting. *SIAM Journal on Scientific Computing, 29*, 49–69.

Gorissen, D., Hendrickx, W., Crombecq, K., & Dhaene, T. (2006). Integrating gridcomputing and metamodelling. *Proceedings of 6th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2006)* (pp. 185–192). Singapore.

Sugiyama, M., & Ogawa, H. (2002). Release from active learning/model selection dilemma: optimizing sample points and models at the same time. *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on* (pp. 2917–2922).

Wang, G. G., & Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design, 129*, 370–380.

Active Learning for Primary Drug Screening

Kurt De Grave, Jan Ramon and Luc De Raedt {KURT.DEGRAVE,JAN.RAMON,LUC.DERAEDT}@CS.KULEUVEN.BE
Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract

We study the task of approximating the k best instances with regard to a function using a limited number of evaluations. We also apply an active learning algorithm based on Gaussian processes to the problem, and evaluate it on a challenging set of structure-activity relationship prediction tasks.

1. Introduction

High Throughput Screening (HTS) is a step in the drug discovery process, in which chemical compounds are screened against a biological assay. The goal of this step is to find a few lead compounds within the entire compound library that exhibit a very high activity in the assay. In this type of application, only partial information can be obtained by testing specific instances for their performance. Such tests correspond to experiments and can be quite expensive. The challenge then is to identify the best performing instances using as few experiments as possible.

2. Problem statement

Our work is especially motivated by the structure-activity relationship domain, where HTS approaches often assume the availability of a large but fixed library of chemical compounds. Hence, we assume the learner must select the next example from a finite pool.

We formally specify the problem as follows:
GIVEN:

- a pool $\mathcal{P}$ of instances,
- an unknown function f that maps instances $x \in \mathcal{P}$ on their target values $f(x)$,
- an oracle that can be queried for the target value of any example $x \in \mathcal{P}$,
- the maximal number N_{max} of queries,
- the number k of best examples searched for.

FIND:

- the top k instances in $\mathcal{P}$, that is, the k instances in $\mathcal{P}$ that have the highest values for f.

From a machine learning perspective, the key challenge is to determine the policy for determining the next query to be asked on the basis of the already known examples. This policy will have to keep the right balance between exploring the whole pool of examples and exploiting those regions in the pool that look most promising.

3. Model and selection strategies

We will use a Gaussian process (GP) model for learning (Gibbs, 1997). Detailed explanations can be found in several textbooks, e.g. (Bishop, 2006). The GP model allows us to calculate the probability distribution of the target value t_* of a new example x_* given the tested examples X_N and their measured target values T_N:

$$t_* | X_N, T_N, x_* \sim \mathcal{N}(\bar{t}_*, var(t_*)) \qquad (1)$$

Different active learning strategies exist. In line with the customary goal of inducing a model with maximal accuracy on future examples, most approaches involve a strategy aiming to greedily improve the quality of the model in regions of the example space where its quality is lowest. One can select new examples for which the predictions of the model are least certain or most ambiguous. Depending on the learning algorithm, this translates to near decision boundary selection, ensemble entropy reduction, version space shrinking, and others. In our model, it translates to *maximum variance* on the predicted value or $\arg\max(var(t_*))$.

(Warmuth et al., 2003) found that in a highly skewed distribution, recall increases quickly when one selects examples for testing that are most likely to belong to the minority class. For our optimization problem we

will test the equivalent method of selecting the example that the current model predicts to have the best target value, or $\arg\max(\bar{t}_*)$. We will refer to this as the *maximum predicted* strategy.

Another strategy is to always choose the example for which the *optimistic* guess is maximal. The idea is not to test the example in the database where the predicted value $\bar{t}_*$ is maximal, but the example where $\bar{t}_* + k_{optimism} \cdot var(t_*)$ is maximal.

An alternative strategy is to select the example x_{N+1} that has the highest probability of improving the current solution, as described by (Lizotte et al., 2007). Let the current step be N, the aggregate value of the set of k best examples be $\|T_N\|_{\text{best}-k}$ and the target value of the k-th best example be $t_{\#(k,N)}$. We can evaluate this probability computing the cumulative Gaussian

$$P(t_* > t_{\#(k,N)}) = \int_{t=t_{\#(k,N)}}^{\infty} (t - \bar{t}_*) P(t_* = t) dt. \quad (2)$$

We call this the *maximum gain probability* strategy.

4. Experimental evaluation

We evaluated the algorithm on the US National Cancer Institute (NCI) 60 anticancer drug screen (NCI60) dataset (Shoemaker, 2006). A pool of 2,000 compounds was randomly selected from each assay.

We used a linear kernel using for each compound 1024 Open Babel FP2 fingerprints as features. The algorithm was bootstrapped with measurements of ten random compounds. Each experiment was repeated 20 times for every assay and the results were averaged.

Budget	10%	15%	20%	25%
Max predicted (0σ)	0.251	0.684	0.040	0.021
Optimistic (0.5σ)	0.521	**Best**	0.251	0.111
Optimistic (1σ)	0.182	0.469	**Best**	**Best**
Optimistic (2σ)	0.618	0.958	0.179	0.298
Max variance ($\infty\sigma$)	ϵ	ϵ	ϵ	ϵ
Max gain prob	**Best**	0.982	0.189	0.052
Random selection	ϵ	ϵ	ϵ	ϵ

Table 1. Best strategy (attaining highest $\|T_{N_{max}}\|_{\text{best}-10}$) and Wilcoxon signed-rank test p-value for the null hypothesis that the difference between the top-10 values of this strategy and those of the best strategy is on average 0. ϵ indicates that $p < 10^{-10}$. Budget shown as % of pool size.

From the results presented in Figure 1 and Table 1 one can see that all strategies, except maximum-variance, clearly perform much better than random example selection. The 1σ optimistic strategy performs best over the widest range of budgets.

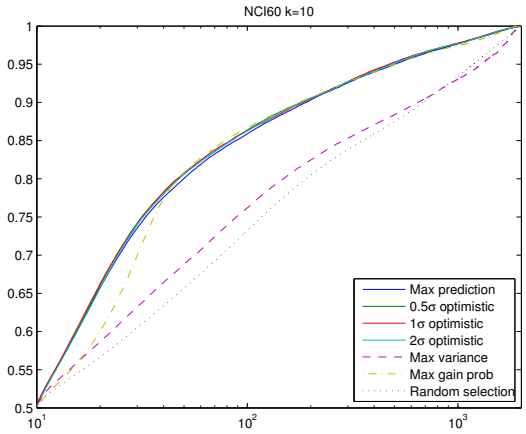


Figure 1. The value of $\|T_N\|_{\text{best}-10}$ as a function of the fraction of compounds tested. The vertical axis is scaled to place the aggregate target value of the overall k best compounds at one and the worst k compounds at zero.

5. Conclusions

To summarize: we introduced the best-k optimization problem in a machine learning context, we proposed an approach based on Gaussian processes to tackle it, and we applied it successfully to a challenging structure activity relationship prediction task.

Acknowledgements

Kurt De Grave is supported by GOA 2003/08 "Inductive Knowledge Bases". Jan Ramon is a postdoctoral fellow of the Fund for Scientific Research (FWO) of Flanders. This research used HPC resources (`http://ludit.kuleuven.be/hpc`).

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Gibbs, M. (1997). *Bayesian gaussian processes for regression and classification*. Doctoral dissertation.

Lizotte, D., Wang, T., Bowling, M. and Schuurmans, D. (2007). Automatic gait optimization with Gaussian process regression. *Proceedings of IJCAI-07*, 944–949.

Shoemaker, R. (2006). The NCI60 human tumour cell line anticancer drug screen. *Nat. Rev. Cancer*, *6*, 813–823.

Warmuth, M. K. et al. (2003). Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.*, *43*, 667–673.

When can we simplify a one-versus-one multi-class classifier to a single ranking?

Willem Waegeman WILLEM.WAEGEMAN@UGENT.BE
Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 913, B-9052 Ghent, Belgium

Bernard De Baets BERNARD.DEBAETS@UGENT.BE
Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, B-9000 Ghent, Belgium

Luc Boullart LUC.BOULLART@UGENT.BE
Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 913, B-9052 Ghent, Belgium

Abstract

We try to answer the question posed in the title from a theoretical point of view. To this end, sufficient conditions are derived for which a one-versus-one ensemble becomes ranking representable, i.e. conditions for which the ensemble can be reduced to a ranking or ordinal regression model such that a similar performance on training data is measured. As performance measure, we use the area under the ROC curve (AUC) and its reformulation in terms of graphs. By means of a graph-theoretic analysis of the problem, we are able to formulate necessary and sufficient conditions for ranking representability. For the three class case, this results in a new type of transitivity for pairwise AUCs that can be verified by solving an integer quadratic program.

1. Introduction

Many machine learning algorithms for multi-class classification aggregate several binary classifiers to compose a decision rule. In the popular one-versus-one ensemble (Fürnkranz, 2002), a classifier is trained on each pair of categories, but do we really need such a complex model for every multi-class classification task? One might agree that different multi-class classification schemes have a different degree of complexity, but no consensus has been reached on which one to prefer. In this work we go one step further and investigate whether a one-versus-one multi-class model can be simplified to a ranking model. We start from the assumption that the optimal complexity of a multi-class model is problem-specific. Reducing a one-versus-one ensemble to a ranking model, can be seen as a quite drastic application of the bias-variance trade-off: a one-versus-one classification scheme is a complex model, resulting in a low bias and a high variance of the performance, while an ordinal regression model is a much simpler model, manifesting a high bias but a low variance. So, we do not claim that a one-versus-one scheme can always be reduced to a ranking model. We rather look for necessary and sufficient conditions for such a reduction.

2. Strict ranking representability

Let $\mathcal{X}$ denote the object space and $\mathcal{Y}$ the unordered set of r labels. We use the notation $\mathcal{Y} = \{\overline{y}_1, ..., \overline{y}_r\}$ to denote the respective categories. Furthermore, we formally define a one-versus-one model as a set $\overline{\mathcal{F}}$ of $r(r-1)/2$ ranking functions $f_{kl} : \mathcal{X} \to \mathbb{R}$ with $1 \leq k < l \leq r$. Thus, we consider one-versus-one schemes for which each binary classifier produces a continuous output resulting in a probability estimate or a ranking of the data for each pair of categories. A dataset of size n will be denoted $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$.

Given a one-versus-one classification model, represented by a set $\overline{\mathcal{F}}$ of $r(r-1)/2$ pairwise ranking functions f_{kl}, when can we reduce this model to a single ranking $f : \mathcal{X} \to \mathbb{R}$ that gives a better performance on unknown test data? Or, equivalently, when can we simplify the one-versus-one model to a ranking model without decreasing the error on training data? Hav-

ing in mind the bias-variance trade-off, it would be appropriate to prefer the single ranking model over the one-versus-one scheme if the training error does not increase. In that case, the former model is complex enough to fit the data well in spite of having a lower variance over different training samples. In its most strict form, we can define ranking representability of a one-versus-one classification scheme as follows.

Definition 2.1. *Let $D \subset \mathcal{X} \times \mathcal{Y}$. We call a set $\overline{\mathcal{F}}$ of pairwise ranking functions f_{kl} strictly ranking representable on D if there exists a ranking function $f : \mathcal{X} \to \mathbb{R}$ such that for all $\overline{y}_k, \overline{y}_l \in \mathcal{Y}$ and any $(\mathbf{x}_i, \overline{y}_k), (\mathbf{x}_j, \overline{y}_l) \in D$*

$$f_{kl}(\mathbf{x}_i) \leq f_{kl}(\mathbf{x}_j) \Leftrightarrow f(\mathbf{x}_i) \leq f(\mathbf{x}_j)\,. \qquad (1)$$

We can define a unique directed graph for a set of pairwise rankings $\overline{\mathcal{F}}$, and strict ranking representability of $\overline{\mathcal{F}}$ can be easily checked with a simple algorithm that verifies whether the corresponding graph is a DAG.

3. AUC ranking representability

It goes without saying that strict ranking representability has a very limited applicability to reduce one-versus-one multi-class schemes, since the condition is too strong to be satisfied in practice. When fitting $r(r-1)/2$ functions to the data in a multi-class setting, it is unrealistic to think that all these functions will impose a consistent ranking, i.e. a ranking satisfying Eq. (1). Yet, is it really necessary to require strict ranking representability in order to exchange a one-versus-one model for a single ranking model? The answer is no, since we are interested in a good performance on independent test data. Therefore, demanding that a single ranking gives exactly the same result on training data as a one-versus-one scheme might be a too strong condition. An obvious relaxation could exist in requiring that a single ranking model yields the same *performance* on training data instead of requiring the same results. This makes a subtle difference since it is now allowed that both models make errors on different data objects, as long as the total error of both models is similar. As claimed above, the single ranking model should attain better results on independent test data when the bias-variance trade-off is taken into consideration.

The performance measure that we will consider is the pairwise AUC, which can be evaluated on a one-versus-one model as well as on a single ranking model. We will respectively use the notations $\widehat{A}_{kl}(\overline{\mathcal{F}}, D)$ and $\widehat{A}_{kl}(f, D)$ for the AUC obtained for categories $\overline{y}_k$ and $\overline{y}_l$. AUC ranking representability is defined as follows.

Definition 3.1. *Let $D \subset \mathcal{X} \times \mathcal{Y}$. We call a set $\overline{\mathcal{F}}$ of pairwise ranking functions f_{kl} AUC ranking representable on D if there exists a ranking function $f : \mathcal{X} \to \mathbb{R}$ such that*

$$\widehat{A}_{kl}(\overline{\mathcal{F}}, D) = \widehat{A}_{kl}(f, D) \quad \forall k, l : 1 \leq k < l \leq r\,. \qquad (2)$$

A graph-theoretic reformulation of AUC ranking representability can be established by defining a set $\mathfrak{G}_{AUC}(\overline{\mathcal{F}}, D)$ of graphs such that $\overline{\mathcal{F}}$ is AUC ranking representable if and only if $\mathfrak{G}_{AUC}(\overline{\mathcal{F}}, D)$ contains at least one acyclic graph. However, unlike strict ranking representability, it is far form trivial to verify whether a set $\overline{\mathcal{F}}$ of pairwise rankings f_{kl} is AUC ranking representable, since examining all graphs in $\mathfrak{G}_{AUC}(\overline{\mathcal{F}}, D)$ will be computationally intractable for large training samples. In the talk we will present a way to tackle the problem by using additional graph concepts and the framework of cycle transitivity (De Baets et al., 2006). Using this framework, we are able to define necessary conditions for AUC ranking representability, since the pairwise AUCs of an AUC ranking representable one-versus-one scheme are reciprocal relations coinciding with dice models (De Schuymer et al., 2003). These conditions can be easily verified in practice by analyzing the pairwise AUCs.

In another way, sufficient conditions for AUC ranking representability can also be translated into the framework of cycle transitivity. To this end, a new type of cycle transitivity is introduced, leading to a verifiable sufficient condition for the three class case. In this way, AUC ranking representability can be checked by solving an integer quadratic program.

Acknowledgment

Willem Waegeman is supported by a grant of the "Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen)".

References

De Baets, B., De Meyer, H., De Schuymer, B., & Jenei, S. (2006). Cyclic evaluation of transitivity of reciprocal relations. *Social Choice and Welfare*, *26*, 217–238.

De Schuymer, B., De Meyer, H., De Baets, B., & Jenei, S. (2003). On the cycle-transitivity of the dice model. *Theory and Decision*, *54*, 164–185.

Fürnkranz, J. (2002). Round robin classification. *Journal of Machine Learning Research*, *2*, 723–747.

Monotone Relabeling of Partially Non-Monotone Data: Restoring Regular or Stochastic Monotonicity

Michael Rademaker MICHAEL.RADEMAKER@UGENT.BE
Bernard De Baets BERNARD.DEBAETS@UGENT.BE
Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, 9000 Gent, Belgium

Hans De Meyer HANS.DEMEYER@UGENT.BE
Department of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281 S9, 9000 Gent, Belgium

Abstract

We examine the problem of non-monotonicity in multi-criteria data, and formulate different optimal relabeling algorithms for different domain constraints. More exactly, we examine the case where labels are ordinal or lacking a distance function, and the case where such a distance function is applicable. Furthermore, we discuss the problem of stochastic monotonicity, and give optimal algorithms for relabeling for this type of domain knowledge. Of central importance is the transitivity of the non-monotonicity relation, which permits formulation of each of these relabeling problems as an independent set problem in a comparability graph. Network flow algorithms can then be applied in order to yield optimal solutions in all but one of the problems discussed.

1. Introduction

We consider the problem of noise in the form of non-monotonicity. Three options present themselves given a data set subject to such noise: keep the data set as it is, identify the noisy instances and remove them from the data set, or identify the noisy instances and relabel them. It is this last option we will discuss here. We will examine regular and stochastic monotonicity, and discuss the ways in which these problems can be translated to independent set problems in comparability graphs. Of key importance is the transitivity of the non-monotonicity relation, permitting the formulation of the problem as one solvable by network flow algorithms and formulation of L1 loss optimal relabeling algorithms.

2. Multi-criteria data, monotonicity and independent sets

In multi-criteria data, instances can be ordered w.r.t. the scores on the different criteria. One instance is said to be strictly better than another instance if it received a score that is at least as good on each of the criteria, with the preference being strict for at least one criterion. For two such instances, one can expect the better instance to receive a label (coming from the collection of labels $\mathcal{L}$) that is at least as good as the worse instance. This background knowledge is known as the monotonicity requirement: an increase in criterion scores cannot result in a decrease in label.

Sometimes, assignment of a single label to each feature vector is too strict. In such cases, a distribution over the different labels is supplied for each feature vector, and regular monotonicity does not apply. Rather, we have stochastic monotonicity: the better feature vector should have a distribution that contains more higher labels than the worse feature vector. This is most easily seen on the basis of the cumulative distributions: the better feature vector has a cumulative distribution that takes higher values at higher labels, while the worse feature vector should take higher values for lower labels (see Section 3).

Of central importance is the fact that both regular and stochastic non-monotonicity are defined as transitive relations: if an instance or feature vector x is non-monotone w.r.t. an instance or feature vector y, it will also be non-monotone w.r.t. all instances or feature vectors which are better than y according to their

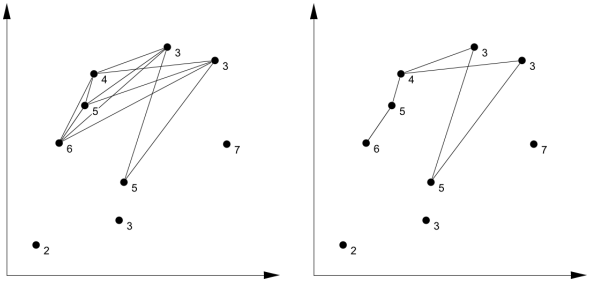


Figure 1. Graph representation of a partially non-monotone data set $\mathbb{S}_1$(numbers represent class labels)

features, and worse according to their label(s).

The independent set concept is relevant to the discussion of non-monotonicity in defining an optimal clean-up (Rademaker et al., 2006). This problem from graph theory deals with a graph G, comprised of a set of vertices V and a set of edges E. Figure 1(a) can be seen as a graph-representation of a data set $\mathbb{S}$ with instances $\mathcal{S}$. The set of instances corresponds to the set of vertices of our graph, and the edges we show denote two instances to be non-monotone (E_{nm}) w.r.t. each other. Though finding a maximum independent set is an NP-complete problem in the general case, it is solvable in our case: the transitivity inherent to the non-monotonicity relation means a graph such as the one in Figure 1(a) is actually a comparability graph by definition. The condensed representation of such a graph is shown in Figure 1(b). Through the use of a network flow algorithm, it is possible to determine the maximum independent set in $\mathcal{O}(|V|^3)$ time (Möhring, 1985). We will show how the problems of restoring regular and stochastic monotonicity through the relabeling of instances can be solved by these methods. Network flow representations are possible and straightforward, but have been omitted here for lack of space.

3. Solving the relabeling problem

Suppose we have the label function $d : \mathcal{S} \to \mathcal{L}$ that returns the label an instance received in the data set, and the label distance function $D : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ that quantifies how far apart two labels are. We now translate the minimal L1 relabeling problem to a weighted maximum independent set problem. To this end, we add a number of instances to the data set: we copy each instance $|\mathcal{L}| - 1$ times, and assign each of these copies a new label so that we end up with in total $|\mathcal{L}|$ copies, one for each label. The weight each of these instances receives is related to the label distance between the old and the new label. The original copy

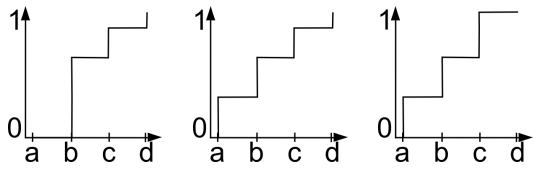


Figure 2. Cumulative frequency label distributions: one step better, original distribution and one step worse (labels denoted by letters)

of the instance, let us call it a_{old}, receives a weight of $A = D(\min(\mathcal{L}), \max(\mathcal{L}))$. Other instances receive a weight depending on their label, i.e. for a relabeled instance a' this weight is $A - D(d(a'), d(a_{old}))$. If we determine a maximum weighted independent set in this data set, we have a relabeling that minimizes the L1 loss.

An analogous technique can be used when stochastic monotonicity is demanded. Each feature vector has an original collection of labels. By stepwise relabeling all instances that received the worst label to the worst-but-one label, we can transitively shift the distribution to the better labels by relabeling instances. An analogous operation is possible to worsen the distribution by relabeling the best instances to the best-but-one label. As evidenced by Figure 2 which contains an original distribution and the next-best and next-worst relabeled version, such a collection of partially relabeled distributions can be ordered from best to worst, with the original distribution obviously being better than all those that have been constructed to be worse, and worse than those that have been constructed to be better. Weights can be assigned on the basis of the number of instances that still carry their original label (as a 0/1 loss equivalent), or on the basis of the number of times an instance needed to be relabeled in order to end up with the distribution in question (as an L1 loss equivalent). Illustrative examples will be provided, and extensions to simultaneously minimize the 0/1 loss and the L1 loss will be formulated.

References

Möhring, R. (1985). Algorithmic aspects of comparability graphs and interval graphs. *Proceedings of the NATO Advanced Study Institute on Graphs and Order* (pp. 41–101). Banff, BC, Canada.

Rademaker, M., De Baets, B., & De Meyer, H. (2006). On the role of maximal independent sets in cleaning data sets for supervised ranking. *Proceedings of the 2006 IEEE International Conference on Fuzzy Systems* (pp. 7810–7815). Vancouver, BC, Canada.

Learning in Kernelized Output Spaces with Tree-Based Methods

Pierre Geurts P.GEURTS@ULG.AC.BE
Louis Wehenkel L.WEHENKEL@ULG.AC.BE
University of Liège, Department of EE and CS & GIGA research Institut Montefiore, Sart Tilman B28, 4000, Liège, Belgium

Florence d'Alché-Buc FLORENCE.DALCHE@IBISC.UNIV-EVRY.FR
University of Evry, IBISC 523 Place des Terrasses de l'Agora, 91000, Evry, France

Abstract

The availability of structured data sets from XML documents to biological structures, such as sequences or graphs, has prompted for the development of new learning algorithms able to handle complex structured output spaces. Motivated by the success of kernel methods for handling complex input spaces, one approach to handle complex output problems is to embed the outputs into a kernelized space and develop algorithms that can work in such a space by exploiting the kernel trick. In this abstract, we present our recent work with tree-based methods in this domain.

1. Learning in kernelized output spaces

The general problem of supervised learning may be formulated as follows: from a learning sample $LS = \{(x_i, y_i)|i = 1, \ldots, N_{LS}\}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, find a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the expectation of some loss function over the joint distribution of input/output pairs:

$$E_{x,y}\{\ell(f(x), y)\}. \quad (1)$$

Let us suppose now that we have a kernelized output space, ie. an output space $\mathcal{Y}$ endowed with a kernel $k : \mathcal{Y} \times \mathcal{Y} \rightarrow I\!R$, and let us denote by $\phi : \mathcal{Y} \rightarrow \mathcal{H}$ the feature map defined by k. In this paper, we consider problems where the loss function is defined as follows:

$$\begin{aligned} \ell(y_1, y_2) &= ||\phi(y_1) - \phi(y_2)||^2 \\ &= k(y_1, y_1) + k(y_2, y_2) - 2k(y_1, y_2), \end{aligned} \quad (2)$$

which depends only on the output kernel.

The resolution of this problem usually involves two steps (see Figure 1):

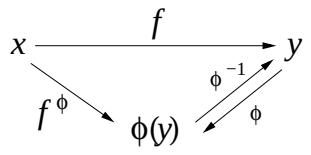


Figure 1. Learning with kernelized outputs

1. Learn an intermediate function $f^\phi : \mathcal{X} \rightarrow \mathcal{H}$ that maps each input vector into the Hilbert space and minimizes the following loss:

$$E_{x,y}\{||f^\phi(x) - \phi(y)||^2\}. \quad (3)$$

2. From a prediction $f^\phi(x)$ in $\mathcal{H}$, get a prediction $f(x)$ in the original output space $\mathcal{Y}$ by solving:

$$f(x) = \arg\min_{y \in \mathcal{Y}} ||\phi(y) - f^\phi(x)||^2. \quad (4)$$

For this solution to be practically feasible, we need to be able to write the algorithm from pairwise dot-products only. Solutions for the learning stage are usually obtained by using the kernel trick at the output of some standard supervised learning method that can handle a vectorial output. For example, Cortes et al. (2005) propose a kernelization of the output of kernel ridge regression. The methods that we present below are obtained by kernelizing the output of (multiple output) regression trees and gradient boosting. The optimization problem (4) is the so called preimage problem, whose solution is kernel specific.

Like other kernel methods, this formulation of the supervised learning problem has the advantage of decoupling the design of the algorithm from the design of the kernel. Given the important literature that exists on kernels, it allows one to develop generic algorithms that can potentially handle a large range of problems in terms of loss functions and output spaces.

2. Output kernel trees

In (Geurts et al., 2006), we propose a kernelization of the output of regression trees that we called OK3 for output kernel trees. The idea of standard regression trees is to recursively split the learning sample with binary tests based on the input variables, trying at each split to reduce as much as possible the (empirical) variance of the output in the left and right subsamples of learning cases corresponding to that split. Our kernelization of this method is based on the exchange of this variance for the average square distance from the center of mass in $\mathcal{H}$ that can be computed from kernel values only:

$$\frac{1}{N}\sum_{i=1}^{N}||\phi(y_i) - \frac{1}{N}\sum_{i=1}^{N}\phi(y_i)||^2 = \frac{1}{N}\sum_{i=1}^{N}k(y_i,y_i) - \frac{1}{N^2}\sum_{i,j=1}^{N}k(y_i,y_j)$$

Predictions at tree leaves are then written as linear combinations of learning sample outputs, which makes expression (4) again computable from kernel values only.

The use of ensemble methods is usually necessary to make tree-based methods competitive with other methods in terms of accuracy. The extension of ensemble methods based on randomization, such as bagging or random forests, to kernelized output spaces is straightforward as these methods only rely on score computations to grow the trees and combine predictions by simply averaging them. The extension of boosting methods is however generally not trivial as these methods manipulate the outputs. In (Geurts et al., 2007b), we show however that a particular type of boosting algorithm, gradient boosting with square loss, can be also modified to handle a kernelized output space. Just like in standard classification and regression settings, the use of ensemble methods usually improves very much with respect to single output kernel trees.

3. Supervised graph inference

One application of these methods is supervised graph inference. In this problem, we assume that we have a set of vertices $\{v_i, i = 1, \ldots, m\}$ that are related between each other by a graph. Each vertex is furthermore described by an input feature vector $x(v_i) \in \mathcal{X}$ and the goal is to find a function $g(x(v), x(v')) : \mathcal{X} \times \mathcal{X} \rightarrow \{0, 1\}$ that predicts as well as possible the existence of a connection between two (potentially new) vertices v and v' described by their input vectors.

Our solution is based on a kernel embedding of the graph (e.g., by a diffusion kernel) and the application of output kernel trees in the resulting kernelized output space. This application gives a model $f^{\phi}(.)$ and edge predictions are then obtained by thresholding the dot-product between the predictions of this model for two (new) input vectors:

$$g(x, x') = 1(\langle f^{\phi}(x), f^{\phi}(x')\rangle > k_{th}).$$

Since f^{ϕ} is expressed as a linear combination of outputs from the learning sample, this latter expression can again be computed from kernel values only.

In (Geurts et al., 2007a), we have applied this methodology to the supervised inference of two biological networks between Yeast genes (protein-protein interaction network and enzyme network) from various information about these genes (microarray expression, localization, and phylogenetic information). Our approach yields competitive results with respect to existing methods.

Acknowledgments

PG is a research associate of the FRS-FNRS (Belgium). This work has been done while he was a postdoc at IBISC (Evry, France) with support of the CNRS (France). FAB's research has been funded by Genopole (France). This work presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

References

Cortes, C., Mehryar, M., & Weston, J. (2005). A general regression technique for learning transductions. *Proceedings of ICML 2005* (pp. 153–160).

Geurts, P., Touleimat, N., Dutreix, M., & d'Alch Buc, F. (2007a). Inferring biological networks with output kernel trees. *BMC Bioinformatics, 8 Suppl 2*, S4.

Geurts, P., Wehenkel, L., & d'Alch Buc, F. (2006). Kernelizing the output of tree-based methods. *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)* (pp. 345–352).

Geurts, P., Wehenkel, L., & d'Alch Buc, F. (2007b). Gradient boosting for kernelized output spaces. *Proceedings of ICML 2007*.

Selective Inductive Transfer

Beau Piccart BEAU.PICCART@CS.KULEUVEN.BE
Jan Struyf JAN.STRUYF@CS.KULEUVEN.BE
Hendrik Blockeel HENDRIK.BLOCKEEL@CS.KULEUVEN.BE
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract

Multi-target models, which predict multiple target variables simultaneously, may predict some of the targets more accurately, and other targets less accurately than a single-target model. This raises the question whether it is possible to find, for a given main target, a subset of the other targets that, when combined with the main target in a multi-target model, results in the most accurate model for the main target. We propose Selective Inductive Transfer, an algorithm that automatically finds such a subset.

1. Introduction

We consider simultaneous prediction of multiple variables, which is known as multi-target or multi-objective prediction. In this setting, the input is associated with a vector of target variables, and all of them need to be predicted as accurately as possible. Multi-target prediction is encountered, e.g., in ecological modelling, where the domain expert is interested in (simultaneously) predicting the frequencies of different organisms in agricultural soil or river water.

It has been shown that multi-target models can be more accurate than predicting each target individually with a separate single-target model (Caruana, 1997). This is a consequence of the fact that when the targets are related (e.g., if they represent frequently co-occurring organisms in the ecological modelling applications mentioned above), they can carry information about each other; the single-target approach is unable to exploit that information, while multi-target models naturally exploit it. This is a form of inductive transfer: the information a target carries about the other targets is transferred to those other targets.

Multi-target models do, however, not always lead to more accurate prediction. For a given target variable, the variable's single-target model may be more accurate than the multi-target model. That is, inductive transfer from other variables can be beneficial, but it may also be detrimental to accuracy. Therefore, the subset of targets that, when combined with a given target (the main target) in a multi-target model, results in the most accurate model for the main target, may be non-trivial, i.e., different from the empty set and from the set of all targets. We call this set the support set for the main target.

2. Selective Inductive Transfer

We propose Selective Inductive Transfer (SIT), a greedy algorithm that approximates the support set for a given main target, and that works as follows.

1. Initialize the support set to the empty set.
2. Consider the multi-target model with as targets the main target and the current support set. Use this model to predict only the main target and estimate the resulting accuracy. (We use 10-fold cross validation to estimate accuracy.)
3. Add each candidate support target in turn to the current support set and estimate the main target's accuracy for each resulting multi-target model.
4. Select the candidate support target (if any) that yielded the largest increase in accuracy over the accuracy of the current support set and add it permanently to the support set. If no candidate improves accuracy, then return the current support set.
5. Go to step 2.

SIT has a number of advantages over related methods. First, other algorithms that exploit transfer selectively, such as the Task Clustering algorithm of Thrun and O'Sullivan (1996), often incorrectly assume transfer to be symmetric. This results in suboptimal models. SIT does not assume transfer to be symmetric; if a is a

support target for main target b, then b is not necessarily a support target for a. A second shortcoming of other methods, such as the η-MTL algorithm of Silver and Mercer (1996), is that they rely on heuristics, such as the linear correlation of the targets, to approximate transfer. Such heuristics may poorly approximate transfer. SIT measures transfer empirically (using cross validation). Therefore, it does not suffer from this problem. A third advantage of SIT is that it is a general method that can be combined with any multi-target learner.

3. Experimental Evaluation

The aim of our experiments is to test to which extent SIT, for a given main target, succeeds in finding a good set of support targets. To this end, we compare SIT to two common baseline models: a single-target model for the main target (ST), and a multi-target model that includes all targets (MT).

SIT has been implemented in the decision tree induction system Clus, which also implements single- and multi-target regression trees (Blockeel et al., 1998), and is available as open source software from `http://www.cs.kuleuven.be/~dtai/clus/`.

We compare for each target variable of 6 ecological modelling multi-target regression datasets, the predictive performance of a traditional single-target regression tree (STRT), a tree constructed by SIT with all other targets as candidate support targets, and a multi-target tree including all targets (MTRT).

Table 1 shows for each method, the cross validated Pearson correlation, averaged over all targets. (This measure is common in ecological modelling; we also use it as accuracy estimate in the internal cross validation performed by SIT.) SIT performs significantly better than STRT for 5 out of 6 datasets and never performs significantly worse. It performs comparable to MTRT or significantly better than MTRT in 5 out of 6 datasets. For one dataset (*Soil 2*), SIT still significantly outperforms STRT, yet it is significantly worse than MTRT. Here, SIT selects too few of the targets (it selects 4.2 out of the 38 targets on average in the different folds). The most likely cause is the large number of targets. SIT implements a greedy hill-climbing search that stops too early and ends up in a local optimum for this dataset.

Table 1. 10-fold cross validated Pearson correlation averaged over all targets and five runs. •,○ denote a statistically significant improvement or degradation of SIT or MTRT over STRT. Significance is determined by the corrected resampled t-test (Nadeau & Bengio, 2003) with significance level 0.01. ■,□ denote a statistically significant improvement or degradation of SIT over MTRT.

Dataset	STRT	SIT	MTRT
Sigmea	0.63±0.40	0.64±0.40	0.64±0.40
Soil 1	0.60±0.18	0.63±0.13•	0.64±0.13•
Soil 2	0.34±0.40	0.41±0.35•□	0.48±0.29•
Soil 3	0.19±0.23	0.24±0.23•	0.26±0.22•
Water 1	0.26±0.17	0.29±0.15•■	0.27±0.15
Water 2	0.37±0.27	0.41±0.25•■	0.39±0.23

4. Conclusions & Further Work

We proposed Selective Inductive Transfer (SIT), an algorithm that searches for the set of support targets that, when predicted together with the main target in a multi-target model, maximally improves the predictive performance with regard to the main target. Experiments show that, in all but one dataset, SIT finds a good set of support targets.

In further work, we plan to compare SIT to other methods that exploit transfer selectively and investigate alternative search strategies to the greedy hill-climbing approach implemented by SIT.

Acknowledgments: Beau Piccart is supported by project G.0255.08 "Efficient microprocessor design using machine learning" funded by the Research Foundation - Flanders (FWO-Vlaanderen). Jan Struyf and Hendrik Blockeel are post-doctoral fellows of the Research Foundation - Flanders (FWO-Vlaanderen).

References

Blockeel, H., De Raedt, L., & Ramon, J. (1998). Top-down induction of clustering trees. *15th Int'l Conf. on Machine Learning* (pp. 55–63).

Caruana, R. (1997). Multitask learning. *Mach. Learn.*, *28*, 41–75.

Nadeau, C., & Bengio, Y. (2003). Inference for the generalization error. *Mach. Learn.*, *52*, 239–281.

Silver, D., & Mercer, R. (1996). The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. *Connect. Sci.*, *8*, 277–294.

Thrun, S., & O'Sullivan, J. (1996). Discovering structure in multiple learning tasks: The TC algorithm. *13th Int'l Conf. on Machine Learning* (pp. 489–497).

Vision as Inference in a Hierarchical Markov Network

Justus Piater JUSTUS.PIATER@ULG.AC.BE
Fabien Scalzo FSCALZO@ULG.AC.BE
Renaud Detry RENAUD.DETRY@STUDENT.ULG.AC.BE
University of Lige, INTELSIG Group, Grande Traverse 10; 4000 Lige – Sart Tilman, Belgium

Abstract

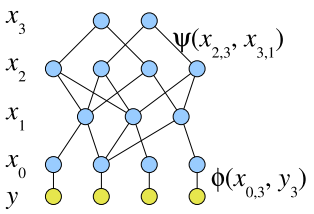
We present a snapshot of our current work on learning visual object models in the form of Markov networks, point out interesting relations to biological vision, and illustrate their performance on diverse tasks such as object recognition and 3D pose estimation.

1. Introduction

Cortical visual processing involves both bottom-up propagation of perceptual stimuli and modulation by top-down signals. Lee and Mumford (2003) suggested that the visual processing stream from the LGN via V1, V2 and V4 to IT might perform Bayesian inference within an undirected Markov chain. A cortical layer x_i (say, V1) computes its activity $P(x_i \mid x_{k\neq i}) = P(x_i \mid x_{i-1}, x_{i+1}) = P(x_{i-1} \mid x_i)P(x_i \mid x_{i+1})/Z_i$, that is, a posterior probability distribution given bottom-up input from x_{i-1} (LGN), under top-down priors x_{i+1} (V2). The parameters of the Markov network are specified via pairwise compatibility potentials ψ, where $P(x_i \mid x_{i-1}, x_{i+1}) = \psi(x_{i-1}, x_i)\psi(x_i, x_{i+1})/Z_i$. These potentials must be learned from experience with the world; Lee and Mumford (2003) do not comment on how this might be done. A crucial aspect of this model is that ambiguities at low levels should persist and propagate upwards until they can be resolved by integrating larger-scale evidence or top-down expectations. As a biologically plausible implementation of inference with arbitrary, possibly multi-modal probability densities, Lee and Mumford (2003) suggest belief propagation using particle representations. Moreover, they provide a wealth of neurophysiological and psychophysical evidence for such a computational model.

2. Principle

We are currently developing representations and methods for visual inference that constitute, at least at the vague level of detail given above, a working computer implementation of central aspects of Lee and Mumford's model. Without making explicit reference to specific cortical layers, our approach is based on a Markov network such as the didactic example shown in the figure, with vertices arranged in layers corresponding to those of Lee and Mumford's. Each vertex is a random variable representing the spatial probability density of the presence of a feature. At level 0, a primitive feature $x_{0,j}$ is the spatial probability density of a given type of locally observable feature. It is inferred from local image appearance y_j via its observation potential $\phi(x_{0,j}, y_j)$. At higher levels, a compound feature (recursively) represents the presence of both of its children, and the compatibility potentials ψ represent pairwise spatial relationships. For example, in the figure, feature $x_{3,1}$ represents the spatial probability density of features $x_{2,1}$ and $x_{2,3}$ occurring in the relative configuration encoded by $\psi(x_{2,1}, x_{3,1})$ and $\psi(x_{2,3}, x_{3,1})$.



We construct such networks, including their topology and compatibility potentials, using unsupervised learning. The input layer y, fixed at the outset, is successively exposed to visual stimuli. The system records the occurrences of known features (primitive or compound), as well as the spatial relations between them. When reoccurring constellations of features $x_{i,a}$ and $x_{i,b}$ are detected, the observed non-uniform spatial co-occurrence probability densities are turned into the compatibility potentials $\psi(x_{i,a}, x_{i+1,c})$ and $\psi(x_{i,b}, x_{i+1,c})$ with respect to a newly-instantiated feature $x_{i+1,c}$, located between them.

3. Results

In practice, we arrive at networks of on the order of 10 layers and on the order of between 10 and 100 vertices per layer. What they represent depends on the training data. For example, we have trained networks that represent individual objects, from a fixed viewpoint or from a variety of viewpoints. If a network is trained on images of several distinct objects, higher-level vertices will specialize to become view-tuned cells of specific objects. If objects share common parts, these are likely to be represented by the same lower-level subgraphs.

Networks learned in this way can be instantiated on a given input stimulus by computing the observations y, optionally instantiating higher-level vertices according to prior expectations, and performing nonparametric belief propagation using particle representations (similar to Sudderth et al., 2003) throughout the network until convergence. Thanks to bidirectional propagation, the network converges to a globally coherent interpretation of the scene, where each vertex $x_{i,j}$ contains its best possible interpretation of its children, under the priors provided by the parents. We have used this procedure successfully for object detection, recognition, and pose estimation, in 2D and in 3D, as well as for inference of occluded object parts.

Acknowledgments

This work was supported by the Belgian National Fund for Scientific Research (FNRS) and by the EU Cognitive Systems project IST-FP6-IP-027657 PACO-PLUS.

References

Lee, T. S., & Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America*, *20*, 1434–1448.

Sudderth, E. B., Ihler, A. T., Freeman, W. T., & Willsky, A. S. (2003). Nonparametric Belief Propagation. *Computer Vision and Pattern Recognition* (pp. 605–612).

Semi-supervised Classification in Graphs using Bounded Random Walks

Semi-supervised learning, large graphs, betweenness measure, passage times

Jérôme Callut JEROME.CALLUT@UCLOUVAIN.BE
Kevin Fançoisse KEVIN.FRANCOISSE@UCLOUVAIN.BE
Marco Saerens MARCO.SAERENS@UCLOUVAIN.BE
UCL Machine Learning Group (MLG)
Louvain School of Management, IAG,
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

Pierre Dupont PIERRE.DUPONT@UCLOUVAIN.BE
UCL Machine Learning Group (MLG)
Department of Computing Science and Engineering, INGI,
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

Abstract

This paper describes a novel technique, called $\mathcal{D}$-walks, to tackle semi-supervised classification problems in large graphs. We introduce here a betweenness measure based on passage times during random walks of bounded lengths in the input graph. The class of unlabeled nodes is predicted by maximizing the betweenness with labeled nodes. This approach can deal with directed or undirected graphs with a linear time complexity with respect to the number of edges and the maximum walk length considered. Preliminary experiments on the CORA database show that $\mathcal{D}$-walks outperforms NetKit (Macskassy & Provost, 2007) as well as Zhou et al. algorithm (Zhou et al., 2005), both in classification rate and computing time.

1. Introduction

This paper is concerned with semi-supervised classification of nodes in a graph. Given an input graph with some nodes being labeled, the problem is to predict the missing node labels. This problem has numerous applications such as classification of individuals in social networks, linked documents categorization or protein function prediction, to name a few.

Several approaches have been proposed to tackle semi-supervised classification problems in graphs. Kernel methods (Zhou et al., 2005; Tsuda & Noble, 2004) embed the nodes of the input graph into an Euclidean feature space where a classifier, such as a SVM, can be estimated. Despite of their good predictive performance, these techniques cannot easily scale up to large problems due to their high time complexity. NetKit is an alternative relational learning approach (Macskassy & Provost, 2007). It has a lower computational complexity but is less simple conceptually and may require to fine tune several of its components.

The approach proposed in this paper, called $\mathcal{D}$-walks, relies on random walks performed on the input graph seen as a Markov chain. More precisely, a betweenness measure, based on passage times during random walks of bounded length, is derived for each class (or label category). Unlabeled nodes are assigned to the category for which the betweenness is the highest. The $\mathcal{D}$-walks approach has the following properties: (i) it has a linear time complexity with respect to the number of edges and the maximum walk length considered; such a low complexity allows to deal with very large graphs, (ii) it can handle directed or undirected graphs, (iii) it can deal with multi-class problems and (iv) it has a unique hyper-parameter that can be tuned efficiently.

2. Discriminative random walks

We are given an input graph $\mathcal{G}$ containing a set of nodes $\mathcal{N}$ and edges $\mathcal{E}$. The (possibly weighted) adjacency matrix is denoted A. The graph $\mathcal{G}$ is assumed partially labeled. The nodes in the *labeled set* $\mathcal{L} \subset \mathcal{N}$ are assigned to a category from a discrete set $\mathcal{Y}$. The *unlabeled set* is defined as $\mathcal{U} = \mathcal{N} \setminus \mathcal{L}$.

Random walks in a graph can be modeled by a

discrete-time Markov chain (MC) describing the sequence of nodes visited during the walk. Each state of the Markov chain corresponds to a distinct node of the graph. The MC transition probability matrix is simply given by $P = D^{-1}A$, with D the diagonal matrix of node degrees. We consider *discriminative random walks* ($\mathcal{D}$-walks, for short) in order to define a betweenness measure used for classifying unlabeled nodes.

Definition 1 ($\mathcal{D}$-walk) *Given a MC defined on the state set $\mathcal{N}$, a class $y \in \mathcal{Y}$ and a discrete length $l > 1$, a $\mathcal{D}$-walk is a sequence of state $q_0, \ldots, q_l$ such that $y_{q_0} = y_{q_l} = y$ and $y_{q_t} \neq y$ for all $0 < t < l$.*

The notation $\mathcal{D}_l^y$ refers to the set of all $\mathcal{D}$-walks of length l, starting and ending in a node of class y. We also consider $\mathcal{D}_{\leq L}^y$ referring to all $\mathcal{D}$-walks *up to* a given length L. The betweenness function $B_L(q, y)$ measures how much a node $q \in \mathcal{U}$ is located "between" nodes of class $y \in \mathcal{Y}$. The betweenness $B_L(q, y)$ is formally defined as the expected number of times the node q is reached during $\mathcal{D}_{\leq L}^y$-walks.

Definition 2 ($\mathcal{D}$-walk betweenness) *Given an unlabeled node $q \in \mathcal{U}$ and a class $y \in \mathcal{Y}$, the* $\mathcal{D}$-walk betweenness function $\mathcal{U} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ *is defined as follows: $B_L(q, y) \triangleq \mathbb{E}\left[\mathrm{pt}(q) \mid \mathcal{D}_{\leq L}^y\right]$, where $\mathrm{pt}(q)$ is the passage times function $\mathcal{N} \rightarrow \mathbb{R}^+$ counting the number of times a node q has been visited.*

This betweenness measure is related to the one proposed by Newman in (Newman, 2005). Our measure is however relative to a specific class y rather than to the whole graph. It also considers random walks up to a given length instead of unbounded walks. Bounding the walk length has two major benefits: (i) better classification results are generally obtained with respect to unbounded walks (ii) the betweenness measure can be computed very efficiently (in $\Theta(|\mathcal{E}|L)$) using forward and backward recurrences, similar to those used in the Baum-Welch algorithm for HMM parameter estimation. Finally, an unlabeled node $q \in \mathcal{U}$ is assigned to the class with the highest betweenness.

3. Experiments

We report here preliminary experiments performed on the `Cora` dataset (Macskassy & Provost, 2007) containing 3582 nodes classified under 7 categories. As this graph is fully labeled, node labels were randomly removed and used as test set. More precisely, we have considered 9 different proportions of labeled nodes in the graph: $\{0.1, 0.2, \ldots, 0.9\}$ and for each labeling rate, 10 random deletions were performed. Comparative performances obtained with NetKit (Macskassy & Provost, 2007) and with the approach of Zhou et al. (Zhou et al., 2005) are also provided. The hyper-parameters of each approach have been tuned using ten-fold cross-validation. Figure 1 shows the correct classification rate on test data obtained by each approach for increasing labeling rates. The $\mathcal{D}$-walk approach clearly outperforms its competitors on these data. The $\mathcal{D}$-walks approach is also the fastest method. It requires typically 1.5 seconds of CPU[1] for every graph classification including the auto-tuning of its hyper-parameter L. NetKit takes about 4.5 seconds per graph classification and our implementation of Zhou et al. approach typically takes several minutes. Large graphs (several millions of edges) were also successfully classified in a few minutes with $\mathcal{D}$-walks while neither NetKit nor Zhou et al. methods could be applied on such large graphs.

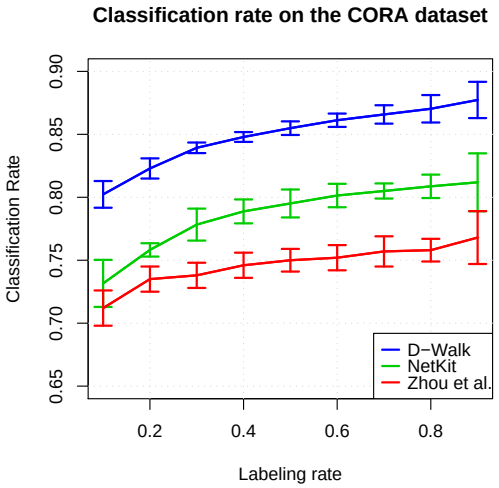


Figure 1. Classification rate of D-walk and two competing methods on the Cora dataset. Error bars report standard deviations over 10 independent runs.

References

Macskassy, S. A., & Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, *8*, 935–983.

Newman, M. (2005). A measure of betweenness centrality based on random walks. *Social networks*, *27*, 39–54.

Tsuda, K., & Noble, W. S. (2004). Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, *20*, 326–333.

Zhou, D., Huang, J., & Schölkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. *ICML '05: Proceedings of the 22nd international conference on Machine learning* (pp. 1036–1043). New York, NY, USA: ACM.

[1]Intel Core 2 Duo 2.2Ghz with 2Gb of virtual memory.

The Sum-Over-Paths Covariance: A novel covariance measure between nodes of a graph

Graph mining, kernel on a graph, correlation measure, semi-supervised classification.

Amin Mantrach AMANTRAC@ULB.AC.BE
Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA-CoDE)
Université Libre de Bruxelles, B-1050 Brussels, Belgium

Marco Saerens MARCO.SAERENS@UCLOUVAIN.BE
Luh Yen LUH.YEN@UCLOUVAIN.BE
UCL Machine Learning Group (MLG), Louvain School of Management,
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

Abstract

This work introduces a link-based covariance measure between the nodes of a weighted, directed, graph where a cost is associated to each arc. To this end, a probability distribution on all possible paths through the network is defined by minimizing the sum of the expected costs between all pairs of nodes while fixing the total relative entropy spread in the network. This results in a probability distribution on the set of paths such that long paths occur with a low probability while short paths occur with a high probability. The covariance measure is then computed according to this probability distribution: two nodes will be highly correlated if they often co-occur together on the same – preferably short – paths. The resulting covariance matrix between nodes (say n in total) is a Gram matrix and therefore defines a valid kernel matrix on the graph; it is obtained by inverting a $n \times n$ matrix. The proposed model could be used for various graph mining tasks such as computing betweenness centrality, semi-supervised classification, visualization, etc, as shown in the experimental section.

1. The sum-over-paths covariance measure

Basic notations and definitions. Consider a weighted directed graph or network, G, with a set of n nodes V (or vertices) and a set of arcs E (or edges). The graph is supposed to be strongly connected. To each arc linking node k and node k', a number $c_{kk'} > 0$ is associated, representing the **immediate cost** of following this arc. The **cost matrix C** is the matrix containing the immediate costs $c_{kk'}$. In a first step, a **random walk** on this graph will be defined. The choice to follow an arc will be made according to transition probabilities representing the probability of jumping from a node k to another node k' belonging to the set $S(k)$ of neighboring nodes (successors S) that can be reached from node k. The transitions probabilities defined on each node k will be denoted as $p_{kk'} = \mathrm{P}(k'|k)$ with $k' \in S(k)$. Furthermore, $\mathbf{P}$ will be the matrix containing the transition probabilities $p_{kk'}$ as elements. If there is no arc between k and k', we simply consider that $c_{kk'}$ takes a large value, denoted by ∞; in this case, the corresponding transition probability will be set to zero, $p_{kk'} = 0$. The *natural random walk* on the graph will be defined by $p^{\mathrm{ref}}_{kk'} = c^{-1}_{kk'} / \sum_{k'} c^{-1}_{kk'}$ and the corresponding transitions-probabilities matrix $\mathbf{P}^{\mathrm{ref}}$. In other words, in this natural random walk, the random walker chooses to follow a link with a probability proportional to the inverse of the immediate cost, therefore favoring links having a low cost. These transition probabilities will be used as reference probabilities later; hence the superscript *ref*.

Definition of the probability distribution on the set of paths. Let us first consider two nodes, an initial node i and a destination node j. We define the (possibly infinite) set of paths (including cycles) connecting these two nodes as $\mathcal{R}_{ij} = \{\wp_{r^{ij}}\}$. Thus, $\wp_{r^{ij}}$ is path number r^{ij}, with path index r^{ij} ranging from 1 to ∞. Let us further define the set of all paths $\mathcal{R} = \bigcup_{ij} \mathcal{R}_{ij}$ and a probability distribution on this set $\mathcal{R}$ representing the probability $\mathrm{P}(\wp_{r^{ij}})$ of following the path numbered r^{ij}. The main idea will be to use the probability distribution $\mathrm{P}(\wp_{r^{ij}})$ *minimizing the expected cost-to-go among all the probability distributions having a fixed relative entropy with respect to the natural random walk on the graph.*

Let us also denote as $E_{r^{ij}}$ the total cost associated to the path r^{ij}, referred to as the **energy** associated to that path. We assume that the total cost associated to a path is additive, i.e. $E(\wp_{r^{ij}}) = \sum_{t=1}^{t_f} c_{k_{t-1}k_t}$ where $k_0 = i$ is the initial state and $k_{t_f} = j$ is the destination state; t_f is the time (number of steps) needed to reach node j. Here, we assume that $\wp_{r^{ij}}$ is a valid path from the initial state to the destination state, that is, every $c_{k_{t-1}k_t} \neq \infty$ along that path.

We now have to find the path probabilities minimizing the sum of the expected energy for reaching node j when starting from i. In other works, we are seeking path probabilities, $\mathrm{P}(\wp_{r^{ij}})$, minimizing $\sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}(\wp_{r^{ij}})E(\wp_{r^{ij}})$ subject to the constraint $-\sum_{r^{ij}=1}^{\infty}\mathrm{P}(\wp_{r^{ij}})\ln(\mathrm{P}(\wp_{r^{ij}})/\mathrm{P}^{\mathrm{ref}}(\wp_{r^{ij}})) = J_0$ where $\mathrm{P}^{\mathrm{ref}}(\wp_{r^{ij}})$ represents the probability of following the path $\wp_{r^{ij}}$ when walking according to the natural random walk, i.e. when using transition probabilities $p^{\mathrm{ref}}_{kk'}$. Here, J_0 is provided a priori by the user, according to the desired degree of randomness he is willing to concede. By defining the Lagrange function

$$\begin{aligned}\mathcal{L} &= \sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}(\wp_{r^{ij}})E(\wp_{r^{ij}}) \\ &+ \lambda\left[\sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}(\wp_{r^{ij}})\ln\frac{\mathrm{P}(\wp_{r^{ij}})}{\mathrm{P}^{\mathrm{ref}}(\wp_{r^{ij}})} + J_0\right] \\ &+ \mu\left[\sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}(\wp_{r^{ij}}) - 1\right], \end{aligned} \quad (1)$$

we obtain the following probability distribution

$$\mathrm{P}(\wp_{r^{ij}}) = \frac{\mathrm{P}^{\mathrm{ref}}(\wp_{r^{ij}})\exp\left[-\theta E(\wp_{r^{ij}})\right]}{\sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}^{\mathrm{ref}}(\wp_{r^{ij}})\exp\left[-\theta E(\wp_{r^{ij}})\right]} \quad (2)$$

where $\theta = 1/\lambda$. Thus, as expected, short paths (having small $E(\wp_{r^{ij}})$) are favoured in that they have a large probability of being followed. When $\theta \to \infty$, only shortest paths are considered in $\mathcal{R}$ while when $\theta \to 0$ all paths corresponding to the natural random walk are taken into account.

Definition of the covariance measure. We now show that the sum-over-paths covariance measure can be computed from a key quantity, defined as

$$Z = \sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}^{\mathrm{ref}}(\wp_{r^{ij}})\exp\left[-\theta E(\wp_{r^{ij}})\right], \quad (3)$$

which corresponds to the **partition function** in statistical physics [2]. It can be shown that the partition function can easily be computed from the cost matrix by inverting a matrix [4].

Indeed, the expected number of times the link $k \to k'$ and the link $l \to l'$ are traversed together along a path can be computed by taking the second-order derivative

$$\begin{aligned}\overline{\eta}(k,k';l,l') &= \frac{1}{\theta^2}\frac{\partial^2(\ln Z)}{\partial c_{ll'}\partial c_{kk'}} \\ &= \sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}(\wp_{r^{ij}})\,\delta(r^{ij};k,k')\delta(r^{ij};l,l') \\ &- \left[\sum_{i,j=1}^{n}\sum_{r^{ij}=1}^{\infty}\mathrm{P}(\wp_{r^{ij}})\,\delta(r^{ij};k,k')\right]^2 \end{aligned} \quad (4)$$

where $\delta(r^{ij};k,k')$ indicates the number of times the link $k \to k'$ is present in path number r^{ij}, and thus the number of times the link is traversed. This last quantity clearly corresponds to the **covariance** between link $k \to k'$ and link $l \to l'$.

Now, the **covariance measure** between node k' and node l' is simply defined as

$$\mathrm{cov}(k',l') = \sum_{k,l=1}^{n}\overline{\eta}(k,k';l,l') \quad (5)$$

which corresponds to the main quantity of interest. This quantity can thus easily be computed from the partition function (the calculus are similar to the one apprering in [4]).

2. Preliminary experiments

Notice that the experiments investigating the betweenness and the visualization capabilities are not reported in this paper because of the lack of space.

Semi-supervised classification. Preliminary experiments on semi-supervised classification have been perfomed on the IMDb dataset (described in [3]). We compare the proposed sum-over-paths kernel to the regularization kernel proposed by [5], the commute-time kernel [1] and the diffusion map kernel [1] in a semi-supervised classification task of unlabeled nodes. An alignment procedure is used in order to classify the unlabeled nodes, as described in [5], for each of the four studied kernels. The hyperparameters of each algorithm have been tunned by using a 5-fold cross-validation and the best value has been retained for the estimation of the classification rate on the test set. In order to reduce random effects on our results, the labeled nodes have been sampled randomly 10 times, and averaged results are finally reported on these ten runs.

The results (ommitted because of the lack of space) show that the sum-over-paths kernel provides competitive results in comparison with the other standard kernels on a graph.

References

[1] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation. *Proceedings of the 2005 IEEE/WIC/ACM International Joint Conference on Web Intelligence*, pages 550–556, 2005.

[2] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[3] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8:935–983, 2007.

[4] M. Saerens, Y. Achbany, F. Fouss, and L. Yen. Randomized shortest-path problems: Two seemingly unrelated problems. *Manuscript submitted for publication*, 2008.

[5] D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. *Proceedings of the 22nd International Conference on Machine Learning*, pages 1041–1048, 2005.

Classification on incomplete data using sparse representations: Imputation is optional

Jort Gemmeke J.Gemmeke@let.ru.nl
Dept. of Linguistics, Radboud University,
P.O. Box 9103, NL-6500 HD, Nijmegen, The Netherlands

Abstract

We present a non-parametric technique capable of performing classification directly on incomplete data, optionally performing imputation. The technique works by sparsely representing the available data in a basis of example data. Experiments on a spoken digit classification task show significant improvement over a baseline missing-data classifier.

1. Introduction

Classification on incomplete data is a challenging task because parametric techniques require that the dimensionality of the data doesn't change between training and classification while non-parametric techniques which can handle incomplete data such as k-nearest-neighbors often deliver suboptimal classification accuracies. In practice, the missing data is often estimated prior to classification through *imputation*. Most imputation methods estimate the missing coefficients based on local information and/or do not fully exploit the structure of the underlying signal. Based on work in *Compressed Sensing* (Donoho, 2006; Candes, 2006) we present a non-parametric method which can not only perform classification directly on the available data but optionally imputes the missing data. The method is based on the premise that a signal can be sparsely represented in a basis of example signals (Yang et al., 2007) and that this sparse representation can be exactly recovered even if only a small part of the data is available (Zhang, 2006). We show the effectiveness of this approach on a spoken digit classification task.

2. Method

2.1. Sparse representation

We consider observation vector $\boldsymbol{y}$ of unknown class and dimensionality K to be a linear combination of feature vectors $\boldsymbol{d}_{i,n}$, where the first index ($1 \leq i \leq I$) denotes one of I classes and the second index ($1 \leq n \leq N_i$) a specific exemplar vector of class i with N_i the number of examples in that class. We write:

$$\boldsymbol{y} = \sum_{i=1}^{I} \sum_{n=1}^{N_i} \alpha_{i,n} \boldsymbol{d}_{i,n}$$

with weights $\alpha_{i,n} \in \mathbb{R}$. The set of exemplars span a $K \times N$ dimensional basis ($N = N_1 + N_2 + \ldots + N_I$):

$$A = (d_{1,1} \ldots d_{1,N_1} \ldots d_{I,1} \ldots d_{I,N_I}) \quad (1)$$

Thus, we can express $\boldsymbol{y}$ as:

$$\boldsymbol{y} = A\boldsymbol{x} \quad (2)$$

with $\boldsymbol{x}$ an N-dimensional vector that ideally will be sparsely represented as $\boldsymbol{x} = [0 \ldots 0 \ \alpha_{i,1}\alpha_{i,2} \ldots \alpha_{i,N_i} \ 0 \ldots 0]^T$ (i.e., most coefficients not associated with class i are zero).

Taking into account that the observation vector $\boldsymbol{y}$ is incomplete we denote its available coefficients by $\boldsymbol{y}_a$ and the missing coefficients $\boldsymbol{y}_m$. Now we can solve the system of linear equations of Eq. 2 using only the available coefficients $\boldsymbol{y}_a$ and the basis A_a, formed by only retaining the rows of A indicated by the available coefficients. Research in the field of *compressed sensing* (Donoho, 2006; Candes, 2006) has shown that if $\boldsymbol{x}$ is sparse, $\boldsymbol{x}$ can be recovered exactly by solving:

$$min||\boldsymbol{x}||_1 \text{ subject to } ||\boldsymbol{y_a} - A_a\boldsymbol{x}||_2 \leq \epsilon \quad (3)$$

with a small constant ϵ such that the error $\boldsymbol{e}$ satisfies $||\boldsymbol{e}||_2 < \epsilon$ and $||.||_1$ the l^1norm.

2.2. Sparse classification (SC)

Following (Yang et al., 2007), we perform classification by comparing the *support* of $\boldsymbol{y_a}$ in parts of A_a associated with different classes i. In other words, we compare how well the various parts of $\boldsymbol{x}$ associated with different classes i can reproduce $\boldsymbol{y_a}$. The reproduction error is called the *residual*. The residual of

class i is calculated by setting the coefficients of $\boldsymbol{x}$ not associated with i to zero while keeping the coefficients associated with i unchanged. Thus the residual is:

$$r_i(\boldsymbol{y_r}) = ||\boldsymbol{y_a} - A_a\delta_i(\boldsymbol{x})||_2 \quad (4)$$

with $\delta_i(\boldsymbol{x})$, the vector selecting only the columns of A that correspond to class i. The class c that is assigned to an observed vector $\boldsymbol{y}$ is the one that gives rise to the smallest residual:

$$c = \underset{i}{\text{argmin}}\ r_i(\boldsymbol{y_r}). \quad (5)$$

2.3. Sparse imputation (SI)

Alternatively, one can use the sparse representation $\boldsymbol{x}$ to impute the missing coefficients. Without loss of generality we reorder $\boldsymbol{y}$ and A as in (Zhang, 2006) so that we can write:

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_a \\ \boldsymbol{y}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_a \\ A_m\boldsymbol{x} \end{bmatrix} \quad (6)$$

with A_m pertaining to the rows of A indicated by the missing coefficients in $\boldsymbol{y}$ and $\boldsymbol{y}_i$ an estimate for the missing coefficients $\boldsymbol{y_m}$. This yields a new observation vector $\hat{\boldsymbol{y}}$ after which ordering can be restored.

3. Experiments

We apply the described method to missing data spoken digit classification task (AURORA-2). In noisy speech, with digits represented by fixed length observation vectors, coefficients are considered missing if their values (representing speech energy in a time-windowed spectrographic representation) are dominated by speech energy rather than noise energy. We explore the effectiveness of our approach by selecting the missing coefficients using knowledge of the true speech and noise signals. Using a setup described in detail in (Gemmeke & Cranen, 2008) we compare the sparse classification technique with a baseline, state-of-the-art, HMM-classifier which performs imputation on a frame-by-frame basis (Van hamme, 2006). Additionally we compare classification accuracies obtained by combining *sparse imputation* and the baseline classifier.

While not strictly linear as a function of signal-to-noise ratio (SNR), the percentage missing coefficients ranges from 0% (clean speech) to $80-95$% (at SNR -5 dB). In Table 1 it is shown that the *sparse classification* and *sparse imputation* methods significantly outperform the baseline, frame-based classifier.

Table 1. AURORA-2 single digit classification accuracy.

	SNR					
method	clean	15	10	5	0	-5
Baseline	99.3	99.1	98.7	96.6	88.4	61.0
SC	98.4	98.4	98.0	97.5	95.8	91.0
SI	99.3	99.0	98.5	97.7	96.5	91.3

4. Discussion and conclusions

Results show that both sparse methods give considerable improvement over the baseline, suggesting that a correct sparse representation can be found even when the majority of the data is missing, provided the redundancy in the structure of the data is exploited by use of example whole-digit observations vectors. The slightly better results using sparse imputation rather than sparse classifications seem to suggest that the sparse classification method does not generalize to observed digits as well as the HMM-based (parametric) approach.

Acknowledgments

The research of Jort Gemmeke was carried out in the MIDAS project, granted under the Dutch-Flemish STEVIN program.

References

Candes, E. J. (2006). Compressive sampling. *Proceedings of the International Congress of Mathematicians.*

Donoho, D. L. (2006). For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics*, *59*, 907–934.

Gemmeke, J., & Cranen, B. (2008). Noise robust digit recognition using sparse representations. *accepted for ISCA ITWR 2008.*

Van hamme, H. (2006). Handling time-derivative features in a missing data framework for robust automatic speech recognition. *Proceedings of IEEE ICASSP.*

Yang, A. Y., Wright, J., Ma, Y., & Sastry, S. S. (2007). Feature selection in face recognition: A sparse representation perspective. *submitted to IEEE Transactions Pattern Analysis and Machine Intelligence.*

Zhang, Y. (2006). When is missing data recoverable? *Technical Report.*

Multi-Agent State Space Aggregation using Generalized Learning Automata

Yann-Michaël De Hauwere YDEHAUWE@VUB.AC.BE
Peter Vrancx PVRANCX@VUB.AC.BE
Ann Nowé ANOWE@VUB.AC.BE
Computational Modeling Lab, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium

Abstract

A key problem in multi-agent reinforcement learning remains dealing with the large state spaces typically associated with realistic distributed agent systems. As the state space grows, agent policies become more and more complex and learning slows. One possible solution for an agent to continue learning in these large-scale systems, is to learn a policy which generalizes over states, rather than trying to map each individual state to an action. In this paper we present a multi-agent learning approach capable of aggregating states, using associative reinforcement learners called generalized learning automata (GLA).

1. Introduction

Reinforcement learning (RL) has already been shown to be a powerful tool for solving single agent Markov Decision Processes (MDPs). Basic RL techniques are not suited for problems with very large state spaces, however, as they mostly rely on a tabular representation for policies and enumerating all possible state-action pairs is not feasible (the so called *curse of dimensionality*). Because of these issues, several extensions have been proposed to reduce the complexity of learning. Methods for representing the agent's policy such as neural networks, decision trees and other regression techniques are already widely used. To our understanding, relatively little work has been done on extending RL for large state spaces to MAS, so far.

2. Generalized Learning Automata

A Generalized Learning Automaton (GLA) is an associative reinforcement learning unit. The purpose of a GLA is to learn a mapping from given inputs or contexts to actions. At each time step the GLA receives an input which describes the current system state. Based on this input and its own internal state the unit then selects an action. This action serves as input to the environment, which in turn produces a response for the GLA. Based on this response the GLA then updates its internal state.

Formally a GLA can be represented by a tuple (X, A, β, u, g, T), where X is the set of possible inputs to the GLA and $A = \{a_1, \ldots, a_r\}$ is the set of outputs or actions the GLA can produce. $\beta \in [0, 1]$ denotes the feedback the automaton receives for an action. The real vector u represents the internal state of the unit. It is used in conjunction with a probability generating function g to determine the action probabilities, given an input $x \in X$. T is a learning algorithm which updates $\mathbf{u}$, based on the current value of $\mathbf{u}$, the given input, the selected action and response β. In this paper we use a modified version of the REINFORCE (WILLIAMS, 1992) update scheme. In (Thathachar & Sastry, 2004) it is shown, that this update mechanism, converges to local maxima of $f(\mathbf{u}) = E[\beta|\mathbf{u}]$, showing that the automata find a local maximum over the mappings that can be represented by the internal state in combination with the function g.

We propose to use the GLA described above in Multi-agent Reinforcement learning problems. In such a system each agent internally uses a set of GLA to learn the different regions in the state space where different actions are optimal. We use the following set-up for the GLA. With every action $a_i \in A$ the automaton can perform, it associates a vector $\mathbf{u_i}$. This results in an internal state vector $\mathbf{u} = [\mathbf{u_1}^\tau \ldots \mathbf{u_r}^\tau]$ (where τ denotes the transpose). With this state vector we use the Boltzmann distribution as probability generating function:

$$g(x, a_i, u) = \frac{e^{\frac{x^\tau u_i}{T}}}{\sum_j e^{\frac{x^\tau u_j}{T}}} \quad (1)$$

Of course since this function is fixed in advance and the environment in general is not known, we have no guarantee that the GLA can represent the optimal mapping. For instance when using the function given in Equation 1 with a 2-action GLA, the internal state vector represents a hyperplane. This plane separates context vectors which give a higher probability to action 1 from those which action 2. If the sets of context vectors where different actions are optimal, are not linearly separable the GLA cannot learn an optimal mapping.

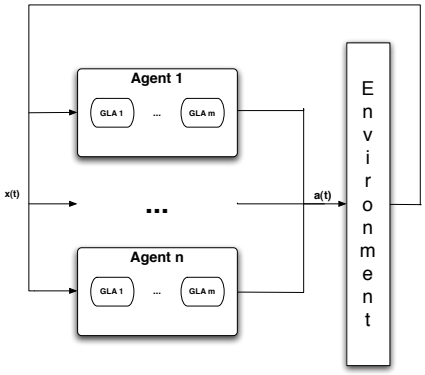


Figure 1. Learning set-up. Each agent receives factored state representation as input. GLA decide action to be performed.

To allow a learner to better represent the desired mapping from context vectors to actions, we can utilize systems composed of multiple GLA units. For instance the output of multiple 2-action GLAs can be combined to allow learners to build a piecewise linear approximation of regions in the space of context vectors. In general, we can use systems which are composed of feedforward structured networks of GLA. In these networks, automata on one level use actions of the automata on the previous level as inputs . If the feedforward condition is satisfied, meaning that the input of a LA does not depend on its own output, convergence to local optima can still be established (Phansalkar & Thathachar, 1995).

Figure 1 shows the general agent learning set-up. Each time step t a vector $\mathbf{x(t)}$ giving a factored representation of the current system state is generated. This vector is given to each individual agent as input. The agents internally use a set of GLA to map an action to the current state. The joint action $\mathbf{a(t)}$of all agents

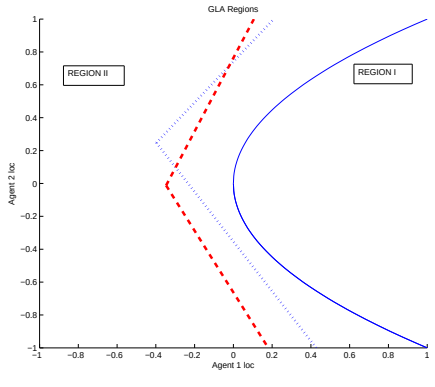


Figure 2. Typical results for approximations for parabola learnt by agents.

serves as input to the environment, which responds with a feedback $\beta(t)$ that agents use to update the GLA.

3. Experiments

In a first simple experiment 2 agents move around randomly in a 2 dimensional state space. Each time step both agents receive the current state (x, y)-values as input. The agents have to learn which joint actions are optimal in different regions of the state space. In this case there are 2 regions determined by a parabola. In region I, given by the inside of the parabola action (a_1, a_1) is optimal with a reward of 0.9. When the joint location of the agents falls outside the parabola, however, action (a_2, a_2) is optimal with reward 0.5. In both cases all other joint actions have a pay-off of 0.1.

Both agents use a system consisting of 2 GLA, connected by an AND operation. Both GLA have 2 actions: 0 and 1. If the automata both choose 1 the agents performs its first action a_1 else it performs action a_2. Figure 2 shows typical results for approximations that the agents learn for the parabola.

References

Phansalkar, V., & Thathachar, M. (1995). Local and global optimization algorithms for generalized learning automata. *Neural Computation*, *7*, 950–973.

Thathachar, M., & Sastry, P. (2004). *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Pub.

WILLIAMS, R. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Reinforcement Learning*, *8*, 229–256.

Efficiently learning timed models from observations

Sicco Verwer, Mathijs de Weerdt, and Cees Witteveen S.E.VERWER@TUDELFT.NL
Delft University of Technology, P.O. Box 5031, 2600 GA, Delft, the Netherlands

1. Learning timed models efficiently

This paper describes an *efficient* algorithm for learning a *timed* model from *observations*. The algorithm is based on the *state merging* method for learning a deterministic finite state automaton (DFA). This method and its problem have been the subject of many studies within the grammatical inference field, see e.g. (de la Higuera, 2005). Consequently, it enjoys a sound theoretical basis which can be used to prove properties of our algorithm. For example, while it has long been known that learning DFAs is NP-complete, it has been shown that DFAs can be learned in the limit from polynomial time and data (efficiently in the limit) using a state merging method.

A DFA is a language model. A language is a set of finite sequences of symbols $\sigma = s_1 s_2 \dots s_n$ known as strings. Adding time to a string can be done by giving every symbol a time value $t \in \mathbb{N}$[1], resulting in a sequence of symbol-time value pairs $\tau = (s_1, t_1)(s_2, t_2) \dots (s_n, t_n)$. Every time value in such a *timed string* represents the time that has elapsed since occurrence of the previous symbol. A set consisting of timed strings is called a *timed language*. There are two main reasons why we want to learn a timed model instead of an untimed model. First, in many applications the data that is obtained by observing a system contains timestamps. For example, this is the case when time series data is obtained from sensors. Second, we believe that learning a timed model from such data is more efficient than learning an untimed model from this data. The reason is that, while it is possible to construct for any timed model an equivalent untimed model by *sampling* the time values, this untimed model is of size exponential in the size of the timed model. Thus, an efficient algorithm that learns a timed system using an untimed model is by definition an *inefficient* algorithm since it is exponential in time and data in the size of the timed model. In contrast, we show it is possible to learn certain types of timed models efficiently. Naturally, we want to focus on learning models that are *efficiently learnable*.

We assume familiarity with the theory of languages and automata. The timed models we consider are known as timed automata (TA) (Alur & Dill, 1994). In these models, time is represented using a finite number of *clocks*. A clock can be thought of as a stopwatch that measures the time since it was last reset. When a transition of a TA executes (fires) it can optionally reset the value of a clock. This clock then measures the time since the last execution of this transition. The value of clocks are used to constrain the execution of transitions in a TA. A transition in a TA can contain a boolean constraint, known as a *clock guard*, that specifies when this transition is allowed to be executed depending on the values of clocks. If the clock values satisfy the constraint, the transition can be executed. Otherwise, it cannot be executed. Thus a transition is executed only if the TA is currently in its source (or parent) state, the current symbol is equal to the transition label, and the current clock values satisfy the transitions clock guard. In this way, the execution of a TA depends not just on the *symbols*, but also on the *time values* occurring in a timed string.

We focus on learning algorithms for a simple timed model known as a deterministic real-time automaton (DRTA) (Dima, 2001) based on the state merging method. A DRTA is a TA that contains a single clock that is reset by every transition. Hence, the execution of a DRTA depends on the symbols, and on the time between two consecutive events of a timed string. The reason for restricting ourselves to these simple TAs is that the before-mentioned learnability results for DFAs can quite easily be adapted to the case of DRTAs. In other words, we can show that DRTAs are efficiently learnable using a state merging method. Due to the expressive power of clocks, we can also show this does not hold for deterministic TAs (DTAs) in general: any algorithm that tries to learn a DTA $\mathcal{A}$ will sometimes require an input set of size exponential in the size of $\mathcal{A}$. Other methods we know of for learning timed models try to learn a sub-

[1] It is more common to use $\mathbb{R}$ for time values. In practice, there always is a finite precision of time and hence this only makes the timed models unnecessarily complex.

class of DTAs known as event recording automata, see e.g. (Grinchtein et al., 2005). These automata are more powerful than DRTAs. However, due to this extra power an algorithm that learns these models is inefficient in the amount of data it requires.

We are currently finishing these (in)efficient learnability proofs. Our aim is to discover the most powerful timed model that is efficiently learnable.

2. Learning from observations

We constructed an algorithm for learning DRTAs from labeled data (Verwer et al., 2007). A high-level view of this algorithm is given in Algorithm 1. The algorithm starts with a prefix tree $\mathcal{A}$ that is constructed from the input set S. This is a DRTA that is such that: all the clock guards are equal to true (it disregards time values), there exists exactly one execution one path to any state (it is a tree), and it is such that the execution of $\mathcal{A}$ on any timed string from S ends in a state in $\mathcal{A}$. Starting from this prefix tree, our algorithm performs the most consistent merge or split as long as consistent merges or splits are possible. A merge of two states a and b replaces a and b with a new state c that contains all the input and output transitions of both a and b. Afterwards, it is possible that $\mathcal{A}$ contains some non-deterministic transitions. These are removed by continuously merging the target states of these transitions until none are left.

A transition splitting process is used to learn the clock constraints of the DRTA. A *split* of a transition d, at time t removes d from $\mathcal{A}$ and adds a transition e that is satisfied by all the clock values that satisfy d up to time t, and a transition f that is satisfied by all the clock values that satisfy d starting at t. The timed strings from S that have an execution path over e and f are subsets of the timed strings from S that had an execution path over d. Therefore, the prefix tree is recalculated starting from the new transitions e and f.

Algorithm 1 State merging and splitting DRTAs

Require: A set of timed strings S.
Ensure: The result is a small consistent DRTA $\mathcal{A}$.
  Construct a timed prefix $\mathcal{A}$ tree from S.
  while States can be merged or split consistently **do**
    Evaluate all possible merges and splits.
    Perform the most consistent merge or split.
  end while
  Return the constructed DRTA $\mathcal{A}$.

There is one important part of Algorithm 1 that we left undefined: What does it mean to be consistent? In the original setting of our algorithm the answer to this question was simple: The algorithm got a labeled data set as input and consistency was defined using these labels. However, for many applications this setting is unrealistic: Usually, only positive data is available. We adapted our algorithm to this setting by using statistics as a consistency measure for our models.

This seems a very natural thing to do and it has been done for the problem of learning probabilistic DFAs (Kermorvant & Dupont, 2002). The main problem to overcome is to come up with a good statistic for the (dis)similarity of two states based on the prefix trees of their suffixes. In our approach, two states in such a tree are treated as being similar if the distributions with which they generate (next) symbols are not significantly different. In addition, since we deal with a timed model, we require that the distributions generating the time values of these symbols are not significantly different. There properties are tested for every state in the prefix tree using Chi-square and Kolmogorov-Smirnov hypothesis tests. The result is many p-values, which we combine into a single test using a standard method for multiple hypothesis testing. The method that was used to learn DFAs does not make use of a multiple hypothesis testing method: it simply rejects if any of the tests fails.

Initial tests of our algorithm on artificial data show that the uses of timed models and multiple hypothesis testing are promising.

References

Alur, R., & Dill, D. L. (1994). A theory of timed automata. *Theoretical Computer Science*, *126*, 183–235.

de la Higuera, C. (1997). Characteristic sets for polynomial grammatical inference. *Machine Learning*, *27*, 125–138.

de la Higuera, C. (2005). A bibliographical study of grammatical inference. *Pattern Recognition*, *38*, 1332–1348.

Dima, C. (2001). Real-time automata. *Journal of Automata, Languages and Combinatorics*, *6*, 2–23.

Grinchtein, O., Jonsson, B., & Leucker, M. (2005). Inference of timed transition systems. *Electronic Notes in Theoretical Computer Science*.

Kermorvant, C., & Dupont, P. (2002). Stochastic grammatical inference with multinomial tests. *ICGI '02* (pp. 149–160). Springer-Verlag.

Verwer, S., de Weerdt, M., & Witteveen, C. (2007). An algorithm for learning real-time automata. *Benelearn'07* (pp. 128–135).

ProSOM: Core promoter identification in the human genome.

Thomas Abeel THOMAS.ABEEL@PSB.UGENT.BE
Yvan Saeys YVAN.SAEYS@PSB.UGENT.BE
Yves Van de Peer YVES.VANDEPEER@PSB.UGENT.BE
Department of Plant Systems Biology, VIB, Technologiepark 927, 9052 Gent, Belgium,
Department of Molecular Genetics, Ghent University, Technologiepark 927, 9052 Gent, Belgium

Abstract

More and more genomes are being sequenced, and to keep up with the pace of sequencing projects, automated annotation techniques are required. One of the most challenging problems in genome annotation is the identification of the core promoter. Better core promoter prediction can improve genome annotation and can be used to guide experimental work.

Comparing the average structural profile of transcribed, promoter and intergenic sequences demonstrates that the core promoter has unique features that cannot be found in other sequences. We show that unsupervised clustering by using self-organizing maps can clearly distinguish between the structural profiles of promoter sequences and other genomic sequences. An implementation of this promoter prediction program, called ProSOM, is available and has been compared with the state-of-the-art.

1. Introduction

Currently, the genomic sequence of over 50 eukaryotic organisms is available. So it is important to automate the identification of genes and regulatory sequences.

The core promoter is the region immediately upstream of the TSS, where the transcription initiation complex assembles.

Core promoters have distinct features that can be used to distinguish them from other sequences. One such property models the local base-stacking energy. High values denote regions that destack or melt easily. Two regions that seem to melt easily are located around -30 from the TSS and on the TSS, and are embedded in a large-scale region that is significantly more stable. We used this large-scale feature in earlier work to predict promoter regions in a wide range of species.

We present a novel promoter prediction technique, called ProSOM, that uses an unsupervised self-organizing map (SOM) to distinguish core promoter regions from the rest of the genome.

2. Material and methods

2.1. Data

We used the human genome assembly (hg17, May 2004). The cap analysis gene expression (CAGE) dataset was retrieved from the Fantom3 project. It contains 123,400 unique TSSs for human. The Ensembl gene annotation has been retrieved using the BioMart tool for Ensembl release 37. Sequences and annotation were retrieved from the ENCODE project. For the training of the SOM we retrieved promoter, transcribed and intergenic sequences from DBTSS and Ensembl.

2.2. Structural profiles

The nucleotide sequence is converted into a sequence of numbers (i.e., a numerical profile). This is done by replacing each dinucleotide with its energy value, which is obtained from experimentally validated conversion tables. We have used the conversion tables for base-stacking energy from Florquin et al. 2005.

2.3. Clustering and promoter prediction

The clustering technique we used is the self-organizing map (SOM), a special type of artificial neural network that can be used both for clustering and class prediction. A SOM consists of a rectangular grid of clusters, each of which has a weighted connection to every input node. In our case, the input nodes represent the different values of a structural profile associated to a potential promoter region. The SOM provides a

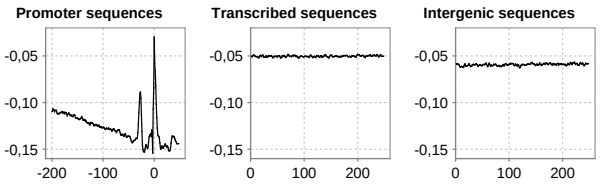


Figure 1. The structural profile of promoter (left), transcribed (center) and intergenic (right) human sequences. The profiles are the averages over all sequences in the respective training sets. We used the base-stacking energy as physical property. The left panel shows the region [-200,50] around the TSS, while for the other two panels there is no reference point and the location are numbered from 0 to 250.

mapping from a higher-dimensional feature space (the structural profile) to a lower-dimensional cluster space.

2.4. Validation

The validation of our technique was done on two sequence sets; first on the entire human genome assembly (hg17, May 2004) and secondly on the ENCODE regions. For both sets we retrieved a set of experimentally characterized TSSs and a gene annotation.

An aggregate measure for the performance of a classifier that is often used in the machine learning field is the F-measure. This is the harmonic mean of the recall (sensitivity) and the precision (specificity).

We have proposed a more objective way to assess the performance of a PPP based on the genome-wide screening for TSSs. This technique is based on the CAGE datasets that have been described earlier. The dataset contains locations where transcription starts. A TP is a known site that has a prediction within 50 bp of a true TSS, a FN is a TSS without a prediction and a FP is a prediction that has no associated TSS in the reference set within 50 bp.

3. Results

Figure 1 shows the average structural profile of base-stacking energy of the three datasets we used for training the SOM. The promoter sequences show a very striking profile with overall lower values than the other two graphs. It has two clear peaks at position -30 (TATA binding protein) and position 0 (TSS).

We used the trained SOM to predict promoter regions. To each cluster we attached a probability that a given sequence assigned to that cluster is a promoter. If the structural profile of a sequence maps to a cluster that has a probability equal to or above the threshold, we predict it as a promoter region.

Table 1. Evaluation of promoter prediction programs using the CAGE dataset with a maximum allowed distance of 50 bp.

program	recall	prec.	**F**
ProSOM	0.17	0.30	**0.22**
Eponine	0.14	0.35	**0.20**
EP3	0.11	0.27	**0.16**
ARTS	0.11	0.27	**0.15**
FirstEF	0.13	0.15	**0.14**

To validate our predictions we use the dataset of CAGE-tags from and a set of genes from Ensembl. To compare with the state-of-the-art, we used a maximum allowed distance from the TSS of 50 bp. Table 1 shows the performance of ProSOM versus a number of other PPPs.

We also analyzed the ENCODE regions of the human genome in more detail. The ENCODE project tries to annotate one percent of the human genome in great detail. ProSOM gets an F-measure of 0.28 on this validation set.

4. Discussion and conclusion

Self-organizing maps provide an intuitive way to cluster DNA sequences. They are unique among unsupervised clustering techniques in their ability to distinguish core promoters from other sequences. We packaged this technique as a full-fledged promoter prediction tool, called ProSOM, that performs as well as the best existing software packages.

Acknowledgments

T.A. is funded by a grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). Y.S. is funded by a post-doctoral grant from the Research Foundation Flanders (FWO-Vlaanderen).

References

Abeel, T., Saeys, Y., Bonnet, E., Rouzé, P., & Van de Peer, Y. (2008a). Generic eukaryotic core promoter prediction using structural features of DNA. *Genome Res*, *18*, 310–323.

Abeel, T., Saeys, Y., Rouzé, P., & Van de Peer, Y. (2008b). ProSOM: Core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics*, *(in press)*, –.

Benchmarking machine learning techniques for the extraction of protein-protein interactions from text

Sofie Van Landeghem SOFIE.VANLANDEGHEM@PSB.UGENT.BE
Yvan Saeys YVAN.SAEYS@PSB.UGENT.BE
Yves Van de Peer YVES.VANDEPEER@PSB.UGENT.BE
Department of Plant Systems Biology, VIB, 9052 Gent, Belgium
Department of Molecular Genetics, University of Ghent, 9052 Gent, Belgium

Bernard De Baets BERNARD.DEBAETS@UGENT.BE
Department of Applied Mathematics, Biometrics and Process Control, University of Ghent, 9000 Gent, Belgium

Abstract

Accurately extracting information from text is a challenging discipline because of the complexity of natural language. We have studied state-of-the art systems that extract biological relations from research articles. It has become clear that this field is still struggling with a heterogeneous collection of data sets, data formats and evaluation methods. While recent developments look promising, there is still plenty of room for improvement.

1. Introduction

In the field of life sciences it is vital to automatically link experimental results to data already published in online literature resources. Fully automated systems that extract biological knowledge from text have thus become a necessity. We have studied the feasibility of applying machine learning approaches for the extraction of protein-protein interactions (PPIs). During our comparative study, it became clear that there is a great need for the standardization of evaluation procedures.

2. Corpora

Over the past few years, different methods have been proposed to extract biological relations from text. The development of standard benchmarking data sets is a step forward towards meaningful comparison between these systems. Such corpora include LLL, AImed and BioInfer, which have all been published in different dataformats. Only recently, software has been introduced to convert these and two smaller data sets into a common dataformat (Pyysalo et al., 2008), which facilitates comparison between different methods.

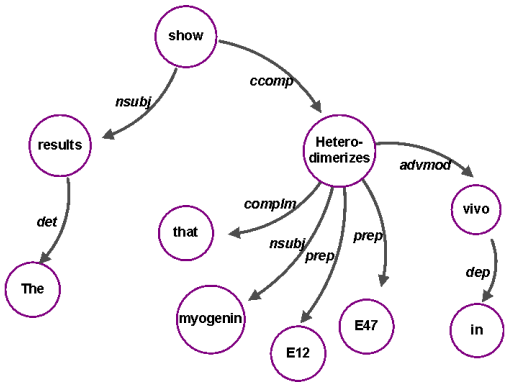


Figure 1. Dependency parse for 'The results show that myogenin heterodimerizes with E12 and E47 in vivo.'

3. PPI extraction

Sentences selected from biomedical text usually contain complex structures with multiple subordinate clauses. Interacting proteins often occur in a sentence with some distance between them. Therefore, pattern-based approaches and algorithms using word order suffer from low recall. On the other hand, techniques solely based on co-occurrence of named entities exhibit low precision. To better capture the semantics of a sentence, recent systems make use of information derived from dependency trees (see Fig 1).

By extracting properties from dependency trees, explicit features can be obtained for each pair of proteins. These feature vectors are used by classifiers such as decision trees, BayesNet and SVM to identify sentences which express a protein-protein interaction. Useful features relate to lexical and syntactic information about the children and ancestors of the proteins in the tree, the presence of common interaction words and depth of the named entities in the tree.

4. Ideas to improve benchmarking

4.1. Common set of benchmark data

A comparative study between different PPI extraction systems is a non trivial task as different studies often benchmark on different data sets. The RelEx system of Fundel et al. (2006) has been reimplemented with the goal of evaluating it on different corpora (Pyysalo et al., 2008). An F score of 0.77 was obtained when benchmarking on LLL, and a score between 0.41 and 0.44 when evaluated on AImed and Bioinfer. We obtain similar results when applying the walk kernel of Kim et al. (2008) to the AImed data set, which results in an F score of 0.44. In contrast, the original paper reports a score of 0.77 for the evaluation on LLL. This shows that for the same extraction method, performance can differ up to 36% depending on the choice of the corpus. It is therefore meaningful to evaluate new algorithms on a collection of different data sets.

4.2. Instance extraction

When benchmarking on the same corpus, different preprocessing steps can yield different instances. Homodimers, which are self-interacting proteins, are sometimes simply discarded. A similar issue is raised by annotations which are nested. The ability of the preprocessing techniques to deal with such annotations influences the final number of instances in the data set and ultimately the performance of the system.

Most corpora do not deal with the construction of negative training data. It has become common practice to adapt the closed world assumption, stating that no interaction exists between two entities when there is no annotated evidence. Even though AImed provides an explicit set of abstracts with no annotated interactions, these are not always used, resulting in different numbers of negative instances in the training set.

Ideally, abstracts for the testing phase should be completely hidden during training. Saetre et al. (2008) pointed out that some evaluations suffer from an artificial boost of performance by using features from the same sentence in both training and testing steps of the machine learning algorithm. This boost of performance has been estimated between 10 and 20%.

4.3. Counting true positives

The definition of true positives varies between different evaluation approaches. Most approaches consider every protein pair as an individual instance and evaluate whether an interaction is stated between these two particular entities. Some however state that an interaction between two proteins may be expressed in the same corpus by more than one instance. To extract a true interaction, retrieving one such instance suffices. The latter evaluation technique exhibits higher recall. Even though this technique may be useful for the evaluation of complete information retrieval systems, we feel the first is more representative for the subtask of extracting interactions between named entities from individual sentences.

4.4. Directed interactions

Finally, the definition of PPI extraction task is not unambiguously defined across corpora. The LLL data set and Bioinfer both consider the role of the different proteins in their interaction and discriminate between effectors and effectees. In AImed however, protein-protein interactions are considered to be symmetrical. This has led to the common practice of treating LLL annotations as symmetrical as well, resulting in artificially higher precision rates.

5. Conclusions

The comparison of different PPI extraction methods is hindered by the lack of standard evaluation procedures. We have pointed out the main problems for such a comparative study and indicated some practical guidelines for setting up a meaningful evaluation.

Acknowledgments

SVL would like to thank the Special Research Fund (BOF) for funding her research. YS would like to thank the Research Foundation Flanders (FWO) for funding his research.

References

Fundel, K., Küffner, R., & Zimmer, R. (2006). Relex–relation extraction using dependency parse trees. *Bioinformatics*, *23*, 365–371.

Kim, S., Yoon, J., & Yang, J. (2008). Kernel approaches for genic interaction extraction. *Bioinformatics*, *24*, 118–126.

Pyysalo, S., Airola, A., Heimonen, J., Bjrne, J., Ginter, F., & Salakoski, T. (2008). Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, *9*.

Saetre, R., Sagae, K., & Tsujii, J. (2008). Syntactic features for protein-protein interaction extraction. *Proceedings of the Second International Symposium on Languages in Biology and Medicine (LBM2007)*.

Predicting sub-Golgi localization of glycosyltransferases

Aalt D J van Dijk AALTJAN.VANDIJK@WUR.NL
Applied Bioinformatics, PRI, Wageningen UR, Droevendaalsesteeg 1, 6708 PB Wageningen, The Netherlands

Dirk Bosch DIRK.BOSCH@WUR.NL
Metabolic Regulation, PRI, Wageningen UR, Droevendaalsesteeg 1, 6708 PB Wageningen, The Netherlands

Cajo J F ter Braak CAJO.TERBRAAK@WUR.NL
Biometris, PRI, Wageningen UR, Bornsesteeg 47, 6708 PD Wageningen , The Netherlands

Sander van der Krol SANDER.VANDERKROL@WUR.NL
Metabolic Regulation, PRI, Wageningen UR, Droevendaalsesteeg 1, 6708 PB Wageningen, The Netherlands

Roeland C H J van Ham ROELAND.VANHAM@WUR.NL
Applied Bioinformatics, PRI, Wageningen UR, Droevendaalsesteeg 1, 6708 PB Wageningen, The Netherlands

Abstract

Proteins fulfill their biological role in specific cellular sub-compartments, and prediction of protein localization is an important topic within bioinformatics. Here we study localization of glycosyltransferases which can reside in any of the cis-, medial-, or trans-Golgi compartments or in the Trans Golgi Network (TGN) compartment. This sub-Golgi localization is important for the order of reactions performed in glycosylation pathways, but it is currently poorly understood. We use a dataset of proteins with experimentally determined sub-Golgi localizations to develop a predictor, making use of a dedicated protein structure-based kernel in an SVM.

1. Experimental dataset

Golgi-localized glycosyltransferases contain one transmembrane helix, which is important for their correct sub-Golgi localization. A dataset of 59 proteins with known sub-Golgi localization was obtained. These were clustered (using the minimum variance method implemented in the R function hclust) based on their sequence identities in order to remove redundant sequences, which would otherwise result in unjustified high performance of the resulting predictor. Based on the sharp rise of maximum inter-cluster similarity when using more clusters, 31 clusters were selected. Of these, 18 had only one entry and 13 had multiple entries with consistent localization, indicating that the available data is consistent. Additional training sequences were obtained based on sequence similarity (via ENSEMBL or BLAST), resulting in 107 sequences with cis-Golgi localization, 117 with medial-Golgi localization, 86 with trans-Golgi localization and 89 with TGN localization.

2. Kernel and SVM

We tested different kernels, both based on the linear sequence (string kernels) and on the modeled 3D-structure of the transmembrane domain (structure kernel). For both kernels, we applied a grouping of amino acids where amino acids were clustered following Shen (2007) into the following 7 groups: AGV, ILFP, YMTS, HNQW, RK, DE, and C. For the string kernel, we took as a starting point the conjoint triad string kernel (Shen, 2007). Triads were redefined to accommodate a fixed spacing of either 0 (the original triad definition) or 1, 2 or 3 (non-sequential triads), since such spacing determines alignment of residues to specific sides of the transmembrane helix. The structure kernel was designed based on observed residue contacts in 3D models of the helix. These models were obtained via structure calculations in CNS (Brunger,1998). [1] Side-chain side-chain contacts were

[1] Dihedral angle and hydrogen bond restraints were defined, and the anneal.inp CNS-script was used to calculate ten structures for each helix. The lowest energy structure was used to obtain the kernel-features.

counted using a distance cutoff of 3.5 Åand each triplet of amino acids within this distance cutoff was counted as one occurrence of a triad.

As SVM implementation SVMlight (Joachims, 1999) was applied. For each type of triad v_i a normalized count was defined as d_i=(f_i-min)/max, where f_i is the raw count and min (max) is the minimum (maximum) over all f_i. Since the number of training examples was relatively small compared to the dimension of the feature space, a linear kernel was expected to be powerful enough. Leave-one-out cross validation was applied to optimize the parameter C, for which a grid [1,2,3,4,5,6,7,8,9,10,15,20,25,30] was used. For the sake of completeness, the radial basis function (RBF) kernel was also tested, where the additional γ parameter was optimized on a grid [500,200,100,50,10,5,1,0.1,0.01,0.001,0.0005,0.0001]. To obtain an unbiased performance estimation, nested cross-validation was used as described previously (Varma, 2006). The leave-one-out cross-validation was performed cluster-wise, meaning that all sequences in one cluster were removed simultaneously.

3. Results

To obtain a multiclass classification, three separate predictors were built: one for cis vs. the other three localizations, one for cis or medial vs. trans or TGN, and one for TGN vs. the other three localizations. This particular ordering was chosen because it coincides with the biologically relevant order cis-medial-trans-TGN. The cis/medial vs. trans/TGN predictor was used to test the performance of the various string kernels. Table 1 shows the prediction accuracies for the various kernels, and indicates that the string-kernels that take structural features of the transmembrane domain into account perform better than kernels that do not take this into account (note that a spacing of 2 or 3 reflects the proximities of residues in 3D-space whereas a spacing of 1 does not). The 3D-structure based kernel has the best prediction performance. Randomly assigning class-labels to each set of clustered sequences and retraining the SVM-predictor resulted in much lower performance (47% accuracy), showing that the performance obtained by the SVM-predictor is non-trivial.

The structure kernel was subsequently used for the other predictors, whose performance was comparable to that of the cis/medial vs. trans/TGN predictor. For each of these three predictors we also tested an RBF instead of linear kernel, which gave comparable results (data not shown). The predictors were combined by using combinatorial logic, e.g. if for a given sequence the cis/medial vs. trans/TGN predictor returns cis/medial and the cis vs. the rest predictor returns not cis then the prediction would be medial. Table 2 shows the confusion table for the resulting predictor. The cross-validated prediction accuracy is 61%.

Table 1. Classification accuracies for cis/medial vs trans/TGN prediction

Kernel	cis/medial	trans/TGN	all
String: spacing 1	64.0	43.0	55.2
String: spacing 0	73.4	58.5	67.2
String: spacing 3	64.2	76.6	69.4
String: spacing 2	64.3	84.3	73.0
Structure based	78.5	72.6	76.1

Table 2. Confusion table for combined predictor

	Predicted			
Experimental	cis	medial	trans	TGN
cis	6	3	1	2
medial	0	5	0	1
trans	0	0	3	0
TGN	1	2	1	5

Application to a variety of glycosyltransferases demonstrates the power of our approach. For example, we obtain consistent predictions when comparing human-mouse orthologs, whereas applying a simple sequence similarity based predictor results in much less consistent predictions. In addition, comparison with a large set of glycan structures, which are the products of the enzymatic actions of the glycosyltransferases, demonstrates a significant correlation between sub-Golgi localization and the predicted ordering of different steps in glycan biosynthesis.

References

Brunger, A.T., et al. (1998) Crystallography & NMR system: A new software suite for macromolecular structure determination, Acta Crystallographica Section D-Biological Crystallography, 54, 905-921.

Joachims, T. (1999) Making large-Scale SVM Learning Practical. In, Advances in Kernel Methods - Support Vector Learning. MIT-Press.

Shen, J.W., et al. (2007) Predicting protein-protein interactions based only on sequences information, Proceedings of the National Academy of Sciences of the United States of America, 104, 4337-4341.

Varma, S. and Simon, R. (2006) Bias in error estimation when using cross-validation for model selection, Bmc Bioinformatics, 7, -.

Prediction of genetic risk of complex diseases by supervised learning

Vincent Botta VINCENT.BOTTA@ULG.AC.BE
Pierre Geurts P.GEURTS@ULG.AC.BE
Louis Wehenkel L.WEHENKEL@ULG.AC.BE
Department of Electrical Engineering and Computer Science
GIGA-Research, University of Liège, B4000 Belgium

Sarah Hansoul S.HANSOUL@ULG.AC.BE
Animal Genomics
GIGA-Research, University of Liège, B4000 Belgium

1. Whole genome association studies

The majority of important medical disorders (f.i. susceptibility to cancer, cardiovascular diseases, diabetes, Crohn's disease) are said to be complex. This means that these diseases are influenced by multiple, often interacting environmental and genetic risk factors. The fact that individuals differ in terms of exposure to environmental as well as genetic factors explains the observed inter-individual variation in disease outcome (i.e. phenotype). The proportion of the phenotypic variance that is due to genetic factors (heritability) typically ranges from less than 10 to over 60 % for the traits of interest. The identification of genes influencing susceptibility to complex traits reveals novel targets for drug development, and allows for the implementation of strategies towards personalized medicine.

Recent advances in marker genotyping technology allow for the genotyping of hundreds of thousands of Single Nucleotide Polymorphisms (SNPs) per individual at less than 0.1 eurocents per genotype, the identification of genomic regions (i.e. loci) that influence susceptibility to a given disease can now be obtained by means of so-called "whole genome association studies" (WGAS).

2. Supervised learning for WGAS

The basic idea behind a GWAS is to genotype a collection of affected (cases) and unaffected (controls) individuals for a very large number of SNPs spread over the entire genome. Genomic regions showing statistical differences among cases and controls are then detected using this dense collection of SNPs. From a machine learning point of view, analysis of this dataset is a binary classification problem, with a very large number of raw symbolic variables, each one corresponding to a different SNP and having only three possible values (homozygous wild, heterozygous and homozygous mutant). On top of this very high p/n ratio, these problems are also generally highly noisy, and the raw input variables are strongly correlated (which is explained by the so-called linkage desiquilibrium).

In this research we study two different representations of the input data for the application of supervised learning, namely the raw genotype data on the one hand, and on the other hand the groups of strongly correlated SNPs (i.e. the haplotype blocks) , representing the observed combinations of about 10 to 100 phased genotypes between the recombination hotspots of the different chromosomes. We report an empirical study based on several simulated datasets where one or two independent or interacting causal mutations on a single chromosome are studied. We provide comparative results of different ensembles of randomized decisions trees adapted to handle the particular nature of these two types of input variables. These methods are assessed in terms of their predictive power as well as their ability to help identifying the genomic regions containing causal mutations.

3. Methods

3.1. Dataset generation

We used the program *gs* (Li & Chen, 2008) to generate samples based on *HapMap* data (Consortium, 2003) so as to keep the linkage desiquilibrium patterns similar to those in actual human populations and focus on chromosome 5. The raw input variables were obtained by taking SNPs spaced by 10 kilobases from the *HapMap* pool to reproduce classical GWAS conditions, and the causal disease loci were removed from the input variables.

Using genotype penetrance tables, 5 different disease models were tested: two for the one locus experiments,

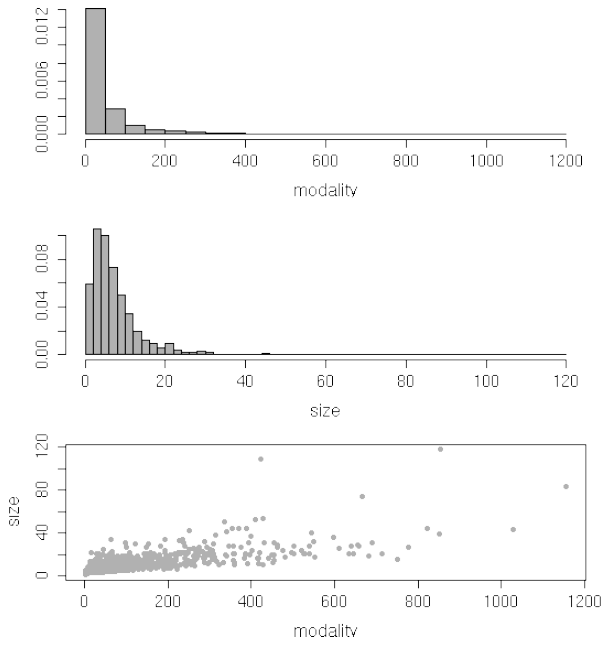


Figure 1. Haplotype block statistics. Top: block modalities; Middle: block length; Bottom: scatter plot

and the three most common disease models with interactions (Li & Reich, 2000) for the two locus case. We considered different noise and penetrance values.

In the first (raw) data representation, the different databases are composed of 14604 symbolic variables with 3 possible values. The second representation is a variant of the first where we group correlated variables into blocks (haplotype blocks chosen according to *HapMap* hotspots). This dramatically reduces the number of variables but it also increases their modalities (up to few hundred possible combinations when the sample comes from a broad population). In total, this yielded 1957 haplotype blocks. Figure 1 shows the histograms of block lengths and number of modalities.

3.2. Supervised learning

We evaluated Random Forests and Extra-Trees (see Geurts et al., 2006 for a precise description of these algorithms and related notions). These methods were customized in an ad hoc way to handle the datasets for the haplotype block variant. Various values of their two main meta-parameters (number of tested attributes and number of trees) were screened while the trees were completely developped.

Learning was repeated 10 times on balanced learning sets (containing between 100 and 1000 controls and as many cases). All models were evaluated on the same independent and balanced test set of size 5000.

The predictive power was assessed using the mean area under the ROC curves and compared to best possible theoretical AUCs which were deduced from the selected disease model.

We ranked SNPs and haplotype blocks using variable importances based on information theory (see Wehenkel, 1998), and provide the mean rank of the SNPs adjacent to the causal mutations, or of the block(s) containing these mutation(s).

4. Preliminary results

Preliminary results show good perspectives. In particular, the different methods obtain rather good AUCs as compared with the theoretical upper bound derived from the disease models. The different methods are also able to predict and to localize the disease loci, rather well. We also observed that most often the direct application of supervised learning to the raw genotype data provides slightly superior results both in terms of risk prediction and loci identification than the application of these methods to haplotype blocks. This essentially suggests that further work should focus on a better determination of the haplotype block structure from the datasets themselves (rather than by extrapolating these structures from other cohorts, as it was the case in these first investigations).

Acknowledgments

This paper presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. Vincent Botta is recipient of a F.R.I.A. fellowship. Sarah Hansoul is a postdoctoral research fellow of the F.R.S.-FNRS and Pierre Geurts is a Research Associates of the F.R.S.-FNRS.

References

Consortium, T. I. H. (2003). The international hapmap project. *Nature*, *426*, 789–796.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees.

Li, J., & Chen, Y. (2008). Generating samples for association studies based on hapmap data. *BMC Bioinformatics*, *9*, 44.

Li, W., & Reich, J. (2000). A complete enumeration and classification of two-locus disease models. *Hum Hered*, *50*, 334–349.

Wehenkel, L. (1998). Automatic learning techniques in power systems.

Component analysis for genome-wide association studies

Gilles Meyer G.MEYER@ULG.AC.BE
Rodolphe Sepulchre R.SEPULCHRE@ULG.AC.BE
Department of Electrical Engineering and Computer Science,
GIGA Bioinformatics Platform,
University of Liège, Belgium.

1. Introduction

This work illustrates the application of component analysis such as principal component analysis (PCA) and independent component analysis (ICA) to analyze SNP databases. The problems of association mapping and population stratification are both addressed with these methods.

2. Component analysis

In the general framework of component analysis, the data matrix $X \in R^{m \times n}$ is approximated by the product of two lower-rank matrices $A \in R^{m \times k}$ and $S \in R^{k \times n}$ with $k \leq m$:

$$X \approx AS \tag{1}$$

where S contains the reduced data and A's columns are the directions spanning the subspace of the reduced data.

If the directions are constrained to be mutually orthogonal and computed to retain as much variance as possible from the original data, the factorization (1) correspond to a principal component analysis of X.

Another possibility is to identify the directions that make the rows of S as statistically independent as possible. This objective is pursued in independent component analysis (Hyvärinen et al., 2001).

3. SNP databases

The human DNA sequence is about 3 billion base pairs of nucleotides (A-T-C-G) arranged into 23 chromosomes. Each individual has one pair of each chromosome, one is inherited from the maternal side and the other one from the paternal side.

In the world's population, there is about 10 million sites or loci that vary between individuals. Such loci referred as single nucleotide polymorphisms (SNPs) are natural candidates for the research of causal differences responsible for diseases or other phenotypes of interest.

At one particular SNP locus, there are usually two alleles (specific nucleotides) observed across the population. Thus, a SNP database can be represented as a matrix whose elements can take 3 discrete values : 2 if the arbitrary reference allele is carried on each chromosome, 1 if the two different alleles are observed and 0 if the non-reference allele is present on each chromosome. To date, an order of magnitude for these databases is the measurement of 10^6 SNPs for 10^3 individuals. These numbers are rapidly increasing with the development of cheaper technologies.

4. Association mapping

The analysis of SNP databases aim at finding loci biologically related to a measured phenotype, for example a particular disease.

The potential of ICA to perform such analysis has been illustrated in (Dawy et al., 2005) on simulated data.

In this work, the method is applied to a real database concerned about the identification of loci involved in Crohn disease.

5. Population stratification

A problem encountered in association mapping is the presence of individuals coming from different populations. This is a source of bias into the observed allele frequencies leading to false discoveries in the mapping process.

In (Price et al., 2006), PCA is used to correct for this stratification effect by computing a component linked to the population structure and then by removing it from the data. The issue will be discussed in the context of a Crohn large database.

Acknowledgments

This work was supported by the Belgian National Fund for Scientific Research (FNRS) through a Research Fellowship at the University of Liège. This paper presents research results of the Belgian Network BioMAGNet (Bioinformatics and Modelling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

References

Dawy, Z., Sarkis, M., Hagenauer, J., & Mueller, J. C. (2005). A novel gene mapping algorithm based on independent component analysis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 381–384). Philadelphia.

Hyvärinen, A., J.Karhunen, & Oja, E. (2001). *Independent component analysis*. Wiley-Interscience.

Price, A., Patterson, N., Plenge, R., Weinblatt, M., Shadick, N., & Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, *38*, 904–909.

Morphological Feature Extraction of Intracranial Pressure Signals via Nonlinear Regression

Fabien Scalzo FSCALZO@MEDNET.UCLA.EDU
Peng Xu PENGXU@MEDNET.UCLA.EDU
Marvin Bergsneider MBERGSNEIDER@MEDNET.UCLA.EDU
Xiao Hu XHU@MEDNET.UCLA.EDU
Division of Neurosurgery, Geffen School of Medicine, University of California, Los Angeles, CA, USA

Abstract

The management of many neurological disorders such as traumatic brain injuries relies on the continuous measurement of intracranial pressure (ICP). Following recent studies, the automatic analysis of ICP pulse seems promising for forecasting intracranial and cerebrovascular pathophysiological changes. MOCAIP algorithm has recently been developed to automatically extract ICP morphological features (in terms of sub-peak positions) in real time. This paper extends MOCAIP by using a regression model instead of Gaussian priors during the peak designation to improve the accuracy of the process. Experimental evaluations conducted on real clinical data indicate that the use of a regression model significantly increases the peak designation accuracy.

1. Introduction

The management of many neurological disorders such as traumatic brain injuries relies on the continuous measurement of intracranial pressure (ICP). Following recent studies (Hu et al., 2008), variations of the ICP signal are linked to the development of intracranial hypertension and cerebral vasospasm, acute changes in the cerebral blood carbon dioxide (CO2) levels, and changes in the craniospinal compliance. Therefore, the automatic and continuous analysis of ICP features appears to be promising for a better monitoring, understanding and forecasting of intracranial and cerebrovascular pathophysiological changes.

Processing ICP signals to extract features in a continuous and reliable way is, however, very challenging and beyond most of state-of-the-art ICP analysis methods.

2. Previous Work

MOCAIP algorithm (Hu et al., 2008) (**Mo**rphological **C**lustering and **A**nalysis of **I**CP **P**ulse) has recently been developed to extract morphological changes of ICP pulse in real time. The algorithm relies on the fact that the ICP waveform is triphasic (*i.e.* three sub-peaks in each ICP pulse). The MOCAIP algorithm offers several interesting properties: it is able to enhance ICP signal quality, to recognize legitimate ICP pulses and to detect the three sub-components in an ICP pulse. This last step is done by considering a set of peak candidates (extracted at curve inflexion points), and by identifying the three peaks among them. During this assignation, MOCAIP makes use of Gaussian priors to set the position of each peak such that the configuration maximizes the probability to observe the peaks given the prior distributions. However, this can be problematic because the position of the peaks within the pulse presents a large variation that is translated into large variance priors and weakens the effectiveness of the peak designation step.

3. Approach

This work introduces an extension of the MOCAIP algorithm to improve the accuracy of the peak designation. The innovative idea is to consider the location of the peaks (p_1, p_2, p_3) as a function $f(x)$ of the pulse signal (discretized as a vector x) (Fig. 1). To this end, a regression model is exploited during the peak designation, instead of the Gaussian priors, to extract the most likely location of each peak,

$$y = f(x) \quad (1)$$
$$\Leftrightarrow (p_1, p_2, p_3) = ax + b \quad (2)$$

Given a set of annotated training pulses, an efficient Spectral Regression (SR) analysis (Cai et al., 2007) is used to estimate the linear function $f(x)$. The Spectral Regression analysis is a recent method which combines

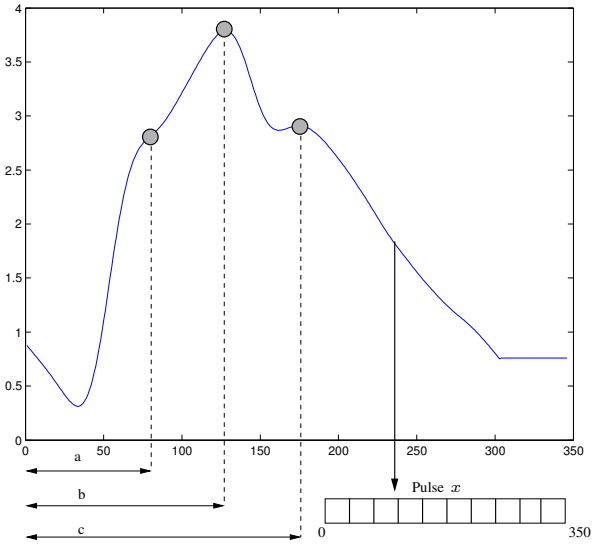


Figure 1. A regression model $f(x)$ is used to predict the positions a, b and c, of the three peaks. The pulse is discretized and normalized into a vector x.

spectral graph analysis and ordinary regression. The main idea of Spectral Regression is to use eigenvectors of the affinity (*i.e.* item-item similarity) matrix to reveal a low-dimensional structure of high-dimensional data. In our framework, we use a RBF kernel to project the data into a higher dimensional space and thus capture the nonlinear relation between the pulse x and the position of the peaks $y = (p_1, p_2, p_3)$.

Once the peak positions have been predicted by the model, a nearest-neighbor matching algorithm is used to assign the candidates to the label of the closest prediction.

4. Experiments

The effectiveness of the proposed extension is evaluated by measuring the accuracy of the algorithm to designate the three ICP peaks on real clinical data. To do so, we assume that the ICP pulses have been previously extracted using MOCAIP and that a set of candidate peaks has been detected.

The dataset used during our experiments contains 13611 ICP pulses that were extracted from the ICP signals of 66 patients. It is a particularly challenging dataset because among the pulses, 1717 have missing P_1, 265 have missing P_2 and 34 have missing P_3. The average accuracy (in terms of True Positive (TP) and False Positive (FP) rates) is recorded using a five-fold cross-validation procedure. The results obtained by the proposed method are reported in Table 1 and compared to MOCAIP. Our extension achieves a very high true positive rate for correctly designating the first two peaks. The significant improvement in terms of True Positive and False Positive rate is confirmed by the combined accuracy (TP+FN)/(TP+FP+TN+FN); MOCAIP obtains 90%, 88%, and 87% for each peak and the results of the proposed extension are 97%, 98% and 88%.

Figure 2 illustrates detection results on four different pulses. We can observe that the detection is successful despite the large variability in shape of the Intracranial Pressure Signals (ICP).

	P_1(TP, FP)	P_2 (TP, FP)	P_3 (TP, FP)
MOCAIP	91%, 18%	88%, 39%	86%, 53%
this work	**99%, 13%**	**98%, 11%**	**88%, 56%**

Table 1. Peak Identification results in terms of True Positive (TP) and False Positive (FP) rates are reported for the MOCAIP algorithm and the proposed Spectral Regression (SR) extension.

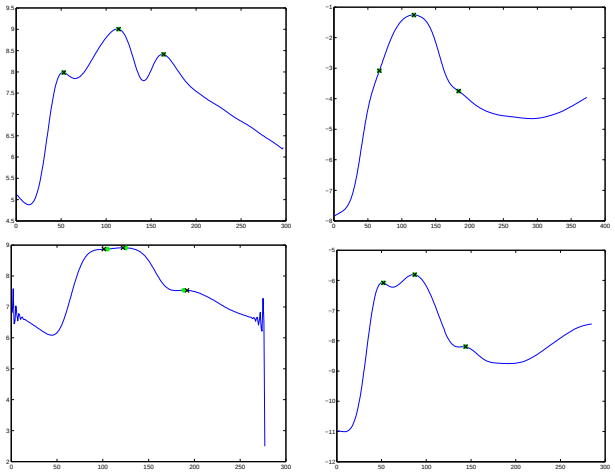


Figure 2. Peak detection on four ICP pulses. The ground truth is marked as a green dot and the output of our framework is depicted as a cross.

Acknowledgments

The present work is supported in part by NINDS grants R21-NS055998, R21-NS055045, R21-NS059797 and R01-NS054881.

References

Cai, D., He, X., & Han, J. (2007). Spectral Regression for Efficient Regularized Subspace Learning. *IEEE International Conference on Computer Vision (ICCV'07).*

Hu, X., Xu, P., Scalzo, F., Miller, C., Vespa, P., & Bergsneider, M. (2008). Morphological Clustering and Analysis of Continuous Intracranial Pressure. *Submitted to IEEE Transactions on Biomedical Engineering.*

An Extended NMF Algorithm for Word Sense Discrimination

Tim Van de Cruys T.VAN.DE.CRUYS@RUG.NL
Humanities Computing, University of Groningen, Oude Kijk in 't Jatstraat 26, 9712 EK Groningen

Abstract

Many words used in natural language are ambiguous: they have various senses. Traditional algorithms dealing with semantic similarity cannot cope with this ambiguity. We present an extension of a dimensionality reduction algorithm called NON-NEGATIVE MATRIX FACTORIZATION that combines both 'bag of words' data and syntactic data, in order to find semantic dimensions according to which both words and syntactic relations can be classified. The use of three way data allows one to determine which dimension(s) are responsible for a certain sense of a word, and adapt the corresponding feature vector accordingly, 'subtracting' one sense to discover another one. The intuition in this is that the syntactic features of the syntax-based approach can be disambiguated by the topical dimensions found by the bag of words approach.

1. Introduction

Most work on semantic similarity relies on the distributional hypothesis (Harris, 1985). This hypothesis states that words that occur in similar contexts tend to be similar. With regard to the context used, two basic approaches exist. One approach makes use of 'bag of words' co-occurrence data; in this approach, a certain window around a word is used for gathering co-occurrence information. Bag of words methods are particularly good at finding topical similarity. One of the dominant methods using this method is LATENT SEMANTIC ANALYSIS (LSA, (Landauer et al., 1998)).

The second approach uses a more fine grained distributional model, focusing on the syntactic relations that words appear with. Typically, a large text corpus is parsed, and dependency triples are extracted.[1] Syntax-based methods are good at finding a tighter, synonym-like similarity. Note that the former approach does not need any kind of linguistic annotation, whereas for the latter, some form of syntactic annotation is needed.

In this research, a framework is explored that tries to combine best of both approaches. The intuition in this is that the syntactic features of the syntax-based approach can be disambiguated by the topical dimensions found by the window-based approach.

2. Methodology

2.1. Extending Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) (Lee & Seung, 2000) is a group of algorithms in which a non-negative matrix $V_{n\times m}$ is factorized into two other matrices, $W_{n\times r}$ and $H_{r\times m}$, subject to the constraint that $W, H \geq 0$.

Typically, r is chosen much smaller than n, m so that both instances and features are expressed in terms of a few components. Practically, the factorization is carried out through the iterative application of update rules.

Since we are interested in the classification of nouns according to both 'bag-of-words' context and syntactic context, we first construct three matrices that capture the co-occurrence frequency information for each mode. The first matrix contains co-occurrence frequencies of nouns cross-classified by dependency relations, the second matrix contains co-occurrence frequencies of nouns cross-classified by words that appear in the noun's context window, and the third matrix contains co-occurrence frequencies of dependency relations cross-classified by co-occurring context words.[2]

We then apply NMF to the three matrices, but we inter-

[1] e.g. dependency relations that qualify *apple* might be 'object of *eat*' and 'adjective *red*'. This gives us dependency triples like $< apple, obj, eat >$.

[2] All co-occurrence information is extracted from the Twente Nieuws Corpus (Ordelman, 2002). The corpus has been parsed with the Dutch dependency parser Alpino (van Noord, 2006).

leave the separate factorizations: the results of the former factorization are used to initialize the factorization of the next matrix. This implies that we need to initialize only three matrices at random; the other three are initialized by calculations of the previous step. The process is represented graphically in figure 1.

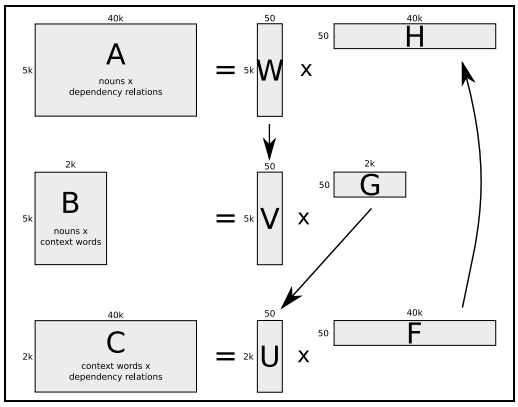


Figure 1. A graphical representation of the extended NMF

When the factorization is finished, the three modes (nouns, dependency relations and context words) are classified according to latent semantic dimensions.

2.2. Sense Subtraction

Next, we want to use the factorization that has been created in the former step for word sense discrimination. The intuition is that we 'switch off' one dimension of an ambiguous word, to reveal possible other senses of the word. From matrix H, we know the importance of each syntactic relation given a dimension. With this knowledge, we can 'subtract' the syntactic relations that are responsible for a certain dimension from the original noun vector.

The last step is to determine which dimension(s) are responsible for a certain sense of the word. In order to do so, we embed our method in a clustering approach. First, a specific word is assigned to its predominant sense (i.e. the most similar cluster). Next, the dominant semantic dimension(s) for this cluster are subtracted from the word vector, and the resulting vector is fed to the clustering algorithm again, to see if other word senses emerge.

3. Results

3.1. Example

Example (1) shows the top-10 similar words for the ambiguous proper name *Barcelona*, which may either refer to the Spanish city or to the Spanish football club. In (a), the results for the original vector are given; the two senses of *Barcelona* are clearly mixed up, showing cities as well as football clubs among the most similar nouns. In (b), where the 'football' dimension has been subtracted, only cities show up. In (c), where the 'cities' dimension has been subtracted, only football clubs remain.

(1) a. *Barcelona*, *Arsenal*, *Inter*, *Juventus*, *Vitesse*, *Milaan* 'Milan', *Madrid*, *Parijs* 'Paris', *Wenen* 'Vienna', *München* 'Munich'

b. *Barcelona*, *Milaan* 'Milan', *München* 'Munich', *Wenen* 'Vienna', *Madrid*, *Parijs* 'Paris', *Bonn*, *Praag* 'Prague', *Berlijn* 'Berlin', *Londen* 'London'

c. *Barcelona*, *Arsenal*, *Inter*, *Juventus*, *Vitesse*, *Parma*, *Anderlecht*, *PSV*, *Feyenoord*, *Ajax*

3.2. Evaluation

Our method has been embedded in an automatic clustering framework and evaluated against Dutch EurowordNet (Vossen et al., 1999). Compared to Pantel and Lin (2002) – considered state of the art in word sense discrimination – our method consistently scores higher with regard to precision, but lower with regard to recall (e.g. with wordnet similarity threshold $\theta = 0.50$, $p = 69\%$ and $r = 56\%$ for our method – $p = 38\%$ and $r = 60\%$ for Pantel and Lin's).

References

Harris, Z. (1985). Distributional structure. In J. J. Katz (Ed.), *The philosophy of linguistics*, 26–47. Oxford University Press.

Landauer, T., Foltz, P., & Laham, D. (1998). An Introduction to Latent Semantic Analysis. *Discourse Processes*, *25*, 295–284.

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *NIPS* (pp. 556–562).

Ordelman, R. (2002). Twente Nieuws Corpus (TwNC). Parlevink Language Techonology Group. University of Twente.

Pantel, P., & Lin, D. (2002). Discovering word senses from text. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 613–619). New York, NY, USA: ACM Press.

van Noord, G. (2006). At Last Parsing Is Now Operational. *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles* (pp. 20–42). Leuven.

Vossen, P., et al. (1999). Eurowordnet, building a multilingual database with wordnets for several european languages.

Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach

Koen Smets KOEN.SMETS@UA.AC.BE
Bart Goethals BART.GOETHALS@UA.AC.BE
Brigitte Verdonk BRIGITTE.VERDONK@UA.AC.BE
Department of Mathematics and Computer Science, University of Antwerp, Antwerp, Belgium

Abstract

Since the end of 2006 several autonomous bots are, or have been, running on Wikipedia to keep the encyclopedia free from vandalism and other damaging edits. These expert systems, however, are far from optimal and should be improved to relieve the human editors from the burden of manually reverting such edits. We investigate the possibility of using machine learning techniques to build an autonomous system capable to distinguish vandalism from legitimate edits. We highlight the results of a small but important step in this direction by applying commonly known machine learning algorithms using a straightforward feature representation. This study demonstrates that elementary features, which are also used by the current approaches to fight vandalism, are not sufficient to build such a system. They will need to be accompanied by additional information which, among other things, incorporates the semantics of a revision.

1. Experiments

We will discuss the setting for our machine learning experiment conducted on simplewiki, the Simple English version of Wikipedia. We first consider the labeling of the data and its representation. Thereafter we discuss the results of two learning algorithms put to test: a Naive Bayes classifier on bags of words (BOW) (McCallum 1996) and a combined classifier built using probabilistic sequence modeling (Bratko et al. 2006), also referred to in the literature as statistical compression models.

1.1. Labeling of Revisions

As a proof of concept and because of space and time constraints, we run the preliminary machine learning experiments on Simple English Wikipedia, a user-contributed online encyclopedia intended for people whose first language is not English.

The data is labeled by inspecting comments that signal a revert action, i.e. an action which restores a page to a previous version. This approach closely resembles the identification of the set of revisions denoted by Priedhorsky et al. (2007) as Damaged-Loose, a superset of the revisions explicitly marked as vandalism (Damaged-Strict).

While labeling based on commented revert actions is a good first order approximation, mislabeling cannot be excluded. If we regard vandalism as the positive class throughout this abstract, then there will be both false positives and false negatives. The former arises when reverts are misused for other purposes than fighting vandalism like undoing changes without proper references or prior discussion. The latter occurs when vandalism is corrected but not marked as reverted in the comment, or when vandalism remains undetected for a long time. Estimating the number of mislabelings is very hard and manual labeling is out of question, considering the vast amount of data.

1.2. Revision Representation

In this case study we use the simplest possible data representation. As for ClueBot (Carter 2007) and VoABot II, the two active vandal fighting bots on Wikipedia nowadays, we extract raw data from the current revision and from the history of previous edits. In particular, for each revision we use its text, the text of the previous revision, the user groups (anonymous, bureaucrat, administrator ...) and the revision comment. We also experimented with including the lengths of the revisions as extra features. The effect

on overall performance is however minimal and thus we discard them in this analysis. Hence the focus lies here more on the content of an edit.

As the modified revision and the one preceding it differ slightly, it makes sense to summarize an edit. Like ClueBot, we calculate the difference using the standard *diff* tool. Processing the output gives us three types of text: lines that were inserted, deleted or changed. As the changed lines only differ in some words or characters from each other, we again compare these using *wdiff*. Basically, this is the same as what users see when they compare revisions visually.

1.3. Analysis and Discussion

Table 1 indicates the performance of a simplified version of ClueBot. It is lower than the performance of the original ClueBot which relies on a user whitelist for trusted users and only reverts edits done by anonymous or new users (Carter 2007). Table 2 shows the results of the machine learning experiments on a 40% test set.

1.3.1. BOW + Naive Bayes

The precision of the Naive Bayes classifier only taking into account the revision diff features as bags of words, both with or without user group information and revision comments, is almost the same as in Table 1. A significant increase can be noticed in terms of recall and F_1, especially when including user group information and comment.

1.3.2. Probabilistic Sequence Modeling

Interesting to note is that the recall is much higher, but that the precision drops unexpectedly. We lack a plausible explanation for this strange behaviour, but the effect can be diminished by setting the threshold parameter to a score higher than zero. This is shown in Figure 1, where we plot the precision/recall curves for varying thresholds for the probabilistic sequence models and for the Naive Bayes models, both with and without user groups and comments. The marks show the results when the log ratio threshold is equal to 0. The tendency is that, despite the worse behavior shown in Table 2, the overall accuracy measured in term of precision and recall is better for the compression based models than for the bag of words model using Naive Bayes.

Acknowledgements

Koen Smets is supported by a Ph. D. fellowship of the Research Foundation - Flanders (FWO).

Table 1. Performance of ClueBot (without user whitelist) on Simple English Wikipedia.

	ACC	PRE	REC	F_1
ClueBot	0.9270	0.6114	0.1472	0.2372

Table 2. Results for Naive Bayes and Probabilistic Sequence Modeling.

(a) revision diff

	ACC	PRE	REC	F_1
NB	0.9303	0.6166	0.2503	0.3561
PSM	0.8554	0.3117	0.7201	0.4351

(b) revision diff + commment + user groups

	ACC	PRE	REC	F_1
NB	0.9314	0.5882	0.3694	0.4359
PSM	0.8436	0.3209	0.9171	0.4755

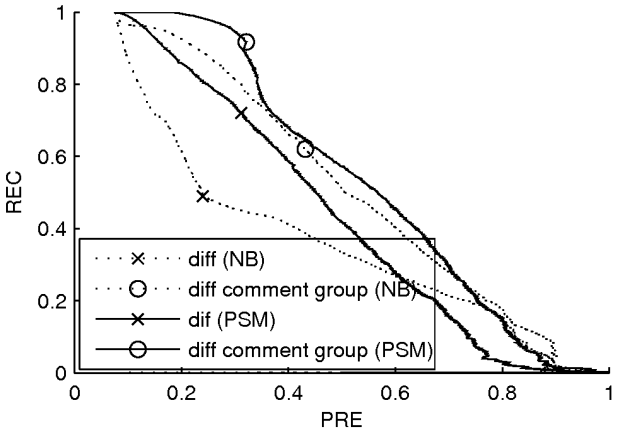


Figure 1. Precision/Recall curves.

References

Bratko, A., Cormack, G. V., Filipič, B., Lynam, T. R., & Zupan, B. (2006). Spam Filtering using Statistical Data Compression Models. *JMLR*, *6*, 2673–2698.

Carter, J. (2007). ClueBot and Vandalism on Wikipedia. Unpublished. Available at `http://24.40.131.153/ClueBot.pdf`.

McCallum, A. K. (1996). Bow: a Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. Available at `http://www.cs.cmu.edu/~mccallum/bow`.

Priedhorsky, R., Chen, J., Lam, S. T. K., Panciera, K., Terveen, L., & Riedl, J. (2007). Creating, Destroying, and Restoring Value in Wikipedia. *Proceedings of the ACM SIGGROUP conference.*

Supervised learning of short-term strategies for generation planning

Bertrand Cornélusse BERTRAND.CORNELUSSE@ULG.AC.BE
Louis Wehenkel L.WEHENKEL@ULG.AC.BE
Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium

Gérald Vignal GERALD.VIGNAL@EDF.FR
OSIRIS department, EDF R&D, 1 avenue du général de Gaulle, F-92141 Clamart Cedex, France

1. Short term electricity generation planning

Short term electricity generation aims at deciding which generation units will be in operation for a certain time period and how much they will produce. Typical generation pools contain a mix of generation units: classical thermal plants, nuclear plants, hydro-electric generators, wind turbines, ... Wind turbines can be considered as non controllable in a time horizon of one day and can be seen as negative loads. On the other hand, the operation of thermal plants and hydro-electric generators has to be planned. Although this problem can be formulated as a multi-stage stochastic programming problem, its complexity is by far too large to allow for an exact solution by available methods. In current practice, the generation plans are generally optimized deterministically, based on a demand forecast and generation units availability assumptions. The generation plans are typically computed the day before they are executed and adjusted in-real-time by good practice rules so as to cope with the differences between real-time conditions and forecasts.

We propose a simulation based approach which uses deterministic optimization methods to compute optimal plannings for a *set* of possible scenarios and extracts by machine learning recourse strategies from these simulations for the next day.

2. Problem description

In the context of short term generation planning, the generation pattern must satisfy some *coupling constraints* (CC) linking all the generation units. First, as the electricity cannot be stored in sufficient quantities, generation must always be close to demand. Secondly, for some more technical reasons, some levels of ancillary services are required. The generation pool is typically divided in 2 categories: thermal units and hydro-electric generation valleys, themselves containing reservoirs, turbines and pumps. The operation of the thermal generation units is restricted by some *dynamical constraints* (DC), because the thermal units must stay in operation for a minimum duration, and because the levels of hydro-reservoirs must stay between acceptable limits. One has thus to decide when to start and when to stop thermal units and to fix their set point if they operate, and for the hydro-valleys one has to decide when to use water to generate electricity, when to store some water by pumping and when to spill water out of dams.

The objective of the generating company is to minimize generation costs including fuel costs, thermal units start-up costs, and opportunity costs for the utilization of water in hydropower plants.

2.1. Mathematical model

For appropriate choices of cost functions and for an appropriate formulation of dynamical constraints, this problem can be modeled as a Mixed Integer Linear Program (MILP):

$$\min_{p_i, s_i, p', s'} \quad \sum_{i \in I} C_i(p_i) + \sum_{t=1}^{T} C_P(p'_t) + \sum_{t=1}^{T} C_S(s'_t) \quad (1)$$

$$\text{s.t.} \quad p'_t = DP_t - \sum_{i \in I} p_{i,t}, \ \forall t \in \{1, ..., T\} \quad (2)$$

$$s'_t = DS_t - \sum_{i \in I} s_{i,t}, \ \forall t \in \{1, ..., T\} \quad (3)$$

$$p_i \in \mathcal{D}_i, s_i \in \mathcal{S}_i, \ \forall i \in I. \quad (4)$$

The optimization is performed over T time periods. The letter i indexes the set of generating units I. p_i (respectively s_i) is the production (ancillary services level) of unit i all along the planning period ($p_i = (p_{i,1}, ..., p_{i,T})$). The constraints (4) indicate that p_i and s_i must stay inside sets encoding the DC. $C_i(\cdot)$ accounts for all the costs linked to generation units. To model the CC, we define some slack variables p'

(2) and s' (3). p' is the mismatch between generation and demand. s' represents the mismatch between the required level of ancillary services and the actual level that is provided. These slack variables penalize the objective though the functions $C_p(\cdot)$ and $C_s(\cdot)$.

To solve this problem efficiently, we use a state of the art *branch and cut* algorithm (ILOG, 2007).

3. Learning of recourses strategies

As mentioned in Section 1, a unique planning is submitted one day before its execution, but it is admitted to take recourses at predefined time steps during the day, say every two hours. Because complete recomputation of the plannings is not achievable during the day, we propose to use the time that is available off-line to make some simulations and to infer some recourse strategies applicable in real-time.

Consider a set D of demand patterns, and a set O of generation units which could become unavailable next day. Let $\pi^\star$ be the optimal planning associated to a reference demand scenario $D^\star \in D$. Suppose that we want to compute an optimal recourse strategy $\sigma^*_{t_r}(\xi_{t_r})$ for a single a priori fixed recourse time $t_r \in \{1, 2, \ldots, T\}$, i.e. we want to know the modifications to bring to all the units from time t_r+1 to T once the real behavior ξ_{t_r} of the system between time 1 and t_r is known. Let S be the set of scenarios made of one demand of D and of a unit outage of O imposed at a time in $\{1, ..., t_r\}$. First, we compute the plannings π_s for each scenario $s \in S$ by imposing the planning $\pi^\star$ for times 1 to t_r and using the optimization problem formulation of Section 2.1 to adjust the planning for time $t_r + 1$ to T. The difference $\pi_s - \pi^\star$ illustrates the impact of the demand variation and the unit outage of scenario s on the reference planning. We exploit these simulations to formulate a supervised learning problem in order to derive a function $\hat{\sigma}^*_{t_r}(\xi_{t_r})$ that will serve as a recourse strategy. The learning set is illustrated in Table 1.

Inputs	**output**
• state of the system at t_r, • observed demand derivation from forecasting until t_r, • unit failure before t_r, • prediction time $t > t_r$,	• ON/OFF status of unit i at t, • and/or power of unit i at t.

Table 1. An item of the Learning set.

To simplify the learning phase, we propose to learn separately the adjustements for the different units and time steps $t \in \{t_r + 1, \ldots, T\}$. This approach has the advantage to lead to a (relatively) simple supervised learning formulation, but it points out two issues:

1. Since we apply a time decomposition, the units DC may not be satisfied; an additional phase will be needed to impose them.
2. Since the strategies are learned unit by unit, the global view of the system is weakened and global constraints are not satisfied; they will also have to be enforced a posteriori.

Although we focus here on the construction of a recourse strategy for a single priod t_r, we can derive strategies for multiple periods during the day by applying this procedure iteratively for an increasing sequence of $t_r \in \{1, 2, ..., T\}$.

4. Validation of the learned strategies

We want to assess the optimality of the learned strategies and to compare them to the ones that are currently used. The latter option is difficult to achieve while this research is carried out through simulations on a reduced generation pool. On the other hand, we can evaluate our strategies on scenarios obtained by Monte-Carlo simulations and compare their cost to

- the reference planning perfectly adjusted to the uncertainties by re-optimization from time $t_r + 1$ to T (lower bound),
- the reference planning not adjusted (upper bound).

Acknowledgments

Bertrand Cornélusse is funded by the FRIA (Belgian Fund for Research in Industry and Agriculture). This paper presents research results of the Belgian Network DYSCO, funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

References

Carpentier, P., Cohen, G., Culioli, J., & Renaud, A. (1996). Stochastic optimization of unit commitment: a new decomposition framework. *Power Systems, IEEE Transactions on*, *11*, 1067–1073.

ILOG (2007). *ILOG CPLEX 11.0 user's manual*.

Performance Evaluation of Machine Learning Techniques for the Localization of Users in Wireless Sensor Networks

Jean-Michel Dricot JDRICOT@ULB.AC.BE
Mathieu Van der Haegen MAVDHAEG@ULB.AC.BE
Yann-Ael Le Borgne YLEBORGN@ULB.AC.BE
Gianluca Bontempi GBONTE@ULB.AC.BE
ULB Machine Learning Group – Université Libre de Bruxelles – 1050 Bruxelles, Belgium

Abstract

In this paper, we introduce a novel and modular framework that is used to evaluate the performance of several user localization techniques in a wireless sensors environment. Three different stages are considered: (i) the signal acquisition and the corresponding distance model, (ii) the terminal positioning, and (iii) the filtering of the estimates over time. Moreover, we investigated how an accelerometer could be included in the filters model in order to further refine the accuracy.

Different implementations of the above-stated modules have been implemented and combined and the corresponding performance is investigated.

1. Motivation

Our research is specific in that it does not aim at implementing a single localization technique but rather focuses on the development of a *testbed* suitable for evaluating the performance of different positioning approaches. The hardware we used includes a mesh of pre-deployed sensors (TMote Invent with CC2420 Chipcon radios and accelerometer) whose position is known. A lightweight computer worn by the user embeds the localization software and collects the real-world data.

The localization process is achieved trough the consecutive steps, as presented on Figure 1. For each step, we provide multiple implementations. First, since the radio signal in an indoor environment presents large variations over time and different beaconing protocols are evaluated to average and de-noise the estimate of the signal strength. Second, we consider several variants of the multilateration technique (Savvides et al., 2003). These are: (i) the simple, (ii) the subsampled, (iii) the nearest-neighbour, and (iv) the weighted nearest-neighbour multilateration. Third, we investigate how the data issued by the accelerometer can be used to detect the user mobility *vs.* a natural body movement (which is considered as noise). The corresponding instantaneous position estimate and the data of the accelerometer are merged into a recursive filtering module. Three different Kalman filters (Brown & Hwang, 1996) are provided, each of them having a different underlying model.

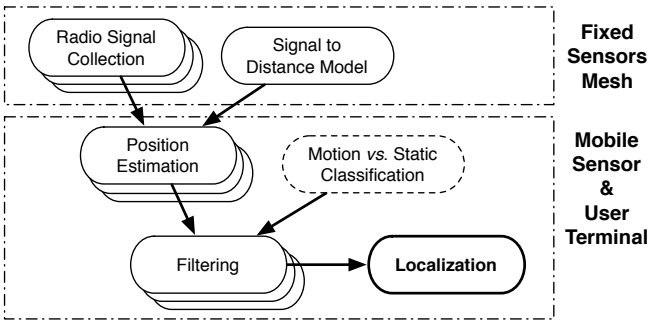


Figure 1. The modular architecture of the localization framework. The three modules on the left of the schema have multiple implementations.

2. Results

The experiments were conducted in the basement of an indoor building made of large corridors and metal structures. In a first round, the data issued from the accelerometer were processed. It has been noted that the variance of the acceleration is a good estimator to detect the user's behaviour, i.e., whether it is static or in motion. On Figure 2, the variance of the acceleration in the x-axis and the y-axis is reported and the corresponding behaviour is annotated. A classification technique knows as Support Vector Machine (Shawe-Taylor & Cristianini, 2000) was used with an overall accuracy of up to 90%.

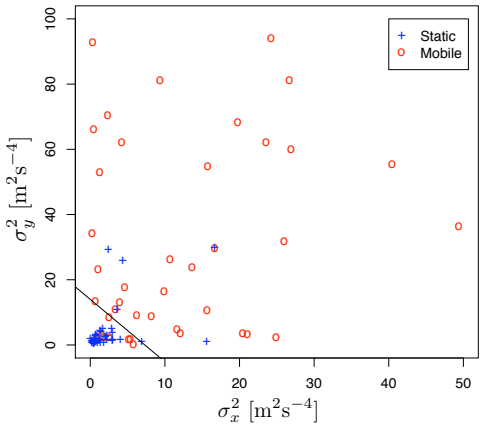


Figure 2. Classification of the terminal mobility. It is characterized by the variance of the acceleration along the two axes of the mobile sensor.

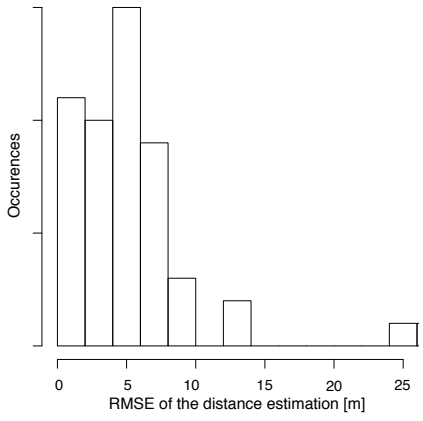


Figure 3. Distribution of the RMSE of the position as evaluated by the the multilateration algorithm.

The Figure 5 presents the underlying model for various filter we implemented. We will now focus on the joint use of the accelerometer to further improve a Kalman filter.

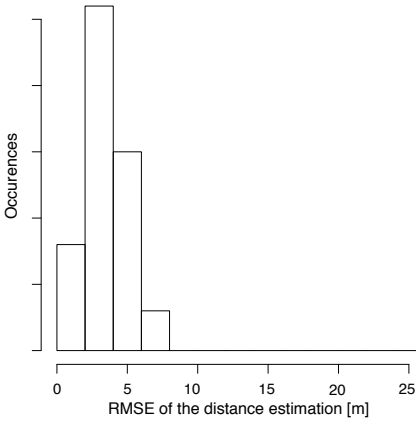


Figure 4. Distribution of the RMSE of the position after filtering and fusion with the information of the accelerometer.

The Figure 3 reports the Root Mean Square Error (RMSE) of the position estimated by the multilateration technique without additional filtering. One can observe that its average positioning error is $\mu = 5.4\,\text{m}$ and that this technique presents a significant variance ($\sigma = 4.9\,\text{m}$). In a second experiment, shown on Figure 4, the samples were collected at the accelerometer and used to further refine the filter model. In that case, the average RMSE of the estimation falls to 3.4 m while, at the same time, the variance of the error is divided by 3, i.e., $\sigma = 1.6\,\text{m}$. Our results suggest that the fusion of multiple sensors has a significant, positive impact on a non-supervised user localization and tracking.

References

Brown, R., & Hwang, P. (1996). *Introduction to random signals and applied kalman filtering.* Wiley Press.

Savvides, A., Park, H., & Srivastava, M. (2003). The n-hop multilateration primitive for node localization problems. *Journal of Mobile Networking Applications.*

Shawe-Taylor, J., & Cristianini, N. (2000). *Support vector machines.* Cambridge University Press.

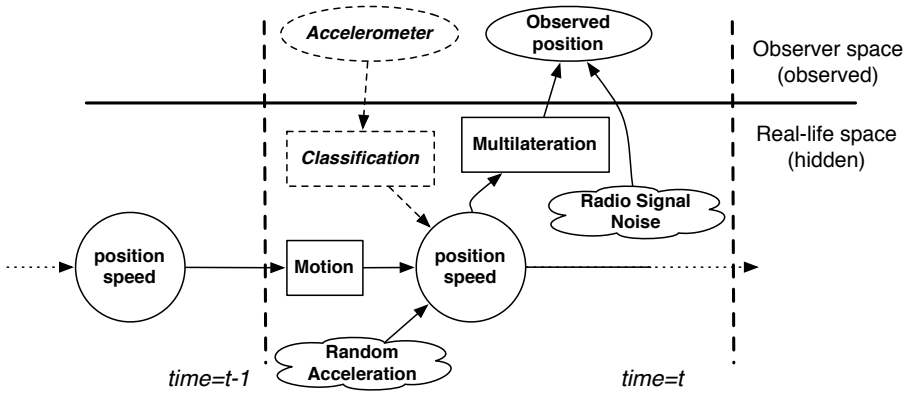


Figure 5. Underlying model for the design of the customizable Kalman flter. The variables are noted as ovals, the transition functions as squares, and the noise processes as a cloud. Note that the samples collected by the accelerometer are only used in selected implementations of the filter. It is therefore noted as dashed.

Bayes-Relational Learning of Opponent Models from Incomplete Information in No-Limit Poker

Marc Ponsen
MICC Universiteit Maastricht, Netherlands

Jan Ramon
Kurt Driessens
Declarative Languages and Artificial Intelligence Group, Katholieke Universiteit Leuven (KUL), Belgium

Tom Croonenborghs
Biosciences and Technology Department, KH Kempen University College, Belgium

Karl Tuyls
Technische Universiteit Eindhoven, Netherlands

Abstract

We propose an opponent modeling approach for No-Limit Texas Hold'em poker that starts from a (learned) prior, i.e., general expectations about opponent behavior and learns a relational regression tree-function that adapts these priors to specific opponents. An important asset is that this approach can learn from incomplete information (i.e. without knowing all players' hands in training games).

1. Introduction

For many board and card games, computers have at least matched humans in playing skill. An exception is the game of poker, offering new research challenges. The complexity of the game is threefold, namely poker is (1) an imperfect information game, with (2) stochastic outcomes in (3) an adversarial multi-agent environment. One promising approach used for AI poker players applies an adaptive imperfect information game-tree search algorithm to decide which actions to take based on expected value (EV) estimates (Billings et al., 2006). This technique (and related simulation algorithms) require two estimations of opponent information to accurately compute the EV, namely a prediction of the opponent's outcome of the game and prediction of opponent actions. Therefore learning an opponent model is imperative and this model should include the possibility of using relational features for the game-state and -history.

In this paper we consider a relational Bayesian approach that uses a general prior (for outcomes and actions) and learns a relational regression tree to adapt that prior to individual players. Using a prior will both allow us to make reasonable predictions from the start and adapt to individual opponents more quickly as long as the choice of prior is reasonable.

2. Learning an Opponent Model

We learn an opponent model for players in the game of No-Limit Texas Hold'em poker. To make the model useful for an AI player, we must be able to learn this model from a limited amount of experience and (if possible) adapt the model quickly to changes in the opponent's strategy. An added, and important, difficulty in poker is that we must be able to learn this model given a large amount of hidden information. We propose to start the opponent model with a prior distribution over possible action choices and outcomes. We will allow the model to adapt to different opponents by correcting that prior according to observed experience.

Consider a player p performing the i-th action a_i in a game. The player will take into account his hand cards, the board B_i at time point i and the game history H_i at time point i. The board B_i specifies both the identity of each card on the table (i.e., the *community cards* that apply to all players) and when they appeared, and H_i is the betting history of all players in the game. The player can *fold*, *call* or *bet*. For simplicity, we consider *check* and *call* to be in the same class, as well as *bet* and *raise* and we don not consider

the difference between small and large calls or bets at this point.[1]

We limit the possible outcomes of a game r_p for a player p to: 1) p folds before the end of the game ($r_p = lose$), 2) p wins without showing his cards ($r_p = win$) and 3) p shows his cards ($r_p = cards(X, Y)$). This set of outcome values also allows us to learn from examples where we did not see the opponent's cards, registering these cases as *win* or *lose*, without requiring the identities of the cards held by the player. The learning task now is to predict the outcome for an opponent $P(r_p|B_i, H_i)$ and the opponent action (given a guess about his hand cards) $P(a_i|B_i, H_{i-1}, r_p)$

2.1. Learning the Corrective Function

We propose a two-step learning approach. First, we learn functions predicting outcomes and actions for poker players in general. These functions are then used as a prior, and we learn a corrective function to model the behavior and statistics of a particular player. The key motivations for this are first that learning the difference between two distributions is an elegant way to learn a multi-class classifier (e.g. predicting distributions over 2+(52*53/2) possible outcomes) by generalizing over many one-against-all learning tasks, and second that even with only a few training examples from a particular player already accurate predictions are possible.

In the following description, the term example references a tuple $(i, p, a_i, r_p, H_{i-1}, B_i)$ of the action a_i performed at step i by a player p, together with the outcome r_p of the game, the board B_i and the betting history H_{i-1}.

Consider the mixture D_{p+*} of two distributions: the distribution D_* of arbitrarily drawn examples from all players and the distribution D_p of arbitrarily drawn examples from a particular player p. Then, consider the learning problem of, given a randomly drawn example x from D_{p+*}, predicting whether x originated from D_* or from D_p. For a given learning setting (either predicting actions from outcomes or predicting outcomes from actions), it is easy to generate examples from D_* and D_p, labeling them with $*$ or p, and learning the function $P(D_p|x)$, giving for each example x the probability the example is labeled with p. We do so by using the relational probability tree learner Tilde (Blockeel & De Raedt, 1998). From this learned 'differentiating' model, we can compute the probability $P(x|D_p)$, for every example x by using Bayes' rule:

$$P(x|D_r) = P(D_r|x) \cdot P(x)/P(D_r) \quad (1)$$

Since we have chosen to generate as many examples for D_* as for D_p in the mixture,

$$P(D_p) = P(D_*) = 1/2 \quad (2)$$
$$P(x) = P(D_*)P(x|D_*) + P(D_p)P(x|D_p) \quad (3)$$

and substituting (2) and (3) into (1) gives:

$$\begin{aligned} P(x|D_p) &= \left(P(D_p|x) \cdot \left(\frac{1}{2}P(x|D_p) + \frac{1}{2}P(x|D_*)\right)\right)/\frac{1}{2} \\ &= P(D_p|x)P(x|D_p) + P(D_p|x)P(x|D_*). \end{aligned}$$

From this, we easily get:

$$P(x|D_p) = \frac{P(x|D_*) \cdot P(D_p|x)}{1 - P(D_p|x)} \quad (4)$$

Here, $P(x|D_*)$ is the learned prior and $P(D_p|x)$ is the learned differentiating function.

Having now explained how to learn a player-specific prediction function given a prior, the question remains as how to learn the prior. We learn the prior by (again) learning a differentiating function between a uniform distribution and the distribution formed by all examples collected from all players. Even though the uniform distribution is not accurate, this is not really a problem as sufficient training examples are available.

3. Experiments and Results

We observed cash games (max 9 players per game) played in an online poker room and extracted examples for players who played more than 2300 games. We randomly selected 20% of the games for the test set, while the remaining games were used to learn an opponent model. We learned one decision tree for all examlpes in the preflop phase and another for all remaning examples from the other phases, i.e, the *postflop* phases. The language bias used by Tilde (i.e., all possible tests for learning the decision tree) includes tests to describe the game history H_i at time i (e.g. game phase, number of remaining players, pot odds, previously executed actions etc.), board history B_i at time i, as well as tests that check for certain types of opponents that are still active in the game. For example, we may find tests such as "there is an opponent still to act in this round who is aggressive after the flop, and this player raised earlier in this game".

To evaluate our learning strategy, we report the log-likelihoods of the learned distributions and compare them with reasonable priors. A model with a higher

[1] We will consider the difference between small and large calls or bets as features in the learned corrective function.

likelihood directly allows an algorithm to sample and estimate the actions and outcome more accurately.

Figure 1 and 2 plot log-likelihoods averaged over 8 players for different training set sizes. The priors are clearly better than uninformed priors (i.e. not using learning). After having observed 200 games, in general the likelihood improves with the size of the training set.

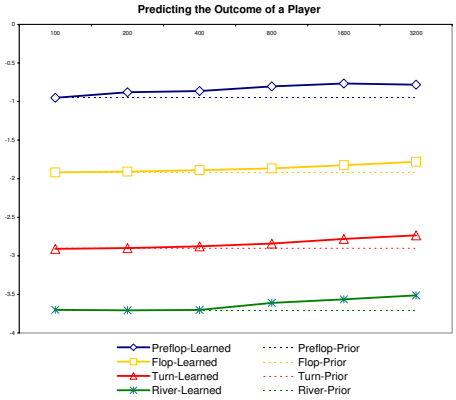


Figure 1. Experiment predicting the outcome, given board and game history. The x-axis represents the number of games used for the training set, and the y-axis the averaged log-likelihood scores on examples in the test set.

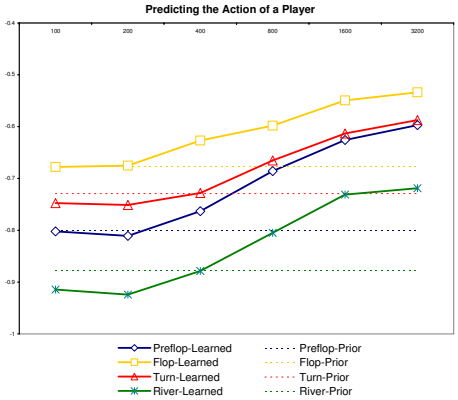


Figure 2. Experiment predicting the action, given outcome, board and game history. The axis are similar to those in Figure 1.

4. Conclusions

We presented a Bayes-relational opponent modeling system that predicts both actions and outcomes for human players in the game of No-Limit Texas Hold'em poker. Both these sources of opponent information are crucial for simulation and game-tree search algorithms, such as the adaptive tree search method by (Billings et al., 2006). The Bayes-relational opponent modeling approach starts from prior expectations about opponent behavior and learns a relational regression tree-function that adapts these priors to specific opponents. Our experiments show that our model adapts to specific player strategies relatively quickly.

Acknowledgments

Marc Ponsen is sponsored by the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. Jan Ramon and Kurt Driessens are post-doctoral fellow of the Research Foundation - Flanders (FWO).

References

Billings, D., Davidson, A., Schauenberg, T., Burch, N., Bowling, M., Holte, R. C., Schaeffer, J., & Szafron, D. (2006). Game-tree search with adaptation in stochastic imperfect-information games. *The 4th Computers and Games International Conference (CG 2004), Revised Papers* (pp. 21–34). Ramat-Gan, Israel: Springer.

Blockeel, H., & De Raedt, L. (1998). Top-down induction of first order logical decision trees. *Artificial Intelligence*, *101*, 285–297.

System modeling with Reservoir Computing

Francis Wyffels FRANCIS.WYFFELS@UGENT.BE
Benjamin Schrauwen BENJAMIN.SCHRAUWEN@UGENT.BE
Dirk Stroobandt DIRK.STROOBANDT@UGENT.BE
Electronics and Information Systems Department, Ghent University, Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium

Abstract

Reservoir Computing is a novel technique which can be applied to a wide range of applications. In this work we demonstrate that Reservoir Computing can be used for black box nonlinear system modeling.We will use Reservoir Computing to model the output flow of a heating tank with variable dead-time.

1. Introduction

Many control engineering techniques, in particular Model Predictive Control strategies, are based on process models. These models are obtained from physical principles or data-driven models (Camacho et al., 2007). Most of the data-driven models are black box models based on Analog Neural Networks (Camacho et al., 2007) which cannot cope with problems that have a strong temporal aspect. Therefore some research has focused on the use of Recurrent Neural Networks which have memory due to the loops inside the network. But unfortunately Recurrent Neural Networks are hard to train.

Reservoir Computing is a recently developed technique for very fast training of Recurrent Neural Networks which has been successfully used in many applications (Jaeger, 2001) such as speech recognition (Skowronski & Harris, 2007; Verstraeten et al., 2007), robot control (Antonelo et al., 2007) and time-series generation (Jaeger, 2001). To accomplish this, Reservoir Computing uses an untrained dynamic system (the reservoir), where the desired function is implemented by a linear, memory-less mapping from the full instantaneous state of the dynamical system to the desired output which can be trained by using linear regression techniques such as ridge regression (Wyffels et al., 2008a). A schematic overview is given in Figure 1.

In this work we will use Reservoir Computing to model the behavior of a nonlinear dynamical system with variable dead-time.

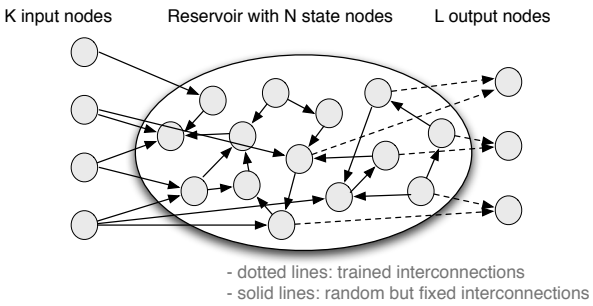


Figure 1. Schematic overview of the Reservoir Computing technique.

2. Experimental setup

The task at hand is the modeling of a heating tank with a variable cold water inlet and a hot water outlet. The heating element of the tank has a constant power thus, the outlet temperature is controlled by varying the cold water flow. Because the temperature of the outlet flow is measured after flowing through a long small pipe, the system has a variable dead-time which adds an extra difficulty in predicting the model. A full description of the plant can be found in (Cristea et al., 2005).

In contrast to most modeling techniques, we don't make any assumptions about the plant neither we split up the plant in different parts. We model both, the tank and the outlet pipe, by using only one reservoir consisting of 400 randomly connected band-pass neurons (see (Wyffels et al., 2008b) for an introduction). The spectral radius was tuned to give the reservoir a near-stable behavior. In order to give the reservoir more nonlinear properties each neuron adds an auxil-

iary input with a constant bias. Because of the variable dead-time, we needed to increase the fading memory of the reservoir by adding feedback from the output to all the neurons. The readout function was trained using 10,000 samples of random input-output examples extracted by simulation. Next, the reservoir was left predicting 3,500 samples based on its input, 1,000 samples were discarded for warming up to eliminate transient effects.

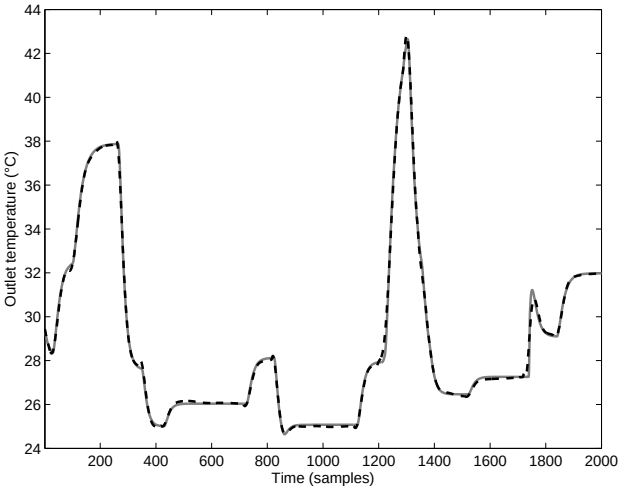


Figure 2. Validation of the model: real outlet temperature (gray solid line), predicted outlet temperature (dashed line).

3. Results

In Figure 2, a comparison of the desired outlet temperature and the predicted outlet temperature is given. Using the previously described reservoir configuration, the outlet temperature was predicted in connection to a variable input flow which was not seen by the reservoir during training. One can see that the reservoir is able to give accurate predictions of the outlet temperature.

4. Conclusions

In previous work, many techniques for system modeling are proposed. But most of them need a lot of experience in the application domain or are difficult to train. In this work we showed that by using Reservoir Computing one is able to model a nonlinear system with variable dead-time based on input-output recordings of the plant. No knowledge in the application domain was needed. For future work we wish to investigate the use of Reservoir Computing as a controller in control engineering tasks.

Acknowledgments

This work was partially funded by FWO Flanders project G.0317.05 and the Photonics@be Interuniversity Attraction Poles program (IAP 6/10), initiated by the Belgian State, Prime Minister's Services, Science Policy Office.

References

Antonelo, E. A., Schrauwen, B., & Campenhout, J. V. (2007). Generative modeling of autonomous robots and their environments using reservoir computing. *Neural Processing Letters*, *26*, 233–249.

Camacho, E., Rubio, F., Berenguel, M., & Valenzuela, L. (2007). A survey on control schemes for distributed solar collector fields. part i: Modeling and basic control approaches. *Solar Energy*, *81*, 1240–1251.

Cristea, S., de Prada, C., & De Keyser, R. (2005). Predictive control of a process with variable dead-time. *CD-Proceedings of the 16th IFAC World Congress.*

Jaeger, H. (2001). *The "echo state" approach to analysing and training recurrent neural networks* (Technical Report GMD Report 148). German National Research Center for Information Technology.

Skowronski, M. D., & Harris, J. G. (2007). 2007 Special Issue: Automatic speech recognition using a predictive echo state network classifier. *Neural Networks*, *20*, 414–423.

Verstraeten, D., Schrauwen, B., D'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, *20*, 391–403.

Wyffels, F., Schrauwen, B., & Stroobandt, D. (2008a). Regularization methods for reservoir computing. *Proceedings of the International Conference on Analog Neural Networks (ICANN).* (submitted).

Wyffels, F., Schrauwen, B., Verstraeten, D., & Stroobandt, D. (2008b). Band-pass reservoir computing. *Proceedings of the International Joint Conference on Neural Networks.*

Index