THE IEEE Intelligent Informatics BULLETIN

IEEE Computer Society
Technical Committee
on Intelligent Informatics

December 2017 Vol. 18 No. 2 (ISSN 1727-5997)

R&D Profile

Feature Articles

Announcements

On-line version: http://www.comp.hkbu.edu.hk/~iib (ISSN 1727-6004)

IEEE Computer Society Technical Committee on Intelligent Informatics (TCII)

Executive Committee of the TCII:

Chair: Chengqi Zhang
University of Technology, Sydney, Australia
Email: chengqi.zhang@uts.edu.au

Vice Chair: Yiu-ming Cheung
(membership, etc.)
Hong Kong Baptist University, HK
Email: ymc@comp.hkbu.edu.hk

Jeffrey M. Bradshaw
(early-career faculty/student mentoring)
Institute for Human and Machine Cognition, USA
Email: jbradshaw@ihmc.us

Dominik Slezak
(conference sponsorship)
University of Warsaw, Poland.
Email: slezak@mimuw.edu.pl

Gabriella Pasi
(curriculum/training development)
University of Milano Bicocca, Milan, Italy
Email: pasi@disco.unimib.it

Takayuki Ito
(university/industrial relations)
Nagoya Institute of Technology, Japan
Email: ito.takayuki@nitech.ac.jp

Vijay Raghavan
(TCII Bulletin)
University of Louisiana- Lafayette, USA
Email: raghavan@louisiana.edu

Past Chair: Jiming Liu
Hong Kong Baptist University, HK
Email: jiming@comp.hkbu.edu.hk

The Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society deals with tools and systems using biologically and linguistically motivated computational paradigms such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality. If you are a member of the IEEE Computer Society, you may join the TCII without cost at http://computer.org/tcsignup/.

The IEEE Intelligent Informatics Bulletin

Aims and Scope

The IEEE Intelligent Informatics Bulletin is the official publication of the Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society, which is published twice a year in both hardcopies and electronic copies. The contents of the Bulletin include (but may not be limited to):

1) Letters and Communications of the TCII Executive Committee

2) Feature Articles

3) R&D Profiles (R&D organizations, interview profile on individuals, and projects etc.)

4) Selected PhD Thesis Abstracts

5) Book Reviews

6) News, Reports, and Announcements (TCII sponsored or important/related activities)

Materials suitable for publication at the IEEE Intelligent Informatics Bulletin should be sent directly to the Associate Editors of respective sections.

Technical or survey articles are subject to peer reviews, and their scope may include the theories, methods, tools, techniques, systems, and experiences for/in developing and applying biologically and linguistically motivated computational paradigms, such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, and self-organization in the research and application domains, such as data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

Editorial Board

Editor-in-Chief:
Vijay Raghavan
University of Louisiana- Lafayette, USA
Email: raghavan@louisiana.edu

Managing Editor:
William K. Cheung
Hong Kong Baptist University, HK
Email: william@comp.hkbu.edu.hk

Assistant Managing Editor:
Xin Li
Beijing Institute of Technology, China
Email: xinli@bit.edu.cn

Associate Editors:
Mike Howard (R & D Profiles)
Information Sciences Laboratory
HRL Laboratories, USA
Email: mhoward@hrl.com

Marius C. Silaghi
(News & Reports on Activities)
Florida Institute of Technology, USA
Email: msilaghi@cs.fit.edu

Ruili Wang (Book Reviews)
Inst. of Info. Sciences and Technology
Massey University, New Zealand
Email: R.Wang@massey.ac.nz

Sanjay Chawla (Feature Articles)
Sydney University, NSW, Australia
Email: chawla@it.usyd.edu.au

Ian Davidson (Feature Articles)
University at Albany, SUNY, USA
Email: davidson@cs.albany.edu

Michel Desmarais (Feature Articles)
Ecole Polytechnique de Montreal, Canada
Email: michel.desmarais@polymtl.ca

Yuefeng Li (Feature Articles)
Queensland University of Technology, Australia
Email: y2.li@qut.edu.au

Pang-Ning Tan (Feature Articles)
Dept of Computer Science & Engineering
Michigan State University, USA
Email: ptan@cse.msu.edu

Shichao Zhang (Feature Articles)
Guangxi Normal University, China
Email: zhangsc@mailbox.gxnu.edu.cn

Xun Wang (Feature Articles)
Zhejiang Gongshang University, China
Email: wx@zjgsu.edu.cn

Publisher: *The IEEE Computer Society Technical Committee on Intelligent Informatics*
Address: *Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (Attention: Dr. William K. Cheung; Email:william@comp.hkbu.edu.hk)*
ISSN Number: *1727-5997(printed)1727-6004(on-line)*
Abstracting and Indexing: *All the published articles will be submitted to the following on-line search engines and bibliographies databases for indexing—Google(www.google.com), The ResearchIndex(citeseer.nj.nec.com), The Collection of Computer Science Bibliographies (liinwww.ira.uka.de/bibliography/index.html), and* ***DBLP*** *Computer Science Bibliography (www.informatik.uni-trier.de/~ley/db/index.html).*
© 2017 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the ***IEEE****.*

Multi-agent research in Transportation Domain

DIM Department (Decision, Interaction and Mobility)
LAMIH UMR CNRS 8201, Univ. de Valenciennes, France

I. Context

The LAMIH[1] laboratory, at the University of Valenciennes is organized into different departments (essentially, Automation, Mechanics, and Computer Science). The Computer Science department, named *Decision, Interaction and Mobility* (DIM), has two teams: operational research and distributed and embedded systems (OptiMob), and interactions and agents (InterA). InterA studies Human-Machine interaction and Distributed Artificial Intelligence / Multi-Agent Systems, (DAI / MAS).

Transportation, and more precisely road traffic simulation, is considered as one of the complex applications where MAS models open new research perspectives. Agent-centered, aka. microscopic, approaches are thus introduced to compete against previous macroscopic approaches. MAS tools take into account a larger variety of behaviors and richer environments, such as geographical databases and ontologies.

InterA investigated in the early 2000s application domains bound to the urban contexts: bus regulation and road traffic simulation. The animation of virtual pedestrians in an urban context contributed to the modeling of realistic environments for traffic simulations in towns.

II. Description

A. Bus traffic regulation

InterA aims to improve the quality of bus transportation by supplying a decision support system (DSS). Buses are modeled with a multi-agent approach. To respect precisely the theoretical schedules announced to the customers / users, it is necessary for the designer to propose a real time regulation. To take into account the incidents (which cause delays), the DSS allows for evaluating alternatives based on different possibles actions (e.g., modifying arrival times). These actions are often realized by a human operator because the process of regulation is not formalized. To meet the needs of the regulator and the satisfaction of the users, InterA designed a tool where agents (buses) change their actions (e.g., respecting the possibility for users to take another bus at a bus stop, in spite of the delays).

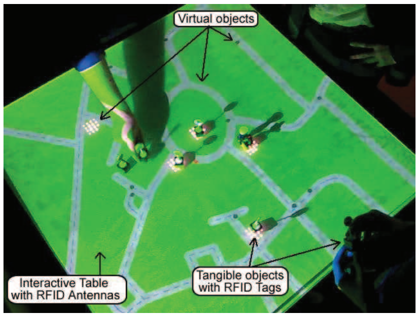
B. Behavioral animation of virtual pedestrians

A model of actions based on a vote mechanisms was applied for virtual agents (pedestrians). An educational simulator was designed for allowing the placing of a child-player in situations produced by interactions between pedestrians and vehicles. InterA contributed to the design and behavioral animation of virtual pedestrians. The actions modeled by these agents correspond to possible movements. A hybrid architecture was proposed, embedding cognitive and reactive properties. The cognitive reasoning searches a path following the adoption of a new goal. The reactive reasoning is happens during simulation. It enables reacting in a appropriate way at every step of the simulation, in a complex environment. The proposed model was also adapted to model the behavior of drivers.

C. Platform based on information spreading in a road traffic

A tool for road traffic simulation was developed to study the implementation of services within the mobility and intermodality contexts. This is part of the *Platform of simulation dedicated to the mobility services* project (PLAiiMOB) of the International Campus on Safety and Intermodality in Transportation (CISIT). Data exchanged between agents may help reacting to unexpected events (e.g., accidents) by reproducing a global behavior.

To facilitate supporting cartographic data from OpenStreetMap[2], an extension was implemented for an interactive table with tangible objects (TangiSense) based on RFID. Support was added for interactions between human users [4].


Virtual objects
Interactive Table with RFID Antennas
Tangible objects with RFID Tags

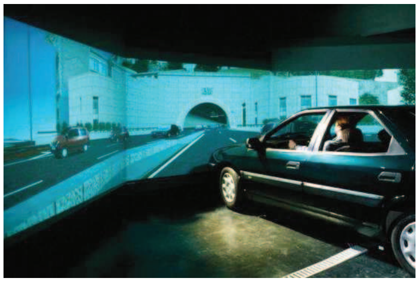
D. Road traffic simulation

InterA worked with Renault on the platform Scaner II for insertion problems on a road. A collaboration with IFSTTAR[3] on the ArchiSim simulation platform addresses traffic in urban contexts.

[1] french acronym for *Laboratoire d'Automatique, de Mécanique et d'Informatique, industrielles et Humaines*

[2] http://www.openstreetmap.org/

[3] french acronym for "Institut franais des sciences et technologies des transport, de l'amnagement et des rseaux".



In this context, a human agent is located into an environment simulating interactions with software agents and other humans. Models for critical situations (e.g., crossroads) were compared with measures of real traffic. Two studies addressed crossroads:

(i) The first one concerns the interactions between vehicles (agents) based on game matrices [5]. Each agent player chooses and selects its actions according to its potential payoff and the gain of other players.

(ii) The second study concerns the environment perceived by each agent as a set of constraints (CSPs) [1], [2]. Every agent tries to anticipate the behavior of the other agents and to detect situations of blocking by detection of incoherence of the network of constraints. This work highlighted the importance of the notion of non-normative agents in a global traffic. In this context, agents will not necessarily respect the traffic rules, considered until now a *inviolable* norm.

Motorbikes or emergency vehicles build virtual lanes (different physical lanes), and their actions based on their reasoning model are not easily described. Our model is based on the notion of *affordance*. The approach considers the properties of objects of the environment, the different possible actions, and also individual agent characteristics [3].

The process of perception of its environment by the driver is a preliminary process in decision-making. Inter-A's model takes into account the perceptive and attentional constraints of the driver. The model includes a double activity for the perception (passive and active), coupled with a quantitative limitation of percepts (due to court-term memorization).

References

[1] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espié. Anticipation based on constraint processing in a multi-agent context. *Journal of Autonomous Agents and Multi-Agent Systems, ISSN 1387-2532, Springer Netherlands Ed*, 17(2):339–361, Oct. 2008.

[2] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espié. Controlling non-normative behaviors by anticipation for autonomous agents. *Web Intelligence and Agent Systems*, 6(1):29–43, 2008.

[3] F. Ksontini, R. Mandiau, Z. Guessoum, and S. Espié. Affordance-based agent model for road traffic simulation. *Autonomous Agents and Multi-Agent Systems*, 29(5):821–849, 2015.

[4] S. Kubicki, Y. Lebrun, S. Lepreux, E. Adam, C. Kolski, and R. Mandiau. Simulation in contexts involving an interactive table and tangible objects. *Simulation Modelling Practice and Theory*, 31:116–131, 2013.

[5] R. Mandiau, A. Champion, J-M. Auberlet J-M., S. Espié, and C. Kolski. Behaviour based on decision matrices for a coordination between agents in urban traffic simulation. *Applied Intelligence*, 28(2):121–138, 2008.

Contact Information:

Prof. René Mandiau &
Prof. Sylvain Piechowiak

E-mail:
rene.mandiau@univ-valenciennes.fr
sylvain.piechowiak@univ-valenciennes.fr

Phone:
+ 33 3 27 51 14 38
Address:
LAMIH UMR CNRS 8201
Univ. of Valenciennes, France
Website:
http://univ-valenciennes.fr/LAMIH

Computer Science in the Information Age

John E. Hopcroft, *Life Fellow, IEEE*, Kun He, *Member, IEEE*

***Abstract*—For fifty years, computer science was focused on making computers useful. The emphasis was on algorithms, programming languages, and operating systems. Today the focus has changed to applications. Some of the drivers are the amount of computing power available, the quantities of data, and the internet. The impact on computer science created an interest in applications and interdisciplinary research. In this article we will review some of the recent work in clustering in social networks, learning theory, the size of data, and the digitalization of records and the need for privacy in computer systems.**

***Index Terms*—Community clustering, deep neural networks, large graph, privacy and security**

I. Clustering

EARLY work in the area of clustering was involved in partitioning vertices of social networks into disjoint clusters. This work evolved into finding overlapping communities. More recently there have been two major advances. One is as the size of social networks increased to billions of vertices, global clustering was replaced by local clustering. In global clustering if one partitions a billion vertex graph into ten or so clusters, the clusters would have hundreds of millions of vertices. With recursive partitioning one may find clusters of size a hundred, but the process is inefficient. Overlapping community detection is also more costly and hard to scale to large networks. Instead one might want local clusters for the seed members of interest. Say a cluster of fifty friends of three or so designated individuals. One method for doing this is to use spectral clustering. In spectral clustering one creates a matrix whose columns are the first few spectral vectors of the adjacency matrix of the network and then finds the minimum one norm vector in the space spanned by the columns [1, 2].

Instead of finding the minimum one norm vector in the space spanned by the singular vectors one might start random walks from a few vertices in the local community, but halt the process when the walks have converged to the stationary probabilities for the vertices in the community but not for the whole graph. The minimum one norm vector in the space spanned by these unconverged vectors will give the local community.

Another direction in clustering is finding hidden structure [3, 4]. Suppose one had images of a number of letters and the letters were in several shades of gray and different type fonts. If you were asked to cluster the images you would probably cluster them by letter. However, one could cluster them by color or by type font. The letter is the dominant structure. The type font and color of letters are the hidden structures, which are weaker and incoherent with the dominant clustering. Real networks have both dominant and several levels of hidden structure. How do you find the hidden structure in a social network?

Select your favorite clustering algorithm and find the dominant structure. Then weaken the dominant structure in the graph by randomly removing edges in the clusters. The again apply your clustering algorithm to the graph and it will find the hidden structure. If you go back to the original graph and now weaken the hidden structure and again find the dominant structure you will probably improve the clustering algorithms. If you alternately weaken the dominant and then the hidden structure you will converge to good clusterings of both the dominant and the hidden structure. Some real world networks have several levels of hidden structure that can be retrieved.

Applying this technique to the Facebook data of Rice University students [5] one gets a dominant structure and three levels of hidden structure. The dominant structure is the dorm the student lives in and one of the hidden levels is the year of the student, freshman, sophomore, junior, or senior. The other two levels have high modularity and are incoherent with earlier levels but we were unable to identify what they corresponded to; maybe sports or other interests.

II. Deep Learning

Machine learning has been extremely valuable for a number of years. Its major tool was the support vector machine. However, in 2012 advances in deep learning changed the field. Until 2012 reducing the classification error in the ImageNet ILSVRC competition [6,7] was very small. ImageNet has 1.2 million images classified in 1,000 categories.

The task is to train a network on a training set of the images and see how well it generalizes to a test set. Prior to 2012 the error rate was approximately 25%. Then in 2012, AlexNet dropped the error rate to 15%, a truly major improvement. Two years later GoogleNet reduced the error rate to 6.7% and in 2015 ResNet reduced it further to 3.6%. The human error rate with training is about 5%. These deep networks outperform humans. Since 2012 deep networks have been applied in many applications and they have performed exceptionally well although little is known as to why they work so well.

This work was supported by US Army Research Office (W911NF-14-1-0477), and National Natural Science Foundation of China (61772219).

John E. Hopcroft is with the Department of Computer Science, Cornell University, Ithaca, NY 14853, USA (e-mail: jeh@cs.cornell.edu).

Kun He, is with the Department of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: brooklet60@hust.edu.cn).

One of the issues with supervised learning is the lack of sufficiently large labeled data sets. Recently there has been progress in unsupervised learning. Instead of training a deep network to classify images one can train the network to reproduce the image. This results in good internal representations of images and raises the question of what internal gates learn. This has significantly increased the research on unsupervised learning.

There are many research problems that might shed some insight into deep learning. Do some gates learn the same thing as other gates? If we train a network twice from different random starting positions do the gates learn the same things or does the network develop entirely different way to classify images [8]? What determines what a gate learns. How does what a gate learns evolve over time. Do gates in the first or second convolution levels learn features of images independent of what the images are? In training a deep network one encounters many local minima with the same error rate. Which local minima will generalize better? Experimentally broader local minima seem to generalize better. This may be because the error function for the training set is very close to the true error function and a small shift will not disturb a broad local minimum as much as it will on a sharp local minimum.

Generative adversarial networks [9] have become very popular. If one wants to generate realistic looking images one might train an adversarial network to distinguish between real images and generated images. Then they could feed the adversarial component the output of the generative component and train the generative component until the adversarial component could not distinguish between the generated image and a real image. At that point one trains the adversarial component to do better. By interacting with the two units one can generate good images.

Another application might be language translation. In the past one used pieces of text where one had the same text in both languages to train a network. But if one does not have sufficient samples in both languages they could use an adversarial network as follows. To create a translator from English to German one first build a translator that will take an English sentence and output German words. The one build an adversarial network that distinguishes between German words and German sentences. Finally, one takes the output of the first devices that outputs German words and builds a device that creates English sentences and compares the sentences generated to the original sentence. Training the three networks forces the output of the first device to be a German sentence rather than just German words. And training the last device forces the German sentence to be a true translation.

There are many other problems researchers are exploring. An interesting one is how one can fool a deep network by making changes to an image that are so small a human cannot detect the changes, but cause the deep network to change the classification of the image [10]. All of a sudden what appears to be an image of a cat is classified as a car. The reason this is possible is that the set of images of a cat map into a manifold of dimension much smaller that the dimension of the activation space. Thus if one moves in activation space perpendicular to the surface of the manifold one is likely to change the classification.

AI programs do not extract the essence of an object and understanding its function or other important aspects. It may be another 40 years before we have another information revolution where function or other property is extracted. This will lead to an enormous range of intellectual ability.

III. Size of Graphs

In early research in the 1960's, graphs had ten to fifteen vertices. When the computer came graph size increased to 1,000 vertices, with faster computers 10^6 vertices. Then sparse graph such as the world wide web came with billions of vertices. Today we compute with 10^{100} graphs with 10^{100} vertices. Remember the number of atoms in the visible universe is only 10^{70}. How do we store a graph with 10^{100} vertices in the computer? We don't. One can do a random walk on a graph without storing it in the computer. All they need is an algorithm which given a vertex will identify the adjacent vertices. All we need to keep is the current vertex the random walk is at. However, how long does it take for a random walk to converge to its stationary probability? It turns out that if the graph is an expander, the random walk will converge to its stationary probability in logarithmic number of steps. For the 10^{100}vertex graph this means some number of step within 100 times some constant. Problems in of this size occurring in many applications are handled every day.

Given the size of data and graphs that are dealt with frequently requires that we randomly sample the data. This might require a random sequence which is a sequence with no short description. How can you store such a sequence? You don't. Instead you use a pseudo random sequence. This raises the question of how much randomness do you need. Usually one only needs two-way randomness. A sequence of zeros and ones is two-way pseudo random if each element is equally likely to be a zero or a one and given the value of one element in the sequence it does not give any information about any other element. If I give you two elements, I may be giving you information about all elements in the sequence.

An example where one uses randomness is in determining the number of distinct elements in a sequence. Suppose you work for a major chain store such as Walmart and want to know how many customers you have. You have access to a data stream of every purchase world-wide along with a credit card number associated with the purchase. You wish to count the number of card numbers. Each number is 16 digits long. You could set up a Boolean vector of length 10^{16}, or you could keep a linked list of numbers, or you could use a hash table or some other technique. However, if you are happy with a good estimate you can do this with only one word of storage. Keep track of the minimum credit card number. If you lay out a sequence of integers from one to $10^{16}$and mark every number you see, the expected distance between elements will be the 10^{16}divided by the number of distinct elements. Hence the minimum is approximately 10^{16} divided by the number of distinct elements. Thus a good approximation for the number of distinct elements is 10^{16} divided by the minimum number seen.

One problem is that algorithm assumes the elements are random. This is not likely to be so since the credit card numbers might not be issued randomly. Thus you want to use a hash function to make the data statistically independent. We cannot store a hash function that will give full independence. However, only two-way independence is needed.

IV. Digitalization of medical records

As we digitize medical records the need for privacy and security becomes critically important. For example, if my entire medical history was digitized and I became ill somewhere in the world, I would like my doctor to be able to see my entire medical history to give me the best possible treatment. However, I do not want my insurance company to see my entire medical history. In fact, the insurance company does not need to see my medical record at all. All they need is a rigorous proof that they owe a doctor a certain amount of money. Medical researchers would like to access every one's medical record to improve medical techniques. How do we allow them to access statistical information without letting them have access to any individual information? Two techniques are emerging to help with this issue: zero knowledge proofs [11] and differential privacy [12].

A. Zero Knowledge Proof

A zero knowledge proof of a statement is a proof that the statement is true without providing any other information. To illustrate a zero knowledge proof, consider the game Sudoku. I can prove that I know how to fill in a Sudoku board without giving you any information on how to do it. I will take pieces of cardboard and write the appropriate number on each piece and place the card board pieces down over the appropriate squares so the numbers are not visible. Now you want to check that I have correctly filled in the first row. I pick up the cardboard pieces from the first row and shuffle them and show you that I have the correct numbers for the row. You check every row, column and three by three square and see that I have correctly filled in each. Actually this is not quite sufficient to prove that I have a correct solution since you don't know that I put the card pieces back in the same order each time. However, if I do not have a solution you will detect it with some probability and as you ask about more rows, columns and three by three squares with repetitions, you can drive the probability that I do not have a solution to zero.

A similar technique can be applied to three coloring a graph so that no two adjacent vertices have the same coloring. This is an NP-complete problem and there is no known polynomial time algorithm for the three coloring problem. Suppose you have a graph with a million vertices that you want to color and I have a business where I provide colorings. However, we cannot do business since we don't trust one another and you do not want to pay me until you know I actually have a coloring for your graph and I do not want to show you the coloring until you pay me. The solution is to give you a proof that I have a coloring without giving you any information as to the coloring. Again we use a zero knowledge proof.

For each vertex I place the appropriate color in an envelope and seal it. You ask to see the color of two adjacent vertices and I allow you to open the two appropriate envelops. This gives you no information about a coloring since one could permute the colors of the vertices to achieve the two colors of these two vertices. However, if I allow you to see another vertex, I have given you some information. So instead I destroy all the envelopes, permute the colors on the graph and recreate envelops with the appropriate color for each vertex. This sound like a lot of work. However, we don't use physical envelops, instead we agree on a digital encoding. When you want to see two vertices I give you the key to decode those two vertices. Since this is all done electronically it takes only a few minutes to convince you I indeed have a coloring and we can do business.

These toy problems are just examples of zero knowledge proofs.

B. Differential Privacy

Privacy is needed in many business applications involving car guidance, supply chains, and transportations systems. For example, the route guidance system in your car does not give you the best route guidance since it does not know the conditions of back roads and thus it keeps you on main roads. If the guidance system could improve routing and reduce mileage by a few percent, it would be a savings in millions of dollars of gasoline.

The route guidance system may record your GPS coordinates for the last month. If when you take your car in for service the GPS coordinates were down loaded, the route guidance system could improve its guidance by making use of the knowledge of local drivers. However, one may not want the GPS coordinates down loaded since one could determine the car owner by where the car was parked at night, where one works, shops, etc. However, if we could provide the condition of back roads without reveling any individual information it would be a success. Many systems that will be created in the future will face such problems of privacy.

V. Conclusion

The availability of large amounts of data, enormous computing power, the internet, and advances in AI are driving an information revolution. Intellectual tasks will be automated, changing the nature of work. In this paper we discussed some of the applications and advances that will influence our future. Those individuals, institutions, and nations that position themselves for the future will benefit enormously.

References

[1] Kun He, Yiwei Sun, David Bindel, John E. Hopcroft and Yixuan Li, "Detecting overlapping communities from local spectral subspaces," in

15th IEEE International Conference on Data Mining (ICDM), Atlantic City, USA, 2015, pp. 769-774.

[2] Yixuan Li, Kun He, David Bindel and John E. Hopcroft. "Uncovering the small community structure in large networks," in 24th International Conference on World Wide Web (WWW), Florence, Italy, 2015, pp. 658-668.

[3] Kun He, Sucheta Soundarajan, Xuezhi Cao, John E. Hopcroft and Menglong Huang, "Revealing multiple layers of hidden community structure in networks," CoRR abs/1501.05700, 2015.

[4] Kun He, Yingru Li, Sucheta Soundarajan and John E. Hopcroft, "Hidden community detection in social networks," Information Sciences, vol. 425, pp. 92-106, Jan. 2018.

[5] Alan Mislove, Bimal Viswanath, P. Krishna Gummadi, Peter Druschel, "You are who you know: inferring user profiles in online social networks," in Proceedings of the 3rd International Conference on Web Search and Web Data Mining (WSDM), New York, NY, USA, 2010, pp. 251-260.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Fei-Fei Li, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, April 2015.

[7] ImageNet Large Scale Visual Recognition Challenge (ILSVRC), [Online]. Available: http://www.image-net.org/challenges/LSVRC/.

[8] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E. Hopcroft, "Convergent learning: Do different neural networks learn the same representations? " in ICLR, 2016.

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, Yoshua Bengio, "Generative adversarial networks," in Advances in Neural Information Processing Systems(NIPS). Montreal, Quebec, Canada, 2014, pp. 2672-2680.

[10] Anh Nguyen, Jason Yosinski, and Jeff Clune, "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.

[11] Blum, Manuel, Feldman, Paul; Micali, Silvio. "Non-interactive zero-knowledge and its applications," in Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC), Chicago, Illinois, USA, 1988, pp. 103–112.

[12] Cynthia Dwork. "Differential Privacy, Automata, Languages and Programming," in 33rd International Colloquium (ICALP), Venice, Italy, 2006, pp. 1-12.

Redescription Mining: An Overview

Esther Galbrun and Pauli Miettinen

***Abstract*—In many real-world data analysis tasks, we have different types of data over the same objects or entities, perhaps because the data originate from distinct sources or are based on different terminologies. In order to understand such data, an intuitive approach is to identify the correspondences that exist between these different aspects. This is the motivating principle behind *redescription mining*, a data analysis task that aims at finding distinct common characterizations of the same objects. This paper provides a short overview of redescription mining; what it is and how it is connected to other data analysis methods; the basic principles behind current algorithms for redescription mining; and examples and applications of redescription mining for real-world data analysis problems.**

***Index Terms*—Redescription mining, alternative characterizations, visualizations, data mining.**

I. Introduction

CONSIDER an ecologist who wants to understand the bioclimatic conditions that define species' habitats.[1] She has data on the regions where the species live and on the bioclimatic conditions (e.g. monthly average temperatures and precipitation) of those regions, and she would like to find explanations such as the following.

> *The areas inhabited by either the Eurasian lynx or the Canada lynx are approximately the same areas as those where the maximum March temperature ranges from* $-24.4\,°\mathrm{C}$ *to* $3.4\,°\mathrm{C}$.

The above is an example of a *redescription*. It *describes* regions of the earth in two different ways; on the one hand, by the fact that certain species inhabit them, and on the other hand, by the fact that they have a certain climate. We can see the areas described above in Figure 1. The medium purple colour denotes the areas where both of the above conditions hold (inhabited by one of the lynx species and with maximum March temperatures in the correct range), light red denotes the areas inhabited by one of the lynx species but where March temperatures are out of the range, and dark blue denotes the areas where the maximum March temperature is in the correct range but neither of the lynxes is found.

Informally, a redescription is a pair of descriptions, both describing roughly the same entities (here, geographical regions). And, as we can see from this example, both the descriptions and what they describe can be of interest. The ecologist is interested in the descriptions in order to understand the *model* of the niche and in the geographical areas in order to understand where the niche is (or is not). While such redescriptions could be constructed manually, the goal of *redescription mining* is to find them automatically without any information other than the raw data (and some user-provided constraints). For instance, the ecologist should not have to define the species she is interested in. Rather, the goal of redescription mining is to find all redescriptions that characterize sufficiently similar sets of entities and adhere to some simple constraints regarding, for example, their type and complexity and how many entities they cover.

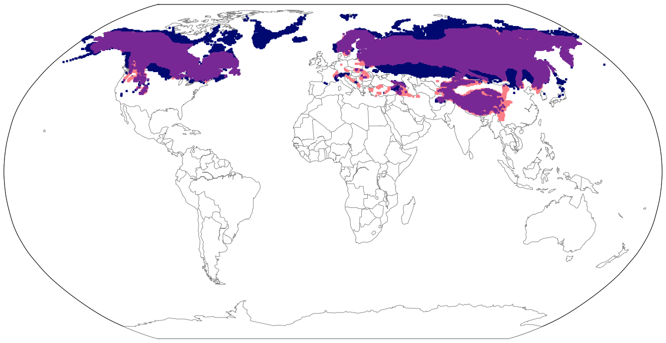


Figure 1. Map of a bioclimatic niche. The areas inhabited by either the Eurasian lynx or the Canada lynx (light red and medium purple) and the areas where the maximum March temperature is between $-24.4\,°\mathrm{C}$ and $3.4\,°\mathrm{C}$ (dark blue and medium purple).

In this article, we present a brief overview of redescription mining. We start by giving the formal definition of the task in the next section. In Section III, we explain the main algorithmic ideas used in redescription mining, before discussing the techniques for removing redundant redescriptions, in Section IV. Sections V, VI, and VII contain, respectively, a brief study of the existing redescription mining tools, an outline of some example applications, and a summary of related methods. We present some open problems and directions to future work in the concluding Section VIII. We will not delve into the details of the different algorithms, tools, or applications. Such details can be found in the original publications, as well as in our recent tutorials[2] and book [1].

II. Formalizing the Task

Redescription mining can, of course, be applied to other use cases than bioclimatic niche finding, but we will use that example as our running example throughout this article. In this section we provide the formal definition of redescription mining. Our definition uses the so-called table-based model [1]; other, more general, formulations exist (see [1]), but that generality is unnecessary for the discussion in this article.

E. Galbrun is with Aalto University, Finland. email: esther.galbrun@aalto.fi
P. Miettinen is with Max Planck Institute for Informatics, Germany. email: pauli.miettinen@mpi-inf.mpg.de

This article is based on our recent tutorials called *An Introduction to Redescription Mining* at ECMLPKDD '16 and SDM '17, and on our upcoming book *Redescription Mining* [1].

[1] In ecology, the task is known as *bioclimatic niche (or envelope) finding* [2, 3].

[2] Slides available at http://siren.mpi-inf.mpg.de/tutorial/

In the table-based model, the data are arranged as a table (or tables; we will discuss that below). The rows of the table $\boldsymbol{D}$ correspond to the *entities* in the data and the columns correspond to the *attributes*; for example, in the bioclimatic niche finding example, the geographical regions are the entities, the table contains one row for each location where observations have been recorded, and the species and bioclimatic variables (that is, the observations) are the attributes. The value of attribute j in entity i is denoted as d_{ij}. The attributes can be of different types, such as binary, categorical, or numerical, and some entity–attribute values might be missing. In our example, the presence or absence of a species in a region constitutes a binary attribute, whereas the bioclimatic variables, such as temperatures or precipitations, are recorded as continuous numerical attributes.

A redescription is a pair of descriptions, and we formalize the descriptions as Boolean queries over the attributes. Each *predicate* in the queries assigns a truth value to (observed) entity–attribute pairs, that is, to the elements of a column of the data table. The queries over the predicates and their negations – together known as *literals* – in turn assign a truth value to each entity. The query can, in principle, be an arbitrary Boolean function of the literals, but it is common to restrict the queries to some *query language* for the sake of interpretability and efficiency of computation. Common query languages include *monotone conjunctive queries*, *linearly parsable queries* (where each variable can appear at most once and both conjunction and disjunction operators have the same precedence), and *tree-shaped queries* (a special case of disjunctive normal forms, encoding the paths from the root to the leaves in a decision tree).

Applying this formalism to our example niche redescription, the query corresponding to *'The areas inhabited by either Eurasian lynx or Canada lynx'* could be written as

$$\mathit{Eurasian\ lynx} \lor \mathit{Canada\ lynx}\ ,$$

and the query *'maximum March temperature ranges from* $-24.4\,°\mathrm{C}$ *to* $3.4\,°\mathrm{C}$*'* as

$$[-24.4 \leq t_3^+ \leq 3.4]\ .$$

To avoid tautological redescriptions (e.g. 'Eurasian lynx lives where Eurasian lynx lives'), we require that the queries do not share any attributes. In many applications, the attributes have a natural division into two disjoint sets. In our running example, the species form one set of attributes and the bioclimatic variables form the other set. In these cases, it is natural to model the data, not as a one, but as two data tables; one table for the species and one table for the bioclimatic variables, in our example. In this setup, the queries of a redescription are required to be over attributes from different tables.

The *support* of a query q, $\mathrm{supp}(q)$, is the set of entities (rows) that satisfy the query.[3] The support of the query $\mathit{Eurasian\ lynx} \lor \mathit{Canada\ lynx}$ contains the regions depicted in light red and in purple in Figure 1, while the support of the query $[-24.4 \leq t_3^+ \leq 3.4]$ contains the regions depicted in dark blue and in purple.

To form a good redescription, the queries should explain roughly the same entities, that is, their supports should be similar. The most common choice for measuring the similarity of the supports is the *Jaccard (similarity) index* J, defined as

$$J(p,q) = J(\mathrm{supp}(p), \mathrm{supp}(q)) = \frac{|\mathrm{supp}(p) \cap \mathrm{supp}(q)|}{|\mathrm{supp}(p) \cup \mathrm{supp}(q)|}\ .$$

The Jaccard index is by no means the only possible similarity measure, but it is by far the most common one. Its use can be motivated in many ways. For example, when using algorithms based on decision-tree (see Section III), it has a natural connection to the information gain splitting criteria [4]. On the other hand, if we consider redescription mining as mining bi-directional association rules (see again Section III), the Jaccard index of a redescription can be interpreted as the lower bound on the confidence of the corresponding association rules $\mathrm{conf}(p \Rightarrow q)$ and $\mathrm{conf}(q \Rightarrow p)$.

How similar should their supports be for the pair (p,q) to be considered a valid redescription is something the user must decide, depending on the data and her needs. Therefore, we say that the supports of p and q are similar enough if $J(p,q) \geq \tau$ for some user-specified constant $\tau \in [0,1]$, and write $p \sim q$.

We can now define what a redescription is. For data that consist of two tables, $\mathbf{D}_1$ and $\mathbf{D}_2$, a redescription is a pair of queries (p,q) expressed over attributes from $\mathbf{D}_1$ and $\mathbf{D}_2$, respectively, such that $p \sim q$. In addition, a redescription might have to satisfy other constraints specified by the user, such as limitations on the size of the support, the maximum p-value, or the complexity of the queries (in terms of the number of variables involved, for instance). Then, the goal of *redescription mining* is to find all valid redescriptions $p_i \sim q_i$ that also satisfy the other potential constraints.

III. Algorithms

Readers familiar with classification and association rule mining might have noticed similarities between redescription mining and these two core data mining tasks. These two tasks provide basic techniques that have been adapted to develop algorithms for mining redescriptions.

Consider a case where one query is fixed and the goal is to find a matching query to make a good redescription; taking the support of the fixed query as the labels of the entities, the problem becomes that of a binary classification problem (see, e.g. [5, Ch. 10]). This fact has inspired a family of iterative algorithms that alternate between the views to build the redescriptions. These algorithms derive target labels from a query obtained at a previous iteration and use classification techniques, typically decision tree induction, to build a matching query in the next iteration. The first algorithm proposed for redescription mining, the `CARTwheels` algorithm [4], is based on the idea of alternatively growing decision trees over one data table with only binary attributes. The decision-tree-based methods for arbitrary data types introduced by Zinchenko et al. [6] also belong to this family of redescription mining algorithms. Predictive clustering trees were used in a similar manner for mining redescriptions by Mihelčić et al. [7].

[3] Some sources call this set the *support set* and reserve the term support for what we call the *size* of the support.

On the other hand, association rule mining (see, e.g. [5, Ch. 4]) can be seen as a precursor of redescription mining, with the latter allowing for more complex descriptions and focusing on equivalences instead of implications [4]. This inspired algorithms that first mine queries separately from the different views before combining the obtained queries across the views into redescriptions. The method proposed by Zaki and Hsiao [8] and the MID algorithm of Gallo et al. [9] both belong to this second family of algorithms. Along similar lines, Zaki and Ramakrishnan [10] studied exact and conditional redescriptions over binary data, focusing on conjunctive queries, while Parida and Ramakrishnan [11] studied the theory of exact redescriptions over binary attributes, where the queries are pure conjunctions, whether in monotone conjunctive normal form or monotone disjunctive normal form.

A third approach for mining redescriptions consists in growing them greedily. Such a strategy of progressively extending the descriptions by appending new literals to either query, always trying to improve the quality of the redescription, was first introduced as the Greedy algorithm of Gallo et al. [9]. Building upon this work, the ReReMi algorithm Galbrun and Miettinen [12] extended the approach to handle categorical and numerical attributes along with binary ones and use a beam search to keep the current top candidates at each step instead of focusing on the single best improvement.

The proposed algorithms can also be divided between exhaustive and heuristic strategies. Mine-and-pair algorithms based on association rule mining techniques are typically exhaustive. Alternating algorithms based on decision tree induction and algorithms that grow the queries greedily typically rely on heuristics.

While the first algorithms only considered binary attributes, more recent ones also allow to handle numerical and categorical attributes, possibly including missing entries. In this latter case, when calculating the supports of the queries and the similarity of the supports, a choice needs to be made about how to handle the entities for which the status of the queries cannot be determined due to missing values. Potential approaches include – but are not limited to – assuming that the queries always evaluate false on such entities [7] or assuming that they evaluate true or false depending on what is the most or the least favorable in terms of support similarity [12]. In fact, evaluating whether there is a way the query can evaluate true is NP-hard in general, though this is not the case with any of the query languages that are used with the existing algorithms. Of course, the actual mining algorithm also has to support missing values. For example, in algorithms using decision tree induction, the induction procedure must be able to handle missing values.

IV. Sets of redescriptions

Redescription mining, as defined above, is an exhaustive enumeration task, the goal being to output *all* valid redescriptions that satisfy the constraints. This is a common approach in data mining (cf. frequent pattern mining), but it can yield many redundant redescriptions. Filtering away the redundant redescriptions, however, requires us to define what redescriptions are redundant.

Perhaps the simplest approach is to consider the supports of the queries. We can order all (valid) redescriptions descending in their similarity, take the topmost redescription, move it to the list of non-redundant redescriptions, and mark the entities in its support 'used'. We can then re-evaluate the remaining redescriptions while only taking into account the non-used entities. All redescriptions that are deemed invalid (e.g. their support becomes too small or their Jaccard index too low) are considered redundant and removed. We repeat the process with the remaining redescriptions and entities until either the list of redescriptions or the set of entities becomes empty.

This simple approach can filter out too many redescriptions, as it only considers their support and not the attributes that appear in the descriptions. Kalofolias et al. [13, 14] presented another approach for defining redundant redescriptions based on maximum-entropy distributions and the subjective interestingness approach of De Bie [15]. They model the data using a maximum-entropy distribution that is constrained so that the already-observed redescriptions have a high probability (or are certain) to occur. The other redescriptions are then ranked based on their likelihood of being true in a data following this model. The redescription that is the least-likely (i.e. the most surprising) is added as a constraint, the model is re-learned, and the remaining redescriptions are re-evaluated.

V. Tools

The Siren tool was developed for mining, visualizing, and interacting with redescriptions [16–18]. It provides a complete environment for redescription mining, from loading the data to finally exporting the results into various formats, through mining, visualizing, and editing the redescriptions.

Having good visualizations is crucial, of course, when designing a tool for visual data analysis. Indeed, visualization is the key to understanding the results of the mining process and we designed several visualizations for redescriptions. *Maps*, like the one presented in Figure 1, are a great way to understand where the queries hold (and do not hold), but require, naturally, that the entities are associated with geographical locations. *Parallel coordinates* plots are especially useful to understand the conditions appearing in the queries, as they allow to visualize the range of values selected by the predicates. Our example redescription depicted in a parallel coordinates plot is shown in Figure 2.

For redescriptions using decision tree induction and for tree-shaped queries more generally, *tree diagrams* reveal the tree structure underlying the queries, facilitating the interpretation of descriptions that can otherwise appear rather convoluted. A tree-shaped equivalent of our example redescription depicted in a tree diagram is shown in Figure 3.

Visualizations in Siren are linked, so that the user can highlight an entity across different visualizations of the same redescription, or interactively adjust the thresholds in the queries through the parallel coordinate plot, for instance. In addition, the tool allows to use different levels of automation when mining redescriptions, from letting the algorithm run fully automatically given a set of parameters, to letting the user edit the results fully manually, through partial automation where

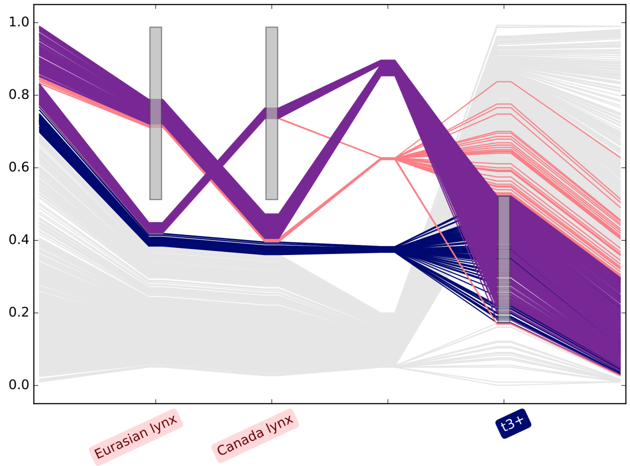


Figure 2. Parallel coordinates plot of our example redescription. Every line corresponds to one geographical location (entity) and the colours of the lines are as in Figure 1, except that grey correspond to locations where neither of the queries hold. The plot has three vertical axes corresponding to the three attributes in the redescriptions. The grey boxes in these axes correspond to the range of the values of the corresponding variable in the query; if a line passes through a gray box, the predicate corresponding to the attribute evaluates true for this entity.

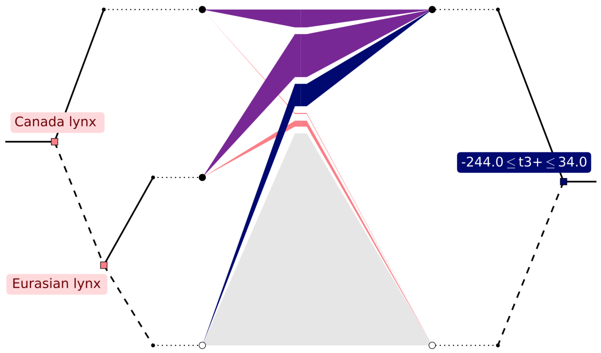


Figure 3. Tree diagram of a tree-shaped equivalent of our example redescription. Solid leaf nodes correspond to paths in the tree where the queries evaluate true, while empty leaves correspond to paths where the queries evaluate false. Lines between the two trees are as in Figure 2.

the algorithm extends and optimizes candidate redescriptions provided by the user.

Siren also allows to perfom support-based filtering on a set of redescriptions as explained in Section IV: the redescriptions are reranked and the redundant ones are marked.

Recently, Mihelčić and Šmuc [19] proposed a tool called InterSet for visualizing and working with sets of redescriptions. The tool allows to cluster redescriptions based on their shared attributes and shared entities. The user can also visualize the statistics of a set of redescriptions, such as the distribution of their Jaccard indices or of their pairwise support overlap, and filter the redescriptions based on those statistics.

VI. Applications

Redescription mining has been applied in various domains. Here, we present three examples from ecology, from biology and from social and policital sciences, respectively.

Instead of modelling the distributions of species directly, as in the niche finding example presented earlier, one might look at the distributions of functional traits of species. Galbrun et al. [20] consider dental traits of large plant eating mammals and bioclimatic variables (derived from temperature and precipitation records) from around the globe, looking for associations between teeth features and climate. The teeth of plant-eating mammals constitute an interface between the animal and the plant food available in its environment. Hence, teeth are expected to match the types of plant food present in the environment, and dental traits are thus expected to carry a signal of environmental conditions. In this study, three global zones are identified, namely a boreal-temperate moist zone, a tropical moist zone, and a tropical-subtropical dry zone, each associated to particular teeth characteristics and a specific climate.

Mihelčić et al. [21] use redescription mining to relate clinical and biological characteristics of cognitively impaired patients, with the aim of improving the early diagnosis of Alzheimer's disease. In this study, one data table consists of biological attributes derived from neuroimaging, from blood tests, and from genetic markers, for instance, while the other data table contains clinical attributes that record patients' answers to several questionnaires, observations by physicians, and results of cognition tests. The results obtained largely confirmed the findings of previous studies. In addition, they highlighted some additional biological factors whose relationships with the disease require further investigation, such as the pregnancy-associated plasma protein-A (PAPP-A), which they found to be highly associated with cognitive impairment in Alzheimer's disease.

Galbrun and Miettinen [22] applied redescription mining to analysing political opinion polls. Specifically, they used data from Finnish on-line voting advice applications, where candidates in the Finnish parliamentary elections have answered to a number of questions regarding their opinions on political matters, and had also provided socio-economical background data. Galbrun and Miettinen [22] analysed, first, the correlations between the socio-economical status and the political opinions of candidates, and, second, compared the answers of candidates who run for both 2011 and 2015 elections between these years. Their findings partially followed the party platforms, but they also found unsuspected connections; for example, candidates who were over 37 years old or who had children were not strongly supporting legalizing euthanasia, and vice versa.

VII. Related methods

As we have seen, the work on redescription mining has significantly expanded and diversified since the task was first formalized by Ramakrishnan et al. [4]. Problem variants have also been introduced: *storytelling* aims at building a chain of redescriptions linking given objects or queries while *relational* *redescription mining* aims to find redescriptions in heterogenous networks.

Beside classification and association rule mining (see Section III), the task also has connections with subgroup discovery, clustering and multi-view approaches, in particular.

In subgroup discovery [23], the input contains features and a target variable over observations, and the goal is to find queries that describe groups that have an 'interesting' behaviour in the target variable, that is, groups of entities that have different statistical properties (e.g. average) in the target variable when compared to the rest of the observations.

Clustering is a classical unsupervised data analysis method with the goal of grouping the entities in such a way that entities in the same group are as similar to each other as possible, and the objects in different groups are as dissimilar from each other as possible. A query can be interpreted as selecting a subset of the attributes and a group of entities that are in some sense 'similar' to each other, although not in the classical sense (e.g. of having short Euclidean distance). Among clustering techniques, redescription mining is most related to subspace clustering [24] and biclustering [25].

An important feature of redescriptions is their ability to describe data from different points of view, i.e. their 'multi-view' aspect. Other examples of *multi-view data mining methods* include *multi-view clustering* [26], where the attributes are divided into two views and the clustering is done separately over each view; *multi-view subgroup discovery* [27], where the subgroup discovery is done over multiple views; and various *multi-view matrix and tensor factorization* [28–30], which use (partially) the same factors to decompose multiple matrices or tensors.

VIII. Conclusion and future work

Redescription mining is a powerful data analysis technique that is gathering wider interest, among data analysis researchers and practitioners alike. The availability of efficient algorithms that can handle heterogeneous data types has undoubtably contributed to the increasing adoption. Yet, redescription mining is, in many ways, in its infancy, and there are still many interesting open questions to be addressed. Developing redescription mining methods that work over time series data is one important future direction. Another interesting direction is to add predicates that are functions of the attributes, such as square roots, logarithms, squares, and so on, and perhaps also multivariate composite attributes. This would naturally allow the query to capture more complex structures, but the exact functions would have to be application-dependant. Finally, redescription mining could also be extended to more complex data (*relational redescription mining* [31] can be seen as one step in that direction), such as graphs and multimodal (e.g. tensor) data.

References

[1] E. Galbrun and P. Miettinen, *Redescription Mining*. Springer, 2018.

[2] J. Soberón and M. Nakamura, "Niches and distributional areas: Concepts, methods, and assumptions," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 106, no. Supplement 2, pp. 19 644–19 650, 2009.

[3] J. Grinnell, "The niche-relationships of the California thrasher," *The Auk*, vol. 34, no. 4, pp. 427–433, 1917.

[4] N. Ramakrishnan, D. Kumar, B. Mishra, M. Potts, and R. F. Helm, "Turning CARTwheels: An alternating algorithm for mining redescriptions," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, 2004, pp. 266–275.

[5] C. C. Aggarwal, *Data Mining: The Textbook*. Cham: Springer, 2015.

[6] T. Zinchenko, E. Galbrun, and P. Miettinen, "Mining predictive redescriptions with trees," in *IEEE International Conference on Data Mining Workshops*, 2015, pp. 1672–1675.

[7] M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "Redescription mining with multi-target predictive clustering trees," in *Proceedings of the 4th International Workshop on the New Frontiers in Mining Complex Patterns (NFMCP'15)*, 2016, pp. 125–143.

[8] M. J. Zaki and C.-J. Hsiao, "Efficient algorithms for mining closed itemsets and their lattice structure," *IEEE Trans. Knowl. Data En.*, vol. 17, no. 4, pp. 462–478, 2005.

[9] A. Gallo, P. Miettinen, and H. Mannila, "Finding subgroups having several descriptions: Algorithms for redescription mining," in *Proceedings of the 8th SIAM International Conference on Data Mining (SDM'08)*, 2008, pp. 334–345.

[10] M. J. Zaki and N. Ramakrishnan, "Reasoning about sets using redescription mining," in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*, 2005, pp. 364–373.

[11] L. Parida and N. Ramakrishnan, "Redescription mining: Structure theory and algorithms," in *Proceedings of the 20th National Conference on Artificial Intelligence and the 7th Innovative Applications of Artificial Intelligence Conference (AAAI'05)*, 2005, pp. 837–844.

[12] E. Galbrun and P. Miettinen, "From black and white to full color: Extending redescription mining outside the Boolean world," *Stat. Anal. Data Min.*, vol. 5, no. 4, pp. 284–303, 2012.

[13] J. Kalofolias, E. Galbrun, and P. Miettinen, "From sets of good redescriptions to good sets of redescriptions," in *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM'16)*, 2016, pp. 211–220.

[14] ——, "From sets of good redescriptions to good sets of redescriptions," *Knowl. Inf. Syst.*, 2018, accepted.

[15] T. De Bie, "Maximum entropy models and subjective interestingness: an application to tiles in binary databases," *Data Min. Knowl. Discov.*, vol. 23, no. 3, pp. 407–446, 2011.

[16] E. Galbrun and P. Miettinen, "A case of visual and interactive data analysis: Geospatial redescription mining," in *Proceedings of the ECML PKDD 2012 Workshop on Instant and Interactive Data Mining (IID'12)*, 2012, Accessed 25 Oct 2017. [Online]. Available: http://adrem.ua.ac.be/iid2012/papers/galbrun_miettinen-visual_and_interactive_geospatial_redescription_mining.pdf

[17] ——, "Siren: An interactive tool for mining and visualizing geospatial redescriptions [demo]," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*, 2012, pp. 1544–1547.

[18] ——, "Interactive redescription mining," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD'14)*, 2014, pp. 1079–1082.

[19] M. Mihelčić and T. Šmuc, "InterSet: Interactive redescription set exploration," in *Proceedings of the 19th International Conference on Discovery Science (DS'16)*, vol. 9956, 2016, pp. 35–50.

[20] E. Galbrun, H. Tang, M. Fortelius, and I. Žliobaitė, "Computational biomes: The ecometrics of large mammal teeth," *Palaeontol. Electron.*, 2017, submitted.

[21] M. Mihelčić, G. Šimić, M. Babić-Leko, N. Lavrač, S. Džeroski, and T. Šmuc, "Using redescription mining to relate clinical and biological characteristics of cognitively impaired and Alzheimer's disease patients," *PLOS ONE*, vol. 12, no. 10, pp. 1–35, 2017.

[22] E. Galbrun and P. Miettinen, “Analysing political opinions using redescription mining,” in *IEEE International Conference on Data Mining Workshops*, 2016, pp. 422–427.

[23] S. Wrobel, “An algorithm for multi-relational discovery of subgroups,” in *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD’97)*, vol. 1263, 1997, pp. 78–87.

[24] P. Kröger and A. Zimek, “Subspace clustering techniques,” in *Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds. New York: Springer, 2009, pp. 2873–2875.

[25] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: A survey,” *IEEE Trans. Comput. Bio. Bioinform.*, vol. 1, no. 1, pp. 24–45, 2004.

[26] S. Bickel and T. Scheffer, “Multi-view clustering,” in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM’04)*, 2004, pp. 19–26.

[27] L. Umek, B. Zupan, M. Toplak, A. Morin, J.-H. Chauchat, G. Makovec, and D. Smrke, “Subgroup discovery in data sets with multi-dimensional responses: A method and a case study in traumatology,” in *Proceedings of the 12th Conference on Artificial Intelligence in Medicine (AIME’09)*, vol. 5651, 2009, pp. 265–274.

[28] P. Miettinen, “On finding joint subspace Boolean matrix factorizations,” in *SIAM International Conference on Data Mining (SDM’12)*, 2012, pp. 954–965.

[29] S. K. Gupta, D. Phung, B. Adams, and S. Venkatesh, “Regularized nonnegative shared subspace learning,” *Data Min. Knowl. Disc.*, vol. 26, no. 1, pp. 57–97, 2013.

[30] S. A. Khan and S. Kaski, “Bayesian multi-view tensor factorization,” in *Proceedings of the 2014 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD’14)*, 2014, pp. 656–671.

[31] E. Galbrun and A. Kimmig, “Finding relational redescriptions,” *Mach. Learn.*, vol. 96, no. 3, pp. 225–248, 2014.

Advances in Inference Methods for Markov Logic Networks

Deepak Venugopal

***Abstract*—Markov Logic Networks (MLNs) are expressive models that can be used to specify complex and uncertain background knowledge in the form of weighted first-order logic formulas. However, inference in MLNs is highly challenging since the underlying probabilistic model can be very large even for simple MLN structures. Lifted inference has emerged as dominant approach for probabilistic inference in MLNs, where the idea is to exploit symmetries in the MLN for scalable inference. In this paper, we provide an overview of MLNs, and major advances in inference techniques for MLNs over the last several years.**

***Index Terms*—Markov Logic Networks, Statistical Relational Learning, Probabilistic Graphical Models, Probabilistic Inference.**

I. Introduction

STATISTICAL Relational AI [1] unifies two corner-stones of Artificial Intelligence, namely, first-order logic and probabilities, to represent relational knowledge in the presence of uncertainty. Several notable SRL models have been proposed over the last several years including Markov Logic Networks (MLNs) [2], [3], Bayesian Logic (BLOG) [4], probabilistic soft logic (PSL) [5] and ProbLog [6]. MLNs are arguably one of the most popular models for SRL, and combine first-order logic with undirected probabilistic graphical models also known as Markov networks [7]. Specifically, an MLN is a set of first-order logic formulas with real-valued weights attached to each formula. The first-order formulas encode knowledge corresponding to an application domain, while the weights represent uncertainty associated with that knowledge. The larger the weight of a formula, greater is our belief in that formula, and vice-versa. Thus, MLNs soften the semantics of first-order logic (where formulas are either true/false). More specifically, MLNs are essentially template models that can encode different probability distributions based on the instantiations of its first-order formulas. Given the constants in a domain-of-interest, the probability distribution in an MLN is represented in factored form as a Markov network. Note that by combining the compactness of first-order logic and Markov networks, MLNs are capable of representing large, complex, uncertain relational knowledge in a succinct manner. Therefore, they have been used in diverse areas including NLP [8], computer vision [9], intelligent tutoring systems [10] and health informatics [11].

However, the expressiveness of MLNs comes at the cost of increased complexity of probabilistic inference, and consequently learning, which typically uses inference as a sub-step. Specifically, with just a few compact formulas, MLNs are able to represent extremely large Markov networks containing thousands of variables and factors. For instance, consider a simple MLN that models the transitive relationship Friends x, Friends $,z$ Friends z,x with weight . Every possible instantiation or *grounding* of the MLN formula for a given domain, represents a factor in the Markov network. That is, suppose we consider 1000 people in our domain, the Markov network underlying our example MLN has 1 billion factors and 1 million variables. Performing inference on MLNs is infeasible using traditional inference algorithms for graphical models, which we refer to as *propositional* methods, since they work on the Markov network representation of the first-order MLN. Thus, the challenge is to perform inference by taking advantage of the *lifted* representation in MLNs.

An interesting aspect about the MLN representation is that the number of weights in the MLN is typically much smaller than the number of factors in the underlying Markov network. In other words, all instantiations of a formula *share* the same weight. This induces symmetries in the probability distributions encoded by an MLN. Therefore, a significant amount of research in MLNs has aimed towards exploiting these symmetries to improve scalability. In particular, starting with the pioneering work by Poole [12], the predominant method for inference is the idea of *lifted inference*, which performs reasoning over groups of indistinguishable variables in the model. For example, if Friends $Alice, Bob$ and Friends $Bob, Carl$ have the same distributions, then inference results for Friends $Alice, Bob$ can be re-used for Friends $Bob, Carl$. The main challenge in developing efficient lifted inference algorithms is to efficiently compute groups of symmetric variables at a first-order level, without explicitly grounding the MLN, which could potentially create an extremely large Markov network.

The aim of this paper is to provide readers an overview of MLNs in general, and in particular, to summarize major advances in inference over the last few years. Fast and scalable inference algorithms are critical to the success of not only MLNs, but the general field of Statistical Relational AI. With a growing interest in Statistical Relational AI due to the expressiveness, and explainability of its models [13], we believe that developments in this area should be of interest to the intelligent systems community in general.

II. Background

A. First-order Logic

The language of first-order logic (cf. [14]) consists of quantifiers (and), logical variables, constants, predicates,

Deepak Venugopal is with the Department of Computer Science, University of Memphis, TN, 38152 USA e-mail: (dvngopal@memphis.edu).

and logical connectives (, , , , and). A predicate is a relation that takes a specific number of arguments as input and outputs either TRUE (synonymous with) or FALSE (synonymous with). The arity of a predicate is the number of its arguments. A first-order formula connects various predicates through the logical connectives. A first-order knowledge base (KB) is a set of first-order formulas. We denote logical variables in a KB by lower case letters (e.g., x, , z) and constants, which model objects in the real-world domain, by strings that begin with an uppercase letter (e.g., A, Ana, Bob).

B. Markov Logic Networks (MLNs)

One of the problems with first-order logic is that it cannot represent uncertainty, i.e., formulas are either true or false. MLNs soften the constraints expressed by each formula, by attaching a weight to it. Higher the weight, higher is our belief of the formula being satisfied, all other things being equal. In MLNs, we assume a restricted version of first-order logic with *Herbrand* semantics. Specifically, we assume that each argument of each predicate is typed and can only be assigned to a finite set of constants. By extension, each logical variable in each formula is also typed. Given a domain of constants, a *ground* atom is obtained by substituting all the variables in a predicate by constants. Similarly, a ground formula is obtained by replacing all variables in the formula with constants. A possible world, denoted by ω, is a truth assignment to all ground atoms in the MLN.

MLNs can also be seen as a first-order template for generating large Markov networks[15], [7], which is an undirected probabilistic graphical model. To illustrate MLNs, we consider the prototypical "friends-smokers" social network domain. We can represent common-sense knowledge that "smoking causes cancer" and "smokers tend to have similar smoking habits" using the following weighted formulas: (i) x Smokes x Cancer x ; and (ii) x, Smokes x Friends x, Smokes where and are the weights. Weights lie between and + and reflect the strength of the constraint. Positive (negative) weights represent that the worlds satisfying the formula have higher (lower) probability than worlds not satisfying the formula. MLNs generalize first-order logic in the sense that weights that are equal to infinity represent hard constraints.

Given a set of constants that represent objects in the domain (e.g. people in the social-network), the Markov network has one random variable for each grounding of each predicate (one for each instantiation of each logical variable in the predicate by a constant) and one feature for each possible grounding of each first-order formula. The weight attached to the feature is the weight attached to the corresponding first-order formula. For instance, given two constants Ana and Bob, the first first-order formula in the friends-smokers MLN yields the following two ground formulas having the same weight : (i) Smokes Ana Cancer Ana ; and (ii) Smokes Bob Cancer Bob . . Similarly, the second first-order formula with the same constants will yield four ground formulas. Formally, given a set of weighted first-order formulas f, and a set of constants, the probability of a world ω, which is a truth-assignment to all the ground atoms, is given by the following log-linear expression: $\omega = Z$ N ω where N ω is the number of groundings of f that are true in ω and Z is a normalization constant, also called the partition function.

Important inference queries in MLNs are computing the partition function, finding the marginal probability of an atom given evidence (an assignment to a subset of variables) and finding the most probable assignment to all atoms given evidence (MAP inference). All these problems are computationally intractable. Therefore, typically approximate inference algorithms are used to solve these problems in practical MLNs. In a typical use case of MLNs, the application designer writes first-order logic formulas that encode prior knowledge about the domain, and then relies on domain independent techniques implemented in software packages such as Alchemy [16] and Tuffy [17] to solve two key tasks: *probabilistic inference* – answering queries (making predictions) given the learned MLN and observations (evidence), and *weight learning* – learning the weights attached to the formulas from data. Weight learning internally uses inference within each sub-step, and therefore developing efficient inference methods is one of the key problems in MLNs.

III. EXACT LIFTED INFERENCE

Lifted inference in MLNs can be viewed as the probabilistic equivalent of reasoning in first-order logic, i.e., theorem proving. Specifically, just as theorem proving does not convert first-order formulas in a knowledge base to propositional formulas but instead reasons directly on the first-order representation, lifted inference aims to perform probabilistic reasoning without creating the full Markov network from the ground formulas. The concept of *domain liftable* MLNs was introduced in [18], [19], which refers to MLN structures on which the complexity of exact inference is polynomial in the number of domain objects. Notable lifted inference algorithms that peform domain-lifted exact inference, include lifted factor-graphs [12], First-order Variable Elimination (FOVE) [20], Weighted First-Order Model Counting (WFOMC) [21] and Probabilistic theorem Proving (PTP) [22]. Next, we will briefly review PTP which is one of the most popular exact lifted inference methods for MLNs.

PTP lifts *weighted model counting* [23] to the first-order. It turns out that the weighted model counting problem is equivalent to computing the partition function of the MLN (cf. [22], [23]). PTP computes the partition functions using two lifting rules, namely, lifted decomposition and lifted conditioning. Just like the well-known DPLL algorithm [24] for SAT solving, PTP recursively applies the lifting rules on the input MLN. Below, we give a informal summary of each lifting rule, and refer the reader to [22], [25] for details.

Lifted Decomposition identifies identical and independent components in the underlying Markov network by only looking at the first-order structure. We illustrate this with a simple example. Consider the MLN , Strong(x) Wins(x). Given the domain, $\Delta = X , X$, the Markov network defined over Strong(X) Wins(X) is identical and independent of the Markov network defined over Strong(X) Wins(X). Thus, $Z = Z$ $Z = Z$.

Lifted Conditioning conditions on a first-order predicate. Conditioning removes the predicate from the MLN by creating MLNs corresponding to each possible assignment to that predicate. Clearly, if a predicate R has d ground atoms, the total number of possible assignments to the ground atoms of R is equal to . However, in some cases, it is possible to group the set of assignments such that in any group, all the MLNs that are generated by the assignments in a group are equivalent to each other, i.e., they have the same partition function. For example, consider the MLN Smokes x Friends x, Asthma x . Here, conditioning on Smokes x implicitly conditions on $^{\Delta}$ different assignments to the groundings of Smokes x . However, it turns out that, in this case, we can form Δ $+$ groups, where any assignment within a group yields the same partition function after conditioning. The grouping can be performed based on the number of atoms among the Δ that are set to true in an assignment. We can then choose one representative from each group, condition on it, and multiply the partition function of the conditioned MLN by group-size. Note that, in general, this rule can only be applied to singleton atoms, namely, atoms whose arity is equal to 1.

In practice, apart from the two lifting rules, PTP leverages advanced SAT techniques such as unit propagation and caching to greatly improve the performance of lifted inference. For more details on these extensions, refer to [22].

IV. Approximate Lifted Inference

Exact lifted inference is highly scalable when the MLN structure has symmetries that can be exploited by algorithms such as PTP. However, as shown in [18], [19], a very restrictive set of MLNs exhibit such symmetries. Specifically, according to current complexity results, MLNs are liftable only if each formula has a maximum of two variables. Therefore, for most practical MLNs, exact inference is unlikely to scale up. Thus, several well-known propositional approximate inference algorithms have been lifted to the first-order level. Next, we will review a few notable ones.

Lifted Belief Propagation. Singla and Domingos [26] lifted belief propagation[27] in MLNs to the first-order level. Specifically, in loopy belief propagation, the MLN is encoded as a factor-graph, where the atoms are the variables, and the ground formulas are the factors. The sum-product algorithm computes the marginals of all variables in the factor graphs by passing messages between the nodes/variables and features/factors that relate the nodes. The message from nodes to features is a product of all incoming messages from other features that the node is connected to, with the variable corresponding to the summed-out from the product. Similarly, the message from a feature to a node is a product of all the messages coming into a feature from nodes connected to the feature. In Lifted Belief Propagation, the main idea is to identify messages that are identical, and send a single aggregate message instead of individual messages. To do this, Singla and Domingos proposed the creation of super-nodes and super-features, which correspond to groups of nodes and features that emit common messages. The grouping of nodes into super-nodes and features into super-features is performed incrementally by observing the messages in BP that are identical to each other.

Lifted Sampling-based Inference. In sampling-based inference methods we draw samples from the target distribution, and compute inference queries as statistics on the drawn samples. Note that in the case of MLNs, sampling worlds directly from the MLN distribution is hard, since the partition function is intractable to compute. In IS, we perform approximate inference by sampling worlds from an easier-to-sample *proposal distribution*. However, to compensate for the fact that we sampled from the approximate distribution, we weigh each sample, and compute statistics over the weighted samples. The quality of estimates from IS depends upon how close the proposal distribution is to the true distribution. Gogate et al. [28] proposed a lifted Importance Sampling (LIS) algorithm, where the main idea is to exploit symmetries to create a more-informed proposal distribution. Specifically, they grouped together symmetric worlds, and sampled a single world from each group, which consequently increases the effective sample-size, and yields lower-variance estimates of the computed inference queries. In order to create a proposal distribution which is tractable to sample from, Gogate et al. relied on lifting rules of PTP [22]. Specifically, in PTP, the lifting rules are applicable only for specific MLN structures. In LIS, the pre-conditions for applying the lifting rules are relaxed, and thus, the lifting rules are applied approximately to non-liftable MLN structures. Samples from the proposal distribution are generated by sampling from a symbolic execution trace of PTP. Further, the proposal distribution is adaptively improved based on prior samples such that the distribution moves closer to the true MLN distribution.

An alternative approach to IS, is to construct a Markov Chain whose stationary distribution is equivalent to the MLN's true distribution. We can then sample from this chain, and answer inference queries based on statistics obtained from the samples. A lifted MCMC method can be visualized as one that works in a *lifted state-space*. That is, we construct a state-space of the sampler that does not explicitly enumerate every possible state but instead groups these states based on symmetries. For instance, a propositional sampler on a MLN with n atoms works in a state-space states, while a lifted state-space can have far fewer number of states. A Markov chain defined on the lifted states can typically make larger jumps as compared to Markov chains defined on a propositional space, and in many cases larger jumps can avoid being struck in regions of local optima. Niepert [29] proposed a lifted MCMC method by grouping together states based on symmetries detected from automorphism groups computed from the MLN's graph structure. Venugopal et al. [30] lifted the blocked Gibbs sampling algorithm [31], which is an advanced variant of the Gibbs sampling [32] algorithm, which is arguably one of the most popular MCMC methods. Lifted Blocked Gibbs (LBG) partitions the atoms in the MLN into domain-liftable blocks, i.e., exact lifted inference must be tractable within each block. Further, the LBG sampler maintains a lifted state-space within each block, where the assignments to all ground atoms within a block are not stored, but sufficient statistics related to these assignments are stored.

This improves the convergence of the sampler, as well as the estimates derived from the sampler.

Lifted MAP Inference. MAP inference is an optimization problem that can be solved using popular randomized local-search solvers such as MaxWalkSAT [33]. These techniques are propositional since they work on the Markov network underlying the MLN. Sarkhel et al. [34] proposed an approach to lift such propositional MAP solvers by pre-processing the MLN, and reducing the size of its underlying Markov network. Specifically, they considered a specific subset of MLNs called *non-shared* MLNs, where no variables are shared across atoms in a formula, and showed that the MAP solution in these MLNs is independent of the number of domain objects. For example, the MLN R x S is equivalent to R S, where, if the assignment to R (or S) in the MAP solution is 1 (or 0), then all ground atoms of R x (or S x) have an assignment equal to 1 (or 0). Using this property, MAP inference on non-shared MLNs can be reduced to propositional MAP inference, where each first-order predicate is replaced by a single propositional variable, since the MAP assignment to all groundings of the predicate are symmetric to each other. Other approaches for MAP inference have lifted Linear Programming solvers based on symmetries [35].

V. Exploiting Approximate Symmetries

One of the key problems with lifted inference methods that exploit exact symmetries is that they are ineffective when the structure of the MLN is complex (e.g. transitive formula) or evidence is presented to the MLN, since evidence typically breaks symmetries. For example, consider the toy MLN given in Figure 1(a). When no evidence is present, all ground atoms of Wins x, have the same marginal probability (exact symmetry). Given evidence (see Figure 1(b)), the marginal probabilities are no longer the same as shown in Figure 1(c) and there are no exact symmetries. In such cases, lifted inference algorithms will ground the MLN, and lifted inference is almost the same as propositional inference. More generally, Broeck and Darwiche [36] showed theoretical results that, in the presence of evidence on binary or higher-arity atoms, MLNs are no longer domain-liftable. Similarly, Kersting et al. [37] showed that as the amount of *symmetry-breaking* evidence increases, the benefits of lifted inference diminishes. Unfortunately, most real-world applications require inference algorithms that can reason in the presence of evidence, and in such cases lifted inference is more or less equivalent to propositional inference. This problem can be averted using approximations which group together variables that are similar, but not identical. For example, from Figure 1(c), we can see that the marginal probabilities of the first three atoms and the last two atoms are roughly the same and they can be treated as indistinguishable for all practical purposes. Below, we highlight some specific approaches that are designed to scale up inference by exploiting approximate symmetries when exact symmetries are absent.

Over-Symmetric Approximation. Broeck and Darwiche [36] proposed the idea of *smoothing* the evidence by introducing more symmetries in the model. In this way, lifted inference methods will have a better chance of finding these symmetries. For example, if we consider the evidences on a predicate Linked, as Linked P , P , Linked P , P , Linked P , P , Linked P , P , the evidence on P and P is not symmetrical. however, by removing Linked P , P and adding Linked P , P , the evidence becomes more symmetrical. Broeck and Darwiche modeled this as a factorization problem in a boolean matrix. Specifically, binary evidence is represented as a boolean matrix, and the idea is to come up with a reduced-rank approximation of this matrix, which in-turn yields more symmetric evidence that is better suited for lifted inference. However, changing the evidence would change the MLN distribution, and therefore, inference results computed from this approximate distribution will not have strong guarantees associated with it. To obtain such guarantees, Niepert and Broeck [38] used the over-symmetric approximation to as a proposal distribution for MCMC algorithms instead of computing inference results directly from the approximate distribution.

Evidence-based Clustering. The key idea here is to preprocess the MLN by reducing the number of objects, replacing several roughly symmetric objects by a single (meta) object. We then run lifted inference using these new, much smaller set of objects. A key challenge is how to find objects that are similar to each other and thus partitioning the set of objects into symmetric subsets. To solve this problem, we defined a distance (similarity) function that takes two objects as input and outputs a number that describes how symmetric or similar the two objects are (smaller the number, greater the chance that the two objects are similar). The problem is now reduced to a standard clustering problem, and algorithms such as K-means can be used to solve it. Venugopal and Gogate [39] proposed a distance function which is based on common sense knowledge that objects having similar neighborhood constraints (Markov blanket) tend to be symmetric in the sense that the marginal probabilities of atoms containing those objects will be roughly the same. Formally, the distance function developed by Venugopal and Gogate is given by: $d\ X, X\ =\ U\ \ U$ where X and X are constants (objects) that belong to the same domain equivalence class (see section 2) and $U\ =\ c\ ,\ldots,c$ and $U\ =\ c\ ,\ldots,c$ are m-dimensional vectors where m is the number of formulas and c is the number of groundings of the formula f that evaluate to true in the MLN obtained from the original MLN by grounding all logical variables having the corresponding object type with X and instantiating evidence. One can think of U as a feature vector describing the neighborhood of the object X in the MLN given evidence. Since computing the number of groundings is a #P-hard problem, the approach by Venugopal and Gogate proposed to approximate the counts by decomposing large formulas into smaller ones. However, one of the major problems with using clustering methods such as K-Means is that the optimal number of clusters is hard to compute. For instance, for some domains with greater symmetry among objects, a small set of meta-objects may suffice, while for other domains, we may require more meta-objects. Venugopal et al. [40] extended the aforementioned approach using a non-parametric clustering method called DP-Means [41], where they computed the optimal number of

Wins(A,A)	0.56
Wins(A,B)	0.56
Wins(A,C)	0.56
Wins(B,A)	0.56
Wins(B,B)	0.56
Wins(B,C)	0.56
Wins(C,A)	0.56
Wins(C,B)	0.56
Wins(C,C)	0.56

(a) Original Marginals

Strong(C)
Wins(A,C)
Wins(B,B)
Wins(B,C)
Wins(C,A)

(b) Evidence

Wins(A,A)	0.6
Wins(A,B)	0.6
Wins(B,A)	0.63
Wins(C,B)	0.85
Wins(C,C)	0.85

(c) New Marginals

Fig. 1. Effect of evidence on an MLN with one formula, Strong( ) Wins(,). The marginal probabilities which were equal in (a) become unequal in (c) due to evidence (b).

clusters for a given bound on the error in samples generated from the approximated MLN.

Apart from the above approaches, other methods have also been proposed for exploiting approximate symmetries in specific inference tasks. Specifically, Sarkhel et al. [42] proposed a refinement approach for MAP inference by adding equality constraints to the MLN when objects are approximately symmetric to each other. For the marginal inference problem, Singla et al. [43] considered approximately similar messages in belief propagation as equivalent messages, and constructed a lifted belief network that is much smaller than the lifted network constructed from exactly symmetric messages.

VI. Exploiting MLN Structure

Since MLNs are deined as logical formulas, propositional inference algorithms can use approaches that exploit MLN structure to avoid constructing the ground Markov network during inference. Along these lines, Shavlik and Natarajan [44] proposed an approach called FROG (Fast Reduction Of Grounded networks). The idea was to pre-process the MLN, and reduce the size of the ground network by efficiently computing formulas that are satisfied due to the logical structure. For example, in R x, S , z T z, x , if we know that majority of the groundings of R x, are false, then majority of the formula groundings are true irrespective of the states of the groundings of S , z and T , z . FROG maintains a representation of these non-satisfied formula groundings, and just stores statistics on the satisfied groundings, which is sufficient for performing inference. Similarly, Poon and Domingos [45] developed an approach to perform *lazy grounding* in MLNs. Here, only a subset of variables and ground formulas are kept in memory, and as inference proceeds, grounding is performed as-needed. The main idea behind this approach is that, many propositional inference algorithms work on one variable of the MLN at a time, and updates to the variable depends upon a small subset of other variables and formulas. More recently, Venugopal et al. [46] proposed an approach to scale up local-search algorithms such as MaxWalkSAT and sampling-based inference algorithms such as Gibbs sampling using efficient counting algorithms. Specifically, a counting problem that MaxWalkSAT or Gibbs Sampling needs to solve is, counting the satisfied groundings of a first-order formula f, given a world ω. This problem is known to be computationally hard [3], and inference algorithms need to solve this problem not just once but several times over thousands of iterations. Venugopal et al. encoded the counting problem as a problem of computing the partition function of a graphical model. Specifically, given that f has k variables, they encoded a graphical model with k nodes, and derived the factors in the graphical model from ω. Importantly, if the *tree-width* of the encoded graphical model is small, then the counting can be performed exactly using methods such as junction-trees [47]. For larger treewidth models, off-the-shelf algorithms such as generalized belief propagation [27] to approximate the counts in a scalable manner.

VII. Joint Inference Applications

MLNs have been used extensively to model joint inference tasks in complex problems. As compared to Integer Linear Prgramming (ILP) formulations which were previously used to model joint inference [48], the first-order structure of MLNs helps us model joint dependencies more compactly. Singla and Domingos [49] developed one of the earliest MLN-based joint inference models for the entity resolution task on the cora and bibserv citation datasets. Poon and Domingos [8] developed an MLN model for information extraction utilizing entity resolution within the model, to jointly segment citation fields in the cora and bibserv datasets. Poon and Domingos [50] also developed an unsupervised model for joint coreference resolution, where they used a predicate to specify clustering of mentions that correspond to a common entity. Poon and Vanderwende [51] developed a joint model for Biomedical event extraction on the Genia dataset, where they detected triggers and arguments jointly. Venugopal et al. [52] further improved the performance of MLNs on the same event extraction task by leveraging linguistic features. Specifically, encoding such features directly as MLN formulas makes learning and inference infeasible due to their high-dimensionality, which results in a large number of ground formulas. To add such features to the MLN, Venugopal et al. used the output of SVMs learned from the high-dimensional features to generate priors for the MLN distribution. Lu et al. [53] designed an MLN for event coreference resolution that jointly performs four tasks in the pipeline: trigger identification and subtyping, argument identification and role determination, entity coreference resolution and event coreference resolution. Similar to Venugopal et al.'s strategy, Lu et al. used an SVM to incorporate high-dimensional features into their MLN.

VIII. Discussion and Conclusion

Though plenty of progress has been made in MLNs over the last several years, there is quite a lot of work that needs to be done to make MLNs a "black-box" for application designers. The blow-up in the size of the probabilistic model, with increased data-size makes inference and learning very hard, and therefore application designers find it hard to use MLNs using existing open-source systems MLNs as-is. Particularly, if MLNs are to work with *big-data problems*, inference algorithms need to become far more scalable than current state-of-the-art. Further, explaining the results of inference is an area that requires active future research in MLNs. Due to their basis in first-order logic, MLNs seem to be a promising candidate to develop interpretable Machine learning models. Integrating MLNs with other models such as deep-models in order to facilitate relational deep learning is also an area where future research seems to be headed.

In this paper, we provided a brief overview of MLNs and major advances in MLN inference methods. Particularly, the idea of lifted inference has received a lot of interest from the research community, where symmetries (both exact and approximate) in the MLN distribution are exploited to perform scalable inference. As MLNs continue to be applied in varied domains, advances in this area should be of interest to the field of Statistical Relational AI, and the Intelligent Systems community in general.

References

[1] L. Getoor and B. Taskar, Eds., *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[2] M. Richardson and P. Domingos, "Markov Logic Networks," *Machine Learning*, vol. 62, pp. 107–136, 2006.

[3] P. Domingos and D. Lowd, *Markov Logic: An Interface Layer for Artificial Intelligence*. San Rafael, CA: Morgan & Claypool, 2009.

[4] B. Milch, B. Marthi, and S. Russell, "BLOG: Relational Modeling with Unknown Objects," in *Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields*. Banff, Canada: IMLS, 2004, pp. 67–73.

[5] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss markov random fields and probabilistic soft logic," *Journal of Machine Learning Research*, vol. 18, pp. 109:1–109:67, 2017.

[6] L. De Raedt, A. Kimmig, and H. Toivonen, "ProbLog: A Probabilistic Prolog and Its Application in Link Discovery," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 2462–2467.

[7] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[8] H. Poon and P. Domingos, "Joint Inference in Information Extraction," in *Proceedings of the 22nd National Conference on Artificial Intelligence*. Vancouver, Canada: AAAI Press, 2007, pp. 913–918.

[9] S. D. Tran and L. S. Davis, "Event modeling and recognition using markov logic networks," in *ECCV*, 2008.

[10] D. Venugopal and V. Rus, "Joint inference for mode identification in tutorial dialogues," in *COLING 2016, 26th International Conference on Computational Linguistics*, 2016, pp. 2000–2011.

[11] S. Natarajan, V. Bangera, T. Khot, J. Picado, A. Wazalwar, V. S. Costa, D. Page, and M. Caldwell, "Markov logic networks for adverse drug event extraction from text," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 435–457, 2017.

[12] D. Poole, "First-Order Probabilistic Inference," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. Acapulco, Mexico: Morgan Kaufmann, 2003, pp. 985–991.

[13] L. D. Raedt, K. Kersting, S. Natarajan, and D. Poole, *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2016.

[14] M. R. Genesereth and E. Kao, *Introduction to Logic, Second Edition*. Morgan & Claypool Publishers, 2013.

[15] A. Darwiche, *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[16] S. Kok, M. Sumner, M. Richardson, P. Singla, H. Poon, D. Lowd, J. Wang, and P. Domingos, "The Alchemy System for Statistical Relational AI," Department of Computer Science and Engineering, University of Washington, Seattle, WA, Tech. Rep., 2008, http://alchemy.cs.washington.edu.

[17] F. Niu, C. Ré, A. Doan, and J. W. Shavlik, "Tuffy: Scaling up statistical inference in markov logic networks using an rdbms," *PVLDB*, vol. 4, no. 6, pp. 373–384, 2011.

[18] G. V. den Broeck, "On the completeness of first-order knowledge compilation for lifted probabilistic inference," in *NIPS*, 2011, pp. 1386–1394.

[19] M. Jaeger, "Lower Complexity Bounds for Lifted Inference," *CoRR*, vol. abs/1204.3255, 2012.

[20] R. de Salvo Braz, "Lifted First-Order Probabilistic Inference," Ph.D. dissertation, University of Illinois, Urbana-Champaign, IL, 2007.

[21] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, and L. De Raedt, "Lifted Probabilistic Inference by First-Order Knowledge Compilation," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011, pp. 2178–2185.

[22] V. Gogate and P. Domingos, "Probabilistic Theorem Proving," in *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2011, pp. 256–265.

[23] M. Chavira and A. Darwiche, "On probabilistic inference by weighted model counting," *Artificial Intelligence*, vol. 172, no. 6-7, pp. 772–799, 2008.

[24] M. Davis and H. Putnam, "A Computing Procedure for Quantification Theory," *Journal of the Association of Computing Machinery*, vol. 7, no. 3, 1960.

[25] A. Jha, V. Gogate, A. Meliou, and D. Suciu, "Lifted Inference from the Other Side: The tractable Features," in *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS)*, 2010, pp. 973–981.

[26] P. Singla and P. Domingos, "Lifted First-Order Belief Propagation," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*. Chicago, IL: AAAI Press, 2008, pp. 1094–1099.

[27] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized Belief Propagation," in *Advances in Neural Information Processing Systems 13*, T. Leen, T. Dietterich, and V. Tresp, Eds. Cambridge, MA: MIT Press, 2001, pp. 689–695.

[28] V. Gogate, A. Jha, and D. Venugopal, "Advances in Lifted Importance Sampling," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[29] M. Niepert, "Markov chains on orbits of permutation groups." in *UAI*. AUAI Press, 2012, pp. 624–633.

[30] D. Venugopal and V. Gogate, "On lifting the gibbs sampling algorithm," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 2012, pp. 1664–1672.

[31] C. S. Jensen, U. Kjaerulff, and A. Kong, "Blocking Gibbs Sampling in Very Large Probabilistic Expert Systems," *International Journal of Human Computer Studies. Special Issue on Real-World Applications of Uncertain Reasoning*, vol. 42, pp. 647–666, 1993.

[32] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[33] B. Selman, H. Kautz, and B. Cohen, "Local Search Strategies for Satisfiability Testing," in *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, D. S. Johnson and M. A. Trick, Eds. Washington, DC: American Mathematical Society, 1996, pp. 521–532.

[34] S. Sarkhel, D. Venugopal, P. Singla, and V. Gogate, "Lifted MAP inference for Markov Logic Networks," *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS-14)*, 2014.

[35] U. Apsel, K. Kersting, and M. Mladenov, “Lifting relational map-lps using cluster signatures,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 2403–2409.

[36] G. van den Broeck and A. Darwiche, “On the complexity and approximation of binary evidence in lifted inference,” in *Advances in Neural Information Processing Systems 26*, 2013, pp. 2868–2876.

[37] K. Kersting, B. Ahmadi, and S. Natarajan, “Counting Belief Propagation,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 277–284.

[38] G. Van den Broeck and M. Niepert, “Lifted probabilistic inference for asymmetric graphical models,” in *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, 2015.

[39] D. Venugopal and V. Gogate, “Evidence-based clustering for scalable inference in markov logic,” in *ECML PKDD*, 2014.

[40] D. Venugopal, S. Sarkhel, and K. Cherry, “Non-parametric domain approximation for scalable gibbs sampling in mlns,” in *Proceedings of the 32nd Conference on Uncertainty In Artificial Intelligence*, 2016, pp. 745–755.

[41] B. Kulis and M. I. Jordan, “Revisiting k-means: New algorithms via bayesian nonparametrics,” *ICML*, 2012.

[42] S. Sarkhel, D. Venugopal, T. A. Pham, P. Singla, and V. Gogate, “Scalable training of markov logic networks using approximate counting,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2016, pp. 1067–1073.

[43] P. Singla, A. Nath, and P. Domingos, “Approximate Lifting Techniques for Belief Propagation,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 2497–2504.

[44] J. W. Shavlik and S. Natarajan, “Speeding up inference in markov logic networks by preprocessing to reduce the size of the resulting grounded network,” in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009, pp. 1951–1956.

[45] H. Poon, P. Domingos, and M. Sumner, “A General Method for Reducing the Complexity of Relational Inference and its Application to MCMC,” in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008, pp. 1075–1080.

[46] D. Venugopal, S. Sarkhel, and V. Gogate, “Just count the satisfied groundings: Scalable local-search and sampling based inference in mlns,” in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[47] S. L. Lauritzen and D. J. Spiegelhalter, “Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 50, no. 2, pp. 157–224, 1988.

[48] D. Roth and W. Yih, “Integer Linear Programming Inference for Conditional Random Fields,” in *Proceedings of the Twenty-Second International Conference on Machine Learning*, 2005, pp. 736–743.

[49] P. Singla and P. Domingos, “Memory-Efficient Inference in Relational Domains,” in *Proceedings of the Twenty-First National Conference on Artificial Intelligence*. Boston, MA: AAAI Press, 2006, this volume.

[50] H. Poon and P. Domingos, “Joint Unsupervised Coreference Resolution with Markov Logic,” in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, HI: ACL, 2008, pp. 649–658.

[51] H. Poon and L. Vanderwende, “Joint inference for knowledge extraction from biomedical literature.” in *HLT-NAACL*, 2010.

[52] D. Venugopal, C. Chen, V. Gogate, and V. Ng, “Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2014, pp. 831–843.

[53] J. Lu, D. Venugopal, V. Gogate, and V. Ng, “Joint inference for event coreference resolution,” in *COLING 2016, 26th International Conference on Computational Linguistics*, 2016, pp. 3264–3275.

Strategic Voting

Reshef Meir

I. Introduction

In a typical voting scenario, a group of voters with diverse preferences need to collectively choose one out of several alternatives. Examples include a committee that selects a candidate for a faculty position or an award, countries in an international forum voting on the adoption of a new environmental treaty, or even automated agents that vote on the preferred meeting time on behalf of their users.

As the satisfaction of each voter is determined by the selected alternative, which is in turn affected by the actions (namely, the ballots) of others, casting a vote is in fact playing a strategic game.

The study of strategic voting is an effort to utilize *game theory*, which merits to model and predict rational behavior in a wide range of economic and social interactions, to explain and even direct the strategic decisions of voters.

This review paper is a hyper-condensed version of a book on strategic voting that is forthcoming this year. [1] The main purpose of the book is to overview the main approaches to strategic voting, in a way that makes these approaches comparable across fields and disciplines. In this paper I will mention the main directions and lines of work, but almost without going into the technical details.

Our starting point will be the seminal Gibbard-Satterthwaite theorem, which states that under a set of natural requirements, one cannot hope to construct a voting rule that is immune to strategic manipulations by the voters. This mean that there will always be situations where some voters have an incentive to misreport their true preferences. From this strong negative result emerged two lines of research. One continues to shape the boundaries and limitations of truthful voting mechanisms, by relaxing some of the assumptions that lead to the G-S impossibility result. The other line forgoes the attempt to elicit truthful votes, and instead applies game theory and equilibrium analysis to understand how strategic voters would vote in existing mechanisms.

II. Basic Notations

We denote sets by upper case letters (e.g., $A = \{a_1, a_2, \ldots\}$) and vectors by bold letters (e.g., $\boldsymbol{a} = (a_1, a_2, \ldots)$).

For a finite set X, we denote by $\mathcal{L}(X)$ the set of all linear (strict) orders over X.

a) Social choice: A voting scenario is defined by a set of *candidates*, or *alternatives*, A, a set of voters N, and a *preference profile* $\boldsymbol{L} = (L_1, \ldots, L_n)$, where each $L_i \in \mathcal{L}(A)$. For $a, b \in A, i \in N$, candidate a precedes b in L_i (denoted $a \succ_i b$) if voter i prefers candidate a over candidate b. We can also think about more general preferences, such as cardinal utilities that we denote by $U_i : A \to \mathbb{R}$.

Definition 1 (Social choice correspondence). *A social choice correspondence (SCC) is a function $F : \mathcal{L}(A)^n \to 2^A \setminus \emptyset$.*

Definition 2 (Social choice function). *An SCC F is* resolute *if $|F(\boldsymbol{L})| = 1$ for all $\boldsymbol{L}$. Resolute SCCs are also called* Social choice functions *(SCF). We typically denote SCFs by a lower case letter f.*

We will reserve the term *voting rule* for a SCF (i.e., a rule with a single winner) unless stated otherwise.

Some common voting rules that are mentioned in the paper are based on computing some score $s(c, \boldsymbol{L})$ for every candidate $c \in A$, and selecting the candidate with the highest score (employing some tie breaking policy if needed). For example, in *Plurality* $s(c, \boldsymbol{L})$ is the number of voters who ranked c in the first place. More generally, a *positional scoring rule* (PSR) sets $s(c, \boldsymbol{L}) = \sum_{i \in N} \alpha_{L_i^{-1}(c)}$ where α is some non-decreasing vector. The *Borda* rule is an example of a PSR where $\alpha = (m-1, m-2, \ldots, 2, 1)$.

b) Game theory:

Definition 3 (Game). *A (finite, n-person, non-cooperative)* game *is a tuple $\langle N, \mathcal{A}, \boldsymbol{u} \rangle$, where:*

- *N is a finite set of n players, indexed by i;*
- *$\mathcal{A} = A_1 \times \cdots \times A_n$, where A_i is a finite set of actions available to player i. Each vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathcal{A}$ is called an* action profile*;*
- *$\boldsymbol{u} = (u_1, \ldots, u_n)$ where $u_i : \mathcal{A} \to \mathbb{R}$ is a real-valued utility (or payoff) function for player i.*

A *game form* is similar to a game, except the utilities remain unspecified. Rather, for each combination of actions, we have an abstract "outcome" from some set A. Any game form $g : \mathcal{A} \to A$ together with cardinal utility functions for each player i, induce a unique game denoted by $\langle g, \boldsymbol{U} \rangle$. This is simply the normal form game $\langle N, \mathcal{A}, \boldsymbol{u} \rangle$, where $u_i(\boldsymbol{a}) \triangleq U_i(g(\boldsymbol{a}))$ for all $i \in N$ and $\boldsymbol{a} \in \mathcal{A}$. We can similarly combine g with an ordinal preference profile $\boldsymbol{L}$ to get an ordinal game.

For example, consider the game form on Fig. 1 (right). The set of players is $N = \{1, 2\}$, where 1 selects a row and 2 selects a column; $A_1 = A_2 = \{C, D\}$ (which stand for the actions "Cooperate" and "Defect"). The game on the left (the famous *prisoner's dilemma*) is obtained by setting a cardinal utility of $U_1(a) = 3, U_2(a) = 3, U_1(b) = 0$ and so on.

A (pure) *Nash equilibrium* (NE) in game $\langle N, \mathcal{A}, u \rangle$ is an action profile $\boldsymbol{a} \in \mathcal{A}$ such that $\forall i \in N \forall a'_i \in A_i$, $u_i(\boldsymbol{a}) \geq u_i(\boldsymbol{a}_{-i}, a'_i)$. That is, every player weakly preferences her current action over any other action *assuming others do*

Reshef Meir is with the faculty of Industrial Engineering and Management, Technion-Israel Institute of Technology, e-mail: (reshefm@ie.technion.ac.il).

[1] *Stategic Voting*, Reshef Meir, Synthesis Lectures on AI and Machine Learning, Morgan-Claypool, forthcoming.

$$\begin{array}{c|c|c|} & C & D \\ \hline C & 3,3 & 0,4 \\ \hline D & 4,0 & 1,1 \\ \hline \end{array} \qquad \begin{array}{c|c|c|} & C & D \\ \hline C & a & b \\ \hline D & c & d \\ \hline \end{array}$$

Figure 1. On the left - one variation of the prisoner's dilemma game. On the right, a 2×2 game form.

not change their own action. For example, the profile (D, D) is the unique NE in the prisoner's dilemma above.

A profile $\boldsymbol{a}$ is *Pareto efficient* if there is no other profile that is weakly better for all agents, and strictly better for at least one. For example, in the prisoner's dilemma, all profiles *excepts* (D, D) are Pareto efficient.

c) Game Forms are Voting Rules: Note that by taking any voting rule (=game form) f, and add a specific preference profile $\boldsymbol{L}$, we get the ordinal game $\langle f, \boldsymbol{L}\rangle$ as explained above, and we can go ahead and analyze its equilibria. However, for most common voting rules, this approach is not very informative. Consider Plurality voting with $n \geq 3$ voters. It is easy to see that any profile in which all voters vote for the same candidate is a Nash equilibrium. This is true even if this candidate is ranked last by all voters in $\boldsymbol{L}$, since no single voter can change the outcome. In Section VI we return to the notion of equilibrium in voting, and consider refinements and variations that are more reasonable and more useful as a solution concept.

III. Strategyproofness and the Gibbard-Satterthwaite Theorem

d) Example of a manipulation: Consider an election using the Borda voting rule with the following preference profile $\boldsymbol{L}$:

L_1	L_2	L_3
b	b	a
a	a	b
c	c	c
d	d	d

Candidate b is the winner, beating candidate a 8 points to 7. However, if voter 3 lies about his preferences and ranks candidate b last (after a, c and d), b's score goes down to 6, and a (voter 3's favorite candidate) wins! This is called a *manipulation*.

A natural question is whether there are voting rules where such manipulations are impossible. That is, where a voter can never gain from lying about her preferences.

Definition 4 (Strategyproofness)**.** *A voting rule f is* strategyproof *if no single voter can ever benefit from lying about her preferences:*

$$\forall \boldsymbol{L} \in \mathcal{L}(A)^n \;\; \forall i \in N \;\; \forall L'_i \in \mathcal{L}(A) \\ f(L'_i, \boldsymbol{L}_{-i}) \preceq_i f(\boldsymbol{L}).$$

For example, Plurality is strategyproof when $|A| = 2$.

A voting rule f is *dictatorial* if there is an individual (the dictator) such that i's most preferred candidate is always chosen:

$$\exists i \in N \quad \forall \boldsymbol{L} \in \mathcal{L}(A)^n : f(\boldsymbol{L}) = \text{top}(L_i).$$

A voting rule f is a *duple* if there are only two possible winners.

Theorem 1 (Gibbard [22],Satterthwaite [61])**.** *A deterministic and onto voting rule is strategyproof if and only if it is dictatorial or a duple.*

It is easy to see that a dictatorial rule is SP, since the dictator is always best off reporting the truth and all other voters have no power; if we allow duples, we can arbitrarily select two candidates a, b and hold a majority vote between them. There are many different proofs of the Gibbard-Satterthwaite (G-S) theorem. Several simple proofs can be found in [66].

The G-S theorem is considered as a strong negative result: Both dictatorial rules and duples have significant shortcomings as voting rules. A dictatorship ignores the will of all voters but one, and a duple may fail to select a candidate even if there is a unanimous agreement among voters that it is best.

e) Extensions: The negative result implied by the theorem is quite robust. Several recent papers show that the number of different profiles in which there is a manipulation is relatively large (a polynomial fraction of all profiles), unless the voting rule is very close to being dictatorial [21], [42].

When also considering manipulations by *coalitions* the situation becomes even worse. For a wide class of voting rules known as "generalized scoring rules," and which contains most common voting rules, Xia and Conitzer [71] showed that "large coalitions" (with substantially more than $\sqrt{n}$ voters) can decide the identity of the winner in almost every profile. These results were later extended by Mossel et al. [41].

Another result demonstrating the robustness of the G-S theorem is by Slinko and White [65], who showed that even if we restrict manipulations by voters to be "safe" (informally, such that any number of like-minded followers will not hurt the manipulator), this does not expand the set of strategyproof voting rules.

IV. Regaining Truthfulness in Voting

We will focus on four approaches, each of which attains truthfulness by relaxing some assumption underlying the G-S theorem. Other approaches that involve monetary payments are discussed in Section V.

A. Domain restriction

Suppose voters are voting on where to place a public library along a street. Naturally, each voter prefers the library to be located as close as possible to her house (whose location is private information). Note that not every preference profile is possible under this assumption.

More formally, a preference profile $\boldsymbol{L}$ is *single peaked* w.r.t. a linear order $\mathcal{O}$ over A, if each voter has some "peak candidate" a^*_i s.t. if x is *between* a^*_i and y then i prefers x over i. See Fig. 2 for an example.

Consider a linear order $\mathcal{O}$ over alternatives A, and a preference profile $\boldsymbol{L}$ that is single-peaked on $\mathcal{O}$. The *Median* voting rule considers the peak locations of all voters, and return their median as the winner. Consider the example in Fig. 2. The median of the five numbers $\{l_1, l_2, l_3, l_4, l_5\} = \{1, 2, 4, 5, 4\}$

L_1	$A \succ E \succ C \succ D \succ B$
L_2	$E \succ A \succ C \succ D \succ B$
L_3	$D \succ B \succ C \succ E \succ A$
L_4	$B \succ D \succ C \succ E \succ A$
L_5	$D \succ C \succ E \succ B \succ A$
L_6	$D \succ C \succ B \succ A \succ E$

			L_5	
L_1	L_2		L_3	L_4
A	E	C	D	B
1	2	3	4	5

Figure 2. The preferences of the first five voters are single peaked w.r.t. the order $\mathcal{O} = A \succ E \succ C \succ D \succ B$. The right figure shows the position of each of the first five voters w.r.t. the order $\mathcal{O}$. For example, $l_4 = \mathcal{O}(B) = 5$ The sixth voter L_6 is not single peaked w.r.t. $\mathcal{O}$.

is 4. Thus either voter 3 or voter 5 can be the median voter. In either case, the outcome is $\text{top}(L_3) = \text{top}(L_5) = D$, which is the *median candidate*.

Theorem 2 (Black [6])**.** *The Median Voter rule is strategyproof.*

A natural question is what other restrictions on preferences give rise to "median-like" mechanisms that are strategyproof. This question has been studied thoroughly [43], [27], [2], [46].

B. Complexity barriers

Even though the G-S theorem states that manipulations exist under any voting rule, a voter trying to manipulate might find it difficult to know *how* to manipulate. This observation led to the idea that some voting rules might be truthful *in practice*, assuming that voters have limited computational resources.

Bartholdi, Tovey, and Trick, formalized the following computational problem, which can be applied to any voting rule f.

$\mathbf{MANIPULATION_f}$: given a set of candidates A, a group of voters N, a manipulator $i \in N$, a preference profile of all voters except i $\boldsymbol{L}_{-i} = (L_1, \dots, L_{i-1}, L_{i+1}, \dots, L_n)$, and a specific candidate $p \in A$: Answer whether the manipulator can provide a preference L_i^* such that $f(\boldsymbol{L}_{-i}, L_i^*) = p$.

Then, they asked whether there is a voting rule f such that computing the outcome $f(\boldsymbol{L})$ is easy, but the problem $\mathbf{MANIPULATION_f}$ is NP-hard. Note that since the number of possible reports is $m!$, a brute-force search is typically infeasible.

At least for some voting rules, it is easy to tell whether a manipulation exists or not. E.g. in Plurality it is sufficient to let i rank p at the top of L_i', followed by all other candidates in an arbitrary order. A manipulation exists if and only if $f(\boldsymbol{L}_{-i}, L_i') = p$. Thus $\mathbf{MANIPULATION_{Plurality}}$ is in $\mathcal{P}$.

Theorem 3 ([4])**.** *There is a voting rule f such that: I) $f(\boldsymbol{L})$ can be computed in polynomial time; II)* $\mathbf{MANIPULATION_f}$ *is an NP-Complete problem.*[2]

The original proof in [4] used a variation of Copeland, and similar hardness results hold for common voting rules such as STV [3]. Note that for a fixed number of candidates m, there is a trivial polynomial-time algorithm for computing a manipulation: simply try all $m!$ possible ballots, which is also a fixed number.

A recent survey of which common voting rules are hard to manipulate by individual or by a coalition appears in [8], Section 6.4.

C. Randomized voting rules

It is easy to see that by allowing randomization, we can find strategyproof voting rules that violate the Gibbard-Satterthwaite conditions. For example, we can think of a rule that return any candidate with equal probability, regardless of the profile. The following theorem by Gibbard characterizes exactly the set of randomized strategyproof voting rules.

Theorem 4 ([23])**.** *A (randomized) voting rule f is strategyproof, if and only if it is a lottery over duples and strategyproof unilateral rules.*

A unilateral rule is a rule that depends on the report of a single voter (e.g., a dictatorial rule). It should be noted that in order to extend the notion of manipulation and strategyproofness to randomized outcomes, Gibbard assumes that each voter has a *cardinal* utility function U_i over alternatives, and a manipulation means that a voter gains in expectation.

Some recent work used Gibbard's characterization to derive strategyproof voting mechanisms with some desired properties. For example, Procaccia [55] proved that for any PSR g there is a strategyproof voting rule (i.e., a mixture of strategyproof unilateral rules and duples) f_g that outputs a candidate with expected score close to the winner of g.

f) Approximate strategyproofness: We get more flexibility if on top of randomization we slightly relax the strategyproofness requirement. Two such approximations were independently suggested by Núnez and Pivato [48] and by Birrell and Pass [5]. Both solutions consider a "target rule" g, and then mix it with some carefully designed noise to obtain a randomized rule f_g that is "close" to g and "almost strategproof," where the formal meaning of these notions differ between the models. The Núnez and Pivato model makes explicit assumptions on the distribution of preferences, whereas the one by Birrell and Pass uses tools from differential privacy.

V. Voting and Mechanism Design

In contrast to the common abstract model of voting, in many specific situations agents have well-defined cardinal utilities for each alternative, and there is a clear social goal. For example - to maximize the sum of utilities. Denote the "optimal" candidate by $a^* = \text{argmax}_{a \in A} \sum_{i \in N} U_i(a)$. A natural question is then whether we can design a strategyproof mechanism that obtains or at least approximates the maximal social welfare.

A. Payments

Adding payments allows us to transfer utility between agents with much flexibility, thereby aligning their incentives.

Suppose that each voter has cardinal utilities U_i over candidates. The Vickrey-Clarke-Groves (VCG) mechanism [70],

[2] We do not formally define here what is an NP-hard problem, and refer the reader to standard textbooks (e.g., [69]) for definitions and further discussion.

[9], [25] collect all utility functions. Then it computes the optimal outcome a^*, and charges each agent i the "damage" that this agent inflicts upon the other agents. That is, the difference between the maximal social welfare in a world where i does not exist, and the maximal social welfare (of all except i) in the current world. For a thorough exposition of VCG and a range of applications see [47].

The VCG mechanism is truthful (it is a dominant strategy for each voter to report her true utilities), and by definition it maximizes the social welfare. Both properties rely heavily on the assumption that voters' utilities are quasi-linear. Relaxing the assumption of quasi-linear utilities even slightly, leads to an impossibility result in the spirit of the Gibbard-Satterthwaite theorem [31].

B. Range Voting

Range Voting allows voters to express their cardinal preferences over candidates (normalized such that $\min_{a\in A} U_i(a) = 0$ and $\max_{a\in A} U_i(a) = 1$), and selects the one maximizing the sum of utilities. I.e. it always returns a^*.

Even without the Gibbard-Sattethwaite theorem, it is obvious that Range Voting is not truthful, as voters can always gain by reporting more extreme preferences. The G-S theorem an an even more negative implication, namely that no deterministic strategyproof mechanism can *approximate* the optimal social welfare by a factor that is sublinear in n.

Filos-Rastikas and Miltersen [20] suggested to find among the class of randomized strategyproof rules (see Sec. IV-C), the ones that give the best possible approximation for the social welfare. Their main result is a tight-to-a-constant approximation bound, that does not depend on n at all, and decreases sub-linearly with m.

C. Facility location

Facility location can be thought of as a special case of voting, where the alternatives A are possible locations for a facility in some metric space $\langle \mathcal{X}, d\rangle$ where d is a metric over the set $\mathcal{X}$. Each agent is assumed to prefer the facility to be as close as possible to her location, thus instead of reporting her entire utility function U_i, she only needs to report her location (say, some point $x_i \in \mathbb{R}^k$ or some vertex of a graph G). The cost (negative utility) of every alternative $a \in A \subseteq \mathcal{X}$ is exactly the distance $d(x_i, a)$.

The *optimal location* $a^* \in A$ is the one minimizing the *social cost* $SC(a, \boldsymbol{x}) = \sum_{i\in N} d(x_i, a)$.

A facility location mechanism is a function $g : \mathcal{X}^n \to A$, mapping the positions of all n agents to a single winning position. The special case where $A = \mathcal{X}$ is called the *unconstrained case*, as the facility can be placed anywhere, and in particular wherever an agent can be placed. Thus the contrained case is more difficult in general. The *cost approximation ratio* of g is the smallest γ s.t. for any input $\boldsymbol{x}$, $E[SC(g(\boldsymbol{x}), \boldsymbol{x})] \leq \gamma \cdot SC(a^*, \boldsymbol{x})$.

Without any restriction on the possible locations of the agents, the impossibility results of general voting rules [22], [61], [23] apply for the constrained facility location problem, which only allows for dictatorial or similar mechanisms. The welfare approximation ratio of such mechanisms can be analyzed, and shown to be $2n-1$ and $3-\frac{2}{n}$ in the deterministic and randomized cases, respectively [38].

Several papers examine variations of the problem, and in particular consider metric spaces $\mathcal{X}$ of specific shapes such as a line or a circle [62], [56], [15], [18]. For example, the Median mechanism we saw in Section IV-A provides an optimal solution for the *unconstrained* problem on a line, as agents' utilities are single-peaked.

VI. Rational Equilibrium

Once we accept that voters are going to behave strategically, and think of voting rules (with preferences) as games, we can analyze them with game theoretic tools like any other game. I will next mention several such models, which differ in their modeling assumptions.

A. Implementation

Consider any (non-resolute) SCC F, i.e. a function that maps strict preference profiles to a possibly empty set of outcomes. In what follows, $F(\boldsymbol{L}) \subseteq A$ can be thought of as some set of socially desirable alternatives under preference profile $\boldsymbol{L}$. Some examples of SCCs we might want to implement are: all Pareto optimal alternatives in $\boldsymbol{L}$; all Borda winners of $\boldsymbol{L}$ (before tie-breaking); all Condorcet winners of $\boldsymbol{L}$ (which may be empty); and so on.

Implementation of F by a mechanism g means that given any (strict) profile $\boldsymbol{L}$, a candidate $c \in A$ is in $F(\boldsymbol{L})$ if an only if voters with preferences $\boldsymbol{L}$ elect c in some *equilibrium* of g.

A most natural question is *which voting rules implement themselves* under some behavior, and if such rules even exist. This question can be extended by allowing arbitrary mechanisms that are not necessarily voting rules, and ask if a voting rule f can be implemented by some mechanism g_f using some notion of equilibrium. For example, a *truthful voting rule* implements itself in dominant strategy equilibrium.

We provide two examples of results in implementation theory that use Nash equilibrium and strong equilibrium. There are many other notions of equilibrium used in the implementation literature. Some such notions are implementation in *undominated strategies* [7], [26], *undominated Nash equilibria* [54], [64], and *mixed Nash equilibria* [26], [40].

g) NE implementation: Denote by $NE_g(\boldsymbol{L}) \subseteq A$ the set of all candidates that win in *some* Nash equilibrium of g for preferences $\boldsymbol{L}$. A mechanism g *implements a SCC F in NE*, if $NE_g(\boldsymbol{L}) = F(\boldsymbol{L})$ for all $\boldsymbol{L} \in \mathcal{L}(A)^n$.

Theorem 5 (Maskin [32])**.** *No voting rule except dictatorships and duples can be implemented in NE by any mechanism.*

Maskin further showed that if we want to implement SCCs rather than SCFs (i.e. rules that allow for more than one winner) then results are more positive, and characterized such SCCs. A trivial example is the SCC $F(\boldsymbol{L}) = A$, which can clearly be implemented (e.g. by Plurality with $n \geq 3$ voters). A less trivial example is $F(\boldsymbol{L})$ which returns all Pareto-optimal outcomes of $\boldsymbol{L}$.

h) SE implementation: Let $SE_g(\boldsymbol{L}) \subseteq A$ be all candidates that win in some strong equilibrium of mechanism g. Recall that an equilibrium is strong if there is no subset of voters that can all gain by deviating. Formally, a mechanism g implements a mapping $G : \mathcal{L}(A)^n \rightarrow 2^A$ in SE, if $SE_g(\boldsymbol{L}) = G(\boldsymbol{L})$ for all $\boldsymbol{L}$.

Note that we do not require that G is a valid SCC, as it may return the empty subset. An example of such a mapping is G_{CON}, which returns the (possibly empty) set of Condorcet winners of profile $\boldsymbol{L}$.

Theorem 6 (Sertel and Sanver [63])**.** *The Plurality voting rule implements G_{CON} in SE, for all odd n.*

B. Bayesian uncertainty in Voting

We have seen that when voters know exactly how others are going to vote, they rarely influence the outcome. Yet it is known that people often do vote strategically, or at least trying to [57]. One possible explanation for this discrepancy is *uncertainty*: since voters do not know exactly the preferences and actions of others, they know they *might* be pivotal, and hence some actions may be better than others in expectation.

The classic game-theoretic approach for *games with partial information*, assumes that each player's *type* (preferences/utilities) is sampled from some distribution and this distribution is common knowledge. Thus each player knows her own type, and some distribution on the other players' types. In equilibrium, each player is playing a mixed strategy contingent on her type, that is a best response to the (mixed) joined strategy of all other players.

Such models have been applied in a series of papers to Plurality, mainly in an attempt to explain the *Duverger Law*, which observes that in equilibrium typically only two candidates get almost all votes [60], [11], [12], [52], [19]. A general version of the model that applies for all scoring rules was suggested by Myerson and Weber [45].

An equilibrium in these models for a particular population (a distribution p_u over utility profiles) is composed of "strategies" (a mapping v from types to ballots) and "outcomes" (a distribution p_s on candidates scores), with the following self-sustaining properties: sampling a profile from p_u and apply voters' strategies v results in scores distributing according to p_s; and given p_s, voters of each type are maximizing their expected utility by voting according to v.

Other models assume voters have (uncertain) information not about other voters' preferences, but about their *actions*. Thus a voting profile is stable if every voter would choose to keep her vote assuming the outcome will be according to current profile with some "noise." This approach is highly related to election polls. Some representatives of this approach are *robust equilibrium* [39] that assumes a small probability that voters fail to cast their vote; *expectationally stable equilibrium* [53] that assumes the actual outcome an be anywhere within some small distance from the expected outcome; and *sampling equilibrium* [51] that assumes each voter is exposed to a small random sample of the other voters.

C. Other Equilibrium Models

There are many more equilibrium models, which typically focus on a specific voting rule or a class of voting rules, and make different assumptions on voters' behavior.

Some such model attribute some small positive cost to casting a ballot ("lazy bias") [13] or to casting a manipulative ballot ("truth bias") [67], [50]. This cost is sufficient to eliminate many of the unreasonable Nash equilibria of common voting rules. Another approach assumes that voters rule out dominated strategies in an iterative way until no strategy can be eliminated. While Moulin [44] showed that dominance solvable voting rules could be designed, in common rules such as Plurality, dominance leads to a unique outcome only in a small class of preferences [14].

Other models assume backward induction reasoning and analyze subgame perfect equilibria. This can apply when voters sequentially vote on parts of the decision [17], [33], [72], or when they have repeated opportunities to offer new candidates that will compete with the current winner [1].

VII. Iterative Voting and Heursitics

In the iterative voting model [37], [36], voters have fixed preferences and start from some announcement (e.g., sincerely report their preferences). Votes are aggregated via some predefined rule (e.g. Plurality), but can change their votes after observing the current announcements and outcome. The game proceeds in turns, where a single voter changes his vote at each turn, until no voter has objections and the final outcome is announced. This process is similar to online polls via Doodle or Facebook, where users can log-in at any time and change their vote. Similarly, in offline committees the participants can sometimes ask to change their vote, seeing the current outcome.

Major questions regarding iterative voting are whether it converges, and how good is the outcome to the society. For Plurality, Meir at al. [37] showed that if each voter in turn plays a best-response to the current votes of others, the game will always converge to a Nash equilibrium. This question was later studied in other voting rules, from which only Veto seems to have similar properties [59], [30], [28].

Interestingly, simulations show that in practice all common voting rules almost always converge. Further, the equilibrium result is often *better for the society* than the truthful outcome [35], [28].

A. Heursitics

Many of the objections raised in the previous sections regarding Nash equilibria, equally apply to any model based on best response, as each particular voter is unlikely to be pivotal. Things become more complicated when voters are assumed to employ more flexible *heuristics*. Such heuristics may be used either if voters only know part of the current state, or are uncertain how well it will reflect the final votes.

Some heuristics are based on common sense reasoning like focusing on the few candidates with highest current scores [58], [24], [49]. Simulations show that these heuristics almost always lead to convergence when applied in an iterative voting setting [24], [49]. Some heuristics are tailored for a specific voting rule, such are Laslier's *Leader Rule* for Approval [29], where a voter approves all candidates strictly preferred to the leader, and approves the leader if it is preferred to the runnerup.

Another approach is to generate heuristics based on dominance relations, by explicitly defining the information sets of the voters under uncertainty [10], [68], [35], [16]. One specific model is *local dominance* [35], [34], where the voter assumes that the actual candidate scores are within certain distance from the poll or current scores. In an iterative Plurality voting, this provably leads to convergence.

Extensive simulations of Plurality voting with Local Dominance heuristics show that a moderate level of uncertainty leads to the highest amount of strategic behavior, in particular when the population is diverse [35]. Further, more strategic behavior in turn leads to higher social welfare.

VIII. Summary: Towards a complete theory of strategic voting

Social choice is perhaps the oldest topic that received formal game-theoretic analysis, much before the term game theory was coined. Yet while economists, political scientists, mathematicians (and now computer scientists) all agree that Nash equilibrium is not an appropriate solution concept for voting, there does not seem to be a single acceptable theory for strategic voting.

This might be due to the fact that, as in other cases that concern with human behavior, strategic voting involves many factors. Some of these factors may be domain specific and/or depend on complex cognitive and social processes that some models ignore or capture in different ways. Meir et al. [35] suggested a *desiderata* which is intended to provide a way to compare the strengths and weaknesses of the many different theories: some make implausible informational or cognitive assumptions, some predict behavior that is contrary to empirical evidence, some are prohibitively difficult to analyze, and some are wonderful but restricted to very specific scenarios.

I sincerely hope that familiarity with the classical and recent approaches to strategic voting will encourage researchers to develop new and better models that with improve our understanding. This, in turn, may lead to the design of improved aggregation mechanisms that lead the society to outcomes that are better for every one.

References

[1] Stéphane Airiau and Ulle Endriss. Iterated majority voting. In *Proceedings of 1st ADT*, pages 38–49, 2009.

[2] Salvador Barberà, Jordi Massó, and Alejandro Neme. Voting under constraints. *journal of economic theory*, 76(2):298–321, 1997.

[3] J. Bartholdi and J. Orlin. Single Transferable Vote resists strategic voting. *Social Choice and Welfare*, 8:341–354, 1991.

[4] J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.

[5] Eleanor Birrell and Rafael Pass. Approximately strategy-proof voting. Technical report, DTIC Document, 2011.

[6] Duncan Black. On the rationale of group decision-making. *Journal of political economy*, 56(1):23–34, 1948.

[7] Tilman Börgers. Undominated strategies and coordination in normalform games. *Social Choice and Welfare*, 8(1):65–78, 1991.

[8] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[9] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[10] Vincent Conitzer, Toby Walsh, and Lirong Xia. Dominating manipulations in voting with partial information. In *Proceedings of 25th AAAI*, 2011.

[11] Gary W Cox. Duverger's law and strategic voting. *unpublished paper, Department of Government, University of Texas, Austin*, 1987.

[12] Gary W Cox. Strategic voting equilibria under the single nontransferable vote. *American Political Science Review*, 88(3):608–621, 1994.

[13] Yvo Desmedt and Edith Elkind. Equilibria of plurality voting with abstentions. In *Proceedings of 11th ACM-EC*, pages 347–356, 2010.

[14] Amrita Dhillon and Ben Lockwood. When are plurality rule voting games dominance-solvable? *Games and Economic Behavior*, 46(1):55–75, 2004.

[15] Elad Dokow, Michal Feldman, Reshef Meir, and Ilan Nehama. Mechanism design on discrete lines and cycles. In *Proceedings of 13th ACM-EC*, pages 423–440, 2012.

[16] Ulle Endriss, Svetlana Obraztsova, Maria Polukarov, and Jeffrey S. Rosenschein. Strategic voting with incomplete information. In *Proceedings of 25th IJCAI*, 2016.

[17] Robin Farquharson. *Theory of Voting*. Yale Uni. Press, 1969.

[18] Michal Feldman, Amos Fiat, and Iddan Golomb. On voting and facility location. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 269–286. ACM, 2016.

[19] Mark Fey. Stability and coordination in duverger's law: A formal model of preelection polls and strategic voting. *American Political Science Review*, 91(1):135–147, 1997.

[20] Aris Filos-Ratsikas and Peter Bro Miltersen. Truthful approximations to range voting. In *International Conference on Web and Internet Economics*, pages 175–188. Springer, 2014.

[21] Ehud Friedgut, Gil Kalai, and Noam Nisan. Elections can be manipulated often. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 243–249. IEEE, 2008.

[22] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–601, 1973.

[23] A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45:665–681, 1977.

[24] Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Restricted manipulation in iterative voting: Condorcet efficiency and Borda score. In *Proceedings of 3rd ADT*, pages 181–192, 2013.

[25] Theodore Groves. Incentives in teams. *Econometrica*, pages 617–631, 1973.

[26] Matthew O Jackson. Implementation in undominated strategies: A look at bounded mechanisms. *The Review of Economic Studies*, 59(4):757–775, 1992.

[27] E. Kalai and E. Muller. Characterization of domains admitting nondictatorial social welfare functions and nonmanipulable voting procedures. *Journal of Economic Theory*, 16:457–469, 1977.

[28] Aaron Koolyk, Tyrone Strangway, Omer Lev, and Jeffrey S. Rosenschein. Convergence and quality of iterative voting under non-scoring rules. In *Proceedings of 26th IJCAI*, 2017. to appear.

[29] Jean-François Laslier. The leader rule: A model of strategic approval voting in a large electorate. *Journal of Theoretical Politics*, 21(1):113–136, 2009.

[30] Omer Lev and Jeffrey S Rosenschein. Convergence of iterative scoring rules. *Journal of Artificial Intelligence Research*, 57:573–591, 2016.

[31] Hongyao Ma, Reshef Meir, and David C Parkes. Social choice for agents with general utilities. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 345–351. AAAI Press, 2016.

[32] Eric Maskin. Nash equilibrium and welfare optimality. *The Review of Economic Studies*, 66(1):23–38, 1999.

[33] Richard D McKelvey and Richard G Niemi. A multistage game representation of sophisticated voting for binary procedures. *Journal of Economic Theory*, 18(1):1–22, 1978.

[34] Reshef Meir. Plurality voting under uncertainty. In *Proceedings of 29th AAAI*, pages 2103–2109, 2015.

[35] Reshef Meir, Omer Lev, and Jeffrey S. Rosenschein. A local-dominance theory of voting equilibria. In *Proceedings of 15th ACM-EC*, 2014.

[36] Reshef Meir, Maria Polukarov, Jeffrey S Rosenschein, and Nicholas R Jennings. Iterative voting and acyclic games. *Artificial Intelligence*, 252:100–122, 2017.

[37] Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nick Jennings. Convergence to equilibria of plurality voting. In *Proceedings of 24th AAAI*, pages 823–828, 2010.

[38] Reshef Meir, Ariel D Procaccia, and Jeffrey S Rosenschein. Algorithms for strategyproof classification. *Artificial Intelligence*, 186:123–156, 2012.

[39] Matthias Messner and Mattias Polborn. Robust political equilibria under plurality and runoff rule. *IGIER Working Paper*, 2005.

[40] Claudio Mezzetti and Ludovic Renou. Implementation in mixed nash equilibrium. *Journal of Economic Theory*, 147(6):2357–2375, 2012.

[41] Elchanan Mossel, Ariel D Procaccia, and Miklós Z Rácz. A smooth transition from powerlessness to absolute power. *Journal of Artificial Intelligence Research*, pages 923–951, 2013.

[42] Elchanan Mossel and Miklós Z Rácz. A quantitative gibbard-satterthwaite theorem without neutrality. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1041–1060. ACM, 2012.

[43] H. Moulin. On strategy-proofness and single-peakedness. *Public Choice*, 35:437–455, 1980.

[44] Hervé Moulin. Dominance solvable voting schemes. *Econometrica: Journal of the Econometric Society*, pages 1337–1351, 1979.

[45] R. B. Myerson and R. J. Weber. A theory of voting equilibria. *The American Political Science Review*, 87(1):102–114, 1993.

[46] Klaus Nehring and Clemens Puppe. The structure of strategy-proof social choice – part I: General characterization and possibility results on median spaces. *Journal of Economic Theory*, 135(1):269 – 305, 2007.

[47] N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9. Cambridge University Press, 2007.

[48] Matıas Núnez and Marcus Pivato. Truth-revealing voting rules for large populations. In *Proceedings of 6th COMSOC*. 2016.

[49] Svetlana Obraztsova, Evangelos Markakis, Maria Polukarov, Zinovi Rabinovich, and Nicholas R Jennings. On the convergence of iterative voting: How restrictive should restricted dynamics be? In *Proceedings of 29th AAAI*, pages 993–999, 2015.

[50] Svetlana Obraztsova, Evangelos Markakis, and David R. M. Thompson. Plurality voting with truth-biased agents. In *Algorithmic Game Theory*, pages 26–37. Springer, 2013.

[51] Martin J Osborne and Ariel Rubinstein. Sampling equilibrium, with an application to strategic voting. *Games and Economic Behavior*, 45(2):434–441, 2003.

[52] Thomas R Palfrey. A mathematical proof of duverger's law. 1988.

[53] Thomas R Palfrey and Howard Rosenthal. Testing game-theoretic models of free riding: New evidence on probability bias and learning. 1990.

[54] Thomas R Palfrey and Sanjay Srivastava. Nash implementation using undominated strategies. *Econometrica: Journal of the Econometric Society*, pages 479–501, 1991.

[55] Ariel D Procaccia. Can approximation circumvent gibbard-satterthwaite? In *Proceedings of 24th AAAI*, 2010.

[56] Ariel D Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 177–186. ACM, 2009.

[57] Michel Regenwetter. *Behavioral social choice: probabilistic models, statistical inference, and applications*. Cambridge University Press, 2006.

[58] Annemieke Reijngoud and Ulle Endriss. Voter response to iterated poll information. In *Proceedings of 11th AAMAS*, pages 635–644, 2012.

[59] Reyhaneh Reyhani and Mark C Wilson. Best-reply dynamics for scoring rules. In *Proceedings of 20th ECAI*, 2012.

[60] William H Riker. The number of political parties: A reexamination of duverger's law. *Comparative Politics*, 9(1):93–106, 1976.

[61] M. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–216, 1975.

[62] J. Schummer and R. V. Vohra. Strategy-proof location on a network. *Journal of Economic Theory*, 104(2):405–428, 2004.

[63] Murat R. Sertel and M. Remzi Sanver. Strong equilibrium outcomes of voting games are the generalized condorcet winners. *Social Choice and Welfare*, 22:331–347, 2004.

[64] Tomas Sjöström. Implementation in undominated nash equilibria without integer games. *Games and Economic Behavior*, 6(3):502–511, 1994.

[65] Arkadii Slinko and Shaun White. Is it ever safe to vote strategically? *Social Choice and Welfare*, 43(2):403–427, 2014.

[66] L.-G. Svensson. The proof of the Gibbard-Satterthwaite theorem revisited. Working Paper No. 1999:1, Department of Economics, Lund University, 1999. Available from: http://www.nek.lu.se/NEKlgs/vote09.pdf.

[67] David R. M. Thompson, Omer Lev, Kevin Leyton-Brown, and Jeffrey S. Rosenschein. Empirical analysis of plurality election equilibria. In *Proceedings of 12th AAMAS*, pages 391–398, 2013.

[68] Hans van Ditmarsch, Jérôme Lang, and Abdallah Saffidine. Strategic voting and the logic of knowledge. In *Proceedings of 14thTARK*, 2013.

[69] Jan Van Leeuwen. *Handbook of theoretical computer science (vol. A): algorithms and complexity*. Mit Press, 1991.

[70] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[71] Lirong Xia and Vincent Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 99–108. ACM, 2008.

[72] Lirong Xia, Vincent Conitzer, and Jérôme Lang. Strategic sequential voting in multi-issue domains and multiple-election paradoxes. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 179–188. ACM, 2011.

Predictive Analytics with Structured and Unstructured data - A Deep Learning based Approach

Lipika Dey, Hardik Meisheri and Ishan Verma, TCS Research, New Delhi, India

***Abstract*— Predictive analytics over structured time-series data has always been an active area of research. With plentitude of textual information generating across different sources on the web, researchers have started combining relevant structured and unstructured data to improve predictions. In this paper, we have present a generic deep learning framework for predictive analytics utilizing both structured and unstructured data. We also present a case-study to validate the functionality and applicability of the proposed framework where we use LSTM for prediction of structured data movement direction using events extracted from news articles. Experimental results shows that the proposed model outperforms existing baseline.**

***Index Terms*— Predictive analytics, deep neural networks, text-driven analytics**

I. Introduction

BUSINESS Intelligence (BI) refers to a collection of technologies that help organizations analyze data to derive actionable intelligence. Traditionally, application of BI technologies has been restricted to structured, numerical business data like those reporting sales figures, customer acquisition figures etc. With the advances in unstructured data analytics techniques over the last one and half decade, businesses have also started looking at unstructured data like customer feedback, social media content, organizational communications etc. to gather intelligence around consumer sentiments, competition landscape etc. However, what is still missing is a common framework that can pull and analyze a multitude of heterogeneous data of both structured and unstructured types, to provide a richer set of insights.

The advantages of linking data from a multitude of sources are many. First of all, while business figures can clearly indicate how the business performed, they can hardly explain why it performed so. Often times, the attributable causes can be known from sources extraneous to the enterprise. For example, possible reasons for the dip in sales for a newly launched car may be learnt from social media. Further, the causal events themselves may be extraneous to the organization. For example, the actual reason for a dip in sales may be due to a higher interest rate levied on automobile loans. Similarly, for certain periods, the reasons for a bad market may be attributed to socio-political disturbances over a region, rather than anything to do with the product or the organization itself.

While Business Analysts do use this kind of information, there exists no formal framework where structured and unstructured information can be simultaneously used for gathering business insights to make informed decisions. One of the key challenges was to learn these models from large volumes of historical data due to the computational complexities of text processing, both from quantitative and qualitative perspectives. Any kind of text processing requires handling of high volumes of data due to the potentially large vocabulary size in any natural language document collection. Further complexities are introduced due to issues like synonyms, lack of fixed structure, spelling errors, ever-changing vocabulary etc.

Deep learning based text-processing methods have provided an alternative way to represent words and text documents as vectors of numbers in a fixed dimensional space. It has been shown that these representations are capable of preserving and detecting semantic similarities without the use of additional linguistic resources. Deep neural networks have been employed to learn predictive models from large volumes of text data. Among these, the particularly interesting variants are those of Long Short-Term Memory (LSTM) networks [27], which exploit the sequential nature of words in a text for text classification. Obviously, LSTMs are not restricted to be used only for text classification, rather can be utilized for any kind of sequence-based prediction tasks.

In this paper, we propose the use of deep neural networks to build predictive models using both structured and unstructured data. The text data is first classified into pre-determined categories using supervised learning. Thereafter, quantification of the text data is done using volumes of category labels aggregated over pre-specified time units. An LSTM based network is thereafter employed to consume a multitude of time-series and predict the future events. We have presented the feasibility and effectiveness of the approach by predicting stock-market movements using both historic stock data and News data. We show that the combined model can predict stock movements more accurately than traditional predictive methods, which uses the time series for

Lipika Dey, Hardik Meisheri and Ishan Verma are with the TCS Research, New Delhi, India; email: (lipika.dey, hardik.meisheri, ishan.verma)@tcs.com

stock values only.

The model can be applied towards predicting any business data like sales, market-share etc. However, availability of large volumes of relevant historical text data is not easily available for most organizations. Typically, the kind of data that would be useful for such tasks would be customer communications, product- or service-related complaints, company policy-related documents etc.

II. Predictive Analytics – a brief overview of techniques used by Business Analysts

For ages, business analysts have relied on predictive analytics models to identify risks and opportunities. These models learn relationships among many factors that are likely to affect business performance. These models are then employed to predict future outcomes depending on current factors, thus allowing analysts to obtain an assessment of risks or opportunities for a particular situation. Analysts also usually run “what if” simulations, wherein they experiment with various values of current factors for potential assessment of risks associated with different situations. Predictive analytics thus enables guided decision making.

Predictive analytics has a wide-range of applications in business. The predictive models provide a probability score for measurable business data elements. It may provide the probability of a customer churning or buying a product or service in the near future. It may provide an estimate of the number of vehicles or insurance policies likely to be sold in the next three months. It may also provide the likelihood of a customer defaulting on a loan based on his or her personal history or characteristics. In yet another set of applications, predictive analytics may be applied to determine whether a transaction is fraudulent or not, based on certain patterns learnt from the past. Applying predictive analytics for predicting stock data movement is an age-old statistical problem. Statistical models are also available for predicting the actual stock price or stock volume of a commodity.

In this section, we will try to provide a very brief overview of various classes of predictive techniques, which will just help understand the different classes of models. Providing in-depth knowledge about any specific technique is beyond the scope of the paper.

As seen from the examples, the context of applying predictive analytics can be quite diverse. The outcomes expected can also vary from simple binary values like YES / NO or TRUE /FALSE for churn or fraud prediction to predicting actual numerical values for stocks or sales. Consequently, the types of methods applied for the tasks are also quite diverse.

Predictive analytics techniques can be broadly grouped into regression techniques and machine learning techniques.

A. Regression Techniques

Regression analysis is a predictive modelling technique which explores the dependence relationship of a target variable, like sales trends, churn possibility or possibility of a transaction to be fraudulent etc. on predictor variables. Regression techniques focus on finding a mathematical equation that can capture the interactions between the different variables in consideration, typically trying to minimize the overall error between the model-predicted values and the real values.

Linear regression models learn to predict the response variable as a linear function of the parameters. These parameters are learnt or adjusted so that a measure of fit like the sum of squared residuals is minimized.

Logistic regression models [2] on the other hand assign probabilities to possible outcomes. A binary outcome variable is transformed to an unbounded continuous variable, and a regular multivariate model is estimated.

Time series models are used for forecasting future behaviour of variables when the past data points exhibit internal structures like autocorrelation, trends or seasonal variations. Stock or sales data are perfect examples of such data. Standard regression techniques cannot model these internal structures. Time series models are capable of decomposing these trends and seasonal components and thereby produce better models. Popular time series models are autoregressive model (AR), moving average (MA), a combination of the two called auto-regressive moving average (ARMA) and auto-regressive integrated moving average (ARIMA) [1].

Classification and regression trees are non-parametric decision tree learning methods that produce classification or regression trees depending on whether the dependent variable is categorical or numeric. These are based on hierarchical optimal discriminant analysis that are generalizations of optimal discriminant analysis. Decision trees [3] are a collection of rules based on variables in the modelling data set, where the rules are deduced to obtain the best split to differentiate observations belonging to different target classes. Decision tree rules are explanatory and are often preferred by analysts. For example, to decide whether a new loan applicant is likely to default or not, a decision tree based model not only predicts the decision but also provides the rule that was applied to come to the decision thereby helping the analyst understand the reasons for it.

B. Machine Learning Techniques

Machine learning based models are also applied for predictive analytics in applications like medical condition diagnostics, fraud detection etc. However, unlike regression or classification trees, the model here remains a black-box without explicit insight into the underlying relationships among the predictor variables, which can be terribly complex, and it is deemed sufficient to predict the dependent variable only.

Among the various machine learning models, the artificial neural networks [4, 5] that were introduced in the sixties, inspired by the human nervous system have gained a huge surge in popularity in recent times, due to their capability to learn very complex relationships among large numbers of predictor variables. There are a wide variety of neural network architectures that are useful for different types of classification tasks. While earlier, neural networks were restricted to using three layers of neurons, the input layer, a hidden layer and an output layer, deep neural networks [6,7], with more than one hidden layers have gained popularity. A large number of neurons and their interconnections are capable of modelling highly non-linear relationships between input and output variables. These architectures are also found to extract useful features by themselves from large volumes of training samples without explicit feature engineering.

Below are some of the most commonly used networks that are suitable for prediction tasks:

1) Multilayer Perceptron

These networks use more than one hidden layer of neurons and are also known as deep feed-forward neural networks.

2) Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [8] are a class of neural networks which performs convolutions between the desired filter and the input data. These Networks are highly efficient in learning hierarchal features from the data by capturing neighbouring relationships among features.

3) Recurrent Neural Network

These are types of neural network in which hidden layer neurons have self-connections, thereby making it possible for a neuron to possess memory. Recurrent neural networks [9] are very suitable for text processing tasks since the interpretation of a word in a text is dependent on its context or its neighbouring words. Thus, networks that can model these interrelationships of words by taking their sequence into account are better at text class prediction tasks than their counterparts, which treat a text as a bag of words.

4) Long-Short Term Memory Network (LSTM)

These are extensions of recurrent neural networks in which memory cell is incorporated inside each hidden layer neuron. LSTMs are good at modelling long-distance relationships also among variables like words within a text separated by many words in between. LSTMs are applicable for analyzing any kind of sequential data.

III. USING UNSTRUCTURED DATA FOR PREDICTIVE ANALYTICS – RELATED WORK

As discussed earlier, till very recently much of predictive analysis in business dealt with structured data alone. With the organizations opening up to the idea of using consumer-generated unstructured text like complaints, service logs, social media data etc. several researchers are exploring the effect of these types of text data on business outcomes. One popular area of research has been to study the effects of online reviews on sales of products and services. Effects of online reviews on few areas like fashion and movies have been studied in depth. While most of these works study the impact of the reviews, few provide mechanisms for including the factors in a predictive model.

In [11], authors have provided a detailed survey of various types of predictive techniques used for predicting the fashion market. The fashion market is affected by several factors like changing weather conditions, trans-continental production facilities, holidays, public events as well as economic situations etc. Along with these, lack of historical data for new kinds of fashion items motivated analysts working in this area to especially look for methods that can combine unstructured data into the predictive process to make better predictions.

One stream of research in this area focuses on the integration of expert judgment and combining it with statistical forecasts, as [13] and [14] have shown that adjustments to the statistical models based on past knowledge of experts led to more accurate forecasts.

However, a second stream of research focuses on the use of machine learning based models that could easily integrate unconventional factors like the ones stated above. Machine learning based models were found to produce much better results than traditional regression techniques for this domain.

Sun et al. [15] proposed the use of extreme learning methods (ELM) for forecasting sales at item levels. In [16], Thomassey and Happiette propose the use of soft computing methods like fuzzy inference systems and neural networks to predict sales. Teucke et al. [17] proposed using a combination of decision trees to determine articles likely to be re-ordered and then support vector machines for the actual forecasts to obtain more accurate results. While several other variations of these models have been proposed by other researchers, as presented in [11], the point to be noted is that most of these works have used only numbers from the retail industry to build their models.

In [10], Yu et al. analyzed large volumes of online movie reviews and showed that both the sentiments expressed in the reviews and quality of reviews have a significant impact on the future sales performance. Sentiments are detected using Probabilistic Latent Semantic analysis (PLSA). Based on the sentiments, they propose an autoregressive sentiment-aware model for sales prediction. This model is further improved by considering the quality factor of a review.

An area where text inputs have been heavily used for prediction is that of News for predicting stock market data. Researchers in a number of studies have analyzed texts from social network services (SNS), blogs and news to analyze correlations between stock prices and public emotion as a reaction to social events and news [28-32]. In [33] Luss and Aspremont show that information extracted from news articles can be used to predict intraday price movements of financial assets using support vector machines. In [34], Verma et. al. presented results to show that stock trends can be better predicted by taking News events into consideration along with actual stock values.

From the above discussion, we find that though machine-learning based models are gaining traction with business analysts, using unstructured data for prediction is yet to pick up. We also see that though they have potential to handle large volumes of data, deep-learning based models have not gained wide attention of researchers in this field. In the next few sections, we propose a deep-learning based model, which can effectively do this.

IV. Deep learning models for Predictive Analytics

We choose deep learning architectures for predictive analytics since these models can learn from large volumes of sequential input data without explicit feature engineering. Additionally, the same architectures can easily handle both structured and unstructured sequential data. Before going into the details of the proposed model, we provide a brief introduction of word vectors and document vectors, which are used for predictions.

Semantic vector space models of language represent each word as a real-valued vector. These vectors can be used as features in a variety of text-based applications, such as information retrieval, document classification, question answering, and named entity recognition. Bengio et al. [22] introduced a model that learns word vector representations as part of a simple neural network architecture for language modelling.

Recently, Mikolov et al. [21] introduced a scheme for learning the representations using skip-grams. They also proposed a new evaluation scheme based on word analogies that probe the finer structure of the word vector space by examining not the scalar distance between word vectors, but rather their various dimensions of difference. For example, the analogy “king is to queen as man is to woman” should be encoded in the vector space by the vector equation king − queen = man − woman. Though this model does well on analogy tasks, they do not utilize the statistics of the corpus since they train on separate local context windows instead of on global co-occurrence counts, and hence perform poorly for most text-based applications. In 2014, Penington et al. [18] present a set of global vectors for words, called GloVe vectors which uses a specific weighted least squares model that trains on global word-word co-occurrence counts and thus makes efficient use of statistics. This model produces a word vector space with meaningful substructure that are found to be suitable for a large number of downstream text processing tasks.

Vector-based representation for words can be further extended to a paragraph or whole document. Mikolov et. al. [23], proposed a scheme where along with learning word representation if explicit information about the paragraph or the document is given, it can effectively encode the whole document into fixed sized vectors. These representations are useful over word representations while processing long text.

In the next section, we present the complete architecture for our proposed predictive systems.

A. Text Classification for Business Data Prediction

In order to effectively quantify without losing human understandability, unstructured business data needs to be assigned to pre-specified categories. One can treat these categories as event labels pertaining to different event type which are known and have defined explanation. One such set of business events is the PESTEL framework where PESTEL represents Political, Economic, Social, Technological, Environmental and Legal events. The set of PESTLE events [12] had been proposed for analyzing general environments of a business organization. These events represent six broad macro-economic factors that may have an impact on an organization.

To evaluate the impact of PESTEL events (or any set of pre-determined business events) on business outcomes, one can use Granger Causality test**.** The Granger causality test is a statistical hypothesis test for determining whether a time-series X is useful in forecasting another time-series Y. Time-series X is accepted as impacting Y, if and only if, prediction of future values of Y improves after taking values of X into consideration.

In this case our null hypothesis is that ‘PESTEL factors reported in News do not Granger-cause changes in indices’. We apply Granger causality test using PESTEL event occurrence time-series and stock indices time-series.

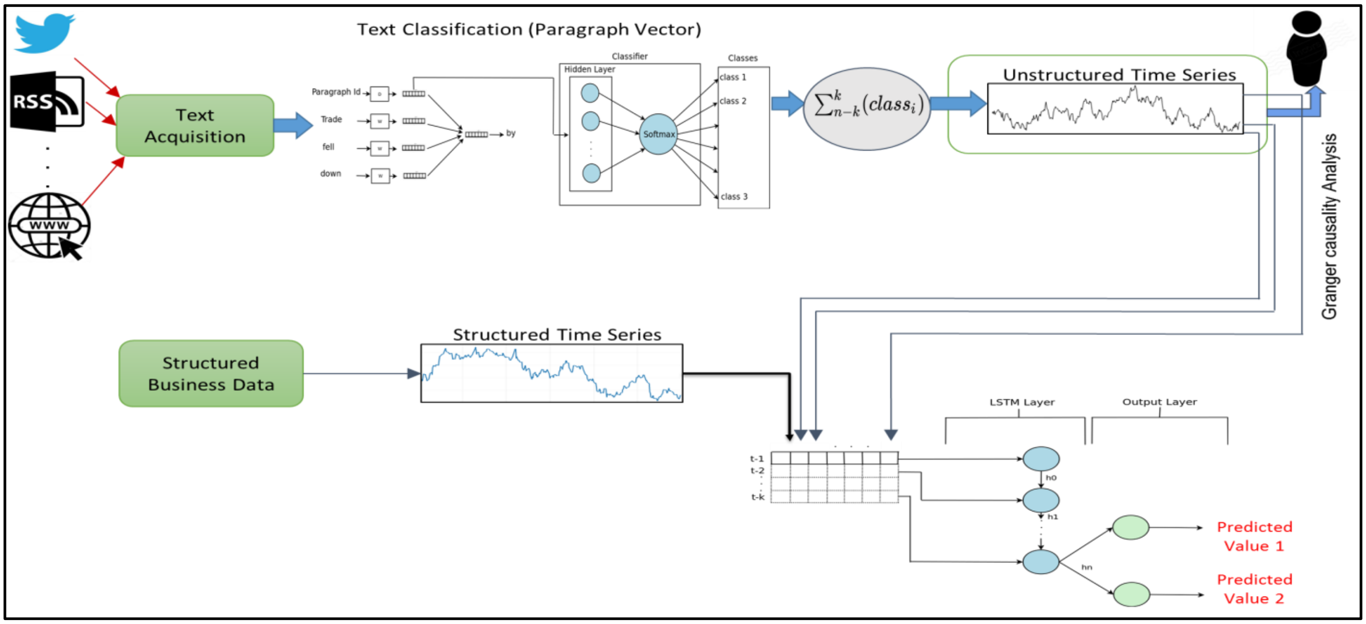


Figure 1 System Architecture

$$y_t = \alpha + \sum_{i=1}^{n} \beta_i y_{t-i} + \varepsilon_t$$

$$y_t = \alpha + \sum_{i=1}^{n} \beta_i y_{t-i} + \sum_{i=1}^{n} \gamma_i x_{t-i} + \varepsilon_t$$

The first one tries to predict future values of Y from past values. The second model takes past values of both X and Y to predict future values of Y, where series X represents quantified PESTEL factors.

V. System Architecture

Figure 1 shows the complete generic system pipeline from information acquisition to final prediction for the proposed predictive system. The upper layer is dedicated for acquiring and processing unstructured text. Text documents are acquired from various sources like news websites, consumer forums, twitter as well as other social media sites, using dedicated crawlers, APIs or RSS feeds. This layer supports stream processing. Structured information can be obtained from various business sources such as market capital, market share, stock etc. These can also be streaming in from different sources or read from internal databases.

All text content is subjected to classification. A multi-label classifier that can assign probabilities for each class to each document is designed for the purpose.

This is achieved through a two-step procedure. First, we use paragraph vector model [23] as stated earlier to represent the long text into a vector with fix dimension. Second, we map this fix dimension vector to our predefined classes using a softmax classifier which gives us the probability of each class in that document.

Paragraph vector model is an extension of learning distributed word representations using gradient descent method. While learning, for every window document ID is given to the model to learn the overall representation of that document. Window here is defined as the number of words model uses to predict the next word in the series. Paragraph vector *D* is learnt along with the word vectors *W*.

Since external world data like tweets, News or social media sentiments stream in continuously, but business data is only aggregated over fixed time-periods like days, weeks or months, an aggregator is introduced over the softmax outputs. The aggregator aggregates the classifier content over fixed time-periods for each class. This aggregation over temporal domain generates a time series from unstructured data. This is then combined with the structured data time-series in the LSTM network to provide the final prediction.

A. LSTM

LSTMs are a special case of RNNs where they tackle the classical problem of vanishing (or exploding) gradients. Information flow in LSTMs are controlled through three different gates, forget gate, output gate and input gate. Along with these gates, there is memory cell which is present, which helps in remembering long sequences. The whole process of learning can be formulated into information flow over a temporal domain. The function of input gate is to regulate new information flowing, output gate regulates the outward flow of information for that time step and forget gate regulates the amount of flow that should be discarded. Single cell of LSTM is shown in Figure 2. <> represents a vector, p denotes the number of hidden units in an LSTM cell. Input gate (denoted by i), Output gate (denoted by o), forget gate (denoted by f) and memory cell (denoted by c) govern the LSTM cell. *t* indicates the time-step or t^{th} time instance. LSTM follows following equations:

$$i_t^p = \sigma(w_{xi}^p x_t + w_{hi}^p h_{t-1}^p + w_{ci}^p c_{t-1}^p + b_i^p)$$

$$f_t^p = \sigma(w_{xf}^p x_t + w_{hf}^p h_{t-1}^p + w_{cf}^p c_{t-1}^p + b_f^p)$$

$$c_t^p = f_t^p c_{t-1}^p + i_t \tanh(w_{xc}^p x_t + w_{hc}^p h_{t-1}^p + b_c^p)$$

$$o_t^p = \sigma(w_{xo}^p x_t + w_{ho}^p h_{t-1}^p + w_{co}^p c_{t-1}^p + b_o^p)$$

$$h_t^p = o_t^p \tanh(c_t^p)$$

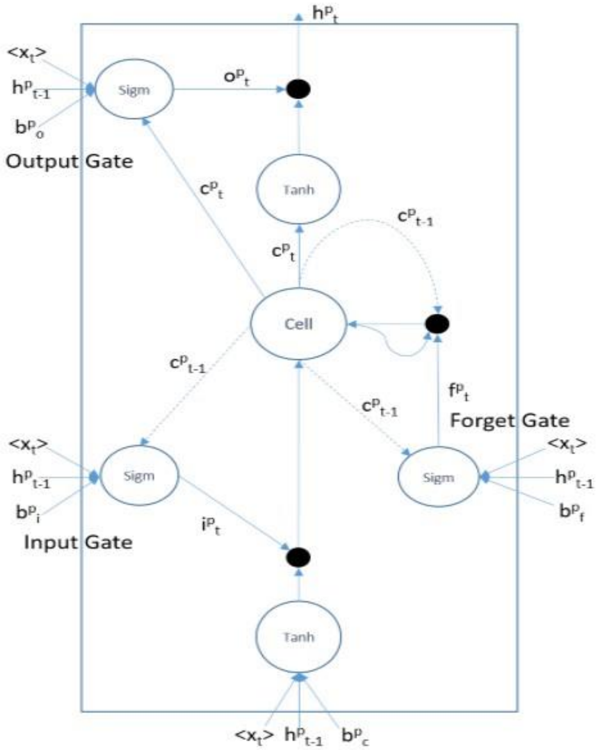


Figure 2 LSTM Cell

Training procedure in LSTM are done through truncated Back Propagation through time (BPTT) [24], which applies backpropagation over time and make changes in the weights after the whole sequence has been passed through LSTM cell.

The non-stationarity present in most of the time series can be modelled using LSTM, where it can store seasonality as well as the overall trend which effectively uses long term and short term dependencies.

VI. Case Study

Due to confidentiality issues associated to internal structured and unstructured business data, we are presenting results for publically available news and stock market data. There is also huge consensus in the business community on the importance of news articles over stock prices. Research results are also available for the same. Hence, we show results over this domain using the predictive analysis framework presented earlier. The accuracy of trend prediction is shown to better than other machine learning models.

For this task, we have used News articles as the source of text information. News articles are pulled from the RSS feeds provided by a number of agencies and crawling news websites. Stock acquisition is done from the official website of National Stock Exchange of India (https://www.nseindia.com/). Next, we have used paragraph vector model with a window size of 8 and fix learning rate of 0.25.

Next, vector representation of each news document is mapped to one the six PESTEL classes as described earlier.

We have used 5 different Indexes from the National Stock Exchange namely, NIFTY 50 (represents about 65% of free floating market capitalization), NIFTY Bank (consists of 12 large Indian banking stocks), NIFTY Auto (15 stocks consisting of Automobile sector including cars & motorcycles, heavy vehicles, auto ancillaries, tires, etc.), NIFTY IT (Companies in this index are those that have more than 50% of their turnover from IT related activities), NIFTY Energy (Index includes companies belonging to Petroleum, Gas and Power sub sectors).

We have investigated the dependency through Granger causality between structured and unstructured time series. In addition to that, we use these PESTEL classes as input to LSTM model for predicting weather stock will rise or fall.

A. Data Description

Documents are collected as mentioned earlier from different sources. The collection contains over 258144 documents collected for the duration of Jan-2013 to Feb-2017. For training paragraph vector model we have used approximately 1200 labelled document equally distributed for each PESTEL class.

Data aggregated over a day is then split into training and testing set. Data from 1st Jan 2013 till 31st Aug 2016 is used for training purpose while data from 1st Sep 2016 till 31st Jan 2017 is used for testing set. Data corresponding to stock is not available for weekend and public holidays due to which total instances for prediction tasks are 912, out of which 820 are used for training and 92 for testing.

B. Experiments and Results

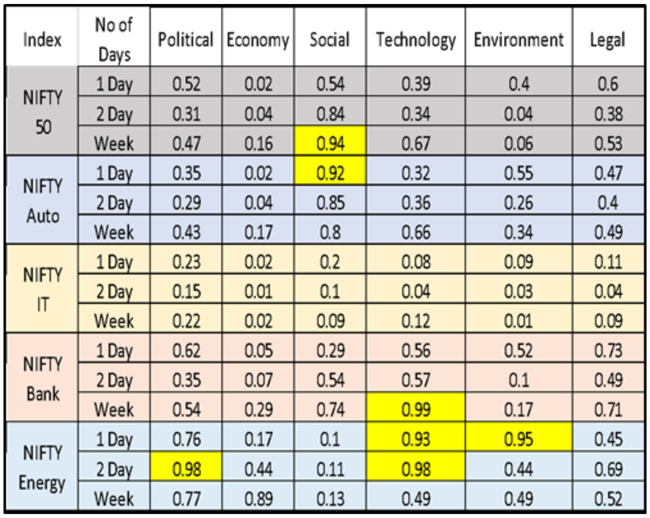
Table 1 shows the confusion matrix of classification over PESTEL classes using paragraph vector model. After 10 fold cross-validation of 70-30 train test split, we achieved 91.94% accuracy for PESTEL classes.

Table 1 Confusion Matrix for PESTEL Classifier

	Pol.	Eco.	Soc.	Tech.	Env.	Leg.
Political	187	0	0	0	0	13
Economy	0	174	0	26	0	0
Social	3	0	184	10	0	3
Technology	0	0	3	190	0	7
Environment	0	10	0	13	174	3
Legal	0	0	0	3	0	197

Table 2 shows results for Granger causality between different stock indices and the PESTEL classes aggregated over a day. Highlighted cells in the table indicate which PESTEL features have a high impact on indices. In addition to this following observations can be made:

Table 2 Granger causality results



Index	No of Days	Political	Economy	Social	Technology	Environment	Legal
NIFTY 50	1 Day	0.52	0.02	0.54	0.39	0.4	0.6
	2 Day	0.31	0.04	0.84	0.34	0.04	0.38
	Week	0.47	0.16	0.94	0.67	0.06	0.53
NIFTY Auto	1 Day	0.35	0.02	0.92	0.32	0.55	0.47
	2 Day	0.29	0.04	0.85	0.36	0.26	0.4
	Week	0.43	0.17	0.8	0.66	0.34	0.49
NIFTY IT	1 Day	0.23	0.02	0.2	0.08	0.09	0.11
	2 Day	0.15	0.01	0.1	0.04	0.03	0.04
	Week	0.22	0.02	0.09	0.12	0.01	0.09
NIFTY Bank	1 Day	0.62	0.05	0.29	0.56	0.52	0.73
	2 Day	0.35	0.07	0.54	0.57	0.1	0.49
	Week	0.54	0.29	0.74	0.99	0.17	0.71
NIFTY Energy	1 Day	0.76	0.17	0.1	0.93	0.95	0.45
	2 Day	0.98	0.44	0.11	0.98	0.44	0.69
	Week	0.77	0.89	0.13	0.49	0.49	0.52

1) NIFTY 50 and NIFTY Bank have a significant long-term impact on social and Technology features respectively.

2) NIFTY Auto also get impacted from social class but from a short-term perspective.

3) NIFTY Energy is affected by multiple PESTEL classes which include the short-term impact from technology and environment while technology and political also affect over two-day lag.

Above observation shows that there is a significant impact of news events over stock value, and the introduction of these features leads to better prediction of stock indices.

Table 2 Stock Index Prediction with PESTEL Events and previous Index Values

		SVM		**LSTM**	
Index	**No of Days**	**Accuracy**	**MCC**	**Accuracy**	**MCC**
India Volatility Index	1	61.96	0.169	61.96	0.154
	2	56.52	0.071	**67.39**	**0.304**
	5	58.24	0.099	**64.83**	**0.250**
NIFTY 50	1	53.26	0.014	**58.69**	**0.163**
	2	51.09	0.043	**52.17**	**0.098**
	5	52.75	0.035	**54.95**	**0.113**
NIFTY Bank	1	57.61	0.267	51.08	0.019
	2	51.09	0.01	**55.43**	**0.114**
	5	54.95	0.049	**56.05**	**0.070**
NIFTY Auto	1	53.26	0.063	50	0.103
	2	52.17	0.051	51.09	0.146
	5	58.24	0.165	51.65	0.105
NIFTY IT	1	55.43	0.097	**56.52**	**0.118**
	2	51.09	0.036	**54.35**	**0.071**

	5	52.75	0.032	**56.04**	**0.122**
NIFTY Energy	1	51.09	0.188	**58.69**	**0.118**
	2	54.35	0.093	**57.61**	**0.079**
	5	59.35	0.133	58.24	0.083

We have conducted experiments taking variable length sequences with sequence length been,1 day, 2 days and 1 week (5 days since the weekend is not counted). We have considered state of the art SVM with Radial Basis kernel as the baseline for our experiments. It can be observed that LSTM outperforms the SVM in most of the cases. We have used accuracy and Matthews correlation coefficient (MCC) as evaluation metrics'. Prediction results are shown in Table 3.

It is observed that in addition, looking at larger sequences improves the performance of LSTMs as compared to SVMs. It can be observed from Table 3 that for most of the indices LSTM predictions are outperforming SVMs by a margin of 2-11% in terms of accuracy.

VII. Conclusion

In this paper, we have presented a generic deep learning framework for predictive analytics using both structured and unstructured data. It is notable that the same architecture can easily handle both structured and unstructured sequential data. We have presented a case study which demonstrate the capability of the proposed framework.

We intend to extend this work in future to cover a wider range of factors along with a more fine-grained representation of generic events and also experiment with recent advancements in deep learning based prediction algorithms.

References

[1] Box, George; Jenkins, Gwilym M.; Reinsel, Gregory C. (1994). Time Series Analysis: Forecasting and Control (Third ed.). Prentice-Hall. ISBN 0130607746.

[2] Cox, DR (1958). "The regression analysis of binary sequences (with discussion)". J Roy Stat Soc B. 20: 215–242.

[3] Rokach, Lior; Maimon, O. (2008). Data mining with decision trees: theory and applications. World Scientific Pub Co Inc.

[4] Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". Psychological Review. 65 (6): 386–408.

[5] Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". Psychological Review. 65 (6): 386–408.

[6] Hinton, G. E.; Osindero, S.; Teh, Y. W. (2006). "A Fast Learning Algorithm for Deep Belief Nets" (PDF). Neural Computation. 18 (7): 1527–1554.

[7] Hinton, G. E.; Osindero, S.; Teh, Y. W. (2006). "A Fast Learning Algorithm for Deep Belief Nets" (PDF). Neural Computation. 18 (7): 1527–1554.

[8] LeCun, Yann. "LeNet-5, convolutional neural networks". Retrieved 16 November 2013.

[9] J. L. Elman. Finding structure in time. Cognitive science, 14(2):179–211, 1990.

[10] Xiaohui Yu, Yang Liu, Xiangji Huang and Aijun An: Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain, IEEE Transactions on Knowledge and Data Engineering, Volume: 24, Issue: 4, April 2012

[11] Samaneh Beheshti-Kashi, Hamid Reza Karimi, Klaus-Dieter Thoben, Michael Lütjen & Michael Teucke: A survey on retail sales forecasting and prediction in fashion markets, Systems Science & Control Engineering, Dec. 2014, Pages 154-161.

[12] Dare, C. (2006). The UK tour-operating industry: A competitive analysis. Journal of Vacation Marketing, 6(4), 357-367.

[13] Davydenko, A., & Fildes, R. (2013). Measuring forecasting accuracy: The case of judgmental adjustments to SKU-level demand forecasts. International Journal of Forecasting, 29(3), 510–522.

[14] Franses, P. H., & Legerstee, R. (2013). Do statistical forecasting models for SKU-level data benefit from including past expert knowledge? International Journal of Forecasting, 29(1), 80–87.

[15] Sun, Z.-L., Choi, T. M., Au, K-F., & Yu, Y.(2008). Sales forecasting using extreme learning machine with applications in fashion retailing. Decision Support Systems, 46(1), 411–419.

[16] Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. International Journal of Production Economics, 128(2), 470–483.

[17] Teucke, M., et al. (2014). Forecasting of seasonal apparel products. In H. Kotzab, J. Pannek, & K.-D. Thoben (eds.), Dynamics in Logistics. Fourth International Conference, LDIC 2014 Bremen, Germany, February 2014 Proceedings. Springer.

[18] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In ICLR Workshop Papers.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In NIPS, pages 3111–3119.

[21] Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In HLT NAACL.

[22] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. JMLR, 3:1137–1155.

[23] Le, Q. V., & Mikolov, T. (2014, June). Distributed Representations of Sentences and Documents. In ICML (Vol. 14, pp. 1188-1196).

[24] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10), 1550-1560

[25] Dare, C. (2006). The UK tour-operating industry: A competitive analysis. Journal of Vacation Marketing, 6(4), 357-367.

[26] Richardson Jr, J. V. (2006). The library and information economy in Turkmenistan. IFLA journal, 32(2), 131-139.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[28] J. R. Nofsinger, "Social Mood and Financial Economics," J. Behav. Finance, vol. 6, no. 3, pp. 144–160, Sep. 2005.

[29] W. S. Chan, "Stock price reaction to news and no-news: drift and reversal after headlines," J. Financ. Econ., vol. 70, no. 2, pp. 223–260, Nov. 2003.

[30] N. Strauß, R. Vliegenthart, and P. Verhoeven, "Lagging behind? Emotions in newspaper articles and stock market prices in the Netherlands," Public Relat. Rev.

[31] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grčar, and I. Mozetič, "The Effects of Twitter Sentiment on Stock Price Returns," PLOS ONE, vol. 10, no. 9, p. e0138441, Sep. 2015.

[32] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” J. Comput. Sci., vol. 2, no. 1, pp. 1–8, Mar. 2011.

[33] Luss, R., & d’Aspremont, A. (2015). Predicting abnormal returns from news using text classification. Quantitative Finance, 15(6), 999-1012.

[34] Ishan Verma, Lipika Dey, and Hardik Meisheri. 2017. Detecting, quantifying and accessing impact of news events on Indian stock indices. In Proceedings of the International Conference on Web Intelligence (WI '17). ACM, New York, NY, USA, 550-557.

State-of-the-Art and Trends in Nature-inspired Coordination Models

Stefano Mariani, *Member, IEEE,* and Andrea Omicini, *Senior Member, IEEE*

***Abstract*—Mostly stemming from closed parallel systems, coordination models and technologies gained in scope and expressive power so as to deal with complex distributed systems. In particular, in the last decade nature-inspired coordination (NIC) models emerged as the most effective approaches to tackle the complexity of pervasive, intelligent, and self-* systems. In this review paper we discuss their evolution, by analysing the main motivations behind the research effort on NIC, the foremost features of the most successful models, and the key issues and challenges they bring along.**

***Index Terms*—Nature-inspired coordination, Coordination models**

I. Introduction: Why NIC Matters

An essential source of complexity for computational systems is *interaction* [1]. According to the early intuition of Wegner [2], in fact (emphasis added):

> "Interaction is a *more powerful paradigm* than rule-based algorithms for computer-based solving, overtiring the prevailing view that all computing is expressible as algorithms."

Coordination, then, is [3]

> "the *glue* that binds separate activities into an ensemble."

or, more operationally, a *coordination model* is [4]

> "a framework in which the interaction of active and independent entities called agents can be expressed."

Tuple-based coordination, in particular, proved over the years to be the most expressive approach to coordination, mostly thanks to some peculiar traits emphasised by Gelernter in his seminal work on the Linda model [5]: *generative communication*, that is, the fact that information items (tuples) live independently of their producer; *associative access*, which means that information can be accessed by looking at their content, with no need to know their name or location; *suspensive semantics*, enabling synchronisation among conflicting activities by suspending and resuming operations based upon availability of data. Altogether, the above features lead to a concise and effective model for space, time, and reference *uncoupled* coordination in distributed systems of any sort.

Recognising interaction as an essential *dimension of computation* [2] impacts on the engineering of computational systems at many different levels:

- the need for brand new **programming languages** specifically devoted to program the *interaction space* arises [3]
- interaction is recognised as an independent design dimension in **software engineering** [6], with its own best practices and recurrent solutions—in the form of *coordination patterns* [7]
- interaction quickly becomes a new source of **artificial intelligence** [8]—for instance, of *social intelligence* [9]

Then, researchers observed that *natural systems* exhibit many features – such as distribution, openness, situatedness, robustness, adaptiveness – which are highly desirable for computational systems, too, and began to analyse them to understand their basic mechanisms. *Nature-inspired computing* soon became a hot research topic offering plenty of solutions to complex problems—see [10] for a short summary. In particular, the prominent role of interaction in the complexity of natural systems – as in the case, e.g., of *stigmergy* [11] – made *nature-inspired coordination* (NIC henceforth) a noteworthy subject of research for the last decades.

In the following sections, we first review some of the main proposals in the field from an historical perspective, discussing how NIC evolved from early models to future generation ones (Section II), then we look forward to the most recent research trends in the field, highlighting the challenges yet to be faced (Section III).

II. Evolution of NIC Models

Many different models have been proposed over time, drawing inspiration from disparate natural system depending on the desirable features to be extracted—i.e. chemistry, biology, ecosystems, physics, etc. In this section we review the most successful ones, highlighting their distinguishing features, and discussing the main issues involved in their engineering and deployment.

A. Early

The first NIC models to gain traction were *stigmergy-based* and *chemical-like*: the former explicitly aimed at coordinating an ensemble of autonomous agents, the latter originally aimed at providing an alternative model of computation, but later exploited and extended towards coordination needs.

1) Stigmergy: Most of NIC models are grounded in studies on the behaviour of social insects, like ants or termites. In fact, it was the zoologist Pierre-Paul Grassè to introduce the very notion of *stigmergy* as the fundamental coordination mechanism in termite societies [11]. There, termites release special chemical markers in the environment – called *pheromones* – to influence other termites activities: in this case, nest building.

S. Mariani is with the Department of Sciences and Methods of Engineering, University of Modena and Reggio Emilia, Italy.
E-mail: stefano.mariani@unimore.it

A. Omicini is with the Department of Computer Science and Engineering, University of Bologna, Italy.
E-mail: andrea.omicini@unibo.it

In this way, a form of indirect communication – called *environment-mediated* communication – favours coordination amongst distributed agents, and the coordination process itself is influenced by the structural properties of the environment: in fact, pheromones evaporates, is usually perceived locally – within some range –, and accumulates—insects perceive their "amount".

Stigmergy-based coordination has been then proficiently brought to the computational world by approaches such as *digital phermonones* [12], fostering digital "signs" (or markers) deposited in a shared environment [13] able to steer interacting agents activities—i.e. in the field of unmanned veichles.

2) Chemistry: Another early source of inspiration for NIC has been *chemistry*. The intuition here is that complex physical phenomena are driven by (relatively) simple chemical reactions, which to some extent "coordinate" the behaviours of a huge amount of components (molecules, for instance), as well as the global system (a cell, an organism) evolution.

Gamma [14] and CHAM (CHemical Abstract Machine) [15] are the earliest and most successful examples of this kind of NIC: the former is a novel approach to computation fostering *multiset rewriting* as the core processing mechanism, later specifically tailored to coordination in shared spaces, whereas the latter is an abstract computational model interpreting processes execution as a *chemical process*.

The two aforementioned models provide the basis for many later models and approaches to chemistry-inspired NIC, among which *biochemical tuple spaces* (Subsection II-B1) and MoK (Subsection III-B)

B. Modern

Based upon the early approaches just described, many models have been conceived as an extension, refinement, or combination of them, either as general-purpose coordination approaches or tailored to specific application domains. Also, thanks to the early success of the models described above, research in NIC further expanded to more heterogeneous sources of inspiration, there including, for instance, physics and swarming.

1) Biochemistry: Chemical tuple spaces [16] developed the Gamma and CHAM models to their full potential: data, devices, and software agents are represented in terms of chemical reactants, and system behaviour is expressed by means of chemical-like coordination rules; these rules are time-dependent and stochastic *exactly* as they are in natural chemistry. *Biochemical tuple spaces* (BST) [17] add a notion of *topology* and *distribution* to the picture, through the notion of compartments and diffusion.

The effectiveness and appeal of (bio)chemical coordination models is witnessed, for instance, by the SAPERE EU project [18], fostering a fully decentralised approach to coordination of pervasive systems deeply rooted in (and also hugely extending) the BTS model.

2) Field-based: Field-based coordination models like Co-fields [19] are inspired by the way masses and particles move and self-organise according to gravitational/electromagnetic fields. There, computational force fields propagate across the (computational) environment, and drive the actions and motion of the interacting agents.

TOTA [20], for instance, is a coordination middleware based on the co-fields model where interacting agents share tuples embedding a rules to autonomously spread in a network so as to create *computational gradients* used to coordinate agent actions and steer their activities towards a collective goal.

3) Swarms: Swarm intelligence [21] has a long tradition of models and algorithms drawing inspiration from *ecological systems* – most notably ant colonies, birds flocks, schools of fishes – to devise out efficient and fully decentralised cooperation/coordination mechanisms—mostly exploited in *swarm robotics* [22]. Along this line, SwarmLinda [23] proposes a tuple-based model for *swarming-based* coordination, where tuples and tuple templates are interpreted as food and artificial ants, respectively, and where the tuple-matching mechanism and tuples distribution in the network of tuple spaces are inspired to food harvesting and brood sorting, respectively.

Many applications in the general area of swarm robotics [24] exploit similar ideas—such as, for instance, cooperative transport [25].

C. Next Generation?

The more NIC becomes mature – and, with it, NIC models gets refined and stable – the more is likely that the original metaphor becomes less visible and somewhat mixed in with other approaches, either nature inspired or not, in order to optimise effectiveness and improve flexibility.

A notable and recent example is *aggregate computing* [26], which promotes a paradigm shift from programming devices to programming ensembles of devices, in a sort of spatio-temporal continuum. The aim is to simplify the design, creation, and maintenance of large-scale software systems [27] such as IoT, cyber-physical systems, pervasive computing, robotic swarms.

The roots of the model are in computational fields [19], chemical coordination [16], as well as *spatial computing* [28]—yet, all those sources of inspiration are blended together to create a very unique and novel programming paradigm.

D. Issues

Despite their heterogeneity, both as regards their source of inspiration and their actual design and implementation, all the models described above share a few critical issues to be dealt with so as to successfully and faithfully realise them:

- **environment** is essential in nature-inspired coordination
  - it works as a *mediator* for agent interaction, through which agents can communicate and coordinate indirectly
  - it is *active*, featuring autonomous dynamics, and affecting agent coordination
  - it has a *structure*, requiring a notion of *locality*, and allowing agents of any sort to move in a topology

For the reasons above, careful *environment engineering* [29] based on well-defined meta-models – such as the

A&A meta-model [30] – inevitably becomes a fundamental step in the software engineering process of a system exploiting NIC

- **probability** is a core mechanism for complex systems
  - randomness without any well-defined probabilistic model (distribution) is *not expressive* enough to capture all the properties of complex systems such as biochemical and social systems
  - probabilistic mechanisms are thus required to enable (possibly simple yet) *stochastic behaviours*

  For the reasons above, NIC primitives should feature some probabilistic semantics, as in the case of *uniform primitives* [31]

It is worth emphasising here that the above mentioned features are issues for NIC in that they represent crucial aspects that requires proper consideration when designing NIC models, but at the same time they are *opportunities* for NIC, as they potentially enable further (and richer) forms of expressiveness.

III. Outlook on Research Trends

In the following section we discuss three research areas in which NIC models either have already shown early promising results, or they are currently under scrutiny by researchers as a promising source of solutions.

A. Simulation

Simulation of complex systems is a multidisciplinary issue ranging from physics to biology, from economics to social sciences. Virtually, no complex system of any sort can be studied nowadays without the support of suitable simulation tools; and, experiments done *in silico* are at least as relevant as those *in vitro* and *in vivo*. Given that interaction is one of the foremost sources of complexity, simulation increasingly amounts to simulating interactions. As a result, simulation platforms and tools are devoting more and more attention and resources to modelling and simulating the coordination rules governing the interaction space of applications.

Therefore, a few research works started considering the option of building simulation frameworks on top of coordination languages and infrastructures, so as to take advantage of their ability to deal with complex interactions elegantly and effectively. For instance, in [32] biochemical tuple spaces [17] are exploited as the core abstraction upon which a simulation tool for simulating intracellular signalling pathways is built [33].

There, the extracellular milieu and intracellular compartments are mapped onto special tuple spaces *programmed* so as to work as *chemical solutions simulators* [34], signalling components such as membrane receptors, proteins, enzymes, and genes map to chemical reactions sets expressed as tuples, signalling molecules, activation, and deactivation signals are represented as reactants and concentrations recorded as tuples in the tuple space.

B. Knowledge-oriented Coordination

Intelligent MAS in *knowledge-intensive environments* (KIE) – as well as complex socio-technical systems, in general – require automatic understanding of data and information [35]. *Knowledge-oriented coordination* exploits coordination abstractions enriched so as to allow for semantic interpretation by intelligent agents [36], [37]. For instance, SAPERE [18] coordination abstractions and *semantic tuple centres* [38] both rely on the semantic interpretation of coordination items.

In KIE scenarios explicit search of information is going to become ineffective while the amount of available knowledge grows at incredible rates, thus knowledge should autonomously organise and flow from producers to consumers, in a sort of *knowledge self-organisation* process. MoK (Molecules of Knowledge [39]) is a nature-inspired coordination model promoting knowledge self-organisation, where sources of knowledge continuously produce and inject *atoms of knowledge* in artificial biochemical compartments (analogously to biochemical tuple spaces), knowledge atoms may aggregate in *molecules* and diffuse, knowledge producers, managers, and consumers are modelled as *catalysts*, whose workspaces are biochemical compartments, and their knowledge-oriented actions become enzymes influencing atoms aggregation and molecules diffusion. All of this so as to make relevant knowledge spontaneously aggregate and autonomously move towards potentially interested knowledge workers.

C. Complex Systems

Simon argues that [40]

> “by a complex system I mean one made up of a large number of parts that *interact* in a non simple way.”

Some “laws of complexity” exist that characterise any complex system, independently of its specific nature [41]: however, the precise source of what all complex systems share is still in some way unknown in essence. We argue that *interaction* – its *nature*, *structure*, *dynamics* – is the key to understand some fundamental properties of complex systems of any kind.

The above considerations are apparent, i.e., in the field of *statistical mechanics*, where introducing interaction among particles structurally changes the macroscopic properties of the system, along with the mathematical ones. In fact, interacting systems in statistical mechanics are systems where particles do not behave independently of each other, thus the probability distribution of an interacting system does not factorise anymore.

In computer science terms, an interacting system is *non-compositional* [2].

1) Sociotechnical Systems: Nowadays, a particularly-relevant class of complex systems is represented by *socio-technical systems* (STS) [42]. There, active components are mainly represented by *humans*, yet interaction is almost-totally regulated by the *software infrastructure*, where *software agents* often play a key role. Examples of such a kind of systems are social platforms such as Facebook [43] and LiquidFeedback [44], but also urban transportation networks, the ICT infrastructure supporting rescue operations, e-government platforms enabling citizens to participate in local administrations' decision making, and, essentially, any kind of CSCW system [45].

It has already been recognised that such a sort of systems may look at nature seeking for solutions [46], mostly because two foremost characteristics provide opportunities for successfully applying NIC: **unpredictability** of human behaviour should be accounted for, thus uncertainty of actions' outcomes and of decision making should be taken as the norm, not an exceptional condition; the fact that an **environment** – either computational such as in the case of CSCW platforms, or physical, as for urban traffic management – exists not because engineers designed it, but because it is an essential part of the application domain. Accordingly, NIC already accounts for the conceptual and technical tools to deal with both: probabilistic coordination mechanisms and environment modelling.

2) Cyberphysical Systems: Cyberphysical systems (CPS) integrate computing and communication technologies with monitoring and control of physical devices [47]. Examples of CPS include power grids, medical devices, manufacturing control systems, etc.

The centrality of a suitable and effective approach to coordination in such a sort of systems has been already recognised [48], and mostly stems from the need to ensure some crucial features in face of distribution and uncertainty of real-world deployments: dependability, reliability, efficiency—to mention a few. Also, the opportunity to resort to NIC has already been considered [49], [50].

3) The Internet of Things: The Internet of Things (IoT) vision lies somewhat at the crossroad between CPS and STS: whereas is true that strictly speaking the IoT deals primarily with interconnecting devices, it is also true that IoT platforms are in their very essence CPS where the devices and the software running in them are mostly indistinguishable, and that IoT devices are to be used and monitored by human users, exploiting them to augment their capabilities. It is thus possible to apply in IoT scenarios the same approaches we mentioned in the previous sections.

Nevertheless, the peculiarities of the IoT application domain allows for developing ad-hoc models and for undertaking specific approaches to NIC. In [51], for instance, the authors take inspiration from natural metaphors to propose a decentralised service composition model based on *artificial potential fields* (APFs). APFs are digital counterparts of gravitational and magnetic potential fields which can be found in the physical world, and are exploited to lead the service composition process through the balance of forces applied between service requests and service nodes. The applicability of the proposed approach is discussed in the context of dynamic and personalised composition of an audio-visual virtual guide service in an IoT network of a trade show venue.

D. Challenges

Many technical challenges are ahead for those who intend to advance the state-of-art in NIC. Instead of just listing them all, in the following we aim at discussing the two main conceptual challenges that we believe are fundamental to drive research on the topic in a focussed and pragmatic way:

- understanding the basic elements of **expressiveness** is crucial to determine to what extent NIC can cope with real-world problems, by understanding the *minimal set* of coordination primitives required to design complex stochastic behaviours. For instance, *uniform coordination primitives* – that is, LINDA-like coordination primitives returning tuples matching a template with a uniform distribution [52] – seemingly capture the full-fledged dynamics of real chemical and biological systems within the coordination abstractions
- engineering unpredictable systems around **predictable** abstractions is fundamental to ensure the predictability of given MAS properties within generally-unpredictable MAS. In fact, since coordination abstractions are often at the core of complex MAS, making the coordinative behaviour predictable makes it possible in principle to make the overall system *partially predictable*.

We believe in fact that only through a deep understanding of how the core mechanisms of NIC influence system evolution research on NIC will enable engineers to consistently design and build predictable yet stochastic systems.

IV. CONCLUSION

Gathering ideas and results from the many research fields dealing with complexity emphasises the central role of *interaction*. Since *coordination models* are meant to provide the conceptual framework to express interaction in parallel, concurrent, distributed systems, they are fundamental in order to deal with complexity in computational systems.

In the last decades *nature-inspired coordination models* worked as powerful sources of inspiration for abstractions and mechanisms aimed at harnessing complexity in distributed, pervasive, intelligent systems. In particular, nowadays application scenarios – such as knowledge-intensive environments, socio-technical systems, and the Internet of Things – are going to propose novel noteworthy challenges that are likely to push research on NIC models to its limits and beyond.

ACKNOWLEDGMENT

The authors would like to thank Prof. Xin Li, the organisers of the 2017 IEEE/WIC/ACM International Conference on Web Intelligence, and the attendants of the tutorial therein delivered from which this paper stems.

REFERENCES

[1] D. Q. Goldin, S. A. Smolka, and P. Wegner, Eds., *Interactive Computation: The New Paradigm*. Springer, Sep. 2006. [Online]. Available: http://www.springerlink.com/content/g3h215/

[2] P. Wegner, "Why interaction is more powerful than algorithms," *Communications of the ACM*, vol. 40, no. 5, pp. 80–91, May 1997. [Online]. Available: http://portal.acm.org/citation.cfm?doid=253769.253801

[3] D. Gelernter and N. Carriero, "Coordination languages and their significance," *Communications of the ACM*, vol. 35, no. 2, pp. 97–107, 1992. [Online]. Available: http://portal.acm.org/cfm?id=129635

[4] P. Ciancarini, "Coordination models and languages as software integrators," *ACM Computing Surveys*, vol. 28, no. 2, pp. 300–302, Jun. 1996. [Online]. Available: http://portal.acm.org/citation.cfm?id=234732

[5] D. Gelernter, "Generative communication in Linda," *ACM Transactions on Programming Languages and Systems*, vol. 7, no. 1, pp. 80–112, Jan. 1985. [Online]. Available: http://portal.acm.org/citation.cfm?id=2433

[6] P. Ciancarini, A. Omicini, and F. Zambonelli, “Multiagent system engineering: The coordination viewpoint,” in *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, ser. LNAI, N. R. Jennings and Y. Lespérance, Eds. Springer, 2000, vol. 1757, pp. 250–259, 6th International Workshop (ATAL’99), Orlando, FL, USA, 15–17 Jul. 1999. Proceedings. [Online]. Available: http://www.springerlink.com/content/703527n23t106302/

[7] D. Deugo, M. Weiss, and E. Kendall, “Reusable patterns for agent coordination,” in *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, Eds. Springer, Mar. 2001, ch. 14, pp. 347–368. [Online]. Available: http://link.springer.com/10.1007/978-3-662-04401-8_14

[8] A. Omicini and G. A. Papadopoulos, “Editorial: Why coordination models and languages in AI?” *Applied Artificial Intelligence: An International Journal*, vol. 15, no. 1, pp. 1–10, Jan. 2001, special Issue: Coordination Models and Languages in AI. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/08839510150204581

[9] C. Castelfranchi, G. Pezzullo, and L. Tummolini, “Behavioral implicit communication (BIC): Communicating with smart environments via our practical behavior and its traces,” *International Journal of Ambient Computing and Intelligence*, vol. 2, no. 1, pp. 1–12, Jan.–Mar. 2010. [Online]. Available: http://www.igi-global.com/bookstore/article.aspx?titleid=40346

[10] J. Liu and K. C. Tsui, “Toward nature-inspired computing,” *Communications of the ACM*, vol. 49, no. 10, pp. 59–64, Oct. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?doid=1164395

[11] P.-P. Grassé, “La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: Essai d’interprétation du comportement des termites constructeurs,” *Insectes Sociaux*, vol. 6, no. 1, pp. 41–80, Mar. 1959.

[12] H. V. D. Parunak, S. Brueckner, and J. Sauter, “Digital pheromone mechanisms for coordination of unmanned vehicles,” in *1st International Joint Conference on Autonomous Agents and Multiagent systems*, C. Castelfranchi and W. L. Johnson, Eds., vol. 1. New York, NY, USA: ACM, 15–19Jul. 2002, pp. 449–450. [Online]. Available: http://dl.acm.org/citation.cfm?doid=544741.544843

[13] H. V. D. Parunak, “A survey of environments and mechanisms for human-human stigmergy,” in *Environments for Multi-Agent Systems II*, ser. LNCS, D. Weyns, H. V. D. Parunak, and F. Michel, Eds. Springer, 2006, vol. 3830, pp. 163–186.

[14] J.-P. Banâtre and D. Le Métayer, “The GAMMA model and its discipline of programming,” *Science of Computer Programming*, vol. 15, no. 1, pp. 55–77, Nov. 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016764239090044E

[15] G. Berry, “The chemical abstract machine,” *Theoretical Computer Science*, vol. 96, no. 1, pp. 217–248, Apr. 1992.

[16] M. Viroli, M. Casadei, E. Nardini, and A. Omicini, “Towards a chemical-inspired infrastructure for self-* pervasive applications,” in *Self-Organizing Architectures*, ser. LNCS, D. Weyns, S. Malek, R. de Lemos, and J. Andersson, Eds. Springer, Jul. 2010, vol. 6090, ch. 8, pp. 152–176, 1st International Workshop on Self-Organizing Architectures (SOAR 2009), Cambridge, UK, 14-17 Sep. 2009, Revised Selected and Invited Papers.

[17] M. Viroli and M. Casadei, “Biochemical tuple spaces for self-organising coordination,” in *Coordination Languages and Models*, ser. LNCS, J. Field and V. T. Vasconcelos, Eds. Lisbon, Portugal: Springer, Jun. 2009, vol. 5521, pp. 143–162, 11th International Conference (COORDINATION 2009), Lisbon, Portugal, Jun. 2009. Proceedings. [Online]. Available: http://www.springerlink.com/content/jx783250031742uu/

[18] F. Zambonelli, A. Omicini, B. Anzengruber, G. Castelli, F. L. DeAngelis, G. Di Marzo Serugendo, S. Dobson, J. L. Fernandez-Marquez, A. Ferscha, M. Mamei, S. Mariani, A. Molesini, S. Montagna, J. Nieminen, D. Pianini, M. Risoldi, A. Rosi, G. Stevenson, M. Viroli, and J. Ye, “Developing pervasive multi-agent systems with nature-inspired coordination,” *Pervasive and Mobile Computing*, vol. 17, pp. 236–252, Feb. 2015, special Issue “10 years of Pervasive Computing” In Honor of Chatschik Bisdikian. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214001904

[19] M. Mamei and F. Zambonelli, *Field-Based Coordination for Pervasive Multiagent Systems. Models, Technologies, and Applications*, ser. Springer Series in Agent Technology. Springer, Mar. 2006. [Online]. Available: http://www.springerlink.com/content/j51v18/

[20] ——, “Programming pervasive and mobile computing applications with the TOTA middleware,” in *Pervasive Computing and Communications*, 2004, pp. 263–273, 2nd IEEE Annual Conference (PerCom 2004), Orlando, FL, USA, 14–17 Mar. 2004. Proceedings. [Online]. Available: http://www.computer.org/portal/web/csdl/doi/10.1109/PERCOM.2004.1276864

[21] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, ser. Santa Fe Institute Studies in the Sciences of Complexity. 198 Madison Avenue, New York, New York 10016, United States of America: Oxford University Press, 1999.

[22] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: a review from the swarm engineering perspective,” *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, Mar 2013. [Online]. Available: https://doi.org/10.1007/s11721-012-0075-2

[23] R. Tolksdorf and R. Menezes, “Using Swarm Intelligence in Linda Systems,” in *Engineering Societies in the Agents World IV*, ser. LNCS, A. Omicini, P. Petta, and J. Pitt, Eds. Springer, Jun. 2004, vol. 3071, pp. 49–65, 4th International Workshops (ESAW 2003), London, UK, 29-31 Oct. 2003. Revised Selected and Invited Papers.

[24] E. Şahin, “Swarm robotics: From sources of inspiration to domains of application,” in *Swarm Robotics*, E. Şahin and W. M. Spears, Eds. Springer, 2005, pp. 10–20, sAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers. [Online]. Available: http://link.springer.com/10.1007/978-3-540-30552-1_2

[25] C. R. Kube and E. Bonabeau, “Cooperative transport by ants and robots,” *Robotics and Autonomous Systems*, vol. 30, no. 1, pp. 85 – 101, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889099000664

[26] J. Beal, S. Dulman, K. Usbeck, M. Viroli, and N. Correll, “Organizing the aggregate: Languages for spatial computing,” in *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments*, M. Mernik, Ed. IGI Global, 2013, ch. 16, pp. 436–501, a longer version available at: `http://arxiv.org/abs/1202.5509`. [Online]. Available: http://www.igi-global.com/chapter/organizing-aggregate-languages-spatial-computing/71829

[27] J. Beal, D. Pianini, and M. Viroli, “Aggregate programming for the internet of things,” *Computer*, vol. 48, no. 9, pp. 22–30, Sept 2015.

[28] M. Viroli, M. Casadei, S. Montagna, and F. Zambonelli, “Spatial coordination of pervasive services through chemical-inspired tuple spaces,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 6, no. 2, pp. 14:1–14:24, June 2011. [Online]. Available: http://doi.acm.org/10.1145/1968513.1968517

[29] M. Viroli, A. Omicini, and A. Ricci, “Engineering MAS environment with artifacts,” in *2nd International Workshop “Environments for Multi-Agent Systems” (E4MAS 2005)*, D. Weyns, H. V. D. Parunak, and F. Michel, Eds., AAMAS 2005, Utrecht, The Netherlands, 26 Jul. 2005, pp. 62–77.

[30] A. Omicini, A. Ricci, and M. Viroli, “Artifacts in the A&A meta-model for multi-agent systems,” *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 3, pp. 432–456, Dec. 2008, special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems. [Online]. Available: http://www.springerlink.com/content/l2051h377k2plk07/

[31] S. Mariani and A. Omicini, “Coordination mechanisms for the modelling and simulation of stochastic systems: The case of uniform primitives,” *SCS M&S Magazine*, 2014.

[32] P. P. González Pérez, A. Omicini, and M. Sbaraglia, “A biochemically-inspired coordination-based model for simulating intracellular signalling pathways,” *Journal of Simulation*, vol. 7, no. 3, pp. 216–226, Aug. 2013, special Issue: Agent-based Modeling and Simulation. [Online]. Available: http://www.palgrave-journals.com/jos/journal/v7/n3/full/jos201228a.html

[33] J. Downward, “Targeting RAS signalling pathways in cancer therapy,” *Nature Reviews Cancer*, vol. 3, no. 1, pp. 11–22, 2003. [Online]. Available: http://www.nature.com/nrc/journal/v3/n1/full/nrc969.html

[34] D. T. Gillespie, “Exact stochastic simulation of coupled chemical reactions,” *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, Dec. 1977. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/j100540a008

[35] A. Omicini, “Self-organising knowledge-intensive workspaces,” in *Pervasive Adaptation. The Next Generation Pervasive Computing Research Agenda*, A. Ferscha, Ed. Austria: Institute for Pervasive Computing, Johannes Kepler University Linz, May 2011, ch. VII: Human-Centric Adaptation, pp. 71–72. [Online]. Available: https://www.pervasive.jku.at/fet11/RAB.pdf

[36] D. Fensel, “Triple-space computing: Semantic web services based on persistent publication of information,” in *Intelligence in Communication Systems*, ser. LNCS, F. A. Aagesen, C. Anutariya, and V. Wuwongse, Eds., vol. 3283, 2004, pp. 43–53, IFIP International Conference (INTELLCOMM 2004), Bangkok, Thailand, 23–26 Nov. 2004. Proceedings.

[37] E. Nardini, A. Omicini, and M. Viroli, “Semantic tuple centres,” *Science of Computer Programming*, vol. 78, no. 5, pp. 569–582, May 2013, special section: Self-Organizing Coordination. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167642312001876

[38] E. Nardini, A. Omicini, M. Viroli, and M. I. Schumacher, “Coordinating e-health systems with TuCSoN semantic tuple centres,” *Applied Computing Review*, vol. 11, no. 2, pp. 43–52, Spring 2011.

[39] S. Mariani, *Coordination of Complex Sociotechnical Systems: Self-organisation of Knowledge in MoK*, 1st ed., ser. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer International Publishing, Dec. 2016. [Online]. Available: http://link.springer.com/10.1007/978-3-319-47109-9

[40] H. A. Simon, “The architecture of complexity,” *Proceedings of the American Philosophical Society*, vol. 106, no. 6, pp. 467–482, 12 Dec. 1962. [Online]. Available: http://www.jstor.org/stable/985254

[41] S. A. Kauffman, *Investigations*. Oxford University Press, Jan. 2003. [Online]. Available: http://ukcatalogue.oup.com/product/9780195121056.do

[42] B. Whitworth, “Socio-technical systems,” in *Encyclopedia of Human Computer Interaction*, C. Ghaou, Ed. IGI Global, 2006, pp. 533–541. [Online]. Available: http://www.igi-global.com/chapter/social-technical-systems/13170

[43] FaceBook, “Home page,” `http://www.facebook.com`, FaceBook, Inc., Menlo Park, CA, USA, 2014. [Online]. Available: http://www.facebook.com

[44] LiquidFeedback, “Home page,” `http://liquidfeedback.org`, Public Software Group, Berlin, Germany, 2014. [Online]. Available: http://liquidfeedback.org

[45] J. Grudin, “Computer-supported cooperative work: history and focus,” *Computer*, vol. 27, no. 5, pp. 19–26, May 1994.

[46] F. Zambonelli, A. Omicini, and P. Scerri, “Coordination in large-scale socio-technical systems: Introduction to the special section,” *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 1, pp. 5–8, Jan 2016.

[47] E. A. Lee, “Cps foundations,” in *Design Automation Conference*, June 2010, pp. 737–742.

[48] ——, “Cyber physical systems: Design challenges,” in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, May 2008, pp. 363–369.

[49] L. Wang, M. Törngren, and M. Onori, “Current status and advancement of cyber-physical systems in manufacturing,” *Journal of Manufacturing Systems*, vol. 37, no. Part 2, pp. 517 – 527, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0278612515000400

[50] S. Senge and H. F. Wedde, “Bee-inpired road traffic control as an example of swarm intelligence in cyber-physical systems,” in *2012 38th Euromicro Conference on Software Engineering and Advanced Applications*, Sept 2012, pp. 258–265.

[51] E. Rapti, A. Karageorgos, and V. C. Gerogiannis, “Decentralised service composition using potential fields in internet of things applications,” *Procedia Computer Science*, vol. 52, no. Supplement C, pp. 700 – 706, 2015, the 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915008790

[52] L. Gardelli, M. Viroli, M. Casadei, and A. Omicini, “Designing self-organising MAS environments: The collective sort case,” in *Environments for MultiAgent Systems III*, ser. LNAI, D. Weyns, H. V. D. Parunak, and F. Michel, Eds. Springer, May 2007, vol. 4389, pp. 254–271, 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers. [Online]. Available: http://www.springerlink.com/content/02441711k44v8224/

RELATED CONFERENCES, CALL FOR PAPERS/PARTICIPANTS

TCII Sponsored Conferences

WI 2018
The 2018 IEEE/WIC/ACM International Conference on Web Intelligence
Chile
December, 2018

After the successful conference in Leipzig, Germany this year, the 2018 IEEE/WIC/ACM International Conference on Web Intelligence will take place in Chile, the first week of December, 2018.

Web Intelligence (WI) aims to achieve a multi-disciplinary balance between research advances in theories and methods usually associated with Collective Intelligence, Data Science, Human-Centric Computing, Knowledge Management, and Network Science. It is committed to addressing research that both deepen the understanding of computational, logical, cognitive, physical, and social foundations of the future Web, and enable the development and application of technologies based on Web intelligence.

Web Intelligence focuses on scientific research and applications by jointly using Artificial Intelligence (AI) (e.g., knowledge representation, planning, knowledge discovery and data mining, intelligent agents, and social network intelligence) and advanced Information Technology (IT) (e.g., wireless networks, ubiquitous devices, social networks, semantic Web, wisdom Web, and data/knowledge grids) for the next generation of Web-empowered products, systems, services, and activities.

WI'18 welcomes both research and application papers submissions. All submitted papers will be reviewed on the basis of technical quality, relevance, significance and clarity. Accepted full papers will be included in the proceedings published by the ACM Press.

ICDM 2018
IEEE International Conference on Data Mining
Singapore
November 17-20, 2018
http://icdm2018.org/

The IEEE International Conference on Data Mining series (ICDM) has established itself as the world's premier research conference in data mining. It provides an international forum for presentation of original research results, as well as exchange and dissemination of innovative, practical development experiences. The conference covers all aspects of data mining, including algorithms, software and systems, and applications. ICDM draws researchers and application developers from a wide range of data mining related areas such as statistics, machine learning, pattern recognition, databases and data warehousing, data visualization, knowledge-based systems, and high performance computing. By promoting novel, high quality research findings, and innovative solutions to challenging data mining problems, the conference seeks to continuously advance the state-of-the-art in data mining. Besides the technical program, the conference features workshops, tutorials, panels.

ICHI 2018
The Sixth IEEE International Conference on Healthcare Informatics
New York, NY, USA
June 4-7, 2018
http://hpr.weill.cornell.edu/divisions/health_informatics/ieee_ichi.html

The Sixth IEEE International Conference on Healthcare Informatics (ICHI'18) will take place in New York City from June 4th to June 7th, 2018 at the Doubletree by Hilton Metropolitan - New York City.

The ICHI series is the premier community forum concerned with the application of computer science principles, information science principles, information technology, and communication technology to address problems in healthcare, public health, and everyday wellness. The conference highlights the most novel technical contributions in computing-oriented health informatics and the related social and ethical implications.

ICHI'18 serves as a venue for discussion of innovative technical and empirical contributions, highlighting end-to-end applications, systems, and technologies, even if available only in prototype form. ICHI'18 will feature keynotes, a multi-track technical program including papers, demonstrations, panels, workshop, tutorials, industrial tracks, and a doctoral consortium.

BigData 2018
The IEEE 2018 7th International Congress on Big Data (BigData Congress 2018)
San Francisco, USA
July 2-7, 2018
http://conferences.computer.org/bigdatacongress/2018/

As cloud computing turning computing and software into commodity services, everything as a service in other words, it leads to not only a technology revolution but also a business revolution. Insights and impacts of various types of services (infrastructure as a service, platform as a service, software as a service, business process as a service) have to be re-examined.

2018 International Congress on Big Data (BigData Congress 2018) aims to provide an international forum that formally explores various business insights of all kinds of value-added "services." Big Data is a key enabler of exploring business insights and economics of services.

BigData Congress 2018's major topics include but not limited to: Big Data Architecture, Big Data Modeling, Big Data As A Service, Big Data for Vertical Industries (Government, Healthcare, etc.), Big Data Analytics, Big Data Toolkits, Big Data Open Platforms, Economic Analysis, Big Data for Enterprise Transformation, Big Data in Business Performance Management, Big Data for Business Model Innovations and Analytics, Big Data in Enterprise Management Models and Practices, Big Data in Government Management Models and Practices, and Big Data in Smart Planet Solutions.

Related Conferences

AAMAS 2018
The 17th International Conference on Autonomous Agents and Multi-Agent Systems

Stockholm, Sweden
July 10-15, 2018
http://aamas18.ifaamas.org

AAMAS (International Conference on Autonomous Agents and Multiagent Systems) is the largest and most influential conference in the area of agents and multiagent systems. The aim of the conference is to bring together researchers and practitioners in all areas of agent technology and to provide a single, high-profile, internationally renowned forum for research in the theory and practice of autonomous agents and multiagent systems. AAMAS is the flagship conference of the non-profit International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

AAMAS'18, the 17th edition of the conference, will be held on July 10-15, in Stockholm, Sweden and is part of the Federated AI Meeting (FAIM), with the other conferences being IJCAI, ICML, ICCBR and SoCS. Please see the preliminary schedule for more information.

The AAMAS conference series was initiated in 2002 in Bologna, Italy as a joint event comprising the 6th International Conference on Autonomous Agents (AA), the 5th International Conference on Multiagent Systems (ICMAS), and the 9th International Workshop on Agent Theories, Architectures, and Languages (ATAL).

Subsequent AAMAS conferences have been held in Melbourne, Australia (July 2003), New York City, NY, USA (July 2004), Utrecht, The Netherlands (July 2005), Hakodate, Japan (May 2006), Honolulu, Hawaii, USA (May 2007), Estoril, Portugal (May 2008), Budapest, Hungary (May 2009), Toronto, Canada (May 2010), Taipei, Taiwan (May 2011), Valencia, Spain (June 2012), Minnesota, USA (May 2013), Paris, France (May 2014), Istanbul, Turkey (May 2015), Singapore (May 2016) and São Paulo (**2017**).

____

AAAI 2018
The 32th AAAI Conference on Artificial Intelligence

New Orleans, Lousiana, USA
February 2–7, 2018
https://aaai.org/Conferences/AAAI-18/

The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18) will be held February 2–7, 2018 at the Hilton New Orleans Riverside, New Orleans, Lousiana, USA. The program chairs will be Sheila McIlraith, University of Toronto, Canada and Kilian Weinberger, Cornell University, USA.

The purpose of the AAAI conference is to promote research in artificial intelligence (AI) and scientific exchange among AI researchers, practitioners, scientists, and engineers in affiliated disciplines. AAAI'18 will have a diverse technical track, student abstracts, poster sessions, invited speakers, tutorials, workshops, and exhibit and competition programs, all selected according to the highest reviewing standards. AAAI'18 welcomes submissions on mainstream AI topics as well as novel crosscutting work in related areas.

____

SDM 2018
The 2018 SIAM International Conference on Data Mining

San Diego, California, USA
May 3 - 5, 2018
http://www.siam.org/meetings/sdm18/

Data mining is the computational process for discovering valuable knowledge from data – the core of modern Data Science. It has enormous applications in numerous fields, including science, engineering, healthcare, business, and medicine. Typical datasets in these fields are large, complex, and often noisy. Extracting knowledge from these datasets requires the use of sophisticated, high-performance, and principled analysis techniques and algorithms. These techniques in turn require implementations on high performance computational infrastructure that are carefully tuned for performance. Powerful visualization technologies along with effective user interfaces are also essential to make data mining tools appealing to researchers, analysts, data scientists and application developers from different disciplines, as well as usable by stakeholders.

SDM has established itself as a leading conference in the field of data mining and provides a venue for researchers who are addressing these problems to present their work in a peer-reviewed forum. SDM emphasizes principled methods with solid mathematical foundation, is known for its high-quality and high-impact technical papers, and offers a strong workshop and tutorial program (which are included in the conference registration). The proceedings of the conference are published in archival form, and are also made available on the SIAM web site.

____

IJCAI 2018
The 27th International Joint Conference on Artificial Intelligence

Stockholm, Sweden
July 13-19, 2018
http://www.ijcai-18.org/

International Joint Conferences on Artificial Intelligence is a non-profit corporation founded in California, in 1969 for scientific and educational purposes, including dissemination of information on Artificial Intelligence at conferences in which cutting-edge scientific results are presented and through dissemination of materials presented at these meetings in form of Proceedings, books, video recordings, and other educational materials. IJCAI consists of two divisions: the Conference Division and the AI Journal Division. IJCAI conferences present premier international gatherings of AI researchers and practitioners and they were held biennially in odd-numbered years since 1969.

IJCAI-ECAI'18 is part of the Federated AI Meeting (FAIM) that takes place in Stockholm July 9-19. The other conferences include AAMAS, ICML, ICCBR and SoCS.

The IJCAI-ECAI'18 registration is expected to open in April 2018.

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903

Non-profit Org.
U.S. Postage
PAID
Silver Spring, MD
Permit 1398