Studies in Computational Intelligence 607

Yacine Laalaoui
Nizar Bouguila *Editors*

Artificial Intelligence Applications in Information and Communication Technologies

Springer

Studies in Computational Intelligence

Volume 607

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Yacine Laalaoui · Nizar Bouguila
Editors

Artificial Intelligence Applications in Information and Communication Technologies

Springer

Editors
Yacine Laalaoui
Taif University
Taif
Saudi Arabia

Nizar Bouguila
Faculty of Engineering and Computer Science
Concordia Institute for Information Systems Engineering, Concordia University
Montreal, QC
Canada

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-319-19832-3 ISBN 978-3-319-19833-0 (eBook)
DOI 10.1007/978-3-319-19833-0

Library of Congress Control Number: 2015942218

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Contents

Part III Ontologies and Multi-agents

Contributors

Jehad Al-Omari Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

Artur Arsenio YDreams Robotics, Universidade da Beira Interior, Covilhã, Portugal

Taoufik Bdiri Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada

Nizar Bouguila Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

Abdelaziz Bouras Computer Science and Engineering Department, College of Engineering, Qatar University, Doha, Qatar

Yassine Drias USTHB, Algeria, Africa

Samir Kechid USTHB, Algeria, Africa

Dalibor Klusáček Faculty of Informatics, Masaryk University, Brno, Czech Republic

Yacine Laalaoui Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

Virgilio López-Morales CITIS, Universidad Autónoma del Estado de Hidalgo, Pachuca, Mexico

J. López Department of Electronics and Communications Technologies, Autonomous University of Madrid (UAM), Madrid, Spain

Thitiya Manakitsirisuthi DISP-Laboratory, Université Lumière Lyon 2, Bron, France

A. Martinez Networking and Information Technology Lab (NetIT Lab), Technical University of Catalonia (UPC), Barcelona, Spain

Mohamed Al Mashrgy Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada

Hedi Mhalla Department of Mathematics and Statistics, The American University of the Middle East, Eqaila, Kuwait

Yacine Ouzrout DISP-Laboratory, Université Lumière Lyon 2, Bron, France

W. Ramirez Advanced Network Architectures Lab (CRAAX), Technical University of Catalonia (UPC), Barcelona, Spain

Hana Rudová Faculty of Informatics, Masaryk University, Brno, Czech Republic

R. Serral-Gracià Networking and Information Technology Lab (NetIT Lab), Technical University of Catalonia (UPC), Barcelona, Spain

Parisa Tirdad Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

M. Yannuzzi Networking and Information Technology Lab (NetIT Lab), Technical University of Catalonia (UPC), Barcelona, Spain

Djemel Ziou DI, Faculté des Sciences, Université de Sherbrooke, Sherbrooke, QC, Canada

Introduction

Information and Communication Technologies (ICT) is one of the important and climbing sectors in industry. It is part of the computing area and includes the human factor. Humans act as ICT managers, system administrators, network administrators and security experts. ICT challenges include user satisfaction, good performance insurance, security issues handling, etc. Great challenges are yet to come with increase of the Internet and mobile access devices. Often, ICT practitioners do not have sufficient Artificial Intelligence (AI) background to deal with real-life problems. The main aim of this book is to help ICT practitioners in managing efficiently their platforms using AI tools and methods, and to bring up what can AI provides to the ICT sector.

This book includes many AI topics namely: search and optimization methods, machine learning, data representation and Ontologies, and multi-agent systems. In "A Metaheuristic for Optimizing the Performance and the Fairness in Job Scheduling Systems", Klusacek and Rudova present a meta-heuristic-based approach to periodically optimize job scheduling plan in large computational systems such as HPC clusters and Grids. The proposed approach is time efficient extension of Conservative backfilling. In fact, this approach aims at minimizing the bad effects of inaccurate run time estimates while improving performance and guaranteeing good fairness with respect to different users of the system. In "Hybrid ACO and Tabu Search for Large Scale Information Retrieval", Drias and Kechid use meta-heuristics to tackle the problem of information retrieval (IR) on large-scale web data sets. The authors use a hybridization of two meta-heuristics, namely Ant Colony Optimization and Tabu Search. In "Hosting Clients in Clustered and Virtualized environment: A Combinatorial Optimization Approach", Laalaoui et al. propose an approach to address the problem of hosting clients (operations systems and applications) on clustered and virtualized environments. The authors describe an integer mathematical model of the problem and use Branch-and-Bound Search to solve the problem. In "On the Application of Artificial Intelligence Techniques to Create Network Intelligence", Arsenio describes the application of AI and machine learning techniques in several areas related to ICT, namely epidemic detection and control, intelligent buildings monitoring, middleware and cloud robotics (inter

process communication), Client profiling and Alarm predication in telecommunication networks. In "A Statistical Framework for Mental Targets Search Using Mixture Models", Bdiri et al. utilize machine learning techniques to search specific target based on visual features. The authors' main objective is to improve the human–machine interaction. In "Variational Learning of Finite Inverted Dirichlet Mixture Models and Applications", Tirdad et al. use statistics and machine learning to solve the problem of data modeling. The proposed framework has been applied to the challenging tasks of natural scene categorization and human activity classification. In "A Fully Bayesian Framework for Positive Data Clustering", Al Mashrgy and Bouguila propose a Bayesian framework for data modeling. The main objective is to enhance the clustering process of positive data. The applicability of the proposed model is shown in the real-life problem of object detection. In "Applying Information Extraction for Abstracting and Automating CLI-Based Configuration of Network Devices in Heterogeneous Environments", Martinez et al. present a tool to automate/simplify the administration of network devices in heterogeneous environments. The proposed tool uses the Information Extraction and knowledge representation, from a designed ontology, and natural language processing of provided Command Line Interfaces. Finally, "MKMSIS: A Multi-agent Knowledge Management System for Industrial Sustainability" by Lopez et al. describes a framework for collaborative development to comply with environmental sustainability using multi-agent technology.

Part I
Search and Optimization

A Metaheuristic for Optimizing the Performance and the Fairness in Job Scheduling Systems

Dalibor Klusáček and Hana Rudová

Abstract Many studies in the past two decades focused on the problem of efficient resource management and job scheduling in large computational systems such as HPC clusters and Grids. For this purpose, the application of Artificial Intelligence-based methods such as metaheuristics has been proposed in many works. This chapter provides an overview of such works that involve metaheuristics and discusses why mainstream resource management and scheduling systems are instead using only a limited set of rather simple scheduling policies. We identify several reasons that are causing this situation, e.g., a common use of overly simplified problem definitions with rather naive job and machine models or an application of unrealistic optimization criteria. In order to solve aforementioned issues, this chapter proposes new complex and well designed approaches that involve the use of metaheuristic which periodically optimizes job scheduling plan using several real life based optimization criteria. Importantly, approaches described in this chapter are successfully used in practice, i.e., within a production job scheduler which manages the computing infrastructure of the Czech Centre for Education, Research and Innovation in ICT (CERIT Scientific Cloud).

Keywords Job scheduling · Metaheuristic · Optimization · Fairness

1 Introduction

The job scheduling problem is known to be very demanding, especially when large and heterogeneous computer environments such as computer clusters and Grids are considered. Here, both users and resource owners should be satisfied simultaneously. Typically, resource owners prefer to keep the overall resource usage reasonably high

D. Klusáček (✉) · H. Rudová
Faculty of Informatics, Masaryk University, Brno, Czech Republic
e-mail: xklusac@fi.muni.cz

H. Rudová
e-mail: hanka@fi.muni.cz

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_1

while users request short wait times. At the same time, fairness has shown to be one of the most important factors to keep users satisfied [1, 2]. Therefore the users should be treated in a fair fashion, such that the available computing power is fairly distributed among them [1]. Last but not least, the predictability, i.e., planning functionality [3, 4] is very useful as it allows users to better understand when and where their jobs will be executed. In fact, even experienced users often do not understand the scheduling decisions as delivered by existing scheduler that does not use planning functionality [2]. In order to meet these goals sophisticated and automated scheduling techniques should be applied, allowing to handle all requirements in an efficient manner. On the other hand, these techniques must remain robust and time-efficient as the system is typically highly dynamic. Then, reactions of the scheduler must remain fast in order to properly reflect continuously changing state of the system.

In order to solve this problem efficiently, many studies considered the application of Artificial Intelligence-based methods such as metaheuristics [5]. Despite this large effort, mainstream resource management and scheduling systems are still using only a limited set of scheduling policies [6]. A typical scheduling system relies on job queues with priorities, while backfilling—a simple optimization of (priority) First Come First Served (FCFS) policy developed in 1995 in order to increase resource utilization—is typically the most advanced option available [7]. Obviously, there must be some good reason why Artificial Intelligence (AI) based methods are not usually used in practice. Therefore, this chapter provides an overview of existing works that involve metaheuristics. We identify several reasons why mainstream systems still rely on rather simple scheduling policies. As we show, existing works using AI methods often use simplified problem definitions with rather naive job and machine models. For example, authors often assume that (exact) job processing times are known in advance, which allow for the construction of accurate job schedules. However, such an assumption is far from reality since only highly inaccurate estimates are usually available, and an actual processing time is only known when a job actually completes its execution [4]. Moreover, although the real system should be represented as a multi-criteria optimization problem, many works use simplified approaches involving single criterion. Also, metaheuristics are frequently considered as "slow" and "static" in contrast to the widely used queue-based approaches that allow for fast and dynamic decisions that are required in real systems.

This chapter—building upon and extending the results of the Ph.D. thesis [8]—demonstrates that aforementioned issues can be successfully addressed by using complex and well designed approaches. In the second part of the chapter, we apply a *metaheuristic algorithm* to optimize the initial job schedule which is created by the well known Conservative backfilling [4] scheduling algorithm. Unlike many existing works that often focus only on the performance-related criteria, we simultaneously optimize several real life based optimization criteria. Notably, we emphasize the fairness with respect to different users of the system, as this is a very important real life based requirement. The application of a metaheuristic largely improves the performance and fairness of Conservative backfilling while also outperforming

other classical techniques. Moreover, our solution outperforms those widely used scheduling techniques even when realistic (very inaccurate) job processing time estimates are used. At the same time, it does not introduce any major overhead thanks to the application of properly designed execution model and efficient data structures [9]. Last but not least, algorithms and approaches described in this chapter were reimplemented within a production job scheduler that manages the computing infrastructure of the Czech Centre for Education, Research and Innovation in ICT (CERIT Scientific Cloud), already demonstrating their usefulness in practice.

This chapter is organized as follows. First, we define the studied problem. Next, an overview of existing scheduling techniques is presented. Especially, the fairness and the problems related to the inaccurate job processing times estimates are emphasized. Also, several existing works that use advanced scheduling methods like metaheuristics are presented. Section 4 describes the proposed complex extension of Conservative backfilling including the metaheuristic. Section 5 presents experimental evaluation of the proposed solution demonstrating good performance of our approach. Finally, we conclude with a short description of the production scheduler that uses the proposed solution in practice and we discuss the future work.

2 Problem Description

In this section we describe the investigated job scheduling problem. In general, the goal of the scheduler is to allocate users' jobs on available machines in time. During this process, jobs requirements concerning corresponding machines parameters must be satisfied. Also, one or more optimization criteria are followed by the scheduler. Therefore, we will further describe characteristics of considered machines and jobs and we will also define a set of optimization criteria that are used when evaluating the quality of generated solutions.

2.1 Machines

A considered system is composed of one or more computer clusters and each cluster is composed of several machines. So far, we expect that all machines within one cluster have the same parameters, e.g., the number of CPUs and the CPU speed. All machines within a cluster use the Space Slicing processor allocation policy [10] which allows parallel execution of several jobs at the cluster when the total amount of requested CPUs is less than or equal to the number of CPUs of the cluster. Therefore, several machines within the same cluster can be co-allocated to process a given parallel job. On the contrary, co-allocation of machines belonging to different clusters is not supported.

2.2 Jobs

A job represents a user's application. In total, there are n jobs in the system which arrive over the time. Therefore the arrival time r_j of each job is specified. There are no precedence constraints among jobs. Job may require one (sequential) or more CPUs (parallel), denoted as $usage_j$. Job runtime is denoted as p_j (processing time). Let S_j and C_j represent the start time and the completion time of the job j respectively. Since we do not consider job preemptions $C_j = S_j + p_j$ holds for every job j. An actual runtime (processing time) is *typically unknown* to a scheduling algorithm until a job completes. Instead of that, only an estimate is usually known in advance. Such an estimate is denoted as ep_j (estimated processing time).

Let us briefly describe the assumptions that can be made concerning job runtime estimates. In a real system, these estimates are usually provided by users and are very inaccurate [4, 11]. Also, users may not specify these estimates at all. Although runtime estimates may not be strictly required, most scheduling systems favor those users who are able to provide such information. Their jobs can be used for backfilling (see Sect. 3.1) which increases utilization and decreases wait times. If no user estimate is provided, the default *queue time limit* can be used instead, since nowadays systems typically use more queues that specify the maximum allowed runtime of a job [12]. A runtime estimate or a queue time limit is the maximum time limit that a job can execute. In case that a job exceeds its available runtime it is killed immediately [4, 11]. As a result, an actual job runtime p_j is always bounded by its estimate, i.e., $p_j \leq ep_j$.

2.3 Optimization Criteria

There are plenty of optimization criteria that are used both in theory and practice. In this chapter, the quality of generated solutions is measured using several performance-related as well as fairness-related criteria.

2.3.1 Performance-Related Criteria

The *avg. response time* [10] represents the average time a job spends in a system, i.e., the time from its submission to its termination (Eq. 1). The *avg. bounded slowdown* [10] is the mean value of all jobs' bounded slowdowns. Bounded slowdown is the ratio of the actual response time of the job to the response time if executed without any waiting[1] (Eq. 2). *Avg. wait time* [13] is the mean time that jobs spend waiting before their executions start (Eq. 3).

[1] To avoid huge slowdowns of extremely short jobs, the minimal job runtime is bounded by some predefined time constant (e.g., 10 s), sometimes called a "threshold of interactivity" [10].

$$RT = \frac{1}{n}\sum_{j=1}^{n}(C_j - r_j) \tag{1}$$

$$BSD = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{C_j - r_j}{\max(10,\, p_j)}\right) \tag{2}$$

$$WT = \frac{1}{n}\sum_{j=1}^{n}(S_j - r_j) \tag{3}$$

As pointed out by Feitelson et al. [10], the use of response time places more weight on long jobs and basically ignores if a short job waits few minutes, so it may not reflect users' notion of responsiveness. Slowdown reflects this situation, measuring the responsiveness of a system with respect to a job length, i.e., jobs are completed within the time proportional to a job length. Wait time criterion supplies the slowdown and the response time. Short wait times prevent the users from feeling that the scheduler "forgot about their jobs".

Of course, also the time complexity of considered scheduling algorithms is of interest. For this purpose we measure the *avg. runtime of a scheduling algorithm per job*. It measures the mean CPU time needed by a scheduling algorithm to develop a scheduling decision for one job (see Eq. 4 where AR_j is the CPU time that a given scheduling algorithm utilizes to develop a scheduling decision(s) for a given job j).

$$AR = \frac{1}{n}\sum_{j=1}^{n} AR_j \tag{4}$$

2.3.2 Fairness-Related Criteria

Previous criteria mostly focused on an overall job performance. Still, good performance is not the only aspect that makes the scheduler acceptable. The scheduler must also be fair, i.e., it must guarantee that the available computing power is fairly distributed among the users of the system. Currently, there is no widely accepted standardized metric to measure fairness and different authors use different metrics [14–16]. An overview of existing techniques including discussion of their suitability can be found in [2].

Very often, the fairness is understood and represented as a *job-related* metric, meaning that every job should be served in a fair fashion with respect to other jobs [14–16]. Such requirements are already partially covered by the common performance criteria like the slowdown and the wait time that were both discussed earlier. However, we aim to guarantee fair performance to *different users* of the system as well. Therefore, we use a different metric which is inspired by the well known fair-share principle [1, 17] that is commonly used in production systems [17] (see Sect. 3.3). Just like fair-share, we try to minimize the differences among (normalized)

mean wait times of all users. Let o be a given user (job owner) in the system and $\mathcal{JOBS}_o$ be the set containing jobs of user o. Then the *normalized user wait time* ($NUWT_o$) for each user (job owner) o is calculated as shown by Eq. 5.

$$NUWT_o = \frac{TUWT_o}{TUSA_o} \tag{5}$$

$$TUWT_o = \sum_{j \in \mathcal{JOBS}_o} \left(S_j - r_j\right) \tag{6}$$

$$TUSA_o = \sum_{j \in \mathcal{JOBS}_o} \left(p_j \cdot usage_j\right) \tag{7}$$

$NUWT_o$ is the *total user wait time* (see $TUWT_o$ in Eq. 6) divided (normalized) by the so called *total user squashed area* (see $TUSA_o$ in Eq. 7), which can be described as the sum of products of a job runtime (p_j) and the number of requested processors ($usage_j$). The normalization is used to prioritize less active users over those who utilize the system resources very frequently [1]. Since users wait times are considered as well, a higher priority is given to users with large wait times.

The normalized user wait time ($NUWT_o$) metric can be used "on the fly" by the scheduling algorithm to dynamically prioritize users. Also, it can be used in graphs with experimental results (see Fig. 3) to reflect the resulting (un)fairness of an applied scheduling algorithm. In this case, the interpretation is following. The closer the resulting $NUWT_o$ values of all users are to each other, the higher is the fairness. If the $NUWT_o$ value is less than 1.0, it means that the user spent more time by computing than by waiting, which is advantageous. Similarly, values greater than 1.0 indicate that the total user wait time is larger than the computational time of his or her jobs.

3 Overview of Existing Scheduling Approaches

In this section we describe existing approaches in the area of (parallel) job scheduling in computing clusters and Grids. First, we describe several widely used scheduling algorithms, discussing their overall suitability. Next, we explain the crucial role of (inaccurate) job runtime estimates on the performance of existing solutions. Then, methods to reflect fairness are explained. Finally, we show how advanced AI-based methods such as metaheuristics can be used to optimize job schedules with respect to applied optimization criteria.

3.1 Standard Scheduling Algorithms

All major production systems such as PBS Pro [12], TORQUE [18], LSF are so called queuing systems. It means that these systems follow the queue-based scheduling approach, using one or more incoming *queues* where jobs are stored until they are scheduled for execution. Job ordering (queue prioritization) and job selection is done using some scheduling policy. Often, these policies are based either on (priority) *First Come First Served (FCFS)* [12, 19] or on the *backfilling* algorithm [19]. FCFS always schedules the first job in the queue, checking the availability of the resources required by such job. If all the resources required by the first job in the queue are available, it is immediately scheduled for the execution, otherwise FCFS waits until all required resources become available. While the first job is waiting for the execution none of the remaining jobs can be scheduled, even if required resources are available. Despite its simplicity, FCFS approach presents several advantages. It does not require an estimated processing time of the job and it guarantees that the response time of a job that arrived earlier does not depend on the execution times of jobs that arrived later. On the other hand, if parallel jobs are scheduled the strict FCFS ordering of job selection often implies a low utilization of the system resources, that cannot be used by some "less demanding" job(s) from a queue [14, 19]. To solve this problem algorithms based on backfilling are frequently used [4].

Algorithms using *backfilling* represent an optimization of the FCFS algorithm that try to maximize resource utilization [19]. There are several variants of backfilling algorithms. The most popular one is the aggressive *EASY backfilling* [4]. It works as FCFS but when the first job in the queue cannot be scheduled immediately, EASY backfilling calculates the earliest possible starting time for the first job using the processing time estimates of running jobs. Then, it makes a reservation to run the job at this pre-computed time. Next, it scans the queue of waiting jobs and schedules immediately every job not interfering with the reservation of the first job [19]. This helps to increase the resource utilization, since idle resources are *backfilled* with suitable jobs, while decreasing the average job wait time. EASY backfilling takes an aggressive approach that allows short jobs to skip ahead provided they do not delay the job at the head of the queue. The price for improved utilization of EASY backfilling is that execution guarantees cannot be made because it is hard to predict the size of delays of jobs in the queue. Since only the first job gets a reservation, the delays of other queued jobs may be, in general, unbounded [4].[2] Therefore, without further control, EASY does not guarantee fairness and may cause a huge job starvation.

In order to prevent such situation, several approaches can be taken. First, the number of reservations can be increased. In case of slack-based [20] and selective backfilling [16] the number of jobs with a reservation is related to their current wait time and slowdown respectively. *Conservative backfilling* [16, 21, 22] makes

[2] If a job is not the first in the queue, new jobs that arrive later may skip it in the queue. While such jobs do not delay the first job in the queue, they may delay all other jobs and the system cannot predict when a queued job will eventually run [4].

reservation for every queued job which cannot be executed at a given moment. It means that backfilling is performed only when it does not delay any previous job in the queue. Clearly, this reduces the core problem of EASY backfilling where jobs close to but not yet at the head of the queue can be significantly delayed. The price paid is that the number of jobs that can utilize existing gaps is reduced, implying that more gaps are left unused in Conservative backfilling than in EASY backfilling [23]. Still, both approaches lead to significant performance improvements compared to FCFS [24]. As the scheduling decisions are made upon job submittal, it can be predicted when each job will run, giving the users execution guarantees. Users can then plan ahead based on these guaranteed response times. Obviously, there is no danger of starvation as a reservation is made for every job that cannot be executed immediately. Apparently, such approach places a greater emphasis on predictability [4, 21] and it is a good compromise between "fair" but inefficient FCFS and "unfair" but efficient EASY backfilling.

3.2 Scheduling with Job Runtime Estimates

The use of reservations in, e.g., the backfilling algorithms brings one major problem. In order to create accurate schedules and reservations, reasonably precise job runtime estimates are required to allow efficient and accurate predictions. The more precise these estimates are, the better results can be expected from these techniques [25, 26]. In current systems, estimates are usually specified either by a user or by a predefined time limit associated with a queue. Sadly, such estimates are typically very inaccurate and overestimated [4, 25, 27] as we have discussed in Sect. 2.2.

3.2.1 Existing Approaches

Surprisingly, many works ignore this problem [28–30] and expect that precise runtime estimates are available. However, this represents rather unrealistic problem. As far as we know, those solutions that use inaccurate runtime estimates often follow similar approach, i.e., they *re-compute the schedule from scratch* when an inaccurate runtime estimate is detected due to an early job completion [3, 4, 31].

Let us have a look at the procedure applied in Conservative backfilling. Here—as a result of the overestimated runtime—jobs can usually start earlier [4, 21] than their reservations allow. The first solution is to leave existing reservations intact, disregarding that earlier time slots are appearing. Sadly, such a solution is very inefficient. Another solution is to cancel all existing reservations and re-backfill all jobs according to a new situation and establish new reservations [4]. However, this may violate the system's execution guarantees that were provided by the previous—now canceled—reservations [4]. Backfilling allows later jobs to skip over jobs that arrived earlier, provided a suitable gap is found in the reservations' schedule. This can only happen when the available gap is smaller than the size of some earlier job, but large

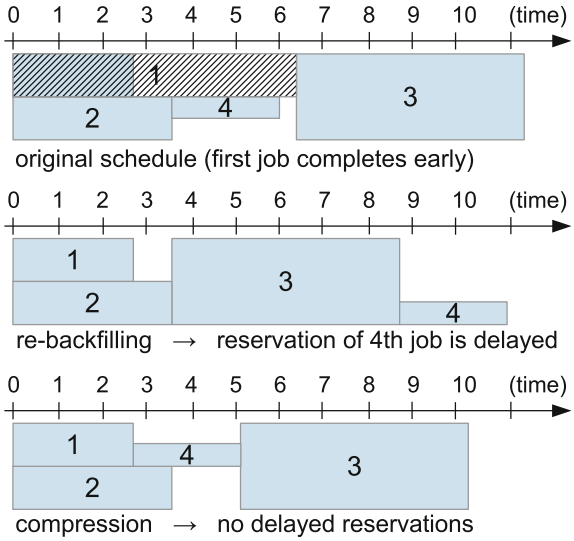

Fig. 1 An early job completion (*top*) and the difference between re-backfilling (*middle*) and the schedule compression (*bottom*) used in Conservative backfilling

enough for the later job. However, if a new round of backfilling is done later, thanks to an earlier termination, these gaps may become large enough for the earlier job so it will get reservation there. Later jobs now may not be backfilled as previously and will therefore run much later than was the guaranteed time of the previous reservation [4]. An example of such a situation is shown in Fig. 1 (middle). It shows how re-backfilling delays fourth job in the schedule with respect to the previous schedule that used overestimated runtime of the first job (see Fig. 1 (top)). To avoid such situations, another approach is taken where existing reservations are *compressed* [4] such that no delays can occur. Existing reservations are checked one by one starting with the earliest (nearest) reservation and they are inserted at the earliest possible start time [4] as shown in Fig. 1 (bottom). Still, gaps may remain in such compressed schedule. However, these provide new backfilling opportunities, that can be exploited in the future by newly arrived jobs [21]. Also, the time complexity of compression is quadratic because the schedule is scanned again for each inserted job [4]. Therefore, for large number of jobs this process may be quite time consuming as was observed and discussed, e.g., in [31] and further confirmed in our own experiments (see Sect. 5.2).

3.2.2 Runtime Prediction Techniques

Since the inaccuracy of runtime estimates represents a significant problem it is not surprising that several automated techniques have been proposed to establish more precise estimates without a user interaction. Typically, historical information together with the statistical analysis of the previously executed jobs are used to predict/refine job runtime or wait time [26, 32] or compute the probability that jobs will exe-

cute within a specified time limit [33]. Still, all these methods represent a major drawback—a newly generated estimate can be smaller than is an actual runtime of a job. As a result, reservations cannot be guaranteed.

3.3 User-to-User Fairness and Fair-Share

As discussed earlier, it is crucial to guarantee fair access to computing resources. All popular resource management systems and schedulers such as PBS Pro [12], TORQUE [18], Maui [17], etc., support some form of so called fair-share mechanism. A nice explanation of Maui's fair-share mechanism can be found in [17]. In order to maintain fairness among users, a priority mechanism is applied, where jobs waiting in queue(s) are ordered according to dynamically updated priorities of users [2]. A priority of a user is computed using the well known *max-min* approach [2], i.e., the highest priority obtains a user with the smallest amount of consumed CPU time and vice versa. Fair-share may also consider other resources, e.g., used RAM memory, using a so called multi-resource aware approach [34]. Furthermore, additional normalization is often done to reflect the overall wait time of given user as well. For example, a priority of a user is computed using the Eq. 5 that represents the normalized user wait time $NUWT_o$. Then, all jobs submitted by a given user get a priority of that user. Once priorities are calculated for all users, their jobs in queue(s) are then ordered in the *highest* $NUWT_o$ *first* order. Priorities of users are updated dynamically, as their jobs arrive and/or complete. Then, all users of the system are prioritized such that their wait times and CPU consumptions are reasonably balanced over the time.

The main benefit of this approach is that it can easily coexist with existing scheduling policies. The only difference is that queue(s) are now prioritized using fair-share priorities. No or little modifications to previously discussed policies are required. For example, a straightforward fair-share prioritization of waiting jobs results in a very fair scheduler [15, 16]. On the other hand, this strict fair job ordering implies that problems related to the low resource utilization remain. As discussed in Sect. 3.1, algorithms based on backfilling are frequently used to solve this problem. Since backfilling selects jobs out of order it tends to dilute the impact of fair-share priorities. It does not eliminate this impact, but it may noticeably decrease it as priorities of users may not be reflected [17]. In reality, existing reservations often represent a pessimistic scenario as jobs are typically completing earlier than their initial estimates suggest [4] and the schedule compression moves reservations to earlier time slots. In such situations, it is beneficial to adjust existing reservations based on the corresponding fair-share priorities such that the highest priority job is allowed to access the newly available resources first. Therefore, high priority jobs get the best chance of improving their start time at each early job completion [17]. Given the pros and cons, such a form of priority backfilling is considered acceptable as its drawbacks are quite minor while its benefits are widespread and significant.

3.4 Advanced Optimization Methods

Algorithms that support planning, e.g., Conservative backfilling, allow to plan job execution ahead using job runtime estimates. Using them, the "plan of job execution" or the "job schedule" is created, i.e., a data structure representing jobs-to-machines mapping in time [3, 29] is continuously maintained. Although job schedule supports predictability, it does not automatically guarantee that resulting solutions will be efficient with respect to considered optimization criteria. For example, Conservative backfilling or the backfill-like policies applied in the CCS [35] scheduling system all use planning but job reservations are still established in the order of job arrivals. In another words, although reservations allow to plan ahead, without an *evaluation* and an *optimization* they represent an ad-hoc solution where decisions are fixed according to initially applied strategy and do not change even when it is clear that such schedule is inefficient.[3]

To avoid problems related to the "ad-hoc" application of the schedule-based solutions an *evaluation* can be applied, analyzing the quality of generated schedules. Then some optimization procedures can be used to improve the quality of the initial schedule. Several metaheuristics have been applied in the literature to perform such an optimization. Authors of [36] propose an application of genetic algorithm and tabu search algorithm. However, they use only static system model, without dynamic job arrivals. Similarly, various techniques such as simulated annealing, genetic algorithms or hill climb search are proposed to minimize the makespan for static problem instances in [29, 37]. However, solved problems are too simple, considering static job pool, unary resources and sequential (non-parallel) jobs only. Same problems, extended by dynamic job arrivals and resource changes, are solved in [38] using a genetic algorithm in a periodical rescheduling procedure. Regrettably, runtime issues of the rescheduling procedure with respect to the problem size are not discussed in the paper. In [39] the authors propose simulated annealing, genetic algorithm and tabu search including their hybrid versions such as genetic-simulated annealing and hybrid genetic-tabu search. Unfortunately, this paper only presents the proposal of the aforementioned algorithms but no experimental evaluation is presented to support the given ideas. Recent applications and related works can be found in [40–42] or in the survey [43]. Sadly, none of the works presented above use a truly realistic model. Typically, these works do not consider dynamic job arrivals, parallel jobs, or do not deal with inaccurate job runtime estimates, assuming that accurate job runtimes are known prior execution. Moreover, none of these works deal with fairness-related issues.

As far as we know, evaluation and/or metaheuristics are not applied in nowadays production systems. A notable exception represents Global Optimising Resource Broker (GORBA) [44]. It is an experimental schedule-based system designed for scheduling sequential and parallel jobs as well as workflows. It uses so called Hybrid

[3] When required, the schedule can be recreated from scratch, e.g., due to a machine failure or early job completion as discussed in Sect. 3.2.1. Still, no optimization or evaluation is applied during this process.

General Learning and Evolutionary Algorithm and Method (HyGLEAM) optimization procedure, combining local search with the GLEAM algorithm [31, 45] which is based on the principles of evolutionary and genetic algorithms. Sadly, this system is a proprietary solution which is not freely available and it seems that it is no longer operational.

3.5 Summary

Unlike pure queue-based techniques that do not support planning, schedule-enabled systems support predictability and can be further improved via evaluation and optimization. On the other hand, these systems are more sensitive to dynamic features of the system like the inaccuracy of runtime estimates. Here, proper reactions must be taken to keep the schedule up to date and consistent. As far as we know, existing works do not provide any advanced method for solving these problems. They either ignore these features using simplified (unrealistic) problem model or perform a total re-computing of an existing solution. None of these two approaches is very good. Simplified problem does not allow modeling of realistic scenarios while (frequent) re-computations represent potential problem concerning speed and scalability, as was discussed in case of GORBA [31] as well as in our own research [9].

4 Metaheuristic Scheduler

In this section we present our time-efficient extension of Conservative backfilling which involves the use of a metaheuristic. The solution aims at minimizing bad effects of inaccurate runtime estimates while improving performance and guaranteeing good fairness with respect to different users of the system.

Conservative backfilling is a good candidate for extension for several reasons. First, it is capable of handling inaccurate job runtime estimates by using so called schedule compression as discussed in Sect. 3.2.1. Second, as each job gets a reservation waiting jobs cannot be delayed by lately arriving jobs, which is fair, at least from the user's point of view. Moreover, job reservations (an execution plan) are good for users as they can get some sort of guarantee and they "know what is happening". Last but not least, the prepared plan (job schedule) can be easily evaluated with respect to selected optimization criteria, covering both performance and fairness-related objectives. Therefore, possible inefficiencies that can appear in classical Conservative backfilling can be identified and fixed. For this purpose some form of metaheuristic seems to be a natural solution.

By default, our solution uses Conservative backfilling to construct the initial schedule, i.e., every time some new job arrives. The extension consists of three fundamental techniques that improve the overall performance of Conservative backfilling. Those techniques are the *linear schedule compression*, the *schedule evaluation* and the *metaheuristic*, which we describe in the rest of this section.

4.1 Linear Schedule Compression

As we have already explained, runtime estimates are usually *overestimated*. Therefore—in most cases—a job finishes earlier than it is specified in the schedule, creating a gap in the schedule [27]. As discussed in Sect. 3.2.1, it would be very inefficient to leave such gaps in the schedule as they represent unused CPU time that can be used by suitable waiting jobs. Therefore, we propose to use a so called *linear schedule compression* that is based on the approach applied in Conservative backfilling [4], described in Sect. 3.2.1. The schedule compression is a fundamental method when dealing with inaccurate runtime estimates, as it is used whenever some job completes earlier, i.e., very often. Our compression algorithm speeds up the approach used in Conservative backfilling where existing reservations are checked one by one starting with the earliest (nearest) reservation and—if possible—are reinserted at the earliest possible start time. Unlike the original method, linear schedule compression respects the existing ordering of expected job start times. Therefore, it is much faster since the schedule is traversed only once. In Conservative Backfilling, the complexity of the compression is quadratic [4] because the schedule is scanned again for each inserted job. The price paid for decreased time complexity is that some gaps are left unused compared to Conservative backfilling. Still, these gaps may be filled with newly arriving jobs via backfilling.

Figure 2 illustrates the difference between our implementation (bottom) of the compression method and the original method of Conservative backfilling (middle). The original schedule with an early job completion is shown in Fig. 2 (top). As our solution does not traverse the schedule from the beginning for each reinserted job, the original ordering of the expected job start times is kept and the fifth job cannot start prior the fourth job.

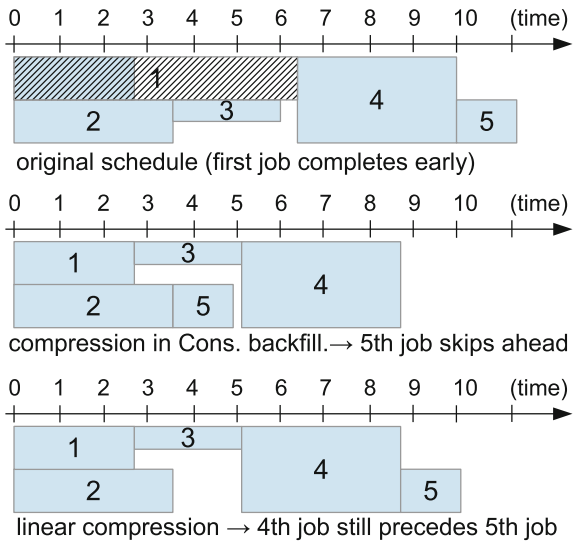

Fig. 2 An early job completion (*top*) and the difference between the schedule compression used in Conservative backfilling (*middle*) and our implementation (*bottom*)

4.2 Schedule Evaluation

As the main purpose of our work is to improve the performance of Conservative backfilling by using a metaheuristic, it is necessary to have a method that can "guide" the optimization process toward better schedules. This is the purpose of the evaluation function which we present here. The function compares two different schedules and decides which one is better with respect to the applied optimization criteria. As we already discussed in Sect. 2, we focus both on the classical and the fairness-related criteria. We use the avg. wait time (*WT*), the avg. response time (*RT*) and the avg. bounded slowdown (*BSD*) to measure the performance of the scheduling algorithm. Each such metric can be easily used when deciding, which solution is better—the one having smaller values of given metric. When considering fairness with respect to different users the situation is more complicated. The fairness-related *normalized user wait time* ($NUWT_o$) described in Sect. 2.3 cannot be directly used as it is a per-user metric. For our purpose we need a function that—given a schedule—returns a *single value*. Therefore, we adopt a criterion called *fairness* (*F*), which we proposed in [2]. It is computed as shown by Eq. 9.

$$UWT = \frac{1}{u} \sum_{o=1}^{u} NUWT_o \tag{8}$$

$$F = \sum_{o=1}^{u} (UWT - NUWT_o)^2 \tag{9}$$

First, we calculate the mean *user wait time* (*UWT*) using the values of $NUWT_o$ as shown in Eq. 8. Then the fairness *F* is calculated by the Eq. 9. The squares used in *F* calculation guarantee that only positive numbers are summed and that higher deviations from the mean value are more penalized than the small ones. This approach has been inspired by the widely used *variance* metric used in statistics to measure dispersion. The fairness (*F*) criterion is used during the evaluation of performed scheduling decisions, i.e., "inside" the optimization procedure. When two possible solutions are available, then the values of *F* are computed for both of them. The one having smaller *F* has smaller dispersion of normalized users' wait times and thus is considered as more fair.[4]

Together, there are four criteria to be optimized simultaneously. Each criterion produces one value characterizing the solution. The final decision on which of the two solutions is better is implemented in separate function, called SELECTBETTER ($schedule_A, schedule_B$) which is shown in Algorithm 1. It is a form of a *weight function*, which is often used when solving multi-criteria optimization problems [9, 43]. The SELECTBETTER function is based on the function that has been already successfully used in our previous works [2, 9].

[4] Equalizing normalized users' wait times is an analogy to the well known fair-share mechanism [17] which is commonly applied in production systems (see Sect. 3.3).

Algorithm 1 SELECTBETTER($schedule_A, schedule_B$)

```
1: compute BSD_A, WT_A, RT_A, F_A of schedule_A;
2: compute BSD_B, WT_B, RT_B, F_B of schedule_B;
3: v_BSD := (BSD_A − BSD_B)/BSD_A;
4: v_WT := (WT_A − WT_B)/WT_A;
5: v_RT := (RT_A − RT_B)/RT_A;
6: v_F := (F_A − F_B)/F_A;
7: weight := v_BSD + v_WT + v_RT + v_F;
8: if weight > 0 then
9:    return schedule_B;
10: else
11:    return schedule_A;
12: end if
```

This function uses two inputs—the two schedules that will be compared. The $schedule_A$ may represent existing (previously accepted) solution while $schedule_B$ represents the newly created *candidate solution*, a product of optimization. First, the values of used objective functions are computed for both schedules (lines 1–2). Using them, decision variables v_{BSD}, …, v_F are computed (see lines 3–6). Their meaning is following: when the decision variable is positive it means that the $schedule_B$ is better than the $schedule_A$ with respect to the applied criterion. Strictly speaking, decision variable defines percentual improvement or deterioration in the value of objective function of $schedule_B$ with respect to the $schedule_A$. Some trivial correction is needed when the denominator is equal to zero, to prevent division by zero error. To keep the code clear we do not present it here. It can easily happen, that for given $schedule_B$ some variables are positive while others are negative. In our implementation the final decision is taken upon the value of the $weight$ (line 1), which is computed as the (weighted) sum of decision variables. If desirable, the "importance" of each decision variable can be adjusted using a predefined weight constant. However, proper selection of these weights is not an easy task. In this particular case, all decision variables are considered as equally important and no additional weights are used. When the resulting $weight$ is positive the candidate $schedule_B$ is returned as the better schedule. Otherwise, the existing $schedule_A$ is returned.

4.3 Metaheuristic for Schedule Optimization

In our solution, the proposed optimization algorithm is used for two purposes. First, the optimization is launched periodically improving the quality of the initial solution delivered by Conservative backfilling. Second, every time an early job completion is handled by schedule compression, quick optimization is then used as a "schedule repair" procedure. The optimization process is always subject to evaluation and only those moves that improve the quality of the schedule are accepted. Let us first describe the proposed optimization algorithm.

4.3.1 Optimization Algorithm

The optimization is carried out by an iterative local search-inspired algorithm that evaluates each move using the applied optimization criteria (see Sect. 4.2). To guarantee fast response of the system, optimization phase represents a *low priority* operation which is always interrupted when a new high priority event such as new job arrival, job completion, machine failure/restart is delivered to the system. The proposed optimization algorithm is called RANDOM SEARCH and its pseudo code is shown in Algorithm 2.

Algorithm 2 RANDOM SEARCH($schedule, iterations, time\_limit$)

```
1: schedule_best := schedule; i := 0;
2: while (i < iterations and time_limit not exceeded) do
3:    i := i + 1;
4:    job := select random job from schedule;
5:    remove job from schedule;
6:    compress schedule;
7:    schedule := move job into a random position in schedule;
8:    schedule_best := SELECT BETTER(schedule_best, schedule);
9:    schedule := schedule_best;    (update/reset candidate schedule)
10: end while
11: return schedule_best;
```

The RANDOM SEARCH optimization algorithm uses three inputs—the schedule that will be optimized ($schedule$), the maximal number of iterations ($iterations$) and the time limit ($time\_limit$). In each iteration, one random job is selected (line 4) and it is removed from its current position. Job removal causes that a gap appears in the schedule. Therefore, the schedule is immediately compressed (see Sect. 4.1), shifting job reservations to available earlier time slots. Next, the removed job is returned to the compressed schedule. RANDOM SEARCH selects totally random position (see line 7), using an aggressive approach where any coordinate can be used to place that job, possibly affecting start times of other jobs. This new schedule is evaluated with respect to the applied criteria in the SELECT BETTER($schedule_{best}, schedule$) function (see Algorithm 1). If this attempt is successful SELECT BETTER returns $schedule$ as the new $schedule_{best}$. Otherwise, the $schedule_{best}$ remains unchanged (line 8). Then the $schedule$ is updated/reset with the $schedule_{best}$ (line 9). The loop continues until the $iterations$ or the given $time\_limit$ are reached (line 2). Then, the $schedule_{best}$ is returned as the newly found solution (line 11).

We may now closely describe the two applications of the proposed optimization algorithm—the periodic optimization and the schedule repair procedure.

4.3.2 Periodic Optimization

As discussed, newly arriving jobs are added into the schedule using Conservative backfilling, where the earliest suitable time slot is always used for the new job. Such an ad-hoc approach may not produce high quality solutions. Therefore, the initial schedule is periodically optimized with the RANDOM SEARCH algorithm. The optimization algorithm is executed every 5 min. Here we were inspired by the actual setup of the TORQUE's scheduler used in the Czech National Grid Infrastructure *MetaCentrum*[5] which performs priority updates of jobs waiting in the queues with the same frequency. The maximum number of iterations is equal to the number of currently waiting jobs multiplied by 2. The (maximum) $time\_limit$ variable was set to be 2 s. Both values have been chosen experimentally—2 s are usually enough to perform all iterations. Moreover, with such a number of iterations there are—on average—two attempts to move each job, which is sufficient to find some improving solution in most cases. However, when some higher priority event such as a new job arrival or a job completion is detected during the optimization phase, the $time\_limit$ is immediately set to 0 and the optimization terminates promptly. Then, a higher priority event is handled accordingly. Therefore, the optimization phase cannot cause any significant delays concerning job processing and the potential overhead of the optimization is practically eliminated [9].

4.3.3 Schedule Repair Procedure

The linear schedule compression which was described in Sect. 4.1 is the primary method used to "repair the schedule" after an early job completion. Still, several gaps may remain in the schedule once the schedule has been compressed (see Fig. 2 (bottom) for an example). These gaps represent unused CPU time that could be utilized by some "later" jobs from the schedule. For this purpose we apply the RANDOM SEARCH as the "schedule repair" optimization procedure. It tries to fill these gaps by finding suitable jobs that can utilize them. Again, the optimization process is guided by the evaluation and only improving moves are accepted. In order to minimize possible overhead, the (maximum) $time\_limit$ variable is set to be only 100 ms in this case, as early job completions typically occur very frequently.

5 Experiments

The main goal of this section is to show that the optimization technique is capable of improving the performance and the fairness of Conservative backfilling.

[5] http://www.metacentrum.cz.

5.1 Simulation Setup

The evaluation was performed experimentally, using the GridSim [46] based Alea simulator.[6] Alea is an advanced job scheduling simulator which allows to perform realistic simulations of the considered job scheduling scenarios. All experiments were computed on an Intel Xeon E5-2665 2.4 GHz machine with 32 GB of RAM.

Six workloads have been used for evaluation: SDSC BLUE (1,152 CPUs, 243,314 jobs during 34 months), CTC SP2 (338 CPUs, 77,222 jobs during 11 months), HPC2N (240 CPUs, 202,876 jobs during 42 months), KTH SP2 (100 CPUs, 28,489 jobs during 11 months), SDSC DataStar (1,664 CPUs, 96,089 jobs during 1 year) and CERIT-SC (2,176 CPUs, 94,900 jobs during 10 months).[7]

These logs were selected for several reasons. First of all, they all contain sufficiently large amount of jobs (28,489–243,314) and represent systems of various sizes, starting with rather small systems (KTH SP2 and HPC2N) and going to larger systems with 1,664 and 2,176 CPUs (SDSC DataStar and CERIT-SC). Last but not least, all these logs represent systems with reasonably high utilization (>50 %) and all workloads contained original (inaccurate) user-provided runtime estimates, which were used in the evaluation, i.e., precise runtimes were unknown to the scheduling algorithms which is a realistic assumption. If available, the recommended "cleaned" versions of the workload logs are always used.

In order to evaluate the suitability of the proposed schedule-based techniques, we have compared their performance with some of the most popular queue-based algorithms that are widely used in the practice as well as in the literature. We have used EASY backfilling (EASY) [4], Conservative backfilling (CONS) [4] as well as their fair-share based versions (EASY-F and CONS-F), where the job queue are dynamically reordered according to fair-share computed job priorities (see Sect. 3.3). Just like in an actual system, priorities of waiting jobs are updated periodically. The queue is always reordered before a new scheduling attempt is performed, such that the highest priority job is at the head of the queue.[8] Our solution was represented by the Random Search (RS) optimization algorithm (see Sect. 4.3). As RS uses a stochastic mechanism, experiments were repeated 10 times for all workloads and their results were averaged.

The evaluation uses the criteria that have been formally defined in Sect. 2.3, i.e., the avg. wait time, the avg. response time and the avg. bounded slowdown. Concerning fairness, the resulting normalized user wait times $NUWT_o$ were collected for all users and we show the arithmetic mean of all such normalized user wait times and the corresponding standard deviation. The smaller the mean and the standard devia-

[6] https://github.com/aleasimulator.

[7] Except for CERIT-SC, all workloads come from the Parallel Workloads Archive [47]. CERIT-SC can be obtained at http://www.fi.muni.cz/~xklusac/workload/.

[8] Other policies like Conservative backfilling using linear compression with RS optimization being disabled (CONS-L), First Come First Served (FCFS) [12], or Shortest Job First (SJF) [16] were also tested, but they performed poorly compared to other algorithms. Therefore, we do not present them in the figures for better visibility.

tion are the lower were the $NUWT_o$ values and the closer (i.e., more fair) they were, respectively. Moreover, the avg. algorithm runtime per job was used to show the runtime requirements of selected algorithms. The runtime of Random Search (RS) is not shown as it cannot be directly compared with the remaining algorithms. As discussed in Sect. 4.3, RS is executed periodically when no higher priority events are detected, i.e., when the system has enough time to perform the optimization. Therefore, its runtime is proportional to the total duration (makespan) of the experiment rather than to the number of jobs. From this point of view, it is more reasonable to measure the avg. runtime of the default scheduling policy, which is used to create the initial schedule after each new job arrival. In our solution it is the Conservative backfilling with the *linear schedule compression* (CONS-L), as discussed in Sect. 4.1. Otherwise, CONS-L is not included in other experiments, as it is a weaker version of CONS policy with a weaker performance and fairness results. Its application makes sense only if accompanied by further RS optimization.

5.2 Experimental Results

The main results are presented in Fig. 3. Bubble charts are used to display corresponding values of wait time and bounded slowdown simultaneously—the y-axis depicts the avg. wait time while the size of the circle represents the avg. bounded slowdown. The actual bounded slowdown value is shown as a label above each circle (see, e.g., Fig. 3). The figure is divided into three columns and six rows. Each row represents results for one workload. Starting from the left, a row shows the results for the wait time and bounded slowdown (left), the average response time (middle), and the mean $NUWT_o$ value and the corresponding standard deviation (right). The results covering our last criterion—the average algorithm runtime per job—are shown in Fig. 5.

Let us first briefly describe the behavior of EASY, CONS and their fair-share based variants (EASY-F, CONS-F). Concerning the avg. wait time and the avg. response time, Conservative backfilling (CONS) does not work very well with respect to EASY, which produces better results in most cases. This is not surprising as these issues have been already addressed in several works [16, 23, 27]. Basically, the problem here is that establishing reservation for every job can be less efficient than aggressive approaches as used in EASY. Reservations decrease the opportunities for backfilling, due to the blocking effect of the reserved jobs in the schedule [16, 21]. On the other hand, for half of the workloads the bounded slowdown (see circle labels in the bubble charts) is slightly better when CONS is applied. This is a normal behavior also observed in previous works [16, 21]. The reason for improved slowdown is that no job can be repeatedly delayed. As soon as fairness is considered (EASY-F and CONS-F) the overall results are mostly improved compared to "unfair" CONS and EASY. This behavior is a result of the prioritization mechanism which dynamically "shuffles" jobs in the queue. Since a user often submits several similar jobs during a session, it may be hard for backfilling to develop an efficient schedule for a group of such similar jobs. However, if the prioritization mechanism "mixes" these jobs in

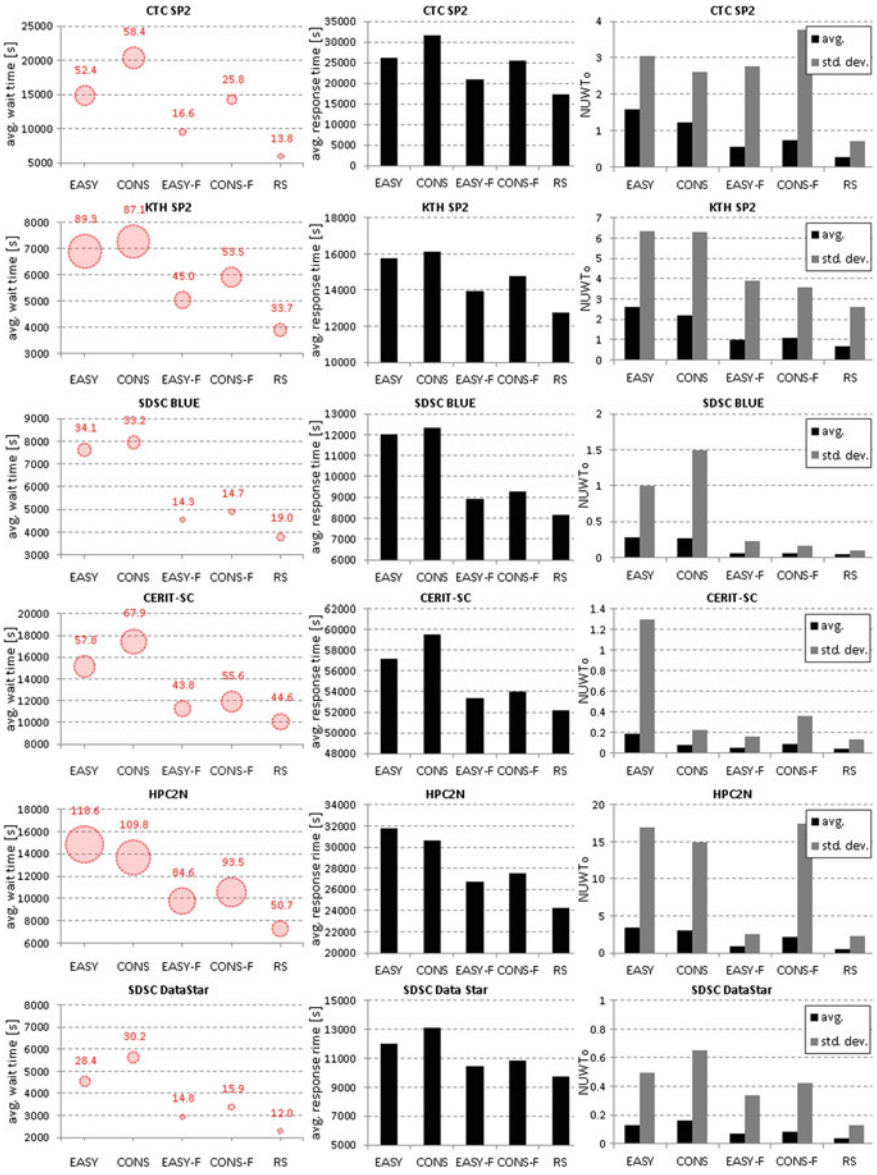

Fig. 3 The avg. wait time and the bounded slowdown (*left*), the avg. response time (*middle*) and the mean and the standard deviation of $NUWT_o$ values (*right*)

the queue with different jobs of other users, backfilling may then often work more efficiently. The more aggressive EASY-F again outperforms CONS-F by producing better slowdowns, wait and response times for all workloads. Concerning fairness, EASY and CONS are usually outperformed by their fair-share based variants (EASY-F, CONS-F), which is natural since EASY and CONS do not apply any technique

to fairly balance resource usage among users. For EASY-F, the normalized user wait times ($NUWT_o$) are always lower on average, with smaller standard $NUWT_o$ deviation compared to EASY. For CONS-F, the average $NUWT_o$ is always smaller than for CONS, while the standard deviation is smaller in 3 cases (50 % of workloads). To conclude the discussion on EASY(-F) and CONS(-F), it is quite clear that EASY-F produces the best results from these four algorithms, at least with respect to the chosen criteria.

Let us now discuss how the proposed Random Search (RS) optimization algorithm influences the performance of the underlying Conservative backfilling. In case of the avg. wait time and the avg. response time, RS produces the best results in all cases. In case of the avg. bounded slowdown, RS either produces the best results (4 workloads), or it generates results that are very close to the best ones (see SDSC-BLUE and CERIT-SC in Fig. 3). In case of fairness, RS again produces the best results by delivering the smallest averages and standard deviations of $NUWT_o$. The results shown in Fig. 3 demonstrate the benefits related to the application of the RS metaheuristic. Especially, the huge improvement obtained with respect to the original Conservative backfilling (CONS) is worth noticing, covering both performance-related criteria and fairness.

Although the results speak for themselves, one may still ask how is it possible, that RS works so well with respect to all other considered algorithms. To answer this question, we have to take a closer look on the data. For example, let us closely analyze job wait times of all algorithms. Instead of just showing average values, we can show a cumulative distribution functions (CDF) of job wait times (see Fig. 4). In this case, a CDF is a $f(x)$-like function showing a fraction of jobs (y-axis) that have their job wait times less than or equal to x. The steeper is the resulting curve and the sooner it reaches the maximum ($y = 1.0$), the better is the performance of a given algorithm. As the resulting distributions have very long tails, the x-axis is not linear. The resulting CDFs are shown in Fig. 4. These CDFs help us to better understand the results from Fig. 3. First of all, they clearly demonstrate that fair-share enabled EASY-F and CONS-F work much better than their original "unfair" versions. Second, they demonstrate why RS outperforms other algorithms. In fact, there is one main reason—as RS is continuously evaluating the schedule, it is able to detect (very) high wait times, slowdowns, etc. Then, it can develop better schedules where these extremes are reduced. As was discussed in Sect. 3.4, neither EASY(-F) nor CONS(-F) can perform such evaluation and/or take some repair actions, as they all work in an ad-hoc fashion. However, with an ad-hoc approach they can only work in a "best effort"-like style, and cannot identify problems as they appear. As a result, the CDFs of wait time for Random Search have shorter tails, i.e., there are less extreme wait times, and those CDFs grow faster compared to EASY(-F) or CONS(-F), i.e., more jobs have smaller wait times. Moreover, it is important that the reduction of extremes is not achieved at the cost of worsening results for the remaining majority of jobs. Similar situation applies for other considered criteria, i.e., response time, bounded slowdown or normalized user wait time.

Last but not least, we show the results concerning the avg. algorithm runtime per job shown in Fig. 5. Since we use inaccurate runtime estimates, the runtime of

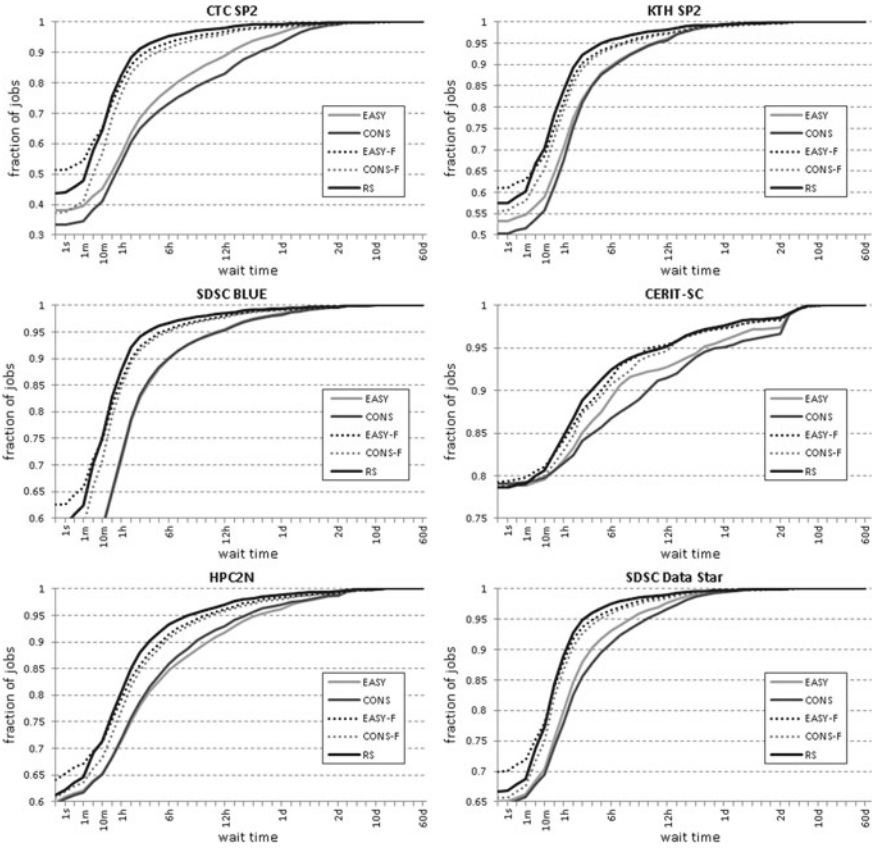

Fig. 4 The CDFs of job wait times for all workloads and algorithms

CONS(-F) grows significantly compared to EASY(-F) since the relatively expensive schedule compression algorithm is applied whenever a job completes earlier than expected. As discussed in Sect. 4.1, our baseline solution uses the linear schedule compression (CONS-L). Thus, CONS-L requires significantly smaller amount of runtime than CONS or CONS-F (see Fig. 5). Still, even if the most time-demanding algorithms like CONS or CONS-F are used, the avg. runtime needed to develop a scheduling decision for one job is always in a decent level (bellow 50 ms) and does not imply any significant overhead.

5.3 Summary

The proposed Random Search (RS) metaheuristic was evaluated using real workloads with realistically inaccurate runtime estimates. Six different workloads have

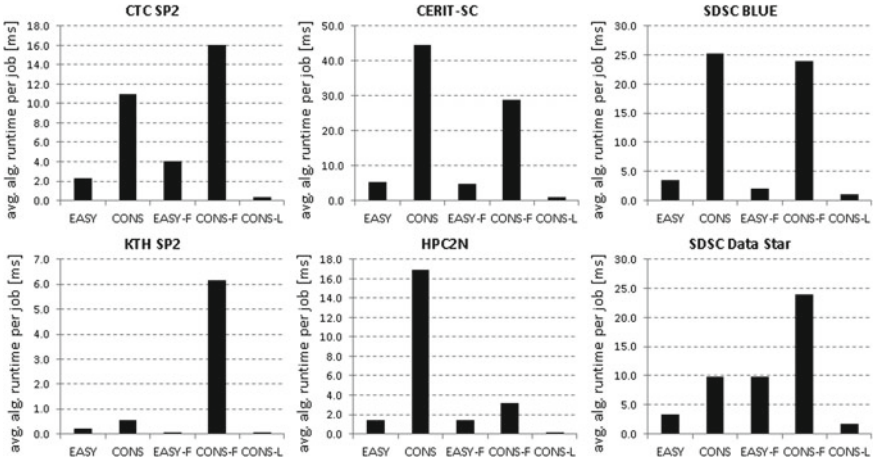

Fig. 5 The avg. algorithm runtime per job for all workloads and algorithms

been used to demonstrate the suitability of the RS metaheuristic using six different objective functions. The experiments indicate that the proposed solution performs very well, being very tolerable to inaccurate runtime estimates. The success of our solution is based on the proposed methodology which combines several approaches in order to develop realistic, fast and efficient solutions. The main idea is to use the linear schedule compression together with the schedule optimization. The optimization is guided by the evaluation which reflects applied objective criteria. Therefore, only good and improving solutions are accepted. It is worth noticing that the optimization is able to reduce (extremely) inefficient job assignments without worsening the performance for other jobs.

6 Conclusion and Future Work

This chapter summarized existing scheduling approaches, starting with standard scheduling policies and their applications in (prioritized) queues. We explored the influence of inaccurate job runtime estimates on the behavior of scheduling algorithms. Fairness-related issues were investigated and considered simultaneously with common performance-related criteria. We discussed existing optimization approaches which were inspired by metaheuristic algorithms. This analysis shows that advanced optimization algorithms deserve more attention as—quite often—their potential was not properly handled by other authors.

The approach involving metaheuristic which was described in this chapter have demonstrated that all important characteristics of the studied problem are properly handled. Also, the performance of the proposed scheduling algorithm has been analyzed with respect to several state-of-the-art queue-based algorithms. It has been

shown that the proposed solution usually outperforms all considered queue-based techniques. Thanks to the proposed schedule compression and the "on demand" correction routine, our solution performs well even when runtime estimates are fully inaccurate, i.e., realistic.

Even more, we were able to use the proposed solution in a real, production scheduler. We have reimplemented our solution which was originally written in Java, preparing a new job scheduler (written in C/C++ language) for the Torque Resource Manager [34]. The new scheduler has been operationally used since July 2014 in the Czech Centre for Education, Research and Innovation in ICT (*CERIT Scientific Cloud*),[9] which is the national center that provides computational and storage capacities for scientific purposes, tightly cooperating with the operator of the Czech National Grid Infrastructure *MetaCentrum*. Currently, this scheduler manages six computer clusters with 4,500 CPU cores. So far, the new scheduler seems to work as intended, providing improved predictability and optimizing job schedules with respect to both the performance and the user-to-user fairness.

Several goals are yet ahead of us. First of all, with the current setup some jobs may start later than initially predicted. These delays are introduced by fair-share and the applied RS optimization algorithm which may postpone planned job start times in order to improve overall quality of the schedule. Our initial experiments suggest that this is not a critical problem as only few percents of jobs ($<5\,\%$) are actually delayed. Also, we have not received any complaints from CERIT users about such behavior. Still, we would like to introduce a simple extension, that would not allow significant job delays to be created by RS. Next, we want to investigate possible benefits of runtime prediction techniques, since our earlier observations [2, 8] suggest that more precise runtime estimates will lead to better performance of our optimization technique.

Acknowledgments We highly appreciate the support of the Grant Agency of the Czech Republic under the grant No. P202/12/0306. The access to the MetaCentrum computing facilities and workloads provided under the program "Projects of Large Infrastructure for Research, Development, and Innovations" LM2010005 funded by the Ministry of Education, Youth, and Sports of the Czech Republic is highly appreciated.

References

1. Kleban, S.D., Clearwater, S.H.: Fair share on high performance computing systems: What does fair really mean? In: Third IEEE International Symposium on Cluster Computing and the Grid (CCGrid'03), pp. 146–153. IEEE (2003)
2. Klusáček, D., Rudová, H.: Performance and fairness for users in parallel job scheduling. In: Cirne, W. (ed.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 7698, pp. 235–252. Springer (2012)

[9] http://www.cerit-sc.cz.

3. Hovestadt, M., Kao, O., Keller, A., Streit, A.: Scheduling in HPC resource management systems: queuing vs. planning. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 2862, pp. 1–20. Springer (2003)
4. Mu'alem, A.W., Feitelson, D.G.: Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. IEEE Trans. Parallel Distrib. Syst. **12**(6), 529–543 (2001)
5. Xhafa, F., Abraham, A.: Metaheuristics for Scheduling in Distributed Computing Environments. Studies in Computational Intelligence, vol. 146. Springer, Berlin (2008)
6. Klusáček, D., Tóth, Š.: On interactions among scheduling policies: finding efficient queue setup using high-resolution simulations. In: Silva, F., Dutra, I., Costa, V.S. (eds.) Euro-Par 2014. LNCS, vol. 8632, pp. 138–149. Springer (2014)
7. Adaptive Computing Enterprises, Inc.: Moab Workload Manager, Jan 2015. http://docs.adaptivecomputing.com/
8. Klusáček, D.: Event-based optimization of schedules for grid jobs. Ph.D. thesis, Masaryk University, 2011
9. Klusáček, D., Rudová, H.: Efficient grid scheduling through the incremental schedule-based approach. Comput. Intell.: Int. J. **27**(1), 4–22 (2011)
10. Feitelson, D.G., Rudolph, L., Schwiegelshohn, U., Sevcik, K.C., Wong, P.: Theory and practice in parallel job scheduling. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 1291, pp. 1–34. Springer (1997)
11. Tsafrir, D., Etsion, Y., Feitelson, D.G.: Modeling user runtime estimates. In: Feitelson, D.G., Frachtenberg, E., Rudolph, L., Schwiegelshohn, U. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 3834, pp. 1–35. Springer (2005)
12. PBS Works: PBS Professional 12.1, Administrator's Guide, Jan 2015. http://www.pbsworks.com
13. Ernemann, C., Hamscher, V., Yahyapour, R.: Benefits of global grid computing for job scheduling. In: GRID'04: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, pp. 374–379. IEEE (2004)
14. Sabin, G., Kochhar, G., Sadayappan, P.: Job fairness in non-preemptive job scheduling. In: International Conference on Parallel Processing (ICPP'04), pp. 186–194. IEEE Computer Society (2004)
15. Sabin, G., Sadayappan, P.: Unfairness metrics for space-sharing parallel job schedulers. In: Feitelson, D.G., Frachtenberg, E., Rudolph, L., Schwiegelshohn, U. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 3834, pp. 238–256. Springer (2005)
16. Srinivasan, S., Kettimuthu, R., Subramani, V., Sadayappan, P.: Selective reservation strategies for backfill job scheduling. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 2537, pp. 55–71. Springer (2002)
17. Jackson, D., Snell, Q., Clement, M.: Core algorithms of the Maui scheduler. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 2221, pp. 87–102. Springer (2001)
18. Adaptive Computing Enterprises, Inc.: TORQUE Resource Manager, Jan 2015. http://docs.adaptivecomputing.com/
19. Lifka, D.A.: The ANL/IBM SP scheduling system. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 949, pp. 295–303. Springer (1995)
20. Talby, D., Feitelson, D.G.: Supporting priorities and improving utilization of the IBM SP scheduler using slack-based backfilling. In: IPPS'99/SPDP'99: Proceedings of the 13th International Symposium on Parallel Processing and the 10th Symposium on Parallel and Distributed Processing, pp. 513–517. IEEE Computer Society (1999)
21. Feitelson, D.G.: Experimental analysis of the root causes of performance evaluation results: a backfilling case study. IEEE Trans. Parallel Distrib. Syst. **16**(2), 175–182 (2005)
22. Li, B., Zhao, D.: Performance impact of advance reservations from the grid on backfill algorithms. In: Sixth International Conference on Grid and Cooperative Computing (GCC 2007), pp. 456–461 (2007)

23. Ngubiri, J.: Techniques and evaluation of processor co-allocation in multi-cluster systems. Ph.D. thesis, Radboud University Nijmegen, 2008
24. Feitelson, D.G., Weil, A.M.: Utilization and predictability in scheduling the IBM SP2 with backfilling. In: 12th International Parallel Processing Symposium, pp. 542–546. IEEE (1998)
25. Chiang, S.-H., Arpaci-Dusseau, A., Vernon, M.K.: The impact of more accurate requested runtimes on production job scheduling performance. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 2537, pp. 103–127. Springer (2002)
26. Smith, W., Taylor, V., Foster, I.: Using run-time predictions to estimate queue wait times and improve scheduler performance. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 1659, pp. 202–219. Springer (1999)
27. Srinivasan, S., Kettimuthu, R., Subrarnani, V., Sadayappan, P.: Characterization of backfilling strategies for parallel job scheduling. In: Proceedings of 2002 International Workshops on Parallel Processing, pp. 514–519. IEEE Computer Society (2002)
28. Yousif, A., Abdullah, A.H., Nor, S.M., Abdelaziz, A.A.: Scheduling jobs on grid computing using firefly algorithm. J. Theor. Appl. Inf. Technol. **33**(2), 155–164 (2011)
29. Abraham, A., Liu, H., Grosan, C., Xhafa, F.: Nature inspired meta-heuristics for grid scheduling: single and multi-objective optimization approaches. In: Metaheuristics for Scheduling in Distributed Computing Environments [5], pp. 247–272 (2008)
30. Abramson, D., Buyya, R., Murshed, M., Venugopal, S.: Scheduling parameter sweep applications on global grids: a deadline and budget constrained cost-time optimisation algorithm. Softw.: Pract. Exper. **35**(5):491–512 (2005)
31. Stucky, K.-U., Jakob, W., Quinte, A., Süß, W.: Solving scheduling problems in grid resource management using an evolutionary algorithm. In: On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE. LNCS, vol. 4276, pp. 1252–1262. Springer (2006)
32. Kumar, R., Vadhiyar, S.: Prediction of queue waiting times for metascheduling on parallel batch systems. In: Cirne, W. (ed.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 8828. Springer (2015)
33. Nurmi, D., Brevik, J., Wolski, R.: QBETS: queue bounds estimation from time series. In: Frachtenberg, E., Schwiegelshohn, U. (eds.) Job Scheduling Strategies for Parallel Processing. LNCS, vol. 4942, pp. 76–101. Springer (2007)
34. Klusáček, D., Chlumský, V., Rudová, H.: Optimizing user oriented job scheduling within TORQUE. In: SuperComputing—The International Conference for High Performance Computing, Networking, Storage and Analysis. Poster, 2013
35. Keller, A., Reinefeld, A.: Anatomy of a resource management system for HPC clusters. Annu. Rev. Scalable Comput. **3**, 1–31 (2001)
36. Subrata, R., Zomaya, A.Y., Landfeldt, B.: Artificial life techniques for load balancing in computational grids. J. Comput. Syst. Sci. **73**(8), 1176–1190 (2007)
37. Ritchie, G., Levine, J.: A fast, effective local search for scheduling independent jobs in heterogeneous computing environments. In: Porteous, J. (ed.) 22nd Workshop of the UK Planning and Scheduling Special Interest Group (PlanSig 03), 2003
38. Carretero, J., Xhafa, F.: Using genetic algorithms for scheduling jobs in large scale grid applications. J. Technol. Econ. Dev. Res. J. Vilnius Gediminas Tech. Univ. **12**(1), 11–17 (2006)
39. Asim YarKhan, J.J.D.: Experiments with scheduling using simulated annealing in a grid environment. In: Parashar, M. (ed.) GRID. LNCS, vol. 2536. Springer (2002)
40. Koodziej, J., Xhafa, F.: Integration of task abortion and security requirements in GA-based meta-heuristics for independent batch grid scheduling. Comput. Math. Appl. **63**(2), 350–364 (2012)
41. Switalski, P., Seredynski, F.: Scheduling parallel batch jobs in grids with evolutionary metaheuristics. J. Sched. 1–13 (2014)
42. Pooranian, Z., Shojafar, M., Abawajy, J., Abraham, A.: An efficient meta-heuristic algorithm for grid computing. J. Comb. Optim. 1–22 (2013)
43. Xhafa, F., Abraham, A.: Computational models and heuristic methods for grid scheduling problems. Future Gener. Comput. Syst. **26**(4), 608–621 (2010)

44. Süß, W., Jakob, W., Quinte, A., Stucky, K.-U.: GORBA: a global optimising resource broker embedded in a Grid resource management system. In: International Conference on Parallel and Distributed Computing Systems, PDCS 2005, pp. 19–24. IASTED/ACTA Press (2005)
45. Jakob, W., Quinte, A. Stucky, K.-U., Süß, W.: Optimised scheduling of Grid resources using hybrid evolutionary algorithms. In: Wyrzykowski, R., Dongarra, J., Meyer, N., Wasniewski, J. (eds.) Parallel Processing and Applied Mathematics, 6th International Conference, PPAM 2005. LNCS, vol. 3911, pp. 406–413. Springer (2005)
46. Sulistio, A., Cibej, U., Venugopal, S., Robic, B., Buyya, R.: A toolkit for modelling and simulating data grids: an extension to GridSim. Concurr. Comput.: Pract. Exper. **20**(13), 1591–1609 (2008)
47. Feitelson, D.G.: Parallel workloads archive (PWA), Jan 2015. http://www.cs.huji.ac.il/labs/parallel/workload/

Hybrid ACO and Tabu Search for Large Scale Information Retrieval

Yassine Drias and Samir Kechid

Abstract This paper presents an attempt to tackle information retrieval (IR) with meta-heuristics. For this aim, we propose two ACO algorithms for information retrieval on large-scale data sets. The main hard issue of this study resides in modeling information retrieval using meta-heuristics that often necessitate links between documents in order to realize move operations from one document to another during the search process. The first novelty in this work is the design of such model to adapt ACO approaches and even other meta-heuristics to IR. The second one resides in the hybridization of ACO approaches with tabu search in order to achieve more efficiency. The designed algorithms and a classical information retrieval method were implemented for comparison purposes. Experiments were conducted on CACM, RCV1 and random benchmarks. Numerical results show that ACO is scalable while achieving the same performance as the traditional IR process in terms of solutions quality.

Keywords Information retrieval · Large-scale data sets · Hybrid meta-heuristics · ACO · Tabu search · CACM · RCV1

1 Introduction

The wide spread information is phenomenal in the web. Information retrieval has consequently shown its great importance in our current life and in many industrial applications. Without the development of this search field, the web would never have aroused the huge interest of the users. Now with the tremendous growth of

Y. Drias(✉) · S. Kechid
USTHB, Bab-Ezzouar Algiers 16111, Algeria, Africa
e-mail: ydrias@usthb.dz

Samir Kechid
e-mail: skechid@usthb.dz

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_2

the contents that the net has known, new appropriate tools to address this problem become necessary to overcome the complexity induced by this situation. In this study, artificial intelligence approaches like ACO algorithms are designed for this purpose.

As we know, real ants apply a stigmergetic way of communication by the use of a hormonal secretion called pheromone. In fact, the ants deposit on ground pheromone when moving and tend to choose the path which has the greatest amount of pheromone. To search food, ants take different directions, but those choosing the shortest path will reach the food more quickly. When they return, the pheromone on the shortest path will be stronger and will attract the successive ants to take this path [1].

In ACO algorithms, artificial ants imitate this way of communication by using artificial pheromone, which is some numerical information saved on the states of the search space of the problem to solve. The first ACO algorithm known in the literature is the ant system AS. It has been proposed by Dorigo [1], and has various extended versions like the Max-Min AS called MMAS [2], the rank based version [3] (ASrank) and the Ant Colony System ACS [4]. The approach has gained a large reputation since it has solved with success many combinatorial optimization problems like the travelling salesman problem (TSP) [5], the quadratic assignment problem (QAP) [6], the vehicle routing problem (VRP) [7], the job shop scheduling problem (JSSP) [8] and the satisfiability problem (SAT) [9].

Motivated by the success and the power of this meta-heuristic and knowing that very few if none of heuristic search techniques have been devoted to investigate information retrieval problem, we designed two ACO algorithms, namely AS-IR and ACS-IR for exploring this domain. The recent works that address IR with meta-heuristics often deal with another features of Information retrieval. For instance, in [10, 11] the authors are interested in documents clustering using meta-heuristics and in [12] the authors focus on query optimization using a genetic algorithm. These studies are completely different from our concern but can be complementary for IR investigations.

The algorithms we designed were tested on real and random collections of documents and comparison of the proposed algorithms to the classical IR method is undertaken.

2 Information Retrieval Background

An information retrieval system handles and manages a collection of documents structured in an internal representation using an indexing process. It consists in finding a set of documents including information expressed in a query specifying user needs. The process involves a matching mechanism between the query and the documents of the collection. Thus three important components are central in such process:

- The document that can be a text, a web page, an image or a video. A document is usually represented by a set of terms or keywords extracted from its source.
- The query that represents a need expressed by a user and specified in a formalism adopted by the system.
- The similarity function that measures the similarity between a document and a query.

Two system evaluations are widely used; the precision which is the fraction of retrieved documents that are relevant and the recall which is the fraction of relevant documents that are retrieved. These indicators serve to evaluate the system model and not the matching mechanism.

In an IR system, an important step is the indexing process in which an internal organization of the documents and the queries is determined in order to access in an efficient way these components. Besides the indexing process, the documents and the queries must be described according to a model.

Many models for IR such as the Boolean model, the vector space model and the probabilistic model exist in the literature. The most widely used which is also appropriate for meta-heuristics is the vector space model. In this model, documents as well as queries are represented as vectors of terms associated with their corresponding weights. Each weight in the vector denotes the importance of the term in the document or in the query. The vector space is built during the indexing process and contains all the terms that the system encounters. In fact, by a well known procedure [13], real documents are translated into data structures containing the most relevant information extracted from the document text in order to be processed by computer programs. A stop list prepared by the system constructor indicates to the system what not to consider as terms when generating the document structure. For instance, only the word roots are kept and articles are suppressed. Generally, there are two main structures; the vector of documents containing terms and the file of terms that includes documents identifiers (called inverted file). According to the vector space model, the set of documents and the set of words or terms are represented respectively as follows:

$$C = (d_1, d_2, d_3, \ldots, d_m)$$
$$T = (t_1, t_2, t_3, \ldots, t_n)$$

T is the set of terms t_i for i=1 to n. For each term, we consider a structure that contains all the documents that include it. The weight of the term in the document is associated with the document in the list. The whole collection C of documents is represented by a file containing all the documents. A document is indexed by its position in the file. Each element of C points towards a list containing all the terms of the documents with their respective weight. The list is sorted according to the identification number of the term for document search efficiency. Besides, the query is modelled exactly as a document.

The weight of a term in a document is computed using the expression $tf * idf$ where *tf* represents the term frequency in the document and *idf* is the inverted frequency computed usually as follows:

$$idf = log(\frac{m}{df})$$

where m represents the total number of documents and df is the number of documents that contain the term. The component tf indicates the importance of the term for the document, while idf expresses the power of discrimination of this term in the whole document collection. This way, a term having a high value of $tf * idf$ is at the same time important in a document and less frequent in others. The weight for a query is computed with the same manner. The similarity of a document d and a query q is computed using one of the following formulas among others [14–16]:

$$f(d,q) = \Sigma i(a_i * b_i) \text{ ..... } (internal\ product)$$

$$f(d,q) = \frac{\Sigma i(a_i * b_i)}{(\Sigma i(a_i)^2 * \Sigma i(b_i)^2)^{1/2}} \text{ .....} (Cosine)$$

$$f(d,q) = \frac{2\Sigma i(a_i * b_i)}{\Sigma i(a_i)^2 + \Sigma i(b_i)^2} \text{ .....} (Dice)$$

$$f(d,q) = \frac{\Sigma i(a_i * b_i)}{(\Sigma i(a_i)^2 + \Sigma i(b_i)^2 - \Sigma i(a_i * b_i))} \text{ .....} (Jaccard)$$

The three last formulas are normalized, a_i and b_i are the weights of term t_i respectively in the document d and in the query q. Only the terms shared by the document and the query are involved in the calculation. One of these important functions is selected in order to be used for matching a document with a query during the search process. The more the matching is strong, the more the document will satisfy the query. The *Cosine* formula will be used in our experiments.

The traditional IR process performs the matching mechanism using the inverted file. Instead of crossing all the documents to find the one that contains a term of the query, the process accesses only those documents that share the term with the query through the inverted file.

3 Lex as a Tool for Documents Indexing

Indexing documents consists in recognizing the significant words of the documents and inserting them in data structures namely, the inverted file and the dictionary for use in the matching process between the documents and the queries.

Lex, a lexical analyzer [17] is a tool for automatically and rapidly implementing a lexer for a programming language or a language defined by any language designer. Lex is widely used in compilers construction but it is also prevalent in many areas that require patterns recognition, such as word processing and natural languages. As the crucial indexing step is based on the recognition of words to classify them according to their importance, it was necessary to find a fast, effective and robust way to implement it. This reason motivates our choice of *Lex* as a working tool.

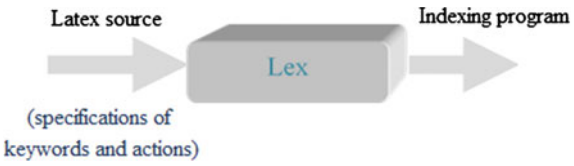

Fig. 1 Using Lex to produce the indexing program (step1)

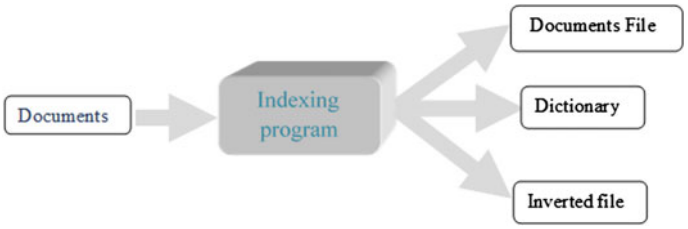

Fig. 2 Executing the indexing program for documents (step2)

Fig. 3 Executing the indexing program for queries

As shown in Fig. 1, lex generates automatically the indexing program from a source containing specifications of words and actions. Each word is described by a regular expression and is followed by an action composed by a fragment of a program. The action is executed each time the corresponding word is recognized during the indexing process.

Afterwards, the indexing program is executed to produce three output files: the dictionary, the documents file and the inverted file, which will be used in the matching process. Figure 2 illustrates the role of the indexing program. The dictionary contains all the relevant terms that appear in the documents as well as their total frequency and their positions in the two other files. The document file includes the documents with their terms, each document with its associated terms and their frequencies in the document. Finally, the inverted file contains all the relevant terms, each one followed by the list of the documents in which it appears in addition to its frequency in each document. Figure 3 shows how the indexing program generates the file of queries in the same manner as for documents. The query file is organized in the same way as the documents file that is, it includes the terms appearing in the query with their frequency. The dictionary helps to check the existence of the term in the indexed documents.

4 AC-IR Algorithm

In this section, we present the ant system algorithm called AS-IR designed for information retrieval. Let us first start with the description of the problem modelling. Based on the natural ants behaviour for finding food from a very large geographical space, the Ant System (AS) algorithm simulates this process for finding optimal solutions from a huge set of potential solutions. The ants move from the hive to a food source and when reaching the latter, they alert their congeners by means of a stigmergetic communication asking them for help to transport the food to the hive. According to animal psychology, this communication is performed thanks to the pheromone that the ants deposit on ground to orient the congeners to the place containing an important amount of food.

4.1 Solutions Encoding

A solution for the ant process will be a document since the core issue is to select from a huge collection those documents that share the maximum number of words with the query. Consequently, we need to evaluate documents during the search process in order to choose those that satisfy this condition. The similarity measure f as defined previously is appropriate.

4.2 Pheromone Table and Probabilistic Decision Rules

The ant algorithm includes several ant generations, each generation is composed of *NbAnts* ants. Two structures are needed to compute the ant algorithm, a table named *Phero* to store the pheromone amount yielded by the ants each time it builds a solution and a table called *sol* to save the best solution found by each ant. *phero[k]* corresponds to the pheromone amount associated with the document found by ant *k* and *sol[k]* is the best solution determined by ant k. Ants will construct new solutions using these structures, which represent a means of communication between the artificial ants. The tables are updated at each generation of ants. Besides, two variables namely *best* and *bestsol* are used to save respectively the best solution found during one generation and the best solution computed since the beginning of the process. Each ant starts building a solution from an initial solution *s* generated randomly. It then constructs a solution using a stochastic process. The ant chooses a solution from its neighborhood, with a probability computed as follows:

$$P(k) = \frac{phero[k]}{\sum_{j=1}^{j=NbAnts} phero[j]} \tag{1}$$

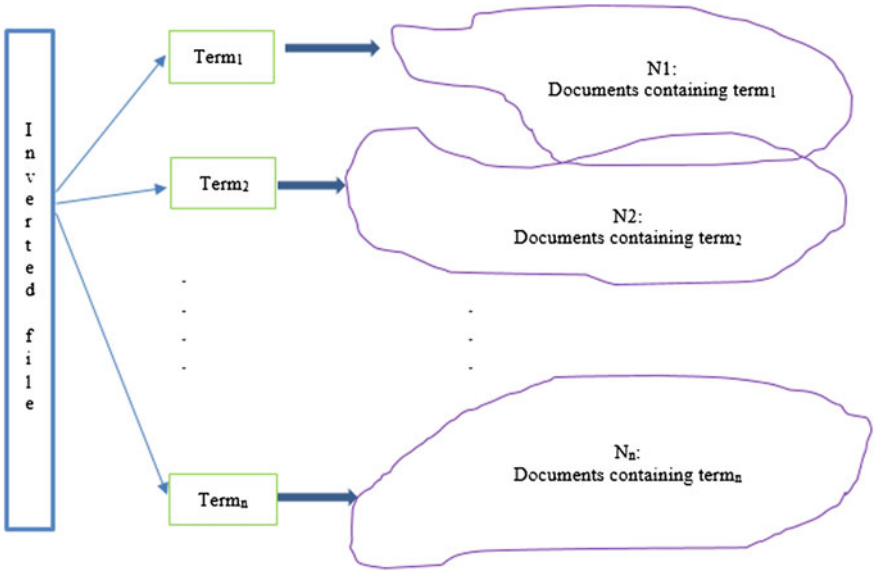

Fig. 4 The definition of neighbourhood of a document

The neighbourhood N_i of a document is the set of documents attached with $term_i$ in the inverted file as shown in Fig. 4. All these documents are neighbours because they share at least one term between themselves. One neighbourhood contains a huge number of documents where the AS algorithm is launched.

The pheromone information is initialized with a small value equal to 0.1 in order to simulate the fact that initially the real ants deposit a very small amount of pheromone on the ground when starting their space exploration. During the search, the pheromone amount, which represents the importance of the document, will be computed and associated to each document found by the ants. The AS-IR framework considers one sub-collection of the inverted file corresponding to one term. It is illustrated in Fig. 5 and outlined in procedure AS-IR().

4.3 Updating the Pheromone

The strategies of updating pheromone simulate the evaporation of natural pheromone followed by a production of this chemical substance. The evaporation phenomenon gives rise to rule (2) where the empirical parameter ρ belongs to the interval [0, 1] and simulates the evaporation rate. For online update performed at each generation of ants, the pheromone added is calculated according to rule (3) whereas for the offline update rule (4) is applied. Recall that *bestsol* is the best solution found during the previous iterations and *best* is the best solution of the current iteration.

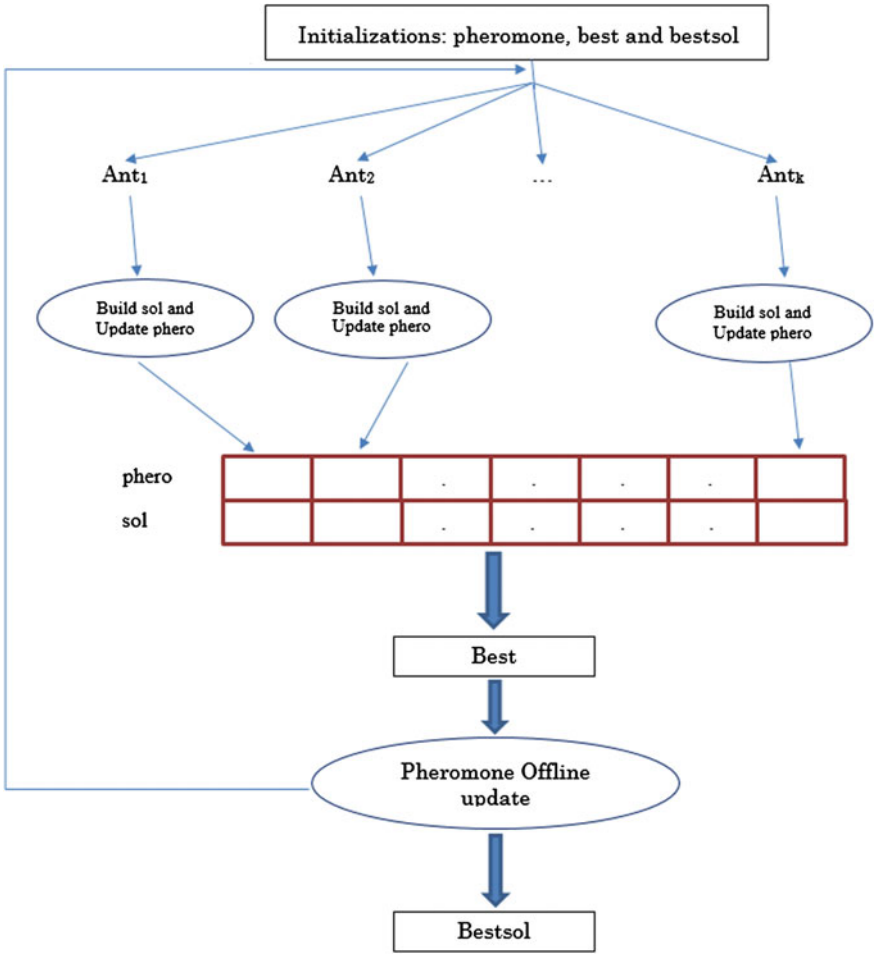

Fig. 5 A schematic view of the AC process

$$phero[k] = (1 - \rho) * phero[k] \tag{2}$$

$$phero[k] = phero[k] + \rho * f(s) \tag{3}$$

$$phero[k] = phero[k] + \rho * f(bestsol)/f(best) \tag{4}$$

4.4 Building and Improving a Solution

Each ant performs the task of exploring the best document in a sub-collection. The method designed for this aim is described through the procedure called build_AS. It merely chooses a solution from its neighbourhood with probability p defined in formula (1).

Algorithm 1 AS-IR

Input: N: a subcollection or neighbourhood of the Inverted File; Q: Query;
Output: bestsol: a document that is the most similar with Q;

```
1: procedure AS- IR
2:     for k=1 to NbAnts do phero[k]= 0.1;                    ▷ pheromone initialization
3:     end for
4:     select at random a solution s from N;                  ▷ a document namely s
5:     best := bestsol := s;
6:     for i=1 to MaxIter do
7:         for  k=1 to NbAnts do
8:             generate a random initial solution s;
9:             sol[k] = s;
10:            s' := build_AS (s) ;
11:            sol[k] := s';                                  ▷ update the best solution of k
12:            update the online pheromone phero[k] using formulas (2) and (3);
13:            if f(s') > f(best) then then best := s';       ▷ f is the similarity function
14:            end if
15:        end for
16:        if f(best) > f(bestsol) then bestsol := best;
17:        end if
18:        apply online-update of pheromone;
19:    end for
20:    return (bestsol);
21: end procedure
```

After its construction, each solution undergoes an improvement of its quality by applying the procedure tabu-search. In the tabu search procedure, the considered neighborhood is the one previously described. The intensification phase starts when the number of iterations without improving the solution quality reaches some limit. After applying the intensification strategy, a diversification technique is launched by choosing the less recently used moves and thus directing the search to new regions of the space. We use the variables *best_s*, which is the best solution found by the tabu process. And in order to set the stop condition we need also the variable, namely *no-improve*, which informs about the number of successive iterations without improvement of *best_s*. The variable *max-no-improve* is the maximum number of iterations without improvement before starting the intensification process. The procedure *tabu-search* outlined below calls both the procedure *neighbor(s)* that returns the best nearest neighbor of *s* which is not tabu nor satisfies the aspiration criterion and the procedure *update-t-length()*, which updates the tabu list length.

Algorithm 2 Build_AS

Input: N_s: the sub-collection containing the document s; Q: Query;
Output: best_s : a document with a better similarity;

```
1: procedure BUILD_AS(var s)
2:     draw at random a document i from N_s;
3:     compute p = P(k) using formula (1);
4:     generate at random a number r from [0, 1];
5:     if r > p then s = i;
6:     end if
7:     best_s = tabu-search(s);
8:     return(best_s)
9: end procedure
```

Algorithm 3 Tabu-search

Input: N_s: the sub-collection containing the document s; Q: Query;
Output: Best_s: the best document computed;

```
1: procedure TABU- SEARCH(var s)
2:     best_s = s;
3:     no-improve = 0;
4:     while (not stop condition) do
5:         s := neighbor(s);
6:         update-t-length();
7:         if (f(s) ≤ f(best_s)) then no-improve =: no-improve + 1;
8:         end if
9:         if (no-improve = max-no-improve) then
10:            no-improve := 0;
11:            best_s := Intensification (s);
12:            if f(s) > f(best_s) then best_s := s;
13:            end if
14:            best_s := diversification (best_s);
15:        end if
16:    end while
17:    return (best_s);
18: end procedure
```

5 ACS-IR Algorithm

The second designed ACO algorithm is the ant colony system (ACS) version. Its main difference with the previous one resides in the design of the probabilistic decision rules and the procedure of building solutions which is called build_ACS.

Algorithm 4 Build_ACS

Input: N_s: a sub-collection of the Inverted File containing the document s; Q: Query;
Output: best_s: a document with good performance;

```
1: procedure BUILD_ACS(var s: solution)
2:     generate a random variable q;
3:     if (q ≤ q0) then
4:         let s be the document with maximal similarity f computed using rule (5);
5:         put s in the tabu list;
6:     else
7:         choose a non tabu document s randomly;
8:         compute probability P(k) by rule (6);
9:         generate a random value r ∈ [0, 1];
10:        if P(k) > r then s := argmax_{j∈V_k} f(j);
11:        end if
12:        put s in the tabu list;
13:    end if
14:    best_s := tabu-search(s);
15:    return (best_s)
16: end procedure
```

q_0 is a tunable parameter and the pseudo-random-proportional rules are computed using respectively the probability of (5) or (6).

$$P(k) = \begin{cases} 1 \text{ if } f(k) = argmax_{j \in V_{sol[k]}}(phero[k]^{\alpha} heur[j]^{\beta}) for k = 1...NbAnts \\ 0 \text{ else} \end{cases} \tag{5}$$

$$P(k) = \frac{phero[k]^{\alpha} heur[j]^{\beta}}{\Sigma_{j=1}^{j=NbAnts} phero[j]^{\alpha} heur[j]^{\beta}} \tag{6}$$

The probability P(k) in (6) is computed using the quantity of pheromone and a heuristic function. α and β are empirical parameters and control respectively the importance of these two components. The heuristic part is calculated as follows:

$$heur[k] = max_{j \in V_{sol[k]}} f(j) \tag{7}$$

In other words, the ant decides stochastically to consider the best solution found in the neighborhoods of the solutions being treated during the current iteration when $q \leq q_0$ and a document drawn at random otherwise, unless the computed probability of formula (6) is greater than a generated random number r.

6 The Overall Algorithm

The ACO algorithms are used to search the sub-collections determined by the inverted file. The overall ant system algorithm called GAS-IR follows the classical schema of information retrieval process and can be outlined in Algorithm 5.

Algorithm 5 GAS-IR

Input: IF: Inverted File; Q: Query;
Output: best: a document that is the most similar with Q;
1: best = a document drawn at random;
2: **for** each term t of Q **do**
3: $N_t = IF(t)$; ▷ the subcollection of Inverted File of term t
4: let d be the document returned by AS-IR();
5: **if** $f(d) > f(best)$ **then** best = d ;
6: **end if**
7: **end for**

This way, only the documents that share at least one term with the query are searched and this leads to optimize the runtime.

The overall ant colony system algorithm GACS-IR is identical to GAS-IR except that it calls ACS_IR procedure instead of AS procedure.

7 Experimental Results

The two designed algorithms were implemented in Java on a personal computer. In order to test their performance, we conducted a series of extensive experiments. The first one consists in setting the empirical parameters that yield high solutions quality such as the ant colony size, the maximum number of iterations and the evaporation rate. A second step is undertaken in order to test the performance of the algorithms.

7.1 Benchmarks

The experiments were performed on well-known public benchmarks that are CACM and RCV1. The first one is of a small size while the other is of a significant size. The purpose of choosing such data is to analyse the impact of testing the developed algorithms on small and large size data.

7.2 Setting the Parameters

Figures 6 and 7 show examples of experimental results for setting respectively the maximum number of iterations for ACS-IR and the evaporation rate for AS-IR for CACM collection. The values for all the empirical parameters are summarized in Table 1 for CACM and in Table 2 for RCV1 respectively.

The parameters α and β are set respectively to 1 and 2 to give more importance to the heuristic component in the decision rule.

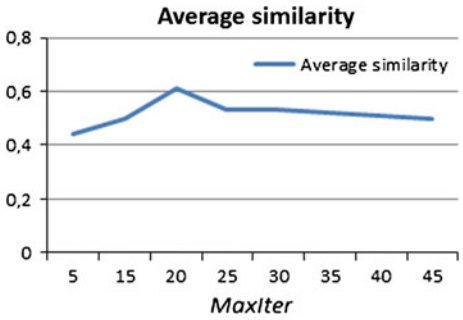

Fig. 6 Setting the maximum number of iterations for ACS-IR for CACM

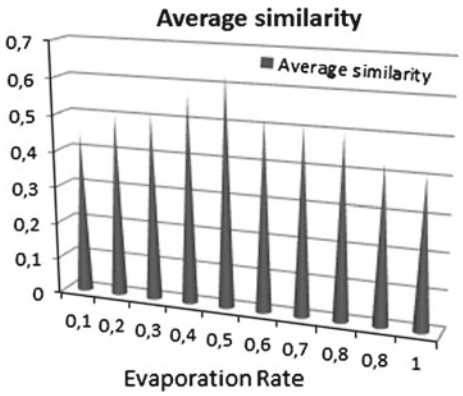

Fig. 7 Setting ρ, the evaporation rate for AS-IR for CACM

Table 1 Empirical parameters values for CACM

Parameter	AS-IR	ACS-IR
MaxIter	25	20
ρ	0.5	0.1
NbAnts	20	30

Table 2 Empirical parameters values for RCV1

Parameter	AS-IR	ACS-IR
MaxIter	200	150
ρ	0.5	0.1
NbAnts	50	50

7.3 Comparison of AS-IR, ACS-IR and CL-IR Algorithms

As the aim of this study is the design and the implementation of intelligent techniques to cope with large-scale information retrieval, it is not necessary to undertake comparisons between these methods using the precision and the recall measures because we are testing the efficacy of retrieval techniques rather than the validation of IR model.

A good performance of our approach implies a high similarity of the best found document and a runtime lower than the one yielded by the exact method. This remark leaded us to use the ratio between the similarity and the runtime to evaluate algorithm efficiency. A high value of this ratio in comparison with the exact approach translates a reduction in runtime and a similarity relatively stable.

An algorithm named CL-IR based on a classical IR approach [14] was developed for comparison purposes. It computes the optimal solution since it proceeds with an exhaustive search. The following series of experiments were performed in order to analyze the behavior of the algorithms AS-IR and ACS-IR relatively to CL-IR. We vary the collection of documents and we compare the performance of the three algorithms in terms of similarity and runtime.

7.3.1 Results of Tests on CACM

Table 3 shows the similarity values as well as the runtime computed for CL-IR and AS-IR for the CACM collection. The schematic view of the time values is illustrated in Fig. 8 and those of the solution performance in Fig. 9. We observe a slight superiority of the classic approach in terms of solution quality. However, AS-IR is faster while performing almost as well as CL-IR.

7.3.2 Results of Tests on RCV1

The same experiments as the ones performed previously are repeated for RCV1 collection. In Table 4 We report the achieved numerical results of the similarity and the computation time for four queries. Figure 8 illustrates the gain in time we obtain from executing AS-IR and ACS-IR. Figure 9 shows the similarity that illustrates the real performance of the different algorithms.

Table 3 AS-IR and CL-IR comparison

	query	doc	f	Time (ms)	sim/time
CL-IR	1	2 945	1.63	10.3	0.16
	2	1 030	1	4.5	0.22
	3	3 012	1.18	12.1	0.09
	4	1 621	3.36	3.2	1.05
AS-IR	1	2 945	1.63	7.9	0.21
	2	137	1	3.8	0.26
	3	1 629	1	6.1	0.16
	4	3 142	3.09	2.3	1.34

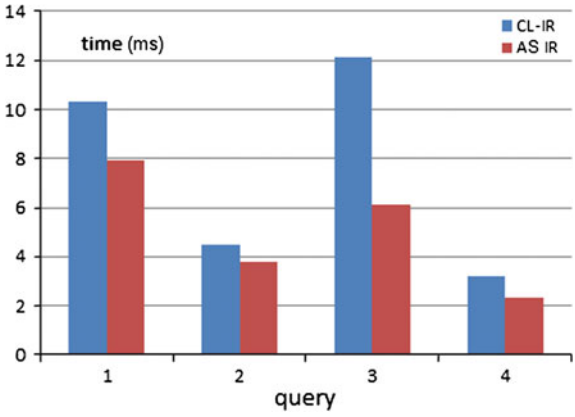

Fig. 8 Comparison between AS-IR and the CL-IR runtime for CACM

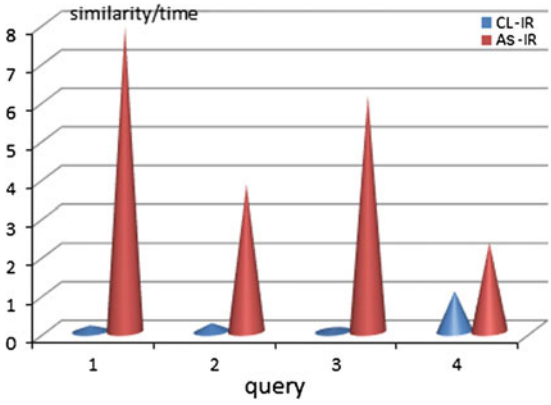

Fig. 9 Performance comparison between AS-IR and CL-IR for CACM

According to these results, we observe the real gain in runtime the algorithms AS-IR and ACS-IR achieved relatively to CL-IR. At the same time, the solution quality of these algorithms expressed by the similarity is competitive with the exact method.

Table 4 Comparison between AS-IR, ACS-IR and CL-IR for RCV1

	query	doc	f	Time (ms)
CL-IR	1	4040	0.31	206.51
	2	93 576	0.62	216.23
	3	511 263	0.79	215.61
	4	536 823	0.56	207.09
AS-IR	1	4 040	0.31	4.65
	2	697 610	0.63	70.23
	3	382 463	0.77	91.63
	4	536 823	0.56	79.31
ACS-IR	1	4 040	0.31	2.16
	2	697 610	0.63	38.09
	3	382 463	0.77	51.86
	4	536 823	0.56	47.30

7.3.3 Results of Testing on the Large Random Collection

The large collection was generated at random in order to test larger data sets. Like for the previous collections, the same experiments were conducted on this collection and the achieved results are shown in Table 5, Figs. 12 and 13. The previous yielded observations are more consolidated by the obtained results of these tests (Fig. 10).

Finally, we can notice through the results of these sets of experiments that the impact of ACO algorithms in reducing computational time for information retrieval is more perceptible on large data sets (Fig. 11).

Table 5 Comparison between AS-IR, ACS-IR and CL-IR for the largest collection

	query	doc	f	Time (s)
CL-IR	1	5 321 985	2.73	35.92
	2	1 298 675	1.97	59.86
	3	7 368 942	2.40	44.98
	4	8 691 245	2.87	29.85
AS-IR	1	6 543 299	2.69	2.45
	2	1 708 043	1.89	2.74
	3	3 876 109	2.33	3.18
	4	8 691 245	2.87	1.52

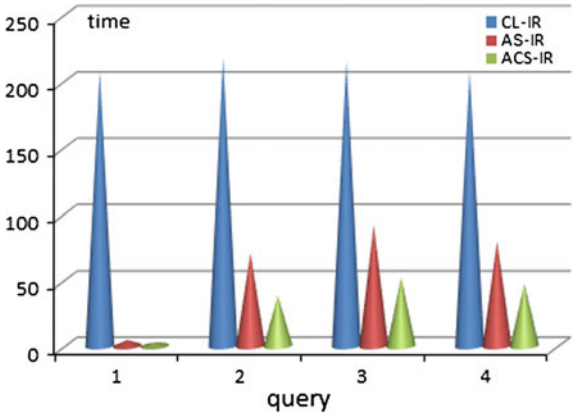

Fig. 10 Time comparison between AC-IR, ACS-IR and CL-IR for RVC1

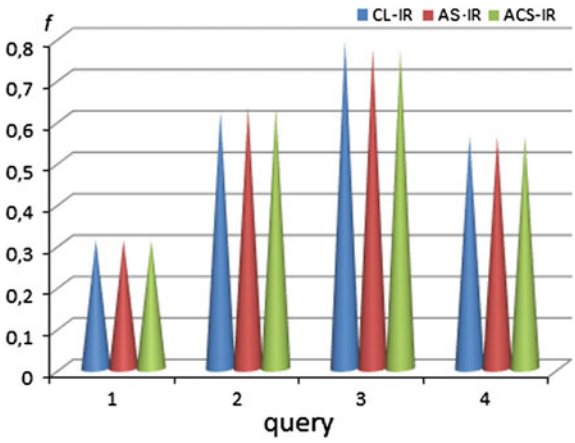

Fig. 11 Performance comparison between AS-IR, ACS-IR and CL-IR for RCV1

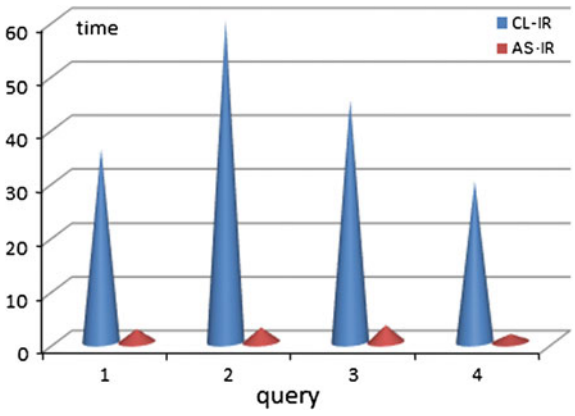

Fig. 12 Time comparison between AS-IR and CL-IR for the largest collection

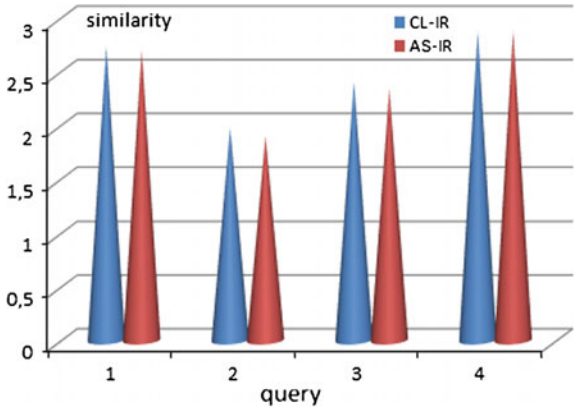

Fig. 13 Performance comparison between AS-IR and CL-IR for the largest collection

8 Conclusion

In this paper, two ACO algorithms named respectively AS-IR and ACS-IR were designed for information retrieval in the large-scale context. The aim of this study is the adaptation of heuristic search techniques to IR and their comparison with classical approaches. Experimental tests were conducted on the well known CACM and RCV1 collections dedicated to researchers for comparison purposes and also on larger random benchmarks.

Through the performed experiments, we observed that both ACO algorithms perform almost as well as the classic approach in terms of solution quality. However, for large-scale data sets their runtime is extremely interesting as it is much faster than the runtime of the conventional approach. In other words, both AS-IR and ACS-IR are scalable for information retrieval unlike the traditional IR techniques. When comparing AS-IR to ACS-IR, it appears that ACS-IR outperforms AS-IR.

As a future work, we think of improving the response time by exploiting the GPU technology for computing the ant tasks in parallel.

Benckmarks

CACM

CACM is a collection of article abstracts published in ACM journal between 1958 and 1979. Table 6 shows its characteristics. Although the designed algorithms aim at searching large scale collections of documents, they work in a good way on this small collection and outperform the exact algorithms in terms of runtime. However the collection remains too small to observe the real impact brought by the proposed approach.

Table 6 The CACM collection dimension

CACM	
Number of documents	3 204
Number of terms	6 468
Average document size	2Ko

Table 7 The RCV1-V2/ LYRL2004 collection dimension

RCV1-v2/LYRL2004	
Number of documents	804 414
Number of terms	47 236
Average document size	2Ko

Table 8 The RANDOM collection dimension

Random collection	
Number of documents	10 000 000
Number of terms	50 000
Average document size	2Ko

RCV1

Reuters Corpus Volume I (RCV1) is a collection of more than 800.000 documents representing archives published by Reuters, Ltd. It is now publicly available for use by researchers, Table 7 shows its parameters sizes.

The Random collection

A larger random collection has been created automatically, Table 8 shows its characteristics.

References

1. Dorigo, M., Di Caro, G., Gambardella, L.M.: Ant algorithms for discrete optimization. Artif. Life. **5–3**, 137–172 (1999)
2. Van Rijsbergen C.J.: Information Retrieval. Information Retrieval Group University of Glasgow, Glasgow (1979)
3. Bulheimer, B., Hartl, R.F., Strauss, C.: A new rank based version of the ant system, a computational study. Technical Report POM -03/97, Institute of Management Science, University of Vienna (1997)
4. Cordon, O., Deviana, I., Herrera, F., Moreno, L.: A new ACO model integrating evolutionary computation concepts: the best-worst ant system. In: From Ant Colonies to Artificial Ants, ANTS 2000, pp. 22–29 (2000)

5. Dorigo, M., Gambardella, L.M.: Ant algorithms for the traveling salesman problem. Biosystems **43**, 73–81 (1997)
6. Hsinchun, C.: Machine learning for information retrieval: neural networks, symbolic learning and genetic algorithms. J. Am. Soc. Inf. Sci. **46**, 194–216 (1995)
7. Doerner, K., Hartl, R.F., Reimann, M.: Cooperative ant colonies for optimizing resource allocation in transportation. LNCS Springer Verlag **2037**, 70–79 (2001)
8. Colorni, A., Dorigo, M., Maniezzo, V., Trubian, M.: Ant system for job-shop scheduling. Belgian J. Oper. Res. Stat. Comput. Sci. **34–1**, 39–53 (1994)
9. Gambardella, L.M., Taillard, E., Dorigo, M.: Ant algorithms for the QAP. Technical Report IDSIA 97–4. Lugano, Switzerland (1997)
10. Manning, C.D., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
11. Pathak, P., Gordon, M., Fan, W.: Effective Information retrieval using genetic algorithms based matching functions adaptation. In: 33rd IEEE HICSS (2000)
12. Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**, 513–523 (1988)
13. Stutzle, T., Hoos, H.: Improving the ant system: a detailed report on the MAX-MIN ant system. In: ICANGA, pp. 245–249. Springer (1997)
14. Baeza-Yates, R., Ribiero-Neto, B.: Modern Information Retrieval. Wesley Longman Publishing Co., Inc., Boston (1999)
15. Mahdavi, M., Chehreghani, M.H., Abolhassani, H., Forsati R.: Novel meta-heuristic algorithms for clustering web documents. Appl. Math. Comput. **201**, 441–451 (2008)
16. Zhengyu, Z., Xinghuan, C., Qingsheng, Z., Qihong, X.: A GA-based query optimization method for web information retrieval. Appl. Math. Comput. **185**, 919–930 (2007)
17. Lesk, M.E., Schmidt, E.: Lex—A lexical analyzer generator. UNIX time-sharing system: UNIX Programmer's Manual, 7th edn, vol. 2B (1975)

Hosting Clients in Clustered and Virtualized Environment: A Combinatorial Optimization Approach

Yacine Laalaoui, Jehad Al-Omari and Hedi Mhalla

Abstract This paper presents a global approach to deal with the problem of allocating a set of clients to a common pool of multiple clusters based on number of connections to advance resources management in virtual environment. To optimize resources allocation in Applications Services Provider's data-centers, we propose a combinatorial optimization look to the problem. First, we describe the corresponding integer mathematical model. Then, we use the IBM CPLEX solver to solve to optimally this problem.

Keywords Hosting clients · Cluster · Virtual machine · Combinatorial optimization

1 Introduction

In the world of IT, each company needs technical skills such as system and network administrators, security experts, and more. Also, it needs servers and powerful machines (CPU and Memory) to host running applications in data-centers. The problem is that the budget increases based on the number of employees and acquired hardware. The recent trend is that non-specialized companies in the IT domain—called clients in this paper- are outsourcing their IT services to new specialized enterprises namely Applications Service Providers (ASP). ASPs host clients and their applications in a sophisticated infrastructures. Clients connect remotely to

Y. Laalaoui (✉) · J. Al-Omari
Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia
e-mail: y.laalaoui@tu.edu.sa

J. Al-Omari
e-mail: jsalomari@tu.edu.sa

H. Mhalla
Department of Mathematics and Statistics, The American University of the Middle East, Eqaila, Kuwait
e-mail: hedi.mhalla@u-picardie.fr

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_3

ASPs infrastructure using simple PCs or disk-less stations through high speed Internet lines. This reduces significantly the number of employees and acquired hardware. The main objective is to guarantee high performance of hosted applications in ASPs' infrastructures according to a service level agreement. All IT services such as hardware, applications, network and security issues will be handled by ASPs' professional staff. Therefore, the market of hosting clients has been climbing recently.

The increased number of clients in the ASP infrastructure has also increased the complexity of IT operations such as software installation, updates, network performance and monitoring. To handle this complexity, the automation of IT operations has been proposed in industry and academia that consists of adopting the most recent and important innovations such as virtualization and cluster technologies.

The main objective of this paper is to help ASP's professional staff in the efficient hosting of clients by optimizing the use of available resources.

1.1 Hardware Virtualization Technology

Hardware Virtualization means the creation of a virtual machine that works like a real computer with an operating system [1, 2]. The main objective of such technology is to increase the exploitation of the current powerful hardware machines (multi-core, multi-threading, huge amount of RAM space, high speed network cards, …etc).

Figure 1 shows an example of running applications using virtualization technology in case of only one computer with one processor.

This computer is running 8 applications and each 2 applications are running in a separated operating system (OS).

Such operating systems are called also Guest OSs. Each OS is running on one Virtual CPU provided by the virtualization layer where the latter is called the *hypervisor*. The virtual layer is an interface between the real hardware and different VMs.

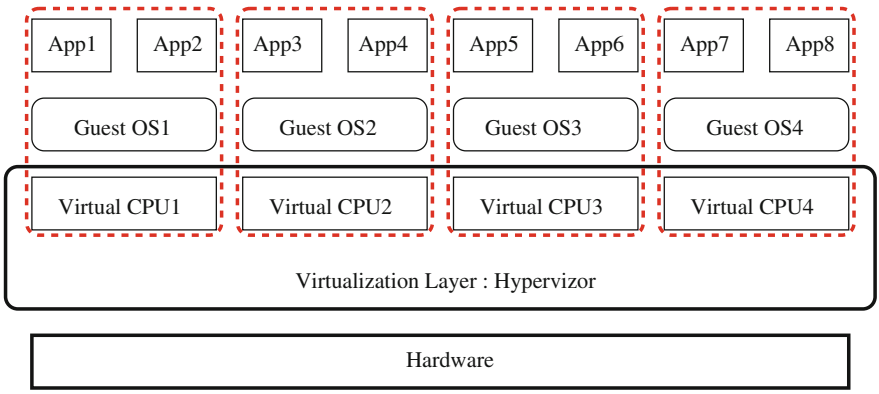

Fig. 1 Example of 4 virtual machines on one physical machine

The number of VMs to be installed is constrained by the amount of available resources (RAM, processor(s) and hard-disk). It is clear that the amount of available resources in the hardware level should not be fully utilized. Otherwise, the virtualization performance would be lost.

- **Xen**: There are two types of hypervisors: The first type is called the "bare-metal" which runs directly on top of the physical Machine. The second type runs within an operating system. Xen is a type1 hypervisor that allows virtualization of multiple virtual guest operating systems simultaneously on a single physical host. Xen supports para-virtualization and full virtualization. Para-virtualization uses enlightened guests (modified guest OS) to make special calls to hypervisors in order to access the hardware and network resources. There is an open source community which provides the XEN hypervisor under GNU General Public License (GPLv2). This community supports vendors including IBM, HP, Dell, Cisco, AMD, SUN, Ret Hat, Mellanox, and others [3]. Also, there are commercial versions of the XEN hypervisor such as the XENServer provided by Citrix [4].
- **VMWare**: VMware is one of the major providers of virtualization and cloud computing software for x86-compatible computers. It offers multiple virtualization solutions for desktop—as well as for server-virtualization based ESX/ ESXi bare metal hypervisor. The architectural difference between ESX and ESXi servers is that the ESXi server allows VMWare agents to run directly on the VMKernal without the use of service console which is not the case with ESX server. VMWare provides another professional server virtualization product which supports functionality such as auto deploy, image builder, firewall which enables more configuration and security possibilities [5].
- **KVM**: The Kernel-based Virtual Machine (KVM) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). The kernel-based virtual machine (KVM) was first introduced and published under General Public Licenses (GPL-v2) in 2006. It is included in the Linux kernel version 2.6.20. This virtualization solution is designed as kernel modules. KVM itself does not do any emulation of hardware components, however it requires the use of Quick Emulator (QEMU) to provide emulated hardware to the guest OS [6].

1.2 Cluster Computing Technology

A cluster is set of interconnected stand-alone nodes working together as a single system to share computing resources providing High Availability (HA), load balancing and parallel processing [7]. The concept of clustering solutions intends to provide instant recovery with minimal downtime for applications in case of hardware or software failure. It is based on virtual infrastructure management to reduce management complexity of aggregating standalone hosts into single cluster with pooled resources [8]. Resource pools permit delegate control over resources of a cluster by separating resources from hardware (hosts). A Virtual Machine (VM) can be given resources

from a resource pool within a cluster, rather than be tied to a specific host. Cluster technology relies on aggregated hardware resource such as processing power, memory, storage devices, and network performance (throughput and latency) [5]. This technology has become cost effective High Performance Computing (HPC). More and more users and businesses become dependent on Applications Services Providers (ASP) and demand a quality of service (QoS). Clustering Technique is widely adopted to improve performance and minimize service disruption or failure [9, 10]. This HA infrastructure continuously provides dynamic monitoring of physical servers using installed agent on each server. Furthermore, it leverages shared SAN storage to replicate data in multiple locations to achieve mobility and automated Disaster Recovery as illustrated in Fig. 2.

1.3 Clients Hosting Problem

When a client requests a hosting service, then the hosting company (ASP provider) asks the number of simultaneous connections and the needed applications to be installed. After that, the ASP would surely creates one virtual machine and not a dedicated physical machine. The new virtual machine would then be loaded into one cluster since the ASP has a sophisticated infrastructure with many connected clusters.

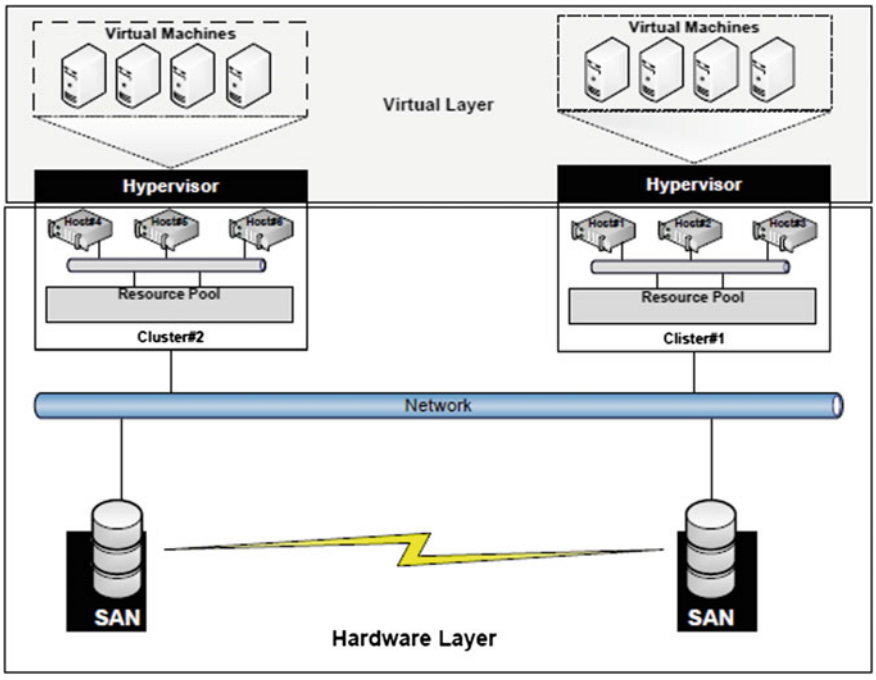

Fig. 2 High availability cluster infrastructure

The process of new demand assignment to a cluster starts by conducting a performance evaluation of all clusters in the virtual and clustered environment. This evaluation is based on resources consumption (Mem/CPU) to compare clusters performance and utilization in order to select the best cluster candidate. The new demand is analyzed based on the new demand number of connections and resource optimization. The new VM is compared with all the VMs on the selected cluster to make sure that the new demand is within the acceptable range. If the selected cluster is the best candidate, the new VM will be added to the cluster; otherwise, the process of selecting best cluster has to be preformed again. After assigning the new VM to the selected cluster a reevaluation of performance and resource utilization must be performed to determine the status of selected cluster. If the cluster is no longer the best candidate to host the new VM, then the process of selecting new cluster has to be conducted again to move the new VM to a different cluster. In fact, this process of assigning a new arriving client to a specific cluster is an on-line approach and it reflects the current real-life assignment method. In the present paper, we describe an off-line approach for the following reasons:

- Off-line approaches offer a complete knowledge about the input problem while on-line approaches don't have the complete knowledge about coming demands.
- In computing point of view, on-line approaches are based on algorithms that have a polynomial time complexity. But, the problem of assigning clients to clusters is an NP-Hard problem and algorithms with a polynomial time that solve this type of problems don't exist [11].

2 Resource Allocation Problem

Resource allocation problem in virtual environment has gained significant consideration due to the growing market of virtualizarion and cluster computing. To understand the latter problem we classify the recent advances in computing technologies into three main layers:

1. Software as a Service (SaaS) at application layer. Researches related to this layer attempted to maximize the service provider's revenue by minimizing the total infrastructure (or hardware) cost and Service-Level-Agreement (SLA) violations [12]. Authors in [13] worked on resources allocation related to the type of systems, specifically to multimedia systems. The latter work attempt to optimize the global workload assignment and resource allocation at VM level under the service response time constraint.
2. Platform as a Service (PaaS) at operating system layer. In [14], Ferretti et al. proposed an architecture related to PaaA that distributes the computational load across the platform resources, and monitors the QoS the platform delivers. The whole architecture is based on SLA to decide the suitable resource allocation in cloud environments. Authors in [15] proposed a smiliar architecture but with multiple levels of QoS.

3. Infrastructure as a Service (IaaS) at hypervisor layer. Research in IaaS utilizing virtual machines resource allocation that run over the cloud [16]. The effort of many researchers focused on power saving of VM resource allocation to improve energy efficiency of the data center in the mean time comply SLAs [17, 18].

To the best of our knowledge the human factor is completely missing in all researches on resource optimization in IT platforms. Involving the human factor in current computing systems is compulsory simply because the autonomy of the later systems has not yet been achieved.

3 Helpful Optimization Problems and Tools

3.1 2-Dimensional Bin-Packing Problem

Bin-Packing is one of the NP-Hard problems in optimization area [19]. This problem is a knapsack problem that consists of packing all items to knapsacks so that the total number of knapsack is minimized. In One-Dimensional Bin-Packing, each item has a weight and the cumulative weight of items assigned to one knapsack should not exceed the capacity of that knapsack. The 2-Dimensional Bin-Packing Problem adds one more parameter for both items and knapsacks, for example the volume[1] [20] and one more constraint to take into account the new parameter.

This problem has been studied extensively and widely applied in many applications domains such as minimizing the number of processors to get a feasible job assignment in real-time systems [21], robot selection with workstation assignment [22] and file placement for multi-device storage system [23].

The 2-Dimensional Bin-Packing Problem can be used also to model the problem of hosting clients in a set of clusters. Each client has the couple of needed processors (or cores) and the amount of RAM. The final target is minimizing the number of cluster to host all clients. The reduction of the cluster leads to the reduction of the cost of extra clusters. Further, it reduces also the energy of those extra servers.

3.2 The Max-Min Problem

The *Max-Min* assignment problem is a combinatorial optimization problem whose aim is to balance the number of assigned elements. For example, in *knapsack sharing problem* (KSP) [24–26] there are items to be assigned into a set of knapsacks. The objective function in KSP is to maximize the number of items within the knapsack that have the smallest number of items.

[1] In literature, items are rectangles and they have the couple of Width and Height.

The *Max-Min* assignment problems, are known to be NP-hard and they received a lot of attentions in computing and Operation Research studies [27–31].

The *Max-Min* assignment problem could be used in the current study in order to balance the use of resources within each cluster. For example there are two clusters; the final assignment should not contain the distribution of 1 and 7 used processors between both clusters if another assignment with 4 processors for each cluster is possible.

3.3 Data-Set and Solving Tool

The present paper describes an approach to solve a real-life problem of hosting clients in a set of clusters in a virtualized environment. The used data-set is taken from a leading Applications Service Provider in USA. The ISP name has been intentionally hidden for confidentiality reason. The used data-set contains 895 clients with different demands. Along the present paper, we divide this data-set into 7 problem instances with 10, 20, 40, 80, 160, 585 and 895 clients in order to enhance the experimental work. All these clients will be assigned to a set of identical clusters where the each cluster is known to have a set of processors with a total of 12 cores and 98291 MB of RAM space.

IBM CPLEX is one of the tools widely used to solve combinatorial optimization problems [32]. IBM CPLEX is an exact solver that uses Branch-and-Bound techniques to search for desired solutions. In the present study, we used IBM CPLEX solver version 12.2 on Laptop MS Windows machine with Intel CORE i5 processor and 6 GB of RAM.

3.4 Branch-and-Bound Search

Branch-and-Bound is a widely used search method in AI and OR areas [33, 34]. It is an exact technique that is able to find the optimal solution if one exists. Branch-and-Bound methods are also able to report failure if the acceptable solution doesn't exists in the state-space. The search process uses a tree where the size of the latter data-structure increases exponentially in accordance to the input problem instance size. The idea behind Branch-and-Bound methods is the use of a problem-dependent bounding function to prune non-promising branches in the search tree and to avoid exhaustive exploration of the state-space. Often, the difficulty of using this type of methods lies in the definition of such bounding function.

The remainder of this paper is organized as follows: Section 2 describes the problem; Section 3 is devoted to the description of the proposed integer programming models with the resolution using IBM CPLEX exact solver; Section 4 includes discussions and the paper is concluded in Sect. 5.

4 Proposed Approach

The problem of hosting clients applications is an assignment problem that consists of assigning one client demand to one specific cluster among a set of available clusters. Each cluster is a set of networked machines (Virtual Machines) managed using a virtualization system (the hypervizor). The cluster capacity is expressed in terms of CPU (number of cores) and RAM space while the Hard Disk space is a common storage device. Each demand from a client is measured in terms of number of simultaneous connections to the system. The objective is to assign all clients' demands to available clusters so that the use of resources is maximized with respect to clusters capacities (Fig. 3).

The problem of hosting clients applications consists, typically, of three phases:

- **Phase 1: System specification**: this phase consists of gathering all information about all clients namely the number of simultaneous connections to the system.
- **Phase 2: Covert demands to (#Cores, #RAM)**: this phase consists of converting clients demands from simultaneous connections to a corresponding amount of CPU and RAM. Tables 1 and 2 are used during this phase. It is worth to notice that the data shown in these tables have been taken from a real-life industrial and it often fit the clients' needs. Each client is assigned firstly a virtual machine with the following characteristics: RAM space, processor (one or more cores) and a

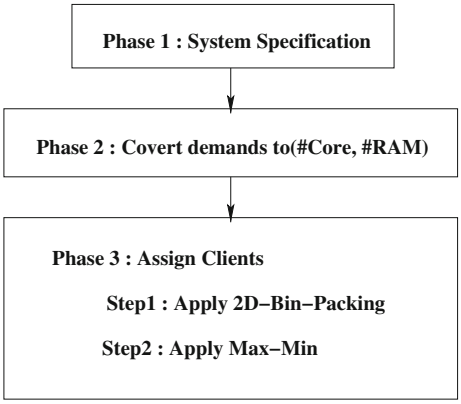

Fig. 3 Clients assignment approach

Table 1 # connections versus allocated CPU

# connections	#cores	CPU freq (Mhz)
1–60	1	2000
61–99	1	2000
100–159	1	2500
160–300	2	4000
301–450	3	6000
451–700	4	8000

Table 2 # connections versus allocated RAM

# connections	Configured RAM	RAM Reserved
1–60	1024 MB	512 MB
61–99	1792 MB	896 MB
100–159	3072 MB	1536 MB
160–300	sold conns * 20MB	Configured RAM * 0.5
301–450	sold conns * 20MB	Configured RAM * 0.5
451–700	sold conns * 20MB	Configured RAM * 0.5

Hard disk space. These characteristics are allocated according to the number of simultaneous connections as it is shown in Table 1. For example, if the maximum number of connections to the system is between 1 and 60 connections per second, then only one core with 2000 Mhz frequency (Table 1) plus amount of RAM equal to 1 GB often suffices (Table 2). Notice that the core frequency column in Table 1 is not considered in the present study and it is subject to future studies. This means that all cores are assumed to have the same frequency in this paper.
As it can be seen from Table 2, Configured RAM column is the allocated amount of RAM space for each client. This value is either fixed or it is a function of the number of sold connections. The number of sold connections is exactly the maximum number of PC/Workstations present in the client's side and that can connect simultaneously to the hosting infrastructure. For example, if the number of sold connections is 380, then the corresponding RAM space is 380*20 which is equal to 5600 MB. The factor 20 MB often fits well clients with a demand above 160 simultaneous connections.
The column called RAM Reserved is the amount of additional RAM space for each client to prevent possible I/O disk swappings. This column is also a function of the Configured RAM result once the number of sold connections exceeds 160. Once the client initial characteristics have been fixed, then the relevant cluster to host this client should be determined which is the task of the proposed global hosting approach.

- **Phase 3: Client Assignment**: this phase consists of two steps, applying the 2-Dimensional Bin-Packing solver to determine the minimum number of needed clusters and then applying the Max-Min solver to distribute fairly the number of used cores between clusters.

5 Integer Programming Models

The mathematical formulation of the problem is as follows:

- Given a set of n clients where each client j must be assigned to a cluster i. All clients have to be allocated to at most n clusters.

- Each client j must be assigned the couple of CPU and RAM (cpu_j, ram_j) where cpu_j is the number of cores and ram is the amount of RAM space, in MB, to be assigned. For simplicty reason, we let ram_j be the configured RAM space for the client j.
- We assume that all processors and their corresponding cores are homogeneous.
- We assume that all clusters are similar and each one has at most tot_cpu cores and tot_ram as the RAM space.

5.1 Minimizing the Number of Clusters

The problem of maximizing the use of resources can be seen as the problem of minimizing the set of clusters to be used to host all clients. Each client has the couple (cpu_j, ram_j). The hard disk is disregarded from the optimization function because it is located out of the cluster. The objective function noted m to be optimized is defined as follows:

$$m = minimize \sum_{i=1}^{n} y_i \tag{1}$$

$subject\ to$:

$$\sum_{j=1}^{n} cpu_j \times x_{ij} \leq tot\_cpu \times y_i, \ \forall i \in \{1, \ldots n\} \tag{2}$$

$$\sum_{j=1}^{n} ram_j \times x_{ij} \leq tot\_ram \times y_i, \ \forall i \in \{1, \ldots n\} \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \ \forall j \in \{1 \ldots n\} \tag{4}$$

$$y_i \in \{0, 1\}, \qquad i \in \{1 \ldots n\}. \tag{5}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j \in \{1 \ldots n\}. \tag{6}$$

where x_{ij} takes 1 if the j^{the} demand is assigned to i^{the} cluster and 0 otherwise. The result of this model is to get the minimum number of clusters, m, needed to host all clients.

Table 3 shows obtained results using IBM CPLEX solver on 6 problem instances taken from a real data-set. CPLEX solver is capable to find the optimal number of cluster to assign 895 clients within a time slot less than 400 s.

Table 3 Minimum number of needed clusters

Problem instances	#Clients	m	Time (s)
I1	10	1	0.03
I2	20	3	0.02
I3	40	5	0.09
I4	80	10	0.17
I5	160	15	0.58
I6	585	61	8.81
I7	895	93	395

5.2 Heavy Clients Distribution

First, let us introduce the following definition:

Definition 1 (*Heavy Client*): A heavy client is a client that consumes much resources (Processors and RAM).

It is clear that the situation in which many heavy clients are present in one specific cluster would increase the consumptions of available resources and the network traffic. This situation would degrade drastically the performance of the whole infrastructure. Thus, there is a need to avoid such situation by the distribution of heavy clients among available clusters. For example, in Fig. 4a, there are three clusters one cluster is assigned all heavy clients (red filled circles) while the remaining clusters are hosting only light clients. Thus, it would be much better to distribute heavy clients among the three clusters as it is shown in Fig. 4b. Since there is no prior knowledge about arriving clients, heavy clients are determined after the first assignment. Often, There are less heavy clients than the total number of hosted clients.

In order to force the allocation of heavy clients to different clusters, we introduce the notion of *Disjunctive Constraint* and it is defined as follows:

Definition 2 (*Disjunctive Constraint*): If two clients j_1 and j_2 are not allowed to be allocated to the same cluster, then j_1 and j_2 are disjunctive.

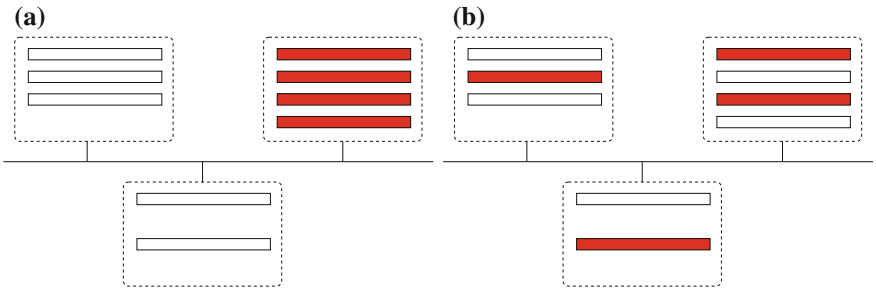

Fig. 4 Example of 3 clusters with heavy and light clients. *Filled rectangles* are heavy clients while *empty rectangles* are light clients. **a** Bad distribution. **b** Good distribution

Disjunctive constraints should be specified between heavy clients only. To this end, we introduce the variable H that takes either 0 or 1 to specify whether the corresponding client is heavy (value 1) or not (value 0).

$$H_j \in \{0, 1\}, \qquad j \in \{1 \ldots n\}. \tag{7}$$

It is worth to note that the variable H is an input data by which the model recognizes heavy clients. It is also worth noting that not each heavy client should be assigned to one separate cluster. This is not practical in the sense of maximizing the use of available resources and reducing the cost of extra clusters. Therefore, we intrduce the notion of *K-Disjuntivity Constraint* to fix the degree of disjuntivity allowed in each cluster.

Definition 3 (*K-Disjunctivity*): Let K be a non-negative integer value, a K-Disjunctivity Constraint is the fact of allowing at most K different heavy clients to be allocated to the same cluster.

Now, the problem of heavy clients distribution could be solved using the above model and the extra constraints shown in the following equation:

$$\sum_{j=1}^{n} H_j \times x_{ij} \leq K, \ \forall i \in \{1, \ldots n\} \tag{8}$$

where K is the maximum number of heavy clients allowed in one cluster. This parameter is an input to the data-model and it is set by the system administrator.

The optimal solution to the example shown in Fig. 4b is two heavy clients instead of four in Fig. 4a.

Table 4 shows results when the maximum number of heavy clients allowed per cluster are fixed to 1, 2 and 3. This tables shows also the total number of heavy clients for each problem instance (column called Hs). Notice that the total number of heavy clients per cluster has been measured using the number of cores requested

Table 4 CPLEX results when heavy clients are considered during the assignment process

			K=1		*K=2*		*K=3*	
Problem instances	*n*	#Hs	*m*	Time	*m*	Time	*m*	Time
I1	10	1	1	0.06	1	0.16	1	0
I2	20	6	6	0.09	3	0.16	3	0.03
I3	40	13	13	0.41	7	0.53	5	0.05
I4	80	17	17	5.54	10	0.23	10	0.19
I5	160	9	15	0.61	15	0.89	15	1.23
I6	585	84	84	438	61	155	61	149
I7	895	130	130	402	93	451	93	429

Time is measured in seconds

by each client. If the number of cores for a given client is above one, then this client is considered as a heavy client. This choice respects well Definition 1.

In practice trial/error paradigm is used to detect heavy clients. Firstly, assign a client to a cluster and then observe its system load. If this client has reached a high system load, then mark this client as a heavy one.

The clear result from Table 4 is that when the K is great (K equal to 1 or 2), then the resulting number of clusters is small. But, when K is set to 1, then the resulting number of clusters needed is at least equal to the number of heavy clients because each client from the heavy set must be hosted in one separated cluster.

In computational intelligence point of view, when K is set to 1, then the problem of assigning clients to clusters is very difficult to solve. A very long time is needed to reach the optimal solution.

5.3 Balancing the Use of Resources

The encountered problem of the described integer mathematical model is that the number of cores distributed among clusters is not balanced. For example, if we have 3 clients j_1, j_2 and j_3 with 1, 3 and 1 cores demands respectively. The above model can distribute j_1 and j_3 in one cluster and j_2 in one more cluster. The result is that the first cluster is assigned only 1 core while the second cluster is assigned 4 cores as shown in Fig. 5a. Therefore, it would be much better if the assignment is done

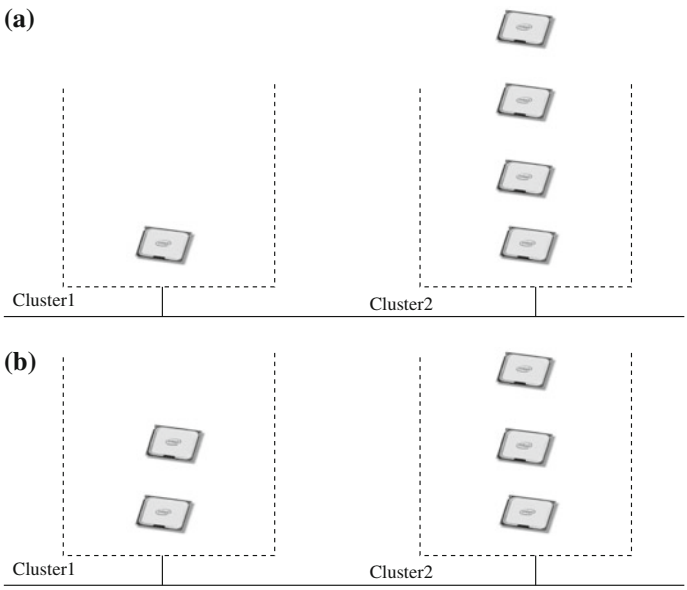

Fig. 5 Example of used cores distribution between two clusters. **a** Bad distribution of used cores. **b** Good distribution of used cores

as follows: clients j_1 and j_3 should be assigned to the first cluster and the client j_2 should be assigned to the second cluster. The result is this case is an almost balanced number of cores for bother clusters (2 cores for the first cluster and 3 cores for the second cluster) as it is shown in Fig. 5b.

Notice that in this example, there is no other possible assignment that can lead to the increase of used cores in the first cluster. The maximization of the minimum number has led to the assignment of 2 cores least for each cluster.

In Combinatorial Optimization point of view, the issue of unbalanced assignment can be seen as a problem of Maximizing the Minimum or Max-Min Problem (MMP). The corresponding integer mathematical formulation is as follows:

$$z = maximize(min_{i=1...m}(\sum_{j=1}^{n} x_{ij} \times cpu_j)) \tag{9}$$

where m is the number of clusters determined using the the 2-Dimensional Bin-Packing model. This optimization function holds subject to the same previous Eqs. 2, 3 and 4 where x_{ij} is the decision variable that takes 1 if the clients j is assigned to the cluster i.

Using this MMP model, the minimum number of cores assigned to each cluster is maximized. The inputs to this model are: (1) the number clusters, (2) the number of clients and (3) clients parameters (cpu and RAM demands). The result of this model is the assignment of all clients to all clusters so that the cpu utilization is maximized between all clusters.

Table 5 shows obtained results using CPLEX solver when minimum number of used cores in each cluster is taken into account. Clearly, for realistic cases (K equal to 2 or 3) the minimum number of used cores in each clusters is almost the same. Therefore, this parameter is optimized and the use of corresponding resource (CPU) is welly balanced.

Table 5 The minimum number of used cores in each cluster from each data-set. Time is measured in seconds

	K=1			*K=2*			*K=3*		
Problem instances	z'	z	Time	z'	z	Time	z'	z	Time
I1	11	**11**	0.00	11	**11**	0.00	11	**11**	0.02
I2	2	**5**	0.03	6	**10**	0.03	6	**10**	0.09
I3	2	**3**	0.03	2	**8**	0.06	11	**11**	0.08
I4	2	**3**	0.12	9	**11**	0.20	9	**11**	0.13
I5	1	**11**	0.31	1	**11**	0.36	1	**11**	0.23
I6	2	8	**11**.58	5	**11**	14.77	11	**11**	11.08
I7	2	**8**	215	8	11	53	4	11	98

6 Discussions

In this paper, we presented a global approach to assign a set of clients to a set of clusters in a virtualized environment. We used two different combinatorial optimization problems, namely 2-Dimensional Bin-Packing and Max-Min problems, to deal with this assignment problem. The IBM CPLEX exact solver has been used to solve considered problem instances.

The first result is that all problem instances have been solved to optimally. This means that the smallest number of needed clusters has been found and no more clusters that could lead to an extra cost and extra energy are needed. The second result is that the resolution time is very reasonable since it is not longer than 500 seconds. This means that real-life problem instances are not very hard to solve and as a result exact solvers are highly recommended.

7 Conclusion

This paper presents a global approach to deal with the problem of allocating a set of clients to a common pool of multiple clusters based on number of connections to advance resources management in virtual environment. To optimize resources allocation in Applications Services Provider's data-centers, we proposed a combinatorial optimization look to the problem. We have used the IBM CPLEX solver to solve to optimally this problem. The proposed approach will help system and network administrators in allocating a set of demands to the appropriate set of clusters.

Future research directions could include the investigation of the energy issue since it is one of the most important challenges in recent decades.

References

1. Barham, P., et al.: Xen and the art of virtualization. In: Proceedings of the 9th ACM symposium on Operating systems principles (SOSP '03), Bolton Landing, NY, USA, pp. 164–177 (2003)
2. Adams, K., Agesen, O.: A Comparison of software and hardware techniques for x86 virtualization. In: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems (ASPLOS XII), San Jose, California, pp. 2–13 (2006)
3. Xen Project. http://www.xenproject.org/
4. Citrix System. http://www.xen.org/about/
5. VMWare Inc: VMware infrastructure architecture overview. http://www.vmware.com/pdf/vi_architecture_wp.pdf
6. Kernel-based Virtual Machine. http://www.linux-kvm.org
7. Wang, D., Xie, W.: Performability analysis of clustered systems with rejuvenation under varying workload. Perform. Eval. **64**(3), 247–265 (2007)
8. VMware Inc: VMware high availability: concepts, implementation, and best practices. http://www.vmware.com/files/pdf/VMwareHA_twp.pdf

9. Fox, A., Gribble, S.D., Chawathe, Y., Brewer, E.A., Gauthier, P.: Cluster-based scalable network services. In: Proceedings of the 16th ACM Symposium on Operating Systems Principles, pp. 78–91 (1997)
10. BEA White Paper: Achieving scalability and high availability for e-business, clustering in bea weblogic server. http://www.bea.com/content/newsevents/whitepapers/BEAWLServerClusteringwp.pdf (2003)
11. Garey, M.R., Johnson, D.S.: Computers and intractability. In: A Guide to the Theory of NP-completeness. Freeman, NewYork, USA (1979)
12. Wu, L., Garg S.K., Buyya, R.: SLA-Based resource allocation for software as a service provider (SaaS) in cloud computing environments. In: Proceedings of 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) (2011)
13. Nan, X., He, Y., Guan, L.: Optimal resource allocation for multimedia application providers in multi-site cloud. In: Proceedings of 2013 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, pp. 449–452 (2013)
14. Ferretti, S., Ghini, V., Panzieri, F., Pellegrini, M., Turrini, E.: QoS aware clouds. In: Proceedings of IEEE 3rd international conference on cloud computing, pp 321–328 (2010)
15. Nathuji, R., Kansal, A., Ghaffarkhah, A.: Q-clouds: managing performance interference effects for QoS-aware clouds. In: Proceedings of EuroSys'10 of the 5th European conference on Computer systems, pp. 237–250 (2010)
16. Lai, G., Song, H., Lin, X.: A service based light weight desktop virtualization system. In: Proceedings of the International Conference on Service Sciences (ICSS'2010), pp. 277–282 (2010)
17. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener. Comput. Syst. **28**(5), 755–768 (2012)
18. Berral, J.L., Gavalda, R., Torres, J.: Adaptive scheduling on power-aware managed data-centers using machine learning. In: Proceedings of 12th IEEE/ACM International Conference on Grid Computing, pp. 66–73 (2011)
19. Martello, S., Toth, P.: Knapsack problems : algorithms and computer implementations. In: Wiley Series in Discrete Mathematics and Optimization, Chapter 8 (1990)
20. Lodi, Andrea, Martello, Silvano, Vigo, Daniele: Recent advances on two-dimensional bin packing problems. Discrete Appl. Math. **123**(1–3), 379–396 (2002)
21. Baruah, S., Fisher, N.: The partitioned multiprocessor scheduling of sporadic task systems. In: RTSS'05 Proceedings of the 26th IEEE International Real-Time Systems Symposium, pp. 321–329 (2005)
22. Cook, J.S., Han, B.T.: Optimal robot selection and workstation assignment for a CIM system. IEEE Trans. Robot. Autom. **10**(2), 210–219 (1994)
23. Han, B.T., Diehr, G.: An algorithm for device selection and file assignment. Eur. J. Oper. Res. **61**, 326–344 (1992)
24. Boyer, V., El Baz, D., Elkihel, M.: A dynamic programming method with lists for the knapsack sharing problem. Comput. Ind. Eng. **61**, 274–278 (2010)
25. Hifi, M., MHalla, H., Sadfi, S.: An exact algorithm for the knapsack sharing problem. Comput. Oper. Res. **32**, 1311–1324 (2005)
26. Yamada, T., Futakawa, M., Kataoka, S.: Some exact algorithms for the knapsack sharing problem. Eur. J. Oper. Res. **106**, 177–183 (1998)
27. Brown, J.R.: Solving knapsack sharing with general tradeoff functions. Math. Program. **5**, 55–73 (1991)
28. Kuno, T., Konno, H., Zemel, E.: A linear-time algorithm for solving continuous maximum knapsack problems. Oper. Res. Lett. **10**, 23–26 (1991)
29. Luss, H.: Minmax resource allocation problems: optimization and parametric analysis. Eur. J. Oper. Res. **60**, 76–86 (1992)
30. Pang, J.S., Yu, C.S.: A min-max resource allocation problem with substitutions. Eur. J. Oper. Res. **41**, 218–223 (1989)

31. Tang, C.S.: A max-min allocation problem: its solutions and applications. Oper. Res. **36**, 359–367 (1988)
32. IBM CPLEX Optimization studio. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
33. Korf, R.E.: Artificial intelligence search algorithms. In: Atallah, M.J. (ed.) Algorithms and Theory of Computation, Handbook. CRC Press, Boca Raton (1998) (ISBN:0849326494)
34. Fukunaga, Alex S., Korf, Richard E.: Bin completion algorithms for multicontainer packing, knapsack, and covering problems. J. Artif. Intell. Res. (JAIR) **28**, 393–429 (2007)
35. Chandra, A., Gong, W., Shenoy, P.: Dynamic resource allocation for shared data centers using online measurements. In: International conference on Measurement and modeling of computer systems (SIGMETRICS '03), pp. 300–301 (2003)
36. Waldspurger, C.A.: Memory resource management in VMware ESX server. In: Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI'02), pp. 181–194 (2002)
37. Khanna, G., Beaty, K., Kar, G., Kochut, A.: Application performance management in virtualized server environments. In: Proceedings of 10th IEEE Network Operations and Management Symposium (NOMS), pp. 373–381 (2006)
38. http://www.vmware.com/
39. Chen, Q., Xin, R.: Optimizing enterprise IT infrastructure through virtual server consolidation. In: Proceedings of the 2005 Informing Science and IT Education Joint Conference, Flagstaff, Arizona, USA, June 2005
40. Kshetri, N.: Cloud computing in developing economies. IEEE Comput **43**(10), 47–55 (2010)

Part II
Machine Learning

On the Application of Artificial Intelligence Techniques to Create Network Intelligence

Artur Arsenio

Abstract Information and Communication Technologies (ICT) growth poses interesting challenges to industry, concerning issues such as scalability, network security, energy management, network monitoring, among others. Several Artificial Intelligence tools can be applied to address many of current ICT challenges. This chapter describes the practical application of Artificial Intelligence (AI) techniques in two main classes of problems: AI for the Internet of Things and Cloud, and usage of AI techniques to manage faults and security issues in traditional Telecommunication networks. Therefore, we will present our research work for the application of AI into different domains, describing for each the current state-of-the-art, and the implemented solution together with the main experimental results. This chapter will demonstrate various benefits achieved from adding an intelligent layer to ICT solutions, in various domains. Finally, we will also address future developments.

Keywords Artificial intelligence · Internet of things · Telecommunication networks · Machine learning · Cloud computing

1 Introduction

According to a joint study by *Groupe Speciale Mobile Association* (GSMA) and Machine Research, by 2020 there will be 24 billion connected devices in the world and over half of them will be non-mobile devices such as household appliances on smart buildings. Such explosive growth is placing stringent requirements on future telecommunication networks, concerning bandwidth, security, and energy consumption. On the other hand, other fields such as the health sector will increasingly depend on ICT. Several Artificial Intelligence (AI) tools can be applied to address many of these current ICT challenges (see [1] for an extensive review).

A. Arsenio (✉)
YDreams Robotics, Universidade da Beira Interior, Covilhã, Portugal
e-mail: arsenio@alum.mit.edu

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_4

This chapter addresses artificial intelligence techniques employed to create network intelligence, by presenting network solutions to exploit the benefits brought by AI. We will discuss different platforms and applications, employing RFID technology, healthcare sensors (for addressing problems such as epidemics prediction) and robots, discussing the main challenges brought by these systems. More specifically, five solutions are described, for different ICT domains of application, employing AI algorithms. These solutions are divided into two groups, namely AI applied to problems in the Internet of Things (IoT) field, as well as solutions targeting the usage of AI on telecommunication networks. More domains of application for AI will be discussed on the conclusions. The reported research work, by the author's research team, is a result of collaborative work under the scope of different research projects.

1.1 AI for Internet of Things

Artificial Intelligence techniques are increasingly being employed for the ever-growing field of the Internet of Things, in which everyday objects become connected. These techniques may consist of heuristic search algorithms, machine learning techniques, deterministic or probabilistic state machines, knowledge-based reasoning systems, natural language processing, graph theory, or environment perception, among others.

We discuss on this chapter three solutions that exploit the emergence of Intelligent Things on the Internet with more power than mere sensors, such as AI processing and actuation capabilities. This enables a new Internet of Things: of intelligent agents that are able to take clever decisions based on the perceived information from the surrounding environment, in such diverse fields such as eHealth (Sect. 2), smart buildings (Sect. 3), or robotics (Sect. 4). Currently, smart devices, such as home appliances and light bulbs, allow for some degree of actuation. In the short term, more intelligent things, like appliance robots (such as smart lamps, or the currently available smart vacuum cleaner) will also be communicating with each other, and with humans, through the Internet.

1.2 AI for Telecommunication Networks

Telecommunication networks are nowadays very complex systems. A single network router may have millions of software lines of code running on it. On the other hand, the large number of network components need to be monitored and remotely configured by network management solutions, that manage faults and security issues. However, fault prediction is becoming more complex. Additionally, there are currently so many alarms being generated on a traditional telecommunications network that the main problem to deal with is not lack of information, but instead an excess of alarms (often correlated) from a multitude of elements. AI techniques are essential for reducing the

number of such alarms by removing correlated information, as well as for predicting the occurrence of future problems based on historic information, as demonstrated on Sect. 5.

Another issue of increasing importance on telecommunication networks is network security. On one hand, it is of foremost importance to detect network attacks, and to identify the hackers behind them. On the other hand, even for regular clients there are security concerns with respect to the resources that they consume. Indeed, it is often debated whether Peer-to-Peer (P2P) traffic should be, or not, allowed by operators, since it typically consumes a large share of the available bandwidth. AI techniques can be employed to learn client profiles that represent they typical pattern of consuming network services, and charge them or implement appropriate security countermeasures according to such profiles, as shown in Sect. 6.

This chapter argues that the application of AI techniques enable intelligent telecommunication networks, providing solutions to some of the most important challenges they face currently.

2 Graph Theory for Virus Epidemic Prediction

Infectious diseases, as demonstrated by the recent surge of Ebola, can pose dangerous threads to large communities. The high level of people mobility creates new challenges for fighting these diseases, requiring new methods for detecting, on a social environment at a global scale, how people get in contact with each other. Furthermore, epidemic models representing such global relationships are also necessary, as well as bio-health sensors that may detect on real-time, at a very initial stage, the appearance of a disease on a person. Sensors on personal devices that gather information from people, and social networks analysis, allow the integration of community data, while data analysis and modeling, through the application of AI techniques, may potentially indicate community-level susceptibility to an epidemic.

2.1 State of the Art

Epidemics are a major public health concern and it has been shown its impact can be reduced by early detection of the disease activity. For instance, it has been shown that the level of influenza-like illness in regions of the US can be estimated with a reporting lag of one day, when compared to clinical methods whose results take a week to be published [2]. However, the field of epidemic prediction has an inherent lack of adequate data [3]. Indeed, most of current models are derived from simulations [3]. In addition, there are similarities between the spreading mechanisms of biological infectious agents and computer virus, and hence some authors have exploited these similarities to address problems in both fields [4].

Developments in epidemic spreading have emphasized the importance of network topology in epidemic modeling. Social and biological systems can be described by complex networks whose nodes represent the system actors, and its links the relationships between them [4]. They may be modeled as a computational network, which is itself modeled as a graph. A graph consists of a set of points called nodes or vertices. Interconnections between these nodes are named links or edges and, in this application, they represent a form of contact or relation. The degree of a node k corresponds to the number of its neighbors [5].

Recently, there has been a surge of interest on systems' approaches applied to epidemiological research, especially well suited for social epidemiology [6]. Such approaches rely on an implicit assumption that the dynamics of a system is different, qualitatively, from those of the sum of its parts. As such, the relation between system components is more relevant than the attributes of its components by themselves. In the case of network approaches, this means a bigger emphasis on the structural characteristics rather than on nodes characteristics. Therefore, the social ties that influence network actors have important consequences in the analysis [6].

Disease, information and social support are among health-relevant factors of interest that may impact networks nodes [6]. Upon the appearance of the Acquired Immunodeficiency Syndrome (AIDS), social network analysis was demonstrated to be suited for infectious contact tracing [7]. Social networks for the investigation of infectious diseases are typically built with resort to personal contacts only, as these contacts are the most traceable means for disease transmission. Nonetheless, in the case of diseases that transmit through other mechanisms besides personal contacts, the inclusion of geographical contacts into social network analysis methods has been proven to reveal hidden contacts [7].

2.2 Architecture and Implementation

The learning solution for the aforementioned epidemics estimation problem is composed by the following architectural modules, as depicted in Fig. 1.

Both the data sampling and data filtering modules constitute the Sensing layer of the system. The aim of this layer is to obtain processed input data. The former module gathers Network Data (i.e. relational data constituting an individual's social network) and Contact Data (i.e. inferred meetings between individuals as detected from personal sensors). The later guarantees sampling complies with user privacy requirements, by performing data anonymization and filtering. Information dissemination comprises the Sharing layer, where system output is returned to users.

The learning layer, constituted by both the data analysis and epidemic prediction modules, composes the solution's core. In this layer, acquired data is transformed and integrated into a model, contributing to the extraction of intelligence in the context of the application problem. These two modules are hereafter described in more detail.

The purpose of the Data Analysis module is to correlate data that may bear distinct viewpoints and resolutions, processing and merging data and enabling the extrac-

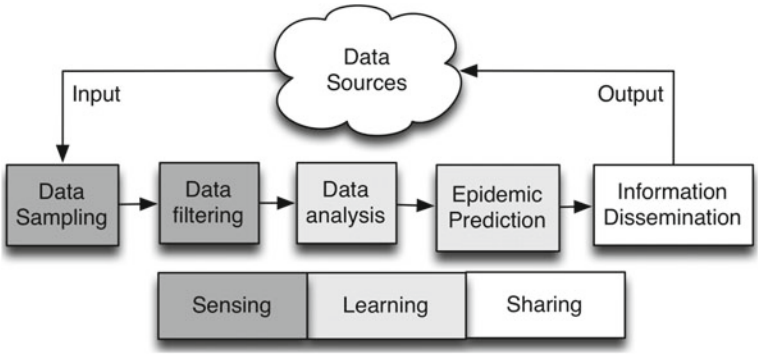

Fig. 1 Solution architecture based on learning through social networks. The output is sent to users. Input data from users' biosensors are feedback into the system

tion of higher-meaning information. Previously acquired contact and network data is merged, concluding for all users with whom they have met and which social connections exist for them. The merging process attributes different importance to the relations associated to user ties observed in the network data by measuring them against perceived user meetings. Given that infectious disease spread is primarily achieved by direct contact, contact-related data has a higher impact in this estimation. This process consists in the weighting of network links with the information provided by contact data [8].

The merged data originates a social contact network, containing all the considered individuals. This network accounts for the relationships between all actors with their associated degree of importance in epidemic spread. As data analysis is a computationally costly process, this module should perform on the set of all aggregated data. For each analysis round the social contact network comprising all users is evaluated. This procedure is represented in Fig. 2.

Social contact networks may exhibit a wide variety of properties. A network's node degree distribution and isoperimetric constant (or epidemic threshold) are among the most relevant in this area of application. Since node degrees are arranged in an exponentially distributed way, it is a scale-free network.

The resulting social contact network is joined with infectious agent data, constituting an epidemic model. This model is fed into the Epidemic Prediction module (the intelligence on the learning layer). By employing the appropriate mathematical formulation and metrics, the later evaluates a social contact network for a given epidemic agent's properties and, as a result, assess the dawning probability of an epidemic. The idea behind this module is the correlation of network properties and analytical methods applied to epidemiology. More precisely, it is the relation between the isoperimetric constant and the epidemic threshold [8]. Implementation-wise, in conformance with methodology criteria and for scalability reasons, it was desired that this module would use memory parsimoniously and run quickly, while for extensibility reasons, its code should be relatively easy to read and modify. Additionally, it was necessary to find a process through which obtain viable data analysis.

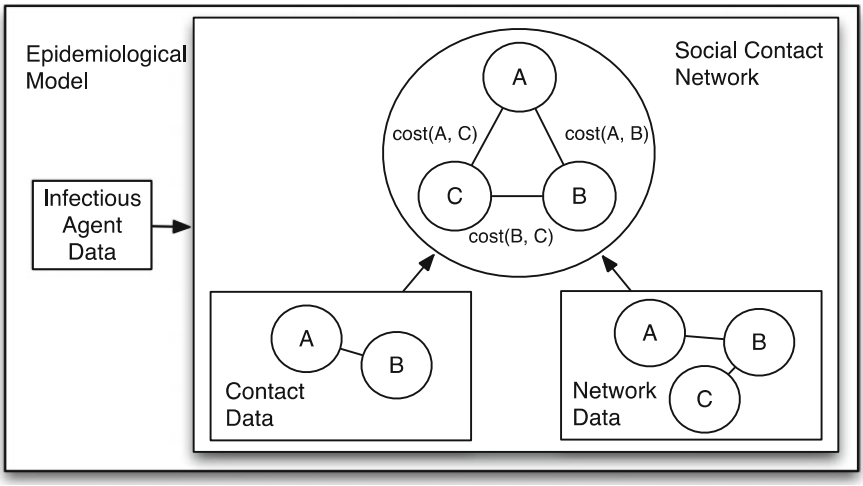

Fig. 2 Solution epidemic model

An important result in network models is the prediction of a non-zero epidemic threshold (λ_c). The higher the node's connectivity, the smaller the epidemic threshold, and consequently, the higher the probability of infection [4]. Equation (1) represents the bounding of the epidemic threshold and defines a condition, that when not met implies the existence of an epidemic:

$$\frac{\beta_a}{\delta} \leq \frac{1}{\lambda_1, A} \tag{1}$$

where $\lambda_{1,A}$ corresponds to the modulus of the largest eigenvalue of the adjacency matrix of the associated contact network topology, also known as the dominant eigenvalue or spectral radius of the network graph, and β_a stands for the average rate of infection along a network edge and δ is the recovery rate of an infected node [5].

The strength of an epidemic can be evaluated with resort to the generalized isoperimetric constant ${\lambda_{1,A}}^{-1}$ of the associated social contact network, also known as Cheeger's constant [5]. By determining and comparing these metrics, it is assessed the network susceptibility to disease outbreaks.

In response to the validation demands of epidemic simulation, an analytic network topology approach was selected. The main idea behind the predictive power of this module is the notion that the topological properties of a social contact network can be used to assess epidemic persistence [5] and, thus, its advent. While this approach provides a limited context, it provides accurate results for the data family in analysis. The prediction algorithm consists in identifying the spectral radius (the maximum of the absolute of the eigenvalues of the adjacency matrix) of the social contact network. The obtained spectral radius and the infectious agent parameters are inserted into (1), which relates the epidemic threshold of the graph associated to the network and the rate of infection transmission and recovery. This process verifies if there is a subset

of users that possess enough connections, of a relatively high weight, to constitute a weight cluster, and thus form a network bridge. If this subset is representative enough, the existence of network weight bottlenecks will be high enough for the successful spread of disease and the creation of an epidemic. The equations yield a Boolean value (forwarded to the next layer, the Sharing layer), which can be translated into network epidemic vulnerability, if false, or the lack of risk if true.

Simulation-oriented strategies demand an extremely complex validation process. Furthermore they are potentially impossible to validate completely [3]. These approaches attempt to validate the model by measuring the output of computationally expensive simulations against the public health statistics of an ongoing epidemic [9]. Alternatively, one may employ data from past epidemics, which compromises the results generalization. There is however a significant lack of data in the joint area of social network analysis and epidemiology [4]. Experimental evaluation of the solution [8] through simulation evidenced that an increase in contacts results in a decrease in epidemic threshold from the initial point of zero contacts in the network. Hence, the impact of contact data in the network is a dominant factor in its susceptibility to an infection [5], in the limit, eclipsing the shape of network data. Thus, with a weighting scheme, it is possible to impact the model without being constrained by the topology of the sampled network. It was also experimentally verified that the introduction of random contacts has an impact over network data, resulting in a lower epidemic threshold that decreases with network size.

3 Machine Learning for Smart Building Energy Management

Heating, Ventilation and Air Conditioning (HVAC) systems account for up to 60 % of the total energy consumption of commercial buildings [10]. In modern buildings, HVAC system settings are configured centrally on the management console of the Building Automation System (BAs). Currently, settings are not adjusted in real-time to take into account fast dynamics of occupant preferences leading to overcooling or overheating. Enabling user participation on the HVAC system, giving them the opportunity to vote on their level of comfort, and using such feedback to control the system, can reduce the number of situations of excessive service delivery that increase energy expenditure. However, it is necessary to find a good trade-of between all users' level of comfort (which may vary significantly) and energy consumption. Learning techniques can be effectively employed to address such problem.

3.1 State of the Art

Most previous research approaches focus on intelligently managing the HVAC system energy, since this is the major source of energy consumption in a building. Different AI technologies have been employed to add this intelligence [11]. Multi-agent

learning systems (e.g. MASBO—Multi-Agent System for Building Control [12] and MACES—Multi-Agent Comfort and Energy System [13]) consist of a collection of communicating software agents that can monitor and control building systems such as lighting and HVAC to reduce energy consumption in a building. Adaptive control systems adapt a controlled system in real time, usually by means of parameter estimation. For instance, the Adaptive Control of Home Environments (ACHE) project employs neural networks to learn how to control the lighting [14]. Another AI system [15] also uses a neural network for achieving minimum energy consumption whilst maximizing user comfort.

Other approaches, such as iDorm [16] employ a Fuzzy control system, an AI technique applied whenever the sources of information are interpreted qualitatively, inexactly or uncertainly. Dodier et al. proposed an Occupancy Sensor Belief Network approach [17], based on probabilistic models of occupancy, representing the probabilistic relationships between occupancy detection through various simple sensors. Pattern mining techniques are commonly used to find existing patterns in data, and have been employed on the Ecosense [18] and iSense projects [19]. Another system combines data from a sensor network with acoustics, lighting, temperature, motion and CO sensors, resorting to an event based pattern algorithm for prediction of user behavior patterns [20], which then controls the HVAC (the authors report energy savings up to 30 %). Erickson [21] employed Markov Chains for occupancy data, retrieved from the wireless camera sensor network, and reported significant results up to 42 % of annual energy savings using a test bed system.

The SpotLight project [22] gives people the possibility to monitor energy consumption by using proximity sensors. People carry an active RFID tag, which is detected by antennas, for predicting the occupants' location. SpotLight reportedly achieved potential energy savings from 10 to 15 %. Several other sensor-based energy saving approaches have been proposed, such as Feedback-based Systems (e.g. [23]), Smart Thermostat Solutions (e.g. [24, 25]), Wireless Sensor Occupancy Prediction [26], or the Occupant Information System [27]. Different occupancy detection strategies are reported, either based on infrared sensors [17, 28], or RFID systems in doors, to count the number of persons entering and exiting a space. Other approaches count the exact number of persons in a space using ultrasonic tracking systems [29] or video cameras; or using both techniques to maximize system accuracy.

3.2 Architecture and Implementation

Aiming at improving smart buildings energy consumption using AI, Mansur et al. proposed a solution [30] that detects room occupants through RFID card-reading, interacts with occupants on their mobile devices' interface, learns appropriate temperature set-points from users' votes, and sends commands to the HVAC sub-system through a gateway (according to Fig. 3). A prototype system was tested on a university building.

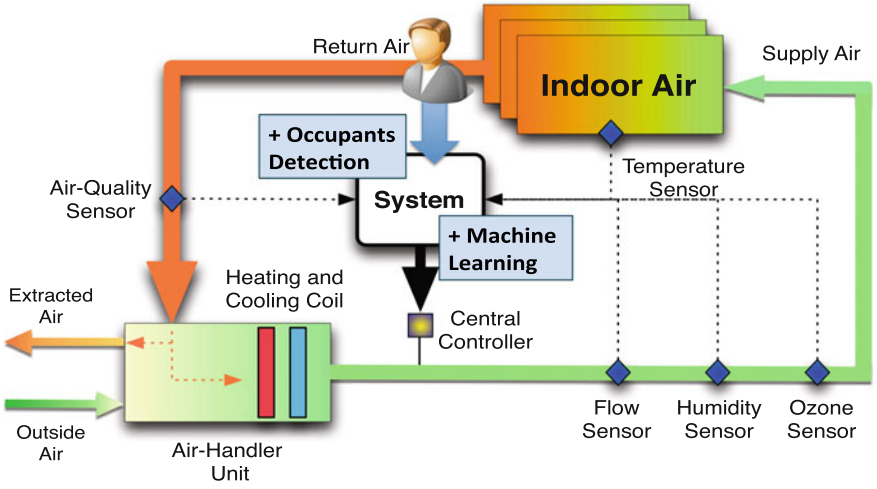

Fig. 3 The overall smart building setup, with the solution incorporating ocuppants detection and machine learning for estimating their preferences

A RFID Module retrieves the users information from the RFID tag embedded on a student card. RFID card readers were mounted on two doors' entrances and wired to two Arduino systems (Arduino Board plus RFID and LCD shields), coupled to a Raspberry Pi, to aggregate the information of the Arduino subsystems and send it to a Storage Module. The AI module is the system brain, performing several functions:

- Occupancy detection: Knowing and deciding if a user is on the room or not, based on information received from the RFID System.
- User voting and feedback handling: Collecting user's votes and sending to users feedback regarding ambient variables.
- Set-point calculation: Collecting the occupant's votes, validating them and calculating a new set-point, which minimizes discomfort.
- Learning algorithm: Learn occupants' behavior and predict the best decision, by employing a k-means clustering algorithm [31]. The input state is the time of the day, number of room occupants and room temperature (other measures can be included as well, such as outside temperature or month of the year), and the output is the users' votes (increase, decrease or maintain temperature).
- Interaction with HVAC System: Sending information about the new set-point and receiving information concerning the ambient variables.

The evaluation was based on simulations [30] of a real life environment using EnergyPlus package. The simulation environment consists of a room where occupants are able to cast a vote with the intent of setting the temperature, which best fits their comfort. Experimental results suggest 23 % energy' savings while maintaining acceptable levels of occupant's comfort.

4 Intelligent Middleware for Cloud Robotics

Cloud Robotics is another potential area for benefiting from the introduction of AI technologies, namely for optimizing resources' usage on mobile or embedded devices.

4.1 State of the Art

Robot operating System [32] is an open source middleware for developing large-scale service robots, supporting multiple programming languages (C++, Python, Octave and LISP). The fundamental concepts of ROS implementation are: node, messages, topics, and services. Nodes are processes that perform computation. ROS is typically comprised of many nodes. The nodes communicate with each other by passing messages. A message is a typed data structure and can be composed of other messages and array of other messages. A node sends a message by publishing it to a given topic (publish-subscribe model). A node that is interested in a certain kind of data will subscribe. In general publishers and subscribers are not aware of each other. Publish-subscribe mode is a flexible communication paradigm but broadcast routing scheme is not appropriate for synchronous transactions.

YARP (Yet Another Robot Platform) [33] is an open-source robot middleware, allowing the movement of a computational process between machines that are in a cluster to redistribute the computational load and to recover from hardware failure. Communication in YARP uses the Observer pattern. In YARP a port is an active object managing multiple connections. The ports can be connected programmatically or at runtime. The communication is asynchronous and as such messages are not guaranteed to be delivered unless occurs a special provision.

PEIS (Physically Embedded Intelligent Systems Kernel) [34] is a middleware that provides a common communication and cooperation model shared among multiple robotic devices. It employs the Ecology concept of Physically Embedded Intelligent Systems. The PEIS Kernel provides a shared memory model, a simple dynamic model for self-configuration and introspection, and supports heterogeneous devices.

Player/Stage system is a middleware platform for mobile robotics applications [35, 36]. The component player is a device repository server with robots sensors and actuators. Each of these devices has an interface and a driver. The interface is used by middleware clients to obtain information (as collected by the sensor) and to control the actuators. The other component (stage) is a graphical simulator that models devices in a user defined environment.

RoboEarth Cloud Engine (Rapyuta) is a cloud robotic platform for robots that implements a platform as a Service (PaaS) open-source framework [37]. It provides a secured customizable computing environment (like a clone) in the cloud so that robots can offload to the cloud some heavy computations. The robots connect to the Rapyuta and can start the computing environment by their own initiative, launching any computational node uploaded by the developer, and communicate with the

launched nodes using the WebSockets protocol. Robots are allowed to process their data inside the computational environment in the cloud without downloading and local processing. All processes within a single environment communicate between them using ROS interprocess communication.

4.2 Architecture and Implementation

Rapyuta employs offload techniques for enabling robots to perform heavy computations on the cloud. But it does not address the optimal usage of device and cloud resources for reducing devices' battery consumption or CPU usage. In our solution [38] whenever the device does not have the capability for running some data, it employs an optimization strategy to send the data, and transfer the flow of execution, to the cloud in real-time.

The system consists of devices running applications (cell phones, tablets, robots, and computers, as shown in Fig. 4) and the cloud that makes data processing and saves data. The communication protocols are TCP, UDP, SSH and HTTP Rest, and a publish/subscribe model for internal communications in the device. Devices run applications developed by programmers, having constraints such as limited memory and battery (contrary to the cloud). These applications will run the management and cloud client side modules for programmers to use our middleware. The former monitors hardware components, and communicates to the cloud client whenever a component reaches a critical condition. The cloud client interchanges application's control messages and data to the cloud server module.

The management module aims to determine hardware components state (battery, CPU and memory), as well as the Wi-Fi connection state. The programmer defines each component's critical state on a configuration file, before the middleware starts to be used. Whenever one of these components achieves a value above a critical value (and WiFi signal is strong enough) a certain execution will no longer run on the device, being transferred to the cloud (system state machines are shown in Fig. 5).

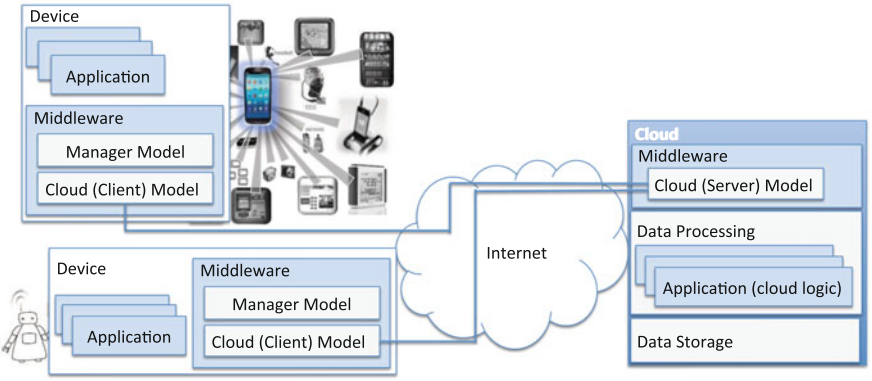

Fig. 4 System architecture and its components

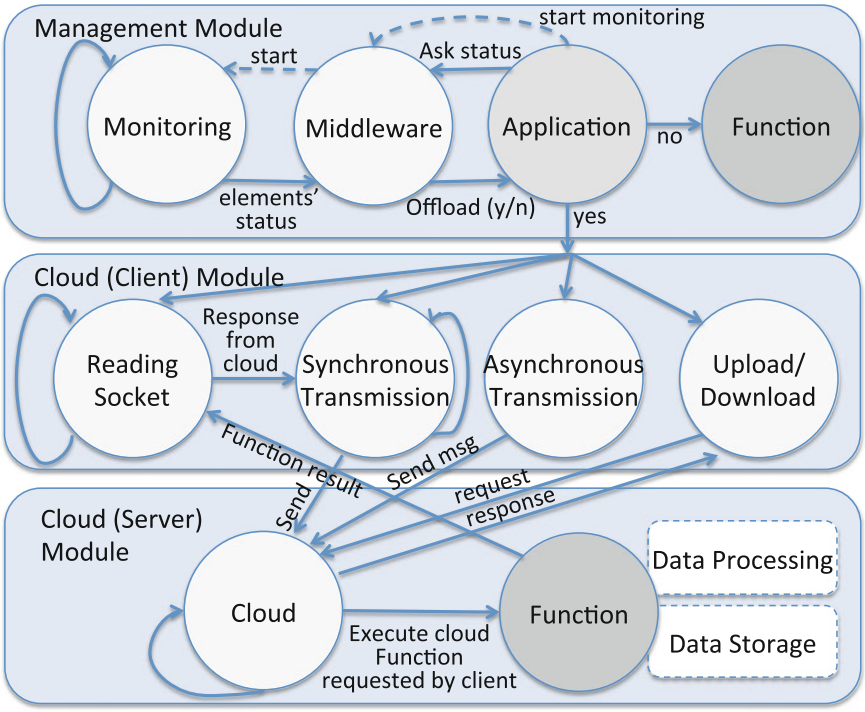

Fig. 5 Management and cloud finite state machines

The "Middleware" state checks continuously the conditions of the Wi-Fi signal quality and the conditions of the battery, CPU and memory, through calls to device hardware that runs the middleware. The values obtained are compared with the critical values stipulated by the application programmer. The load CPU analysis is a bit different from the other checks, because a notification is only sent if the measured values are above the critical value for three times in a row (to avoid reactions to sporadic peaks). If the signal quality of the wireless network is below the critical value stipulated by the programmer the remaining monitoring tests will not be performed. After each monitoring cycle of the hardware components, the "Monitoring" state goes into sleep mode for one minute.

The "Application" state sends requests to the "Middleware" state on the conditions of a component (e.g. battery). If the reply is "False" (transition between the "Application" and "Function" states), it means no action is needed, since the state of the component is below the critical status (and hence running with enough resources on the device). Consequently, the programmer's application can continue to run without any changes, and no event is initiated. In case a component exceeds the critical value, the offload to the cloud is activated. In this case the programmer chooses the actions to take after receiving the message. One possible option is to use cloud platform

provided services for performing certain actions. This way it is removed some load on the device that is running the application, releasing resources (e.g. memory).

The middleware in the cloud replies to an application request for storage (transitions between "Application", "Upload/Download" and "Cloud") either: (i) with a confirmation that the information was successfully saved; or (ii) there was an error while performing the storage operation; or (iii) the information as requested by the get command; or (iv) error due to failure on obtaining the requested information. The communication model accepts either blocking and non-blocking message transmission. Messages contain a function ID and its arguments. The "Function" state at the cloud consists of the execution of the functions that were chosen by the programmer to run in the cloud. At the end of the execution of a function a message is sent to the client (state "Reading Socket) with the function ID and its result. It is also sent a small packet to identify the type of transmission (synchronous or asynchronous).

Experimental tests [38] employing data from sensor devices, including heavy streaming of video data, demonstrated that the use of the cloud to solve the lack of resources problem in devices is quite advantageous because it allows the CPU load to be reduced to lower values. This leads to battery with extended autonomy, thereby providing less inconvenience to device users. It also avoids applications entering in a blocking state due to lack of memory, and allows running more applications in simple device that otherwise would exceed the available resources. But most importantly, the decision whether to run an application locally or remotely is done dynamically, according to the status of the available resources as checked through active monitoring.

5 Multiple Neural Networks for Client Profiling on Telecommunication Networks

Network traffic monitoring is an essential activity for active and passive management of networks of various dimensions, and may be performed through the observation of packets or flows. Great efforts have been made in the scientific community with the ambitious aim of understanding how the characteristics of traffic and diverse applications affect the behavior of the network infrastructure. Thus, measurement strategies, together with AI techniques, can bring important contributions to identify abnormal behavior (e.g. elevated or sudden increase in network traffic).

5.1 State of the Art

Even though the monitoring activity has become mainstream with the help of tools existent in routers (e.g. CISCO Netflow), there are still some problems that must be addressed. Currently, the main obstacle of the traffic monitoring based on measurement (of packets or flows) is the lack of scalability on the capacity of the links. That

is, the traffic monitoring of links with high capacities generate an enormous volume of data. With the increase of the capacity of the links and the number of flows, keeping counters for each flow that crosses the routers becomes expensive and difficult to execute.

Thus, diverse sampling strategies have been proposed as ways to optimize the process of selection of packets (for accounting of flows) or selection of flows (for statistical analysis of original traffic). Some papers [39–41] focus on the importance of choosing a good sampling method to enable the estimation of the original traffic from the sample traffic. Even though sampling is necessary, because it is completely unaffordable to evaluate all the packets that travel through the network, the knowledge of flow statistics in the sampled flow is still useful to understand the properties and sources of the traffic and to know the consumption of resources in the network.

Deep Packet Inspection (DPI) involves the complete analysis of the packets that cross the network, examining not only the header, as it is the case for the Shallow Packet Inspection (SPI), but also its body. However, Internet packets do not consist only of payload data added in one header. In each layer of the multi-layer architecture there exists a header and a body, and the payload of one layer contains the header of the superior layer. Thus, a more adequate definition is based on the border between the IP header and the IP payload. Therefore, Deep Packet Inspection is the action of any network device that is not an extremity of a communication, to use any field in a layer superior to the IP addresses. It contrasts with the SPI, which only verifies a portion of the header of one packet [42]. Modern network devices use deep packet inspection for the execution of sophisticated services, such as detection and prevention of intrusion, traffic shaping, load balancing, firewalls, spam detection, and antivirus, among others. Deep packet inspection constitutes a powerful mechanism to perform the correspondence of the criteria over the packets [42].

Previous work employed approaches based on estimation of degree of auto-similarity (equivalent to identify fractal patterns on traffic) to identify network attacks [43] on the dark (without packet inspection). Learning approaches [44] have also been proposed for the optimization of traffic filtering rules, which produced significant improvements on attack detection accuracy.

Due to the high volume of traffic that runs daily through the Internet, especially P2P traffic, many companies, institutions and ISPs were forced to apply restrictions, mainly to the P2P traffic, not only for legal and political reasons, but also to ensure a good performance of the network for its users. The methods and instruments available to ensure this type of work have known a great evolution to be able to keep up with the new and most recent applications, cipher algorithms and encryption processes. From the simplest firewall rules to state of the art software, a long path has been travelled. Just like in a war, where the appearance of a new weapon implies a matching counter measure, a successful method to detect P2P traffic forces developers to find new or better alternatives to keep it stealth.

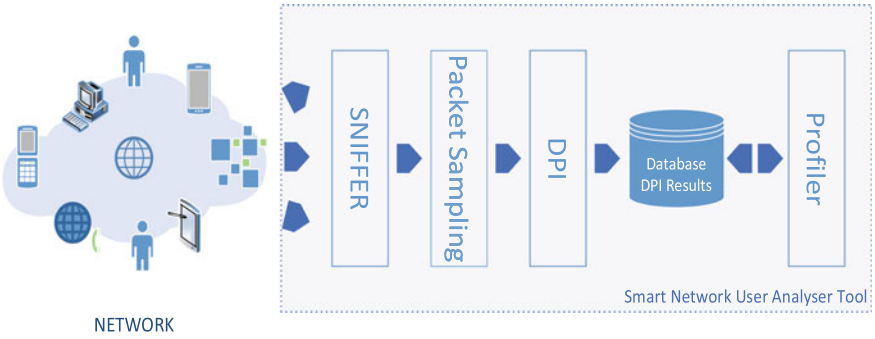

Fig. 6 SNUAT architecture

5.2 Architecture and Implementation

Instead of trying to block P2P traffic, our solution aims at profiling the client usage of P2P traffic and other network data (such as multimedia, game, etc.) in order to charge clients fairly according to the network resources they consume (i.e., their profile). A new architecture was conceived, employed back-propagation neural networks for users' profile learning. Figure 6 presents the architecture solution, named "Smart Network User Analyzer Tool" (SNUAT). The solution is composed by the following main modules: (i) Sniffer; (ii) Packet Sampling; (iii) DPI; (iv) Profiler and Data Base.

The SNUAT is an efficient and scalable software solution for network traffic analysis that enables the network operator to learn the profile that best matches the pattern of network usage associated to a given client. This profile is induced from real-time smart packets sampling. That sampling is performed in an intelligent way (smart sampling) using adaptive sampling. Depending on external factors: (i) CPU load; (ii) network data rate; (iii) sampling rate; the solution is able to find the ideal sampling interval, using a neural network.

A good location to install the SNUAT is in one of the network nodes, the best option being the installation in a second aggregation router (in the ISP network). The routers of second aggregation are the devices installed after the DSLAMs (Digital Subscriber Line Access Multiplexer). Usually named EDGE Routers, they have a great capacity of storage and processing. Alternatively, the installation of the SNUAT may also be done in the Set-top-box (STBs) of the ISP clients, since nowadays many ISPs offer triple Play STBs—a solution with TV, telephone, Internet, all in the same box. Currently, STBs possess great processing capacity. With this solution, better and more efficient results are obtained, and, therefore, a better classification of the client is achieved, since the traffic inspected is all relative to only one client. Since the ISPs have remote access to the STB's, in case we choose this alternative, there is no impact on the architecture and solution presented, since the operator may access the STB and observe the SNUAT output remotely.

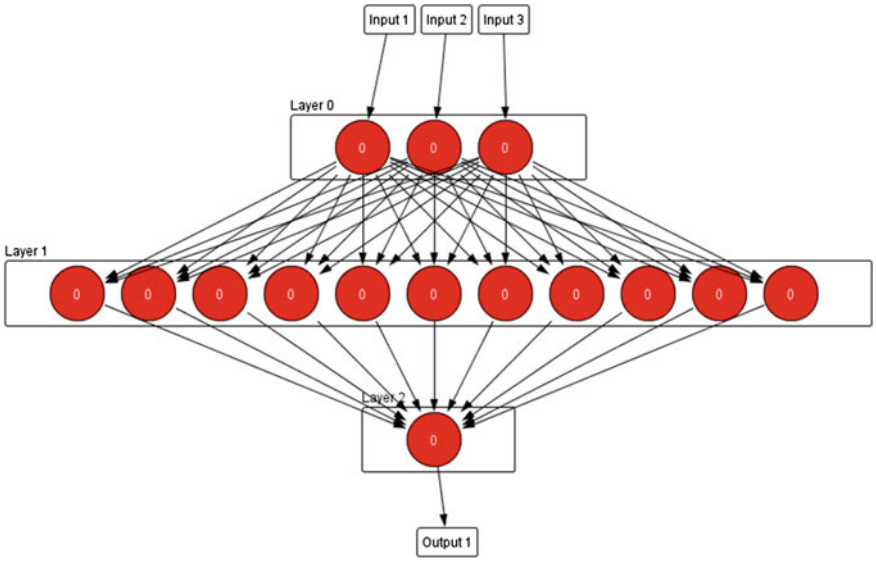

Fig. 7 Neural network architecture for smart sampling

The Sniffer Module is the entity responsible for capturing all the Ethernet traffic that crosses the network node. This sniffer acts in a promiscuous mode by capturing all the packets that cross the node.

The Packet Sampling Module samples the traffic delivered by the sniffer. It implements a method of systematic sampling based on time, i.e., all the packets that cross the node during a time interval are captured and analyzed. When that period expires, all the system stays offline exactly during the same period of time. This module is also responsible for implementing Smart Sampling.

Smart Sampling is a clever method, to find the optimum sampling rate. This method is composed by a Neural Network (see Fig. 7) with 3 inputs: (i) The Total % of CPU Load (ii) Network Rate (i.e. Data Link Rate) in Mbps (input 2). (iii) Sampling Rate (Mbps) (i.e. Total Sampled Traffic as a function of sample interval). The network has only one output: (i) a sampling interval between 10 and 35 seconds.

The careful choice of each of these variables is of vital importance. If the CPU load is elevated, it is desirable that the sampling intervals are higher (fewer samples), since it is not desirable that the normal operation of the system where the architecture is installed is affected—we want the system to maintain its performance. If there is a lower CPU load, it will generate more samples, thus smaller sampling intervals. Having a larger sampling interval will lead, on one hand, to more traffic to analyze (more processing) during the time when the capture is active. However, on the other hand, the system will be in offline mode during the same period, which clears the system during that period.

The Deep Packet Inspection Module is responsible for performing the inspection of the packets, for inferring to which technology the sampled packet belongs and for output data storage. This module receives the sampled packets, after being received by the previous module (Packet Sampling), and inspects only the ones that correspond to the client that we want to classify. That filter is based on the IP address. If a certain IP packet corresponds to the client we want to classify in its source or destination IP address, then the inspected packet is classified.

The Profiler module reads the data from the database after sampling, and for infers accordingly the clients' profile. The inducing of the profile is once again performed using a multi-layer back-propagation neural network. The network has 10 input neurons, a hidden layer of 20 neurons and 6 output neurons. The training group is composed by 168 inputs and respective outputs, being 230 iterations necessary to learn. The 10 input neurons are (given as %):

- Traffic related with chat and instant messaging applications (e.g. MSN).
- Traffic related with Cloud Computing applications (e.g. DropBox).
- Web surf traffic (e.g. Google, GoogleMail, HTTP).
- Stream traffic (e.g. Sopcast).
- % Voip traffic.
- Gaming traffic: traffic sourced by online games (e.g. WorldofWarcraft).
- Traffic for music applications (e.g. iTunes, GrooveShark, Spotify).
- Traffic P2P (e.g. BitTorrent).
- Traffic related with social networks (e.g. Facebook).
- Other protocols supported by the DPI and not contemplated in the present study.

Experimental evaluation showed a 100 % hit rate in 5 of the 6 scenarios tested [45], and 77 % on the remaining one. The average processing time for each sample was approximately 200 ms, which was considered acceptable to overcome some of the current problems faced by ISPs, institutions and telecommunication companies.

6 Alarm Prediction on Telecommunication Networks

Telecommunication networks operation and management is based on the data provided by the network elements [46]. This data consists of log entries containing alarms that represent events that occurred in the network elements. In large networks, a single problem may generate a big volume of alarm messages, which makes it difficult for the telecommunications monitoring operators to analyze and manage the network faults. However, the sequence of alarms generated is often correlated, because they lead to the same fault. These event correlations can be exploited to predict the occurrence of target alarms using machine-learning techniques, and thus support decision making on the monitoring operator side.

6.1 State of the Art

Deterministic and probabilistic machine learning systems can be employed for sequential classification to address event prediction, using the knowledge base generated by the knowledge discovery system. The machine learning solution has to satisfy the following two requirements:

1. Sequences can differ from each other in terms of size, and these sizes are unknown in advance;
2. The system receives as input eventSets (sequences of events associated with a particular target event), and uses it to update the model's parameters, meaning the machine learning model has to be more accurate after a new training phase, in terms of classification.

Deterministic machine learning systems have the objective of matching a new testing sequence of events, to the correspondent ones in the knowledge base. In our case, the Machine Learning system has to recognize sequential patterns. A possible approach to address this problem is based on Finite State Machines (FSM). The Acceptor Finite State Machine (AFSM) works as a sequence recognizer, and can be implemented with timeouts [47]. During an offline periodic training phase, the machine learning system is trained with new sets of data. This training phase consists in the creation of n AFSMs, where n is the number of sequence patterns. In order to use a machine learning model based on AFSM, one has to apply one AFSM to each sequence pattern in the knowledge base. This way, each AFSM can have its own distinct size, so requirement 1 is met. Requirement 2 is also met, because each pattern on the knowledge base is an eventSet.

Non-deterministic Machine Learning systems [48] have the particularity of being able to train themselves with the patterns from the knowledge base, forming a classification model. An Artificial Neural Network (ANN) may be seen as a set of hidden processors (units), connected by unidirectional connections, which carry numerical data. This set of processors serves to map an input to an output [49]. Neural Networks have the ability to learn from training examples in the form 'input→output', to adjust the weights of the connections.

This format of input matches the knowledge base generated from the EventSets method, the association rule (a→b). This way it is possible to feed an Artificial Neural Network with training examples, based on association rules. The result is an ANN with a set of events as input, a hidden layer of processors, and a specific target event as output.

A Markov chain is an extension of the FSM, with probabilistic transitions. The Hidden Markov Model further extends the Markov Chain model. It is used to model simple stochastic processes, where we have a number of states, each state corresponding to an observable event [50]. An HMM may be used as a sequence classifier, where the class labels, in our case, are the target events.

ANNs can classify sequences of alarms, even when the input data is noisy, and be trained with vector-coded patterns of alarms through supervised training, and

generally, ANN's have high statistical prediction accuracy [51]. However, the ANN solution fails to meet the requirement 1, because the ANN has a predefined number of input values, but the sequences used in our case have unknown sizes.

On the other hand, Hidden Markov Models are used for sequential classification. However, in order to build HMMs, one has to define the initial parameters of the model (initial distribution, state transition probabilities, emission distribution, and number of states). Additionally, HMMs prediction's probabilities tend to grow inversely with the sequence size, being still very useful for such problems as speech recognition with a finite alphabet (being the best prediction taken comparatively among all HMMs output). But this is not the case for learning patterns on telecommunication event logs.

Other approaches have been proposes, such as Timeweaver [51], which employs a Genetic Algorithm for identifying predictive patterns in sequences of events.

6.2 Architecture and Implementation

On the proposed Preventive Monitoring system architecture (researched under the scope of Nokia Siemens Networks' NaDM tool), the classifying process has N classes, each class referring to a target event. For that reason, N Evaluation Engine (EE) instances are created. An EE is a machine-learning instance created for each target event. We implemented it based on Acceptor FSMs, so an EE contains one finite state machine for each sequence pattern in the knowledge base.

The system has an offline flow, responsible for pre-processing the alarm data, and executing the pattern mining algorithms to generate the knowledge base of learned patterns. The online flow represents the Evaluation of the testing event sequence, using the Evaluation Engines, created through the patterns of the knowledge base. The decision maker has the role of selecting the predictions that have a likelihood probability greater than the threshold. Figure 8 presents the functional blocks architecture, which are described hereafter.

Alarm data—represents the source of historical alarm data to be used for pattern learning. Typically it is a database containing a log of alarm occurrences, for a certain period of time.

Data Pre-processor—represents the process of eliminating alarm fields and organizing events in sequential form, generating the Sequential Events.

EventSets Creator—this block represents the execution of the EventSets method. The objective is to extract all the sequences of events that precede a target event, according to a time window, and storing them in the Apriori Context.

Apriori Selector—here, the Apriori algorithm is executed. The objective is to extract the most meaningful sequence patterns that are present in the Apriori Context, according to a value of minimum support. The resulting sequences are stored in the Apriori Result.

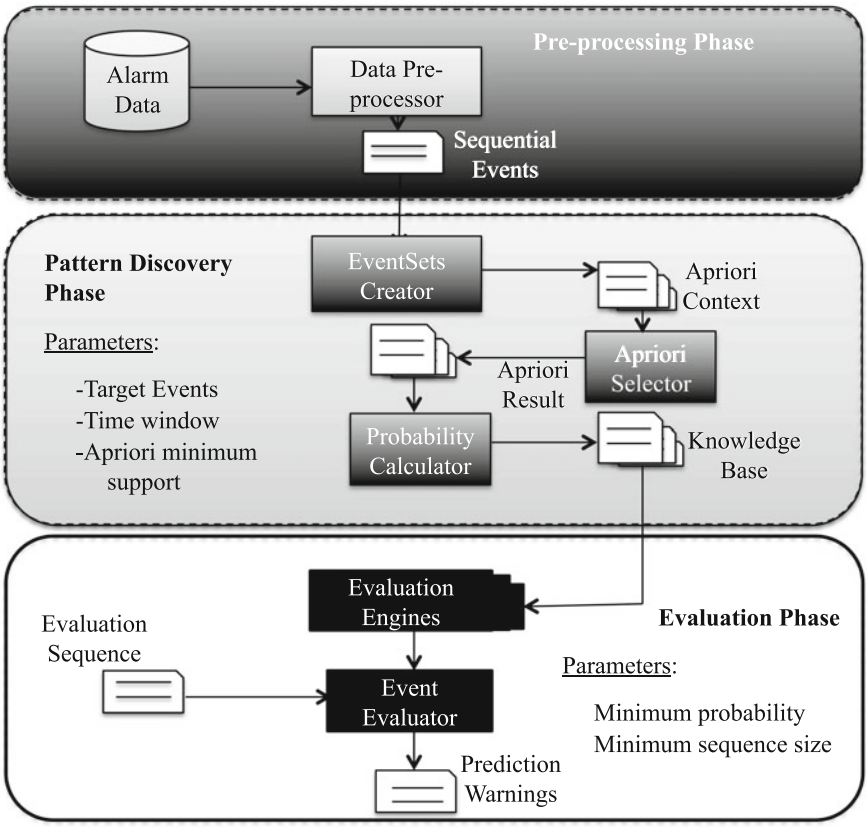

Fig. 8 Preventive monitoring tool's functional blocks architecture

Probability Calculator—in this block, the likelihood probabilities of each pattern in the Apriori Result will be calculated, so that the Knowledge Base is generated, and the pattern learning process is completed.

Evaluation Engines—these engines represent the Machine Learning instances used in the Evaluation phase. Each EE contains the FSM's representing the sequence patterns of the Knowledge Base.

Event Evaluator—this block represents the execution of the Evaluation algorithm. The objective is to compare the evaluation sequence and the state machines on the Evaluation Engines, and check if any pattern was recognized.

The *Decision Maker* is built into the Event Evaluator. The goal is to select, from the recognized patterns, only those that have probability greater than minimum probability.

In terms of prediction accuracy, this solution was able to sustain a rate of 72 % correct predictions, for patterns learned from 6 days of historical alarm data acquired from a real network infrastructure.

7 Conclusions and Further Applications of AI in ICT

The high pace of technological development on ICT requires increasingly new approaches to deal with current problems and new challenges due to increased complexity. This chapter presented some examples whether the application of AI techniques bring significant benefits. Experimental results showed evidence for significant improvements on the different application scenarios. However, further improvements are still possible by comparing the performance of different AI approaches to solve the aforementioned problems.

7.1 Future Work on the Reported Solutions

There are significant improvements that can be carried out in future work for the presented solutions, concerning the usage of AI techniques. For instance, the work on Virus Epidemic Prediction could be extended to integrate a large number of real biosensors for sampling people (or fetus) health status among a social network group.

Concerning Smart Building Energy Management, we are currently evaluating other AI approaches: employing user occupation detection technologies based on Bluetooth4.0 iBeacons; applying alternative learning algorithms (e.g. neural networks, more complex than k-means clustering); and employing game strategies to motivate and guide room occupants' vote (e.g. gathering green points in social games for user's votes and energy consciousness behaviors).

Intelligent Middleware for Cloud Robotics could also benefit from the application of a probabilistic model to optimize the usage of resources. Furthermore, a more powerful AI technique could allow extra functionality, such as taking decisions based on applications logic besides the monitored resources.

Client Profiling on Telecommunication networks should be tested with alternative, more efficient, machine learning approaches, such as Support Vector Machine (SVMs). Concerning dataming for prediction of rare events such as telecommunication alarms, we are investigating currently other approaches for improving the classifier, such as:

- Feeding the generated probabilistic rules into a decision tree algorithm, to further refine the rules prediction precision with respect to the training set
- Markov Logic Networks [52] as an alternative technology for the classifier.

The described solutions are examples of AI techniques being employed to solve ICT problems. Indeed, many other ICT areas can benefit from AI on order to solve pressing problems, as addressed by current research work and described hereafter.

7.2 Further Applications of AI on ICT

The World is experiencing a continuous technological development enabling the deployment of miniaturized and low-cost electronics in a variety of sectors. Agriculture is surely one of them. Playing an important role in the economy of every nation, agricultural production has been experiencing the continuous improvement of its processes and techniques, which is the focus of the precision agriculture (PA) concept. The intention of precision agriculture is, by collecting real-time data from the environment, to improve products quality as well as maintaining a sustainable agriculture. To accomplish this, there is a need of optimizing the resources used in the agricultural processes, mainly in the irrigation system. Water plays a crucial role in plants lifecycle, including germination, photosynthesis and nutrition processes. Most of the times, the water provided by natural precipitation is not enough in order to provide the amount of water the plants need to grow in a healthy way. Currently, agriculture consumes about 70% of the fresh water [53]. This percentage can be decreased performing efficient water management through precision Irrigation. The water is applied in an efficient and optimized way, in the right place, at the right time and in the right amount. It brings wide benefits, such as water savings, money savings as well as the improvement of crop quality. Hence, we recently developed a general architecture divided in three main components: a wireless sensor and actuator network component, a cloud platform component running a smart irrigation algorithm (using web-based weather services), and a user application component, to address a variety of distinct scenarios, such as agriculture, greenhouses, golf courses and landscapes. The solution was recently tested in a farm for the irrigation of a field with peach trees.

All these concepts have the potential to positively impact the livestock sector, changing the way monitoring is done on this sector. A cloud-based WSN prototype system for the livestock sector is currently being tested on dairy sheep and cows, in different Portuguese farms, validating the solution in real application scenarios. This monitoring is centered on the collection of the location of each animal and processing it, extract valuable information for the farmer. We are currently employing machine learning techniques for detecting alterations in the normal bustle of each animal, for inferring the propagation of possible diseases in the flock of sheep, thefts, etc.). Other potential applications of AI may bring important benefits, such as from combining this system with other sources of information (e.g., milking systems, etc.) in order to have a broader vision of the business.

On a different sector, that of law enforcement agencies (LEAs), over the past years many commercial Real-Time Tracking Management systems (RTMS) were introduced into the market. The solutions started to be designed for simple tracking purposes, but vendors soon realized that tracking systems would benefit from the design of new services (service layer). Recently we conceived a RTMS solution and corresponding service layer for LEAs [54]. This solution was created according to a list of requirements gathered from one of the Portuguese LEAs, GNR (Guarda Nacional Republicana) to fully understand the generic daily challenges.

Main requirements raised by GNR strive with issues such as: cost, data exchange security, bi-directional communication services, network performance, fault tolerance and user-friendly interface. The solution provides a risk map to law officers according to data gathered from the system. We are currently investigating AI techniques to classify regions on the risk map according to previous events reported to the system.

Wireless Local Area Networks (WLANs) are being widely deployed at private homes and public spaces, with a tendency to spread further. At the same time, end-users expect to have access to broadband mobile services at low cost and in a way that is at the same time easy and intuitive. This is not a trivial problem, given that the deployment of new radio systems implies costs for access operators, which need strong reasons to invest on these new systems. Moreover, the currently deployed private WLAN environments are normally not used at their full capacity.

Network virtualization is a technique that when applied to wireless end-user devices is the basis to e.g. take advantage of having several wireless access points around a single user, which in normal conditions this would result in worst signal conditions or signal fading. Furthermore, it is common for users nowadays to have electronic devices with multiple networking capabilities. Personal computing devices, e.g., laptops, PDAs, smartphones, are typically equipped with several networking interfaces ranging from different flavors of Wireless Fidelity (Wi-Fi) to Ethernet, GPRS, UMTS, and Bluetooth. Adding to the diversity of network interfaces embedded on end-user devices, the typical Internet end-user has at her disposal a set of applications with significantly different bandwidth requirements and which comprise multimedia services, gaming, as well as collaboration, among others. However, most services provided today to the end-user simply take advantage of one network interface at a time. This perspective is bound to change due to the fact that more and more, different Service Providers (SP) serves the same household or enterprise location. Hence, we are using optimization techniques [55, 56] for taking advantage of the simultaneous usage of multiple network interfaces, considering virtualization on the end-user device to be able to take advantage of available resources. This traffic-engineering technique, similar to multihoming and load-balancing techniques (which have been used to give networks some redundancy and redirect traffic flows based on the device necessities, such as power, signal strength, available bit rate, etc.) schedules in real-time the destination interface for each traffic flow, based on monitoring of network conditions, to maximize the overall throughput.

Considering green telecommunication networks, the current Internet infrastructure consumes large amounts of energy because network elements are always working at their full capacity even with low traffic demands. This waste of energy (by the Internet infrastructures) can be reduced, by allowing some network elements to enter into energy saving modes. However, this may lead to a network performance decrease. We have described a new proposal [57] using optimization techniques for an integrated energy saving model to both the current Internet and Publish-Subscribe architectures, taking into account the tradeoff between energy saving and network performance. This is mainly achieved by classifying the network elements according to their importance in the packet delivery process. The solution was implemented

and evaluated for both current and future Internet networks, using the NS3 simulator, and experimental results showed evidence for 45 and 23 % of average energy consumption reduction in scenarios of low and high traffic demand, respectively.

Telecommunication operators need to deliver their clients not only new profitable services, but also good quality, personalized and interactive content. One of the main concerns for current multimedia platforms is the provisioning of content to end-users that provide them a good Quality of Experience. This can be achieved through new interactive, personalized content applications, as well by improving the image quality delivered to the end-user. Hence, in the multimedia field, it was recently proposed the application of computer vision techniques for exploiting new video coding mechanisms [58]. The solution employs computer vision techniques to produce extra object information, using the à priori availability of 3D models of objects, which further expands the range of video personalization possibilities on the presence of new video coding mechanisms. Another work in this field proposes intelligent approaches for adaptation and distribution of personalized multimedia content [59]. The reported solution will allow SPs to provide the end-user with automatic ways to adapt and configure the (on-line, live) content to their tastes–and even more—to manipulate the content of live (or off-line) video streams (in a way that Photo Editing did for images or Video Editing, into a certain extent, to off-line videos).

Acknowledgments Different parts of this work have been carried in cooperation with companies: YDreams Robotics, SenseFinity and Nokia Siemens Networks. Parts of this work have also been partially funded by different research projects: CMU-Portuguese program through Fundação para Ciência e Tecnologia, project AHA—Augmented Human Assistance, AHA, CMUP-ERI/HCI/0046/2013. Harvard Medical School Portugal Collaborative Research Award HMSP-CT/SAU-ICT/0064/2009: Improving perinatal decision-making: development of complexity-based dynamical measures and novel acquisition systems. The author wishes to thanks the different contributions of researchers on his team for the research work, namely João Andrade, Vitor Mansur, Rui Francisco, Diogo Teixeira, Ivan Caravela, José Almeida, Nelson Sales and João Ambrósio, as well as the research collaboration of Paulo Carreira on the smart building project, Orlando Remédios from SenseFinity on AI for agriculture and geofence, as well Nuno Borges from Nokia Siemens Networks.

References

1. Arsenio, A., Serra, H., Francisco, R., Andrade, J., Serrano, E., Nabais, F.: Internet of intelligent things—bringing artificial intelligence approaches for communication networks. In: Inter-Cooperative Collective Intelligence: Techniques and Applications, vol. 495, pp. 1–37. Springer (2014)
2. Zhang, D., Guo, B., Li, B., Yu, Z.: Extracting social and community intelligence from digital footprints: an emerging research area. In: Ubiquitous Intelligence and Computing, pp. 4–18. Springer (2010)
3. Rothenberg, R., Costenbader, E.: Empiricism and theorizing in epidemiology and social network analysis. Interdisc. Perspect. Infect. Dis. **2011** (2011)
4. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. Phys. Rev. Lett. **86**(14), 3200–3203 (2001)

5. Schumm, P., Scoglio, C., Gruenbacher, D., Easton, T.: Epidemic spreading on weighted contact networks. In: IEEE 2nd BioInspired Models of Network Information and Computing Systems, pp. 201–208 (2007)
6. El-Sayed, A., Scarborough, P., Seemann, L., Galea, S.: Social network analysis and agent-based modeling in social epidemiology. Epidemiol. Perspect. Innovations EP+I **9**(1) (2012)
7. Chen, Y., Tseng, C., King, C.: Incorporating geographical contacts into social network analysis for contact tracing in epidemiology: a study on Taiwan SARS data. In: Advances in Disease Surveillance: Abstracts from the 2007 Conference of the International Society for Disease Surveillance (2007)
8. Andrade, J., Arsenio, A.: Epidemic estimation over social networks using large scale biosensors. Advanced research on hybrid intelligent techniques and applications. IGI Global (2015)
9. Gorder, P.: Computational epidemiology. Comput. Sci. Eng. **12**(1), 4–6 (2010)
10. Yang, R., Wang, L.: Development of multi-agent system for building energy and comfort management based on occupant behaviors. Energ. Build. **56**, 1–7 (2013)
11. Mansur, V.: Energy efficiency optimization through occupancy detection and user preferences. M.Sc. thesis, IST-UTL, June 2014
12. Qiao, B., Liu, K., Guy, C.: A multi-agent system for building control. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 653–659 (2006)
13. Klein, L., Kwak, J., Kavulya, G., Jazizadeh, F., Becerik-Gerber, B., Varakantham, P., Tambe, M.: Coordinating occupant behavior for building energy and comfort management using multi-agent systems. Autom. Constr. **22**, 525–536 (2012)
14. Mozer, M.: Lessions from an adaptive house. Smart Environ. 271–294 (2005)
15. Sierra, E., Hossian, A., Rodriguez, D., Britos, P.: Intelligent systems applied to optimize building's environments performance. Sierra **276**, 237–244 (2008)
16. Doctor, F., Hagras, H., Callaghan, V.: A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. IEEE Trans. Syst. **35**(1), 55–65 (2005)
17. Dodier, R.H., Henze, G.P., Tiller, D.K., Guo, X.: Building occupancy detection through sensor belief networks. Energ. Build. **38**(9), 1033–1043 (2006)
18. Rashidi, P., Cook, D.J.: Mining and monitoring patterns of daily routines for assisted living in real world settings. In: Proceedings of the ACM international Conference on Health Informatics—IHI'10, pp. 336–345 (2010)
19. Padmanabh, K.: iSense: a wireless sensor network based conference room management system. In: Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy—Efficiency in Buildings, pp. 37–42 (2009)
20. Dong, B., Andrews, B.: Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings. In: Proceedings International IBPSA Conference, pp. 1444–1451 (2009)
21. Erickson. V.: OBSERVE: occupancy-based system for efficient reduction of HVAC energy. In: 10th International Conference on Information Processing in Sensor Networks (IPSN), pp. 258–269 (2011)
22. Kim, Y., Charbiwala, Z.: Spotlight: personal natural resource consumption profiler. In: Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets) (2008)
23. Murakami, Y., Terano, M., Mizutani, K., Harada, M., Kuno, S.: Field experiments on energy consumption and thermal comfort in the office environment controlled by occupants' requirements from PC terminal. Build. Environ. **42**(12), 4022–4027 (2007)
24. Gao, G., Whitehouse, K.: The self-programming thermostat: optimizing setback schedules based on home occupancy patterns. In: Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (2009)
25. Lu, J., Sookoor, T., Srinivasan, V.: The smart thermostat: using occupancy sensors to save energy in homes. In: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (2010)
26. Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., Weng, T.: Occupancy-driven energy management for smart building automation. In: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building—BuildSys'10 (2010)

27. Oldewurtel, F., Sturzenegger, D., Morari, M.: Importance of occupancy information for building climate control. Appl. Energy **101**, 521–532 (2013)
28. Delaney, D., O'Hare, G., Ruzzelli, A.: Evaluation of energy-efficiency in lighting systems using sensor networks. In: Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings—BuildSys'09 (2009)
29. Harle, R.K., Hopper, A.: The potential for location-aware power management. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 302–311 (2008)
30. Mansur, V., Carreira, P., Arsenio, A.: Learning approach for energy efficiency optimization by occupancy detection. In: Proceedings of The First International Conference on Cognitive Internet of Things Technologies, Rome, Italy (2014)
31. MacKay, D.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge. pp. 316–322 (2003)
32. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Wheeler, R., Andrew, N.: ROS: an open- source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, no. 3, 2 (2009)
33. Metta, G., Fitzpatrick, P., Natale, L.: YARP: yet another robot platform. Int. J. Adv. Robot. Syst. (2006)
34. Broxvall, M., Seo, B., Kwon, W.: The PEIS kernel: a middleware for ubiquitous robotics. In: Proceedings of the IROS-07 Workshop on Ubiquitous Robotic Space Design and Applications (2007)
35. Kranz, M., Rusu, R., Maldonado, A., Beetz, M., Schmidth, A.: A Player/Stage System for Context-Aware Intelligent Environments. In: Proceedings of the System Support for Ubiquitous Computing Workshop (UbiSys) (2006)
36. Rusu, R., Maldonado, A., Beetz, M., Kranz, M., Mosenlechner, L., Holleis, P., Schmidt, A.: Player/Stage as Middleware for Ubiquitous Computing. In: Proceedings of the 8th Annual Conference on Ubiquitous Computing (Ubicomp) (2006)
37. Hunziker, D., Gajamohan, M., Waibel, M., D'Andrea, R.: Rapyuta: the roboearth cloud engine. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2013)
38. Francisco, R., Arsenio, A.: Intelligent multi-platform middleware for wireless sensor and actuator networks. In: Proceedings of the First Conference on Cognitive Internet of Things Technologies (Coiote2014). Selected paper on Lecture Notes in Computer Science: Users-centric IoT. Springer (2015)
39. Duffield, N., Lund, C., Thorup, M.: Estimating flow distributions from sampled flow statistics. In: Proceedings of the ACM SIGCOMM (2003)
40. Fernandes, S., Correia, T., Kamienski, C., Sadok, D., Karmouch, A.: Estimating properties of flow statistics using bootstrap. In: IEEE MASCOTS (2004)
41. Duffield, N., Lund, C., Thorup, M.: Properties and prediction of flow statistics from sampled packet streams. In: ACM SIGCOMM Internet Measurement Workshop (2002)
42. Chen, H., You, F., Zhou, X., Wang, C.: The study of DPI identification technology based on sampling. Inf. Eng. Comput. Sci. (2009)
43. Inácio, P.: Study of the impact of intensive attacks in the self-similarity degree of the network traffic in intra-domain aggregation points. Ph.D. thesis, University of Beira Interior, Covilhã, Dec 2009
44. Neto, M., Gomes, J., Freire, M., Inácio, P.: Real-time traffic classification based on statistical tests for matching signatures with packet length distributions. In: Proceedings of the 19th IEEE International Workshop on Local and Metropolitan Area Networks (IEEE LANMAN 2013), Brussels, Belgium, 10–12 April 2013
45. Diogo, T.: Smart and automatic network configuration. M.Sc. thesis, Universidade Tecnica de Lisboa (2013)
46. Hätönen, K., Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H.: Knowledge discovery from telecommunication network alarm databases, pp. 115–122 (1996)
47. Graf, S., Prinz, A.: Time in state machines. Abstr. State Mach. 217–232 (2005)

48. Hitike, K., Khalifa, O.: Comparison of supervised and unsupervised learning classifiers for human posture recognition. Comput. Commun. Eng. 1–6 (2010)
49. Yao, X.: Evolving artificial neural networks. In: Proceedings of the IEEE, vol. 87, pp. 1423–1447 (1999)
50. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE, vol. 77, pp. 257–286 (1989)
51. Weiss, G.: Timeweaver: a genetic algorithm for identifying predictive patterns in sequences of events. In: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann (1999)
52. Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. **62**(1–2), 107–136 (2006)
53. Fry, A.: Facts and trends. water. In: World Business Council for Sustainable Development. Earthprint Ltd (2005)
54. Almeida, J.: Tracking management system for security enhancement. M.Sc. thesis, Universidade Tecnica de Lisboa (2014)
55. Mota, J., Arsenio, A., Sofia, R.: Combining heterogeneous access net works with ad-hoc networks for cost-effective connectivity. API Rev. **1**(1), 41–43 (2010)
56. Maurício, J., Arsenio, A., Sofia, R.: Wireless resources aggregation: leveraging multiple WiFi virtual interfaces. API Rev. **1**(1), 47–49 (2010)
57. Arsenio, A., Silva, S.: Energy efficiency and failure recovery mechanisms for communication networks. In: Khan, S., Mauri, J. (eds.) Green Networking and Communications: ICT for Sustainability. CRC Press (2013)
58. Arsenio, A.: Application of computer vision techniques for exploiting new video coding mechanisms. In: Srivastava, R., Singh, S., Shukla, K. (eds.) Research Developments in Biometrics and Video Processing Techniques, pp. 156–182. Information Science Reference, Hershey, PA (2014)
59. Arsenio, A.: Intelligent approaches for adaptation and distribution of personalized multimedia content. In: Kanellopoulos, D. (ed.) Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools, pp. 197–224. Information Science Reference, Hershey, PA (2013)

A Statistical Framework for Mental Targets Search Using Mixture Models

Taoufik Bdiri, Nizar Bouguila and Djemel Ziou

Abstract Image retrieval is usually based on specific user needs that are expressed under the form of explicit queries that lead to retrieve target images. In many cases, a given user does not possess the adequate tools and semantics to express what he/she is looking for, thus, his/her target image resides in his/her mind while he/she can visually identify it. We propose in this work, a statistical framework that enables users to start a search process and interact with the system in order to find their target "mental image", using visual features only. Our bayesian formulation provides the possibility of searching multi target classes within the same search process. Data are modeled by a generalized inverted Dirichlet mixture that also serves to quantify the similarities between images. We run experiments including real users and we present a case study of a search process that gives promising results in terms of number of iterations needed to find the mental target classes within a given dataset.

Keywords Mental search · Image retrieval · Bayesian models · Generalized inverted dirichlet · Mixture models

1 Introduction

New technological achievements during the recent years caused the appearance of large data collections that are complex to represent, analyze and search. The amount of information that could be derived from such collections can vary depending on

T. Bdiri (✉)
Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada
e-mail: t_bdiri@encs.concordia.ca

N. Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

D. Ziou
DI, Faculté des Sciences, Université de Sherbrooke, Sherbrooke, QC, Canada
e-mail: djemel.ziou@usherbrooke.ca

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*, Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_5

the purpose, context and intention of the users dealing with them. Therefore, many systems adopt query-based structures in order to satisfy context-aware needs from a given data collection [20]. In many real life situations, a search problem occurs naturally when a human being is interested in one or several concepts that exist within a certain amount of data. Such concepts could be visual, textual, or within any other information container that is usually represented by the system in a well defined feature space. A common application of such search processes comes with the recommendation frameworks where the system uses different feed backs from users in order to suggest them what could be categorized as "interesting items" [21, 27]. Some other systems collect data about the users behaviors, then search and suggest advertising items for commercial purposes [14, 16]. In many cases, a user is intentionally searching an item but has no clue about the features that a system is using to represent a concept, or lacks the needed tools and semantics to describe his/her needs and express them. Thus, he/she cannot provide a precise query to the system in order to tune the search process. We refer to this problem as the semantic gap between the user and the system. In order to solve such a problem, some works have proposed a mental matching which consists of a search process during which the system tries to identify the mental target of a given user without having an explicit query. An overview about mental matching can be found in [11] where the authors track the early works on this field and propose a new bayesian framework to search an image category based on a mental image. They mainly design a system where several images are displayed to a user, and he/she has to select the image that appears to be the closest to the image he/she has in mind, until the system shows a target image. In this paper, we propose to extend the work of [11] which has the advantage of being query-free, in order to (1) cover multi targets category search, (2) include the possibility of a multiple images selection and a no preference choice when looking for a target category during the search process, and (3) use a new data model that serves to model the data and to measure similarities between different images using the generalized inverted Dirichlet distribution [4, 19].

This paper is structured as follows. In Sect. 2 we present the problem and the techniques that have been proposed to solve it, as well as the main contributions of this work. The proposed framework is introduced in Sect. 3. In Sects. 4, 5, 6 and 7, we detail the Data Model, Update Model, Answer Model and Display Model, respectively. We present our experimental results in Sect. 8. In Sect. 9 we interpret the results and we conclude in Sect. 10.

2 Related Work

When they first appeared, images databases were carefully annotated with keywords by experts, in order to easily browse them and retrieve images by query. With the tremendous increase of the volume of available images that are published on a daily bases by millions of internet users, manual annotation turns to be impossible to realize. To solve this issue, researchers started by proposing an automated tagging based

on annotation propagation such as in [13, 22]. Although those methods were successful, it is not always possible to formulate an explicit query in order to retrieve images. Many users lack the necessary vocabulary and are not able to formulate a system understandable query that could lead them to their target images. Thus, many researchers based their work on the query by image content (QBIC) framework that was initially published in [12], where a query is an image that can serve to find similar images within a database. Still, the QBIC system needs example images to serve as a graphical query. Those example images that serve to start the search process are not always available which represents a problem known as the page zero problem. On the other hand, to help the system making suggestion that may interest users, many researchers have proposed the relevance feedback such as in [15, 26, 28]. Relevance feedback uses high level information provided by the user about a given concept, using an iterative fashion, in order to adjust the data representation and converge toward the discovery of new elements that represent a value to that user. In order to construct a solid model founded on a strong mathematical background, many model-based framework have been developed for content-based images retrieval. The work in [25] proposes to minimize the probability of retrieval error by combining feature selection and similarity measures into a Bayesian formulation, while the work in [5] uses the context-aware formulation to identify several cases where the context serves to generate more accurate recommendations. Some other works propose hybrid models that combine both generative and discriminative learning, such as in [29] where the authors propose a hybrid model-based framework for efficient image retrieval . While those works were not dedicated to the page zero problem, they prove that statistical frameworks can give accurate results. The first model-based work that was trying to solve the page zero problem is the mental matching which tries to find a "mental image" that resides in the mind of a user without having an explicit query. It was first pioneered by the work in [6]. The search process was done iteratively, and at every round the user was asked to choose the closest image to the target image that resides in his/her mind, among two displayed images by the search engine. The work in [6] served as a basis for the framework proposed by [11] which extended their work to cover semantic target category search using a bayesian model that includes a pair of positive and negative answer models. The statistical framework proposed in [11] has been adopted and extended for large-scale image collections of millions of images in the HEAT retrieval system proposed in [23] , and extended in [24]. Still, the framework proposed by [11] is limited to one single target category within the same search process. The user has to repeat the process N times if he/she is targeting N categories. Also, a user is always forced to choose an image that is closest to the image in his/her mind even if there is not anyone displayed that matches the targeted mental images. Moreover, a user is allowed to have one single selection, while he/she could be interested in several images that could lead to the target category. In this paper we extend the work proposed in [11], to cover many target images search within a single mental search process, including the non selection and multi selection preferences. We also use a new data model constructed by the generalized inverted Dirichlet (GID) mixture that has proven its capability to robustly model and cluster

multi-dimensional data [4, 19]. This work focuses on visual concepts and mainly deals with images but it is noteworthy to mention that a concept can be any other information representation that a user can use to interact with the system.

3 The Framework Structure

Let $\Omega = \{X_1, X_2, \ldots, X_N\}$ be a set of N images. The main purpose of this work is to determine a subset $S \subset \Omega$ that represents a set of target images that matches the visual interests in the mind of a given user. S could be formed by diverse images' categories $\{S_j\}$ that do not necessarily have visual similarities at the system level, or have different semantic meanings for the user. Thus, S can be written under the following form:

$$S = \bigcup_{j=1}^{M} S_j \tag{1}$$

where $S_j \subset \Omega$, $j = 1, \ldots, M$, are the different target classes forming S and M represents their number. We assume that the user knows exactly how many classes M he/she is targeting. We also assume that if an image that belongs to a target class is displayed, the user will be able to identify it. If the user is not interested anymore in a given target class S_k the search problem is reduced to find:

$$S = \bigcup_{\substack{j=1 \\ j \neq k}}^{M} S_j \tag{2}$$

The mental search consists of several iterations during which a set of different images $D \subset \Omega$ is displayed to the user. If $D \cap S_j = \emptyset$ the user specifies the set of images $\{X_i\}_j \subset D$ that are the closest to what he/she might have in mind in order to approach S_j. We associate a random variable Y_{X_k} with each image $X_k \in \Omega$ such that $Y_{X_k} = j$ if $X_k \in S_j$ and $Y_{X_k} = 0$ if $X_k \notin S$. In other terms we have:

$$\begin{aligned} S &= \{X_k \in \Omega \ / \ Y_{X_k} \neq 0\} \\ S_j &= \{X_k \in \Omega \ / \ Y_{X_k} = j\} \end{aligned} \tag{3}$$

During the search process Y_{X_k} is updated according to the user responses to the different displays $\{D_t\}$. Let B_t denote the responses of the user for the first t displays. The distribution of $Y_{X_k} = j$ knowing B_t is represented by:

$$p_t(X_k)_j = P(Y_{X_k} = j | B_t), \quad j = 1, \ldots, M \tag{4}$$

As S is seen by the system as a random subset, we have no prior knowledge about it. Therefore we initialize the values of $p_0(X_k)_j$ as follows:

$$P_0(X_k)_j = \frac{1}{M+1}, \quad j = 1, \ldots, M, \quad k = 1 \ldots N \tag{5}$$

which is a uniform distribution over all the $M+1$ classes, the $(M+1)$th component is an eventual additional class which represents the negative set of images that represent no interest to the user. Note that if we consider the specific case $M = 1$ we find the same framework structure proposed in [11]. Our work treats a generalized case where a user can choose not to select any image, or to select multiple images. We also suggest a different model describing the data that is discussed in Sect. 4. The framework we propose is as follows: we design a graphical interface through which the user will interact with the system in order to identify his/her target classes within a dataset using the visual features only. This interaction is realized through different iterations, during which the system would display at iteration t a set of images D_t such that:

$$\begin{aligned} |D_t| &= N_t \\ N_t &= \sum_{j=1}^{M} N_t(j) \end{aligned} \tag{6}$$

N_t is the total number of displayed images at time t, and $N_t(j)$ is the number of images proposed for a specific target class S_j, $j = 1, \ldots, M$. The user would select the images that he/she might find interesting and specify their corresponding classes among the M available classes defined at the beginning. e.g. say we have a scrolling list that can be used for each displayed image in order to specify its class j (see Fig. 1). The user responses enable the system to update its parameters and come up

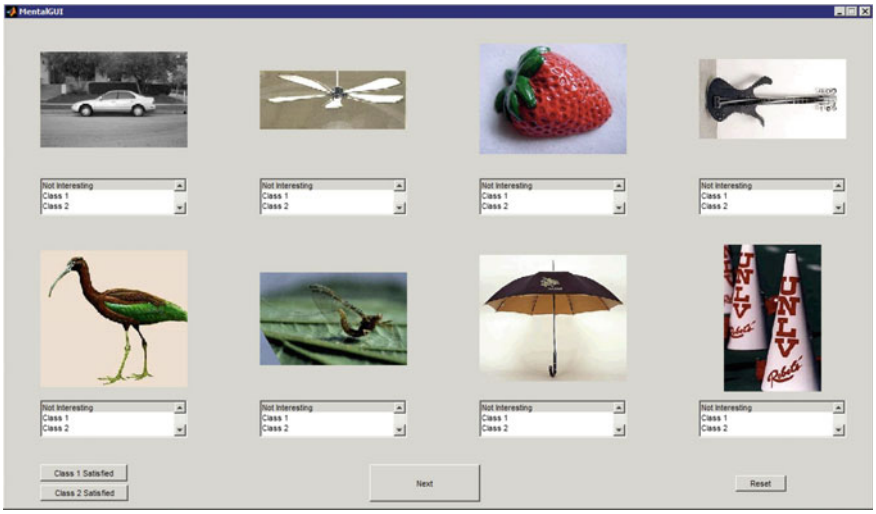

Fig. 1 Example of the first display of 8 images

with a new display D_{t+1}. The user repeats the same selection process until he/she is satisfied of the search process concerning a target class S_j. In order to develop a such search process, we mainly design four models:

- **Data Model**: This component covers the images representation at the system level, and how the system perceives the data. It can be constructed in a supervised or unsupervised way.
- **Update Model**: This component computes for each image category, $p_{t+1}(X_k)_j$ in terms of $p_t(X_k)_j$ and the user's answers at step t.
- **Answer Model**: This component specifies for each image $X_k \in \Omega$ the probability that a user chooses an image $X_i \in D$ given $Y_{X_k} = j$.
- **Display Model**: This component specifies which images to display at step t, basing on the search history.

In the following we develop each of these components.

4 Data Model

We propose to model the data using the statistical framework proposed in [1–3]. We develop the framework using the GID mixture model which is a powerful mathematical tool to model data and cluster it [4, 19]. To build the data model we need to (1) define a feature space to which we map our images in order to enable the system to represent their contents, (2) to build a GID mixture model and estimate the parameters of each of its components (3) to set an update approach to the model when new images are added to the database. We propose to use the local Histogram of Oriented Gradient (HOG) descriptor proposed in [7], in order to extract the images features. Known for its efficiency in terms of detecting local characteristics of images, the HOG descriptor is often used in computer vision and image processing areas for the purpose of object detection. The GID mixture model supposes that the data is positive, which makes the HOG suitable for consideration as it generates descriptors having positive values. In our experiment each image was represented by a feature vector whose dimension is $D = 81$. Let us consider a set Υ of N D-dimensional vectors, that represents the features extracted from the set of images $\Omega = \{X_1, X_2, \ldots, X_N\}$, such that $\Upsilon = \{\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_N\}$, where $\boldsymbol{F}_n$ is the feature vector of the image X_n. Let C denotes the number of different components forming a flat mixture model at the system level. We assume that Υ is controlled by a mixture of GID distributions such that the vectors follow a common probability density function $p(\boldsymbol{F}_n|\Theta)$, where Θ is the set of its parameters. The GID distribution was introduced by Lingappaiah [17] and the GID mixture is expressed as follows [1, 4, 19]:

$$p(\Upsilon|\Theta, \boldsymbol{\pi}) = \prod_{n=1}^{N} \Big(\sum_{j=1}^{C} \pi_j \prod_{d=1}^{D} \frac{\Gamma(\alpha_{jd} + \beta_{jd})}{\Gamma(\alpha_{jd})\Gamma(\beta_{jd})} \frac{F_{nd}^{\alpha_{jd}-1}}{(1 + \sum_{l=1}^{d} F_{nl})^{\gamma_{jd}}} \Big) \tag{7}$$

where $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_C\}$, with $\boldsymbol{\theta}_j = (\alpha_{j1}, \beta_{j1}, \ldots, \alpha_{jD}, \beta_{jD})$, $\gamma_{jd} = \beta_{jd} + \alpha_{jd} - \beta_{j\{d+1\}}$ for $d = 1, \ldots, D$ with $\beta_{j\{D+1\}} = 0$. $\boldsymbol{\pi}$ is the vector of mixing weights such that $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_C)$, $\pi_j > 0$ and $\sum_{j=1}^{C} \pi_j = 1$. We propose to model our features vectors by a GID mixture model, that can serve to build a hierarchical model if needed using the approach proposed in [1–3]. The details about the estimation of the parameters, the model construction and its update when new data is available are out of the scope of this work and are discussed in details in [1]. The proposed model serves to cluster the data into classes at the system level.

5 Update Model

The update model updates the posterior probabilities $p_t(X_k)_j$ according to the responses of the users. Let us denote X_{D_t} the user response to the display D_t, then we have:

$$X_{D_t} = \{X_{D_{t0}}, \ldots, X_{D_{tM}}\} \tag{8}$$

X_{D_t} is the whole user response set for a display D_t, and $X_{D_{tj}}$ is the user selection for a specific class j at time t for the same display D_t. As we consider $(M+1)$ selection sets in Eq. (8), we consider the non selected images as a "user selection" for the negative images class, so when a user decides not to select specific images, this is considered as being a choice for the class containing the "not interesting" images set. Naturally, the system has no prior knowledge about the user mental targets, therefore, in order to constitute D_1, the most trivial method is to select a number N_1 of images with a selection probability equal to $\frac{1}{|\Omega|}$. A more adequate method is to use our data model in order to suggest the initial display. Recall that we have a mixture model composed of C clusters that group similar images together. We propose to randomly select M clusters with the probability $\frac{1}{C}$, and then select from each selected cluster j, $N_1(j)$ images to display in D_1, with the probability $\frac{1}{|j|}$, with $|j|$ the cardinal of the cluster j and $j = 1 \ldots M$. Thus, we can construct D_1, composed of N_1 images such that $N_1 = \sum_{j=1}^{M} N_1(j)$, with $N_1(j) \geq 0$, $j = 1 \ldots M$. At time t, the search history is expressed as follows:

$$B_t = \{X_{D_1}, X_{D_2}, \ldots, X_{D_t}\} \tag{9}$$

we construct B_{t+1} such that

$$B_{t+1} = B_t \cup \{X_{D_{t+1}}\} \tag{10}$$

Note that our proposed model is not limited to a binary model for each target class, where we have a positive and a negative model as proposed in [11]. We rather have

$M+1$ classes, with the $(M+1)$th representing the negative class. The consideration of M binary models, where we have for each class a positive and negative model, increases the complexity and time cost when targeting many classes. Moreover when considering $M+1$ classes, we take into consideration that when an image $X_k \in \Omega$ does not belong to a target class S_j, it should not be systemically considered as a negative image, as it could belong to another target class S_k, with $k \neq j$. The selection probability of a displayed image X_D when $D = D_{t+1}$ and given $Y_{X_k} = j$, in order to approach the target class S_j, can be written under the form:

$$p(X_D = i | Y_{X_k} = j, D_{t+1} = D) = p_+(i, X_k, D)_j \tag{11}$$

We consider Eq. (11) as being the "positive answer model" for the class j, and basing on the work of [11], we update the distribution of $Y_{X_k} = j$ knowing B_{t+1} as follows:

$$\begin{aligned} p_{t+1}(X_k)_j &= p(Y_{X_k} = j | B_{t+1}) \\ &= \frac{p(X_D = i | Y_{X_k} = j, D_{t+1} = D) p_t(X_k)_j}{C_{t+1}} \\ &= \frac{p_+(i | X_k, D)_j \, p_t(X_k)_j}{C_{t+1}} \end{aligned} \tag{12}$$

where C_{t+1} is a normalizing factor such that:

$$C_{t+1} = \sum_{j=0}^{M} p_+(i | X_k, D)_j \, p_t(X_k)_j \tag{13}$$

note that we have:

$$\sum_{j=0}^{M} p_t(X_k)_j = 1, \quad \forall t, \quad \forall X_k \in \Omega \tag{14}$$

6 Answer Model

The answer model is based on the data model. The assumption that the user is not satisfied yet by the displayed images still hold. When considering the mixture model, we assume that the more an image is close to a mental target image belonging to a class j, the more likely their posterior probabilities will be close to each others. Let $(X_i)_j$ be the image selected by the user to be the closest image to the class j and let $(X_k)_j \in S_j$. We define the pseudo metric $d((X_i)_j, (X_k)_j)$ that we denote $d_j(X_i, X_k)$, between a selected image $(X_i)_j$ and a target image $(X_k)_j$, where $Y_{(X_k)_j} = j$, as follows:

$$d_j(X_i, X_k) = \sum_{c=1}^{C} \left(p(c|\boldsymbol{F}_i, \Theta, \boldsymbol{\pi}) - p(c|\boldsymbol{F}_k, \Theta, \boldsymbol{\pi})\right)^2 \tag{15}$$

Notice that $d_j(X_i, X_k)$ can be turned into a full metric distance, if we consider an equivalence classes such that: if $d_j(X_i, X_k) = 0$ it implies that $X_i \sim X_k$. We adopt the answer model form proposed by [11] for each target class S_j such that:

$$p_+(i|X_k, D)_j = \frac{\phi_+(d_j(X_i, X_k))}{\sum\limits_{X_d \in D} \phi_+(d_j(X_d, X_k))} \tag{16}$$

where $\phi_+(x)$ is a monotonically decreasing function, such that if $X_i, X_j \in D$ and $d_j(X_i, X_k) < d_j(X_j, X_k)$ we expect $p_+(i|X_k, D)_j > p_+(j|X_k, D)_j$. We propose to consider $\phi_+(x)$ as a Gaussian distribution with mean $\mu = 0$ and standard deviation σ. Thus, we have:

$$p_+(i|X_k, D)_j = \frac{e^{-\frac{1}{2}\left(\frac{d_j(X_i, X_k)}{\sigma}\right)^2}}{\sum\limits_{X_d \in D} e^{-\frac{1}{2}\left(\frac{d_j(X_d, X_k)}{\sigma}\right)^2}} \tag{17}$$

σ can be seen as a precision parameter, to specify how the distance metric should be perceived.

6.1 The No Preference/Selection Case

As we mentioned before, the user has the choice not to select any image, therefore, all the images that are not selected, are considered to be part of the "not interesting" class. When a user does not select any image that might be semantically and visually close to what he/she has in mind concerning a target class j, we propose to base the next suggestions on the last selection belonging to that class and reestablish all the display process basing on it such that:

$$X_{D_{t\,j}} = X_{D_{t-1\,j}} \tag{18}$$

The special case of not having a previous selection comes to surface when a user decides that the earliest displays are not interesting and considers that there is not any interesting image targeting the class j during all the previous iterations such that:

$$X_{D_{i\,j}} = \emptyset, \quad i = 1 \ldots t \tag{19}$$

In that case we use the negative answer to have an implicit selection of the target class j in order to be able to update our model. We assume that in terms of the

distance metric, the further an image is from the images of the negative class, the more probable it will belong to a target class. Thus, we consider a set $\{X_h\}$ that represents those furthest images from the selected images of the negative class and consider it as a selection for the target class j. This assumption enables the model to conserve its integrity in the early displays when no selection of target classes takes place. In this case we consider that D implicitly contains $\{X_h\}$ and the equations using D are used with a new $D = D_{implicit}$ such that:

$$D_{implicit} = D \cup \{X_h\} \tag{20}$$

still, when a target class has no selections, the system will not consider the implicit selections as a real user selection and will display a random selection from different non-displayed clusters when it suggests images for that class, as described in Sect. 5.

6.2 The Multi Selection Case

The multi selection case takes places when a user wants to select more than one single image in order to approach a target class j. We assume that the user selections are independent, so if we consider that the user selects J images to target the class j, we calculate the probability of the multi selection as follows:

$$p_+(i_1, \ldots, i_J | X_k, D)_j = \prod_{l=1}^{J} p_+(i_l | X_k, D)_j, \quad J \leq |D| \tag{21}$$

7 Display Model

According to our framework, at time t, we should display N_t images including $N_t(j)$ suggestions for the target class j. Naturally, an image should not be displayed twice, and should not have been already displayed in the past. In order to suggest images for a target class j, we rank the images, according to their $p_t(X_k)_j$ and we pick up the first $N_t(j)$ images. We define D_{t+1} as follows:

$$D_{t+1} = \bigcup_{j=1}^{M} D_{t+1_j} \tag{22}$$

where

$$D_{t+1_j} = \underset{\substack{D_j \subset \{\Omega \setminus B_t\} \\ |D_j| = N_{t+1}(j) \\ D_j \cap \{D_i\} = \emptyset}}{\arg\max} \sum_{X_k \in D_j}^{N_{t+1}(j)} p_t(X_k)_j \tag{23}$$

where $\{D_i\}$ in Eq. 23 is the set of the already picked images to be displayed in the display D_{t+1} for any other target classes k, such that $k \neq j$. We also propose, at first, to rank only the images that belong to the same cluster of the images selected by the user, that is defined by the mixture model, which leads to suggest images that are semantically alike. If a cluster does not have enough images to display, we extend the application of Eq. 23 on the whole set of images Ω. If the user selects images from different clusters we have two choices, either we select a defined number from each cluster to form the display of the target class j, or we build a hierarchical model forming a parent cluster containing the sub clusters (such as in [2, 3]) and then we select the images from it using Eq. 23.

8 Experimental Results

In order to test our framework we consider the publically avaiable Caltech101 dataset [10]. Caltech101 is composed of 8676 images of different objects representing 101 categories. These objects have different shapes and colors as shown in Fig. 2 which illustrates two randomly chosen samples from each category of the dataset. In our experiment we used all the 101 categories, and we have considered two target classes such that $M = 2$. The construction of the data model was established using the GID mixture model basing on the construction strategy that is proposed in [1], and using the HOG features. The modeling process resulted 124 components that compose the GID mixture. Those components can be considered as the system perception for the data in the HOG feature space. In order to estimate the number of representing classes of the data, we used the MML criteria, the GID Kullback-Leibler distance and a perception tolerance rate equal to 20 % to measure the similarity between the different components as it has been proposed in [1]. We designed a Graphical User Interface (GUI) that displays eight images at each iteration and we fixed the display number for each category such that $N_1(1) = N_1(2) = 4$, which means that the system should suggest four images from each target class. We also considered a precision parameter $\sigma = 1$ for the Eq. 17 to have the standard normal distribution.

The system starts by displaying eight images randomly, such that an image from each constructed component of the mixture is selected as explained in Sect. 5. Figure 1 shows a screenshot of the developed GUI that shows a first display D_1. The user selection is set, by default, to 'not interesting'. If some interesting images are displayed, the user selects them by specifying their corresponding classes. Basing on those selection, the system tries to find images that could belong to the mental target classes of the user. Using a normal PC, the suggestion of a new display takes an average of 1 second. The performance analysis of such systems remains hard to interpret as it is related to the satisfaction of users, and it involves human psychology and decision making. As a first approach, we have tried to measure users satisfaction by asking 20 users that are not familiar with the system to use the search interface in order to target some images from a target class. We asked them to give a satisfaction grade that goes from 1 to 10, one being the worse grade and 10 represents the full satisfaction grade.

Fig. 2 CalTech 101: two randomly chosen samples for each category

We obtained an overage equal to 7.95/10. We noticed that the dissatisfaction usually occurs with users who have missed an image from their category class, or mislead the system toward other categories. Indeed, it takes 16 iterations, for the system to show at least one image from each component of the mixture, considering 124 components with a screen size of 8 images to display. For those who miss an image that would lead them to the component containing their target class, they may spend a longer time to find their target images which lower their satisfaction of the system. In order to quantify the displaying results, we propose to illustrate how the posterior $p_t(X_k)_j$

Fig. 3 Target images from "buddha" and "sunflower" classes

changes over time. We choose two target classes that are constructed from the categories "sunflower" and "buddah". Our main purpose during the search process is to find eight specific images from each class. For the "sunflower", we propose to find the target class that represent the sunflowers with a tall stem, and for the "buddah" we propose to pick up the buddah that are constructed of gold. Figure 3 shows the 16 targeted images.

We report on the results after 23 displays, in Figs. 4 and 5, where we illustrate the posterior probability of the "buddha" class and the "sunflower" class, respectively. The blue stars shows the already displayed images that do not make part of the target images set, while the green ones represent the target images that are already displayed. The lines with red stars show the probabilities of target images. At the first display all the images have the same probability as the system has no input from the user. The first appearance of an image of buddha takes place in iteration 9 as shown in Fig. 4b, and the first appearance of a sun flower image takes place in Iteration 14 as shown in Fig. 5e. As the images of buddha appear before the appearance of the "sunflower" images, it is expected that the system displays a target image from the buddha set before showing a target image from the "sunflower" set. The first target image of buddha appears in iteration 11 as shows in Fig. 4c, and the first appearance of a target image of a "sunflower" appears in iteration 15 as shown in Fig. 5f. Figure 6 shows the display at Iteration 18, note that some target images have been already displayed at that stage. After 23 displays, the system was able to suggest 5 target images of "sunflower" and 7 target images of "buddha", which makes a total of 12 images out of 16.

9 Discussion

The system was able to converge toward the specified needs of the user, nevertheless it is noteworthy to mention that the framework is still very sensitive to the used features in order to measure the similarity between different concepts. The distance metric that we have used combines the HOG features and the probabilistic data model that we have constructed, further improvement may be done by using new features in order to compare the similarity between images, and new metrics approaches such

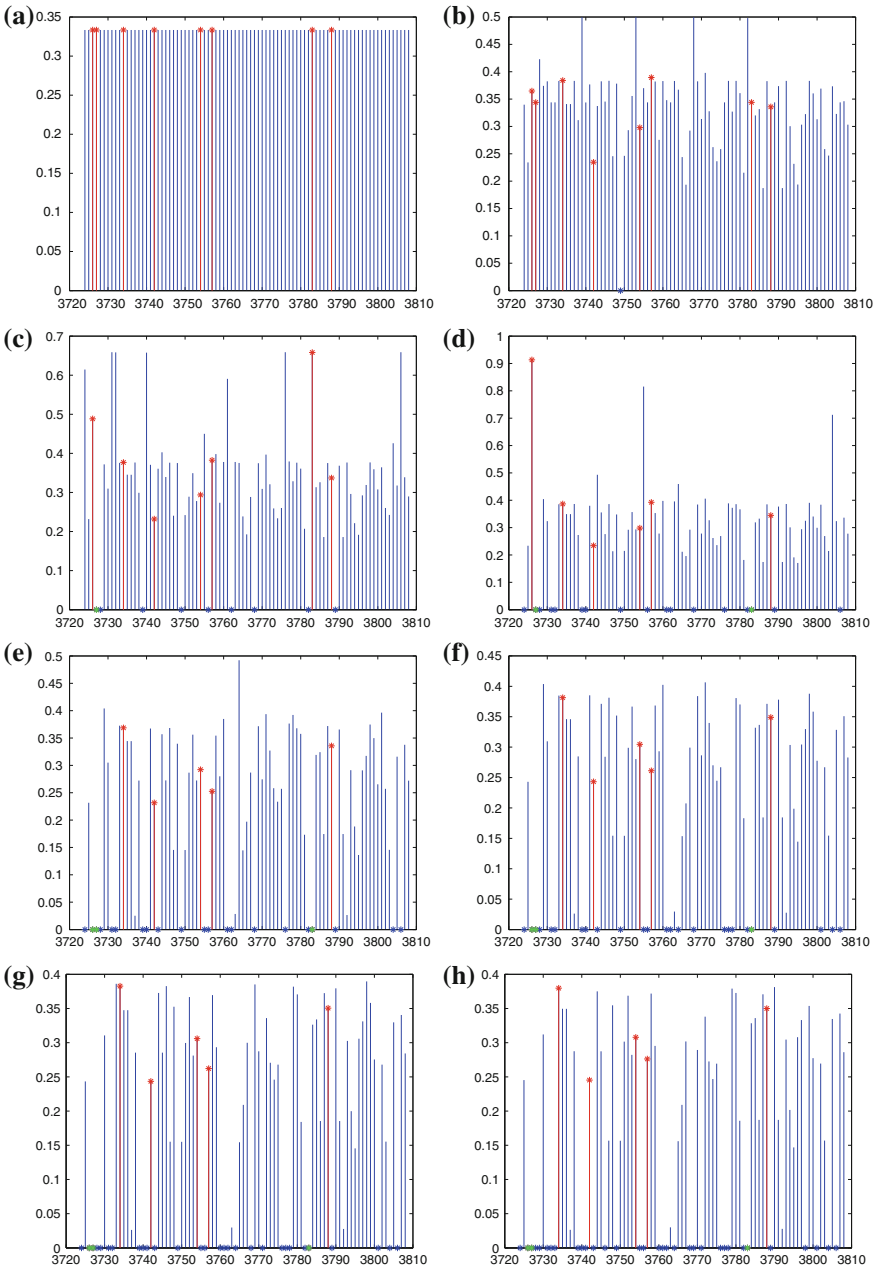

Fig. 4 Class "buddha": $p_t(X_k)_j$ evolution during different iterations **a** Iteration 1 **b** Iteration 9 **c** Iteration 11 **d** Iteration 13 **e** Iteration 14 **f** Iteration 15 **g** Iteration 16 **h** Iteration 17 **i** Iteration 18 **j** Iteration 19 **k** Iteration 20 **l** Iteration 21 **m** Iteration 22 **n** Iteration 23

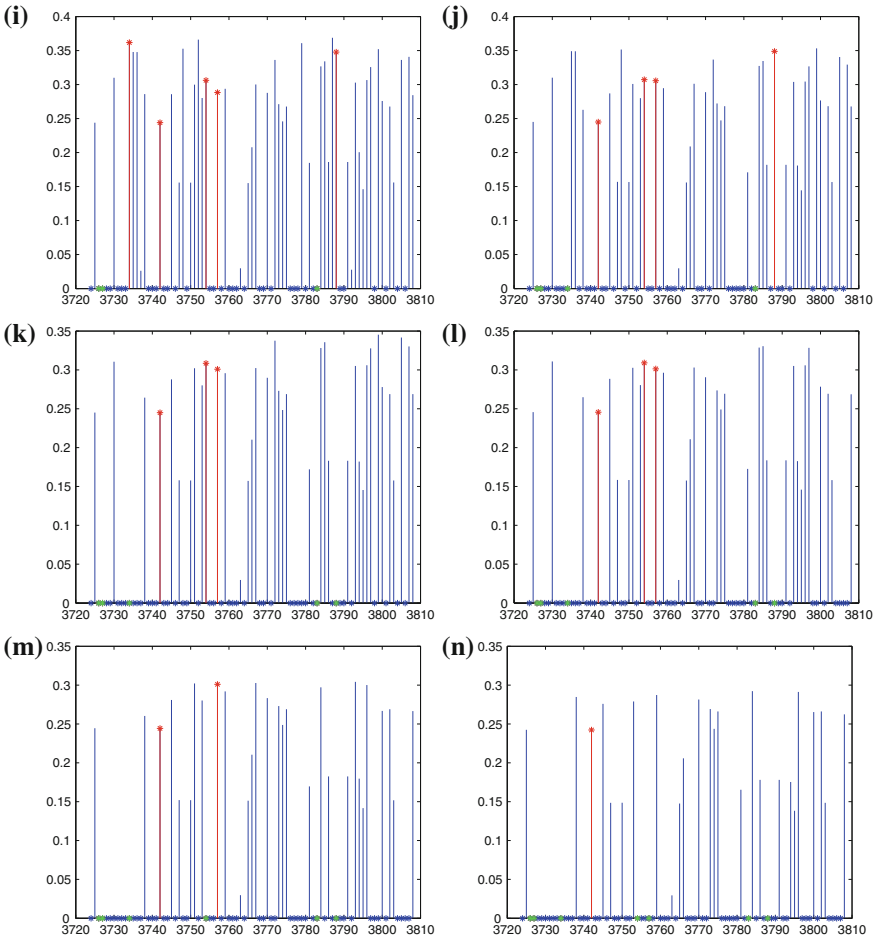

Fig. 4 (continued)

as in [18]. Also it is clear that the better the clustering of the data, the better are the suggestions of the system. The construction of the model is semi supervised in terms of the class injections, but unsupervised in terms of the representation of each class by a certain number of components basing on the MML criterion. Such framework may be built using a totally unsupervised data model using any clustering technique including the mixture models. The search process can be longer when a component contains a large subset of images, another perspective of research can be dedicated to find more appropriate techniques to discard "not interesting" images within a component. The strategy followed by [11] to discard clusters that are not interesting to the user is not appropriate, as a parent cluster that is discarded could contain sub clusters that are interesting to the user. When compared with the work in [11], we introduced the possibility of having multiple selections, and also the no preference

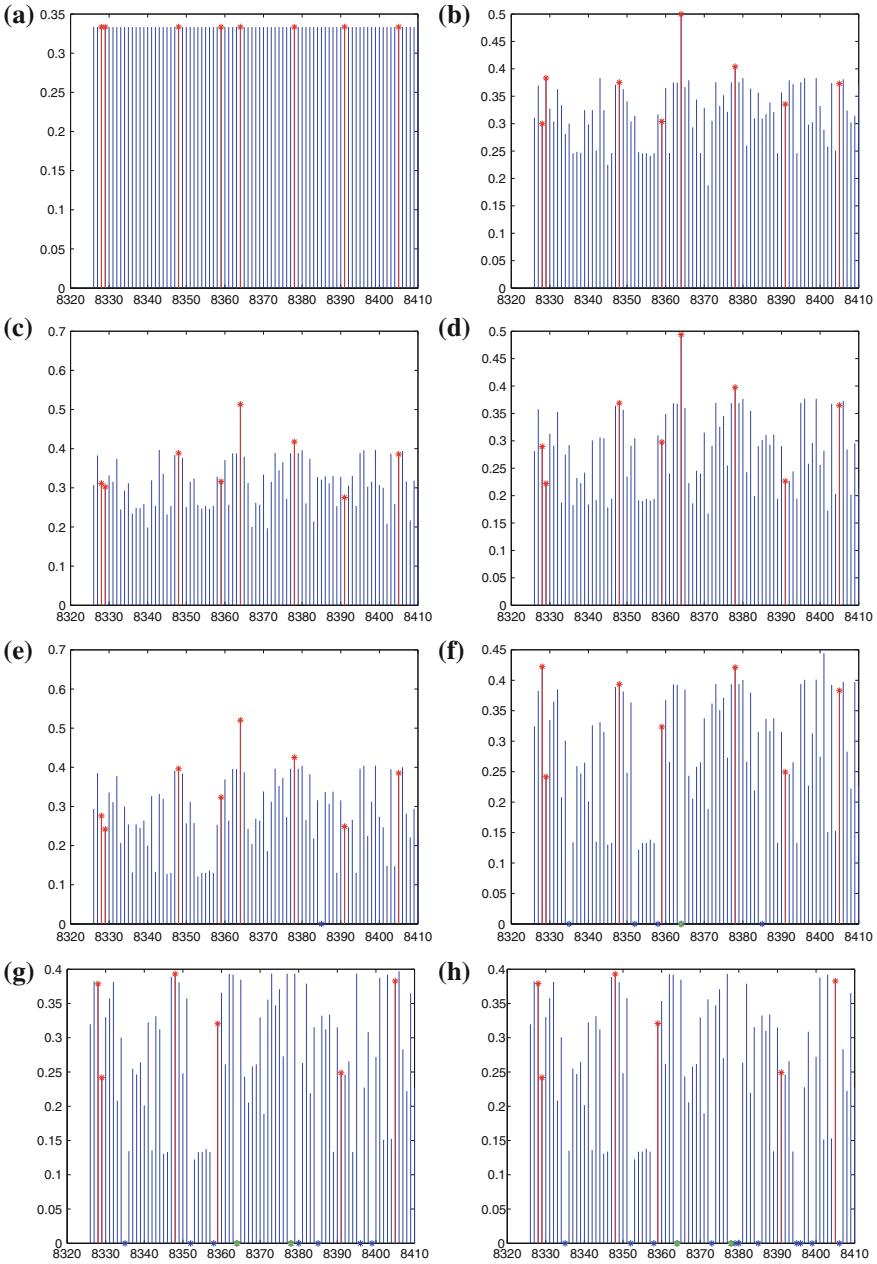

Fig. 5 Class "sunflowers": $p_t(X_k)_j$ evolution during different iterations **a** Iteration 1 **b** Iteration 9 **c** Iteration 11 **d** Iteration 13 **e** Iteration 14 **f** Iteration 15 **g** Iteration 16 **h** Iteration 17 **i** Iteration 18 **j** Iteration 19 **k** Iteration 20 **l** Iteration 21 **m** Iteration 22 **n** Iteration 23

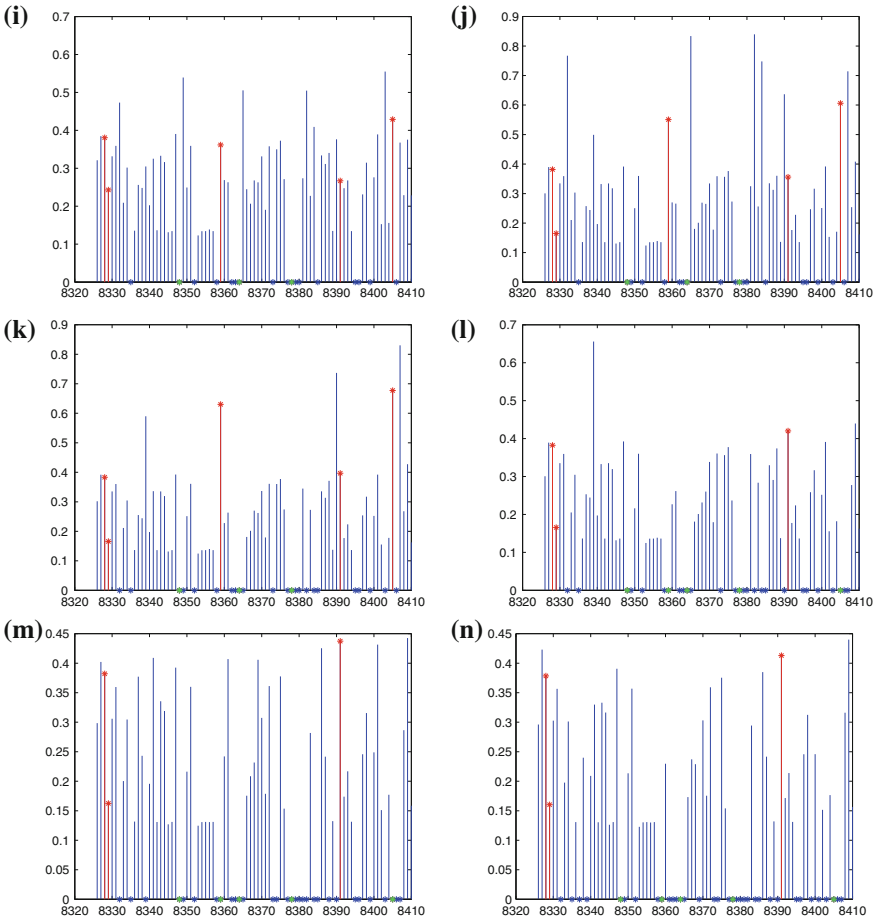

Fig. 5 (continued)

option. In many cases the user decides that many images are close to his/her target image and selects them all. Also, having a no preference selection enables the user not to mislead the system which may happen in the framework proposed by [11]. We have introduced the possibility of having many target classes within the same search process. Indeed, in many cases, the target class is composed of many sub classes that can be mapped to several target classes, and the user does not have to redo the search process in order to find them. For example, in the framework proposed by [11], it is not possible for the user to have a suggested display that contains a mix of pictures including "buddha" and "sunflower", as the system would only suggest one target class. The framework we have proposed is rather a generalization to what has been proposed to cover multiple target classes within a single framework. We also do not

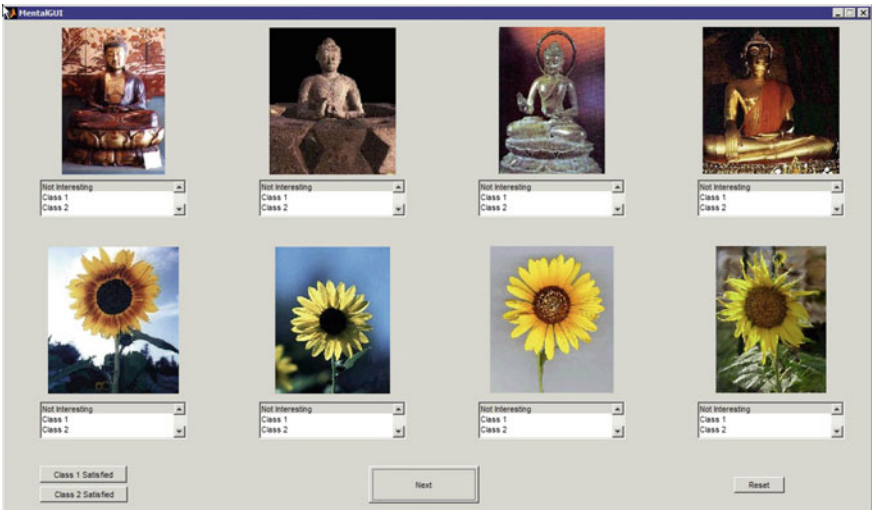

Fig. 6 Display 18

discard any cluster, but we rather zoom on a cluster to select images from it, and other clusters can reappear according to the choices of the user.

10 Conclusion

We have proposed a statistical framework whose purpose is to help a user to find target concepts within a set of data, without expressing specific query to the system but rather basing on a mental process. The system tries to distinguish the user needs basing on his/her selections. Such search processes are justified by the fact that, in many cases, users cannot express what they have in mind, or lack the needed vocabulary to describe a concept. We have mainly included the multi target classes search within one single search process, and also included the multi selection and no preference options compared to the work in [11]. We have also proposed a new data model basing on the Generalized Inverted Dirichlet mixture. A such model, can be used to construct different models of the data according to the needs of the system designer, and basing on the hierarchical model proposed in [1–3]. A better representation of the data in terms of modeling and clustering, can be performed by establishing a feature selection such as in [8], or by a better estimation of the parameters using the variational process such as in [9]. Further research perspectives could cover the construction of a link between two target classes like for example finding an image of buddha with a sunflower.

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Bdiri, T., Bouguila, N., Ziou, D.: A statistical framework for online learning using adjustable model selection criteria. Eng. Appl. Artif. Intell. (2014) (manuscript submitted for publication)
2. Bdiri, T., Bouguila, N., Ziou, D.: Object clustering and recognition using multi-finite mixtures for semantic classes and hierarchy modeling. Expert Syst. Appl. **41**(4, Part 1), 1218–1235 (2014)
3. Bdiri, T., Bouguila, N., Ziou, D.: Visual scenes categorization using a flexible hierarchical mixture model supporting users ontology. In: IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 262–267 (2013)
4. Bourouis, S., Mashrgy, M., Bouguila, N.: Bayesian learning of finite generalized inverted dirichlet mixtures: application to object classification and forgery detection. Expert Syst. Appl. **41**(5), 2329–2336 (2014)
5. Boutemedjet, S., Ziou, D.: Long-term relevance feedback and feature selection for adaptive content based image suggestion. Pattern Recogn. **43**(12), 3925–3937 (2010)
6. Cox, I., Miller, M., Minka, T., Papathomas, T., Yianilos, P.: The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. IEEE Trans. Image Process. **9**(1), 20–37 (2000)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 886–893. IEEE Computer Society (2005)
8. Fan, W., Bouguila, N., Ziou, D.: Unsupervised hybrid feature extraction selection for high-dimensional non-gaussian data clustering with variational inference. IEEE Trans. Knowl. Data Eng. **25**(7), 1670–1685 (2013)
9. Fan, W., Bouguila, N., Ziou, D.: Variational learning of finite dirichlet mixture models using component splitting. Neurocomputing **129**, 3–16 (2014)
10. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: Computer Vision and Pattern Recognition Workshop, CVPRW '04. pp. 178–178 (2004)
11. Ferecatu, M., Geman, D.: A statistical framework for image category search from a mental picture. IEEE Trans. Pattern Anal. Mach. Intell. **31**(6), 1087–1101 (2009)
12. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Qian, H., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: the qbic system. Computer **28**(9), 23–32 (1995)
13. Jia, L., Wang, J.: Real-time computerized annotation of pictures. IEEE Trans. Pattern. Anal. Mach. Intell. **30**(6), 985–1002 (2008)
14. Kaasinen, A., Yong-Ik, Y.: Service engagement model for mobile advertising based on user behavior. In: International Conference on Information Networking (ICOIN). pp. 131–134 (2013)
15. Kherfi, M., Ziou, D.: Relevance feedback for cbir: a new approach based on probabilistic feature weighting with positive and negative examples. IEEE Trans. Image Process. **15**(4), 1017–1030 (2006)
16. Kim, S., Qin, T., Liu, T., Yu, H.: Advertiser-centric approach to understand user click behavior in sponsored search. Inf. Sci. **276**, 242–254 (2014)
17. Lingappaiah, G.S.: On the generalised inverted dirichlet distribution. Demonstratio Math. **9**, 423–433 (1976)
18. Lokoc, J., Grosup, T., Cech, P., Skopal, T.: Towards efficient multimedia exploration using the metric space approach. In: 12th International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1–4 (2014)

19. Mashrgy, M., Bdiri, T., Bouguila, N.: Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted dirichlet mixture models. Knowl. Based Syst. **59**, 182–195 (2014)
20. Pan, J., Ren, Y., Wu, H., Zhu, M.: Query generation for semantic datasets. In: Proceedings of the Seventh International Conference on Knowledge Capture. pp. 113–116. K-CAP '13. ACM (2013)
21. Shahab Saquib, S., Jamshed, S., Rashid, A.: User feedback based evaluation of a product recommendation system using rank aggregation method. In: Advances in Intelligent Informatics, Advances in Intelligent Systems and Computing, vol. 320, pp. 349–358. Springer International Publishing (2015)
22. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. **22**(12), 1349–1380 (2000)
23. Suditu, N., Fleuret, F.: Heat: Iterative relevance feedback with one million images. In: IEEE International Conference on Computer Vision (ICCV), pp. 2118–2125 (2011)
24. Suditu, N., Fleuret, F.: Iterative relevance feedback with adaptive exploration/exploitation tradeoff. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. pp. 1323–1331. CIKM'12. ACM (2012)
25. Vasconcelos, N., Lippman, A.: A probabilistic architecture for content-based image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 216–221 (2000)
26. Yong, R., Huang, T., Ortega, M., Mehrotra, S.: Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Trans. Circuits Syst. Video Technol. **8**(5), 644–655 (1998)
27. Zhimin, Y., Xiangzhan, Y., Hongli, Z.: Commodity recommendation algorithm based on social network. Advances in Computer Science and Its Applications. Lecture Notes in Electrical Engineering, vol. 279, pp. 27–33. Springer, Berlin Heidelberg (2014)
28. Zhou, X., Huang, T.: Relevance feedback in image retrieval: a comprehensive review. Multimedia Syst. **8**(6), 536–544 (2003)
29. Ziou, D., Hamri, T., Boutemedjet, S.: A hybrid probabilistic framework for content-based image retrieval with feature weighting. Pattern Recogn. **42**(7), 1511–1519 (2009)

Variational Learning of Finite Inverted Dirichlet Mixture Models and Applications

Parisa Tirdad, Nizar Bouguila and Djemel Ziou

Abstract Statistical modeling provides a useful and well grounded framework to conduct inference from data. This has given rise to the development of varied rich suite of models and techniques. In particular, finite mixture models have received a lot of attention by offering a formal approach to unsupervised learning which allows to discover the latent structure expressed in observed data. In this chapter, we propose a mixture model based on the inverted Dirichlet mixture which provides a natural way of clustering positive data. An EM-style algorithm is developed based upen variational inference for learning the parameters of the mixture model. The proposed statistical framework is applied to the challenging tasks of natural scene categorization and human activity classification.

Keywords Inverted dirichlet · Mixture models · Variational learning · Scene categorization · Human activity recognition

1 Introduction

Advances in technology has allowed to generate and store large amounts of multimodal data (text, image, video, audio). A crucial problem is the statistical modeling and analysis of these data. This is evidenced by many information retrieval systems (see, for instance, [1, 2]), and various data mining and machine learning techniques. Clustering, in particular, has been the topic of extensive research in the

P. Tirdad · N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

P. Tirdad
e-mail: parisa.tirdad@gmail.com

D. Ziou
DI, Faculté des Sciences, Université de Sherbrooke, Sherbrooke, QC, Canada
e-mail: djemel.ziou@usherbrooke.ca

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_6

past and several parametric and nonparametric approaches have been proposed (see, for instance, [3–9]). Among these approaches finite mixtures have received a lot of attention. The most popular mixture model is the Gaussian mixture [10–12] which has been deployed in several applications [13–15]. However, this choice is not appropriate when the data partitions are not Gaussians as shown in several previous works [16–21]. Indeed, it is important to take the form of the data and the kind of latent structure it expresses into account. This is exactly the case of positive data for which the inverted Dirichlet, that we shall consider in this paper, has been shown to be an efficient alternative to the Gaussian [19–21].

Two challenging problems when dealing with finite mixtures are the determination of the number of the mixture components and the estimation of the mixture's parameters. Concerning parameters estimation, two families of approaches could be considered namely the frequentist and the Bayesian techniques. The maximum likelihood (ML), while the most popular among frequentist estimation techniques, to mixture learning has several shortcomings for its application since it can easily get caught in saddle points or local maxima and it depends on the initially set parameters. It was implemented in [20], via an expectation-maximization algorithm [22], in the case of the inverted Dirichlet mixture. To address these drawbacks, Bayesian framework could be adopted [23]. Bayesian learning has several interesting properties. For instance, it allows to incorporate prior knowledge in a natural way, it permits the manipulation of uncertainty consistently, and it does not suffer from over-fitting problems. However, fully Bayesian learning is generally computationally intractable which has forced researchers in the past to adopt approximation techniques such as Laplace approximation and Markov Chain Monte Carlo (MCMC) sampling. In particular MCMC-based sampling approaches have received a lot of attention, yet they suffer from significant computational complexity. Thus, variational learning has been proposed, as an efficient deterministic approximation to fully Bayesian learning, to overcome the problems related to MCMC sampling and have been widely adopted [24]. Variational learning has made it possible to fit large class of learning models and then to explore real-world complexity of data [25, 26]. It can be viewed as an approximation to the exact pure Bayesian learning where the true posterior is approximated with a simpler distribution. The main goal of this paper is to propose a variational approach that can simultaneously estimate the model's parameters and automatically select the appropriate number of mixture components. The proposed approach is different from classic techniques most widely used for model selection, which are based on selecting the best mixture models from a set of candidates with different number of components which is time consuming since it requires the estimation of the parameters of multiple mixture models. In addition to showing the validity of the proposed variational approach in parameter estimation and model selection on different synthetic datasets, we have shown the usefulness of our method in challenging real world applications. First application is natural scene categorization which plays an important role in several tasks such as content-based image retrieval. Human activity classification is the second task that has attracted lots of attention for its important applications in security systems and surveillance

for public environments. Moreover, we compared the performance of our model with Gaussian mixture model.

In Sect. 2, a brief overview about inverted Dirichlet mixtures is given. The specific variational learning approach that we have developed is discussed in details in Sect. 3. The complete learning algorithm is also presented in Sect. 3, with extensive experimental results shown and described in Sect. 4. The last section summarizes this paper and presents the conclusions.

2 Finite Inverted Dirichlet Mixture Model

The main reason for adopting inverted Dirichlet distribution as the standard distribution for mixture model is that, inverted Dirichlet can generate models based on the nature of the data and as shown in Fig. 1, unlike Gaussian distribution, it is considerably flexible and can perform in both symmetric and asymmetric modes. The inverted Dirichlet distribution has many interesting properties and has applications in various fields [27–29].

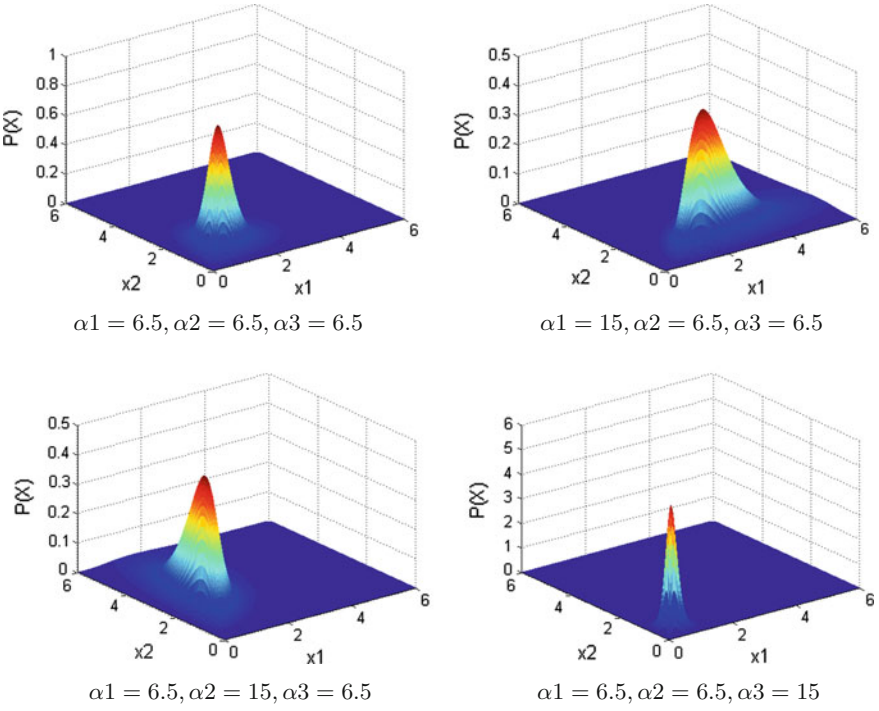

Fig. 1 Bivariate inverted Dirichlet distributions, in symmetric and asymmetric modes

Assume that a D-dimensional positive vector $\mathbf{X}_i = (X_{i1}, \ldots, X_{iD})$ is sampled from a finite inverted Dirichlet mixture model with M components, then we have:

$$p(\mathbf{X}_i|\boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{j=1}^{M} \pi_j \mathcal{ID}(\mathbf{X}_i|\boldsymbol{\alpha}_j) \tag{1}$$

where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M)$ and $\boldsymbol{\pi} = (\pi_1, ..., \pi_M)$ denotes the mixing coefficients with the constraints that they are positive and sum to one. $\mathcal{ID}(\mathbf{X}_i|\boldsymbol{\alpha}_j)$ represents the jth inverted Dirichlet distribution with parameter $\boldsymbol{\alpha}_j$ and is defined in [27] as

$$\mathcal{ID}(\mathbf{X}_i|\boldsymbol{\alpha}_j) = \frac{\Gamma(\sum_{l=1}^{D+1} \alpha_{jl})}{\Pi_{l=1}^{D+1} \Gamma(\alpha_{jl})} \Pi_{l=1}^{D} X_{il}^{\alpha_{jl}-1} (1 + \sum_{l=1}^{D} X_{il})^{-\sum_{l=1}^{D+1} \alpha_{jl}} \tag{2}$$

where $0 < X_{il} < \infty$ for $l = 1, \ldots, D$. In addition, $\boldsymbol{\alpha}_j = (\alpha_{j1}, ..., \alpha_{jD})$ such that $\alpha_{j1} > 0$ for $l = 1, \ldots, D+1$. The mean, variance and covariance of the inverted Dirichlet distribution are given by

$$E(X_l) = \frac{\alpha_l}{(\alpha_{D+1} - 1)} \tag{3}$$

$$var(X_l) = \frac{\alpha_l(\alpha_j + \alpha_{D+1} - 1)}{(\alpha_{D+1} - 1)^2(\alpha_{D+1} - 2)} \tag{4}$$

$$cov(X_a, X_b) = \frac{\alpha_a \alpha_b}{(\alpha_{D+1} - 1)^2(\alpha_{D+1} - 2)} \tag{5}$$

Next, we introduce an M-dimensional binary random vector $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iM})$ for each observed vector $\mathbf{X}_i$, such that $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^{M} Z_{ij} = 1$, and $Z_{ij} = 1$ if $\mathbf{X}_i$ belongs to component j and 0, otherwise. Notice that, $\mathcal{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$ are called the *membership vectors* of the mixture model and are also considered as the latent variables since they are actually hidden variables that do not appear explicitly in the model. Furthermore, the conditional distribution of $\mathcal{Z}$ given the mixing coefficients $\boldsymbol{\pi}$ is defined as

$$p(\mathcal{Z}|\boldsymbol{\pi}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \pi_j^{Z_{ij}} \tag{6}$$

Then, the likelihood function with latent variables, which is indeed the conditional distribution of data set $\mathcal{X}$ given the class labels $\mathcal{Z}$ can be written as

$$p(\mathcal{X}|\mathcal{Z}, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \mathcal{ID}(\mathbf{X}_i|\boldsymbol{\alpha}_j)^{Z_{ij}} \tag{7}$$

Moreover, we assume that the parameters of the inverted Dirichlet are statistically independent and for each parameter α_{jl}, the Gamma distribution $\mathcal{G}$ is adopted to approximate the conjugate prior:

$$p(\alpha_{jl}) = \mathcal{G}(\alpha_{jl}|u_{jl}, v_{jl}) = \frac{v_{jl}^{u_{jl}}}{\Gamma(u_{jl})}\alpha_{jl}^{u_{jl}^{-1}} e^{-v_{jl}\alpha_{jl}} \tag{8}$$

where u_{jl} and v_{jl} are positive hyperparameters. Thus, the joint distribution of all the random variables, conditioned on the mixing coefficients can be written as

$$\begin{aligned} p(\mathcal{X}, \mathcal{Z}, \boldsymbol{\alpha}|\boldsymbol{\pi}) &= p(\mathcal{X}|\mathcal{Z}, \boldsymbol{\alpha})p(\mathcal{Z}|\boldsymbol{\pi})p(\boldsymbol{\alpha}) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\pi_j \frac{\Gamma(\sum_{l=1}^{D+1}\alpha_{jl})}{\prod_{l=1}^{D+1}\Gamma(\alpha_{jl})}\prod_{l=1}^{D} X_{il}^{\alpha_{jl}-1}\right. \\ &\quad \left.\times (1+\sum_{l=1}^{D} X_{il})^{-\sum_{l=1}^{D+1}\alpha_{jl}}\right]^{Z_{ij}} \\ &\quad \times \prod_{j=1}^{M}\prod_{l=1}^{D+1}\frac{v_{jl}^{u_{jl}}}{\Gamma(u_{jl})}\alpha_{jl}^{u_{jl}-1} e^{-v_{jl}\alpha_{jl}} \end{aligned} \tag{9}$$

A directed representation of this model is illustrated in Fig. 2.

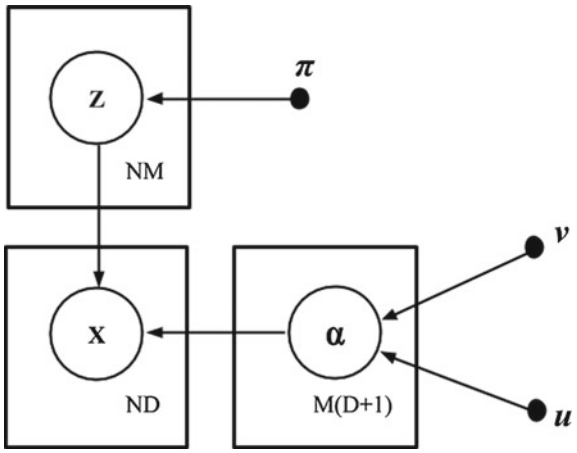

Fig. 2 Graphical model representation of the finite inverted Dirichlet mixture. Symbols in circles denote random variables; otherwise, they denote model parameters. Plates indicate repetition (with the number of repetitions in the lower right), and arcs describe conditional dependencies between variables

3 Variational Learning

Variational inference is a deterministic approximation scheme, which is used to formulate the computation of a marginal or conditional probability in terms of an optimization problem. In this section, following the methodology proposed in [30], we develop a variational framework for learning the finite inverted Dirichlet mixture model. The main idea of variational learning is to approximate the intractable joint distribution with a simpler distribution by lower bounding the log evidence using Jensen's inequality. To simplify the notation without loss of generality, we define $\Theta = \{\mathcal{Z}, \boldsymbol{\alpha}\}$. The main idea in variational learning is to find an approximation $Q(\Theta)$, which approximates the true posterior distribution $p(\Theta|\mathcal{X}, \boldsymbol{\pi})$. The logarithm of the model evidence $p(\mathcal{X}|\boldsymbol{\pi})$ can be decomposed as

$$\ln p(\mathcal{X}|\boldsymbol{\pi}) = \mathcal{L}(q) - \underbrace{\int Q(\Theta) \ln \left[\frac{p(\Theta|\mathcal{X}, \boldsymbol{\pi})}{Q(\Theta)}\right] d\Theta}_{-KL(Q\|p)} \tag{10}$$

where $KL(Q \parallel p)$ is the Kullback-Leibler (KL) divergence between $Q(\Theta)$ and the true posterior distribution $p(\Theta|\mathcal{X}, \boldsymbol{\pi})$. $\mathcal{L}(q)$ is the variational lower bound of $\ln p(\mathcal{X})$ and is defined by

$$\mathcal{L}(q) = \int Q(\Theta) \ln \left[\frac{p(\mathcal{X}, \Theta|\boldsymbol{\pi})}{Q(\Theta)}\right] d\Theta \tag{11}$$

In our work, a mean field approximation [31, 32] is adopted for the variational inference. Hence, $Q(\Theta)$ can be factorized into disjoint tractable distributions as follows:

$$Q(\Theta) = Q(\mathcal{Z})Q(\boldsymbol{\alpha}) \tag{12}$$

In order to maximize the lower bound $\mathcal{L}(q)$, we need to make a variational optimization of $\mathcal{L}(q)$ with respect to each of the factors in turn. For a specific factor $Q_s(\Theta_s)$ the general variational solution is given by

$$Q_s(\Theta_s) = \frac{\exp \langle \ln p(\mathcal{X}, \Theta|\boldsymbol{\pi}) \rangle_{\neq s}}{\int \exp \langle \ln p(\mathcal{X}, \Theta|\boldsymbol{\pi}) \rangle_{\neq s} d\Theta} \tag{13}$$

where $\langle . \rangle_{\neq s}$ denotes an expectation with respect to all factor distributions, except for s. We can obtain the following variational solutions for the finite inverted Dirichlet mixture model:

$$Q(\mathcal{Z}) = \prod_{i=1}^{N} \prod_{j=1}^{M} {r_{ij}}^{Z_{ij}} \tag{14}$$

$$Q(\boldsymbol{\alpha}) = \prod_{j=1}^{M} \prod_{l=1}^{D+1} \mathcal{G}(\alpha_{jl} | u_{jl}^{*}, v_{jl}^{*}) \tag{15}$$

where we have defined

$$r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^{M} \rho_{ij}} \tag{16}$$

$$\begin{aligned} \rho_{ij} = \exp \Bigg\{ & \ln \pi_j + \tilde{\mathcal{R}}_j + \sum_{l=1}^{D} \left(\bar{\alpha}_{jl} - 1\right) \ln X_{il} \\ & - \Big(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}\Big) \ln\Big(1 + \sum_{l=1}^{D} X_{il}\Big) \Bigg\} \end{aligned} \tag{17}$$

$$\begin{aligned} \widetilde{\mathcal{R}}_j = & \ln \frac{\Gamma\left(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}\right)}{\prod_{l=1}^{D+1} \Gamma\left(\bar{\alpha}_{jl}\right)} \\ & + \sum_{l=1}^{D+1} \bar{\alpha}_{jl} \left[\Psi\left(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}\right) - \Psi\left(\bar{\alpha}_{jl}\right) \right] \left[\left\langle \ln \alpha_{jl} \right\rangle - \ln \bar{\alpha}_{jl} \right] \\ & + \frac{1}{2} \sum_{l=1}^{D+1} \bar{\alpha}_{jl}^{2} \left[\Psi'\left(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}\right) - \Psi'\left(\bar{\alpha}_{jl}\right) \right] \left\langle \left(\ln \alpha_{jl} - \ln \bar{\alpha}_{jl} \right)^{2} \right\rangle \\ & + \frac{1}{2} \sum_{a=1}^{D+1} \sum_{\substack{b=1 \\ (b \neq a)}}^{D+1} \bar{\alpha}_{ja} \bar{\alpha}_{jb} \left[\Psi'\left(\sum_{l=1}^{D+1} \bar{\alpha}_{jl} \left(\left\langle \ln \alpha_{ja} \right\rangle - \ln \bar{\alpha}_{ja} \right)\right) \right. \\ & \left. \times \left(\left\langle \ln \alpha_{jb} \right\rangle - \ln \bar{\alpha}_{jb} \right) \right] \end{aligned} \tag{18}$$

$$\begin{aligned} u_{jl}^{*} = u_{jl} + \sum_{i=1}^{N} \langle Z_{ij} \rangle \bar{\alpha}_{jl} \Bigg[& \Psi\Big(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}\Big) - \Psi(\bar{\alpha}_{jl}) \\ & + \sum_{k \neq l}^{D+1} \bar{\alpha}_{k} \Psi'\Big(\sum_{l=1}^{D+1} \bar{\alpha}_{l}\Big) \left(\langle \ln \alpha_{k} \rangle - \ln \bar{\alpha}_{k} \right) \Bigg] \end{aligned} \tag{19}$$

$$v_{jl}^{*} = v_{jl} - \sum_{i=1}^{N} \left\langle Z_{ij} \right\rangle \left[\ln X_{il} - \ln\left(1 + \sum_{l=1}^{D} X_{il}\right) \right] \tag{20}$$

where Ψ (.) is diagamma function. The expected values in the above formulas are

$$\langle Z_{ij} \rangle = r_{ij} \tag{21}$$

$$\bar{\alpha}_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}}{v_{jl}} \tag{22}$$

$$\langle \ln \alpha_{jl} \rangle = \Psi \left(u_{jl} \right) - \ln v_{jl} \tag{23}$$

Note that, $\tilde{\mathcal{R}}_j$ is the approximate lower bound of $\mathcal{R}_j$, where $\mathcal{R}_j$ is defined as $\mathcal{R}_j = \left\langle \ln \frac{\Gamma\left(\sum_{l=1}^{D+1} \alpha_{jl}\right)}{\prod_{l=1}^{D+1} \Gamma(\alpha_{jl})} \right\rangle$. Since a closed form expression cannot be found for $\mathcal{R}_j$, the standard variational inference can not be applied directly. Therefore, we applied the second-order Taylor series expansion to find a lower bound approximation $\tilde{\mathcal{R}}_j$ for the variational inference.

In our case, the mixing coefficients $\boldsymbol{\pi}$ are treated as parameters, and point estimations of their values are evaluated by maximizing the variational likelihood bound $\mathcal{L}(Q)$. Setting the derivative of this lower bound with respect to $\boldsymbol{\pi}$ to zero gives:

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} r_{ij} \tag{24}$$

It is noteworthy that components that provide insufficient contribution to explain the data would have their mixing coefficients driven to zero during the variational optimization. Thus, by starting with a relatively large initial value of M and then remove the redundant components after convergence, we can obtain the correct number of components. Variational learning, is able to trace the convergence systematically by monitoring the variational lower bound during the re-estimation step [33]. Indeed, at each step of the iterative re-estimation procedure, the value of this bound should never decrease. Specifically, the bound $\mathcal{L}(Q)$ is evaluated at each iteration and terminate optimization if the amount of increase from one iteration to the next is less than a threshold. For the variational inverted Dirichlet mixture model, the lower bound in (11) is evaluated as

$$\begin{aligned} \mathcal{L}(Q) &= \sum_{\mathcal{Z}} \int Q(\mathcal{Z}, \boldsymbol{\alpha}) \ln \left\{ \frac{p(\mathcal{X}, \mathcal{Z}, \boldsymbol{\alpha} | \boldsymbol{\pi})}{Q(\mathcal{Z}, \boldsymbol{\alpha})} \right\} d\boldsymbol{\alpha} \\ &= \langle \ln p(\mathcal{X} | \mathcal{Z}, \boldsymbol{\alpha}) \rangle + \langle \ln p(\mathcal{Z} | \boldsymbol{\pi}) \rangle + \langle \ln p(\boldsymbol{\alpha}) \rangle \\ &\quad - \langle \ln Q(\mathcal{Z}) \rangle - \langle \ln Q(\boldsymbol{\alpha}) \rangle \end{aligned} \tag{25}$$

The variational inference for finite inverted Dirichlet mixture model can be performed via an EM-like algorithm and is summarized in Algorithm 1.

Algorithm 1 Variational learning of inverted Dirichlet mixture model

1: Set the initial number of components M.
2: Initialize the values of the hyper-parameters u_{jl} and v_{jl}.
3: Initialize the value of r_{ij} by K-means algorithm.
4: **repeat**
5: The variational E-step: update the variational solutions for $Q(\mathcal{Z})$ (14) and $Q(\boldsymbol{\alpha})$ (15).
6: The variational M-step: maximize the lower bound $\mathcal{L}(Q)$ with respect to the current value of $\boldsymbol{\pi}$ (24).
7: **until** Convergence criterion is reached.
8: Detect the optimal number of components M by eliminating the components with small mixing coefficients close to 0.

4 Experimental Results

This section shows the experimental results of applying the proposed variational inverted Dirichlet mixture model (varIDM) on synthetic data and its applications in natural scene categorization and human activity classification. In all the experiments, the number of components M is initialized to 20 with equal mixing coefficients. The initial values of hyperparameters u_{jl} and v_{jl} were 1 and 0.01, respectively. Our experiments are performed using MATLAB on a Windows platform machine.

5 Synthetic Data

To show the validity of the proposed approach in parameter and model selection, it is applied on six, two-dimensional synthetic datasets. Please note that, $D = 2$ is chosen for ease of representation. The number of the components is set to 20 as a start point. Table 1 shows the real and estimated parameters, resulted from VarIDM. For estimating the number of components a threshold of ($T = 10^{-4}$) is applied to remove the redundant components that have mixing coefficients close to zero. As it is shown in Table 1, for all the synthesized datasets, the proposed approach could successfully estimate the number of components with a good accuracy. To prove that finite inverted Dirichlet mixtures are considerably flexible and can perform in both symmetric and asymmetric modes, some examples with symmetric and asymmetric shapes are demonstrated in Fig. 3. Figure 4 represents the variational lower likelihood bound. As shown in this figure, the likelihood bound increases very fast when one of the mixing coefficients is close to zero, which indicates that the estimated component should be eliminated. In each diagram (Fig. 4), the value of the likelihood bound is maximum at the point in which, the true number of components is estimated. Therefore, the variational likelihood bound can be used as a model selection criterion. In this case, there is no need to eliminate the redundant components by applying a threshold.

Table 1 VarIDM real and estimated parameters on different synthetic datasets

	n_j	j	α_{j1}	α_{j2}	α_{j3}	π_j	$\bar{\alpha}_{j1}$	$\bar{\alpha}_{j2}$	$\bar{\alpha}_{j3}$	$\bar{\pi}_j$
Dataset 1 (N = 400)	200	1	20	70	4	0.50	20.47	70.70	4.07	0.5010
	200	2	40	50	5	0.50	36.42	45.16	4.62	0.4990
Dataset 2 (N = 400)	133	1	10	40	4	0.33	9.54	38.51	4.22	0.3333
	133	2	20	30	5	0.33	22.16	32.07	5.34	0.3458
	133	3	30	20	5	0.33	29.27	18.88	4.79	0.3209
Dataset 3 (N = 600)	200	1	10	40	4	0.33	9.62	38.23	3.72	0.3359
	200	2	20	30	5	0.33	18.93	28.67	4.39	0.3080
	200	3	30	20	5	0.33	29.71	19.89	5.03	0.3561
Dataset 4 (N = 600)	150	1	10	40	4	0.25	10.87	42.12	4.30	0.2446
	150	2	20	30	5	0.25	18.67	27.95	4.44	0.2710
	150	3	30	20	5	0.25	33.71	20.64	5.24	0.2533
	150	4	40	10	4	0.25	35.96	8.81	3.54	0.2311
Dataset 5 (N = 800)	200	1	10	40	4	0.25	10.43	41.66	4.01	0.2493
	200	2	20	30	5	0.25	19.14	28.42	4.81	0.2630
	200	3	30	20	5	0.25	28.42	18.66	4.49	0.2359
	200	4	40	10	4	0.25	38.04	9.30	3.86	0.2517
Dataset 6 (N = 1000)	200	1	10	40	4	0.20	8.77	39.30	3.60	0.1943
	200	2	20	30	5	0.20	17.70	29.46	4.87	0.2020
	200	3	5	60	2	0.30	4.54	56.60	2.03	0.2892
	200	4	30	20	5	0.20	28.52	19.71	4.89	0.2155
	200	5	40	10	4	0.10	36.74	9.47	3.54	0.0990

In this Table, N denotes the total number of elements, n_j shows the number of elements in cluster j. $\alpha_{j1}, \alpha_{j2}, \alpha_{j3}$, and π_j denote the real parameters. $\bar{\alpha}_{j1}, \bar{\alpha}_{j2}, \bar{\alpha}_{j3}$, and $\bar{\pi}_j$ are the estimated parameters by variational inference

6 Natural Scene Categorization

Scene categorization is playing an important role in understanding the world through images. Human beings' brain is able to perceive complex natural scenes, understand their contents, and categorize them rapidly with little or no attention [34]. However, in machine vision, scene classification is a very challenging task due to variability in texture and color, ambiguity, wide range of illumination, scale and location of the objects in the scene [35], all of which, make it difficult to classify scenes belonging to the same class, and avoiding confusing scenes from different classes. In this section, the proposed approach is tested on scene categorization using bag-of-visual-words representation [36, 37]. Every image contains multiple salient patches (usually around the corners and edges) called keypoints, which contain rich local information about that image. Using k-means clustering these keypoints can be grouped into different clusters each of which is considered a "visual-word", and the group of visual-words are called visual-word vocabulary. With this definition, an image is represented as a "bag of visual words" [35, 36].

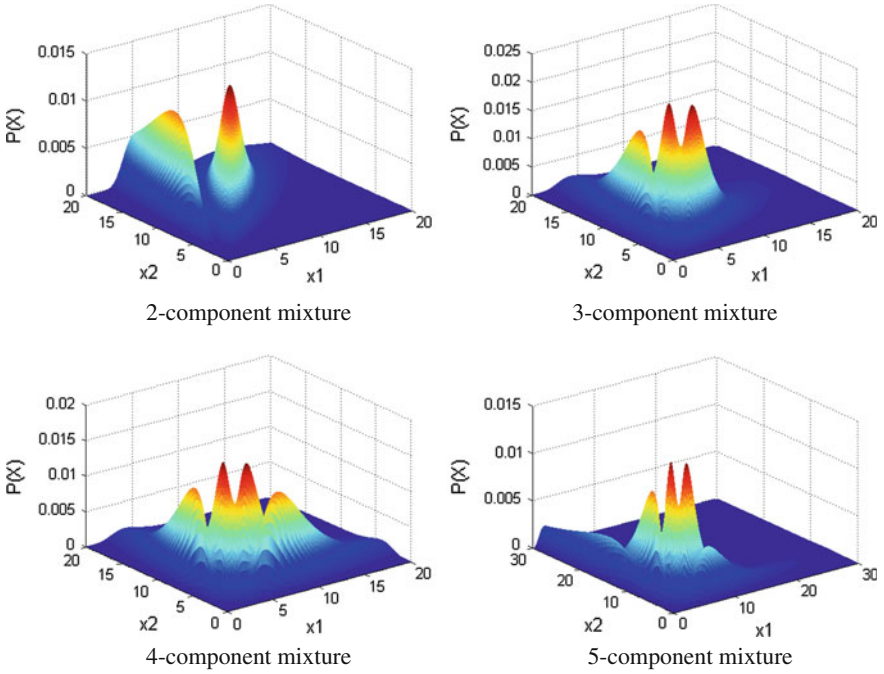

Fig. 3 Two-dimensional inverted Dirichlet mixtures

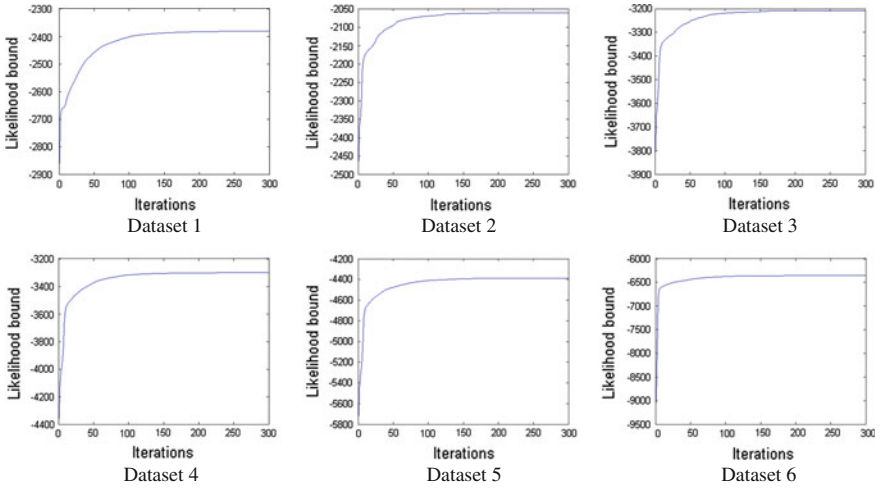

Fig. 4 Variational lower likelihood bound in each iteration for synthesized datasets

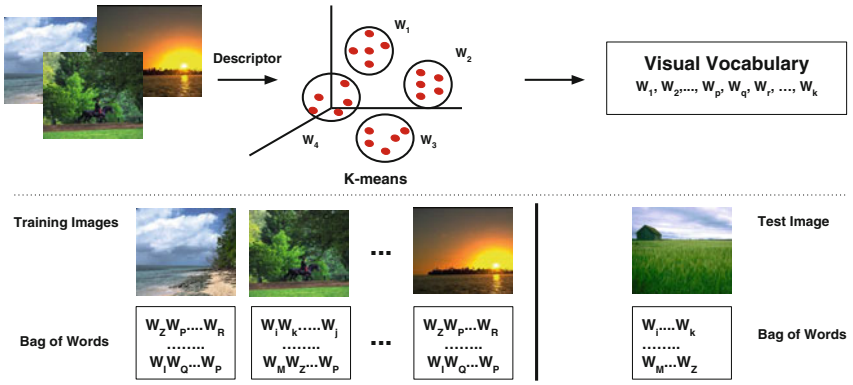

Fig. 5 Overview of visual vocabulary formation. (edited from [35])

Before extracting the keypoints, a 5×5 window Gaussian filter with sigma = 0.5 is applied on the images to reduce the effect of noise on the extracted keypoints. The preprocessed images were fed into scale invariant feature transform (SIFT) descriptor [38] and the extracted keypoints were quantized through K-Means clustering to form our visual words. Having the visual vocabulary, each image can be represented as a d-dimensional vector containing the frequency of each visual word in that image. Figure 5 demonstrates this process.

For the evaluations, MIT natural scene dataset [39] is used. This dataset contains eight categories of complex scenes namely, highway (260 images), inside city (308 images), tall buildings (365 images), street (292 images), forest (328), coast (360 images), mountains (374 images), open country (410 images). The dataset is collected from COREL images and personal photographs. Based on color and texture features, the classes are divided into indoor (highway, inside city, tall buildings, street) and outdoor (forest, coast, mountains, open country) categories. Figure 6 shows some examples from each class.

The performance of VarIDM and GMM are visualized in confusion matrices shown in Tables 2 and 3 for indoor and outdoor categories, respectively. In each confusion matrix:

- Each column represents the number of samples in a predicted class, and each row illustrates the samples in an actual class. In other words the value of entry $ConfMat(i, j)$ shows the number of instances which belong to class i but are classified as category j.
- The diagonal entries $ConfMat(i, j)_{i=j}$, represent the number of correctly classified samples.
- The off diagonal entries $ConfMat(i, j)_{i \neq j}$, show the system's false positives (FP) and false negatives (FN).

The same dataset was fed to Gaussian mixture model (GMM) and the results were computed. Tables 4 and 5 illustrate the GMM's confusion matrices for indoor and out-

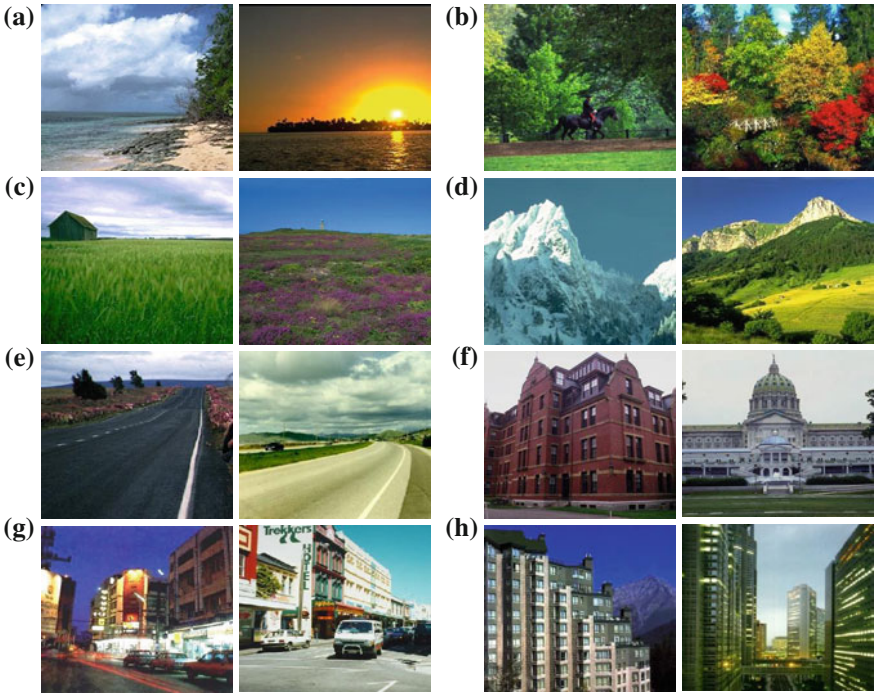

Fig. 6 Sample frames of MIT natural scene dataset. **a** Coast. **b** Forest. **c** Open country. **d** Mountain. **e** Highway. **f** Inside city. **g** Street. **h** Tall building

Table 2 VarIDM confusion matrix of indoor natural scene

	Highway	Street	Inside city	Tall building
Highway	**87**	30	6	7
Street	24	**107**	9	6
Inside city	3	7	**133**	11
Tall building	4	19	15	**145**

Table 3 VarIDM confusion matrix of outdoor natural scene

	Coast	Open country	Forest	Mountains
Coast	**155**	15	8	2
Open country	28	**143**	11	23
Forest	14	9	**104**	37
Mountains	3	17	32	**135**

Table 4 GMM confusion matrix of indoor natural scene

	Highway	Street	Inside city	Tall building
Highway	**70**	43	7	10
Street	30	**94**	14	8
Inside city	6	14	**108**	26
Tall building	5	37	16	**125**

Table 5 GMM confusion matrix of outdoor natural scene

	Coast	Open country	Forest	Mountains
Coast	**136**	27	11	6
Open country	39	**119**	17	30
Forest	12	19	**84**	49
Mountains	5	23	44	**115**

Table 6 Average accuracy of VarIDM and GMM on indoor and outdoor natural scene dataset

	VarIDM	GMM
Outdoor	72.96	61.68
Indoor	76.99	64.76

door datasets respectively. The overall accuracies for VarIDM and GMM are shown in Table 6. Running student's t-test on our results shows that VarIDM outperforms GMM at the significance level of 0.05, and p values were 0.0038 and 0.0001 for indoor and outdoor datasets, respectively.

7 Human Activity Classification

Automatic human activity classification has attracted lots of attention for its important applications in surveillance for public environments such as banks and airports and subway stations, or security systems in industry and commerce for intruder detection, real-time monitoring of patients, children or elderly people, and human-computer interaction [40, 41]. Variation in environment, caused by moving background, scene surroundings, camera motion, illumination changes, people varying in expression, posture, motion and clothing and actions, make human activity classification a challenging problem [42]. This section demonstrates our experimental results on Weizmann dataset [43] which contains 93 video sequences showing nine different people, each performing ten actions such as run, walk, skip, jumping-jack (jack), jump-forward-on-two-legs (jump), jump-in-place-on-two-legs (pjump), gallop-sideways (side), wave-two-hands (wave2), wave-one-hand (wave1) and bend. The video res-

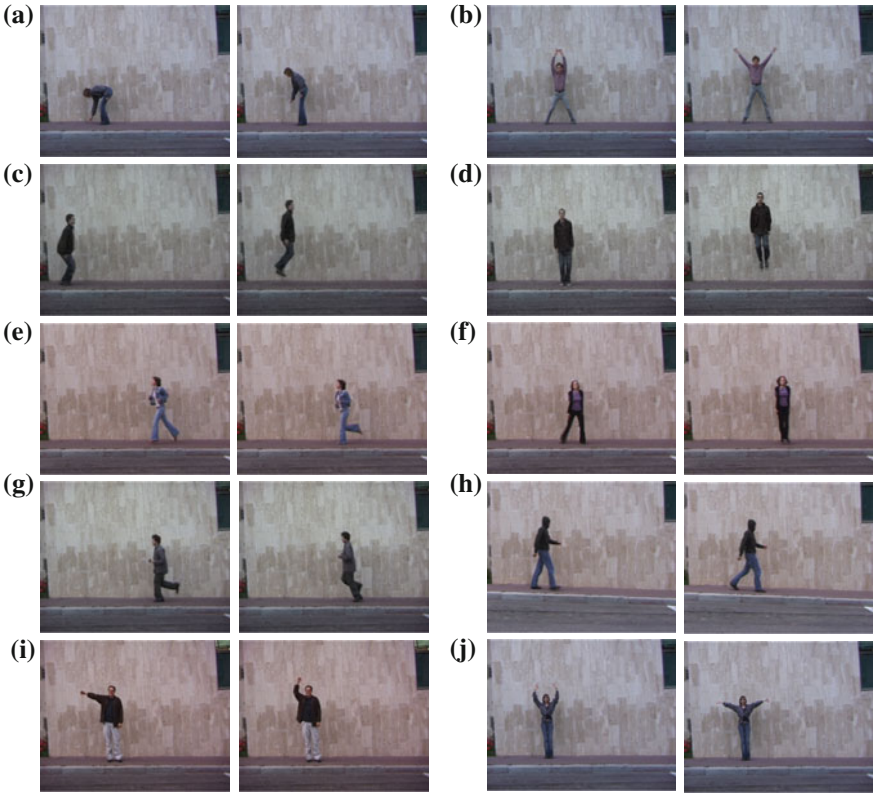

Fig. 7 Sample frames of Weizmann dataset. **a** Bend. **b** Jack. **c** Jump. **d** Pjump. **e** Run. **f** Side. **g** Skip. **h** Walk. **i** Wave1. **j** Wave2

olution is 180×144. Samples of each class are shown in Fig. 7. In the experiments, two different types of features are considered for video categorization. First group are local spatio-temporal features. Laptev et al. [44] detector is used to detect the space-time interest features in each video sequence. Second group are optical flow features [45], after computing the optical flow matrix on subsequent frames, a threshold ($T = 0.8$) is considered to extract the strong optical flow responses. A mask of size 5×5 is defined around the positions with the strong optical flow values to form the total feature set. These features are fed into the minimum Redundancy Maximum Relevance (mRMR) feature selection method [46] in order to choose the most discriminative features. Using K-means algorithm, a bag of visual words is constructed, and each video is represented as a frequency histogram of the visual words. Finally, the vector of frequencies is passed to VarIDM for classification. Tables 7 and 8 illustrate VarIDM and GMM confusion matrices for Weizmann dataset, respectively. We fed the same feature set to Gaussian mixture model (GMM) and computed the results. The overall accuracies for VarIDM and GMM are 87.49 and 81.3 respectively. As

Table 7 VarIDM confusion matrix on Weizmann action dataset

	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	**320**	0	0	0	0	0	0	0	0	0
Jack	0	**365**	0	0	0	0	0	0	0	0
Jump	0	0	**128**	0	25	23	53	0	0	0
Pjump	0	0	0	**269**	0	0	0	0	0	0
Run	0	0	0	0	**174**	0	22	32	0	0
Side	0	0	23	0	0	**195**	4	0	0	0
Skip	0	0	32	0	58	0	**153**	0	0	0
Walk	0	0	0	0	0	0	0	**356**	0	0
Wave1	0	0	0	0	0	0	0	0	**291**	36
Wave2	0	21	0	0	0	0	0	0	30	**261**

Table 8 GMM confusion matrix on Weizmann action dataset

	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	**320**	0	0	0	0	0	0	0	0	0
Jack	0	**331**	34	0	0	0	0	0	0	0
Jump	0	0	**116**	10	25	28	50	0	0	0
Pjump	0	0	11	**258**	0	0	0	0	0	0
Run	0	0	0	0	**160**	0	23	45	0	0
Side	0	0	47	0	0	**165**	10	0	0	0
Skip	0	0	45	0	59	0	**139**	0	0	0
Walk	0	0	0	0	11	0	0	**345**	0	0
Wave1	0	0	0	0	0	0	0	0	**259**	68
Wave2	0	23	0	0	0	0	0	0	48	**241**

our experimental results show, the action videos can be modeled better by VarIDM rather than Gaussian mixture model. This is also illustrated by a student's t-test at the significance level of 0.05 (p value=0.0005).

8 Conclusion

Clustering is a fundamental and widely applied methodology in understanding and exploring data. Finite mixture models have been widely applied for clustering. As an approach to positive data clustering, this paper presents a practical variational algorithm for unsupervised learning based on inverted Dirichlet mixtures. There are several promising avenues for future research. For instance, the covariance structure imposed by the inverted Dirichlet is strictly positive. This is rather restrictive and

does not allow for modeling the data in a flexible way. Thus, it is possible to consider the generalized inverted Dirichlet [47, 48] to enlarge the applicability of the proposed model.

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Dr. Wentao Fan for helpful discussions and suggestions.

Appendix A: Proof of Equations (14) and (15)

The general expression for the variational solution $Q_s(\Theta_s)$ (Eq. (13)), can be written as:

$$\ln Q_s(\Theta_s) = \langle \ln p(\mathcal{X}, \Theta) \rangle_{j \neq s} + const \tag{26}$$

where *const* is an additive constant denoting all the terms that are independent of $Q_s(\Theta_s)$. Considering the joint distribution represented in Eq. (9), the following variational solutions for $Q(\mathcal{Z})$ and $Q(\boldsymbol{\alpha})$ can be developed.

Proof of Eq. (14): Variational Solution to $Q(\mathcal{Z})$

$$\ln Q(Z_{ij}) = Z_{ij}\left[\ln \pi_j + \mathcal{R}_j + \sum_{l=1}^{D+1} (\bar{\alpha}_{jl} - 1) \ln X_{il} \right] + const \tag{27}$$

where

$$\mathcal{R}_j = \left\langle \ln \frac{\Gamma(\sum_{l=1}^{D+1} \alpha_{jl})}{\prod_{l=1}^{D+1} \Gamma(\alpha_{jl})} \right\rangle_{\alpha_{j1},\ldots,\alpha_{jD+1}} \tag{28}$$

and

$$\bar{\alpha}_{jl} = \langle \alpha_{jl} \rangle = \frac{u_{jl}}{v_{jl}} \tag{29}$$

Since a closed-form expression cannot be found for $\mathcal{R}_j$, the standard variational inference cannot be applied directly. Therefore, to obtain a closed-form expression, a lower bound approximation is proposed. To provide traceable approximations, the second-order Taylor series expansion is applied in variational inference [49, 50]. In fact, the function $\mathcal{R}_j$ is approximated using a second-order Taylor expansion about the expected values of the parameters $\boldsymbol{\alpha}_j$. Here, $\widetilde{\mathcal{R}_j}$ is defined to denote the

approximation of $\mathcal{R}_j$, and $(\bar{\alpha}_{j1}, ..., \bar{\alpha}_{jD+1})$ to represent the expected values of $\boldsymbol{\alpha}_j$. Replacing $\mathcal{R}_j$ by $\widetilde{\mathcal{R}}_j$ makes the optimization in Eq. (27) traceable.

Considering the logarithmic form of Eq. (6) , formula (27) can be written as

$$\ln Q(\mathcal{Z}) = \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij} \ln \rho_{ij} + const \tag{30}$$

where

$$\ln \rho_{ij} = \ln \pi_j + \widetilde{\mathcal{R}}_j + \sum_{l=1}^{D} (\bar{\alpha}_{jl} - 1) \ln X_{il} \tag{31}$$

All the terms, independent of Z_{ij} can be added to the constant part, therefore, it is straight forward to show

$$Q(\mathcal{Z}) \propto \prod_{i=1}^{N}\prod_{j=1}^{M} \rho_{ij}^{Z_{ij}} \tag{32}$$

In order to find the exact formula for $Q(\mathcal{Z})$ the Eq. (32) needs to be normalized. Simple calculations lead to

$$Q(\mathcal{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{M} r_{ij}^{Z_{ij}} \tag{33}$$

where

$$r_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^{M} \rho_{ij}} \tag{34}$$

Note that $\sum_{j=1}^{M} r_{ij} = 1$, therefore, the standard result for $Q(\mathcal{Z})$ will be

$$\langle Z_{ij} \rangle = r_{ij} \tag{35}$$

Proof of Eq. (15): Variational Solution to $Q(\boldsymbol{\alpha})$

Having a mixture model with M components and assuming the parameter α_{jl} are independent, $Q(\boldsymbol{\alpha})$ can be factorized as

$$Q(\boldsymbol{\alpha}) = \prod_{j=1}^{M}\prod_{l=1}^{D+1} Q(\alpha_{jl}) \tag{36}$$

Consider the variational optimization for the specific factor $Q(\alpha_{js})$. The logarithm of the optimized factor is given by

$$\ln Q(\alpha_{js}) = \sum_{i=1}^{N} r_{ij}\mathscr{T}(\alpha_{js}) + \alpha_{js}\sum_{i=1}^{N} r_{ij}\ln X_{is} + (u_{js}-1)\ln\alpha_{js} - v_{js}\alpha_{js} + const \tag{37}$$

where

$$\mathscr{T}(\alpha_{js}) = \left\langle \ln \frac{\Gamma\left(\alpha_s + \sum_{l\neq s}^{D+1}\alpha_{jl}\right)}{\Gamma(\alpha_s)\prod_{l\neq s}^{D+1}\Gamma(\alpha_{jl})} \right\rangle_{\Theta\neq\alpha_{js}} \tag{38}$$

$\mathscr{T}$ is a function of α_{js}. Since $\mathscr{T}(\alpha_{js})$ is intractable, a lower bound estimation should be found for it. Hence, a first-order Taylor expansion [49] around $\bar{\alpha}_{js}$ (the expected value of α_{js}) is used (See Appendix 8).

$$\mathscr{T}(\alpha_{js}) \geq \bar{\alpha}_{js}\ln\alpha_{js}\left\{\Psi\Big(\sum_{l=1}^{D+1}\bar{\alpha}_{jl}\Big) - \Psi(\bar{\alpha}_{js}) + \sum_{l\neq s}^{D+1}\bar{\alpha}_{jl}\times\Psi'\Big(\sum_{l=1}^{D+1}\bar{\alpha}_{jl}\Big) \times\left(\langle\ln\alpha_{jl}\rangle - \ln\bar{\alpha}_{jl}\right)\right\} + const \tag{39}$$

Substituting this lower bound into Eq. (37), results in an optimal solution for α_{js}

$$\begin{aligned}\ln Q(\alpha_{js}) &= \sum_{i=1}^{N} r_{ij}\bar{\alpha}_{js}\ln\alpha_{js}\Bigg[\Psi\Big(\sum_{i=1}^{D+1}\bar{\alpha}_{jl}\Big) - \Psi(\bar{\alpha}_{js}) + \sum_{l\neq s}^{D+1}\Psi' \\ &\quad\times\Big(\sum_{l=1}^{D+1}\bar{\alpha}_{jl}\Big)\bar{\alpha}_{jl}\left(\langle\ln\alpha_{jl}\rangle - \ln\bar{\alpha}_{jl}\right)\Bigg] \\ &\quad+\alpha_{js}\sum_{i=1}^{N} r_{ij}\ln X_{is} + (u_{js}-1)\ln\alpha_{js} - v_{js}\alpha_{js} + const \\ &= \ln\alpha_{js}(u_{js}+\phi_{js}-1) - \alpha_{js}(v_{js}-\nu_{js}) + const\end{aligned} \tag{40}$$

where

$$\phi_{js} = \sum_{i=1}^{N} r_{ij}\bar{\alpha}_{js}\Bigg[\Psi\Big(\sum_{l=1}^{D+1}\bar{\alpha}_{jl}\Big) - \Psi(\bar{\alpha}_{js}) + \sum_{l\neq s}^{D+1}\Psi'\Big(\sum_{l=1}^{D+1}\bar{\alpha}_{jl}\Big)\times\bar{\alpha}_{jl}\left(\langle\ln\alpha_{jl}\rangle - \ln\bar{\alpha}_{jl}\right)\Bigg] \tag{41}$$

$$\nu_{js} = \sum_{i=1}^{N} r_{ij}\ln X_{is} \tag{42}$$

By taking the exponential of Eq. (40) which is the logarithmic form of Gamma distribution, we will have

$$Q(\alpha_{js}) \propto \alpha_{js}^{u_{js}+\phi_{js}-1} e^{-(v_{js}-v_{js})\alpha_{js}} \tag{43}$$

The optimal solution to the parameters are as

$$\begin{aligned} u_{js}^{*} &= u_{js} + \phi_{js} \\ v_{js}^{*} &= v_{js} - v_{js} \end{aligned} \tag{44}$$

Proof of Equations (18) and (39)

Lower bound of $\mathcal{R}_j$: Proof of Eq. (18)

Since $\mathcal{R}_j$ in Eq. (28) is intractable, a non-linear approximation of the lower bound of $\mathcal{R}_j$ is calculated using the second-order Taylor expansion. First, function $\mathcal{H}$ is defined as follow

$$\mathcal{H}(\boldsymbol{\alpha}_j) = \mathcal{H}(\alpha_{j1}, \ldots, \alpha_{jD+1}) = \ln \frac{\Gamma(\sum_{l=1}^{D+1} \alpha_{jl})}{\prod_{l=1}^{D+1} \Gamma(\alpha_{jl})} \tag{45}$$

where $\alpha_{jl} > 1$. Using the second-order Taylor expansion for $\ln \boldsymbol{\alpha}_j = (\ln \alpha_{j1}, \ldots, \ln \alpha_{jD+1})$ around $\ln \boldsymbol{\alpha}_{j,0} = (\ln \boldsymbol{\alpha}_{j1,0}, \ldots, \ln \boldsymbol{\alpha}_{jD+1,0})$, the lower bound of $\mathcal{H}(\boldsymbol{\alpha}_j)$ is obtained as

$$\begin{aligned} \mathcal{H}(\boldsymbol{\alpha}_j) \geq \mathcal{H}(\boldsymbol{\alpha}_{j,0}) &+ (\ln \boldsymbol{\alpha}_j - \ln \boldsymbol{\alpha}_{j,0})^T \nabla \mathcal{H}(\boldsymbol{\alpha}_{j,0}) \\ &+ \frac{1}{2!} (\ln \boldsymbol{\alpha}_j - \ln \boldsymbol{\alpha}_{j,0})^T \nabla^2 \mathcal{H}(\boldsymbol{\alpha}_{j,0}) (\ln \boldsymbol{\alpha}_j - \ln \boldsymbol{\alpha}_{j,0}) \end{aligned} \tag{46}$$

where $\nabla \mathcal{H}(\boldsymbol{\alpha}_{j,0})$ is the gradient of $\mathcal{H}$ at $\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_{j,0}$ and $\nabla^2 \mathcal{H}(\boldsymbol{\alpha}_{j,0})$ is the Hessian matrix, which gives

$$\begin{aligned} \mathcal{H}(\boldsymbol{\alpha}_j) \geq \mathcal{H}(\boldsymbol{\alpha}_{j,0}) &+ \sum_{l=1}^{D+1} \frac{\partial \mathcal{H}(\boldsymbol{\alpha}_j)}{\partial \ln \alpha_{jl}} |_{\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_{j,0}} (\ln \alpha_{jl} - \ln \alpha_{jl,0}) \\ &+ \frac{1}{2} \sum_{a=1}^{D+1} \sum_{b=1}^{D+1} \frac{\partial^2 \mathcal{H}(\boldsymbol{\alpha}_{j,0})}{\partial \ln \alpha_{ja} \partial \ln \alpha_{jb}} |_{\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_{j,0}} (\ln \alpha_{ja} - \ln \alpha_{ja,0}) \\ &\times (\ln \alpha_{jb} - \ln \alpha_{jb,0}) \end{aligned} \tag{47}$$

Taking the expectation of Eq. (47), the lower bound of function $\mathcal{R}_j$ will be

$$\begin{aligned}\mathcal{R}_j \geq \widetilde{\mathcal{R}}_j &= \ln \frac{\Gamma(\sum_{l=1}^{D+1} \alpha_{jl,0})}{\prod_{l=1}^{D+1} \Gamma(\alpha_{jl,0})} \\ &+ \sum_{l=1}^{D+1} \alpha_{jl,0} \left[\Psi\left(\sum_{l=1}^{D+1} \alpha_{jl,0} \right) - \Psi(\alpha_{jl,0}) \right] \times [\langle \ln \alpha_{jl} \rangle - \ln \alpha_{jl,0}] \\ &+ \frac{1}{2} \sum_{l=1}^{D+1} \alpha_{jl,0}^2 \left[\Psi'\left(\sum_{l=1}^{D+1} \alpha_{jl,0} \right) - \Psi'(\alpha_{jl,0}) \right] \times \langle (\ln \alpha_{jl} - \ln \alpha_{jl,0})^2 \rangle \\ &+ \frac{1}{2} \sum_{a=1}^{D+1} \sum_{\substack{b=1 \\ (b \neq a)}}^{D+1} \left\{ \alpha_{ja,0} \alpha_{jb,0} \Psi'\left(\sum_{l=1}^{D+1} \alpha_{jl,0} \right) (\langle \ln \alpha_{ja} \rangle - \ln \alpha_{ja,0}) \right. \\ &\left. \times (\langle \ln \alpha_{jb} \rangle - \ln \alpha_{jb,0}) \right\} \end{aligned} \tag{48}$$

To prove that the second-order Taylor expansion of $\mathcal{H}(\boldsymbol{\alpha}_j)$ is a lower bound of $\mathcal{H}(\boldsymbol{\alpha}_j)$, it is shown that $\Delta\mathcal{H}(\boldsymbol{\alpha}_j) \geq 0$, where $\Delta\mathcal{H}(\boldsymbol{\alpha}_j)$ denotes the difference between $\mathcal{H}(\boldsymbol{\alpha}_j)$ and its second-order Taylor expansion. The Hessian of $\Delta\mathcal{H}(\boldsymbol{\alpha}_j)$ with respect to $(\ln \alpha_{jl}, \ldots, \ln \alpha_{jD+1})$ is given by (49).

$$Hess = \begin{pmatrix} \begin{matrix} \alpha_{j1}[\Psi(\sum_{l=1}^{D+1} \alpha_{jl}) - \Psi(\alpha_{j1})] \\ +\alpha_{j1}^2[\Psi'(\sum_{l=1}^{D+1} \alpha_{jl}) - \Psi'(\alpha_{jl})] \\ -\bar{\alpha}_{j1}^2[\Psi'(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}) - \Psi'(\bar{\alpha}_{j1})] \end{matrix} & \cdots & \begin{matrix} \alpha_{j1}\alpha_{jD+1}\Psi'(\sum_{l=1}^{D+1} \alpha_{jl}) \\ -\bar{\alpha}_{j1}\bar{\alpha}_{jD+1}\Psi'(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}) \end{matrix} \\ \vdots & \ddots & \vdots \\ \begin{matrix} \alpha_{j1}\alpha_{jD+1}\Psi'(\sum_{l=1}^{D+1} \alpha_{jl}) \\ -\bar{\alpha}_{j1}\bar{\alpha}_{jD+1}\Psi'(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}) \end{matrix} & \cdots & \begin{matrix} \alpha_{jD+1}[\Psi(\sum_{l=1}^{D+1} \alpha_{jl}) - \Psi(\alpha_{jD+1})] \\ +\alpha_{jD+1}^2[\Psi'(\sum_{l=1}^{D+1} \alpha_{jl}) - \Psi'(\alpha_{jD+1})] \\ -\bar{\alpha}_{jD+1}^2[\Psi'(\sum_{l=1}^{D+1} \bar{\alpha}_{jl}) - \Psi'(\bar{\alpha}_{jD+1})] \end{matrix} \end{pmatrix} \tag{49}$$

Substituting $(\ln \alpha_{j1}, \ldots, \ln \alpha_{jD+1})$ by the critical point $(\ln \alpha_{j1,0}, \ldots, \ln \alpha_{jD+1,0})$, reduces (49) to a positive-definite diagonal matrix. Since $(\ln \alpha_{j1,0}, \ldots, \ln \alpha_{jD+1,0})$ is the only critical point, and for all $\alpha_{jl} > 1$, $\Delta\mathcal{H}(\boldsymbol{\alpha}_j)$ is continuous and differentiable, the critical point $(\ln \alpha_{j1,0}, \ldots, \ln \alpha_{jD+1,0})$ is also the global minimum of $\Delta\mathcal{H}(\boldsymbol{\alpha}_j)$. When $(\ln \alpha_{j1}, \ldots, \ln \alpha_{jD+1}) = (\ln \alpha_{j1,0}, \ldots, \ln \alpha_{jD+1,0})$, the global minimum value 0, is reached, therefore, the second-order Taylor expansion is definitely a lower bound of $\mathcal{H}$.

Lower Bound of $\mathscr{T}(\alpha_{js})$: Proof of (39)

The lower bound of $\mathscr{T}(\alpha_{js})$ is approximated in [51] by a first-order Taylor expansion. The first-order Taylor expansion of a convex function is a tangent line of that function at a specific value. Function $\mathcal{F}(\alpha_{js})$ is defined as

$$\mathcal{F}(\alpha_{js}) = \ln \frac{\Gamma\left(\alpha_{js} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right)}{\Gamma(\alpha_{js}) \prod_{l\neq s}^{D+1} \Gamma(\alpha_{jl})} \tag{50}$$

Convexity of $\mathcal{F}(\alpha_{js})$

Since it cannot directly be shown that $\mathcal{F}(\alpha_{js})$ is a convex function of α_{js}, a relative convexity similar to [49] is considered. It can be demonstrated that $\mathcal{F}(\alpha_{js})$ is convex relative to $\ln \alpha_{js}$. Function $\mathcal{F}$, is considered to be convex on an interval if and only if its second derivative is nonnegative in that interval. The first and second derivatives of $\mathcal{F}(\alpha_{js})$ with respect to $\ln \alpha_{js}$ are

$$\frac{\partial \mathcal{F}(\alpha_{js})}{\partial \ln \alpha_{js}} = \left[\Psi\left(\alpha_{js} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right) - \Psi(\alpha_{js})\right]\alpha_{js} \tag{51}$$

$$\begin{aligned}\frac{\partial^2 \mathcal{F}(\alpha_{js})}{\partial (\ln \alpha_{js})^2} &= \left[\Psi\left(\alpha_{js} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right) - \Psi(\alpha_{js})\right]\alpha_{js} \\ &\quad + \left[\Psi'\left(\alpha_{js} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right) - \Psi'(\alpha_{js})\right]\alpha_{js}^2 \\ &= \alpha_{js} \int_0^\infty \frac{1 - e^{-(\sum_{l\neq s}^{D+1} \alpha_{jl})t}}{1 - e^{-t}} e^{-\alpha_{js} t}(1 - \alpha_{js} t) dt\end{aligned} \tag{52}$$

The integral representation of $\Psi(x)$ and $\Psi'(x)$ are defined by

$$\Psi(x) = \int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1 - e^{-t}}\right) dt \tag{53}$$

$$\Psi'(x) = \int_0^\infty \frac{t e^{-xt}}{1 - e^{-t}} dt \tag{54}$$

Considering (53) and (54), Eq. (52) can be written as

$$\frac{\partial^2 \mathcal{F}(\alpha_{js})}{\partial (\ln \alpha_{js})^2} = \alpha_{js} \int_0^\infty f_1(t) f_2(t) dt \tag{55}$$

where $f_1(t)$ and $f_2(t)$ are

$$f_1(t) = \frac{1 - e^{-(\sum_{l\neq s}^{D+1} \alpha_{jl})t}}{1 - e^{-t}} \tag{56}$$

$$f_2(t) = e^{-\alpha_{js}t}(1 - \alpha_{js}t) \tag{57}$$

when $\sum_{l\neq s}^{D+1} \alpha_{jl} > 1$

- if $t > \frac{1}{\alpha_{js}}$ then $f_1(t) < f_1(\frac{1}{\alpha_{js}})$, and $f_2(t) < 0$
- if $t < \frac{1}{\alpha_{js}}$ then $f_1(t) > f_1(\frac{1}{\alpha_{js}})$, $f_2(t) > 0$

Therefore Eq. (55) can be rewritten as

$$\begin{aligned}
\frac{\partial^2 \mathcal{F}(\alpha_{js})}{\partial(\ln \alpha_{js})^2} &= \alpha_{js}\left\{\int_0^{\frac{1}{\alpha_{js}}} f_1(t) f_2(t) dt + \int_{\frac{1}{\alpha_{js}}}^{\infty} f_1(t) f_2(t) dt\right\} \\
&> \alpha_{js}\left\{\int_0^{\frac{1}{\alpha_{js}}} f_1(\frac{1}{\alpha_{js}}) f_2(t) dt + \int_{\frac{1}{\alpha_{js}}}^{\infty} f_1\left(\frac{1}{\alpha_{js}}\right) f_2(t) dt\right\} \\
&= \alpha_{js} f_1\left(\frac{1}{\alpha_{js}}\right) \int_0^{\infty} f_2(t) dt \\
&= \alpha_{js} f_1\left(\frac{1}{\alpha_{js}}\right) \lim_{t\to\infty} t e^{-\alpha_{js}t} = 0
\end{aligned} \tag{58}$$

Hence, it is proven, when $\sum_{l\neq s}^{D+1} \alpha_{jl} > 1$, $\mathcal{F}(\alpha_{js})$ is convex relative to $\ln \alpha_{js}$.

Evaluating Lower Bound by First-Order Taylor Expansion

The lower bound of $\mathcal{F}(\alpha_{js})$ can be calculated by applying the first-order Taylor expansion of $\mathcal{F}(\alpha_{js})$ for $\ln \alpha_{js}$ at $\ln \alpha_{js,0}$, since is a convex function relative to $\ln \alpha_{js}$.

$$\begin{aligned}
\mathcal{F}(\alpha_{js}) &\geq \mathcal{F}(\alpha_{js,0}) + \frac{\partial \mathcal{F}(\alpha_{js})}{\partial \ln \alpha_{js}} | \alpha_{js} = \alpha_{js,0} (\ln \alpha_{js} - \ln \alpha_{js,0}) \\
&= \mathcal{F}(\alpha_{js,0}) + \frac{\partial \mathcal{F}(\alpha_{js})}{\partial \alpha_{js}} \frac{\partial \alpha_{js}}{\partial \ln \alpha_{js}} | \alpha_{js} = \alpha_{js,0} (\ln \alpha_{js} - \ln \alpha_{js,0}) \\
&= \ln \frac{\Gamma\left(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right)}{\Gamma(\alpha_{js,0}) \prod_{l\neq s}^{D+1} \Gamma(\alpha_{jl})} + \left[\Psi\left(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right) - \Psi(\alpha_{js,0})\right] \\
&\times \alpha_{js,0} (\ln \alpha_{js} - \ln \alpha_{js,0})
\end{aligned} \tag{59}$$

Note that when $\alpha_{js} = \bar{\alpha}_{js}$, the equality is reached. Substituting (59) in (39) will result in

$$\mathscr{T}(\alpha)_{js} \geq \left\langle \ln \frac{\Gamma\left(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right)}{\Gamma(\alpha_{js,0}) \prod_{l\neq s}^{D+1} \Gamma(\alpha_{jl})} + \left[\Psi\left(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right) - \Psi(\alpha_{js,0})\right] \right.$$
$$\left. \times \alpha_{js,0}(\ln \alpha_{js} - \ln \alpha_{js,0}) \right\rangle_{\boldsymbol{\alpha}\neq\alpha_{js}} \tag{60}$$
$$\ln \alpha_{js}\alpha_{js,0} \left\{ \left\langle \Psi\left(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl}\right) \right\rangle_{\boldsymbol{\alpha}\neq\alpha_{js}} - \Psi(\alpha_{js,0}) \right\} + const$$

The calculation of the expectation $\langle \Psi(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl}) \rangle_{\boldsymbol{\alpha}\neq\alpha_{js}}$ (in (60)) is analytically intractable. With the same approach explained in Sect. 8, it can be inferred, for $l = \{1, \ldots, D+1\}$ and $l \neq s$, $\Psi(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl})$ is a convex function relative to $\ln \alpha_{js,0}$. To calculate the lower bound, a first-order Taylor expansion for the function $\Psi(\sum_{i=1}^{n} x_i + y)$ at $\ln \hat{x}$, $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$ is applied

$$\Psi\left(\sum_{i=1}^{n} x_i + y\right) \geq \Psi\left(\sum_{i=1}^{n} \hat{x}_i + y\right) + \sum_{i=1}^{n} (\ln x_i - \ln \hat{x}_i) \Psi'\left(\sum_{i=1}^{n} \hat{x}_i + y\right) \hat{x}_i \tag{61}$$

Considering (61), the approximation lower bound of $\langle \Psi(\alpha_{js,0} + \sum_{l\neq s}^{D+1} \alpha_{jl}) \rangle_{\boldsymbol{\alpha}\neq\alpha_{js}}$ is given by

$$\left\langle \Psi\left(\sum_{l\neq s}^{D+1} \alpha_{jl} + \alpha_{js,0}\right) \right\rangle_{\boldsymbol{\alpha}\neq\alpha_{js}} \geq \Psi\left(\sum_{l=1}^{D+1} \alpha_{jl,0}\right) + \sum_{l\neq s}^{D+1} \alpha_{jl,0} \Psi'\left(\sum_{l=1}^{D+1} \alpha_{jl,0}\right) \tag{62}$$
$$\times \left(\langle \ln \alpha_{jl} \rangle - \ln \alpha_{jl,0}\right)$$

The lower bound of $\mathscr{T}(\alpha_{js})$ can be calculated by substituting (62) to (39)

$$\mathscr{T}(\alpha_{js}) \geq \ln \alpha_{js}\alpha_{js,0} \left\{ \Psi\left(\sum_{l=1}^{D+1} \alpha_{jl,0}\right) - \Psi(\alpha_{js,0}) + \sum_{l\neq s}^{D+1} \alpha_{jl,0} \Psi'\left(\sum_{l=1}^{D+1} \alpha_{jl,0}\right) \right.$$
$$\left. \times \left(\langle \ln \alpha_{jl} \rangle - \ln \alpha_{jl,0}\right) \right\} + const \tag{63}$$

References

1. Picard, R.W.: Light-years from lena: video and image libraries of the future. In: Proceeding of the IEEE International Conference on Image Processing (ICIP), vol. 1, pp. 310–313 (1995)
2. Ortega, M., Rui, Y., Chakrabarti, K., Porkaew, K., Mehrotra, S., Huang, T.S.: Supporting ranked boolean similarity queries in mars. IEEE Trans. Knowl. Data Eng. **10**(6), 905–925 (1998)

3. Guha, S., Rastogi, R., Shim, K.: Cure: an efficient clustering algorithm for large databases. Inf. Syst. **26**(1), 35–58 (2001)
4. Pal, N.R., Biswas, J.: Cluster validation using graph theoretic concepts. Pattern Recognit. **30**(6), 847–857 (1997)
5. Comaniciu, D., Meer, P.: Distribution free decomposition of multivariate data. Pattern Anal. Appl. **2**(1), 22–30 (1999)
6. Basu, S., Banerjee, A., Mooney, R.J.: Semi-supervised clustering by seeding. In: Proceeding of the Nineteenth International Conference on Machine Learning (ICML), pp. 27–34 (2002)
7. Dougherty, E.R., Brun, M.: A probabilistic theory of clustering. Pattern Recognit. **37**, 917–925 (2004)
8. Bagirov, A.M., Ugon, J., Webb, D.: Fast modified global k-means algorithm for incremental cluster construction. Pattern Recognit. **44**(4), 866–876 (2011)
9. Law, M.H.C., Topchy, A.P., Jain, A.K.: Multiobjective data clustering. In: Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. II-424–II-430 (2004)
10. Hastie, T., Tibshirani, R.: Discriminant analysis by gaussian mixtures. J. Roy. Stat. Soc. Ser. B **58**(1), 155–176 (1996)
11. Garcia, V., Nielsen, F., Nock, R.: Levels of details for gaussian mixture models. In: Zha, H., Taniguchi, R., Maybank, S.J. (eds.) ACCV (2), volume 5995 of Lecture Notes in Computer Science, pp. 514–525. Springer (2009)
12. Dixit, M., Rasiwasia, N., Vasconcelos, N.: Adapted gaussian models for image classification. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 937–943 (2011)
13. Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) Computer Vision - ECCV 2000, 6th European Conference on Computer Vision, Dublin, Ireland, 26 June– 1 July 2000. Proceedings, Part II, volume 1843 of Lecture Notes in Computer Science, pp. 751–767. Springer (2000)
14. Liu, L., Fan, G.: Combined key-frame extraction and object-based video segmentation. IEEE Trans. Circ. Syst. Video Technol. **15**(7), 869–884 (2005)
15. Song, X., Fan, G.: Joint key-frame extraction and object segmentation for content-based video analysis. IEEE Trans. Circ. Syst. Video Technol. **16**(7), 904–914 (2006)
16. Allili, M.S., Bouguila, N., Ziou, D.: Finite generalized gaussian mixture modeling and applications to image and video foreground segmentation. In: Proceeding of the Fourth Canadian Conference on Computer and Robot Vision (CRV), pp. 183–190 (2007)
17. Bouguila, N.: Spatial color image databases summarization. In: Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), pp. 953–956 (2007)
18. Bouguila, N., Ziou, D.: Online clustering via finite mixtures of dirichlet and minimum message length. Eng. Appl. Artif. Intell. **19**(4), 371–379 (2006)
19. Bdiri, T., Bouguila, N.: Learning inverted dirichlet mixtures for positive data clustering. In: Kuznetsov, S.O., Slezak, D., Hepting, D.H., Mirkin, B. (eds.) Rough Sets, Fuzzy Sets, Data Mining and Granular Computing - 13th International Conference, RSFDGrC 2011, Moscow, Russia, 25–27 June 2011. Proceedings, volume 6743 of Lecture Notes in Computer Science, pp. 265–272. Springer (2011)
20. Bdiri, T., Bouguila, N.: Positive vectors clustering using inverted dirichlet finite mixture models. Expert Syst. Appl. **39**(2), 1869–1882 (2012)
21. Bdiri, T., Bouguila, N.: Bayesian learning of inverted dirichlet mixtures for SVM kernels generation. Neural Comput. Appl. **23**(5), 1443–1458 (2013)
22. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc. Ser. B **39**, 1–38 (1977)
23. Bouguila, N., Ziou, D., Hammoud, R.I.: On bayesian analysis of a finite generalized dirichlet mixture via a metropolis-within-gibbs sampling. Pattern Anal. Appl. **12**(2), 151–166 (2009)
24. Ghahramani, Z., Beal, M.J.: Variational inference for bayesian mixtures of factor analysers. In: Advances in Neural Information Processing Systems (NIPS), pp. 449–455 (1999)

25. Archambeau, C., Opper, M., Shen, Y., Cornford, D., Shawe-Taylor, J.: Variational inference for diffusion processes. In: Advances in Neural Information Processing Systems (NIPS) (2007)
26. Opper, M., Sanguinetti, G.: Variational inference for markov jump processes. In: Advances in Neural Information Processing Systems (NIPS) (2007)
27. Tiao, G.G., Cuttman, I.: The inverted dirichlet distribution with applications. J. Am. Stat. Assoc. **60**(311), 793–805 (1965)
28. Yassaee, H.: Inverted dirichlet distribution and multivariate logistic distribution. Can. J. Stat. **2**(1–2), 99–105 (1974)
29. Ghorbel, M.: On the inverted dirichlet distribution. Commun. Stat. Theory Methods **39**(1), 21–37 (2009)
30. Corduneanu, A., Bishop, C.M.: Variational Bayesian model selection for mixture distributions. In: Proceeding of the International Conference on Artificial Intelligence and Statistics (AISTAT), pp. 27–34 (2001)
31. Saul, L., Jordan, M.I.: Exploiting tractable substructures in intractable networks. In: Advances in Neural Information Processing Systems 8, pp. 486–492. MIT Press, Cambridge (1995)
32. Jaakkola, T.S., Jordan, M.I.: Computing upper and lower bounds on likelihoods in intractable networks. In: Proceeding of the Twelfth International Conference on Uncertainty in Artificial Intelligence, UAI'96, pp. 340–348, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA (1996)
33. Attias, H.: A variational bayesian framework for graphical models. In: Advances in Neural Information Processing Systems 12, pp. 209–215. MIT Press, Cambridge (2000)
34. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005. vol. 2, pp. 524–531 (2005)
35. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via plsa. In: Proceeding of the 9th European Conference on Computer Vision - Volume Part IV, ECCV'06, pp. 517–530, Springer, Berlin, (2006)
36. Yang, J., Jiang, Y.G., Hauptmann, A.G., Ngo, C.W.: Evaluating bag-of-visual-words representations in scene classification. In: Proceeding of the International Workshop on Workshop on Multimedia Information Retrieval, pp. 197–206, ACM, New York, USA, 2007
37. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1447–1454 (2006)
38. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99, pp. 1150-1157, IEEE Computer Society, Washington, DC, USA (1999)
39. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vision **42**(3), 145–175 (2001)
40. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. **43**(3):16:1–16:43 (2011)
41. Luo, J., Wang, W., Qi, H.: Feature extraction and representation for distributed multi-view human action recognition. IEEE J. Emerg. Sel. Top. Circ. Syst. **3**(2), 145–154 (2013)
42. Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G.: Effective codebooks for human action representation and classification in unconstrained videos. IEEE Trans. Multimedia **14**(4), 1234–1245 (2012)
43. Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G.: Effective codebooks for human action representation and classification in unconstrained videos. IEEE Trans. Multimedia **14**(4), 1234–1245 (2012)
44. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
45. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA (1980)

46. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1226–1238 (2005)
47. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1226–1238 (2005)
48. Bouguila, N.: A model-based discriminative framework for sets of positive vectors classification: Application to object categorization. In: 2014 1st International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 277–282 (2014)
49. Ma, Z., Leijon, A.: Bayesian estimation of beta mixture models with variational inference. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2160–2173 (2011)
50. Woolrich, M.W., Behrens, T.E.: Variational bayes inference of spatial mixture models for segmentation. IEEE Trans. Med. Imaging **25**(10), 1380–1391 (2006)
51. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. **37**(2), 183–233 (1999)

A Fully Bayesian Framework for Positive Data Clustering

Mohamed Al Mashrgy and Nizar Bouguila

Abstract The main concern with mixture modeling is to describe data in which each observation belongs to one of some number of different groups. Mixtures of distributions provide a flexible and convenient class of models for density estimation and their statistical learning has been studied extensively. In this context, fully Bayesian approaches have been widely adopted for mixture estimation and model selection problems and have shown some effectiveness due to the incorporation of the prior knowledge about the parameters. In this chapter, we propose a fully Bayesian approach for finite generalized Inverted Dirichlet (GID) mixture model learning using a reversible jump Markov chain Monte Carlo (RJMCMC) approach [23]. RJMCMC enables us to deal simultaneously with model selection and parameters estimation in one single algorithm. The merits of RJMCMC for GID mixture learning is investigated using synthetic data and a real interesting application namely object detection.

1 Introduction

Finite mixture model provides a natural representation of heterogeneity when data are assumed to be generated from two or more distributions mixed in varying proportions. Finite mixture models provide a powerful, flexible and well principled statistical approaches and have been commonly used to model complex data in many applications [1, 6, 10, 22]. Model selection and estimation of parameters are the

M.A. Mashrgy
Department of Electrical and Computer Engineering, Concordia University,
Montreal, QC, Canada
e-mail: m_almash@encs.concordia.ca

N. Bouguila (✉)
Concordia Institute for Information Systems Engineering, Concordia University,
Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_7

fundamental problems in mixture modeling. To date, Gaussian mixture modeling has been the subject of much research because of its relative simplicity [22]. The Gaussian assumption is, however, not realistic in the majority of signal and image processing problems [4, 5, 9]. In general, numerous approaches have been developed for learning of mixture parameters (i.e., both model selection and parameter estimation). These approaches can be categorized into deterministic and Bayesian methods. Deterministic inference is an important branch of inference methodologies, and it has been actively studied, especially during the past 15 years. Using deterministic methods, data are taken as random while parameters are taken as fixed and unknown and the inference is generally based on the likelihood of the data. Among these approaches, the expectation maximization (EM) [15] algorithm has been extensively used in the case of maximum likelihood estimation. However, it has been shown that maximum likelihood estimation suffers from singularities, convergence to local maxima [21], leads to more complex models and then to overfitting [24]. Moreover, likelihood is non-decreasing function on the number of components, thus maximum likelihood approach cannot be used as a model selection criterion. To overcome this problem, many model selection approaches based on Bayesian approximation have been proposed such as Bayesian information criterion, minimum message length, and maximum entropy criterion.

Pure Bayesian techniques can be used as an alternative to learn mixture models and generally provide good results. Using Bayesian approaches for finite mixture modeling is performed by introducing suitable prior distributions for the model's parameters. Moreover, Bayesian approaches provide us with a valid inference without relying on the asymptotic normality assumption since simulation from the posterior distribution of the unknown parameters is feasible [26]. This simulation is generally based on MCMC which is an important tool for statistical Bayesian inference [16, 24]. In this paper, we consider a special MCMC technique, which performs simultaneously parameters estimation and model selection for generalized Inverted Dirichlet (GID) mixture, namely reversible jump MCMC (RJMCMC) sampling previously proposed in [18]. RJMCM has been applied successfully in the past in the case of the Gaussian [23, 27] and Beta [7] mixtures. It provides a general framework for Markov chain Monte Carlo (MCMC) simulation in which the dimension of the parameter space can vary between iterates of the Markov chain.

This chapter is organized as follows. In Sect. 2, we introduce the GID mixture model. Section 3 defines our fully Bayesian framework for learning the GID mixture using the RJMCMC technique. In Sect. 4, split/merge and birth/death moves are explained in detail. Section 5 presents the experimental results, for generated and real data, to show the merits of the proposed approach. Finally, conclusions and future works are presented in Sect. 6.

2 GID Mixture Model

Let us consider a data set $\mathcal{Y}$ composed of N D-dimensional positive vectors, $\mathcal{Y} = (\vec{Y}_1, \vec{Y}_2, \ldots, \vec{Y}_N)$. We assume that $\mathcal{Y}$ is governed by a weighted sum of M generalized Inverted Dirichlet (GID) component densities with parameters $\Theta_M = (\vec{\theta}_1, \vec{\theta}_2, \ldots, \vec{\theta}_M, p_1, p_2, \ldots, p_M)$ where $\vec{\theta}_j$ is the parameters vector of the jth component and $\{p_j\}$ are the mixing weights which are positive and sum to one:

$$p(\vec{Y}_i|\Theta_M) = \sum_{j=1}^{M} p_j p(\vec{Y}_i|\vec{\theta}_j) \tag{1}$$

where $p(\vec{Y}_i|\vec{\theta}_j)$ is the GID distribution with parameters $\vec{\theta}_j = (\alpha_{j1}, \beta_{j1}, \alpha_{j2}, \beta_{j2}, \ldots, \alpha_{jD}, \beta_{jD})$. In mixture-based clustering, each data point $\vec{Y}_i$ is assigned to all classes with different posterior probabilities $p(j|\vec{Y}_i) \propto p_j p(\vec{Y}_i|\vec{\theta}_j)$. The GID distribution allows the factorization of the posterior probability as shown in [19].

$$p(j|\vec{Y}_i) \propto p_j \prod_{l=1}^{D} p_{IBeta}(X_{il}|\theta_{jl}) \tag{2}$$

where we have set $X_{i1} = Y_{i1}$ and $X_{il} = \frac{Y_{il}}{1+\sum_{k=1}^{l-1} Y_{ik}}$ for $l > 1$. $p_{IBeta}(X_{il}|\theta_{jl})$ is an inverted Beta distribution with parameters $\theta_{jl} = (\alpha_{jl}, \beta_{jl})$, $\beta_{jl} > 2, l = 1, \ldots, D$. Thus, the clustering structure underlying $\mathcal{Y}$ is the same as the one underlying $\mathcal{X} = (\vec{X}_1, \ldots, \vec{X}_N)$, where $\vec{X}_i = (X_{i1}, \ldots, X_{iD})$, $i = 1, \ldots, N$, governed by the following mixture model with conditionally independent features:

$$p(\vec{X}_i|\Theta_M) = \sum_{j=1}^{M} p_j \prod_{l=1}^{D} p_{IBeta}(X_{il}|\theta_{jl}) \tag{3}$$

where

$$p_{IBeta}(X_{il}|\theta_{jl}) = \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} X_{id}^{\alpha_{jl}-1}(1 + X_{id})^{-\alpha_{jl}-\beta_{jl}} \tag{4}$$

The mean and the variance of the inverted Beta distribution are as following

$$\mu_{jl} = \frac{\alpha_{jl}}{\beta_{jl} - 1} \tag{5}$$

$$\sigma_{jl}^2 = \frac{\alpha_{jl}(\alpha_{jl} + \beta_{jl} - 1)}{(\beta_{jl} - 2)(\beta_{jl} - 1)^2} \tag{6}$$

Using Eqs. 5 and 6, the parameters α_{jl} and β_{jl} of inverted Beta distribution can be written with respect to the mean and the variance as following

$$\alpha_{jl} = \frac{{\mu_{jl}}^2(1+\mu_{jl}) + \mu_{jl}\sigma_{jl}^2}{\sigma_{jl}^2} \tag{7}$$

$$\beta_{jl} = \frac{\mu_{jl}(1+\mu_{jl}) + 2\sigma_{jl}^2}{\sigma_{jl}^2} \tag{8}$$

then, the probability density function of the inverted Beta, as a function of its mean and variance, can be written as following

$$\begin{aligned} p_{IBeta}(X_{il}|\mu_{jl},\sigma_{jl}^2) = & \frac{1}{\mathcal{B}\left(\frac{{\mu_{jl}}^2(1+\mu_{jl})+\mu_{jl}\sigma_{jl}^2}{\sigma_{jl}^2}, \frac{\mu_{jl}(1+\mu_{jl})+2\sigma_{jl}^2}{\sigma_{jl}^2}\right)} \\ & \times X_{il}^{\left(\frac{{\mu_{jl}}^2(1+\mu_{jl})+\mu_{jl}\sigma_{jl}^2}{\sigma_{jl}^2}-1\right)} \\ & \times \left(1+X_{il}\right)^{-\left(\frac{{\mu_{jl}}^2(1+\mu_{jl})+\mu_{jl}\sigma_{jl}^2+\mu_{jl}(1+\mu_{jl})+2\sigma_{jl}^2}{\sigma_{jl}^2}\right)} \end{aligned} \tag{9}$$

where $\mathcal{B}$ is the beta function which is defined as $\mathcal{B}(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$.

The main problem when dealing with mixture models is to estimate the parameters. A huge number of methods in the literature has been developed in the past [22]. Among these methods, maximum likelihood estimation which maximizes the likelihood through the expectation maximization (EM) algorithm [15, 21] has received a lot of attention. However, EM algorithm suffers from some drawbacks. First, it is highly dependent on the initialization which is the main reason for convergence to local maxima. Second, it suffers from overfitting problem. In EM-based formulation, a latent allocation vector is introduced $\vec{Z}_i = (Z_{i1}, \ldots, Z_{iM})$ and indicates to which mixture component each vector $\vec{X}_i$ belongs to, such that $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^{M} Z_{ij} = 1$ and $Z_{ij} = 1$ if $\vec{X}_i$ belongs to component j and 0, otherwise. $\mathcal{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$ is known as the set of "membership vectors" of the mixture model and its different elements Z_i are supposed to be drawn independently from the following distribution

$$p(Z_i = j) = p_j \quad j = 1, \ldots, M. \tag{10}$$

Thus, the distribution of $\vec{X}_i$ given the class label $\vec{Z}_i$ is

$$p(\vec{X}_i|\Theta_M, \vec{Z}_i) = \prod_{j=1}^{M}\left(\prod_{l=1}^{D} p_{IBeta}(X_{il}|\theta_{jl})\right)^{Z_{ij}} \tag{11}$$

3 GID Bayesian Learning Using RJMCMC

One of the main concerns in mixture modeling is model selection (i.e. determine the number of components). Within Bayesian modeling, many approaches have been proposed to infer the optimal number of components. Examples of Bayesian model selection approaches include Bayes factors, Bayesian information criterion (BIC), deviance information criterion (DIC), RJMCMC, and birth and death processes [3, 11, 17]. In this work we develop a RJMCMC-based method. It allows us to successfully perform both model selection and parameter estimation in one single algorithm. In our proposed Bayesian framework, the number of component M, the parameters which govern the mixture $\vec{\theta}$ components, and the mixing weights $\vec{P} = (p_1, \ldots, p_M)$ are considered as drawn from appropriate distribution. The joint distribution of all variables can be written as

$$p(M, \vec{P}, Z, \vec{\theta}, \mathcal{X}) = p(M)p(\vec{P}|M)p(Z|\vec{P}, M)p(\vec{\theta}|Z, \vec{P}, M)p(\mathcal{X}|\vec{\theta}, Z, \vec{P}, M) \tag{12}$$

where the conditional independencies $p(\vec{\theta}|Z, \vec{P}, M) = p(\vec{\theta}|M)$ and $p(\mathcal{X}|\vec{\theta}, Z, \vec{P}, M) = p(\mathcal{X}|\vec{\theta}, Z)$ are imposed. The joint distribution can be written as following:

$$p(M, \vec{P}, Z, \vec{\theta}, \mathcal{X}) = p(M)p(\vec{P}|M)p(Z|\vec{P}, M)p(\vec{\theta}|M)p(\mathcal{X}|\vec{\theta}, Z, \vec{P}, M) \tag{13}$$

Now, the main goal of the Bayesian inference is to generate realizations from the conditional joint density $p(M, \vec{P}, Z, \vec{\theta}|\mathcal{X})$.

3.1 Priors and Posteriors

In this section, we will define the priors of the different parameters in our hierarchical Bayesian model. These parameters are supposed to be drawn independently. For our model, we have chosen an inverted Beta and inverse gamma distributions as priors for the mean μ_{jl} and the variance σ_{jl}^2, respectively.

$$p(\vec{\mu}_j|\varepsilon, \zeta) = \prod_{l=1}^{D} \frac{1}{\mathcal{B}\left(\frac{\varepsilon^2(1+\varepsilon)+\varepsilon\zeta}{\zeta}, \frac{\varepsilon(1+\varepsilon)+2\zeta}{\zeta}\right)} \mu_{jl}^{\left(\frac{\varepsilon^2(1+\varepsilon)+\varepsilon\zeta}{\zeta}-1\right)} \times \left(1+\mu_{jl}\right)^{-\left(\frac{\varepsilon^2(1+\varepsilon)+\varepsilon\zeta+\varepsilon(1+\varepsilon)+2\zeta}{\zeta}\right)} \tag{14}$$

where ε_{jl} is the location and ζ_{jl} is the shape parameter for the inverted Beta distribution. A common choice as a prior for the variance $\vec{\sigma}_j^2 = (\sigma_{j1}^2, \ldots, \sigma_{jD}^2)$ is inverse gamma distribution, then

$$p(\vec{\sigma}_j^2|\vartheta, \varpi) \sim \prod_{l=1}^{D} \frac{\vartheta^{\varpi}}{\Gamma(\varpi)} {\sigma^2}_{jl}^{-\varpi-1} \exp\left(\frac{\vartheta}{\sigma^2}\right) \tag{15}$$

where ϑ and ϖ represent the shape and scale parameters of inverse Gamma distribution, respectively. Using Eqs. 14 and 15, we have

$$p(\vec{\Theta}|M, \tau) = \prod_{j=1}^{M} p(\vec{\mu}_j|\varepsilon, \zeta) p(\vec{\sigma}_j^2|\vartheta, \varpi) \tag{16}$$

where $\tau = (\varepsilon, \zeta, \vartheta, \varpi)$ are the hyperparameters of $\vec{\Theta}$. Therefore, the full conditional posterior distribution for the mean $\vec{\mu}_j$ and the variance $\vec{\sigma}_j^2$ can be written as following:

$$p(\vec{\mu_j}|\ldots) \propto \prod_{j=1}^{M} p(\vec{\mu}_j|\varepsilon, \zeta) p(\vec{\sigma}_j^2|\vartheta, \varpi) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\Theta}_{Z_i}) \propto p(\vec{\mu}_j|\varepsilon, \zeta) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\theta}_{Z_i}) \tag{17}$$

$$p(\vec{\sigma}_j^2|\ldots) \propto \prod_{j=1}^{M} p(\vec{\mu}_j|\varepsilon, \zeta) p(\vec{\sigma}_j^2|\vartheta, \varpi) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\theta}_{Z_i}) \propto p(\vec{\sigma}_j^2|\vartheta_j, \varpi_j) \prod_{i=1}^{N} p(\vec{X}_i|\vec{\theta}_{Z_i}) \tag{18}$$

The $|\ldots$ is used to denote conditioning on all other variables. In addition, the typical prior choice for the mixing weight $\vec{P}$ is the Dirichlet distribution since it is defined under the constraint of $p_1, \ldots, p_M : \sum_{j=1}^{M} p_j = 1$. Then, the prior can be written as following:

$$p(\vec{P}|M, \delta) = \frac{\Gamma(\sum_{j=1}^{M} \delta_j)}{\prod_{j=1}^{M} \Gamma(\delta_j)} \prod_{j=1}^{M} p_j^{\delta_j - 1} \tag{19}$$

Also, the prior of the membership variable Z is:

$$p(Z|P, M) = \prod_{j=1}^{M} p_j^{n_j} \tag{20}$$

where n_j represents the number of vectors belonging to jth cluster. Using Eqs. 19 and 20 we get

$$\begin{aligned} p(\vec{P}|\ldots) &\propto p(Z|\vec{P}, M) p(\vec{P}|M, \delta) \\ &\propto \prod_{j=1}^{M} p_j^{n_j} \frac{\Gamma(\sum_{j=1}^{M} \delta_j)}{\prod_{j=1}^{M} \Gamma(\delta_j)} \prod_{j=1}^{M} p_j^{\delta_j - 1} \propto p_j^{n_j + \delta_j - 1} \end{aligned} \tag{21}$$

which is simply proportional to a Dirichlet distribution with parameters $(\delta_1 + n_1, \ldots, \delta_M + n_M)$. Besides, using Eq. 10 the membership variable posterior can be obtained as following

$$p(Z_i = j|\ldots) \propto p_j \prod_{l=1}^{D} p_{IBeta}(X_{il}|\Theta_{jl}) \tag{22}$$

Another hierarchical level can be introduced to represent the priors of the hyperparameters in the model. First, the hyperparameters, ε and ζ which are associated with $\vec{\mu}_j$ are given uniform and inverse Gamma priors, respectively:

$$p(\varepsilon) \sim \mathcal{U}_{[a,b]} \tag{23}$$

$$p(\zeta|\varphi, \varrho) \sim \frac{\varrho^{\varphi} \exp(-\varrho/\zeta)}{\Gamma(\varphi)\zeta^{\varphi+1}} \tag{24}$$

where $a = \min\{X_{il}, i = 1, \ldots, N; l = 1, \ldots, D\}$ and $b = \max\{X_{il}, i = 1, \ldots, N; l = 1, \ldots, D\}$. According to the two previous equations, the conditional posterior for ε and ζ can be written as:

$$p(\varepsilon|\ldots) = p(\varepsilon) \prod_{j=1}^{M} p(\vec{\mu}_j|\varepsilon, \zeta) \tag{25}$$

$$p(\zeta|\ldots) = p(\zeta|\varphi, \varrho) \prod_{j=1}^{M} p(\vec{\mu}_j|\varepsilon, \zeta) \tag{26}$$

Also, the hyperparameters for ϑ and ϖ, which are associated with the variance $\vec{\sigma}_j^2$, are given inverse Gamma and exponential priors, respectively:

$$p(\vartheta|\lambda, \nu) \sim \frac{\nu^{\lambda} \exp(-\nu/\vartheta)}{\Gamma(\lambda)\vartheta^{\lambda+1}} \tag{27}$$

$$p(\varpi|\phi) \sim \phi \exp(-\phi\varpi) \tag{28}$$

From these two pervious equations, the conditional posteriors for ϑ and ϖ are written as:

$$p(\vartheta|\ldots) \propto p(\vartheta|\lambda, \nu) \prod_{j=1}^{M} p(\vec{\sigma}_j^2|\vartheta, \varpi) \tag{29}$$

$$p(\varpi|\ldots) \propto p(\varpi|\phi) \prod_{j=1}^{M} p(\vec{\sigma}_j^2|\vartheta, \varpi) \tag{30}$$

Finally, for the number of components, M, the common choice is uniform distribution between 1 and a predefined integer M_{max}.

4 RJMCMC Moves

Practically, RJMCMC allows moves between parameter subspaces by allowing the following six types of moves:

1. Update the mixing parameters $\vec{P}$
2. Update the parameters $\vec{\mu}_{jl}$ and $\vec{\sigma}^2_{jl}$
3. Update the membership variable Z
4. Update the hyperparmeters ε, ζ, ϑ, and ϖ
5. Split one component into two, or merge two into one
6. The birth or death of an empty component

Each step is called a move $t = 1, \ldots, 6$ and a sweep is defined as a complete pass over the six moves. Since the first four moves do not change the number of clusters, they can be considered as classic Gibbs sampling moves. On the other hand, moves 5 and 6 involve changing the number of component, M, by 1.

Assume that we are in state Δ_M, where $\Delta_M = (Z, P, M)$. The MCMC step representing move (5) takes the form of a Metropolis-Hastings step by proposing a move from a state Δ_M to $\hat{\Delta}_M$ with target probability distribution (posterior distribution) $p(\Delta_M|\chi)$ and proposal distribution $q_t(\Delta_M, \hat{\Delta}_M)$ for the move t. When we are in the current state Δ_M, a given move t to destination $\hat{\Delta}_M$ is accepted with probability

$$p_t(\Delta_M, \hat{\Delta}_M) = \left(1, \frac{p(\hat{\Delta}_M|\chi)q_t(\hat{\Delta}_M, \Delta_M)}{p(\Delta_M|\chi)q_t(\Delta_M, \hat{\Delta}_M)}\right) \tag{31}$$

In the case of a move type where the dimension of the parameter does not change we use an ordinary ratio of densities. A move from a point Δ_M to $\hat{\Delta}_M$ in a higher dimensional space is done by drawing a vector of continuous random variables u, independent of Δ_M and the new state $\hat{\Delta}_M$ is determined by using an invertible deterministic function of Δ_M and u: f(Δ_M, u) [23]. On the other hand, the move from $\hat{\Delta}_M$ to Δ_M can be carried out using the inverse transformation. Hence, the move acceptance probability is given by

$$p_t(\Delta_M, \hat{\Delta}_M) = \min\left(1, \frac{p(\hat{\Delta}_M|\chi)r_m(\hat{\Delta}_M)}{p(\Delta_M|\chi)r_m(\Delta_M)q(u)}\left|\frac{\partial(\hat{\Delta}_M)}{\partial(\Delta_M, u)}\right|\right) \tag{32}$$

where $r_m(\Delta_M)$ is the probability of choosing move type m when we are in state Δ_M, and $q(u)$ is the density function of u. The last term $\frac{\partial(\hat{\Delta}_M)}{\partial(\Delta_M, u)}$ is the Jacobian function arising from the variable change from (Δ_M, u) to state $\hat{\Delta}_M$. All RJMCMC moves are discussed in the following subsections.

4.1 Gibbs Sampling Move

The first four steps of RJMCMC are based on simple Gibbs sampling where the parameters are drawn from their known full conditional distributions. The first move is to draw the mixing weight from a Dirichlet distribution as shown in Eq. 21. The second move is based on drawing the mixture's parameters using Eqs. 14 and 15. According to these equations, it is clear that the full conditional distributions are complex and are not in well-known forms. So, Gibbs sampling is not an appropriate choice in this case. The Metropolis-Hastings (M-H) algorithm [12, 13] could be used. At sweep t, the mean μ_{jl} can be generated using the M-H algorithm as following:

1. Generate $\hat{\mu}_j \sim q\left(\mu_j | \mu_j^{(t-1)}\right)$ and $u \sim \mathcal{U}_{[0,1]}$
2. Calculate $r = \frac{p(\hat{\mu}_j|\ldots) q\left(\mu_j^{(t-1)}|\hat{\mu}_j\right)}{p(\mu_j^{(t-1)}|\ldots) q\left(\hat{\mu}_j|\mu_j^{(t-1)}\right)}$
3. if $r < u$ then μ_j^t else $\mu_j^t = \mu_j^{(t-1)}$

The most important issue in M-H algorithm is choosing the candidate generating density q (proposal distribution) in order to keep the mean within the range of $\mu \in [a, b]$. A popular choice for q is a random walk where the previously simulated parameter$\hat{a}$ μ_{jl}), value is used to generate the following value $\hat{\mu_{jl}}$. We propose to generate the new mean $\mu_j^{(t)}$ from inverted Beta $\mathcal{IB}$ distribution (w.r.t mean and variance), where its mean is the previous mean value $\mu_j^{(t-1)}$ and its variance is a constant value C (we take $C = 2.5$). The new generated value of the mean using the proposal distribution is

$$\hat{\mu}_j \sim \mathcal{IB}\left(\mu_j^{(t-1)}, C\right) \tag{33}$$

For the variance $\sigma^{2(t)}$ we have

1. Generate $\hat{\sigma^2}_j \sim q\left(\sigma^2{}_j | \sigma^2{}_j^{(t-1)}\right)$ and $u \sim \mathcal{U}_{[0,1]}$
2. Calculate $r = \frac{p(\hat{\sigma^2}_j|\ldots) q\left(\sigma^2{}_j^{(t-1)}|\hat{\sigma^2}_j\right)}{p(\sigma^2{}_j^{(t-1)}|\ldots) q\left(\hat{\sigma^2}_j|\sigma^2{}_j^{(t-1)}\right)}$
3. if $r < u$ then $\sigma^{2t}{}_j$ else $\sigma^{2t}{}_j = \sigma^{2(t-1)}_j$

where the proposal distribution q is given by

$$\hat{\sigma^2}_j \sim \mathcal{LN}\left(\sigma_j^{2(t-1)}, e^2\right) \tag{34}$$

where $\mathcal{LN}$ refers to the lognormal distribution with mean $\log\left(\sigma^{2(t-1)}_j\right)$ and variance e^2. The third move is to generate the missing data $Z_i (1 \geq i \leq N)$ from a simulated standard uniform random variables u_i, $Z_i = j$ if $p(Z_i = 1|\ldots) + \cdots + p(Z_i =$

$j-1|\ldots) < u_i \leq p(Z_i = 1|\ldots)+\ldots p(Z_i = j|\ldots)$. Finally, the Gibbs sampling is used to update the hyperparameters ε, ζ, ϑ,and ϖ given by Eqs. 25, 26, 29, and 30, respectively.

4.2 Split and Combine Moves

In move (5), we make a random choice between attempting to split or combine, with probabilities a_M and b_M where $b_M = 1-a_M$, respectively. It is clear that, $a_{M_{max}} = 0$ and $b_1 = 0$, otherwise we choose $a_M = b_M = 0.5$ for $M = 1, \ldots, M_{max}$, where M_{max} is the maximum value allowed for M. The combine move is constructed by randomly choosing a pair of components (j_1, j_2), which must be adjacent; in other words they must meet the following constraint: $\mu_{j1} < \mu_{j2}$, where there is no other μ_j in the interval $[\mu_{j1}, \mu_{j2}]$. Then, these two components can be merged and M is reduced by 1. We denote the new formed component by j^* which contains all the observations that were allocated to j_1 and j_2. Finally, we generate the parameter values for the new components $p_{j*}, \mu_{j*}, \sigma^2_{j*}$ by preserving the zeroth, first, and second moments which are calculated as follows

$$p_{j*} = p_{j1} + p_{j2} \tag{35}$$

$$p_{j*}\mu_{j*} = p_{j1}\mu_{j1} + p_{j2}\mu_{j2} \tag{36}$$

$$p_{j*}(\mu_{j*} + \sigma^2_{j*}) = p_{j1}(\mu_{j1} + \sigma^2_{j1}) + p_{j2}(\mu_{j2} + \sigma^2_{j2}) \tag{37}$$

For the split type move, a component $j*$ is chosen randomly and we split it into two components $j1$ and $j2$ with new parameters p_{j1}, μ_{j1}, and σ^2_{j1} and p_{j2}, μ_{j2},and σ^2_{j2}, respectively, which confirm Eqs. 35, 36, and 37. Since there are 3 degrees of freedom in achieving this, we need to generate, from a Beta distribution, a three-dimensional random vector $u = [u_1, u_2, u_3]$ to define the new parameters [23]. And we set

$$p_{j1} = w_{j*}u_1 \quad p_{j1} = w_{j*}(1-u_1) \tag{38}$$

$$\begin{aligned} \mu_{j1} &= \mu_{j*} - u2\sqrt{\sigma^2_{j*}\frac{p_{j2}}{p_{j1}}} \\ \mu_{j2} &= \mu_{j*} + u2\sqrt{\sigma^2_{j*}\frac{p_{j1}}{p_{j2}}} \end{aligned} \tag{39}$$

$$\sigma_{j1}^2 = u_3(1-u_2^2)\sigma_{j*}^2\frac{p_{j*}}{p_{j1}}$$
$$\sigma_{j2}^2 = (1-u_3)(1-u_2^2)\sigma_{j*}^2\frac{p_{j*}}{p_{j2}} \tag{40}$$

For the new generated components, the adjacency condition defined in the combine move must be checked to make sure that the split/combine is reversible or not. If this condition is rejected, split/combine move is not reversible, the split move is rejected. Otherwise, the split move is accepted and we reallocate the $j*$ into the new components j_1 and j_2 using Eq. (10). According to Eq. 32, the acceptance probability R for the split and combine moves types can be calculated using the following

$$R = \frac{P(Z, P, M+1, \varepsilon, \zeta, \varpi, \vartheta|\mathrm{X})b_{M+1}}{p(Z, P, M, \varepsilon, \zeta, \varpi, \vartheta|\mathrm{X})a_M P_{alloc} q(u)}\left|\frac{\partial\hat{\Delta}_M}{\partial(\Delta_M, u)}\right| \tag{41}$$

where the acceptance probability for the split is $min(1, R)$, and for the combine move is $min(1, R^{-1})$. P_{alloc} is the probability of making this particular allocation to components j_1 and j_2:

$$P_{alloc} = \prod_{Z_i=j_1}\frac{p_{j1}p(x_i|\mu_{j1}, \sigma_{j1}^2)}{p_{j1}p(x_i|\mu_{j1}, \sigma_{j1}^2) + p_{j2}p(x_i|\mu_{j2}, \sigma_{j2}^2)}$$
$$\times \prod_{Z_i=j_2}\frac{p_{j1}p(x_i|\mu_{j2}, \sigma_{j2}^2)}{p_{j1}p(x_i|\mu_{j1}, \sigma_{j1}^2) + p_{j2}p(x_i|\mu_{j2}, \sigma_{j2}^2)} \tag{42}$$

Also, $\left|\frac{\partial\hat{\Delta}_M}{\partial(\Delta_M,u)}\right|$ is the Jacobian of the transformation from the state $(w_{j*}, \mu_{j*}, \sigma_{j*}^2, u_1, u_2, u_3)$ to state $(w_{j1}, \mu_{j1}, \sigma_{j1}^2, w_{j2}, \mu_{j2}, \sigma_{j2}^2)$

$$\left|\frac{\partial\hat{\Delta}_M}{\partial(\Delta_M, u)}\right| = \frac{|\mu_{j1}-\mu_{j2}|p_{j*}\sigma_{j1}^2\sigma_{j2}^2}{u_2(1-u_2^2)u_3(1-u_3)\sigma_{j*}^2} \tag{43}$$

4.3 Birth and Death Moves

In Death/Birth move, we first make a random choice between birth and death with the same a_m and b_m as above. If the birth move is chosen, the values of the parameters of the new components $(\mu_{j*}, \sigma_{j*}^2)$ are drawn from the associated prior distributions given by Eqs. 17 and 18, respectively. Also, the mixing weight of the new component is drawn from:

$$p_{j*} \sim \mathcal{B}e(1, M) \tag{44}$$

In order to keep the constraint of $\sum_{j=1}^{M} p_j + p_{j*} = 1$, we re-scale the previous value of p_j, $j = 1 : M$ by multiplying them with $1 - p_{j*}$. The acceptance probabilities for the birth and death are $\min\{1, R\}$ and $\min\{1, R^{-}1\}$, respectively, where

$$R = \frac{p(M+1)}{p(M)} \frac{1}{\mathcal{B}(\delta, M\delta)} p_{j*}^{\delta-1} (1 - p_{j*})^{N+M\delta-M} (M+1) \frac{a_{M+1}}{M_0 b_M} \frac{1}{p(p_{j*})} (1 - p_{j*}^{M}) \tag{45}$$

where $\mathcal{B}$ is Beta function and M_0 is the number of empty components before the birth.

5 Experimental Results

In this section, experiments are carried out in order to evaluate the benefits of using the proposed model. The simulations are conducted on both synthetic and real data extracted from a challenging application namely object detection.

5.1 Synthetic Data

This section is dedicated to the generated datasets. Its main aim is to investigate the ability of our algorithm to estimate the mixture parameters and to select the number of clusters correctly. We generated three different multidimensional datasets (3-dimensional) from the GID mixture model. The first dataset was generated from a 2-components model. The second one was generated from a 3-components mixture. The third dataset was generated from a 4-components model. Table 1 shows the real and estimated parameters obtained for these datasets. On the other hand Table 2 shows the estimated posterior probabilities for the considered number of components for the three datasets as well as the percentage of accepted split-combine and birth-death moves. According to this table it is clear that our algorithm favours each time the correct number of components. Figure 1 shows how the algorithm moves between the components which is shown by plotting the number of components as a function of the number of sweeps. According to the obtained results, we can conclude that our algorithm has an excellent learning ability.

5.2 Object Detection

Advances in multimedia technology has caused an exponential increasing of the number of images generated everyday. This huge amount of visual data needs to be efficiently organized and indexed. To solve this problem, a lot of different approaches

Table 1 Real parameters used to generate the synthetic data sets (n_j represents the number of elements in cluster j) and the estimated ones using our RJMCMC algorithm

		Real Parameters			Estimated Parameters		
Dataset	j	μ_j	σ_j^2	p_j	$\hat{\mu}_j$	$\hat{\sigma}_j^2$	$\hat{p}_j$
Dataset1	1	1.00	2.00	0.50	0.98	2.10	0.45
	2	7.00	4.00	0.50	7.50	6.00	0.55
Dataset2	1	1.00	1.25	0.33	0.98	1.56	0.26
	2	4.50	2.00	0.33	4.57	3.62	0.34
	3	9.00	3.30	0.33	9.37	5.17	0.40
Dataset3	1	1.00	1.25	0.25	0.86	1.21	0.21
	2	3.50	2.50	0.25	2.21	6.7	0.24
	3	6.50	1.67	0.25	2.24	6.80	0.25
	4	12.00	11.11	0.25	12.07	14.06	0.30

Table 2 The estimated posterior probabilities of the number of components given the data for the three datasets

Datasets	N	$p(k\|y)$		
Dataset 1	200	$p(1\|y) = 0.0017$	$\mathbf{p(2\|y) = 0.9123}$	$p(3\|y) = 0.0697$
		$p(4\|y) = 0.0113$	$p(5\|y) = 0.0050$	$p(>5\|y) = 0.0$
Dataset 2	300	$p(1\|y) = 0.0003$	$p(2\|y) = 0.1903$	$\mathbf{p(3\|y) = 0.4703}$
		$p(4\|y) = 0.2948$	$p(5\|y) = 0.1420$	$p(>5\|y) = 0.0929$
Dataset 3	400	$p(1\|y) = 0.0$	$p(2\|y) = 0.0$	$p(3\|y) = 0.2657$
		$\mathbf{p(4\|y) = 0.4180}$	$p(5\|y) = 0.2190$	$p(>5\|y) = 0.0973$

have been developed in the past [2, 8]. Object detection is an important problem challenging problem related to content-based image indexing and retrieval and has several applications (e.g. video surveillance, object recognition). In this section we shall focus on the application of our model to pedestrian and car detection problems. An important step in object detection is the extraction of low level features to describe the images. Many visual descriptors have been proposed in the past (see, for instance, [25]). Here, we use local Histogram of Oriented Gradient (HOG) descriptor which generates positive features and which has been shown to be efficient and convenient for different object detection tasks [14]. Experiments are conducted by considering three windows for the HOG descriptor which allows representing each image by 81-dimensional vector of features. The experimental results are conducted by considering our proposed GID mixture model using RJMCMC-based learning (GID-RJMCMC). The performance obtained by the proposed model (GIDM-RJMCMC) is compared with GID and GMM mixture models using EM-based learning [20] (GIDFS/GMMFS) without feature selection and also when feature selection is considered (GIDnoFS/GMMnoFS) as developed in [19].

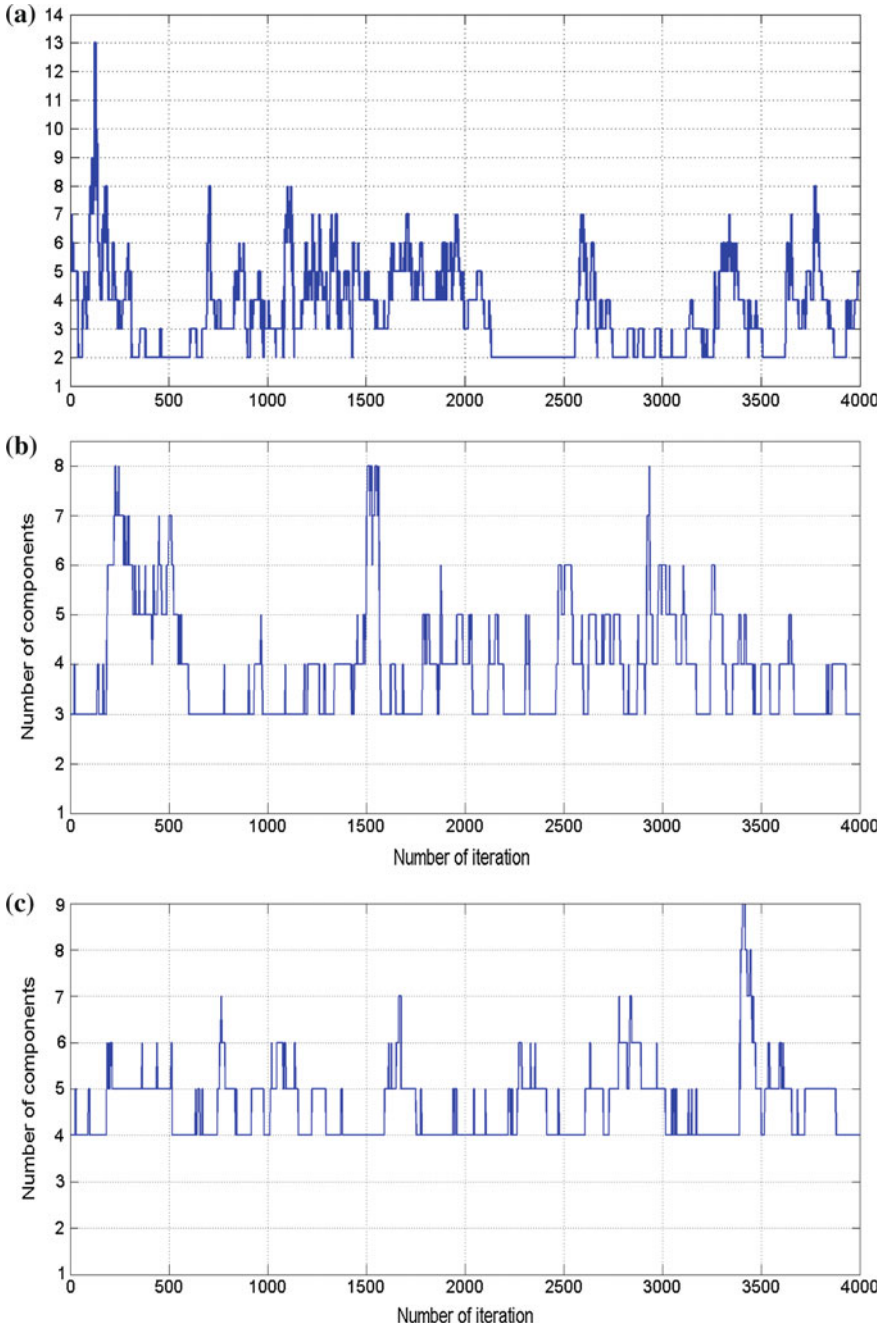

Fig. 1 The number of components versus the number of sweeps for **a** Dataset 1, **b** Dataset 2, and **c** Dataset 3

Fig. 2 Examples of car images

Fig. 3 Examples of non-car images

Table 3 Car detection accuracy when different approaches are considered

Model	Accuracy (%)
GIDM-RJMCMC	85.88
GIDFS	84.59
GIDnoFS	80.76
GMMFS	74.00
GMMnoFS	72.77

5.2.1 Car Detection

The dataset that we consider here contains images of cars side views which were collected at UIUC.[1] The dataset consists of 1050 images (550 car and 500 non-car images). Figures 2 and 3 show examples of images from this dataset. The first 100 images from both car and non-car images are used for training and the rest for testing. Table 3 shows the detection accuracies using GIDM-RJMCMC, Gaussian and GID mixtures learned via EM with and without feature selection. According to this table, it is clear that GIDM-RJMCMC outperforms the other tested approaches.

5.2.2 Human Detection

Another challenging task that we consider here is human detection. We consider the INRIA Static Person dataset[2] to evaluate the proposed model. The data consists of

[1] http://cogcomp.cs.illinois.edu/Data/Car/.

[2] http://pascal.inrialpes.fr/data/human/.

Fig. 4 Examples of images containing humans

Fig. 5 Examples of negative images used for human detection task

both positive (containing humans) and negative examples (images that do not contain humans). 400 images are used for training (200 positive examples and 200 negative ones). On the other hand, the testing set consists of 741 images, 288 of them are positive examples and the remaining 453 are negative examples. Figures 4 and 5 show samples of positive and negative examples, respectively.

Table 4 shows the classification accuracy for the INRIA dataset. According to this table, it is clear that the proposed model GIDM-RJMCMC outperforms GIDFS, GIDnoFS, GMMFS, and GMMnoFS.

On the other hand, for model selection, and for both datasets (car and human), our algorithm successfully determined the correct number of components as shown in Table 5.

Table 4 Human detection accuracies when different approaches are considered

Model	Accuracy (%)
GIDM-RJMCMC	72.60
GIDFS	68.55
GIDnoFS	65.56
GMMFS	57.35
GMMnoFS	53.00

Table 5 The estimated posterior probabilities of the number of components for the car and human datasets

Datasets	$p(k\|y)$		
Car detection	$p(1\|y) = 0.1361$	$\mathbf{p(2\|y) = 0.7810}$	$p(3\|y) = 0.0571$
Dataset	$p(4\|y) = 0.0223$	$p(5\|y) = 0.0027$	$p(> 5\|y) = 0.0007$
Human detection	$p(1\|y) = 0.4049$	$\mathbf{p(2\|y) = 0.5660}$	$p(3\|y) = 0.0291$
Dataset	$p(4\|y) = 0.0000$	$p(5\|y) = 0.0000$	$p(> 5\|y) = 0.0000$

6 Conclusion

This chapter describes a RJMCMC algorithm for fully Bayesian learning of GID mixtures. The proposed learning approach allows simultaneous model selection and parameter estimations in one single algorithm. We presented experimental results using synthetic data and a challenging real-life application namely object detection. According to the obtained results it is clear that the proposed approach is promising. A potential future work that we are currently working on, using the same methodology proposed in [19], is the introduction of feature selection in the proposed framework which may improve further the learning results.

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Bouguila, N.: Spatial color image databases summarization. In: The IEEE International Conference on Acoustics. Speech, and Signal Processing (ICASSP), pp. 953–956. Honolulu, HI, Apr 2007
2. Bouguila, N., Daoudi, K.: Learning concepts from visual scenes using a binary probabilistic model. In: Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP), pp. 1–5 (2009)
3. Bouguila, N., Wang, J.H., Hamza, A.B.: Software modules categorization through likelihood and bayesian analysis of finite dirichlet mixtures. J. Appl. Stat. **37**(2), 235–252 (2010)
4. Bouguila, N., Ziou, D.: Dirichlet-based probability model applied to human skin detection. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 521–524 (2004)
5. Bouguila, N., Ziou, D.: A powerful finite mixture model based on the generalized Dirichlet distribution: Unsupervised learning and applications. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR), pp. 280–283 (2004)
6. Bouguila, N., Ziou, D.: Improving content based image retrieval systems using finite multinomial dirichlet mixture. In: The IEEE Workshop on Machine Learning for Signal Processing (MLSP), pp. 23–32. Sao Luis, Brazil, Oct 2004
7. Bouguila, N., Elguebaly, T.: A fully bayesian model based on reversible jump MCMC and finite beta mixtures for clustering. Expert Syst. Appl. **39**(5), 5946–5959 (2012)

8. Bouguila, N., ElGuebaly, W.: A statistical model for histogram refinement. In: Kurková, V., Neruda, R., Koutník, J. (eds.) Artificial Neural Networks - ICANN 2008, 18th International Conference, Prague, Czech Republic, September 3–6, 2008, Proceedings, Part I. Lecture Notes in Computer Science, vol. 5163, pp. 837–846. Springer (2008)
9. Bouguila, N., ElGuebaly, W.: Discrete data clustering using finite mixture models. Pattern Recognit. **42**(1), 33–42 (2009)
10. Bouguila, N., Ziou, D.: Online clustering via finite mixtures of dirichlet and minimum message length. Eng. Appl. Artif. Intell. **19**(4), 371–379 (2006)
11. Bouguila, N., Ziou, D., Hammoud, R.I.: On bayesian analysis of a finite generalized dirichlet mixture via a metropolis-within-gibbs sampling. Pattern Anal. Appl. **12**(2), 151–166 (2009)
12. Casella, G., George, E.I.: Explaining the gibbs sampler. Am. Stat. **46**(3), 167–174 (1992)
13. Chib, S., Greenberg, E.: Understanding the metropolis-hastings algorithm. Am. Stat. **49**(4), 327–335 (1995)
14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 886–893 (2005)
15. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc. B **39**(1), 1–38 (1977)
16. Elguebaly, T., Bouguila, N.: Bayesian learning of finite generalized gaussian mixture models on images. Sig. Proc. **91**(4), 801–820 (2011)
17. Elguebaly, T., Bouguila, N.: A bayesian approach for the classification of mammographic masses. In: Sixth International Conference on Developments in eSystems Engineering (DeSE), 2013, pp. 99–104, Dec 2013
18. Green, P.J.: Reversible jump markov chain monte carlo computation and bayesian model determination. Biometrika **82**, 711–732 (1995)
19. Mashrgy, M.A., Bdiri, T., Bouguila, N.: Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted dirichlet mixture models. Knowl.-Based Syst. **59**, 182–195 (2014)
20. Mashrgy, M.A., Bouguila, N., Daoudi, K.: A statistical framework for positive data clustering with feature selection: Application to object detection. In: 21st European Signal Processing Conference, EUSIPCO 2013, Marrakech, Morocco, September 9–13, 2013, pp. 1–5 (2013)
21. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions. Wiley Series in Probability and Statistics, 2nd edn. Wiley, Hoboken (2008)
22. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley Series in Probability and Statistics. Wiley, New York (2000)
23. Richardson, S., Green, P.J.: On Bayesian analysis of mixtures with an unknown number of components. J. Roy. Stat. Soc. B **59**(4), 731–792 (1997)
24. Robert, C.P.: The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation, 2nd edn. Springer, Berlin (2007)
25. Rocha, A., Goldenstein, S.: PR: More than meets the eye. In: Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV), pp. 1–8 (2007)
26. Schnatter, F.S.: Finite mixture and Markov switching models. Springer, New York (2006)
27. Zhang, Z., Chan, K., Wu, Y., Chen, C.: Learning a multivariate gaussian mixture model with the reversible jump mcmc algorithm. Stat. Comput. **14**(4), 343–355 (2004)

Part III
Ontologies and Multi-agents

Applying Information Extraction for Abstracting and Automating CLI-Based Configuration of Network Devices in Heterogeneous Environments

A. Martinez, M. Yannuzzi, J. López, R. Serral-Gracià and W. Ramirez

Abstract With the continuous growth of current networks, configuration management has become increasingly relevant to the Information and Communication Technologies (ICT) field. Despite numerous standardization efforts, network administrators continue to rely on Command-Line Interfaces (CLIs) to modify and control the configuration of network devices. Nevertheless, network administrators must deal with the complexities that derive from this practice. On one hand, CLI-based configuration hinders the automation of network configuration tasks which are typically required in autonomic management. The only means for achieving a certain degree of automation is the creation of custom scripts, which is neither scalable nor practical, and is the reason why configuration management tasks are mainly performed through manual intervention. On the other hand, CLIs are generally both device and vendor-specific. In the context of heterogeneous network infrastructures—i.e., networks typically composed of multiple devices from different vendors—the use of several CLIs raises serious Operation, Administration and Management (OAM) issues. Moreover, multi-vendor configurations not only differ syntactically. Overall, the utilization of proprietary mechanisms allows neither

A. Martinez (✉) · M. Yannuzzi · R. Serral-Gracià
Networking and Information Technology Lab (NetIT Lab),
Technical University of Catalonia (UPC), Barcelona, Spain
e-mail: annym@ac.upc.edu

M. Yannuzzi
e-mail: yannuzzi@ac.upc.edu

R. Serral-Gracià
e-mail: rserral@ac.upc.edu

J. López
Department of Electronics and Communications Technologies,
Autonomous University of Madrid (UAM), Madrid, Spain
e-mail: jorge.lopez_vergara@uam.es

W. Ramirez
Advanced Network Architectures Lab (CRAAX),
Technical University of Catalonia (UPC), Barcelona, Spain
e-mail: wramirez@ac.upc.edu

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_8

reusing the configurations nor sharing knowledge consistently between vendors' domains. Due to this heterogeneity, CLIs typically provide a help feature which is in turn a useful source of knowledge to enable semantic interpretation of a configuration space. The large amount of information a network administrator must learn and manage makes Information Extraction (IE) and other forms of natural language analysis of the Artificial Intelligence (AI) field key enablers for the network device configuration space. In this chapter we present an Ontology-Based Information Extraction (OBIE) System from the Command-Line Interface (CLI) of network devices. This system exploits natural language resources already available in CLIs in order to extract relevant information and automatically build the semantics of each configuration space. Overall, our solution provides network administrators with a simple tool which entirely automates and abstracts the complexities and heterogeneity of underlying configuration environments in order to reduce time and effort in the configuration of network devices. With such a tool, network administrators will no longer have to read hundreds of manuals, and configuration scripts can be automatically updated for new devices or system upgrades. We developed a prototype implementation to show how we complete the loop from the process of IE, to the configuration of network devices and final testing.

Keywords Ontology-based information extraction · Configuration management · Autonomic management · CLI · Semantics

1 Introduction

Network management is key to network administrators as a means to remotely manage and control their networks. This broad functional area encompasses the monitoring, provisioning and configuration functions required to ensure reliable performance, operation, recovery, administration and maintenance of the network and its related assets [1]. Although many aspects of network management are fully covered in practice by well-known protocols already in place—such as the Simple Network Management Protocol (SNMP) [2] for network monitoring, which has actually become the de facto standard in the field, to the extent that, currently, almost any device in the network is compliant with SNMP—other several aspects still remain largely unsolved. One of these aspects includes the *configuration* of network devices [3, 4]. The problem of network configuration management in multi-vendor networks has been around for long time. Nevertheless, in an effort to cope with the ever increasing traffic demand, industry has mainly focused on the development of data and control plane technologies leaving network management innovation far behind.

The task of network configuration has become one of the most critical and complex areas in network management [5]. Network configuration typically deals with the maintenance, setup, repair and expansion of services and network functions. It is performed for multiple reasons, but overall, it aims to deploy end-to-end network services, ensure performance, minimize downtime, support rollback in case of failures,

enable device software management as well as collect configuration data. Notice that, in general, the term *network configuration* refers to the configuration of the network as a whole, for example, the deployment of a Virtual Private Network (VPN) service is a typical configuration task, usually expressed through high-level requirements which translate into low-level (individual) device configurations. Our focus in this paper is on the configuration of network devices rather than the network as a whole.

The lack of standard protocols for network device configuration makes network management increasingly complex for network administrators—particularly in the context of heterogeneous network infrastructures. Multi-vendor networks—i.e., networks conformed by devices from multiple vendors—emerge in an effort to avoid single-vendor dependencies which respond to technical, practical and business-driven matters. In this context, the absence of standard protocols has prompted the use of proprietary mechanisms for the configuration of network devices. The Command-Line Interface (CLI) is in fact the preferred mechanism for network device configuration—near 95 % of current network devices are configured through proprietary CLIs [5]. Despite its widespread use for configuration purposes, CLIs have numerous limitations, some of which we will briefly overview.

Proprietary Protocol. The most significant limitation of CLIs is their *proprietary* nature, i.e., CLI-based configuration environments are not standard, and thus, are specific to each vendor. This is exacerbated by the fact that within each vendor's space a CLI can also be specific to a device model or operating system version. This means that, the terminology, commands, configuration operations and related concepts can be dissimilar even among devices of a single vendor. In order to cope with the ever changing configuration environment, network administrators must develop advanced skills, gain specialized knowledge and continuously update to encompass the full range of devices available in the network market. Figure 1 depicts an extract of the CLI environment for two different vendors (Juniper and Cisco, respectively). From this figure we can observe that different sets of terminologies and commands are used by each vendor to refer to the *same* configuration operation, namely, configuration of an IP standard Access Control List (ACL). Notice that, not only commands differ syntactically but the granularity and arrangement of the hierarchy also differs among vendors.

Lack of semantic interoperability. Furthermore, the foundations of the configuration problems are not restricted to syntactical differences, but most importantly, they extend to semantic dissimilarities between CLIs as well. Due to their proprietary nature, device vendors customize their CLIs as a way to unequivocally distinguish from their competitors (as shown in Fig. 1). This leads to definitions of the configuration space in their own terms. In some cases vendors set the hallmark by launching their own terminology, while in other cases their interpretation of the domain leads to misleading use of terms with respect to the common routing domain knowledge, or even overlapping meanings in relation to different terminologies with other vendor configuration spaces. The lack of common standards for the conceptualization of the routing domain has led to scenarios wherein the same terms are used to refer to different concepts or where different sets of terminologies have the same meaning. As CLIs were devised for human operators, in practice, this issue is addressed by

Fig. 1 CLI configuration commands for configuring an access control list on two different router vendors. **a** Router Vendor: Juniper. **b** Router Vendor: Cisco

featuring help descriptors which aim to resolve the semantics of the configuration domain by providing users with a concise natural language description of commands and variables. Generally, help descriptors are a way to narrow down to the common and shared conceptualizations of the domain of knowledge and guide users through the configuration process. In this context, help descriptors are unquestionably valuable resources for network administrators as a means to disambiguate the semantics of configuration commands. However, in the context of automated environments, there are no means for achieving interoperability at a semantic level.

Human/User-driven. The CLI was ideally designed for user-driven operation, i.e., manual-based configuration. This explains its text-based interface and natural language help feature. Moreover, the heterogeneity of CLIs also hinders the automation of network configuration in multi-vendor networks. The only means to achieve automation to a certain extent relies on the use of configuration scripts, an approach that aims to simplify recurrent configuration tasks. However, scripting is neither scalable nor practical as network administrators must adjust and fine-tune their configurations for each system upgrade or network equipment purchase. For this reason, CLI-based configuration is rather challenging and mostly performed in a manual way. Manual-based configurations are likely undesired, not only because they are error-prone and increasingly complex, but mostly because they significantly increase the Operational Expenditures (OPEX).

High Operational Expenditures. Performing network device configuration through CLI-based mechanisms entails high operational expenditures. Network device configuration is considered in fact the major contributor to OPEX for current telecom providers [5]. This stems from the fact that manual configuration is

expensive. The delay in service provisioning naturally leads to increased costs, in addition to the costs that carry network outages whenever there are mistakes in the configuration process and the expenses required to continuously train network administrators in the configuration of each vendors' space.

Error-prone. Configuration errors can lead to network outages, performance degradation and increased operational expenditures [3]. CLI-based configurations are highly sensitive to errors due to their manual nature, and note that this can potentially affect other network devices or the network as a whole.

According to the previous facts, the proprietary, user-driven and manual-based nature of CLI-based mechanisms make this technology an unsuitable solution to automate network device configuration in the context of heterogeneous networks. Overall, the issue with CLI-based mechanisms is the lack of semantic interoperability between configuration spaces. The means for semantically defining the configuration domain are solely expressed in the form of natural language textual resources targeting human network operators. These facts lead us to the belief that semantic technologies and Natural Language Processing (NLP) techniques can significantly contribute to bridge the semantic interoperability gap in the network device configuration domain, by exploiting the knowledge natively provided by vendors in the CLIs of network devices.

The remainder of this chapter is organized as follows. Section 2 provides an overview on state-of-the-art approaches for network device configuration. Section 3 presents our semantic approach for the abstraction and automation of CLI-based configurations. Section 4 provides performance evaluation of our OBIE System. Finally, Sect. 5 concludes the chapter.

2 Ontology-Based Approaches to Network Management: State of the Art

The seamless configuration of routers is yet far from being accomplished in the network domain. Based on the limitations of CLI-based mechanisms for the configuration of network devices—in the context of heterogeneous networks—telecom providers have raised their concerns regarding the need of a practical and scalable solution to this largely unsolved issue. The fact that this problem is still in force is mainly because, in the absence of standard mechanisms, vendor's best interests have imposed. Indeed, proprietary approaches are an effective way of distinguishing from others and lead in a such competitive market. Nevertheless, in the recent years operators' interest for enabling interoperability in such contexts has pushed toward different lines of work. These efforts have emerged both from industry and standardization bodies.

The Network Configuration Protocol (NETCONF) [6] is the most recent effort of the Internet Engineering Task Force (IETF) to develop a standard network configuration protocol for IP devices. NETCONF aims to fill the gap in network device

configuration by featuring state maintenance, concurrency, configuration locking, transactional-safe operations across multiple devices, automatic roll-back and operation ordering, distinction between configuration and operational data and consistency in a standard and practical way of use. The major limitation to the adoption of NETCONF is the absence of standards for network management information definition, i.e., the protocol does not foresee a formal and explicit way of specifying the resources being managed (e.g., interfaces, access-lists, ports, etc.). The lack of a data modeling language pushes the interoperability problem to a new level. Initially, several data models emerged as potential candidates for network management definition [7–9], but neither of them found a path in practice, either because of their increased complexity or lack of semantics. Other NETCONF implementations were built over proprietary data models which in turn raised clear interoperability issues. Most recently, YANG [10]—an IETF proposed standard—has become the most solid contribution for a common data model language for NETCONF. However, almost a decade later and in spite of initial implementations of NETCONF and YANG, the problem remains and CLI continues to be the preferred protocol for network device configuration among network administrators [5]. It is clear that there is still a long path before YANG and NETCONF truly become leading standards in the network configuration arena.

Moreover, the emergence of Software-Defined Networks [11], a new paradigm which aims at providing openness and programmability of network functions, offers new possibilities for network management. In [12] authors discuss on the potential benefits of SDN and OpenFlow [13]—the most common protocol for enabling SDN—to improve various aspects of network management. They suggest that these technologies cannot only ease configuration through software programmability instead of fixed set of commands but also benefit from centralized management. Undoubtedly, OpenFlow and SDNs are key to improve and ease overall network management [12]. However, its flow-oriented nature makes it unsuitable to resolve basic network management operations. This is the case for configurations targeting the router itself (and not actually things going "through" the router, for which it would be indeed largely effective) or administrative tasks (e.g., setting user access rights, such as a user password, etc.).

In a different line of work, industry has approached the device configuration issue by developing dedicated software agents as a means to force static mappings between commands of different configuration spaces. The downside of such approach is that it is not scalable in the context of dynamic environments and requires skillful development teams dedicated to update and maintain these static agents.

From academia, several works have emerged in an effort to integrate ontologies and other semantic technologies to achieve interoperability at the semantic level. Next, we will review the status of this research field.

2.1 Semantic-Based Approaches for Device Configuration in Other Domains

Configuration management is not restricted to IP network devices and is in fact an essential task to many other domains, for example, the case of configuration in the scope of smart homes, manufacturing, e-commerce or service deployment. There is clear evidence on the use of ontologies and other semantic technologies to address the configuration issue in these domains [14–20]. It is clear that, due to the inherent differences and dissimilar requirements between domains the configuration issue is approached from different perspectives. Nevertheless, the overall aim is to automate the configuration process by using ontologies to formally represent the domain's knowledge, as a means to support reasoning and enable reuse of shared conceptualizations.

The research work led by authors in [14, 15] introduces a novel holistic system for intelligent Smart Home (SH) environments to support device auto-configuration and intelligent control under energy efficiency requirements. Their solution is based on an ontology framework, capable of providing efficient control logics and intelligent decision making. In addition, they develop a semantic extension to the standard Universal Plug and Play (UPnP) protocol to enable communication capabilities of intelligent devices in the context of SHs.

In [16] authors develop a recommender system to support requirement elicitation in a product configuration system. They capture customer requirements represented in an OWL-ontology and assess consistency with respect to manufacturers' specifications (constraints) represented in the Semantic Web Rule Language (SWRL). It also identifies mandatory requirements yet not specified by the customer and suggest them as a means to complete the configuration of the product. In a similar line of research, the work presented in [18–20] targets product configuration systems for e-commerce through ontology-based approaches.

Moreover, in [17] authors introduce an ontology-based approach for a product configuration system in the e-business field. The aim is to identify customer requirements—expressed in natural language—and output the configuration design of the product that best meets his needs. Herein, three ontologies were developed to represent, (i) customer needs (ii) product functionalities and (iii) product configuration. Finally, an ontology mapping approach and the use of a Bayesian Network enables automatic conversion between customer needs and product configuration.

Indeed, the scope of configuration as targeted in the aforementioned research initiatives differs from the low-level aspect of the configuration issue in network device configuration. However, they unveil the potential of semantic technologies to reconcile the differences in configuration environments and support translation from custom requirements (in our case "custom" CLIs) to common shared foundations of the domain knowledge.

2.2 Semantic-Based Approaches for the Router Configuration Domain

As introduced earlier, network configuration is one of many functional areas of network management. Indeed, several ontology-based research proposals have emerged in the scope of network management *monitoring*, *security*, autonomic management and *information models*. Readers are referred to [21] for a deeper study on ontology-based network management proposals. Nevertheless, semantic approaches to the configuration aspect of network management are scarce. Next, we will survey the state-of-the-art in ontology-based network configuration management solutions. To this end, we classify semantic approaches into two groups, (i) ontology-based approaches to network configuration and (ii) ontology-based approaches to network device configuration. The former refers to solutions wherein configuration is devised for the network as a whole, while the later refers to semantic approaches targeting the device configuration issue.

(i) Ontology-based Network Configuration Approaches.
The approaches grouped within this category abstract from the heterogeneity of device configurations and deal instead with the autonomic configuration of the network—at a higher level [22–24]. Overall, the aim of these approaches is to provide the network with self-management and context-awareness configuration capabilities. They propose—in different contexts—the use of semantic technologies to provide a smart environment in which configuration services are triggered whenever a network condition is given. To this end, they combine the use of the Ontology Web Language (OWL) and the Semantic Web Rule Language (SWRL) to model the management information and network behavior, respectively. SWRL enables rule integration into the ontology, so whenever a condition is fulfilled a service is automatically invoked and then executed. In the context of IP networks this service can be represented by a configuration script, which includes the CLI device-specific commands to the requested service. As can be seen, these works are restricted to static device configurations as underlying issues remain unsolved. This means that to ensure scalability and flexibility of their solutions, an approach to resolve device configuration heterogeneity is required.

(ii) Ontology-based Network Device Configuration Approaches.
Within this category we classify ontology-based approaches to the semantic interoperability problem in network device configuration management. In the work presented in [25] authors introduce an ontology-driven approach to the semantic interoperability problem in network management and validate it for the case of multi-vendor router configuration. Their major contribution is a generic similarity-based ontology mapping strategy which can be seamlessly applied across the ITU-T Telecommunication Management Network (TMN) layer model. For validating the configuration use case, they built vendor-specific ontologies (one per network vendor), wherein CLI commands where properly modeled and classified. They further

applied the mapping strategy to a set of selected commands to assess the semantic match between both configuration spaces. Beyond the limitations of the similarity function and computational methods—pointed out by the authors in [25]—scalability of this approach is an issue, in as much as, ontologies for CLI environments are not given by vendors' in advance. Thus, formal representations of the CLI knowledge must be manually built by domain experts—a task which is in essence sufficiently complex and challenging—and continuously updated as new features, vendors or releases emerge. As if that was not enough, the ontology expert would require to gain expertise in the new CLI environment beforehand. The ideal scenario would be to assume that vendors handle ontologies in advance for every CLI, in a similar way as drivers are provided for every device. Nevertheless, this is a demanding requirement which is far from vendors' road-map. To the best of our knowledge there are no further efforts in this line. Nevertheless, we firmly believe that ideas from the ontology research arena can still be brought to the CLI configuration domain to achieve interoperability at the semantic level. The semantic interoperability problem in network configuration management requires a solution capable of adapting to the dynamics of current configuration environments in an easy and automated way.

3 Ontology-Based Information Extraction from the Command-Line Interface

Herein, we present an Ontology-Based Information Extraction (OBIE) System from the Command-Line Interface (CLI) of network devices. In Sect. 3.1 we will describe the fundamentals of our approach, in Sect. 3.2 we will introduce the general architecture of our system, in Sect. 3.3 we will provide insights on our specified domain ontology and finally, in Sect. 3.4 we will delve into the specifics of our semantic approach for IE.

3.1 Fundamentals

Overall, our work is based on the assumption that device configuration knowledge can be automatically extracted from the information natively provided by vendors in their configuration CLIs. An important aspect of CLIs is that because of its human-oriented nature it is largely based on natural textual language. For this reason, Information Extraction (IE)—a form of natural language analysis—becomes a key technology to automatically find and retrieve relevant information from the CLI. Furthermore, OBIE—a recently emerged sub-field of IE [26]—which incorporates the use of a formal ontology can help improve this process. The general notion is that an ontology—*"a formal and explicit specification of a shared conceptualization"* [27]—provides

a model of the information to be extracted. Accordingly, the OBIE system has the ability to link natural textual resources to formal semantic models.

From a macro perspective, the design of the OBIE system poses two main challenges: (i) defining a formal knowledge model of the network device configuration domain (i.e., the domain ontology) and (ii) developing a learning approach for IE from the configuration CLI. Regarding the first challenge, we have created our own structured knowledge-base of the switch/router configuration space, which we have named, **O**ntology for the **N**etwork **D**evice **C**onfiguration domain (ONDC) [28]. ONDC formally specifies the most relevant concepts of the domain and integrates a lexicon of the networking vocabulary. In the context of our approach, this ontology provides a comprehensive and vendor-neutral coverage of the device configuration knowledge. Overall, concepts in the ontology conform to networking standards and well-known technical terminology, reflecting the knowledge of the networking domain regardless of vendors' specifics. The main motivation for this, is that—apart from proprietary technologies—the vast majority of protocols and features to be set on a network device are common across multi-vendor platforms, otherwise, there would be no means to provide network interoperability. In light of this, configuration capabilities are just about the same for all devices, what essentially changes is the way in which vendors express this knowledge in their CLIs. The differences among CLI-based environments are basically determined by, (i) the granularity of the tree structure, (ii) the arrangement of commands in the hierarchy, (iii) the syntax—e.g., the use of different terminologies for expressing the same concepts—and (iv) the semantics—e.g., the use of the same terminologies for expressing different concepts. In light of all this, we require a solution capable of exploiting this information to unambiguously determine the semantics of each space. This leads us to the second challenge, that is, developing an approach for IE which allows to reconcile the differences between heterogeneous CLIs. Because of the differentiating features of CLIs, we developed a methodology which not only exploits the (explicit) knowledge given in the form of natural textual language, but moreover, we actually exploit the structure of the CLI itself, i.e., implicit knowledge. The reason for this, is that commands are arranged in a hierarchy structure by relation, therefore, contiguous levels in the CLI are semantically associated either because they are specifications of an upper level command or attributes of the same.

3.2 General Architecture

The general architecture of our OBIE System is depicted in Fig. 2. The design is fully-modular in order to allow the system to be easily extended to other application domains. The inputs of our system are limited to (i) the domain ontology (cf., Sect. 3.3)—which formally defines the knowledge of the network device configuration domain—and (ii) the CLI—as natively provided by vendors (i.e., unprocessed). Whereas, the output is a device-specific version of the target domain-ontology, i.e., the ontology populated with instances of the configuration commands. Because of

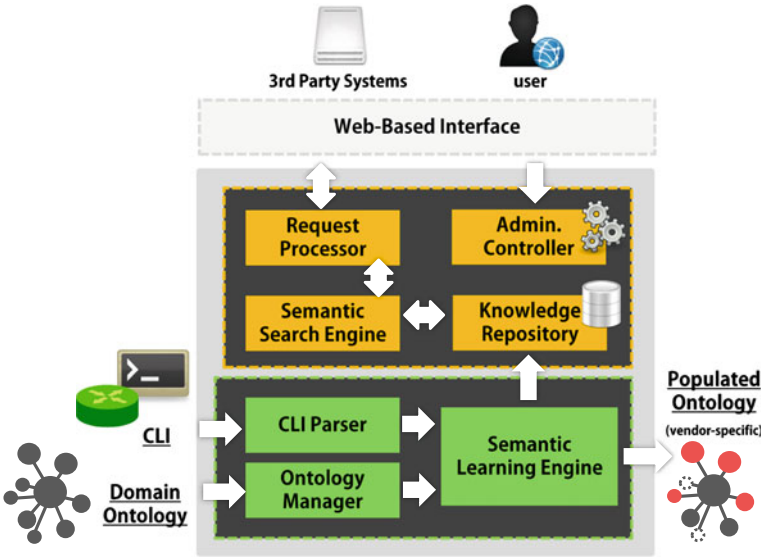

Fig. 2 General architecture of the OBIE system for the device configuration domain

the nature of our approach, we foresee the portability of our system to other application domains, wherein configuration relies on the extended use of CLIs. For instance, consider the configuration of printers in a large business organization or the initial setup of voting machines for an electoral process. In other words, the methodology herein developed is independent of the underlying domain (e.g., IP network devices), but instead, considers features that are specific to any CLI-based environment, such as, hierarchical knowledge or structural data (commands/helps).

The basic architectural components of our system are, namely, (1) CLI Parser, (2) Ontology Manager, (3) Semantic Learning Engine, (4) Knowledge Repository, (5) Administration Controller, (6) Request Processor, and (7) Semantic Search Engine (cf. Fig. 2). The **CLI Parser** is responsible of manipulating the structure of the CLI in order to separate commands (*cmd*) and variables (*var*) from help descriptors (*help*) (Fig. 3 depicts the typical structure of a CLI). Though all three structural elements are key to derive the semantics from the CLI, it is important to distinguish between each type as only commands and variables are target of instantiation. While the information provided in the *helps* can serve to contextualize, disambiguate and identify relevant concepts, which will ultimately assist in the process of semantic instantiation. Furthermore, the CLI Parser also scans the hierarchy to form the complete set of valid configuration statements (cf., Fig. 3), i.e., executable sequences of commands and variables. The reason for this is that given the hierarchical and relational nature of CLIs—in most cases—single commands are not sufficient to provide the complete semantics, instead, it is the combination of commands in the hierarchy which build the meaning of a configuration action. The **Ontology Manager** relies on the OWL

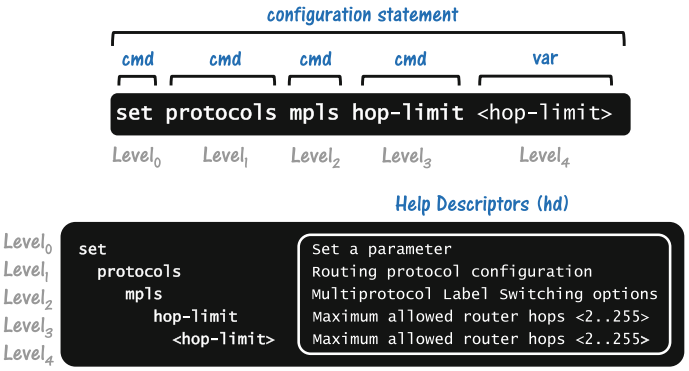

Fig. 3 Typical CLI structure

API [29] to read, access and manipulate the domain-ontology. Furthermore, it creates a Java-based graph of the input ontology, which will be key to our IE algorithm. It also exports all available configuration operations via Web Services to third-party applications. The **Semantic Learning Engine** carries the algorithms for the extraction of configuration knowledge from the CLI. This module is the core of our system and comprises the logic which finally makes up to the instantiation of commands into semantic categories (cf., Sect. 3.3.). The **Knowledge Repository** provides storage capabilities for the generated semantic models. Moreover, ontologies are stored on the basis of its heterogeneity, i.e., vendor, model and Operating System (OS) release of the device. The **Administration Controller** enables user-level administrative control of the system. It provides functions such as, adding, updating or deleting semantic models from the repository; requesting the system to parse a new CLI or manually manipulating ontologies. The **Request Processor** manages all configuration requests in order to retrieve the corresponding commands for a given semantic operation. The configurations are requested by means of (a) atomic operations—which have been exposed by the Ontology Manager—and (b) device tuples (vendor, device model and OS version). For instance, consider requesting configuring a DHCP address pool for the Cisco 7200 OS v-12.4 or the HUAWEI NE20E-S. Finally, the **Semantic Search Engine** is the module responsible of building and formatting the valid sequence of commands and variables from the semantic structure.

Moreover, observe that modules are grouped into blocks, which define the functional modes of our system, i.e., *offline* (lower block) and *online* (upper block). The *offline* functionality is responsible for performing the semantic abstraction of multi-vendor (i.e., heterogeneous) configuration environments. Accordingly, the system's *intelligence* lies within this mode. The Semantic Learning Engine (cf., Fig. 2)—which represents the core module of this functionality—carries the logic and algorithms for extracting and interpreting the information of the CLIs. In Sect. 3.4 we will show how the semantic instantiation of commands is done by thoroughly describing our approach for IE from the CLI. Moreover, the *online* functionality provides a web-based interface through which users or third-party systems can retrieve

semantic-based configurations for heterogeneous (i.e., dissimilar) network devices. In other words, this mode enables access to the semantic models generated in the offline mode, so external systems or applications can benefit from multi-vendor configuration abstraction. To illustrate the online functionality, consider a system or administrator that must perform the real deployment of an initial network planning strategy, which involves a high number of configurations across multiple devices from different vendors. A process of this nature, could make single requests to our system for each required configuration operation and this would retrieve the commands for each available device model. For instance, request the commands for setting the domain's name for all available devices in the network.

3.3 Ontology for the Network Device Configuration Domain

The Ontology for Network Device Configuration (ONDC) acts as a general semantic foundation for the configuration of network devices. As previously stated, it provides a common and shared conceptualization of the configuration knowledge, regardless of vendors' specifics. The ontology was formally defined using the Web Ontology Language (OWL)—the de-facto language for encoding knowledge over the Semantic Web—and built with Protègè, a powerful free open-source ontology editing tool and knowledge acquisition system developed by Stanford for the creation, edition and manipulation of ontologies. Moreover, we used the Protègè API to access, create and manipulate ontology resources.

We have defined over 600 concepts relevant to the routing configuration domain and near 320 operations. We have developed our ontology based on the use of all OWL constructs, namely, classes, individuals, properties, restrictions, etc., in order to enrich the domain knowledge model. We defined, hierarchical (i.e., taxonomic relations) and non-hierarchical relationships between concepts, in an effort to improve the information content of the domain. Moreover, we modeled user-defined datatypes using the pattern facet restriction feature of OWL2 to define custom types to match regular expressions. This feature will enable us to validate domain-specific types of data, e.g., to identify ranges or an IPv4 address—which is a 32 bit number expressed by a standard notation of the form 192.45.32.120 where dot separated numbers range from 0 to 255. The design of an ontology is closely related to the ultimate use or purpose of the knowledge representation model. In the context of our approach, ONDC constitutes a valuable resource to guide the configuration information extraction from the CLI. In light of this, we have determined the need of defining two distinct layers, namely, the router operation layer and the router resource layer. The notion of a layered structure of the ONDC ontology is depicted in Fig. 4. The former defines the entities, concepts and resources of the domain both, physical (e.g., an interface or a LAN port for the routing domain) and virtual (e.g., a routing protocol or the OSPF hello interval). Moreover, the latter defines the functional concepts of the domain, i.e., the set of operations that can be performed over virtual and physical resources—e.g., configure a router host-name or remove a static IP route, etc. Notice

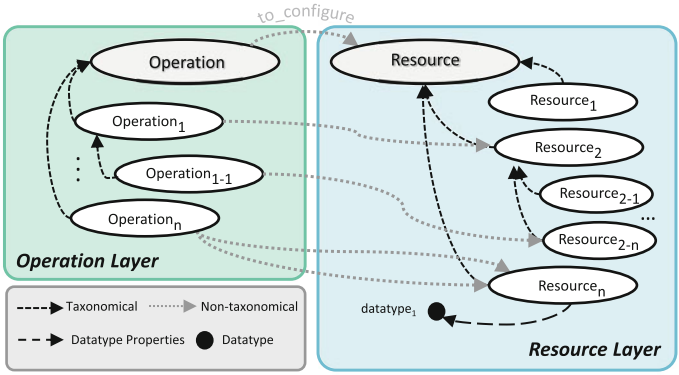

Fig. 4 ONDC layered structure

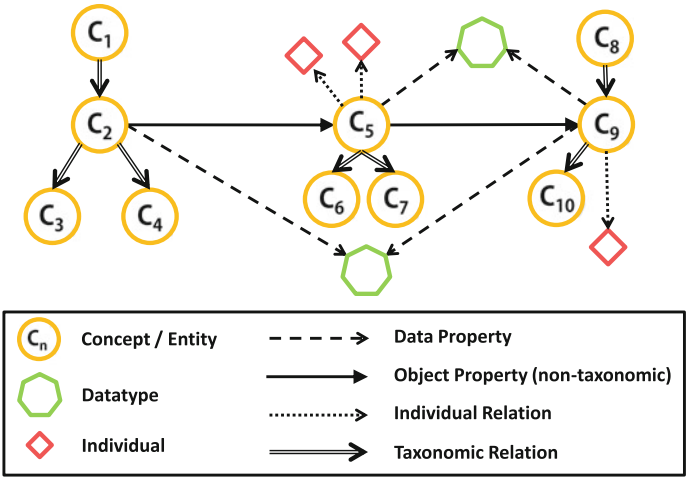

Fig. 5 Ontology modeled as a semantic graph

that concepts in the operation layer are specified in terms of verb phrases (e.g., set, delete, configure, show, etc.) and semantically associated to concepts in the resource layer (cf., Fig. 4). In short, a resource represents a component that can be supplied or consumed in an operation.

For illustration purposes, from now on we will consider the ontology as a semantic graph G (cf., Fig. 5).

Definition 1 The **Semantic Network** (G) is a directed graph where nodes represent concepts of the networking domain, and edges represent attributes or relations between concepts.

For the sake of readability, we will use the terms "*node*", "*concept*" and "*entity*" interchangeably for the rest of this chapter. Likewise, the terms "*edge*" and

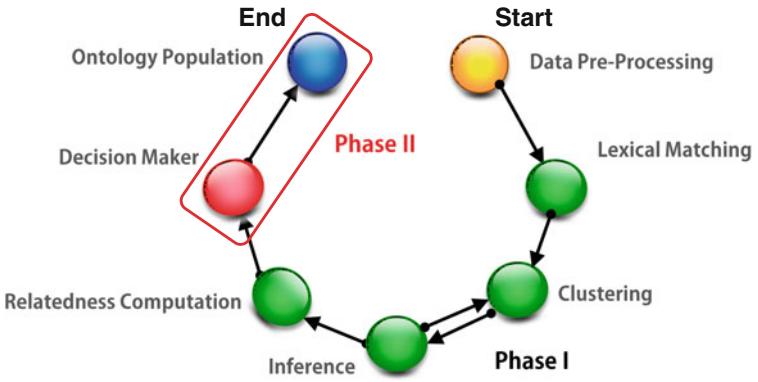

Fig. 6 Configuration knowledge extraction algorithm (Phase I and II)

"*relation*" will be used interchangeably to refer to the semantic links between ontological concepts. Notice that there are different types of relations, namely, *taxonomic*, which reflect subsumer relations, *non-taxonomic*, which reflect object-type of relations (i.e., between entities), *individual*, which represent membership relations and finally, *data-property*, which represent datatype attributes.

3.4 Semantics Extraction from the Command-Line Interface

In this section, we present our methodology for configuration knowledge extraction from the CLI. Figure 6 depicts a general diagram of the IE algorithm. Overall, our methodology is based on a two-phase process. In the first phase (*Stage 1*) we independently identify *verb phrases* and *networking entities* (concepts) from the CLI. The identification of entities is done with respect to particular components of the *resource* layer of our domain ontology. In the second phase (*Stage 2*) we integrate this knowledge (i.e., verbs and domain entities) in an effort to classify commands into their corresponding semantic categories—with respect to the components of the *operation* layer. The methodology we have developed is a multi-step process (cf., Fig. 6) which combines the use of NLP techniques and other semantic resources to unveil the semantics of the CLIs. Next, we will explain each step of our IE algorithm.

3.4.1 Data Pre-Processing

The first step of our algorithm consists in applying shallow Natural Language Processing (NLP) tools over the CLI data. Overall, Data Pre-processing includes the following resources, namely, (i) Part-Of-Speech (POS) Tagger, (ii) Tokenizer and (iii) Stemmer. The **POS Tagger** resource allows the identification of verb phrases from the CLI. Notice that in-depth POS analysis is far from being required as typical

CLIs lack of grammar rules and verbosity. Moreover, the number of configuration actions (i.e., operations) are finite and common across multi-vendor platforms, e.g., set, delete, add, merge, show, load, reset, etc. In our implementation, we have used the Stanford Log-linear POS Tagger [30] for this purpose. The next resource to be applied is the **tokenizer** which separates data into tokens for further processing. We have used the tokenization tool in the Apache OpenNLP library [31]. Notice that commands are typically expressed with single short suggestive keywords as to ease manual configuration. Nevertheless, in the context of CLIs, we can often find the use of hyphenated words to improve the expressiveness of a command—e.g., "source-port", "source-class", etc. In an effort to preserve the semantics of the CLI—in the context of our solution—hyphenated words account as single tokens. Moreover, we remove stop words, i.e., we filter irrelevant words from the CLI data, e.g., articles or prepositions. It is important to highlight the fact that, the number of stop words in our domain is not particularly significant, as commands and variables are single keywords and help descriptors are generally short and concise phrases with poor grammar. However, we have developed a custom list of stop words to avoid returning or processing unnecessary information. The final resource to complete the step of Data Pre-Processing is the **Stemmer**, which allows to reduce inflectional forms of a word to its common base form. We used the Stanford NLP stemmer (CoreNLP) [30]—which is based on the Porter stemming algorithm. Herein—as done by many search engines—words with the same stem are treated as synonyms. This will improve the performance of our system by increasing the probability when performing lexical matching.

3.4.2 Lexical Matching

The next step in the algorithm is the identification of concepts with respect to components in the domain-ontology by means of *lexical matching*. This strategy fits with the general notion that—overall, and despite CLIs heterogeneity—concepts must converge to well-known technical terminologies, otherwise, it becomes increasingly complex to achieve interoperability and moreover, stay competitive in an industry led by standards. Consider for instance, a device vendor using custom terminologies to refer to standard concepts of the networking domain, e.g., an IP Address or a networking protocol (MPLS, DHCP, etc.). In such a context, the interpretation of CLIs becomes overly intricate, unless vendors provide mappings to standard terminologies. In light of all this, the notion of CLIs having to rely on technical (standard) terms—at least those likely to be referents in the field—makes lexical matching a feasible strategy to identify key concepts of the domain. Nevertheless, it is clear that in a field full of terminologies and ever-changing technologies, there is still space for syntactic and semantic ambiguity, as typically, vendors use different terms to refer to the same concepts or on the contrary, use the same terms to refer to different concepts.

If we consider the graph-based representation of the ontology (cf., Fig. 5), the lexical matching stage results in the activation or highlight of nodes and links of

the Semantic Graph G. Notice that we admit both, partial and exact matching. If an exact match is found for a given term, partial matches are discarded. Moreover, if more than one partial match is found for a term and these are taxonomically related, we hold the Least Common Subsumer (LCS), i.e., the most concrete taxonomic ancestor. In other words, we generalize in the absence of information. Notice that there are cases for which candidate concepts do not have a LCS. In these cases, concepts are considered disjoint, i.e., only one can accurately define the semantics of the given CLI term. It is important to realize that even if concepts are not identified by lexical matching, mainly because of the use of custom or dissimilar terminologies, the Semantic Analysis can identify relevant concepts by inference. Therefore, we do not make limited use of the ontology—such as names of classes—moreover, we use the ontological structure to enhance our assessment.

3.4.3 Semantic Analysis: Clustering and Inference

In this stage activated resources are grouped into semantic clusters (cf., Fig. 7). We form clusters between *directly* connected resources of adjacent levels (e.g., $\{C_2, C_3\} \in G$). If an activated resource is disconnected to other active concepts

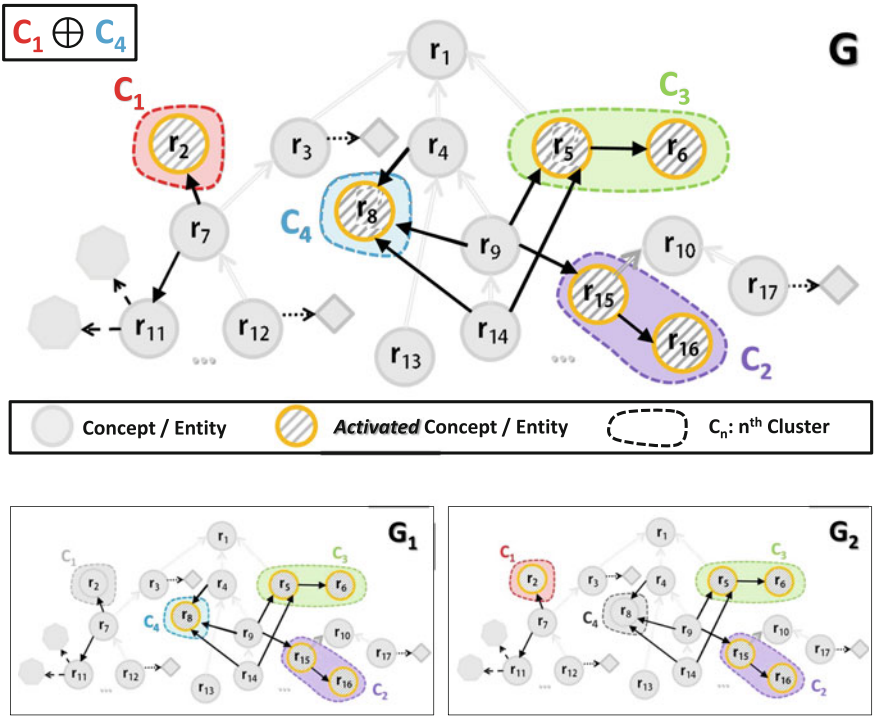

Fig. 7 General Notion of the clustering stage

in the graph, it uniquely forms a cluster (e.g., $\{C_1, C_4\} \in G$). Notice that the notion of nodes being part of fully interconnected clusters is based on the premise that commands are arranged in the hierarchy by semantic association—i.e., commands become more specific down in the tree structure. Accordingly, the concepts that derive from commands and variables in contiguous levels are expected to be semantically related to a certain extent—directly or not. Ideally, activated resources would form part of a single cluster (i.e., interconnected concepts), however, the degree to which entities are actually related will also depend on the granularity of the CLI—which varies for every vendor. For this reason, clustering is not sufficient and we require other means to measure (i.e., quantify) the degree to which concepts are semantically related, this is part of the *relatedness computation* stage, which will be described later in this section. Notice that we restrict clusters to non-disjoint nodes. This means that concepts that have been triggered by the same activation keywords necessarily belong to different clusters—even if they are directly connected—as they are (from a lexical perspective) equally likely candidates for the same set of concepts. For instance, consider in Fig. 7 resources r_2 and r_8 to be candidate concepts for the same set of keywords. Accordingly, clusters C_1 and C_4 are disjoint, as only one can fairly represent the semantics of the given term(s) and thus, each belong to a different subgraph—i.e., valid combinations of non-disjoint clusters (e.g., G_1 and G_2). In further stages, semantic relatedness is computed over each subgraph in an effort to promote closest nodes (i.e., those with higher density) as the most suitable concepts for defining a given configuration statement.

Furthermore, we perform *semantic inference* as a means to derive knowledge that is not explicitly expressed in the CLI. To this end, we exploit the ontological structure and reason over the facts and axioms formally defined in the router/switch configuration domain ontology. The inference stage has a two-fold purpose. First, to discover potential concepts that were not identified in previous stages, either because of (i) the use of very dissimilar terminologies—i.e., use of a vocabulary which is not well aligned to the domain lexicon—or (ii) because of the granularity of the hierarchy. For instance, for less granular hierarchies, knowledge is most likely to be implicit and thus, concepts can fail to be identified. To illustrate the inference stage, consider the example shown in Fig. 8. If we identify from the CLI the concepts "$\langle administrative-distance \rangle$" ($r_8$), "$\langle destination-prefix \rangle$" ($r_5$) and "$\langle bandwidth \rangle$" ($r_{15}$), we can infer from the equivalent axioms of the ontology that most likely we are referring to the concept "$\langle route \rangle$" (r_9), for which these 3 concepts are exclusive properties. Based on this. we can further activate the inferred resource (r_9) and build the semantics of the given configuration statement. The second purpose of this stage is to generalize or specify already active concepts by taking into account contextual information. Consider the following example, if the concept "$\langle routing-protocol \rangle$" has been identified for a given level of the hierarchy and we then identify the concept "$\langle OSPF-area \rangle$", we can infer that we are referring to the OSPF protocol—which is both a routing-protocol and exclusively related to the attribute "$\langle OSPF-area \rangle$". In any of both cases, if this stage results in the activation of a node by inference (e.g., r_9 in Fig. 8) we perform clustering once more to group nodes by direct association (e.g., C_2' in Fig. 8).

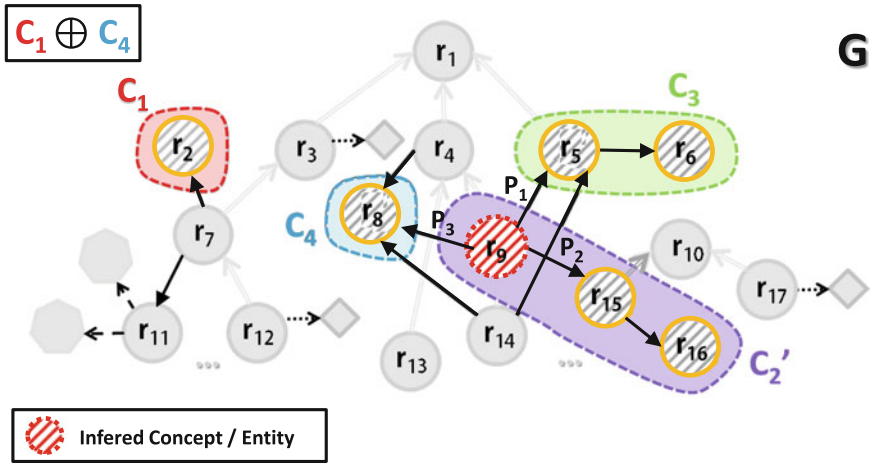

Fig. 8 General notion of the inference stage

3.4.4 Semantic Analysis: Relatedness Computation

The rationale of computing semantic relatedness for the set of candidate concepts is to promote closest nodes, based on the premise that, CLIs are arranged by association, accordingly, successive commands in the hierarchy are expected to represent highly interrelated concepts of the domain. Overall, relatedness will contribute to the identification of potential outliers (command sense disambiguation) and information extraction. In the context of our solution, we define ***relatedness*** ($\mathcal{R}$) as a function that computes the strength of semantic association between a set of clusters $\{\mathcal{C}_k\} \in G_k$. Notice that, in contrast to state of the art approaches, $\mathcal{R}$ is not restricted to a given "pair" of concepts, but instead, extends to reflect the proximity in meaning of a "set" of concepts. Typical existing measures of relatedness based on ontologies exploit only taxonomic relations, accordingly, they are more a measure of similarity rather than relatedness. In light of this, our measure $\mathcal{R}$ is computed by interpreting the paths between clusters, based on both taxonomic and non-taxonomic relations.

Definition 2 The Semantic Relatedness ($\mathcal{R}$) between the set of concepts $c_k \in G_k$ is defined as the product of the density (d) and the Information Content (I) (cf., Eq. 1). As the majority of approaches we consider semantic relatedness symmetric.

$$\mathcal{R}(G_k) = d(G_k) \cdot \mathcal{I}(G_k) \tag{1}$$

Next, we will formally define the components for the computation of semantic relatedness, namely, **density** (d) and **Information Content** (I).

Definition 3 The graph interconnectivity which is captured under the notion of density (d) (cf., Eq. 2) is computed as the relation between the number of active edges along the shortest path between any pair of clusters in graph G_k, and the total number of edges in those shortest paths (i.e., path length).

The number of *active* edges is computed as the total number of activated entities along the path ($\mathcal{A}(\mathcal{P})-1$). Recall that clusters are composed of activated entities only, thus, source and destination of any given path P are always activated entities, hence the number of *active* edges is ($\mathcal{A}(\mathcal{P}) - 1$). Notice that, similarly to the approach of Tsatsaronis et al. in [32] a *path* ($\mathcal{P}$) can be a combination of different types of edges—including both taxonomic and non-taxonomic. However, we consider that relations are equally weighted.

$$d(G_k) = \frac{\sum_{i=1}^{|\mathcal{C}_k|-1} \sum_{j=i+1}^{|\mathcal{C}_k|} [\mathcal{A}(\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j)) - 1]}{\sum_{i=1}^{|\mathcal{C}_k|-1} \sum_{j=i+1}^{|\mathcal{C}_k|} \mathcal{H}(\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j))} \leq 1 \tag{2}$$

Next, we introduce the notations to our relatedness measure $\mathcal{R}$.

$\mathcal{A}(\mathcal{P})$ a function that returns the number of activated entities along a given *path* P.
$\mathcal{H}(\mathcal{P})$ a function that returns the length of a given *path* P.
$\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j)$ a function that returns the *shortest path* between a given pair of clusters.

Definition 4 The Information Content (I) is a measure of the knowledge enclosed by a cluster (cf. Eq. 3).

In formulae, we use t_k^{il} to denote the total number of terms that trigger the activation of a domain entity $c_k^{il} \in \mathcal{C}_k^i$ in the semantic graph G_k. Moreover, m_k^{il} is a matching factor which represents the probability of an entity of being the asserted concept with respect to the total number of entities identified for the same set of terms. This coefficient takes the maximum value of "1" whenever a domain entity has been identified by perfect lexical match, or $\frac{1}{(e+1)}$ in all other cases, with e the total number of entities also identified for the same terms. Finally, the o_k^{il} factor is calculated by counting the frequency of occurrence of an entity for all levels, over the total number of occurrences of its exclusive disjoint candidate entities.

$$\mathcal{I}(G_k) = \sum_{i=1}^{|\mathcal{C}_k|} \sum_{l=1}^{|c_k^{il}|} t_k^{il} . m_k^{il} . o_k^{il} \tag{3}$$

After computing semantic relatedness for all set of candidate concepts ($\forall G_k$), we select the set with maximum relatedness (cf., Eq. 4) as the most suitable set of

resources representative of the CLIs knowledge. If semantic relatedness is the same for more than one set of clusters we compute outdegree as tiebreak.

$$\max_{k} \quad \mathcal{R}(G_k) \tag{4}$$

3.4.5 Decision Maker and Ontology Population

In this stage we combine the information of identified network resources and verb phrases to semantically derive the operation(s) that a command or set of commands represent—i.e., with respect to components defined within the operation layer of our domain ontology. In other words, we reason over the ontology to determine the set of operations that most likely represent the semantics of a given configuration statement. If a single atomic operation does not fully-define the extracted information (i.e., network resources and verb phrases at the same time), we build semantic connections between atomic operations, which furthermore represent the way in which commands are organized in the hierarchy. Consider for instance the example shown in Fig. 9, where the set of identified resources are interface (L2), if-name (L3), if-description (L4) and verb phrases ***set*** (L1). The nature of our decision maker is to select on a level-basis the most concrete atomic operation and finally build a semantic flow among them. Therefore, in our example the output of this stage will be the concatenation of atomic operations OP1–OP3. Under this scenario, whenever a user requests the semantic operation OP3, we will be able to automatically build the sequence of commands by following the path of semantic links. Ontology population is the actual instantiation of commands into the semantic categories. In the case that a single

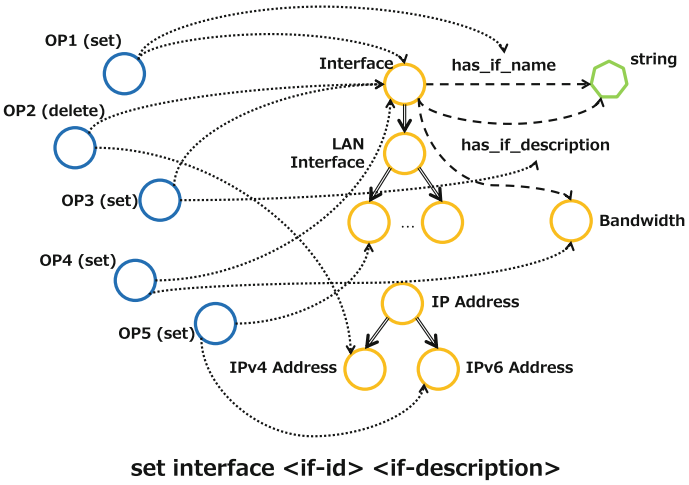

Fig. 9 Example decision maker stage

atomic operation is represented by several commands or combination of commmands and variables we include ordering information for adequately retrieving configuration statements in the exact order.

4 Evaluation

Regarding implementation of the OBIE System, all modules were developed in Java, as it ideally suits the needs of integration with already available libraries for both, OWL Ontology Management (OWL API [33]) and Natural Language text Processing (NLP Stanford [30]).

In order to assess the performance of our system we carried out end-to-end experiments over the configuration spaces of well-known routing devices—both proprietary and open-source. More specifically, we performed evaluations over a Juniper Router (Model M7i - JUNOS 10.4.R13.4), a Cisco Router (Cisco IOS Release Version 12.4(16a) FC2) and the Quagga routing software (Release 0.99.21). To this end, we manually limited each CLI to a set of commands which semantically represent the operations most commonly performed by network administrators across their networks. Notice that these operations were determined by thoroughly analyzing real configuration files of actual core routers. We selected a total of 250 operations, which individually mapped to different sets of commands for every configuration space.

In the literature, ***precision*** (P), ***recall*** (R) and ***F-measure*** (F) have become the absolute metrics for the performance evaluation of traditional Information Extraction Systems [34]. Overall, these measures reveal the ability of a system to identify information from text. Most specifically, *precision* is a measure of correctness—i.e., a percentage of correct instances with respect to the total number of identified items. *Recall* is a measure of completeness—i.e., a percentage of correct instances with respect to the total number of expected identified instances. Finally, *F-measure* is the geometric mean of both, precision and recall. However, when Information Extraction is done with respect to components of an ontology, traditional metrics are insufficient [35]. This is mainly because ontological classification can more or less have different degrees of correctness. For instance, consider classifying the instance 'FE80::0202:B3FF:FE1E:8329' as an $\langle IP\_Address \rangle$—rather than $\langle IPv6\_Address \rangle$. In this case, classification is clearly less wrong than if classified as an instance of the $\langle MAC\_Address \rangle$ class. Accordingly, we require of other mechanisms which provide the means to quantify the degree of correctness of instantiations with respect to the ontological structure. One of the most common used metrics for the evaluation of OBIE systems is the Balanced Distance Metric (BDM) proposed by authors in [35]. In order to compute this cost-based component we used the BDM Computation Resource plug-in available for the GATE platform [36]. This tool outputs a file with the BDM scores for all pair of classes in the ontology. Notice that BDM by itself, does not provide the means to evaluate the system's overall

Table 1 Performance results of our OBIE process (Augmented and Traditional)

	Augmented						Traditional					
	Stage 1 (%)			Stage 2 (%)			Stage 1 (%)			Stage 2 (%)		
	AP	AR	AF	AP	AR	AF	P	R	F	P	R	F
Juniper	92.7	93.8	93.3	91.9	94.2	93.0	89	90	90	74	61	67
Cisco	89.0	89.2	89.1	90.0	89.9	90.0	84	86	85	85	80	82
Quagga	88.9	82.9	85.8	91.6	85.8	88.6	88	82	85	84	63	72
Overall	90.9	88.4	89.3	**91.3**	**90.0**	**90.6**	87	86	86	80	68	73

performance. For this reason, we computed augmented versions of traditional precision, recall and F-measure as defined by the same authors in [37].

In order to compute performance metrics, we manually built a gold standard for each configuration space, i.e., a benchmark data set against which to compare the system's output. The gold standard was created by a group of networking experts as a fully-compliant reference set. We created a gold standard for each stage of our instantiation methodology, namely, one for the resource identification stage (**Stage 1**) and another for the operation identification stage (**Stage 2**). Results for **Stage 1** are a measure of the ability of our system to extract information of networking resources from the CLI, while results for **Stage 2** measure the ability of our system to derive and infer the configuration operations that commands in the CLI actually represent. Table 1 depicts the final percentage values for Augmented Precision (AP), Recall (AR) and F1-Measure (AF1), as well as traditional values (P, R and F1) for both stages.

Overall, our system's performance is around 92 % AP and 90 % AR, which is nearly human-level performance. When comparing these results to traditional metrics of P, R and F1-Measure (cf., Table 1) we can confirm that our algorithm has the ability to semantically approximate concepts in the cases when CLI information is not sufficient. Moreover, an interesting finding was to discover an added capability of our system in determining the correspondence and consistency of CLI knowledge with the literature. In other words, results show that when CLI knowledge is inconsistent with networking facts, our system approximates (either generalizing or specifying) to the nearest concept wherein these assertions are true. For instance, if a protocol is categorized in the CLI hierarchy under a subtype which according to the literature is not true, our system approximates to the nearest concept for which such statement is true. Notice that with 91 % AP achieved in a fully automated matter, near 9 % of commands would not be adequately instantiated within their corresponding semantic categories. Nevertheless, considering that significant tedious work of mapping would be done and that we can have good suggestions produced by the system, it is reasonable to think of a human in the final loop of deployment. Indeed, a network administrator would have a significantly easier time verifying the remaining 8 % of commands than navigating and semantically interpreting the entire hierarchy.

Moreover, in an effort to assess the suitability of our design decisions, we developed potential improved versions of our instantiation algorithm by taking into account

Table 2 Performance results of our OBIE process + ontology depth

	Stage 1 (%)			Stage 2 (%)		
	AP	AR	AF	AP	AR	AF
Juniper	92.7	93.8	93.3	91.9	94.2	93.0
Cisco	89.1	89.2	89.1	90.0	89.8	90.0
Quagga	88.9	83.0	85.9	91.8	85.6	88.6
Overall	90.3	88.6	89.4	**91.2**	**89.9**	**90.5**

other variables in the decision process. The first path of enhancement that we performed was considering the ontology *depth* as a variable to the computation of semantic relatedness. Ontological depth is one of many variables considered in several state-of-the-art semantic relatedness metrics, which is based on the notion that deeper nodes in the taxonomy—i.e., more specific concepts—have stronger semantic association than higher (generic) concepts. In light of this, we included a component that considers the relative depth of clusters (denoted as $\mathcal{D}$) (cf., Eqs. 5 and 6), which is a relation of the sum of actual depths of a pair of clusters over the sum of the maximum depths. Table 2 depicts the performance results when considering ontology depth. Notice that when compared to the original version of our algorithm (cf., Table 1) performance results are not affected by the inclusion of a new variable to our semantic relatedness measure. We believe that because of the generalization feature performed during lexical matching and the subsequent reasoning in the inference stage, concepts within a branch of the ontology (i.e., taxonomically related concepts) have already been pruned on this basis. As such, by the time we perform semantic relatedness computation pairs of disjoint candidate concepts on a same taxonomy branch do not exist, therefore, ontology depth has no actual weight in the final scores. For this reason, we performed new experiments by removing the generalization and specialization features of our algorithm in order to evaluate the performance of our system considering taxonomic concept pruning based on semantic relatedness measures rather than on our initial approach. The results of this experiment are shown in Table 3. Herein, *Original** refers to our initial algorithm without the generalization/specialization feature, while *Depth** is the later with considerations of ontology depth in semantic relatedness computation. Observe that overall the performance of the system for both scenarios is lower than the one obtained for the original version of our algorithm (cf., Table 1) and that considering the ontology depth as the absolute criteria for taxonomic pruning of concepts does not significantly contribute to enhancing the performance of the system. We strongly believe that ontology depth does not compensate to the performance of our generalization feature, basically because our decisions for pruning concepts in the hierarchy rely on the lexical knowledge obtained from the CLI in combination with ontological structure, while depth only takes a decision based on the ontological structure. In light of all this, we can definitely conclude on the suitability of our initial design premises as the system achieves the highest performance values for this scenario.

Table 3 Performance results—no generalization/specialization + depth

	Original *			Depth *		
	Stage 2 (%)			Stage 2 (%)		
	AP	AR	AF	AP	AR	AF
Juniper	88.2	93.6	90.8	87.1	93.2	90.0
Cisco	87.1	84.0	85.5	87.2	83.4	85.2
Quagga	91.8	84.1	87.7	91.4	84.0	87.5
Overall	**88.9**	**87.0**	**88.1**	**88.4**	**86.7**	**87.3**

$$d'(G_k) = d(G_k) \cdot \sum_{i=1}^{|\mathcal{C}_k|-1} \sum_{j=i+1}^{|\mathcal{C}_k|} \mathcal{D}(\mathcal{C}_i, \mathcal{C}_j) \tag{5}$$

$$\mathcal{D}(\mathcal{C}_i, \mathcal{C}_j) = \frac{depth(\mathcal{C}_i) + depth(\mathcal{C}_j)}{max\_depth(\mathcal{C}_i) + max\_depth(\mathcal{C}_j)} \tag{6}$$

depth($\mathcal{C}_n$) a function that returns the depth of a cluster as the depth of the deepest node in the cluster.

max_depth($\mathcal{C}_n$) a function that returns the maximum depth of a cluster computed as the deepest concept among all branches of the cluster.

5 Conclusions

In this chapter, we have described a tool which can ease the complex task of network administrators in the configuration of network devices in the context of heterogeneous (i.e., multi-vendor) networks. We have shown the potential of Information Extraction and other semantic technologies to exploit the knowledge natively provided by vendors in the form of natural language in their CLIs. Most specifically, we have presented an Ontology-Based Information Extraction (OBIE) System from the Command-Line Interface (CLI) of network devices and thus, developed an ontology for the network device configuration domain and an IE methodology for the semantic categorization of commands. Furthermore, we have concluded on the suitability of our design decisions for which we achieved a maximum performance of 91 % precision and 90 % recall. The advantage of this approach is that network administrators will no longer have to hold off the top of their heads hundreds of commands nor read a large number of manuals for the actual configuration of network devices. Moreover, a solution of this nature has the potential to enable autonomic networking, by assisting third-party applications in the execution of network device (re)configuration.

References

1. Subramanian, M., Gonsalves, T.A., Rani, N.U.: Network Management: Principles and Practice. Dorling Kindersley, Noida (2010)
2. Case, J., Fedor, M., Schoffstall, M., Davin, J.: Simple network management protocol (SNMP). RFC 1157, Internet Engineering Task Force, pp. 1–36. May 1990
3. Caldwell, D., Gilbert, A., Gottlieb, J., Greenberg, A., Hjalmtysson, G., Rexford, J.: The Cutting EDGE of IP router configuration. SIGCOMM Comput. Commun. Rev. **34**(1), 21–26 (2004)
4. Lee, S., Wong, T., Kim, H.S.: To automate or not to automate: on the complexity of network configuration. In: IEEE International Conference on Communications, 2008, ICC '08, pp. 5726–5731. May 2008
5. Chappell, C.: The business case for NETCONF/YANG in network devices. White Paper, 2013
6. Enns, R., Bjorklund, M., Schoenwaelder, J., Bierman, A.: Network configuration protocol (NETCONF). In: RFC 6241, IETF, June 2011. http://tools.ietf.org/html/rfc6241
7. Cui, H., Zhang, B., Li, G., Gao, X., Li, Y.: Contrast analysis of NETCONF modeling languages: XML Schema, Relax NG and YANG. In: International Conference on Communication Software and Networks, 2009, ICCSN '09, pp. 322–326, 2009
8. Chisholm, S., Clemm, A., Tjong, J.: Using XML Schema to define NETCONF Content. Internet-Draft, Network Working Group (2008)
9. Johansson, L.: NETCONF Configuration Data Modeling Using OWL. Internet-draft, Internet Engineering Task Force (IETF) (2008)
10. Bjorklund, M.: YANG—A data modeling language for the network configuration protocol (NETCONF). RFC 6020, IETF, Oct 2010
11. Nadeau, T., Gray, K.: SDN: software defined networks an authoritative review of network programmability technologies. O'Reilly Media, 2013
12. Kim, H., Feamster, N.: Improving network management with software defined networking. Commun. Mag. IEEE **51**(2), 114–119 (2013)
13. Open Networking Foundation. OpenFlow Switch Specification. Version 1.1.0. http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf
14. Grassi, M., Nucci, M., Piazza, F.: Towards a semantically-enabled holistic vision for energy optimisation in smart home environments. In: IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 299–304, April 2011
15. Nucci, M., Grassi, M., Piazza, F.: Ontology-based device configuration and management for smart homes. Neural Nets and Surroundings. Smart Innovation, Systems and Technologies, vol. 19, pp. 301–310. Springer, Heidelberg 2013
16. Wicaksono, H., Schubert, V., Rogalski, S., Laydi, Y.A., Ovtcharova, J.: Ontology-driven requirements elicitation in product configuration systems. In: Hoda A., ElMaraghy (eds.) Enabling Manufacturing Competitiveness and Econoomic Sustainability, pp. 63–67. Springer, Heidelberg 2012
17. Colace, F., De Santo, M., Napoletano, P.: Product configurator: an ontological approach. In: Ninth International Conference on Intelligent Systems Design and Applications, 2009, ISDA '09, pp. 908–912, Nov 2009
18. Dong, M., Yang, D., Su, L.: Ontology-based service product configuration system modeling and development. Expert Syst. Appl. **38**(9), 11770–11786 (2011)
19. Yang, D., Miao, R., Wu, H., Zhou, Y.: Product configuration knowledge modeling using ontology web language. Expert Syst. Appl. **36**(3, Part 1):4399–4411 (2009)
20. Yang, D., Dong, M., Miao, R.: Development of a product configuration system with an ontology-based approach. Comput.-Aided Des. **40**(8), 863–878 (2008)
21. López De Vergara, J.E., Guerrero, A., Villagrá, V.A., Berrocal, J.: Ontology-based network management: study cases and lessons learned. J. Netw. Syst. Manage. **17**(3), 234–254 (2009)
22. López de Vergara, J. E., Villagrá, V. A., Fadón, C., González, J. M., Lozano, J. A., Álvarez Campana, M.: An autonomic approach to offer services in OSGi-based home gateways. Comput. Commun. **31**(13):3049–3058 (2008) Special Issue: Self-organization and self-management in communications as applied to autonomic networks

23. Xu H., Xiao, D.: A common ontology-based intelligent configuration management model for IP network devices. In: First International Conference on Innovative Computing, Information and Control, 2006, ICICIC '06, vol. 1, pp. 385–388. Aug 2006
24. Xu, H., Xiao, D.: Applying semantic web services to automate network management. In: 2nd IEEE Conference on Industrial Electronics and Applications, 2007, ICIEA, pp. 461–466. May 2007
25. Wong, A.K.Y., Ray, P., Parameswaran, N., Strassner, J.: Ontology mapping for the interoperability problem in network management. IEEE J. Sel. Areas Commun. **23**(10), 2058–2068 (2005)
26. Wimalasuriya, D.C., Dou, D.: Ontology-based information extraction: an introduction and a survey of current approaches. J. Inf. Sci. **36**(3), 306–323 (2010)
27. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data Knowl. Eng. **25**(12), 161–197 (1998)
28. ONDC: Ontology for network device configuration. http://www.netit.upc.edu/ondc, 2014
29. OWL web ontology language API. http://owlapi.sourceforge.net/
30. Stanford CoreNLP. http://nlp.stanford.edu/software/corenlp.shtml
31. Apache OpenNLP. https://opennlp.apache.org/
32. Tsatsaronis, G., Varlamis, I., Vazirgiannis, M.: Text relatedness based on a word thesaurus. J. Artif. Int. Res. **37**(1), 1–40 (2010)
33. Horridge, M., Bechhofer, S.: The OWL API: a Java API for OWL ontologies. Semant. web **2**(1), 11–21 (2011)
34. Esuli, A., Sebastiani, F.: Evaluating information extraction. In: Agosti, Maristella, Ferro, Nicola, Peters, Carol, de Rijke, Maarten, Smeaton, Alan (eds.) Multilingual and Multimodal Information Access Evaluation. Lecture Notes in Computer Science, vol. 6360, pp. 100–111. Springer, Berlin Heidelberg (2010)
35. Maynard, D.: Metrics for evaluation of ontology-based information. In: Proceedings of the WWW 2006 Workshop on Evaluation of Ontologies for the Web, 2006
36. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I.,Gorrell, G. , Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.,Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE, vol.6. 2011
37. Maynard, D., Peters, W., Li, Y.: Evaluating evaluation metrics for ontology-based applications: Infinite reflection. In: LREC, 2008

MKMSIS: A Multi-agent Knowledge Management System for Industrial Sustainability

Virgilio López-Morales, Yacine Ouzrout, Thitiya Manakitsirisuthi and Abdelaziz Bouras

Abstract Industrial Sustainability in most companies face several problems when a firm tries to promote a greener image while focusing on maintaining profitability in a long-term approach. This is a necessary strategic approach, as customers are becoming more committed to buying products that consider environmental concerns and additional regulations need to be followed. However, traditional marketing and distribution methods, management tools and the homogenization of requirements fail to fully integrate the implication of environmental regulations into their processes. In this scenario, a system for supporting the integration of environmental concerns, endogenous and exogenous regulations, and market trends would be very well received. This is a complex task to be realized by just one system, even when a firm's various departments have an efficient networked communication system. A Distributed Decision Making System (DDMS) could be a useful approach. This paper introduces a multi agent network for collaborative knowledge management. Since regulations and environmental issues are at the core of those processes and influence the final products, several sustainability phases can be addressed within this network, from the design to the marketing and distribution stages. The aim of our system is to deal with the management of these different phases in a collaborative platform by considering the sustainability knowledge related to several regulations.

V. López-Morales (✉)
CITIS, Universidad Autónoma del Estado de Hidalgo,
Carr. Pachuca-Tulancingo km. 4.5, Hgo, 42184 Pachuca, Mexico
e-mail: virgilio@uaeh.edu.mx

Y. Ouzrout · T. Manakitsirisuthi
DISP-Laboratory, Université Lumière Lyon 2,
160 Bd de L'Université, 69676 Bron, France
e-mail: yacine.ouzrout@univ-lyon2.fr

T. Manakitsirisuthi
e-mail: thitiya.manakitsirisuthi@univ-lyon2.fr

A. Bouras
Computer Science and Engineering Department,
Qatar University, College of Engineering, 2713, Doha, Qatar
e-mail: abdelaziz.bouras@qu.edu.qa

Y. Laalaoui and N. Bouguila (eds.), *Artificial Intelligence Applications in Information and Communication Technologies*,
Studies in Computational Intelligence 607, DOI 10.1007/978-3-319-19833-0_9

Keywords Industrial sustainability · Multi-agent system · Knowledge management · Expert systems · Distributed decision making

1 Introduction

When a project's marketing is launched, Product, Price, Place and Promotion are important issues to bring a brand or product to the market and sell it with the best return of investment. Our current global economy is evolving into an interconnected world which demands inter-organizational Business to Business (B2B) or Business to Customer (B2C) information sharing. In this framework, system analysts, database developers, statisticians, graphic designers and customers service professionals yield a heterogeneous scenario [1] that must be organized efficiently. In order to construct a support structure for collaborative decision making, including information from final customers, an efficient knowledge-based system is paramount. As highlighted in [2], more companies are paying more attention to customer relationship management and their direct marketing strategies to reduce costs and increase profitability by acquiring information directly from those data sources. Information Technologies (IT) are playing a major role in this interaction; inter-organizational or customer-specific information can be collected and analyzed by an automated system to quickly improve the product lifecycle and comply with market trends.

However, as mentioned in [3], to integrate a collaborative system in a Department Network[1] (DN) is a complex and critical task that requires a collaborative framework and a decision-making unit among departments to solve multiple issues, such as technical problems, after sale service, environmental regulations, etc. Consequently, innovation and availability need to be well-balanced to meet the expectations of long-term buyers. Sometimes, customers do not want a change in a product, but a modification or a price increase needs to be made to comply with environmental regulations. A Distributed Decision Making System (DDMS) would thus be useful for a number of potential concerns, including:

- How will a modification be communicated?
- Where is the value to be found?
- What questions/complaints will arise?
- Does our marketing audit integrate the implications of actions into the marketing process?
- What kind of changes could, or should, be performed in the manufacturing supply chain of a corresponding department(s)? and
- What is the cost/benefit for the company and/or the customer?

A high degree of collaboration and transparency is helpful for solving critical technical aspects in the Department Network. Even if a particular firm does not have a large

[1]Composed by heterogeneous Departments or Users, as for instance service providers, designers, decision makers, implementers, dispatchers and executive managers.

number of departments, the increasing concerns about the environment and energy conservation will lead it to rethink their market position, reformulate their strategy and re-engineer their business processes [4]. Customers are becoming more and more committed to making their purchasing decisions based on products that are in compliance with health and/or environmental quality standards, as well as their use of recycled materials (not to mention fair labor practices). These trends have led many companies to utilize Reverse Logistics as a key strategy for product handling and the disposal of returned products from their inter-organization or customers. Reverse Logistics can help to recycle resources and to better manage waste materials, with a reasonable cost, from customers back to the production point [5]. Environmental issues, regulation standards and customers' advice can help engineers in their decision making during the design, manufacturing and recovery phases. Organizations could effectively minimize their waste and improve their environmental performances. Managing returned products efficiently not only reduces the amount of waste generated, it also encourages a company to redesign their processes and products to avoid the use of hazardous substances and to promote better manufacturing practices [6].

Consequently, in a new strategic approach it will be necessary to adopt or to develop a new eco-mindset, and this mindset must be integrated across all organizational areas and activities. This approach can help a DN to develop new products and long-term profits, as well as reduce negative environmental impacts [7]. Furthermore, as suggested in [8], 70 % of a product's environmental harms are built into the product along with the associated production process. Firms need to incorporate environmental standards into products and processes at the initial stages of new product development. The objective is to use lifecycle analysis to evaluate a product's ecological impact at each production stage, making it possible to identify alternative methods of designing or producing goods while lowering production costs.

In the past decade, Efficient Product Lifecycle Management (PLM) strategies have successfully been applied to product information and knowledge management throughout the lifecycle of products. It has specifically taken advantage of information sharing to decrease the time-frame for product development and production processes [9]. Moreover, PLM strategies have the potential to enhance environmental sustainability; developing affordable products that make the best use of the available resources to satisfy the needs of large markets [10]. Current PLM systems deal with the integration of information regarding environmental regulations and industrial standards in the products stages, such as product design, manufacturing and recovery.

In line with these efforts, the key concept in our work is that by integrating a PLM system with a DDMS containing several knowledge bases and a set of expert systems in a distributed collaborative platform, it could be simpler and faster to have a trend line going through marketing, industrial policies, design, process plans, etc., even including distribution strategies. This system could achieve a transparent marketing policy while maintaining a high sustainability index. The set of expert systems will contain complete, updated information on the environmental regulations and the main problems collected from an inter-organizational market and final user returned

products. Furthermore, to accommodate the changing interests of final customers, DDMS-PLM will help managers to capitalize on their product strategies and create competitive advantages, for instance, improving the management of sustainable benefits and green marketing to meet customers' new needs. By using a DDMS-PLM-based system, a positive impact can be realized with an efficient management of products, as illustrated by the Ford Motor Company when they quickly adapted their assembly line configuration to satisfy changing user requirements [11].

Finally, there are currently some large companies which have put into practice a strategy that addresses environmental issues across all disciplines right down to the way they manage their supply chain [12, 13]. When enterprizes are mainly concerned about marketing and promotion before sales, they may ignore the roles and responsibilities of green marketing after sales, even though the after-sales reputation can further improve a company's reputation and customer loyalty [4]. Consequently, the integration of a DDMS-PLM system designed to enhance, qualify and quantify regulations and their corresponding implications on the manufacturing supply chain has not been yet addressed. A complete integration of the information on environmental concerns, regulations, market trends and a firm's main objectives could be achieved by using a collaborative multi-expert platform in each area (manufacturing distribution, marketing, etc.).

The paper is organized as follows: The background of the main technologies necessary for the development of MKMSIS modules are given in Sect. 2, followed by a presentation of the core module of MKMSIS, in Sect. 3. The agent's roles within the inference engine are also explained in Sect. 3. Next, in Sect. 4, Eco-KMS module's communication tasks are described through a model diagram. A case study for a Mobile Phone Process is analyzed in Sect. 5. Sections 6 and 7, offer some discussions and conclusions about the main advantages of the system and the direction of future work.

2 Information Technologies for Green Strategies and Industrial Sustainability

Despite the increasing salience of being greener and sustainable, there is no holistic framework to guide the construction of green industrial brands [14], a framework which would require not only green operations, but also green marketing. When considering green marketing, many people tend to focus incorrectly on specific individual activities. Our PLMS (Product Lifecycle Management System) focuses on the whole lifecycle, in order to foster collaborative processes. In the same manner, and as suggested in [15], green-marketing activities can occur at three levels in a firm: strategic, quasi-strategic, and tactical. The goal of our approach, based on information technology and decision support systems, is to help to change corporate philosophy and to facilitate a substantial change in manufacturing and business practices.

2.1 Multi Agent Systems in Green and Industrial Sustainability

Multi Agent Systems (MAS) are an evolution of methods that appeared in distributed artificial intelligence and of contributions focusing on logical extensions of rational behavior [16], designed to capture human expertise in narrow domains to be stored on re-usable and sharable knowledge repositories. By considering multiple cognitive entities acting in communities [17], and with the evolution of network-based computing technology the agent paradigm of computing has been based on the Internet, mobile computing and the ubiquity of computing, as well as novel, human-oriented software engineering methodologies [18–20]. This is the essence of a MAS framework which can facilitate the development of a collaborative platform for the most efficient management of resources and task allocations.

In order to have feedback from the manufacturing modules, a firms' targets, market behavior, and customers, as well as to assess the impact of environmental regulations, a complete integration of the different modules must be done, right from the conception of the entire communication design of the Department Network. The concept of reverse logistics has already been analyzed in [21], wherein some levels of the recovery processes are stated:

1. cleaning and repair;
2. product remanufacturing;
3. refurbishing;
4. cannibalization;
5. recycling of packaging materials; and
6. energy recovery by incineration, added to the supply chain [21].

Enterprize Resource Planning (ERP) and Advanced Planning Systems (APS) provide useful advice that can be implemented in a supply chain, which must be designed to support the return flows based on information and communication technologies for reverse logistics. These systems (ERP and APS) can contribute to the implementation of reverse logistics in

- manufacturing and recovery process;
- distribution and collection;
- supplier collaboration; and
- customer collaboration with respect to the different levels of the recovery process.

Consequently, production plans depend on the estimated quality and supply of the components, numerous regulations and customer preferences. In the same spirit, our contribution attempts to show that a management support system and this type of strategic approach, could work together and be integrated within a manufacturing system.

2.2 Product Lifecycle and Industrial Marketing Management

Some organizations are continuously improving their product lifecycle processes to satisfy customer requirements and environmental regulations by increasing the sustainability of their products. Enterprizes are indeed recognizing the need to develop their green competence in order to implement effective and efficient reverse logistics. From the marketing point of view, this is an opportunity to sell product at a higher price thanks to a green image and meeting social responsibility requirements [4]. This aspect includes an efficient use of raw materials and a reuse of components and materials from the returned products. Reductions in production costs, including energy use, and/or in waste generation [22], and the associated improvement in profitability are a desirable outcome [23]. Accordingly, "Reverse logistics" approaches, are being implemented more and more, and utilized as a competitive advantage in organizations [24].

From the environmental perspective, and to achieve a certain degree of sustainability, several regulations exist to guide organizations in reducing their consumption of non-renewable resources and to decrease the amount of waste materials generated. These include the WEEE, RoHS, and ISO14000.

Environmentally Conscious Manufacturing and Product Recovery (ECMPRO) is a useful approach to integrate environmental concerns into new product development, namely, the design process, material selection, manufacturing and management of the end-of-life products [6]. Some examples are:

- A take-back and recycling program introduced by CISCO to collect and dispose of their end-of-life products, which fully complies with the EU WEEE directive [25]; and
- Products designed by Hewlett-Packard (i.e. printers and laptops) to be environmentally-friendly using recyclable products and reducing the energy employed in production processes, as well as reducing the use of hazardous materials [26].

In general, the different subsystems must cover certain critical areas, from product design, manufacturing and process planning, to delivery and further disposal. In order to achieve a well balanced autonomy and intelligence within a collaborative framework, re-engineering could allow the various modules to cooperate and behave as intelligent social systems.

3 MKMSIS: Multi-agent Knowledge Management System for Industrial Sustainability

We provide a description of our proposed system, based on each department involved in the DN and that participates in a phase of the manufacturing process. Several knowledge management modules should take into account the main requirements associated with their corresponding raw materials and the manufacturing process. All

of the different partial knowledge bases are made available in a collaborative platform for the other departments in the DN. Each system has an optimal product defined as the best product to be manufactured, comprising all of the current restrictions and fulfilling the complete set of requirements. Based on the previous analysis of the materials (bills, availability, etc.), ongoing changing requirements must be attended to and shared among departments in the complete manufacturing and marketing processes. Each sub-system has a set of expert systems driven by a multi-agent system containing a knowledge base valid for the current time. Naturally, several complete rounds in the DN would have to be performed before a final stable state can be achieved.

3.1 Eco-KMS and the Feedback Department Network

By considering an organization as a Feedback Department Network (FDN) that can be regulated in order to accomplish a sustainable performance, we can define a User as a Department, as a distribution associate or as a final client. The Optimal Product Requirement includes definitions from internal organization policies, market trends, external environmental regulations, customer requirements, etc. Figure 1 depicts a FDN.

By dealing with the different departments in a FDN, it can be possible to regulate them by means of the Eco-KMS module (as explained later) and to characterize a Multi-agent Knowledge Management System for Industrial Sustainability (MKMSIS). With this information, an initial sustainability analysis can subsequently be performed, based on the generation and scheduling of the production orders and the most common failures reported by customers.

The first step to reach this goal is to obtain a performance model of a collaborative supply chain, which should involve a complete set of criteria of the main and additional key performance indicators as noted in [27]. Next, the main problems can be collected directly from the FDN and the customers. These problems can then be compared against the expected Optimal Product Requirements. A performance model can be synthesized via an evolutive and/or a fuzzy approach [28, 29] and then tuned more precisely according to the flow of information in the main loop (cf. Fig. 1).

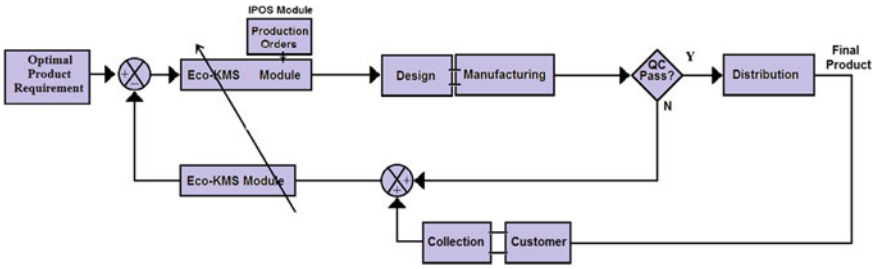

Fig. 1 Feedback Department Network

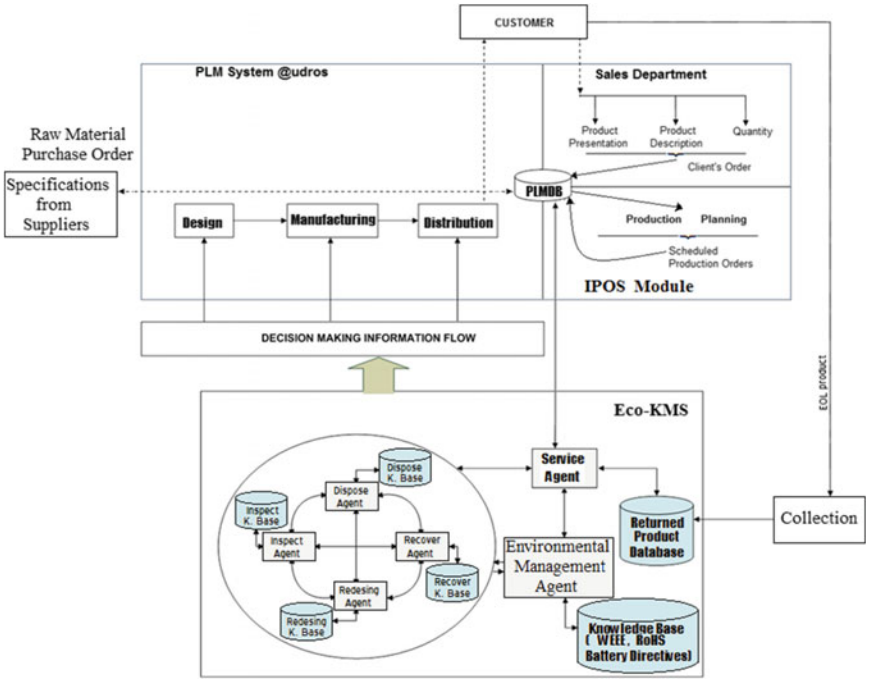

Fig. 2 Organization of the modules in the MKMSIS

The whole system can thus be constantly refined and updated with the flow of information. Different strategies can be applied to fulfill the Optimal Product Requirements. A performance model may involve technical goals, environmental regulations, distribution strategies, etc. based on a forecasted target, or on a concrete standard.

Next, we explain a module built to control or regulate the system. Our proposed FDN is detailed in Fig. 2, where the production process begins with the client's orders from the Sales Department.

The proposed Organization of the Modules in the FDN is the backbone of our MKMSIS and is composed according to scheduled production orders, where an automated system can be used along with Eco-KMS modules based on a Multi Agent System and a set of Expert Systems to manage the knowledge related to regulations. The proposed system will facilitate efficient decision making by constructing the link between the Agent's knowledge of sustainability and the PLM system. With this Network, the impact on the final product can be considered in each stage of the decision-making process and for the whole lifecycle of the product.

In a previous work, a methodology to model a semantic network for a collaborative system was given [30] by defining an intelligent cooperative system. The same procedure was applied here in order to have a basis of entities issued from the application domain, from which a family of sets was synthesized. A network of

elementary and complex concepts linked in a decision tree was finally obtained. In fact, we use ontologies to formalize the knowledge of the production orders and the product's end-of-life; defining each object and its relationship in their corresponding domain.

3.2 Automated Production Orders

The integration of an Intelligent Production Order System[2] (IPOS) will help to increase the quality of production orders. The aim is to decompose the decision-making process conducted by human managers into a series of tasks performed by software agents. These tasks are:

1. Reading the PLM Database to discover newly created sales orders, and to acquire relevant variables;
2. Determining all of the necessary tooling information;
3. Determining the associated machine; and
4. Establishing a sequence for launching production orders by assigning a priority and to feed it to Eco-KMS.

The primary goal of the IPOS module is to generate and schedule production orders. If a new design is furnished according to the Eco-KMS requirements, then IPOS module waits for this new design. If the current design verifies the whole current set of requirements (Optimal Product Requirement), the production orders are transmitted to the design-manufacturing module. Or, Eco-KMS can obtain a new set of instructions via the Quality Control module and the Collection module (see Figs. 1 and 2).

At the IPOS module, the entire communication process ends as soon as the production agent orders updates the PLM database. The production orders can then be put on hold to be sent to the Eco KMS module.

Once the set of requirements are fulfilled and production order is completed with all the information to be employed during the manufacturing execution, the information can be sent to the database and a Service Agent will send it also to the Eco KMS module.

In our case, all the systems' modules are coded in JAVA. The inference engine of Expert Systems (based on inference RULEs) provides two reasoning methods, forward and backward chaining, to solve the inference rules with the current conditions given by the User [20]. The Multi-Agent System coordinates and maintains data consistency all along the process [32]. A methodology for the development and the main required features of expert systems in the area of production planning and scheduling is proposed in [33].

The Eco-KMS Module can manage the knowledge for an efficient decision-making process, wherein the information about the product development and the

[2] An example of such a system can be found in [31].

production process are recovered from the PLM database system. The Eco-KMS module is based on another MAS to be coordinated with the IPOS Multi-Agent System and an agent driving the PLMDB (a centric database).

4 Eco-KMS: A Knowledge Management System

The Eco-KMS is connected to the PLM database where the product information and the production order schedule are stored. It is also connected to the environmental knowledge base, which is shared by agents and by a User, at each stage of the PLM process. Each agent has its own knowledge base that contains the knowledge related to regulations and performance. The Eco-KMS module is described in detail in the following subsections.

4.1 Eco-KMS Agents

The Eco-KMS module includes six agents: Service Agent, Inspect Agent, Recover Agent, Dispose Agent, Redesign Agent and Environmental Management Agent. Each agent is autonomous however, they communicate and interact to share knowledge and information via the Agent Communication Language (ACL) that complies with FIPA (Foundation for Intelligent Physical Agents) specifications [34]. The ACL provides agents with the means to exchange information and knowledge, and defines the types of messages.

The role(s) and the reasoning process of each agent are described next.

Service Agent: This agent handles tasks involving the interface with the PLM system. It receives some requests from the agents involved and provides them the requested information after extracting it from the technical product database of the PLM system (cf. Fig. 3). It uses a blackboard to keep all the data and information requested from Agents, to be integrated later into a Case Base Reasoning approach. At the beginning of the process, the Service Agent requests the Inspect Agent to inspect a returned product. If the Service Agent is asked for more detailed information about the product's technical data (bill of material, manufacturing range, etc.), it connects to the PLM database to extract the information and sends this information to the Inspect Agent.

Inspect Agent: It receives requests from the Service Agent to inspect returned products, along with the parameters of a particular problem to assess if there are similar problems in its knowledge base, using the Case-Based Reasoning (CBR) approach [35, 36]. After the inspection is made by a human team and when more detailed data and information are needed, this will be requested to the Service Agent. A number of rounds are necessary to terminate this process and close down this communication. Since the inspection process is the most important step for classifying the end-of-life of a product (repaired, recycled or redesigned), to complete this task the

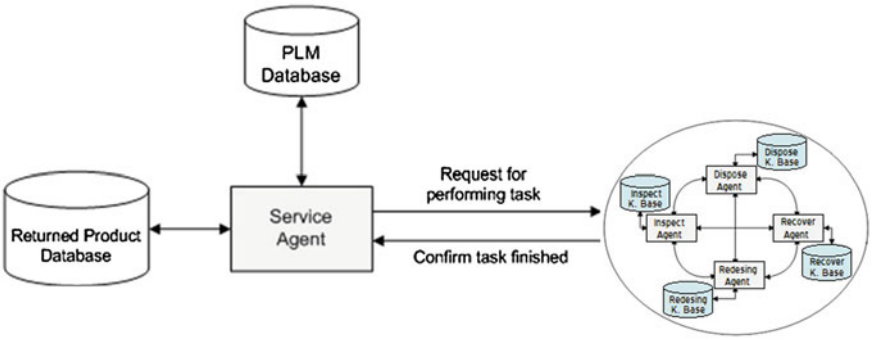

Fig. 3 The interaction between agents and databases

Inspect Agent needs supporting information from the Recover and Dispose Agents. Inspect Agent sends the requested information together with the product number, the component number, and the number of the step involved in the current production order. The expertise of different agents, and their sharing of information and specialized knowledge, allows the system to find the proper solution for each analyzed product. When the product is to be repaired or redesigned and based on CBR, the Inspect Agent sends the Redesign Agent some rules and solutions to manage the analyzed product. The Inspect Agent needs to know the environmental regulations to support its inspection process. It sends a "REQUEST" message with the product information (such as product type, return reason, product's destination, etc.) to the Environmental Management Agent (EM), asking the EM to pass this message to the Service Agent, and waits for the results from the EM Agent. This product information will be the facts that the EM Agent and the inference engine will utilize.

Recover Agent: This agent has a twofold objective: to deal with products to be repaired and with products to be recycled. Thus, after receiving a message from the Inspect Agent specifying that a product needs to be repaired or recycled, the Recover Agent looks into its knowledge base which contains the rules on how to repair those products, the environmental legislations and any internal directives. Once the solution is fully characterized, the Recover Agent sends the list of materials and the average amount of resources employed in the reparation or recycling process associated with the components (as previously specified by the Inspect Agent) to the Repairing or Recycling Department (cf. Fig. 6).

Dispose Agent: Its knowledge base contains information based on the different regulations, ex., from the RoHS and the WEEE, Battery directives, etc., and on how to eliminate hazardous materials and/or reuse materials contained in the analyzed product. This agent helps the FDN to reduce the use of resources and energy by reusing materials or parts for the second life of a product. When a product destination and its treatment have been fully characterized, it sends this information to the Disposal Department (cf. Fig. 6). Whenever a particular case has not been previously treated, the Dispose Agent asks the Environmental Management Agent for more information on environmental regulations or for a new procedure.

Redesign Agent: Once the Inspect Agent has analyzed and determined that the returned product needs to be redesigned so that it will meet the Optimal Product Requirements, some solutions are proposed by the Redesign Agent. Making use of its regulation knowledge base, this agent takes into account the impacts of different processes related to the product based on the bill of material, rate routing, etc., to inform its suggested solutions. When many components are involved in the same failure (based, for example, on statistical analysis), the Redesign Agent may propose a change of the product design to utilize modular components or to minimize the number of parts and simplify the disassembly process, thereby helping to save time, costs and energy in the recycling process phase. The Redesign Agent sends all the solutions to the User in the design module of the PLM system (see Fig. 2).

Environmental Management Agent: The EM Agent receives the requested messages from other agents. This agent is driven by an expert system which has a knowledge base containing the facts and rules related to environmental regulations such as WEEE, RoHS and battery directives, for example. When agents send a request for information about materials, procedures, a process or environmental regulations and that information has not already been classified, a new procedure is created. This new procedure triggers several commands and alerts the involved departments (human teams) to deal with this contingency. After running its inference engine, the EM Agent sends the requested information back to the appropriate agent(s) as well as to the Service Agent to feed the CBR database.

Figure 4 illustrates the interaction between agents (Inspect Agent and Environmental Management (EM) Agent) and the environmental knowledge base.

The items in Table 1 give an example of the rules in the EM Agent's knowledge base. The EM Agent's expert system contains the environmental domain knowledge

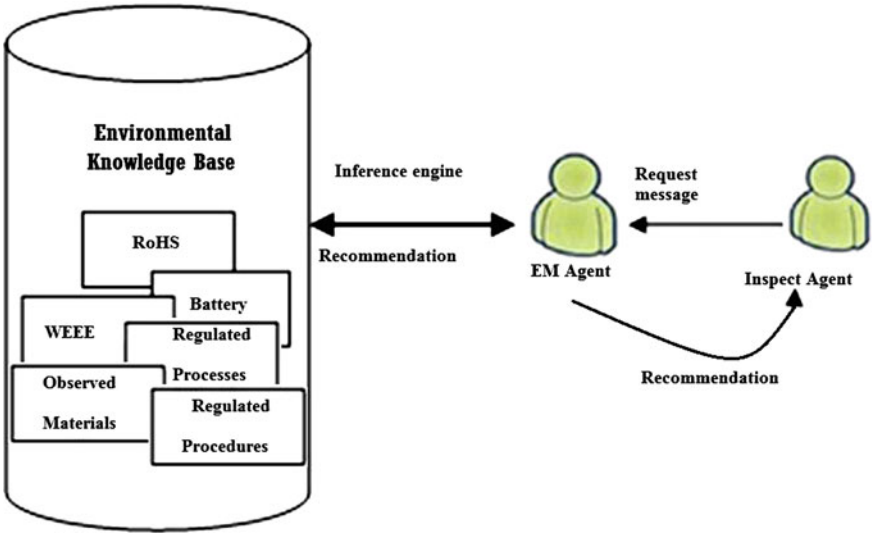

Fig. 4 Example of EM Agent interactions

Table 1 An excerpt of an EM Agent's Rule-based system for WEEE directives

Rules	WEEE directives
Rule 1	IF product depends on electric or electromagnetic fields THEN product is EEE product
Rule 2	IF product does not depend on electric nor electromagnetic fields THEN product is non-EEE product
Rule 3	IF EEE products falls under categories 3 or 4 and EEE product is mobile phone and mobile phone destination is recycle THEN the rate of recovery shall be increased to a minimum of 75 % by an average weight per appliance and component, material and substance reuse and recycling shall be increased to a minimum of 65 % by an average weight per appliance $WEEEdirectiveArt7AnnexIA$
Rule 4	IF LCD is component of mobile phone and LCD has surface > 100 cm^2, THEN LCD with surface > 100 cm^2 must be removed from separated collection of waste mobile phone $WEEEdirectiveAnnexII(1)$

related to the analyzed products. In our system, the knowledge is represented as a set of rules based on environmental regulations, materials, procedures and processes. Each rule specifies a relation and a recommendation or solution, and has an If/Then structure. When the condition parts of a rule are satisfied, the rule is fired and the action part is executed. Its inference engine carries out the reasoning to reach a solution.

Ontologies are used to formalize this knowledge. As an example, Fig. 5 depicts a part of the semantic of an electronic product's end-of-life; based on different environmental regulations (v. gr. WEEE, RoHS, Battery directives, etc.). The EM Agent knowledge base mainly contains information related to the environmental performance (based on the performance model) and regulations (for an excerpt cf. Table 2).

Our proposed system maintains close communication with each department (User) involved in the FDN. Several knowledge management modules have a relationship with the main requirements, environmental regulations, materials, regulated processes and procedures.

5 Mobile Phone End-of-Life Case Study

In order to validate a part of our proposed system, an industrial case study is analyzed concerning the lifecycle of a mobile phone (MP), the Nokia-N95. The technical data for this product are stored in a PLM database. The Multi-Agent System's main interest is highlighted in this case study; the proposed MKMSIS allows the structuring of the distributed knowledge bases (environmental regulations and norms) and the reasoning modules based on the different phases of the product lifecycle and the reverse logistic process. The Multi-Agent System also manages the multiple interactions

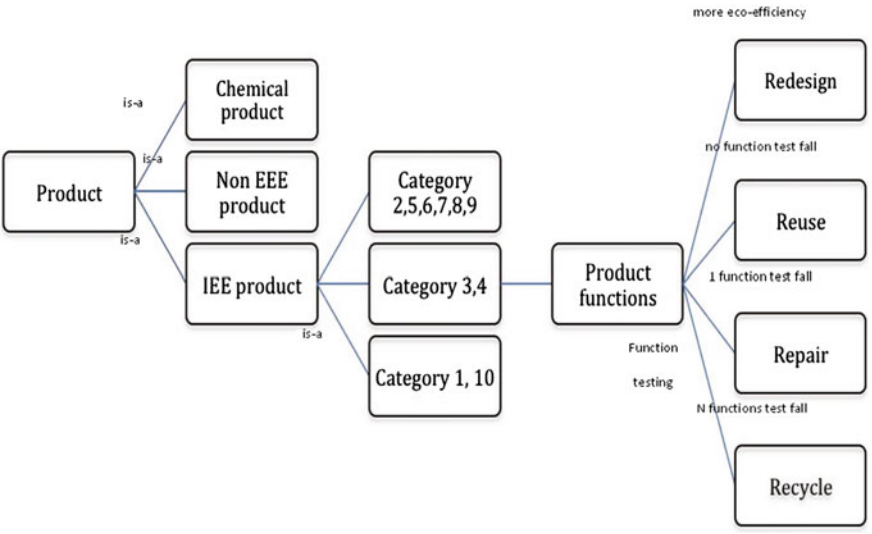

Fig. 5 Example of electronic product domain ontology

Table 2 An excerpt of an EM Agent's Rule-based system for WEEE directives

Rules	EM Agent: Environmental regulation rules
Rule 1	IF product depends on electronic in order to work properly THEN Product is an EEE Product
Rule 2	IF EEE product's return reason is "end-of-life" THEN the priority of waste should devote to reuse, recovery and then recycle, and producers should integrate material recovery in new equipment (WEEE directive para. 18)
Rule 3	IF EEE products fall under category and returned product's destination is "recycle" THEN the rate of recovery shall be increased to a minimum of 75 % by an average weight per appliance and component, and material and substance reuse and recycling shall be increased to a minimum of 65 % by an average weight per appliance (WEEE directive Art7. (Annex IA))
Rule 4	IF the mobile phone's functionality depends on electricity and the mobile phone falls under category 3 THEN mobile phone is an EEE product

between the actors of the system and the information systems in the Organization. In our scenario, each returned MP is inspected by a worker in the company. To analyze the returned product, some information are collected from the customer; mainly about their perception and their opinions of the functionalities, the design, the quality, etc., of the corresponding product. The results of the inspection are then transmitted to the Inspect Agent using the web application interface. The Inspect Agent also

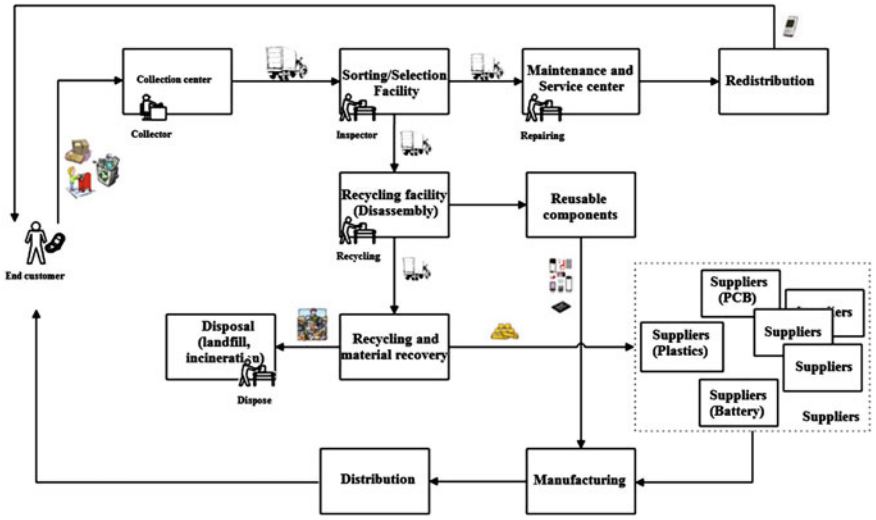

Fig. 6 Supply chain and reverse logistics of a mobile phone

receives some technical information such as the MP model (family), bill of material (BOM) or the manufacturing range from the Service Agent (after a request in the PLM system). The inspection process requires some environmental knowledge, e.g. the WEEE directive, which is contained in the environmental knowledge base.

In Fig. 6 it is shown the supply chain of the mobile phone including the return process of the unused mobile phones. This supply chain is composed by different Departments or Users involved in the product lifecycle process (supply, production, maintenance, distribution, etc.). The main actors of these processes are: designers, collectors, manufacturers, distributors, etc. In the proposed example, agents interact with Users working at different stages of the lifecycle (design, manufacturing, recycling, distribution and recollection). The agents of the proposed Multi-Agent System are implemented in the local system (computers, mobile applications, etc.) of each department, and interact amongst themselves using the PLM information system and Web applications. We have developed some services to facilitate the interactions among the Organization actors.

At the mobile phone sorting phase, an Inspect Agent provides information and guidelines for inspecting the mobile phone. This information is used for decision-making, to decide if a mobile phone should be recycled or not. The Recover Agent gives the recommendations to carry out: repair, replacement of damaged parts, or dismantling and recycling.

In order to determine if the Nokia-N95 should be reused, repaired or recycled for material recovery, a mobile phone function test requires the User to apply it. This can be done when the microphone and speaker are functional, but the LCD screen, the keypad and the battery are not functional. The Eco-KMS provides some recommendations to the User to help them to finalize the destination of each Nokia-

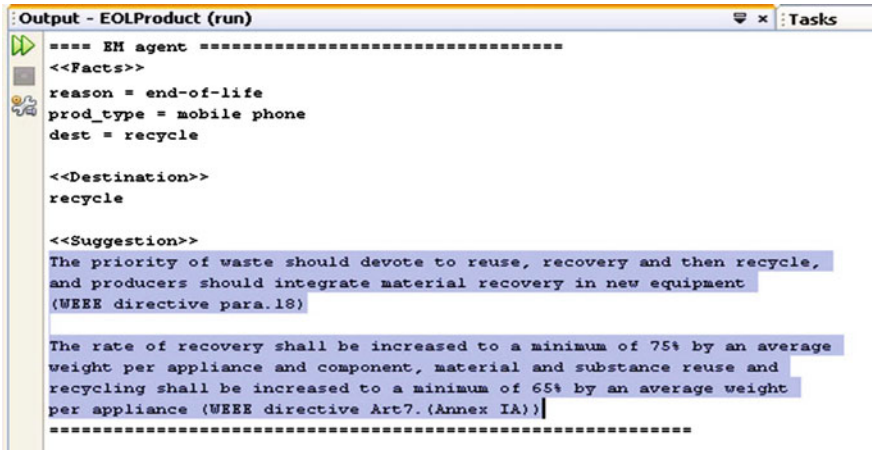

Fig. 7 Some recommendations of an EM Agent

N95. As an example, the recommendation from the EM Agent in Fig. 7 is that "The rate of recovery shall be increased to a minimum of 75 % by …", which can support Users who are concerned about the environmental impact and the economic benefit of their end-of-life mobile phone. At the end of the recycling phase of the mobile phone, the Dispose Agent proposes some recommendations for recycling the battery, based on its knowledge base. In addition, the Dispose Agent receives the recommendations from the EM Agent. These recommendations help users to handle and recycle the mobile phone battery in a way that improves the environmental performance and complies with the European Battery directive (in this case). This case study illustrates a part of the complete system proposed here.

6 Discussion

Sustainable development and reverse logistics applied to Green Manufacturing have been gaining more and more attention from companies and researchers over the last decade. Green Manufacturing becomes a credible strategy when marketers can use local materials, or materials that can be reused or recycled (low competitive advantage, high sustainability advantage). A Green Manufacturing approach can enhance recycling efforts as part of the firm's sustainability program [10].

Whether sustainable/green supply chains can be integrated with green industrial marketing in building greener organizations and industrial brands is still unclear.

In order to create a competitive edge in the marketplace, an open question for industrial organizations is the use of both supply chain sustainability and green industrial marketing. Green industrial branding could be an important industrial marketing effort in conveying the capability of sustainability.

The key research question of our work is focused on how to coherently integrate green manufacturing based on sustainable development and reverse logistics as a whole, combined with sustainable supply chain management. Our main objective states that, green customer's needs can be better met from both the demand and the supply sides.

7 Conclusions and Future Work

Several conditions based on Manufacturing, Marketing, Internal Policies or Directives, Norms and environmental regulations shared on a B2B or B2C networks are correlated with the use and reuse of materials and substances in products and production processes. Consumer awareness of environmental issues is increasing, changing their behaviors in terms of purchases and their returning of used products. Organizations therefore need to improve their products in terms of time, cost and quality, pursuing a high sustainability performance. A Collaborative and Intelligent Knowledge Management System for Industrial Sustainability, based on Multi-Agent systems, has been proposed here, with the aim of obtaining the best of product based on product knowledge, costs, safety, and environmental regulations, including taking advantage of reverse logistics activities. Environmental restrictions, such as the WEEE, RoHS and ISO14000 directives encourage organizations to take responsibility by producing sustainable products. Sharing this regulation knowledge among the different users of our system at every stage of the product lifecycle process improves the efficiency of managing products through to their end-of-life. By using the knowledge-based theory and the Multi-Agent System as a foundation, this study examines how to help managers make the best decisions in the different phases of the product lifecycle.

We have tested our system on two case studies, but only presented the lifecycle of a specific mobile phone (Nokia-N95). In both case studies, recommendations concerning the environment are given by the MKMSIS to Users, who take them as a guideline with which to deal with the product's end-of-life. In this manner, Users get the benefits of a better product, the environment is protected and the product complies with the regulations' requirements. The Feedback Department Network proposed here is a first step toward the integration of a scheduled production, from the client's order to the redesign of the entire product process so that it will comply with sustainable and more environmentally friendly product regulations. Future works will focus on extending the implementation to a larger industrial case.

Acknowledgments The authors would like to express their gratitude to the Editors and their team, and to the anonymous reviewers for their very useful comments that we used to improve this paper. Thanks are also due to Mr. John Harbison for improving the writing style.

References

1. Hanson, C.: What are the best methods for increasing diversity at a digital marketing company? DMNews 1 (2009)
2. López-Morales, V., López-Ortega, O.: Direct marketing based on a distributed intelligent system, 1st edn. Studies in Fuzziness and Soft Computing, vol. 258, pp. 223–239. Springer, Berlin Heidelberg (2010) doi:10.1007/978-3-642-15606-9-17
3. Harrison, M., Hague, P., Hague, N.: Why is business-to-business marketing special?. White paper of B2B International in Market Research with Intelligence, pp. 1–12. B2B International, Manchester (2011)
4. Lee, C., Lam, J.: Managing reverse logistics to enhance sustainability of industrial marketing. Ind. Mark. Manage. **41**, 589–598 (2012)
5. Rogers, D.S.: Tibben-Lembke: an examination of reverse logistics practices. J. Bus. Logistics **22**, 129–148 (2001)
6. Gungor, A., Gupta, S.M.: Issues in environmentally conscious manufacturing and product recovery: a survey. Comput. Ind. Eng. **36**, 811–853 (1999)
7. Polonsky, M., Rosenberger III, P.: Re-evaluating to green marketing - an integrated approach. Bus. Horiz. **44**, 21–30 (2001)
8. Ashley, S.: Designing for the environment. Mech. Eng. **115**, 52–55 (1993)
9. Ouzrout, Y., Geryville, H., Bouras, A., Sapidis, N.: A product information and knowledge exchange framework: a multiple viewpoints approach. J. Prod. Lifecycle Manage. (IJPLM) **4**, 270–289 (2009)
10. Sharma, A., Iyer, G.R.: Resource-constrained product development: implications for green marketing and green supply chains. Ind. Mark. Manage. **41**, 599–608 (2012)
11. Raza, M.B., Kirkham, T., Harrison, R., Reul, Q.: Knowledge based flexible and integrated PLM system at Ford. J. Inf. Syst. Manage. **1**, 66–74 (2011)
12. Bauer, B., Odell, J.: Uml 2.0 and agents: how to build agent based systems with the new uml standard. Eng. Appl. Artif. Intell. **18**, 141–157 (2005)
13. Sandholm, T., Levine, D., Concordia, M.P.M., Hughes, R., Jacobs, J., Begg, D.: Changing the game in strategic sourcing at procter & gamble: expressive competition enabled by optimization. Interfaces **36**, 55–68 (2006)
14. Chan, H., He, H., Wang, W.: Green marketing and its impact on supply chain management in industrial markets. Ind. Mark. Manage. **41**, 557–562 (2012)
15. Menon, A., Menon, A.: Enviropreneurial marketing strategy: the emergence of corporate environmentalism as market strategy. J. Mark. **61**, 51–67 (1997)
16. Bond, A., Gasser, L.E.: Readings in distributed artificial intelligence, Morgan-Kaufmann 1 (1988)
17. Doyle, J., Brown, T.D.: Strategic directions in artificial intelligence. ACM Comput. Surv. **28**, 651–670 (1996)
18. Luck, M., McBurney, P., Shehory, O., Willmott, S.: Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing). AgentLink (2005)
19. Monostori, L., Vncza, J., Kumara, S.R.T.: Agent-based systems for manufacturing. Cirp Ann. Manufact. Technol. **55**, 697–720 (2006)
20. Joseph, P.B., Jennifer, B.: Constructing intelligent agents using Java. Wiley, NY (2001)
21. Van Hillesgersberg, J., Zuidwijk, R., Van Nunen, J., Van Eijk, D.: Supporting return flows in the supply chain. Commun. ACM **44**, 71–81 (2001)
22. Ayres, R., Ferrer, G., Van Leynseele, T.: Eco-efficiency, asset recovery and remanufacturing. Eur. Manage. J. **15**, 557–574 (1997)
23. Giuntini, R., Andel, T.: Master the six r's of reverse logistics. Trans. Distrib. **36**, 93–98 (1975)
24. Marien, E.J.: Reverse Logistics as competitive strategy. SCM Review (1998)
25. SYSTEMS, C.: Cisco takeback and recycle program. CISCO Documents WEEE 1 (2008)
26. Packard, H.: Designing. HP, http://www.hp.com/hpinfo/globalcitizenship/environment/productdesign/design.html (2011)

27. Gunasekaran, A., Kobu, B.: Performance measures and metrics in logistics and supply chain management: a review of recent literature (1995–2004) for research and applications. Int. J. Prod. Res. **45**, 2819–2840 (2007)
28. Maertens, K., De Baerdemaeker, J., Babuska, R.: Genetic polynomial regression as input selection algorithm for non-linear identification. Soft Comput. **10**, 785–795 (2006)
29. Derrouiche, R., Holimchayachotikul, P., Leksakul, K.: Predictive performance model in collaborative supply chain using decision tree and clustering technique. In: Proceedings of 4th International Conference on Logistics (LOGISTIQUA), pp. 412–417 (2011)
30. López-Morales, V., López-Ortega, O.: A distributed semantic network model for a collaborative intelligent system. Int. J. Intell. Manuf. **16**, 515–525 (2005)
31. López-Ortega, O., López-Morales, V., Villar-Medina, I.: Intelligent and collaborative multi-agent system to generate and schedule production orders. J. Intell. Manuf. **19**, 677–687 (2008)
32. TiLAB: JADE- Java Agent Development. TILAB, http://jade.tilab.com/ (2011)
33. Metaxiotis, K., Askounis, D., Psarras, J.: Expert systems technology in production planning and scheduling. Intell. Knowl.-Based Syst. **03**, 797–817 (2005)
34. FIPA: FIPA SPECS, http://www.fipa.org/specs/fipa00061/SC00061G.pdf (2002)
35. Tung, Y., Tseng, S., Wenga, J., Lee, T., Liao, A.Y.H., Tsai, W.: A rule-based cbr approach for expert finding and problem diagnosis. Expert Syst. Appl. **37**, 2427–2483 (2010)
36. Guo, Y., Hu, J., Peng, Y.: Research on cbr system based on data mining. Appl. Soft Comput. **11**, 5006–5014 (2011)